

UCLA

UCLA Electronic Theses and Dissertations

Title

Numerical Subdivision Surfaces for Simulation and Data Driven Modeling of Woven Cloth

Permalink

<https://escholarship.org/uc/item/30g0h9r5>

Author

Clyde, David

Publication Date

2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Numerical Subdivision Surfaces for Simulation and
Data Driven Modeling of Woven Cloth

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

David Corwin Clyde

2017

© Copyright by
David Corwin Clyde
2017

ABSTRACT OF THE DISSERTATION

Numerical Subdivision Surfaces for Simulation and
Data Driven Modeling of Woven Cloth

by

David Corwin Clyde

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2017

Professor Joseph M. Teran, Chair

We present the derivation details necessary for simulation of thin shells with finite strains based on the Kirchhoff-Love assumptions. With an eye towards cloth simulation, we combine this with a nonlinear orthotropic constitutive model framework. We leverage a conforming spatial discretization using Catmull-Clark subdivision surfaces to ensure convergence under refinement, which we confirm by numerical experiments. The dynamics are handled in a fully implicit fashion to allow for large timesteps and solution of quasistatic problems.

Accurate constitutive modeling and parameter estimation for woven fabrics is essential in many fields. To achieve this we first design an experimental protocol for characterizing real fabrics based on commercially available tests. Next, we present a new orthotropic hyperelastic constitutive model for woven fabrics. Finally, we create a method for accurately fitting the material parameters to the experimental data. The last step is accomplished by solving inverse problems using our Catmull-Clark subdivision finite element discretization of the Kirchhoff-Love equations. Through this approach we are able to reproduce the fully nonlinear behavior corresponding to the captured data with a small number of parameters while maintaining all fundamental invariants from continuum mechanics. The resulting constitutive model can be used with any discretization and not just subdivision finite elements, which we demonstrate by providing an alternate implementation based on simple triangle meshes. We illustrate the entire process with results for five types of fabric and compare

photo reference of the real fabrics to the simulated equivalents.

The dissertation of David Corwin Clyde is approved.

Luminita Vese

Rasmus Tamstorf

Jeffrey Eldredge

Christopher Anderson

Joseph M. Teran, Committee Chair

University of California, Los Angeles

2017

TABLE OF CONTENTS

1	Introduction	1
1.1	Subdivision surface based cloth simulation	1
1.2	Data driven modeling for woven materials	3
1.3	Related Work	5
1.3.1	Subdivision surface based cloth simulation	5
1.3.2	Data driven modeling for woven materials	6
1.4	Contributions	7
2	Subdivision Surface Based Cloth Simulation	9
2.1	Kirchhoff-Love kinematics	9
2.1.1	Deformation gradient and strain tensor	11
2.2	Hyperelasticity	12
2.2.1	Orthotropy	13
2.2.2	Interpretation of orthotropic invariants	16
2.3	Governing PDE	17
2.3.1	Mass density and kinetic energy	17
2.3.2	Weak form derivation	18
2.3.3	Strong form and boundary terms	21
2.4	Spatial discretization	24
2.4.1	Catmull-Clark subdivision surfaces	24
2.4.2	Finite element discretization of the weak form PDE	27
2.5	Quadrature	28

2.6	Enforcement of boundary conditions	30
2.7	Implicit integration	32
2.8	Linear solver	35
2.8.1	Alternative: Schur complement reduction	36
2.8.2	Alternative: Nullspace methods	36
2.9	Derivatives of energy density	38
2.10	Evaluation of subdivision surfaces	40
2.11	Implementation	41
2.11.1	Precomputation	41
2.11.2	Simulation loop	42
2.11.3	Code structure for energy density computation	44
2.11.4	Optimizing low-level functions	46
2.11.5	Avoiding Hessian computations	46
2.12	Convergence under mesh refinement	47
3	Data Driven Modeling for Woven Materials	52
3.1	Experimental design	52
3.2	Constitutive model	55
3.2.1	Parameterization and derivatives	56
3.2.2	Comparison to an existing model	57
3.2.3	Extrapolation	59
3.2.4	Alternative discretization using triangle mesh	60
3.3	Fitting strategy	62
3.3.1	Optimization objective	63
3.3.2	Fitting in stages	64

3.3.3	Choice of degree parameters	66
3.4	Initial guesses for fitting solves	67
3.4.1	Power balance	69
3.4.2	Potential energy as a function of stretch	71
3.4.3	Implementation and predictive accuracy	72
3.5	Main fitting solver	74
3.5.1	Equilibrium configuration	75
3.5.2	Differentiating the equilibrium configuration	78
3.5.3	Multiresolution meshing and curve sampling	81
3.5.4	BFGS vs. Gauss-Newton	82
3.5.5	BFGS implementation	82
3.5.6	Multi-valued objective	83
3.5.7	Step size management	85
3.6	Results	87
3.7	Limitations and future work	93
References	97

LIST OF FIGURES

2.1	Side view of a Kirchhoff-Love plate	9
2.2	Visual summary of the 2D and 3D cloth spaces and maps	10
2.3	Demonstration of 2D strain penalizing bend deformations	16
2.4	Decomposition of parameter space boundary into components with horizontal or vertical normals	23
2.5	Smoothing a 3D mesh with Catmull-Clark subdivision	24
2.6	Catmull-Clark subdivision on an irregular 2D mesh	26
2.7	Direction vectors for normal direction constraint formulation	30
2.8	Test setups for convergence under refinement	47
2.9	Plots of convergence under mesh refinement in four norms	49
3.1	Comparison between simulated drape experiments and photographs of real cloth	52
3.2	Experimental test setups	53
3.3	Unsuitability of the Itskov model for woven cloth	58
3.4	Approximate kinematics for the warp stretch test	67
3.5	Approximate kinematics for the bias stretch test	68
3.6	Finite element equilibria for stretch tests	73
3.7	Experimental and fitted force-elongation curves for tensile tests of five fabrics	89
3.8	Demonstration of the improvement of fit quality as model complexity increases	90
3.9	Relative error plots for tensile tests at 22.5° , 45° , and 67.5°	90
3.10	Picture frame test with fitted materials	92
3.11	Corner pull simulation with fitted materials	93
3.12	Collision example with fitted materials using triangle mesh discretization . . .	94

LIST OF TABLES

2.1	Orders of convergence under mesh refinement	50
3.1	Complete list of parameters for determining cloth behavior in our model	60
3.2	Measured properties of the test materials	87
3.3	Final fitted parameter sets for the five test materials	88
3.4	Average relative error of the fitted models vs. the experimental data	89
3.5	Bend test drape angles for the fitted models	91
3.6	Average relative error of fitted models on the validation test	91

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Joseph Teran, who kept me at UCLA by showing me how exciting applied math can be. I thank Rasmus Tamstorf, my mentor and manager through several internships at Disney Animation, for wisdom shared through near-daily meetings and emails and attention far beyond what was required. Together you made this project possible - and more importantly, you taught me how to be a scientist.

Special thanks go to Colin Eckart and particularly Norman Joseph of Walt Disney Animation Studios, who provided final rendering of simulated examples.

This dissertation includes content from the previously published work [CTT17], available online at <https://doi.org/10.1145/3099564.3099577>.

During my research, I was partially supported by the following grants: NSF (CCF-1422795), ONR (N000141110719, N000141210834), DOD (W81XWH-15-1-0147) as well as a gift from Disney Research.

VITA

- 2011 B.S. (Mathematics), University of Michigan, Ann Arbor.
- 2012-2017 Teaching Assistant, Mathematics Department, UCLA
- 2011-2017 Research Assistant, Mathematics Department, UCLA
- 2015-2017 Research Intern, Walt Disney Animation Studios

PUBLICATIONS

D. Clyde, J. Teran, R. Tamstorf. “Modeling and Data-Driven Parameter Estimation for Woven Fabrics.” In ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 2017

CHAPTER 1

Introduction

Simulation of cloth is an area with broad-ranging applications, from the textile industry to entertainment to engineering applications such as fiber reinforcement in composites. However, numerical cloth simulation software faces several challenges which can make it cumbersome to achieve accurate or even visually attractive simulations. Most existing simulators rely on a triangle mesh cloth discretization, which greatly limits convergence properties and also requires a conversion process since cloth models are usually created as smooth subdivision or NURBS surfaces. We derive and implement a cloth simulation scheme which discretizes the Kirchhoff-Love shell kinematics using subdivision surfaces, thus enabling isogeometric simulation for applications such as animated movies while maintaining strong convergence properties. Arguably a more important difficulty lies in choosing an appropriate constitutive model together with the attendant material parameters. For graphics applications, this is typically accomplished by trial-and-error, which is time consuming and inconsistent. We address the problem by presenting a versatile nonlinear constitutive law together with a data-driven technique which obtains all necessary parameters based on experimental data from commercially available test procedures. The parameter fitting process benefits from the accuracy of the subdivision surface cloth model, but the model with its final fitted parameters is also adapted for the triangle mesh discretization to enable easier adoption.

1.1 Subdivision surface based cloth simulation

Chapter Two of the thesis provides a detailed derivation and implementation guide for thin shell simulation based on the Kirchhoff-Love kinematic assumptions together with an arbi-

trary orthotropic hyperelastic energy.

Most physics-based cloth simulation techniques can be derived from the Kirchhoff-Love thin shell equations with varying levels of approximation accuracy [GGR06, Zor05, ZPA17]. We state the Kirchhoff-Love kinematics in §2.1; then in §2.2 and §2.3 we apply the Kirchhoff-Love theory to a general hyperelastic material to derive the implied PDE in weak and strong forms. However, direct discretization of the resulting weak form PDE requires a finite element basis which is at least H^2 -smooth.

Historically, the H^2 regularity requirement is a key obstacle for finite element approaches, and is not easily resolved by for example storing additional positions and derivatives per element [COS00]. Instead, typical simulation techniques perform a series of approximation steps to enable simulation using triangle mesh finite elements, but in the process, convergence rate is greatly reduced or sacrificed entirely. Instead, we introduce in §2.4 a conforming finite element method using Catmull-Clark subdivision surfaces to satisfy the H^2 regularity requirement and preserve convergence behavior. The use of Catmull-Clark surfaces constitutes a particular advantage for application in the animated movie industry, where subdivision surfaces are the de facto standard for building and rendering surfaces. Use of the same geometry for simulation is termed isogeometric analysis ([HCB05]) and saves several geometry conversion steps.

Several subtle theoretical details must be addressed to obtain an effective simulator. It is necessary to choose an appropriate integral quadrature rule (§2.5), and the non-interpolating nature of the subdivision surface finite elements necessitates special care in enforcement of Dirichlet boundary conditions (§2.6). To enable simulation of large time steps, in §2.7 we apply backward Euler to achieve implicit time integration, together with a minimization solver formulation which improves efficiency and robustness despite the ill-conditioning commonly present in cloth problems. The poor conditioning means that both accuracy and efficiency are key concerns when selecting a linear solver; that choice is analyzed in §2.8.

Many existing H^2 discretizations of the Kirchhoff-Love theory incorporate a simplified constitutive model directly into their implementations [COS00, CO01, Gre03, WHP11].

However, we provide a simulation framework which allows relatively easy plug-and-play modification of the constitutive model. This is achieved by constructing the hyperelastic energy using a multi-layer chain rule, in which only the top layer must be reimplemented for each new energy. The necessary differentiation formulas are obtained in §2.9 and §2.10, and §2.11 provides a detailed implementation guide for the energy computation.

Finally, Chapter Two of the thesis concludes in §2.12 with a convergence study. The convergence under refinement property of our implementation is confirmed in a variety of norms and for several tests.

1.2 Data driven modeling for woven materials

In Chapter Three of the thesis we address the data-driven cloth modeling problem.

For cloth simulation in engineering and also for e-commerce applications such as virtual try-on, predictive power is of paramount significance. By contrast, absolute accuracy is typically a lower priority in entertainment. For graphics applications, it is usually up to an artist to come up with a reasonable set of simulation parameters. Unfortunately, different artists often end up with different parameters for the same materials and since the approach is based on trial-and-error it can be difficult to reach satisfactory results. The desire for easier workflows when setting up simulations and also higher accuracy has motivated many researchers to look for experimental ways to determine simulation parameters. The appeal of such data-driven approaches is clear as it potentially allows for the creation of libraries of material parameters which can be used repeatedly and augmented as necessary.

In order for a data-driven approach to work, multiple components must be in place. First, a set of experiments must be designed which exercise all important deformation modes of the material response. From a practical point of view these experiments should be reproducible and since most end-users of cloth simulation are not experts in material testing, it is desirable that the tests are commercially available through a service bureau. Second, a sufficiently versatile, accurate and preferably parsimonious cloth model must be established. Finally, a

robust fitting method needs to be devised to fit the model parameters to the experimentally obtained data. In this dissertation we endeavor to provide all these components.

Much existing work related to characterization of fabrics is based on the Kawabata system, [Kaw80]. However, these systems can be hard to find and are quite expensive. A more recent system, called FAST for “Fabric Assurance by Simple Testing”, is much cheaper but limited to small strain deformations, [Min95]. In the graphics literature alternate methods have been proposed, but most of these require nonstandard equipment [WOR11, MBT12]. The experimental protocol we propose, (§3.1), is based on existing ASTM standards¹ along with extensions which can be implemented using the same instruments as for the standard tests.

There is a plethora of different types of fabrics including knitted fabrics, woven fabrics, and a variety of specialty fabrics such as lace, 3D fabrics, and non-wovens. A single model is unlikely to work well for all these different types of fabric; thus this work will focus on woven fabrics. In order to fit a wide range of woven materials while minimizing mesh dependent behavior, a continuum assumption is used in the mechanical model. Since real clothing typically undergoes large deformations including large strains (especially in the shear component), it is essential to use a fully nonlinear model. To this end we propose a generalized orthotropic model characterized by a small number of parameters (§3.2). The orthotropy reflects the basic symmetry present in most woven fabrics, and by introducing nonlinearity in a way akin to Ogden’s constitutive model we are able to represent highly nonlinear behavior using a single model for the entire strain regime.

Data fitting is conceptually simple, but in practice fraught with peril. By using a fully implicit method for quasistatic simulation and by leveraging an infinitely smooth constitutive model, we are able to provide exact derivatives with respect to the model parameters in our optimization method based on the Broyden-Fletcher-Goldfarb-Shanno method (BFGS). Combined with good initial guesses based on bootstrap estimates this allows us to circumvent some of the data fitting challenges experienced in previous methods. We present our

¹Corresponding ISO standards also exist.

data fitting methodology in §3.3, §3.4, and §3.5. All of this is based on the Catmull-Clark subdivision finite element (FEM) discretization of the Kirchhoff-Love thin shell equations. Although the Kirchhoff-Love thin shell model is rather complex, its continuum basis is essential for providing parameters that depend minimally on mesh resolution and associated discrete anisotropy. However, §3.2.4 has established that the Kirchhoff-Love assumption, as well as the subdivision basis for FEM, while useful for the fitting process, are not necessary in practical simulation; in particular the constitutive model can be used with simple linear strain triangles and commonly used graphics approaches for bending, e.g. [BMF03, GHD03].

1.3 Related Work

1.3.1 Subdivision surface based cloth simulation

Cloth simulation is a vast topic in both graphics and engineering. Here we discuss just some of the work most related to ours. Recent work has proposed yarn-level simulations as a way to achieve accurate and very detailed results [KJM08, CLM14]. These methods can create beautiful details, but require millions if not billions of degrees of freedom to represent typical garments. By adopting a continuum mechanics approach we aim to capture the cloth behavior at a tiny fraction of this cost. A middle-ground between the two approaches is the mesostructurally-based continuum model presented in [KJS05]. This can capture yarn-level effects, but consequently also requires experimental data that characterizes the fabric at that level.

The method presented here is an example of “isogeometric analysis” (IGA), where subdivision surfaces are used to represent both the geometry (of cloth) and the basis functions for finite element analysis. This concept has been credited to [HCB05], but subdivision finite elements were originally introduced by [COS00], where they were applied to linear elastic Kirchhoff-Love thin shells. The method was later extended to finite strain analysis in [CO01], where they considered a Neo-Hookean material but used explicit integration for the temporal evolution. Both [COS00] and [CO01] are based on the Loop subdivision scheme. In graphics,

[TWS06] combined the subdivision finite elements with a co-rotational linear elasticity model and introduced implicit time integration using the backward Euler scheme. More recently, [LBC12] considered shear deformable shells, but limited their discussion to quasistatic problems. The work by [VSJ13] investigates growth problems using the St. Venant-Kirchhoff material model along with an explicit Newmark predictor-corrector integration scheme. All of this prior work is based on Loop subdivision on triangle meshes.

[WHP11] considers the Catmull-Clark subdivision scheme with linear elasticity as an alternative to Loop subdivision, with a variety of applications that do not involve dynamics. We extend that work through the use of a fully implicit time integration scheme and a nonlinear constitutive model. A similar extension based on B-splines has been presented by [KHW15].

In this work we limit ourselves to elastic deformations, but it is well-known that cloth exhibits significant amounts of hysteresis due to internal friction [Wil10, MTB13]. Additionally we do not consider rate-dependent behavior as they do in [RCY07], but we believe that leveraging these ideas is an important area of future work.

1.3.2 Data driven modeling for woven materials

The estimation of clothing simulation parameters from real fabric deformation data has been investigated in recent years. Lubile and Magnenat-Thalmann [LM08] compared the Kawabata system (KES) to the FAST system but found that they both have limitations. More recently [Pow13] compared results from the FAST system to a newer Fabric Testing Kit (FTK) from Browzwear, but still found limitations. Despite these limitations the work by [MLV07] uses the stress-strain curves from KES to drive an “accurate particle system”. This approach was further elaborated upon in [VMF09]. Bhat et al [BTH03] developed an optimization procedure for estimating clothing simulation parameters directly from video data of moving fabrics. Beyond the already mentioned methods, existing methods from the textile industry for testing the mechanical properties of fabric are surveyed in [WLD08].

A number of papers in graphics have proposed alternative and simple methods for measuring fabric properties. Wang et al. [WOR11] propose a sequence of tensile measurements using small loads (less than 15 N/m) for which they then fit a piecewise linear model. Miguel et al. [MBT12] use a more complicated setup for capturing data which include more complicated examples with shear buckling. However, they also primarily focus on the low strain domain (typically less than 200 N/m). By comparison KES tests up to 500 N/m. While the smallest strain regime is most important for typical cloth simulations, higher strains do occur even in everyday use of garments and are therefore also important to consider. Examples of higher strains occur especially near garment seams where parts of the fabric are physically constrained.

The experimental data of Wang et al. [WOR11] is based on a sparse set of measurements, and thus may not be suitable for fitting a more complex range of strain regimes. The experiments of Miguel et al. [MBT12] rely on complex, hand-constructed machinery, which requires careful manual labor by researchers. Our approach uses a wide range of experimental data from the readily available ASTM standard tests, and we examine our experimental error by repeating experiments with additional samples.

Similar to our approach, Miguel et al. [MMO16] construct a single hyperelastic energy function for fabric. Unlike our approach they do not assume that cloth possesses orthotropic symmetry but instead allow for a more general anisotropic form, which they then fit against a subset of the data from [MBT12]. Other papers in graphics have considered orthotropic hyperelastic energies with a focus on how to make it intuitive for an artist to design and control orthotropic behavior, [LB14, XSZ15]. We believe this is complementary to our work as the goal in animation is often plausibility rather than true reality.

1.4 Contributions

The specific contributions of this thesis may be summarized as follows:

- Detailed derivation a conforming finite element discretization of the Kirchhoff-Love

shell theory using Catmull-Clark subdivision surface finite elements.

- An implementation guide for a robust, convergent cloth simulation framework which allows a variety of constitutive models to be simulated with minimal additional investment.
- Experimental procedures for measuring deformation response of woven materials using readily available [ASTM](#) tests.
- A hyperelastic constitutive model that separates the large strain from small strain behaviors in an intuitive way and thus allows for natural fitting to data.
- Efficient and accurate fitting procedures tailored to our experiment selection and model.

CHAPTER 2

Subdivision Surface Based Cloth Simulation

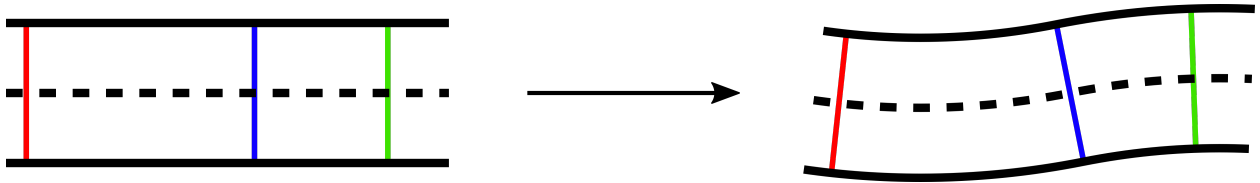


Figure 2.1: *Side view of a Kirchhoff-Love plate in flat rest configuration (left) and in a deformed configuration (right). The midsurface is shown as a dashed line and three surface-normal cross sections are marked in red, blue, and green before and after deformation. The deformed cross sections are lines perpendicular to the midsurface with the same length as the original segments.*

2.1 Kirchhoff-Love kinematics

We model cloth as a deformable solid subject to kinematic constraints imposed by the Kirchhoff-Love theory of thin shells. The Kirchhoff-Love assumptions fully determine a 3D cloth object’s position from a 2D midsurface by assuming that straight lines initially perpendicular to the midsurface remain straight lines perpendicular to the midsurface and retain their length during deformation. A visual representation of the kinematic constraints is provided in Figure 2.1. As shown in [CL96], the Kirchhoff-Love model is asymptotically in agreement with the unconstrained 3D solid mechanics as the shell thickness approaches 0.

In this work, we assume a constant thickness τ per cloth object. We represent the deformed midsurface by a map \mathbf{x} from a 2D parameter space ω to worldspace Ω . The corresponding map describing the volumetric cloth object is denoted by $\mathbf{r} : \omega^\tau \rightarrow \Omega^\tau$, where

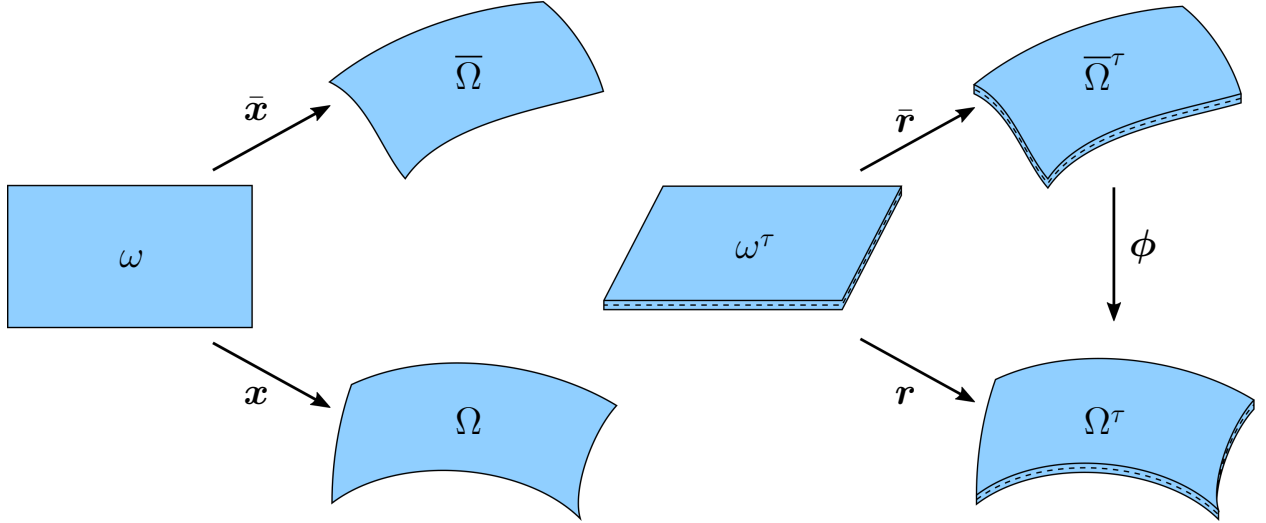


Figure 2.2: Visual summary of the 2D midsurface kinematics (left) and the solid kinematics (right). The midsurface is shown in the 3D shells as a dashed line.

ω^τ is the 3D parameter space, and Ω^τ is the region of \mathbb{R}^3 occupied by the shell. The curvilinear coordinates for a point in the shell are denoted by $\boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3)$, where superscripts indicate contravariant indices (not to be confused with exponentiation). Given these conventions, the Kirchhoff-Love assumptions directly imply

$$\mathbf{r}(\xi^1, \xi^2, \xi^3) = \mathbf{x}(\xi^1, \xi^2) + \xi^3 \mathbf{a}_3(\xi^1, \xi^2), \quad -\frac{\tau}{2} \leq \xi^3 \leq \frac{\tau}{2}, \quad (2.1)$$

where $\mathbf{a}_3(\xi^1, \xi^2)$ is the unit normal to the deformed midsurface. In the following, we use Greek letters for indices in $\{1, 2\}$, lowercase Latin letters for indices in $\{1, 2, 3\}$, and uppercase Latin letters for indices that range from 1 to n , where $n > 2$. Furthermore we use the comma notation to denote partial derivatives with respect to ξ^α such that $\mathbf{x}_{,\alpha} = \partial \mathbf{x} / \partial \xi^\alpha$. For convenience, let $\mathbf{a}_\alpha = \mathbf{x}_{,\alpha}$ denote the covariant basis vectors of the midsurface or, equivalently, the columns of the midsurface mapping Jacobian. We can then write the surface normal as

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\|\mathbf{a}_1 \times \mathbf{a}_2\|}.$$

Also, let \mathbf{g}_i denote the covariant curvilinear basis vectors $\frac{\partial \mathbf{r}}{\partial \xi^i}$, and use \mathbf{g}^i to denote the

corresponding contravariant basis vectors; thus, $\mathbf{g}^i \cdot \mathbf{g}_j = \delta_i^j$ and

$$\begin{pmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 \end{pmatrix} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \xi^3 \mathbf{a}_{3,1} & \xi^3 \mathbf{a}_{3,2} & \mathbf{a}_3 \end{pmatrix} \quad (2.2)$$

Finally, define $\mathbf{G} = \frac{\partial \mathbf{r}^T}{\partial \boldsymbol{\xi}} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}}$ as the covariant metric tensor with entries $g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j$. Throughout this document, we use overbar notation for quantities related to the undeformed configuration. Thus, for example, $\bar{\mathbf{x}} : \omega \rightarrow \bar{\Omega}$ denotes the undeformed midsurface map; analogous definitions are made for $\bar{\mathbf{r}}$, $\bar{\mathbf{a}}_3$, and so on. Figure 2.2 provides a reference for the definitions of the various maps and spaces.

2.1.1 Deformation gradient and strain tensor

The deformation function from the undeformed shell to the deformed shell is given by

$$\boldsymbol{\phi}(\bar{\mathbf{x}}) = \mathbf{r}(\bar{\mathbf{r}}^{-1}(\bar{\mathbf{x}}))$$

and the deformation gradient $\mathbf{F} : \bar{\Omega} \rightarrow \mathbb{R}^{3 \times 3}$ is the Jacobian of $\boldsymbol{\phi}(\bar{\mathbf{x}})$:

$$\mathbf{F} = \frac{\partial \boldsymbol{\phi}}{\partial \bar{\mathbf{x}}} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}} \left(\frac{\partial \bar{\mathbf{r}}}{\partial \boldsymbol{\xi}} \right)^{-1} = \mathbf{g}_i \otimes \bar{\mathbf{g}}^i. \quad (2.3)$$

Note that here, as well as in the following, there is an implied summation over repeated indices. Given the deformation gradient, the Green-Lagrange strain is

$$\begin{aligned} \mathbf{E} &= \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \\ &= \frac{1}{2} ((\bar{\mathbf{g}}^i \otimes \mathbf{g}_i)(\mathbf{g}_j \otimes \bar{\mathbf{g}}^j) - \mathbf{I}) \\ &= \frac{1}{2} (g_{ij} \bar{\mathbf{g}}^i \otimes \bar{\mathbf{g}}^j - \mathbf{I}) \end{aligned} \quad (2.4)$$

The entries g_{ij} satisfy

$$\begin{aligned} g_{\alpha\beta} &= (\mathbf{a}_\alpha + \xi^3 \mathbf{a}_{3,\alpha})^T (\mathbf{a}_\beta + \xi^3 \mathbf{a}_{3,\beta}) \\ g_{\alpha 3} &= 0 \\ g_{33} &= 1, \end{aligned} \tag{2.5}$$

in which $g_{\alpha 3}$ has been simplified using $\mathbf{a}_\alpha \cdot \mathbf{a}_3 = \mathbf{a}_{3,\alpha} \cdot \mathbf{a}_3 = 0$. By using

$$\mathbf{a}_{3,\alpha} = \frac{1}{\|\mathbf{a}_1 \times \mathbf{a}_2\|} (\mathbf{I} - \mathbf{a}_3 \mathbf{a}_3^T) (\mathbf{a}_{1,\alpha} \times \mathbf{a}_2 + \mathbf{a}_1 \times \mathbf{a}_{2,\alpha}), \tag{2.6}$$

we obtain $g_{\alpha\beta}$ from the inputs $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{2,2}, \xi^3$. The analogous formula for $\bar{\mathbf{a}}_{3,\alpha}$ yields $\bar{\mathbf{g}}_j$ and ultimately $\bar{\mathbf{g}}^j$ from $\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2, \bar{\mathbf{a}}_{1,1}, \bar{\mathbf{a}}_{1,2}, \bar{\mathbf{a}}_{2,2}, \xi^3$. We define the column vector $\mathbf{z} = \mathbf{z}(\boldsymbol{\xi}) = \mathbf{z}(\boldsymbol{\xi}; \mathbf{x})$ in \mathbb{R}^{15} as the concatenation of $\mathbf{z}_1 = \mathbf{a}_1, \mathbf{z}_2 = \mathbf{a}_2, \mathbf{z}_3 = \mathbf{a}_{1,1}, \mathbf{z}_4 = \mathbf{a}_{1,2},$ and $\mathbf{z}_5 = \mathbf{a}_{2,2}$; that is,

$$\mathbf{z} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_{1,1} \\ \mathbf{a}_{1,2} \\ \mathbf{a}_{2,2} \end{pmatrix}. \tag{2.7}$$

In the following, it will be convenient to use the notation $\mathbf{E} = \mathbf{E}(\mathbf{z}, \xi^3)$, leaving implicit the dependence on the rest configuration.

2.2 Hyperelasticity

In this work we model cloth objects as hyperelastic solids. That is, we assume the existence of a strain energy density $\psi = \psi^{\mathbf{E}}(\mathbf{E})$ such that stress within the material is given by $\mathbf{S} = \frac{\partial \psi^{\mathbf{E}}}{\partial \mathbf{E}}$. This implies the internal forces are conservative, since ψ can be computed from \mathbf{E} and thus

depends on the undeformed and deformed configurations but not on the specific deformation path. The notation $\psi^{\mathbf{E}}(\mathbf{E})$ is meant to emphasize that this is ψ viewed as a function of \mathbf{E} ; the same energy density ψ could of course also be viewed as a function $\psi^{\mathbf{F}}(\mathbf{F})$, or in terms of various other quantities which will be defined in the following analysis. The stress tensor \mathbf{S} is the second Piola-Kirchhoff stress, which maps material configuration area weighted normal vectors to material configuration forces. \mathbf{S} is related to the Cauchy stress tensor $\boldsymbol{\sigma}$ via the transformation

$$\mathbf{S} = \det(\mathbf{F})\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T}$$

and to the first Piola-Kirchhoff stress via

$$\mathbf{S} = \mathbf{F}^{-1}\mathbf{P}.$$

2.2.1 Orthotropy

A hyperelastic material is orthotropic provided that there exists a curvilinear orthonormal basis in the undeformed space such that the elastic potential is invariant under replacement of \mathbf{E} with $\mathbf{Q}^T\mathbf{E}\mathbf{Q}$ for any element \mathbf{Q} of the orthotropic symmetry group. Specifically, this means that it is invariant under reflection across any of the given basis vectors. The thread directions in woven cloth create a close approximation to orthotropic symmetry, albeit with slight discrepancies introduced by, for example, asymmetric weave patterns such as satins and twills. Thus we describe woven cloth by orthotropic models with orthotropy basis specified by the direction matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$, where \mathbf{d}_1 , \mathbf{d}_2 , and \mathbf{d}_3 correspond to the (normalized) material warp, weft, and normal directions, respectively.

Let \mathbf{L}_j be the reflection matrix $\mathbf{I} - 2\mathbf{d}_j\mathbf{d}_j^T$; then, to demonstrate orthotropic symmetry, we must establish that \mathbf{F} and $\mathbf{F}\mathbf{L}_j$ produce the same energy density ψ for any \mathbf{F} and for each j . That is, we must have $\psi^{\mathbf{E}}(\mathbf{E}) = \psi^{\mathbf{E}}(\mathbf{L}_j\mathbf{E}\mathbf{L}_j)$.

To construct such an energy density, define the reduced strain $\tilde{\mathbf{E}}$ as $\tilde{\mathbf{E}} = \mathbf{D}^T\mathbf{E}\mathbf{D}$. By

definition of the deformation gradient, $\mathbf{F}\mathbf{d}_\alpha$ lies in the deformed configuration tangent plane, while the Kirchhoff-Love assumptions imply $\mathbf{F}\mathbf{d}_3$ is a unit vector in the deformed configuration's normal direction. Thus, $\mathbf{d}_\alpha^T \mathbf{F}^T \mathbf{F} \mathbf{d}_3 = 0$ and $\mathbf{d}_3^T \mathbf{F}^T \mathbf{F} \mathbf{d}_3 = 1$, so $\tilde{\mathbf{E}}$ has the following block structure:

$$\tilde{\mathbf{E}} = \begin{pmatrix} \tilde{E}_{11} & \tilde{E}_{12} & 0 \\ \tilde{E}_{12} & \tilde{E}_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

As a result, choosing an orthotropic energy density $\psi^{\mathbf{E}}(\mathbf{E})$ just amounts to choosing an orthotropic energy density $\psi^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22})$.

In fact, we claim that the quantities $\tilde{E}_{\alpha\alpha} = \mathbf{d}_\alpha^T \mathbf{E} \mathbf{d}_\alpha$ and $\tilde{E}_{12}^2 = (\mathbf{d}_1^T \mathbf{E} \mathbf{d}_2)^2$ are invariant under the replacement $\mathbf{E} \rightarrow \mathbf{L}_j \mathbf{E} \mathbf{L}_j$ for any j . Indeed,

$$\begin{aligned} \mathbf{d}_\alpha^T \mathbf{L}_j \mathbf{E} \mathbf{L}_j \mathbf{d}_\alpha &= \begin{cases} (-\mathbf{d}_\alpha)^T \mathbf{E} (-\mathbf{d}_\alpha) & \text{if } \alpha = j \\ \mathbf{d}_\alpha^T \mathbf{E} \mathbf{d}_\alpha & \text{if } \alpha \neq j \end{cases} \\ &= \mathbf{d}_\alpha^T \mathbf{E} \mathbf{d}_\alpha \end{aligned}$$

and

$$\begin{aligned} (\mathbf{d}_1^T \mathbf{L}_j \mathbf{E} \mathbf{L}_j \mathbf{d}_2)^2 &= \begin{cases} (-\mathbf{d}_1^T \mathbf{E} \mathbf{d}_2)^2 & \text{if } 1 \leq j \leq 2 \\ (\mathbf{d}_1^T \mathbf{E} \mathbf{d}_2)^2 & \text{if } j = 3 \end{cases} \\ &= (\mathbf{d}_1^T \mathbf{E} \mathbf{d}_2)^2. \end{aligned}$$

From this it follows that any energy density ψ which can be written in the form

$$\begin{aligned} \psi &= \psi^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22}), \\ \psi^{\tilde{\mathbf{E}}} &\text{ independent of sign of } \tilde{E}_{12} \end{aligned} \tag{2.8}$$

is invariant under all orthotropic symmetries. Further, any function of $\tilde{\mathbf{E}}$ which does depend on the sign of \tilde{E}_{12} cannot possibly be orthotropy-invariant, since applying \mathbf{L}_1 on both sides will always reverse the sign of \tilde{E}_{12} while leaving $\tilde{E}_{\alpha\alpha}$ unaffected. Thus we conclude that Equation (2.8) provides a fully general framework for defining orthotropic energy densities on the Kirchhoff-Love surface. In the language of the invariant classification literature [WP64], one says that $\{\tilde{E}_{11}, |\tilde{E}_{12}|, \tilde{E}_{22}\}$ forms a function basis for the space of all orthotropic functions of $\tilde{\mathbf{E}}$.

In the following PDE derivation and implementation details we will work with a general function $\psi^{\tilde{\mathbf{E}}}$; as a result, the addition of new models to be simulated will require only routines for evaluating $\psi^{\tilde{\mathbf{E}}}$ and its first two derivatives with respect to entries of $\tilde{\mathbf{E}}$. A specific choice of $\psi^{\tilde{\mathbf{E}}}$ will be recommended in §3.2 as part of a data driven modeling approach. However, for the sake of example we also provide the energy density functions necessary for implementation of two simpler material models. For the orthotropic St. Venant-Kirchhoff model [BIE00], the energy density is stated in terms of scalar model parameters a_{11} , a_{12} , a_{22} , and G_{12} :

$$\psi_{STVK}^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22}) = \frac{a_{11}}{2} \tilde{E}_{11}^2 + a_{12} \tilde{E}_{11} \tilde{E}_{22} + \frac{a_{22}}{2} \tilde{E}_{22}^2 + G_{12} \tilde{E}_{12}^2. \quad (2.9)$$

The compressible Mooney-Rivlin model ([Cia88]) is isotropic, and is often used to simulate rubbery materials. In terms of a reduced Cauchy-Green deformation tensor

$$\tilde{\mathbf{C}} = \begin{pmatrix} 1 + 2\tilde{E}_{11} & 2\tilde{E}_{12} \\ 2\tilde{E}_{12} & 1 + 2\tilde{E}_{22} \end{pmatrix} = \text{top-left } 2 \times 2 \text{ block of } \mathbf{D}^T \mathbf{F}^T \mathbf{F} \mathbf{D}$$

together with scalar material parameters c_1 , c_2 , and D , the Mooney-Rivlin energy density is

$$\begin{aligned} \psi_{MR}^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22}) &= c_1 [\text{tr}(\tilde{\mathbf{C}}) - 2] + c_2 \left[\frac{\text{tr}(\tilde{\mathbf{C}})^2 - \text{tr}(\tilde{\mathbf{C}}^2)}{2} - 2 \right] \\ &\quad + D_1 [\det(\tilde{\mathbf{C}}) - 1]^2 - D_2 \log(\det(\tilde{\mathbf{C}})). \end{aligned}$$

A slight variation of the Mooney-Rivlin model was used with subdivision surface shells in

[CO01].

2.2.2 Interpretation of orthotropic invariants

To make Equation (2.8) more intuitive, we analyze the intuitive meaning of each entry of $\tilde{\mathbf{E}}$ by explicitly computing $\tilde{\mathbf{E}}$ for some basic deformation modes. First, suppose the cloth object undergoes a pure stretch along warp direction by a multiple of c . Then we have

$$\mathbf{F} = c\mathbf{d}_1 \otimes \mathbf{d}_1 + \mathbf{d}_2 \otimes \mathbf{d}_2 + \mathbf{d}_3 \otimes \mathbf{d}_3$$

$$\mathbf{E} = (c^2 - 1)\mathbf{d}_1 \otimes \mathbf{d}_1$$

$$\tilde{\mathbf{E}} = \begin{pmatrix} c^2 - 1 & 0 \\ 0 & 0 \end{pmatrix},$$

so we view \tilde{E}_{11} as describing the degree of warp stretch in a given deformation state. An analogous computation shows that \tilde{E}_{22} describes the weft stretch. Finally, consider a pure shear by angle θ with respect to warp and weft directions; that is, $\mathbf{d}_1 \mapsto \cos(\theta)\mathbf{d}_1 + \sin(\theta)\mathbf{d}_2$

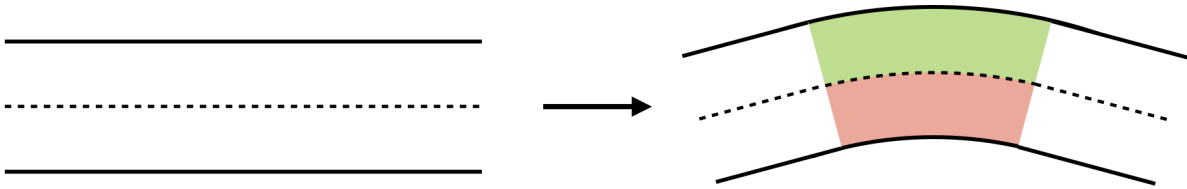


Figure 2.3: Side view of a Kirchhoff-Love plate with flat rest configuration (left), showing the midsurface as a dashed line. When the midsurface experiences a length-preserving deformation (right), nearby off-midsurface points undergo stretching (green) or compression (red).

and $\mathbf{d}_2 \mapsto \sin(\theta)\mathbf{d}_1 + \cos(\theta)\mathbf{d}_2$. This implies

$$\begin{aligned}\mathbf{F} &= \cos(\theta)\mathbf{d}_1 \otimes \mathbf{d}_1 + \sin(\theta)\mathbf{d}_1 \otimes \mathbf{d}_2 + \sin(\theta)\mathbf{d}_2 \otimes \mathbf{d}_1 + \cos(\theta)\mathbf{d}_2 \otimes \mathbf{d}_2 + \mathbf{d}_3 \otimes \mathbf{d}_3 \\ \mathbf{E} &= \frac{\sin(2\theta)}{2}(\mathbf{d}_1 \otimes \mathbf{d}_2 + \mathbf{d}_2 \otimes \mathbf{d}_1) \\ \tilde{\mathbf{E}} &= \begin{pmatrix} 0 & \frac{\sin(2\theta)}{2} \\ \frac{\sin(2\theta)}{2} & 0 \end{pmatrix},\end{aligned}$$

so our third orthotropic invariant, \tilde{E}_{12}^2 , can be viewed as describing the degree of shearing present in the current deformation.

All of the deformations considered in this section thus far are purely in-plane. In fact, due to the sparsity pattern in \mathbf{E} forced by the kinematic constraints of Kirchhoff-Love, even bending deformations are penalized in the strain tensor via these same three deformation modes. See Figure 2.3 for a visual explanation. Essentially, when the midsurface undergoes a pure bending deformation, the Kirchhoff-Love kinematic constraints dictate that nearby off-midsurface points are stretched or compressed. Thus a constitutive model written in terms of the 2D reduced strain $\tilde{\mathbf{E}}$ can still respond to out-of-plane deformations.

2.3 Governing PDE

2.3.1 Mass density and kinetic energy

To derive a PDE we need a concept of mass density. Let $\bar{\rho} : \bar{\Omega}^\tau \rightarrow \mathbb{R}$ be initial mass density defined on the 3D reference configuration, and let $\rho : \Omega^\tau \rightarrow \mathbb{R}$ be the current mass density in the deformed cloth object. Thus for example the mass of an arbitrary rest configuration subset $\mathbf{B} \subseteq \bar{\Omega}^\tau$ is

$$\text{mass}(\mathbf{B}) = \int_{\mathbf{B}} \bar{\rho}(\bar{\mathbf{p}}) d\bar{\mathbf{p}}.$$

Conservation of mass requires

$$\int_{\mathbf{B}} \bar{\rho}(\bar{\mathbf{p}}) d\bar{\mathbf{p}} = \int_{\phi(\mathbf{B})} \rho(\mathbf{p}) d\mathbf{p},$$

and after change of variables and a limiting argument, we obtain the pointwise condition

$$\bar{\rho}(\bar{\mathbf{p}}) = \rho(\phi(\bar{\mathbf{p}})) \det(\mathbf{F}(\bar{\mathbf{p}})).$$

Let \bar{J} denote the Jacobian determinant $|\frac{\partial \bar{\mathbf{r}}}{\partial \bar{\boldsymbol{\xi}}}|$ and define $\hat{\rho} : \omega^\tau \rightarrow \mathbb{R}$ to be the pullback of the initial rest configuration density; that is, $\hat{\rho} = \bar{\rho} \circ \bar{\mathbf{r}}$. Then the total kinetic energy of the surface is

$$\begin{aligned} \frac{1}{2} \int_{\Omega^\tau} \rho(\mathbf{p}) \|\dot{\phi}(\phi^{-1}(\mathbf{p}))\|^2 d\mathbf{p} &= \frac{1}{2} \int_{\bar{\Omega}^\tau} \rho(\phi(\bar{\mathbf{p}})) \|\dot{\phi}(\bar{\mathbf{p}})\|^2 \det(\mathbf{F}(\bar{\mathbf{p}})) d\bar{\mathbf{p}} \\ &= \frac{1}{2} \int_{\bar{\Omega}^\tau} \bar{\rho}(\bar{\mathbf{p}}) \|\dot{\phi}(\bar{\mathbf{p}})\|^2 d\bar{\mathbf{p}} \end{aligned} \quad (2.10)$$

$$= \frac{1}{2} \int_{\omega^\tau} \hat{\rho}(\boldsymbol{\xi}) \|\dot{\mathbf{r}}(\boldsymbol{\xi})\|^2 \bar{J}(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (2.11)$$

2.3.2 Weak form derivation

Because we model cloth as a 3D manifold, the governing equations could be derived from standard continuum mechanics laws. Given a constant gravity field $\mathbf{g} \in \mathbb{R}^3$, Newton's second law can be expressed from a Lagrangian viewpoint as

$$\bar{\rho}(\bar{\mathbf{p}}) \ddot{\phi}(\bar{\mathbf{p}}) = \nabla^{\bar{\mathbf{P}}} \mathbf{P}(\bar{\mathbf{p}}) + \bar{\rho}(\bar{\mathbf{p}}) \mathbf{g}$$

which could then be integrated against test functions to produce a weak form. However, the PDE derivation and discretization for this problem are simpler when starting from Lagrangian mechanics. The two approaches are of course mathematically equivalent, so we are free to choose the easier route.

The Kirchhoff-Love assumptions fully dictate the cloth motion based on the midsurface

map \mathbf{x} , so we must derive a PDE for evolution of \mathbf{x} . Assume $\mathbf{x} \in H^k(\omega \rightarrow \mathbb{R}^3)$, where the differentiation degree k of the Sobolev space is not yet specified; the appropriate k will be determined later based on the weak form PDE. Define the functional $T = T(\mathbf{x}, \dot{\mathbf{x}})$ to be the total kinetic energy for the 3D surface, and the functional $V = V(\mathbf{x})$ to be the total potential energy. For a functional F mapping from $H^k(\omega \rightarrow \mathbb{R}^3)$ to \mathbb{R} , we let $\frac{\partial F}{\partial \mathbf{x}}$ denote the variational derivative; that is, for any $\mathbf{x} \in H^k(\omega \rightarrow \mathbb{R}^3)$, the function $\frac{\partial F}{\partial \mathbf{x}}(\mathbf{x}) : \omega \rightarrow \mathbb{R}^3$ is almost everywhere defined and satisfies

$$\int_{\omega} \frac{\partial F}{\partial \mathbf{x}}(\mathbf{x}) \mathbf{v} d\xi^1 d\xi^2 = \left. \frac{\partial F(\mathbf{x} + \varepsilon \mathbf{v})}{\partial \varepsilon} \right|_{\varepsilon=0}$$

for an arbitrary test function $\mathbf{v} \in H^k(\omega \rightarrow \mathbb{R}^3)$.

The Lagrangian functional $L = L(\mathbf{x}, \dot{\mathbf{x}})$ is defined as the difference $L = T - V$ between total kinetic energy T and total potential energy V . The governing equations are then the Euler-Lagrange equations

$$\frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) = \frac{\partial L}{\partial \mathbf{x}}$$

and the equivalent weak form is

$$\int_{\omega} \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) \mathbf{v} d\xi^1 d\xi^2 = \int_{\omega} \left(\frac{\partial L}{\partial \mathbf{x}} \right) \mathbf{v} d\xi^1 d\xi^2, \quad (2.12)$$

where \mathbf{v} again denotes an arbitrary test function in $H^k(\omega \rightarrow \mathbb{R}^3)$.

By substituting Equation (2.1) into Equation (2.11), we obtain the kinetic energy functional T in integral form

$$T(\mathbf{x}, \dot{\mathbf{x}}) = \frac{1}{2} \int_{\omega_{\tau}} \hat{\rho}(\boldsymbol{\xi}) \|\dot{\mathbf{x}}(\xi^1, \xi^2) + \xi^3 \dot{\mathbf{a}}_3(\xi^1, \xi^2)\|^2 \bar{J}(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (2.13)$$

The term $\xi^3 \dot{\mathbf{a}}_3(\xi^1, \xi^2)$ introduces complex dependencies on both \mathbf{x} and $\dot{\mathbf{x}}$, but the multiplier ξ^3 makes this term small in most contexts. In our analysis we simply discard this term,

following the example of [CO01, KHW15]. After this simplification, the resulting variational derivatives satisfy

$$\begin{aligned} \int_{\omega} \left(\frac{\partial T}{\partial \mathbf{x}} \right) \mathbf{v} d\xi^1 d\xi^2 &= 0 \\ \int_{\omega} \frac{\partial}{\partial t} \left(\frac{\partial T}{\partial \dot{\mathbf{x}}} \right) \mathbf{v} d\xi^1 d\xi^2 &= \int_{\omega^\tau} \hat{\rho} \ddot{\mathbf{x}}^T \mathbf{v} \bar{J} d\xi. \end{aligned} \quad (2.14)$$

Given a general hyperelastic energy density function $\psi^{\mathbf{E}}(\mathbf{E})$ and a constant gravity field $\mathbf{g} \in \mathbb{R}^3$, define $\psi^{\mathbf{z}}(\mathbf{z}) = \psi^{\mathbf{z}}(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5)$ to equal $\psi(\mathbf{E}(\mathbf{z}))$, where \mathbf{z} and \mathbf{z}_j are defined as in Equation (2.7). Then the potential energy is

$$\begin{aligned} V(\mathbf{x}) &= \int_{\bar{\Omega}^\tau} \psi^{\mathbf{z}}(\mathbf{z}(\bar{\mathbf{r}}^{-1}(\bar{\mathbf{p}}))) + \bar{\rho}(\bar{\mathbf{p}}) \mathbf{g}^T \phi(\bar{\mathbf{p}}) d\bar{\mathbf{p}} \\ &= \int_{\omega^\tau} (\psi^{\mathbf{z}}(\mathbf{z}(\boldsymbol{\xi})) + \hat{\rho}(\boldsymbol{\xi}) \mathbf{g}^T \mathbf{r}(\boldsymbol{\xi})) \bar{J}(\boldsymbol{\xi}) d\boldsymbol{\xi} \\ &= \int_{\omega^\tau} (\psi^{\mathbf{z}}(\mathbf{z}(\boldsymbol{\xi}; \mathbf{x})) + \hat{\rho}(\boldsymbol{\xi}) \mathbf{g}^T \mathbf{r}(\boldsymbol{\xi}; \mathbf{x})) \bar{J}(\boldsymbol{\xi}) d\boldsymbol{\xi}. \end{aligned}$$

The functional derivative with respect to \mathbf{x} satisfies

$$\begin{aligned} \int_{\omega} \frac{\partial V}{\partial \mathbf{x}}(\mathbf{x}) \mathbf{v} d\xi^1 d\xi^2 &= \left. \frac{\partial}{\partial \varepsilon} V(\mathbf{x} + \varepsilon \mathbf{v}) \right|_{\varepsilon=0} \\ &= \left. \frac{\partial}{\partial \varepsilon} \int_{\omega^\tau} (\psi^{\mathbf{z}}(\mathbf{z}(\boldsymbol{\xi}; \mathbf{x} + \varepsilon \mathbf{v})) + \hat{\rho}(\boldsymbol{\xi}) \mathbf{g}^T \mathbf{r}(\boldsymbol{\xi}; \mathbf{x} + \varepsilon \mathbf{v})) \bar{J}(\boldsymbol{\xi}) d\boldsymbol{\xi} \right|_{\varepsilon=0} \\ &= \left. \frac{\partial}{\partial \varepsilon} \int_{\omega^\tau} (\psi^{\mathbf{z}}(\mathbf{x}_{,1} + \varepsilon \mathbf{v}_{,1}, \mathbf{x}_{,2} + \varepsilon \mathbf{v}_{,2}, \mathbf{x}_{,11} + \varepsilon \mathbf{v}_{,11}, \mathbf{x}_{,12} + \varepsilon \mathbf{v}_{,12}, \mathbf{x}_{,22} + \varepsilon \mathbf{v}_{,22}) \right. \\ &\quad \left. + \hat{\rho} \mathbf{g}^T (\mathbf{x} + \varepsilon \mathbf{v} + \xi^3 \mathbf{a}_3(\mathbf{x} + \varepsilon \mathbf{v})) \right) \bar{J} d\boldsymbol{\xi} \Big|_{\varepsilon=0} \\ &= \int_{\omega^\tau} \left(\frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,1}} \mathbf{v}_{,1} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,2}} \mathbf{v}_{,2} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,11}} \mathbf{v}_{,11} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,12}} \mathbf{v}_{,12} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,22}} \mathbf{v}_{,22} \right. \\ &\quad \left. + \hat{\rho} \mathbf{g}^T \left(\mathbf{v} + \xi^3 \frac{\partial \mathbf{a}_3}{\partial \mathbf{x}} \mathbf{v} \right) \right) \bar{J} d\boldsymbol{\xi}. \end{aligned}$$

In analogy to our simplifying assumption in the kinetic energy, we discard the term $\xi^3 \frac{\partial \mathbf{a}_3}{\partial \mathbf{x}} \mathbf{v}$

from the final integrand, yielding

$$\int_{\omega} \frac{\partial V}{\partial \mathbf{x}} \mathbf{v} d\xi^1 d\xi^2 = \int_{\omega^\tau} \left(\frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,1}} \mathbf{v}_{,1} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,2}} \mathbf{v}_{,2} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,11}} \mathbf{v}_{,11} \right. \\ \left. + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,12}} \mathbf{v}_{,12} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,22}} \mathbf{v}_{,22} + \hat{\rho} \mathbf{g}^T \mathbf{v} \right) \bar{J} d\xi. \quad (2.15)$$

Since V has no dependence on $\dot{\mathbf{x}}$,

$$\int_{\omega} \frac{\partial}{\partial t} \left(\frac{\partial V}{\partial \dot{\mathbf{x}}} \right) \mathbf{v} d\xi^1 d\xi^2 = 0. \quad (2.16)$$

Plugging the variational derivatives from Equation (2.14), Equation (2.15), and Equation (2.16) into the weak formulation of the Euler-Lagrange equations in Equation (2.12) then gives the weak form PDE for our system:

$$\int_{\omega^\tau} \hat{\rho} \ddot{\mathbf{x}}^T \mathbf{v} \bar{J} d\xi = - \int_{\omega^\tau} \left(\frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,1}} \mathbf{v}_{,1} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,2}} \mathbf{v}_{,2} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,11}} \mathbf{v}_{,11} \right. \\ \left. + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,12}} \mathbf{v}_{,12} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,22}} \mathbf{v}_{,22} + \hat{\rho} \mathbf{g}^T \mathbf{v} \right) \bar{J} d\xi \quad (2.17)$$

Recall we originally assumed $\mathbf{x}, \mathbf{v} \in H^k(\omega \rightarrow \mathbb{R}^3)$ for an unspecified k . The explicit weak form PDE above requires weak derivatives of \mathbf{x} and \mathbf{v} up to second order, so the minimum regularity requirement to make the weak form well-defined is $k = 2$.

2.3.3 Strong form and boundary terms

To derive the strong form PDE, the weak form terms are integrated by parts to pull spatial derivatives off of the test function \mathbf{v} . We demonstrate the computation for one of the weak form terms, using $\mathbf{n} = (n_1, n_2, n_3)$ to denote the outward normal on the boundary $\partial\omega^\tau$ and

$d\xi_S$ to denote the area element for boundary integrals over $\partial\omega^\tau$:

$$\begin{aligned}
\int_{\omega^\tau} \frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \mathbf{v}_{,11} \bar{J} d\xi &= \int_{\omega^\tau} \nabla \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \mathbf{v}_{,1} \mathbf{e}_1 \right) - \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \right)_{,1} \mathbf{v}_{,1} d\xi \\
&= \int_{\partial\omega^\tau} n_1 \frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \mathbf{v}_{,1} d\xi_S - \int_{\omega^\tau} \left[\nabla \left(\left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \right)_{,1} \mathbf{v} \mathbf{e}_1 \right) + \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \right)_{,11} \mathbf{v} \right] d\xi \\
&= \int_{\partial\omega^\tau} n_1 \left[\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \mathbf{v}_{,1} - \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \right)_{,1} \mathbf{v} \right] d\xi_S - \int_{\omega^\tau} \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \right)_{,11} \mathbf{v} d\xi
\end{aligned}$$

This process is repeated for the other terms in Equation (2.17), with a given weak form term requiring one application of the divergence theorem per spatial derivative initially present on \mathbf{v} . The strong form PDE is then obtained by ignoring the boundary terms (integrals over $\partial\omega^\tau$) to get an equation of the form $\int_{\omega^\tau} \boldsymbol{\zeta}^T \mathbf{v} d\xi = 0$, followed by replacing that weak criterion with the pointwise version $\boldsymbol{\zeta} = \mathbf{0}$. The resulting strong form PDE is

$$\begin{aligned}
\hat{\rho} \ddot{\mathbf{x}} \bar{J} &= -\hat{\rho} \mathbf{g} \bar{J} + \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,1}} \bar{J} \right)_{,1} + \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,2}} \bar{J} \right)_{,2} \\
&\quad - \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \right)_{,11} - \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,12}} \bar{J} \right)_{,12} - \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,22}} \bar{J} \right)_{,22}
\end{aligned} \tag{2.18}$$

As seen in our sample computation, the integration by parts produces several boundary terms on the right-hand side of the integral version of Equation (2.18). Let θ be an arbitrary scalar. Then the boundary terms are:

$$\begin{aligned}
&\int_{\partial\omega^\tau} \left\{ n_1 \left[\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,1}} \bar{J} - \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} \right)_{,1} - \theta \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,12}} \bar{J} \right)_{,2} \right] \mathbf{v} \right. \\
&\quad + n_2 \left[\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,2}} \bar{J} - \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,22}} \bar{J} \right)_{,2} - (1-\theta) \left(\frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,12}} \bar{J} \right)_{,1} \right] \mathbf{v} \\
&\quad + \left[n_1 \frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,11}} \bar{J} + n_2 \theta \frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,12}} \bar{J} \right] \mathbf{v}_{,1} \\
&\quad \left. + \left[n_2 \frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,22}} \bar{J} + n_1 (1-\theta) \frac{\partial\psi^{\mathbf{z}}}{\partial\mathbf{x}_{,12}} \bar{J} \right] \mathbf{v}_{,2} \right\} d\xi
\end{aligned} \tag{2.19}$$

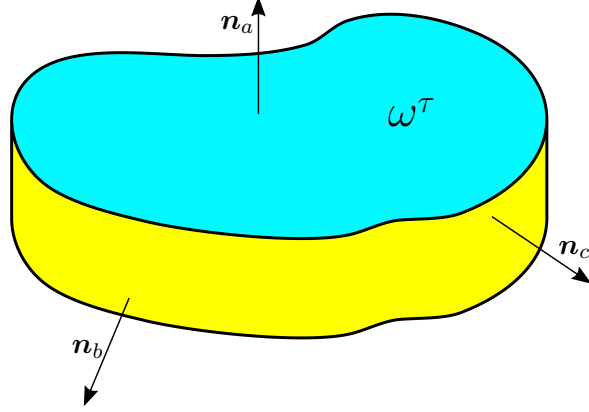


Figure 2.4: *Decomposition of the parameter space boundary $\partial\omega^\tau$ into components $\omega \times \{-\tau, \tau\}$ (blue) and $\partial\omega \times [-\tau, \tau]$ (yellow). Normal vectors such as \mathbf{n}_a which originate on the blue section are vertical; normal vectors such as \mathbf{n}_b and \mathbf{n}_c which originate on the yellow section have no vertical component.*

The arbitrary scalar θ comes from the weak form term $\int_{\omega^\tau} \frac{\partial\psi^z}{\partial x_{,12}} \mathbf{v}_{,12} \bar{J} d\xi$, which can be integrated by parts in two distinct ways depending on the order in which the derivatives with respect to ξ^1 and ξ^2 are removed from $\mathbf{v}_{,12}$. The two approaches give nominally different formulas, but of course either option will give the same result, as will any linear combination of the two methods with coefficients summing to 1.

The boundary $\partial\omega^\tau$ may be broken into intuitively distinct parts via

$$\partial\omega^\tau = (\partial\omega \times [-\tau, \tau]) \cup (\omega \times \{-\tau, \tau\});$$

as shown in Figure 2.4. At any point on the “top and bottom” boundary components $\omega \times \{-\tau, \tau\}$, the outward normal vector will be $\mathbf{n} = (0, 0, \pm 1)$, which means the integrand of Equation (2.19) will be 0. Thus the only real contribution to the boundary terms comes from the domain $\partial\omega \times [-\tau, \tau]$.

In practice, we set the Neumann boundary conditions equal to 0 and impose any desired constraints by Dirichlet means instead. We implement three distinct types of boundary conditions. A clamped boundary has specified positions and normals for \mathbf{x} ; a supported boundary has specified positions only; and a free boundary is not subject to any Dirichlet

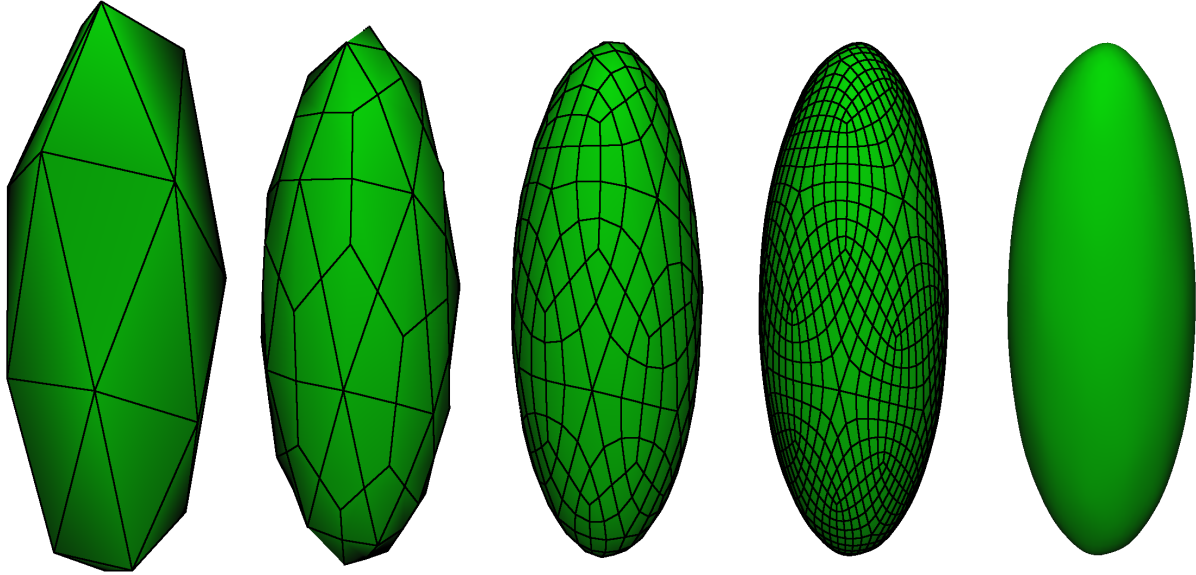


Figure 2.5: *A coarse mesh (far left) is visibly smoothed with each iteration of the Catmull-Clark subdivision scheme and quickly converges to the H^2 -regular limit surface (far right).*

constraint.

2.4 Spatial discretization

We spatially discretize the weak form PDE in Equation (2.17) using the finite element method (FEM). However, the need for H^2 regularity precludes the use of a typical piecewise-linear finite element basis, and even tracking additional node positions and derivatives on each element does not easily resolve the problem [COS00]. Instead we choose a more specialized surface representation.

2.4.1 Catmull-Clark subdivision surfaces

This project represents cloth objects' Kirchhoff-Love midsurfaces as Catmull-Clark subdivision surfaces. [CC78] and [Sta98] provide useful references on Catmull-Clark subdivision surfaces; for completeness, we restate the relevant properties here. The Catmull-Clark subdivision scheme amounts to a set of rules which take as input an arbitrary polygonal mesh and return a denser, subdivided mesh. Limiting application of the subdivision process yields

a smooth surface; this process is demonstrated in Figure 2.5. The original polygonal mesh is referred to as the ‘control cage’ or ‘control mesh’ corresponding to the final smooth subdivision surface. The output mesh from any Catmull-Clark subdivision step consists of only quadrilateral faces, independent of the polygon types present in the original mesh. By replacing the control cage with its first subdivision if necessary, we assume without loss of generality that all control cage faces are quadrilateral. Control cage vertices which have degree 4 are called regular; vertices of any other degree are termed extraordinary. A control cage face is termed regular if its one ring is topologically a 4×4 grid of vertices (and thus a 3×3 grid of quadrilateral faces). By the one ring of a face F , we mean the set of all faces which border F together with the vertices and edges of all such faces.

For a given control mesh, denote the worldspace locations of mesh vertices by row vectors $\mathbf{q}_I = (q_{I1}, q_{I2}, q_{I3}) \in \mathbb{R}^3$, where $I = 1, \dots, n_v$ and n_v is the number of control vertices. Then there exists a parameter space ω and functions $N_I \in H^2(\omega \rightarrow [0, \infty))$ corresponding to each control vertex such that the final subdivision surface can be represented as

$$\mathbf{x}(\mathbf{q}; \xi^1, \xi^2) = \mathbf{q}_I^T N_I(\xi^1, \xi^2), \quad x_j(\mathbf{q}; \xi^1, \xi^2) = q_{Ij} N_I(\xi^1, \xi^2). \quad (2.20)$$

Here summation is implied on the repeated index I . The functions N_I additionally satisfy

$$\sum_I N_I(\xi^1, \xi^2) = 1$$

for every $(\xi^1, \xi^2) \in \omega$. Equation (2.20) means that $\{N_I \mathbf{e}_j\}_{j=1,2,3; I=1, \dots, n_v}$ forms an appropriate finite element basis for spatial discretization of Equation (2.17), with the control vertices \mathbf{q}_I as the corresponding generalized coordinates. However, the finite element basis is not interpolating, i.e. the surface need not pass through any of the worldspace points \mathbf{q}_I . This represents a departure from piecewise linear finite elements on triangle mesh surfaces, where the \mathbf{q}_I would simply be the worldspace surface vertices. The set $\{\mathbf{q}_{Ij}\}$ can be thought of as a $n_v \times 3$ matrix of control point locations. With a slight abuse of notation, we use \mathbf{q} to denote the corresponding column vector of all the control vertices. With \mathbf{x} discretized according to

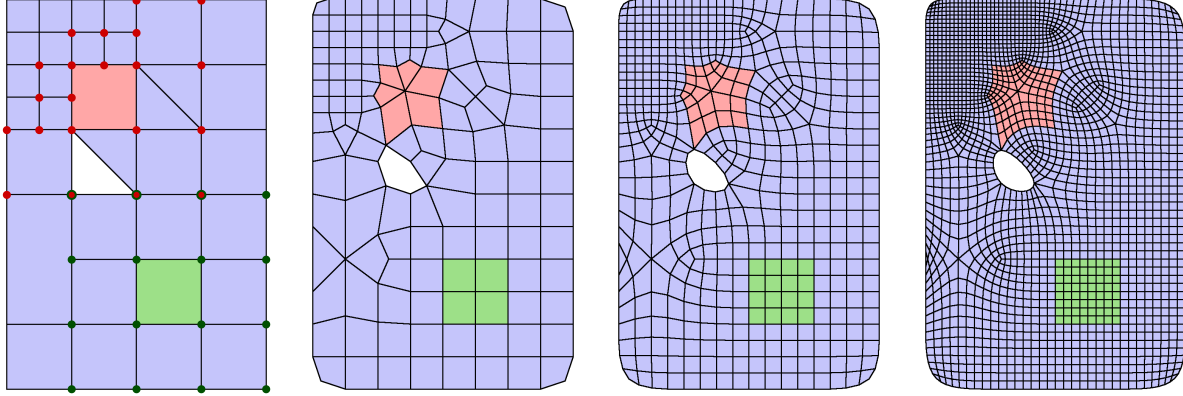


Figure 2.6: *The Catmull-Clark subdivision scheme is demonstrated for a highly irregular initial mesh (far left). Green and red colored dots mark the control cage vertices whose positions affect evolution of the same-colored patch. Note the difference in behavior between a patch beginning in a regular region (green) compared to an irregular region (red).*

Equation (2.20), we can compute the energy density ψ as a function $\psi^{\mathbf{q}}(\mathbf{q}; \boldsymbol{\xi}) = \psi^{\mathbf{z}}(\mathbf{z}(\mathbf{q}; \boldsymbol{\xi}))$. The dependence on $\boldsymbol{\xi}$ will sometimes be left implicit, leading to the notation $\psi^{\mathbf{q}}(\mathbf{q})$.

Some additional properties of Catmull-Clark surfaces will prove useful for intuition and for designing an efficient implementation. The final subdivision surface Ω admits a natural separation into patches homeomorphic to rectangles which correspond one-to-one with the non-boundary faces of the control cage. The surface patch corresponding to a given control cage face depends only on that face's one ring in the control cage. As a result, for any fixed $(\xi^1, \xi^2) \in \omega$, only a sparse subset of the finite element basis values $N_I(\xi^1, \xi^2)$ will be nonzero. For example, if all nearby vertices are regular, there will be exactly 16 nonzero values N_I . Figure 2.6 provides examples of the stencil of control vertices which affect a given limit surface patch. If a control cage face's one ring is topologically a 4×4 rectangular grid, then the corresponding section of \boldsymbol{x} simply reduces to a uniform bicubic B-spline. Thus Catmull-Clark subdivision surfaces can be viewed as a direct generalization of bicubic B-splines to allow more complex topologies.

For the purposes of this project, any choice of subdivision scheme could be applied as long as it guarantees sufficient smoothness. For example, the triangle-based Loop subdivision scheme could be used instead of Catmull-Clark [CO01]. NURBS surfaces [PT12] are

another generalization of B-splines which provide a viable alternative surface representation for finite element methods [SFH08], and have the advantage of easy generalization to basis functions using even higher order polynomials if desired. However, NURBS patches can only describe surfaces which are topologically either spheres, cylinders, or tori. It is possible to achieve arbitrary surface topologies by stitching together several NURBS patches using appropriate constraints, but the stitching process introduces significant added complexity. By comparison, a single subdivision surface is sufficient to achieve any desired surface topology.

When considering application in the animated film industry, simulation with Catmull-Clark surfaces provides the useful isogeometric property ([HCB05]). Cloth objects are typically designed and rendered as subdivision surfaces, so a simulator based on a different discretization necessitates geometry conversion steps. On the other hand, in computer-aided design (CAD), smooth surfaces are generally represented as NURBS objects, so for CAD applications a NURBS-based simulator would be the isogeometric option. The present research has been performed in association with Walt Disney Animation Studios, which partially informed the decision to use Catmull-Clark surfaces over NURBS as the underlying discretization.

2.4.2 Finite element discretization of the weak form PDE

We now substitute the discretized \mathbf{x} from Equation (2.20) into the weak form of Equation (2.17) with $\mathbf{v} = N_K \mathbf{e}_j$ for arbitrary K and standard basis vectors $\mathbf{e}_j \in \mathbb{R}^3$. The substitution gives

$$\int_{\omega^\tau} \hat{\rho} \ddot{\mathbf{q}}_{Ij}^T N_I N_K \bar{J} d\xi = - \int_{\omega^\tau} \left(\frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,1}} \mathbf{e}_j N_{K,1} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,2}} \mathbf{e}_j N_{K,2} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,11}} \mathbf{e}_j N_{K,11} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,12}} \mathbf{e}_j N_{K,12} + \frac{\partial \psi^{\mathbf{z}}}{\partial \mathbf{x}_{,22}} \mathbf{e}_j N_{K,22} + \hat{\rho} g_j N_K \right) \bar{J} d\xi. \quad (2.21)$$

This equation can be further simplified. We chose $\mathbf{x} = \mathbf{q}_I^T N_I$, so $\mathbf{x}_{,1} = \mathbf{q}_I^T N_{I,1}$ and $\frac{\partial \mathbf{x}_{,1}}{\partial \mathbf{q}_{Kj}} = N_{K,1} \mathbf{e}_j$. Using these together with analogous expressions for the other ξ derivatives, the

right-hand side of Equation (2.21) can be rewritten as

$$\int_{\omega^\tau} \hat{\rho} \ddot{\mathbf{q}}_{Ij}^T N_I N_K \bar{J} d\xi = - \int_{\omega^\tau} \left(\frac{\partial \psi^z}{\partial \mathbf{x}_{,1}} \frac{\partial \mathbf{x}_{,1}}{\partial \mathbf{q}_{Kj}} + \frac{\partial \psi^z}{\partial \mathbf{x}_{,2}} \frac{\partial \mathbf{x}_{,2}}{\partial \mathbf{q}_{Kj}} + \frac{\partial \psi^z}{\partial \mathbf{x}_{,11}} \frac{\partial \mathbf{x}_{,11}}{\partial \mathbf{q}_{Kj}} + \frac{\partial \psi^z}{\partial \mathbf{x}_{,12}} \frac{\partial \mathbf{x}_{,12}}{\partial \mathbf{q}_{Kj}} + \frac{\partial \psi^z}{\partial \mathbf{x}_{,22}} \frac{\partial \mathbf{x}_{,22}}{\partial \mathbf{q}_{Kj}} + \hat{\rho} g_j N_K \right) \bar{J} d\xi,$$

which by the chain rule is simply

$$\int_{\omega^\tau} \hat{\rho} \ddot{\mathbf{q}}_{Ij}^T N_I N_K \bar{J} d\xi = - \int_{\omega^\tau} \left(\frac{\partial \psi^{\mathbf{q}}}{\partial \mathbf{q}_{Kj}} + \hat{\rho} g_j N_K \right) \bar{J} d\xi.$$

This weak form must hold for all choices of K and j . Gathering the resulting system of equations and pulling out the exact mass matrix entries as a separate expression gives

$$\begin{aligned} \tilde{M}_{KI} \ddot{\mathbf{q}}_I &= - \int_{\omega^\tau} \left(\frac{\partial \psi^{\mathbf{q}}}{\partial \mathbf{q}_K} + \hat{\rho} \mathbf{g}^T N_K \right) \bar{J} d\xi \\ \tilde{M}_{KI} &= \int_{\omega^\tau} \hat{\rho} N_I N_K \bar{J} d\xi. \end{aligned} \tag{2.22}$$

This is the same mass matrix formula that arises in typical non-Kirchhoff-Love finite element methods. However, we note that this simple form is due to the term we ignored in Equation (2.13). Without that simplification, the “mass matrix” would be time-dependent and the kinetic energy term would become overall much more costly in implementation.

2.5 Quadrature

Evaluating Equation (2.22) numerically using a quadrature rule with evaluation sites $\xi^{(r)} \in \omega^\tau$ and corresponding weights $d\xi^{(r)}$ gives

$$M_{KI} \ddot{\mathbf{q}}_I = \mathbf{f}_K(\mathbf{q}), \tag{2.23}$$

where

$$\begin{aligned}
M_{KI} &= \sum_r \hat{\rho}^{(r)} N_I^{(r)} N_K^{(r)} \bar{J}^{(r)} d\boldsymbol{\xi}^{(r)} \\
\mathbf{f}_K(\mathbf{q}) &= - \sum_r \left(\frac{\partial \psi^{\mathbf{q}}}{\partial \mathbf{q}_K}(\mathbf{q}; \boldsymbol{\xi}^{(r)}) + \hat{\rho}^{(r)} \mathbf{g}^T N_K^{(r)} \right) \bar{J}^{(r)} d\boldsymbol{\xi}^{(r)}.
\end{aligned} \tag{2.24}$$

Let $\mathbf{f}(\mathbf{q})$ denote the column vector obtained by concatenating all vectors $\mathbf{f}_K(\mathbf{q})^T$. We have $M_{KI} \ddot{\mathbf{q}}_I^T = M_{KI} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \ddot{\mathbf{q}}_I^T$, so defining the global mass matrix \mathbf{M} as the Kronecker product

$$\mathbf{M} = \{M_{KI}\}_{K,I=1,\dots,n_v} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

allows Equation (2.23) to be condensed as

$$\mathbf{M} \ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}). \tag{2.25}$$

In practice, we use a lumped mass matrix, which combines each row's entries onto the diagonal with the result

$$M_{II} = \sum_r \hat{\rho}^{(r)} N_I^{(r)} \bar{J}^{(r)} d\boldsymbol{\xi}^{(r)}$$

and $M_{IK} = 0$ for $I \neq K$.

Regardless of which mass matrix is being used, $\hat{\rho}^{(r)}$ must be computed. For a uniform mass distribution, this is trivial. For a non-uniform mass distribution, the density will typically be specified at each control vertex, so we denote the corresponding set of values by $\hat{\rho}_L$, where $L = 1, \dots, n_v$. To compute $\hat{\rho}^{(r)}$, we then interpolate the density function at quadrature points:

$$\hat{\rho}^{(r)} = \hat{\rho}_L N_L^{(r)}.$$

For the lumped mass matrix this leads to

$$M_{II} = \sum_r \hat{\rho}_L N_L^{(r)} N_I^{(r)} \bar{J}^{(r)} d\boldsymbol{\xi}^{(r)}. \tag{2.26}$$

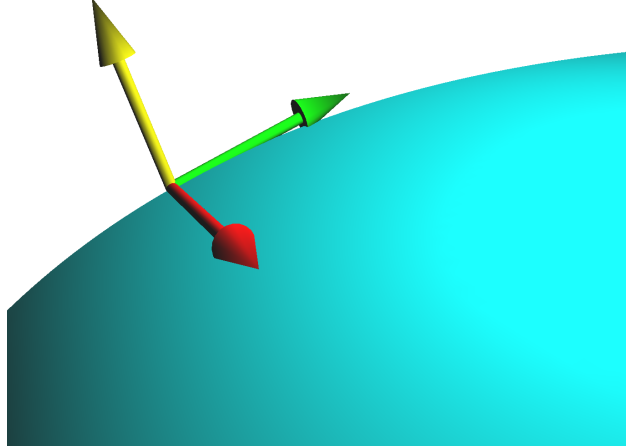


Figure 2.7: *Direction vectors used in formulating clamped boundary conditions to constrain cloth positions and normals. \mathbf{b}_n (yellow) is the constrained normal direction. \mathbf{t} (red) is the in-plane direction perpendicular to the tangent vector to \mathbf{b}_s (tangent vector shown in green).*

Similar to [CO01], we use a three-point Simpson’s rule for integration through the thickness and Gauss quadrature for integration across the surface. It is important to choose a sufficient number of Gaussian quadrature points. For Loop subdivision, a single point per triangle is sufficient [COS00]. Catmull-Clark basis functions on the other hand are of polynomial degree 3 in regular regions, so integrands involving just the basis functions have total degree 6, and thus require 4×4 quadrature points [Coo97]. For nonlinear integrands, more quadrature points may in general be necessary, but we have found 4×4 quadrature points to be sufficient for our examples. The integration gets more complicated around extraordinary vertices. Most recently, this has been studied by [JMP16] for Loop subdivision and [WP16] for Catmull-Clark schemes.

2.6 Enforcement of boundary conditions

To enforce a supported boundary condition (positions constrained), we add the requirement $\mathbf{x}(\xi^1, \xi^2) = \mathbf{b}_s(\xi^1, \xi^2)$ for some specified list of parameter space points (ξ^1, ξ^2) along the boundary segment in question. Each mesh element corresponds to a quadrilateral region in parameter space and, in practice, we apply boundary conditions (if any) to the points

at the corners of each of these regions. The condition at each such point discretizes as $\mathbf{b}_s(\xi^1, \xi^2) = \mathbf{q}_I^T N_I(\xi^1, \xi^2)$, which is a linear constraint on \mathbf{q} . For a clamped boundary segment (positions and normals constrained), we start with the supported boundary condition \mathbf{b}_s , and then additionally enforce normality of a specified direction $\mathbf{b}_n(\xi^1, \xi^2)$ to the surface. Let $\mathbf{t}(\xi^1, \xi^2)$ denote the surface tangent vector perpendicular to the boundary edge. The roles of \mathbf{b}_s , \mathbf{b}_n , and \mathbf{t} are summarized in Figure 2.7.

\mathbf{t} is an in-plane direction and so must equal some linear combination of $\mathbf{a}_1 = \mathbf{q}_I^T N_{I,1}$ and $\mathbf{a}_2 = \mathbf{q}_I^T N_{I,2}$, with coefficients depending on the boundary edge orientation in parameter space. Thus the necessary condition

$$\mathbf{t}(\xi^1, \xi^2) \cdot \mathbf{b}_n(\xi^1, \xi^2) = 0$$

amounts to another linear constraint on \mathbf{q} . For compatibility, the curve \mathbf{b}_s must be perpendicular to the desired normal \mathbf{b}_n , which (approximately) ensures the surface tangent vector along the boundary edge will be normal to \mathbf{b}_n without requiring any additional constraints.

We write the combination of all linear constraints for a given simulation via the requirement $\mathbf{B}\mathbf{q} = \mathbf{b}(t)$. For implementation purposes, the matrix \mathbf{B} of linear constraint coefficients may be precomputed since it depends only on the topology of the parameter space mesh. We incorporate this extra requirement into Equation (2.25) using a Lagrange multiplier vector $\boldsymbol{\lambda}$ as

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} &= \mathbf{f} + \mathbf{B}^T \boldsymbol{\lambda} \\ \mathbf{B}\mathbf{q} &= \mathbf{b} \end{aligned} \tag{2.27}$$

This constraint approach is based on the ideas presented by [GT04].

2.7 Implicit integration

For simulation purposes, the ODE in Equation (2.27) is discretized temporally using backward Euler. The result is a nonlinear system to be solved at each timestep for $(\mathbf{q}^{n+1}, \boldsymbol{\lambda}^{n+1})$:

$$\begin{aligned} \frac{\mathbf{M}}{h^2}(\mathbf{q}^{n+1} - \mathbf{q}^n - h\dot{\mathbf{q}}^n) &= \mathbf{f}(\mathbf{q}^{n+1}) + \mathbf{B}^T \boldsymbol{\lambda}^{n+1} \\ \mathbf{B}\mathbf{q}^{n+1} &= \mathbf{b}^{n+1}. \end{aligned}$$

Here h denotes the time step size.

Our implementation augments the nonlinear system with the Rayleigh damping model of [GS14]. Let $\mathbf{K} = -\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ denote the energy Hessian. Then the damping force $\tilde{\mathbf{f}}^d$ at time step $n+1$ with nodal positions \mathbf{q}^{n+1} is written in terms of a nonnegative scalar parameter β as

$$\tilde{\mathbf{f}}^d(\mathbf{q}^{n+1}) = -\beta \mathbf{K}(\mathbf{q}^{n+1}) \dot{\mathbf{q}}^{n+1}.$$

We replace $\tilde{\mathbf{f}}^d$ for implementation purposes with a lagged and clamped version

$$\hat{\mathbf{f}}^d(\mathbf{q}^{n+1}) = -\beta \tilde{\mathbf{K}}(\mathbf{q}^n) \dot{\mathbf{q}}^{n+1} \quad (2.28)$$

where $\tilde{\mathbf{K}}$ is a specialized modification of \mathbf{K} which is guaranteed to be positive definite. Lagged evaluation at \mathbf{q}^n instead of \mathbf{q}^{n+1} keeps the damping force gradient from depending on third derivatives of $\psi^{\mathbf{q}}$, which depending on the choice of energy density could be computationally prohibitive or even undefined. Use of $\tilde{\mathbf{K}}$ instead of \mathbf{K} ensures that the damping force is dissipative. The modified damping term is discretized by backward Euler to obtain

$$\mathbf{f}^d(\mathbf{q}^{n+1}) = -\beta \tilde{\mathbf{K}}(\mathbf{q}^n) \left(\frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{h} \right), \quad (2.29)$$

which is added to the nonlinear equations for the implicit time step:

$$\begin{aligned} \frac{\mathbf{M}}{h^2}(\mathbf{q}^{n+1} - \mathbf{q}^n - h\dot{\mathbf{q}}^n) &= \mathbf{f}(\mathbf{q}^{n+1}) + \mathbf{f}^d(\mathbf{q}^{n+1}) + \mathbf{B}^T \boldsymbol{\lambda}^{n+1} \\ \mathbf{B}\mathbf{q}^{n+1} &= \mathbf{b}^{n+1} \end{aligned} \quad (2.30)$$

To compute the new configuration \mathbf{q}^{n+1} , we use the optimization formulation proposed in [RO99, KYT06, GS14]. Discretizing the cloth potential energy V with the same finite element approximation of \mathbf{x} and quadrature rule as used for the weak form PDE gives a discrete potential energy $\Psi(\mathbf{q})$ computable by

$$\Psi(\mathbf{q}) = \sum_r \left(\psi^{\mathbf{q}}(\mathbf{q}; \boldsymbol{\xi}^{(r)}) + \hat{\rho}^{(r)} \mathbf{g}^T \mathbf{q}_K N_K^{(r)} \right) \bar{J}^{(r)} d\boldsymbol{\xi}^{(r)}. \quad (2.31)$$

Comparison to Equation (2.24) reveals $\frac{\partial \Psi(\mathbf{q})}{\partial \mathbf{q}} = -\mathbf{f}(\mathbf{q})$. We also define a ‘‘potential’’ Ψ^d corresponding to the Rayleigh damping force \mathbf{f}^d by

$$\Psi^d(\mathbf{q}) = (\mathbf{q} - \mathbf{q}^n)^T \left(\frac{\beta}{2h} \tilde{\mathbf{K}}(\mathbf{q}^n) \right) (\mathbf{q} - \mathbf{q}^n). \quad (2.32)$$

Of course, Ψ^d is clearly not a potential energy in the physical sense, given its dependence on \mathbf{q}^n and h . However, the key property $\frac{\partial \Psi^d(\mathbf{q})}{\partial \mathbf{q}} = -\mathbf{f}^d(\mathbf{q})$ does hold. We now define an incremental potential $E(\mathbf{q})$ by

$$E(\mathbf{q}) = (\mathbf{q} - \mathbf{q}^n - h\dot{\mathbf{q}}^n)^T \left(\frac{\mathbf{M}}{2h^2} \right) (\mathbf{q} - \mathbf{q}^n - h\dot{\mathbf{q}}^n) + \Psi(\mathbf{q}) + \Psi^d(\mathbf{q}). \quad (2.33)$$

By construction of E , the system in Equation (2.30) is exactly the first-order necessary condition for \mathbf{q}^{n+1} to be a minimizer in the constrained problem

$$\begin{aligned} &\underset{\mathbf{q}}{\text{minimize}} && E(\mathbf{q}) \\ &\text{subject to} && \mathbf{B}\mathbf{q} = \mathbf{b}^{n+1}. \end{aligned} \quad (2.34)$$

To solve for the new configuration $(\mathbf{q}^{n+1}, \boldsymbol{\lambda}^{n+1})$ we use Newton’s method. Recasting the

nonlinear system as an optimization problem provides the ability to augment the Newton step directions with a line search procedure, ensuring each step satisfies the strong Wolfe conditions based on the incremental potential; see [NW06] for details. In practical examples we find the inclusion of line search significantly improves convergence rates and thus runtime, as well as making the solver more robust in ill-conditioned steps. Notice that the hyperelastic nature of the constitutive models we consider is essential to enable the optimization formulation. In a model which relies on forces defined directly from the strain, the necessary incremental potential may be undefined or otherwise not readily computable.

To derive the Newton step we begin with a current iterate $(\mathbf{q}^*, \boldsymbol{\lambda}^*)$ and set $(\mathbf{q}^{n+1}, \boldsymbol{\lambda}^{n+1}) = (\mathbf{q}^* + \Delta\mathbf{q}, \boldsymbol{\lambda}^* + \Delta\boldsymbol{\lambda})$. Linearizing Equation (2.30) around $(\mathbf{q}^*, \boldsymbol{\lambda}^*)$ yields

$$\begin{aligned} \frac{\mathbf{M}}{h^2} (\mathbf{q}^* + \Delta\mathbf{q} - \mathbf{q}^n - h\dot{\mathbf{q}}^n) &= \mathbf{f}(\mathbf{q}^*) + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}(\mathbf{q}^*) \Delta\mathbf{q} - \frac{\beta}{h} \tilde{\mathbf{K}}(\mathbf{q}^n) (\mathbf{q}^* - \mathbf{q}^n) - \frac{\beta}{h} \tilde{\mathbf{K}}(\mathbf{q}^n) \Delta\mathbf{q} \\ &\quad + \mathbf{B}^T \boldsymbol{\lambda}^* + \mathbf{B}^T \Delta\boldsymbol{\lambda} \\ \mathbf{B}\mathbf{q}^* + \mathbf{B}\Delta\mathbf{q} &= \mathbf{b}^{n+1}, \end{aligned}$$

so the Newton step $(\Delta\mathbf{q}, \Delta\boldsymbol{\lambda})$ is determined via the linear system

$$\begin{aligned} \begin{pmatrix} \mathbf{K}(\mathbf{q}^*) + \beta h^{-1} \tilde{\mathbf{K}}(\mathbf{q}^n) + h^{-2} \mathbf{M} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ -\Delta\boldsymbol{\lambda} \end{pmatrix} \\ = \begin{pmatrix} \mathbf{f}(\mathbf{q}^*) + \mathbf{B}^T \boldsymbol{\lambda}^* + h^{-2} \mathbf{M} (h\dot{\mathbf{q}}^n + \mathbf{q}^n - \mathbf{q}^*) + \beta h^{-1} \tilde{\mathbf{K}}(\mathbf{q}^n) (\mathbf{q}^n - \mathbf{q}^*) \\ \mathbf{b}^{n+1} - \mathbf{B}\mathbf{q}^* \end{pmatrix}. \end{aligned} \quad (2.35)$$

We begin each timestep with the initial guess $\mathbf{q}^* = \mathbf{q}^n$.

2.8 Linear solver

According to Equation (2.35), a saddle point or KKT system must be solved at each Newton step. Let \mathbf{Y} denote the system's upper-left block, i.e.

$$\mathbf{Y} = \mathbf{K}(\mathbf{q}^*) + \beta h^{-1} \tilde{\mathbf{K}}(\mathbf{q}^n) + h^{-2} \mathbf{M}.$$

It will also prove useful to consider the alternate upper-left block $\mathbf{Y} = \mathbf{K}(\mathbf{q}^*)$, which will arise in §3.5.1 when dealing with quasistatics problems. Assume the constraints are linearly independent, i.e. \mathbf{B} has full row rank. If \mathbf{Y} is positive semidefinite then the overall system matrix will be indefinite, with exactly one negative eigenvalue per row of \mathbf{B} . In our practical usage, \mathbf{Y} is usually but not always symmetric positive definite (SPD): large simulation time steps or quasistatics problems can create tiny or negative eigenvalues in \mathbf{Y} and overall KKT systems with condition numbers in excess of 10^{10} . Therefore our choice of linear solver is heavily influenced by the need to preserve accuracy as much as possible. Some methods assuming \mathbf{Y} is SPD are more affected than others by the occasional indefinite \mathbf{Y} . For example, linear solves with system \mathbf{Y} generally produce acceptable results when using a conjugate gradient algorithm with early return on indefiniteness as in [GS14], but Cholesky factorization of \mathbf{Y} is not a reliable approach.

Our implementation handles indefinite systems using the PARDISO direct solver in the Intel MKL; see [SG06]. PARDISO solves symmetric indefinite systems using a sparse LU decomposition, and in our tests provides solutions with relative residual near the order of floating point roundoff when paired with iterative refinement using quad precision to store the linear residual. The runtime of linear solves using PARDISO is less than 20% of total solver runtime, largely due to the runtime expense of computing entries of the energy Hessian. Thus we believe this project would not reap significant benefit from heavy investment of analysis and implementation time into speeding up the linear solves. Nevertheless, we acknowledge that saddle point systems may often be solved efficiently by reducing to an equivalent SPD system. [Ben05] provides an extensive discussion of such solution methods; we reproduce the

most relevant aspects here. There exist several distinct families of methods for reduction of a saddle point problem to equivalent SPD matrix solves; here we discuss Schur complement reduction and nullspace methods.

2.8.1 Alternative: Schur complement reduction

Schur complement reduction reduces the saddle point problem to two separate linear solves with system matrices $\mathbf{BY}^{-1}\mathbf{B}^T$ and \mathbf{Y} respectively. Explicitly building $\mathbf{BY}^{-1}\mathbf{B}^T$ would require a linear solve with system \mathbf{Y} per column of \mathbf{B}^T , i.e. per linear constraint in the simulation. Our examples typically use several hundred constraints which makes explicit construction impractical. Another approach is simply to apply the matrix $\mathbf{BY}^{-1}\mathbf{B}^T$ at each step of an iterative solution method such as conjugate gradient. A linear solve with system \mathbf{Y} is then required at every conjugate gradient iteration. These solves can introduce additional error and be computationally expensive, especially since \mathbf{Y} is not always SPD and thus we cannot simply build and reuse a Cholesky decomposition for \mathbf{Y} . Another problem is that the system $\mathbf{BY}^{-1}\mathbf{B}^T$ can be ill-conditioned compared to \mathbf{Y} , exacerbating the precision problems already inherent in cloth simulation using large timesteps or quasistatics problems. In light of these concerns, we discard Schur complement reduction as a linear solver approach for our purposes.

2.8.2 Alternative: Nullspace methods

Nullspace methods require access to a matrix \mathbf{Z} whose columns form a basis for the nullspace of \mathbf{B} . The KKT solve is then replaced by a linear solve with system $\mathbf{Z}^T\mathbf{Y}\mathbf{Z}$ followed by a linear least squares solve with system \mathbf{B}^T . The reduction from \mathbf{Y} to $\mathbf{Z}^T\mathbf{Y}\mathbf{Z}$ does not change the condition number if \mathbf{Z} has orthonormal columns but can otherwise worsen the conditioning. Common methods for producing \mathbf{Z} include QR or singular value decomposition of \mathbf{B} , use of a so-called fundamental basis as in [HWS12], or (ideally) exploiting some known problem-specific structure in the constraint matrix \mathbf{B} .

The QR and singular value decompositions of \mathbf{B} are in general dense. The dimension of the reduced system $\mathbf{Z}^T \mathbf{Y} \mathbf{Z}$ is the number of vertex degrees of freedom less the number of linear constraints. Our simulation examples typically use tens of thousands of vertices but only hundreds of constraints, so a dense reduced system is not useful. However, careful reordering and approximation steps can yield a fairly sparse QR factorization [Mat94]. Such an approach could potentially be usable for the current project; the main drawback is simply the implementation time that would be required in order to discover whether a slight speedup could be obtained.

The fundamental basis approach for constructing \mathbf{Z} relies on rearranging columns of \mathbf{B} by a permutation matrix \mathbf{P} to produce a block matrix structure $\mathbf{B}\mathbf{P} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 \end{pmatrix}$ where \mathbf{B}_1 is square and invertible. Then \mathbf{Z} is chosen as

$$\mathbf{Z} = \mathbf{P} \begin{pmatrix} -\mathbf{B}_1^{-1} \mathbf{B}_2 \\ \mathbf{I} \end{pmatrix}.$$

The permutation \mathbf{P} must be chosen carefully: aside from making \mathbf{B}_1 invertible, a ideal permutation will balance controlling the reduced system condition number with making \mathbf{Z} as sparse as possible. This is a difficult problem in general, although useful heuristic methods exist; the situation is exacerbated further by the extremely poor conditioning of some of our KKT systems.

The matrix \mathbf{B} generally remains constant over the course of a simulation, which means \mathbf{Z} can be precomputed and stored for reuse. Unfortunately, we are not aware of any reliable problem-specific tricks for generating an appropriate \mathbf{Z} . Each row of \mathbf{B} has approximately 9 nonzero entries and the sparsity patterns do not admit a simple factorization. This is a significant difference from the more traditional triangle mesh based cloth simulation setting. On a triangle mesh, the standard piecewise-linear finite element basis is interpolating and the \mathbf{q}_I are just the worldspace locations of mesh vertices. As a result each Dirichlet constraint on one worldspace vertex position creates three rows in \mathbf{B} with a single nonzero entry each. This structure in \mathbf{B} makes it trivial to produce an orthonormal nullspace basis \mathbf{Z} , which

typical triangle mesh cloth solvers exploit to inexpensively reduce each KKT matrix to an SPD system without affecting the conditioning.

2.9 Derivatives of energy density

Construction of the discretized system Equation (2.35) requires evaluation of $\psi = \psi^{\mathbf{q}}(\mathbf{q})$ together with its derivatives $\frac{\partial \psi^{\mathbf{q}}}{\partial \mathbf{q}}$ and $\frac{\partial^2 \psi^{\mathbf{q}}}{\partial \mathbf{q} \partial \mathbf{q}}$. However, in practice the hyperelastic model is defined and computed in terms of $\psi^{\mathbf{E}}$, while we recall $\psi^{\mathbf{q}}$ is defined by $\psi^{\mathbf{q}}(\mathbf{q}) = \psi^{\mathbf{E}}(\mathbf{E}(\mathbf{z}(\mathbf{q})))$. The necessary derivatives of $\psi^{\mathbf{q}}$ may thus be evaluated via the chain rule given routines for computing the functions $\psi^{\mathbf{E}}(\mathbf{E})$, $\mathbf{E}(\mathbf{z})$, and $\mathbf{z}(\mathbf{q})$ together with their first and second derivatives. In particular,

$$\begin{aligned} \psi^{\mathbf{q}}(\mathbf{q}) &= \psi^{\mathbf{E}}(\mathbf{E}(\mathbf{z}(\mathbf{q}))) \\ \frac{\partial \psi^{\mathbf{q}}}{\partial \mathbf{q}} &= \frac{\partial \psi^{\mathbf{E}}}{\partial \mathbf{E}} \frac{\partial \mathbf{E}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{q}} \\ \frac{\partial^2 \psi^{\mathbf{q}}}{\partial \mathbf{q} \partial \mathbf{q}} &= \frac{\partial \mathbf{z}^T}{\partial \mathbf{q}} \frac{\partial \mathbf{E}^T}{\partial \mathbf{z}} \frac{\partial^2 \psi^{\mathbf{E}}}{\partial \mathbf{E} \partial \mathbf{E}} \frac{\partial \mathbf{E}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{q}} + \frac{\partial \psi^{\mathbf{E}}}{\partial \mathbf{E}} \left(\frac{\partial \mathbf{z}^T}{\partial \mathbf{q}} \frac{\partial^2 \mathbf{E}}{\partial \mathbf{z} \partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{q}} \right) \end{aligned} \quad (2.36)$$

where we have used the fact $\frac{\partial^2 \mathbf{z}}{\partial \mathbf{q} \partial \mathbf{q}} = 0$. To check this, we recall that the vector $\mathbf{z}(\mathbf{q}; \boldsymbol{\xi})$ is the concatenation of vectors $\mathbf{x}_{,\alpha}$ and $\mathbf{x}_{,\alpha\beta}$. The necessary fact then follows by applying the discretization in Equation (2.20) to $\mathbf{x}_{,\alpha}$ and $\mathbf{x}_{,\alpha\beta}$:

$$\begin{aligned} \mathbf{x}_{,\alpha} &= \mathbf{q}_I^T N_{I,\alpha} & \mathbf{x}_{,\alpha\beta} &= \mathbf{q}_I^T N_{I,\alpha\beta} \\ \frac{\partial \mathbf{x}_{,\alpha}}{\partial q_{Ii}} &= N_{I,\alpha} & \frac{\partial \mathbf{x}_{,\alpha\beta}}{\partial q_{Ii}} &= N_{I,\alpha\beta} \\ \frac{\partial^2 \mathbf{x}_{,\alpha}}{\partial q_{Ii} \partial q_{Jj}} &= 0 & \frac{\partial^2 \mathbf{x}_{,\alpha\beta}}{\partial q_{Ii} \partial q_{Jj}} &= 0. \end{aligned} \quad (2.37)$$

Note Equation (2.37) also provides the values for $\frac{\partial \mathbf{z}}{\partial \mathbf{q}}$.

The strain $\mathbf{E}(\mathbf{z})$ and its \mathbf{z} derivatives are computed from the formulas in §2.1.1. First, the vectors $\bar{\mathbf{g}}_j$ may be prestored for each quadrature point at the start of the solve and reused thereafter. Recall that $\mathbf{G} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}}^T \frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}}$. Then Equation (2.4) means that during the solve,

evaluation of \mathbf{E} (and its \mathbf{z} derivatives) just requires evaluation of \mathbf{G} (and its \mathbf{z} derivatives). According to Equation (2.5), we can write

$$\mathbf{G}(\xi^1, \xi^2, \xi^3) = \mathbf{A}(\xi^1, \xi^2) + \xi^3 \mathbf{B}(\xi^1, \xi^2) + (\xi^3)^2 \mathbf{C}(\xi^1, \xi^2), \quad (2.38)$$

where

$$\begin{aligned} \mathbf{A}_{\alpha\beta} &= \mathbf{a}_\alpha \cdot \mathbf{a}_\beta & \mathbf{A}_{\alpha 3} &= 0 & \mathbf{A}_{33} &= 1 \\ \mathbf{B}_{\alpha\beta} &= \mathbf{a}_\alpha \cdot \mathbf{a}_{3,\beta} + \mathbf{a}_\beta \cdot \mathbf{a}_{3,\alpha} & \mathbf{B}_{\alpha 3} &= 0 & \mathbf{B}_{33} &= 0 \\ \mathbf{C}_{\alpha\beta} &= \mathbf{a}_{3,\alpha} \cdot \mathbf{a}_{3,\beta} & \mathbf{C}_{\alpha 3} &= 0 & \mathbf{C}_{33} &= 0 \end{aligned} \quad (2.39)$$

For a given evaluation point ξ^1, ξ^2 on the midsurface, we must evaluate \mathbf{G} and its \mathbf{z} derivatives at several values of ξ^3 (depending on the quadrature rule). We minimize repeated computation by first building \mathbf{A} , \mathbf{B} , \mathbf{C} and their \mathbf{z} derivatives. All required evaluations of \mathbf{G} are then obtained using Equation (2.38). To compute \mathbf{z} derivatives of \mathbf{A} , \mathbf{B} , and \mathbf{C} , we first use the chain rule based on Equation (2.6) to store the often-reused expressions $\frac{\partial \mathbf{a}_{3,\alpha}}{\partial \mathbf{z}}$ and $\frac{\partial^2 \mathbf{a}_{3,\alpha}}{\partial \mathbf{z} \partial \mathbf{z}}$. Then \mathbf{A} , \mathbf{B} , and \mathbf{C} are differentiated by the product rule on Equation (2.39). Throughout the above process, all Hessians with respect to \mathbf{z} are stored as sparse symmetric matrices, which significantly improves runtime because several differentiated quantities have extremely simple dependencies on \mathbf{z} . The differentiation of more complex terms using the product and chain rules then amounts to a series of sparse matrix-matrix products. Note that \mathbf{E} exhibits a complex (nonquadratic) dependence on \mathbf{z} through the $\mathbf{a}_{3,\alpha}$ terms, and thus on the generalized coordinates \mathbf{q} . This is in contrast to more standard volumetric FEM and is the source of greater implementation complexity for Kirchhoff-Love FEM approaches.

The derivatives $\frac{\partial \psi^{\mathbf{E}}}{\partial \mathbf{E}}$ and $\frac{\partial^2 \psi^{\mathbf{E}}}{\partial \mathbf{E} \partial \mathbf{E}}$ are constructed by using the definition $\tilde{\mathbf{E}} = \mathbf{D}^T \mathbf{E} \mathbf{D}$ to obtain

$$\begin{aligned} \frac{\partial \psi^{\mathbf{E}}}{\partial E_{ab}} &= D_{qa} \frac{\partial \psi^{\tilde{\mathbf{E}}}}{\partial \tilde{E}_{qr}} D_{rb} \\ \frac{\partial^2 \psi^{\mathbf{E}}}{\partial E_{ab} \partial E_{cd}} &= D_{qa} D_{sc} \frac{\partial^2 \psi^{\tilde{\mathbf{E}}}}{\partial \tilde{E}_{qr} \partial \tilde{E}_{st}} D_{rb} D_{td}. \end{aligned} \quad (2.40)$$

The derivatives $\frac{\partial \psi^{\tilde{\mathbf{E}}}}{\partial \mathbf{E}}$ and $\frac{\partial^2 \psi^{\tilde{\mathbf{E}}}}{\partial \mathbf{E} \partial \mathbf{E}}$ depend on the specific choice of $\psi^{\tilde{\mathbf{E}}}$. Their computation for the specific constitutive model we recommend will therefore be postponed until that model is introduced in §3.2.

2.10 Evaluation of subdivision surfaces

To complete the evaluation of the derivatives in the previous section, we need to evaluate the basis functions, N_I , and their derivatives at the quadrature points. As previously mentioned, we use the Catmull-Clark subdivision scheme to define the basis functions, but our method is not inherently tied to this scheme. Any other subdivision scheme with sufficient smoothness could in theory be substituted. In particular, this will work fine with a triangle-based method like the Loop subdivision. However, different subdivision schemes may require different numbers of quadrature points.

In practice, we use the [OpenSubdiv](#) library from Pixar, which is what presents a restriction. The current release (version 3.3.0 at the time of this writing) only has sufficient support for Catmull-Clark subdivision surfaces. In particular, OpenSubdiv has not implemented computation of limit surface locations and tangents for any other scheme, which we need for evaluation of the basis functions N_I and their derivatives.

Given a mesh description and an input point (ξ^1, ξ^2) in parameter space, OpenSubdiv is able to efficiently evaluate the basis functions $N_I(\xi^1, \xi^2)$ and their first derivatives $N_{I,\alpha}(\xi^1, \xi^2)$ for all control control points I . For this project we additionally implemented the ability to evaluate second derivatives $N_{I,\alpha\beta}(\xi^1, \xi^2)$; this feature has since been included in version 3.2.0 of the publicly available library.

To evaluate the basis functions and their derivatives OpenSubdiv first isolates irregular vertices by adaptively subdividing the control mesh, up to some user-specified maximum level. If the parameter-space input point now lies in a regular patch, OpenSubdiv evaluates using standard bicubic B-spline basis functions for that patch. If the parameter space input point is still in an irregular patch, OpenSubdiv performs an approximate evaluation using

the Gregory patch computation technique of [LSN09].

Since our energy computation at each timestep always requires evaluation at the same quadrature points we can avoid Gregory patches altogether. For 4-point Gauss quadrature on $[0, 1]$, the smallest evaluation point is

$$\frac{1}{2} \left(1 - \sqrt{\frac{3}{7} + \frac{2}{7} \sqrt{\frac{6}{5}}} \right) \approx 0.0694 > 0.0625 = 2^{-4}.$$

By symmetry, it follows that for our 4×4 point quadrature rule on $[0, 1] \times [0, 1]$, all evaluation points are at least distance ≈ 0.0694 away from the boundary. Additionally, note that if we set `OpenSubdiv` to use s_a adaptive subdivisions, then any remaining irregular patches must lie in the $2^{-s_a} \times 2^{-s_a}$ corner regions of the original square. Thus, if we choose an adaptive subdivision level $s_a \geq 4$, we find that our quadrature points never fall in an irregular patch, which lets us eliminate the Gregory patch approximations as a possible source of error. By the same reasoning, any choice $s_a > 4$ has no effect on the accuracy of the weights. Thus, the current code uses $s_a = 4$ as the number of adaptive subdivision levels. If we wanted to use a different quadrature rule besides the current 4×4 Gauss points per patch, we would still be able to avoid Gregory patch evaluation. However, the simple analysis above would have to be repeated to find the best number of adaptive subdivisions for the new scheme.

2.11 Implementation

Our implementation separates the computations into a precomputation phase, and steps that have to be repeated within the main simulation loop.

2.11.1 Precomputation

The mass matrix is assumed constant in our simulations and can be computed directly in the precomputation phase using Equation (2.26).

Our method supports the constraint coefficient matrix \mathbf{B} changing over the course of a simulation, but \mathbf{B} is constant in our examples and thus is simply constructed and stored at the start of the simulation.

For each Newton iteration and/or time step during the simulation we need to evaluate Equation (2.39) along with the associated \mathbf{z} derivatives. This requires evaluation of \mathbf{z} at all quadrature points, which can be done using Equation (2.37). We note, however, that the derivatives of the basis functions in these expressions will always be evaluated at the same parameter space locations, namely the quadrature points. We precompute these weights using OpenSubdiv.

Finally, obtaining \mathbf{E} from \mathbf{G} requires the basis vectors $\bar{\mathbf{g}}^j$ and our numerical integration requires \bar{J} . These depend only on the rest configuration, so we precompute and store them for all quadrature points at the start of the simulation.

2.11.2 Simulation loop

Inside the main simulation loop we compute the hyperelastic energy Hessian contribution from each quadrature point $\boldsymbol{\xi}^{(r)}$ and then sum these to build the full energy Hessian. To reduce synchronization in our parallel implementation, one thread sums the contributions from all quadrature points within a specified element. The per-element stiffness matrix contributions are built concurrently, and are then added into the final Hessian one at a time. Combining the contributions per element works well because with subdivision surfaces, all $\boldsymbol{\xi}^{(r)}$ within a given element necessarily have the same stencil (that is, will share the same set of nonzero N_I). The energy computation algorithm is summarized in Algorithm 2.1, where the bulk of the work happens in line 5. As described in §2.9, we can compute the strain $\mathbf{E}(\mathbf{z})$ and its \mathbf{z} derivatives if we know the values, \mathbf{z} -gradients, and \mathbf{z} -Hessians for $\mathbf{a}_\alpha \cdot \mathbf{a}_\beta$, $\mathbf{a}_\alpha \cdot \mathbf{a}_{3,\beta}$, and $\mathbf{a}_{3,\alpha} \cdot \mathbf{a}_{3,\beta}$.

The terms $\mathbf{a}_\alpha \cdot \mathbf{a}_\beta$ are trivial to compute in terms of \mathbf{z} . To clarify the dependence of the

Algorithm 2.1 Parallelization structure for energy computation

- 1: Initialize overall system energy/gradient/Hessian as 0.
 - 2: **for all** non-boundary quad faces of the control cage **do** {in parallel}
 - 3: Initialize face energy/gradient/Hessian contributions as 0.
 - 4: **for all** quadrature points in the chosen face **do** {in serial}
 - 5: Compute energy/gradient/Hessian at this quadrature point and add these to the face's contributions.
 - 6: **end for**
 - 7: Add the face's total accumulated energy/gradient/Hessian contributions into the final system. {Use mutex to avoid race conditions!}
 - 8: **end for**
-

remaining terms on \mathbf{z} , we introduce the notation $\tilde{\mathbf{a}}_3 = \mathbf{a}_1 \times \mathbf{a}_2$ and $m = \|\tilde{\mathbf{a}}_3\|$. Then we have:

$$\begin{aligned}\mathbf{a}_3 &= \frac{\tilde{\mathbf{a}}_3}{m} \\ \mathbf{a}_{3,\alpha} &= \frac{\tilde{\mathbf{a}}_{3,\alpha}}{m} - \frac{\tilde{\mathbf{a}}_3}{m^2} m_{,\alpha} \\ m_{,\alpha} &= \frac{\tilde{\mathbf{a}}_3 \cdot \tilde{\mathbf{a}}_{3,\alpha}}{m}.\end{aligned}$$

After some simplification, we can rewrite

$$\begin{aligned}\mathbf{a}_\alpha \cdot \mathbf{a}_{3,\beta} &= -\frac{\mathbf{a}_{\alpha,\beta} \cdot \tilde{\mathbf{a}}_3}{m} \\ \mathbf{a}_{3,\alpha} \cdot \mathbf{a}_{3,\beta} &= \frac{1}{m^2} \tilde{\mathbf{a}}_{3,\alpha} \cdot \tilde{\mathbf{a}}_{3,\beta} - \frac{1}{m^4} (\tilde{\mathbf{a}}_3 \cdot \tilde{\mathbf{a}}_{3,\alpha}) (\tilde{\mathbf{a}}_3 \cdot \tilde{\mathbf{a}}_{3,\beta}).\end{aligned}$$

Thus, we can compute \mathbf{E} and its \mathbf{z} derivatives by the chain and product rules if we know the values, gradients, and Hessians of the simple ‘building block’ scalar quantities

$$m \quad \mathbf{a}_\alpha \cdot \mathbf{a}_\beta \quad \mathbf{a}_{\alpha,\beta} \cdot \tilde{\mathbf{a}}_3 \quad \tilde{\mathbf{a}}_3 \cdot \tilde{\mathbf{a}}_{3,\alpha} \quad \tilde{\mathbf{a}}_{3,\alpha} \cdot \tilde{\mathbf{a}}_{3,\beta}.$$

The strain $\mathbf{E}(\mathbf{z})$ and its \mathbf{z} derivatives are functions of our kinematic assumptions alone. As such, all of the above is equally applicable for any hyperelastic constitutive model.

2.11.3 Code structure for energy density computation

Computation of the energy density and its \mathbf{q} -derivatives incurs the majority of the runtime cost in Algorithm 2.1, and thus benefits from an efficient implementation. The computation proceeds in two stages. First, the energy density and its \mathbf{z} -derivatives are computed; then the \mathbf{q} -derivatives are obtained from Equation (2.36) using the precomputed data $\frac{\partial \mathbf{z}}{\partial \mathbf{q}}$. Regardless of the constitutive model chosen, the computation of energy density and \mathbf{z} -derivatives proves to be the key runtime-critical step; thus to facilitate future implementations we now provide a slightly simplified but fairly explicit description of our C++ code for that step.

For storage, we create a struct `Entry` that holds a double, a 15-dimensional vector, and a symmetric 15×15 matrix. These should be interpreted as some scalar value together with its gradient and Hessian with respect to \mathbf{z} . Next, we define a second struct `Gdata` that holds \mathbf{G} and its gradient and Hessian. In other words, `Gdata` stores 3 `Entry` structs, one for each nonconstant unique entry in \mathbf{G} .

Our low-level implementation consists of ‘building-block’ functions named `add_a1a1`, `add_a1a2`, and so on going through the list above. These functions accept as input an arbitrary double scalar, plus a reference to `Entry` and the input vector \mathbf{z} . They add (scalar)·(specified building block) to the input `Entry`.

The next complexity level consists of functions with names like `addG12Linear`, `addG11Quadratic`, and so on for the other terms. These functions take as input a reference to `Gdata` giving the current status of \mathbf{G} , the input vector \mathbf{z} , and the value of ξ^3 ; they call some combination of building-block functions to add the term described in their name into \mathbf{G} . So for example, `addG11Quadratic` adds the value, gradient, and Hessian of $(\xi^3)^2 \mathbf{a}_{3,1} \cdot \mathbf{a}_{3,1}$ to the (1,1) `Entry` in our `Gdata` object.

We create a function `potentialEnergyHelper` which accepts a reference to `Entry` (called `LocalEnergy`), a const reference to `Gdata` (called `G`), and a const reference to a templated `ConstitutiveModel` object. It also takes in the precomputed basis vectors $\bar{\mathbf{g}}^j$ and the orthotropy direction matrix \mathbf{D} . This function computes $\tilde{\mathbf{E}}_{\alpha\beta}$ and its \mathbf{z} derivatives, and then

Algorithm 2.2 Computation of potential energy density and \mathbf{z} -derivatives in function `potentialEnergy`

Input: Entry `psi`, `ConstitutiveModel` `model`, `PrecomputedData` `pre`, Vector `z`

```

1: // Initialize data structure.
2: Gdata G = 0;
3:
4: // Build and add contribution at  $\xi^3 = 0$ .
5: for all  $\alpha\beta$  in {11, 12, 22} do
6:   addG $\alpha\beta$ Constant(G, z);
7: end for
8: potentialEnergyHelper(psi, G, model, pre);
9:
10: // Build and add contribution at  $\xi^3 = \frac{\tau}{2}$ .
11: for all  $\alpha\beta$  in {11, 12, 22} do
12:   addG $\alpha\beta$ Linear(G, z,  $\tau/2$ );
13:   addG $\alpha\beta$ Quadratic(G, z,  $\tau/2$ );
14: end for
15: potentialEnergyHelper(psi, G, model, pre);
16:
17: // Build and add contribution at  $\xi^3 = -\frac{\tau}{2}$ .
18: for all  $\alpha\beta$  in {11, 12, 22} do
19:   addG $\alpha\beta$ Linear(G, z,  $-\tau$ );
20: end for
21: potentialEnergyHelper(psi, G, model, pre);

```

calls a member function in the `ConstitutiveModel` class to build $\psi(\tilde{\mathbf{E}})$, $\frac{\partial\psi}{\partial\tilde{\mathbf{E}}}$, and $\frac{\partial^2\psi}{\partial\tilde{\mathbf{E}}\partial\tilde{\mathbf{E}}}$. The chain rule is then used to produce the derivatives $\frac{\partial\psi^z}{\partial\mathbf{z}}$ and $\frac{\partial^2\psi^z}{\partial\mathbf{z}\partial\mathbf{z}}$, and the local energy contribution ψ and its \mathbf{z} derivatives are added to `LocalEnergy`. The Hessian $\frac{\partial^2\tilde{\mathbf{E}}_{\alpha\beta}}{\partial\mathbf{z}\partial\mathbf{z}}$ is mostly dense, so we have not made significant optimizations here. Instead, that computation essentially follows the steps a rudimentary automatic differentiation would use.

Finally, we create the top-level function `potentialEnergy`, which makes use of previously defined functionality to add the current quadrature point's contributions for ψ , $\frac{\partial\psi}{\partial\mathbf{z}}$, and $\frac{\partial^2\psi}{\partial\mathbf{z}\partial\mathbf{z}}$; see Algorithm 2.2 for details.

2.11.4 Optimizing low-level functions

It is possible to store a building-block’s gradient and Hessian in sparse form instead of recomputing the entries each time it is used. This ‘precomputation’ would still need to be repeated for every Gauss point and for each new energy evaluation, but it can potentially help if the same building block is added many times. The extra step is worth our runtime for some building blocks more than others.

Several of the building-block functions described above require very little computation to add the required terms. For example, the Hessian of $\mathbf{a}_1 \cdot \mathbf{a}_1$ has 3 nonzero entries, all of which are constant. Other building-block functions are more complex, but are not called very often. The $\mathbf{a}_{3,1} \cdot \mathbf{a}_{3,2}$ building block is relatively expensive to add, but is only added once since it appears in just one quadratic term.

In our current implementation, the only building block derivatives we prestore are the gradient and Hessian of m , and the gradients of all building blocks involving $\tilde{\mathbf{a}}_3$. We have found this to provide a runtime-efficient balance between the cost of memory allocation and the cost of recomputing the same entries multiple times.

2.11.5 Avoiding Hessian computations

The preceding sections have detailed the energy computation procedure as if every energy evaluation requires Ψ , $\frac{\partial \Psi}{\partial \mathbf{q}} = -\mathbf{f}$, and $\frac{\partial^2 \Psi}{\partial \mathbf{q} \partial \mathbf{q}} = \mathbf{K}$. Of these three, the energy Hessian $\frac{\partial^2 \Psi}{\partial \mathbf{q} \partial \mathbf{q}}$ is by far the most computationally costly. Fortunately, the Hessian computation is not always necessary. Using the optimization-based backward Euler integrator detailed in §2.7, energy evaluations can be divided into two categories: evaluations at the start of a Newton step, and evaluations during line search. Computation of Newton steps requires the energy Hessian to build the linear system of Equation (2.35), but the more common scenario is line search evaluation, for which only the energy and energy gradient are necessary.

To take advantage of this structure, we include an extra flag in the energy computation procedure which dictates whether the energy Hessian must be computed. If the flag is not

set, all levels of the chain rule based computation will skip every Hessian related computation and memory allocation, resulting in a far faster computation. This is the key optimization that makes the optimization formulation of the time integration worthwhile, since choosing a good step size via line search is very cheap compared to taking additional Newton iterations.

2.12 Convergence under mesh refinement

A useful feature of high-order finite element methods is that they typically enjoy strong convergence properties as the mesh is refined. Let Δx denote the length of the longest edge in the finite element mesh. Then a method is said to offer convergence rate $O(\Delta x^k)$ for some $k > 0$ provided that, for fixed input data, the difference between the exact PDE solution and the finite element approximation is bounded by a constant times $(\Delta x)^k$ in some specified function norm. A method's convergence rate depends on the norm chosen to measure the error. Theoretical analysis is most commonly ([Koi, COS00]) based in the Sobolev spaces H^s with corresponding norms

$$\|\mathbf{x}\|_s = \sum_{|\alpha| \leq s} \|D^\alpha \mathbf{x}\|_{L^2}$$

where we note the special case $H^0 = L^2$. However, many numerical experiments are carried out instead in the L^∞ or displacement norm ([BSL85, GT04]) due to its more direct applicability in real-world scenarios.

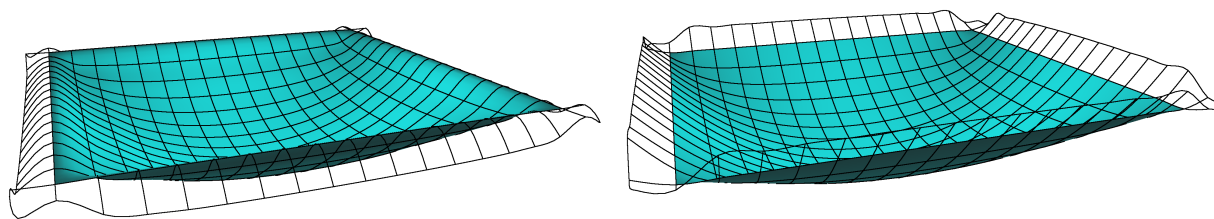


Figure 2.8: *Clamped boundary test (left) and simply supported boundary test (right) used in verifying convergence under refinement for our implementation. With the ghost face boundary treatment, the outer ring of faces in the black mesh is not part of the cloth object. Pictured meshes show the 1m × 1m test, but have been scaled by 3 in the vertical direction to make the deformation more visible.*

Several convergence results have previously been established for finite element solutions of plate and shell problems. The Koiter thin shell theory of [Koi] may be derived starting from the Kirchhoff-Love assumptions but provides a more tractable problem for theoretical convergence analysis by using an isotropic, linear stress-strain relationship and often also a linearized strain tensor. [Cia02] studies shell problems using the fully linear Koiter theory in terms of a general finite element scheme which can represent all polynomials of degree up to some integer k . The resulting error bound is $O(\Delta x^{k-2})$ in H^2 . A plate formulation with additional small-strain hypotheses is also addressed, obtaining an H^2 error bound of $O(\Delta x^{k-1})$. Due to the close relationship between Catmull-Clark subdivision surfaces and bicubic B-splines, this theoretical result is most analogous to our method when fixing $k = 3$, which suggests an H^2 bound of $O(\Delta x)$ for shell problems and $O(\Delta x^2)$ for plates. However, several factors preclude these theorems from directly applying to our implementation. First, our method considers a nonlinear strain tensor and allows highly nonlinear stress-strain relationships due to use of a general hyperelastic energy. Second, the theorem imposes several additional conditions on the problem data which may not be met for real-life problems; for example, the exact solution must lie in H^4 . Finally, the correspondence between Catmull-Clark surfaces and cubic B-splines breaks down in the presence of extraordinary vertices.

To address the greater complexities of convergence for nonlinear Kirchhoff-Love finite elements, Cirak [COS00] turns to numerical experiments. An H^2 convergence order of $O(\Delta x^2)$ is demonstrated for a plate problem, and L^∞ convergence of unspecified order are observed in several shell tests with the surface representation given by the triangle-based Loop subdivision scheme. The convergence rate for Kirchhoff-Love plates is improved by Green [GT04], in which $O(\Delta x^4)$ convergence rates are obtained in both H^2 and L^∞ , using both Loop and Catmull-Clark surfaces. The key to allowing this upgraded convergence lies in the treatment of the subdivision surface boundary. The subdivision rules in Cirak’s paper allow all faces of the control mesh to correspond to smooth surface patches, and the boundary control cage nodes are interpolated by the final surface, which simplifies imposition of Dirichlet boundary conditions. By contrast, the treatment of Green is the approach adopted in §2.4.1 and

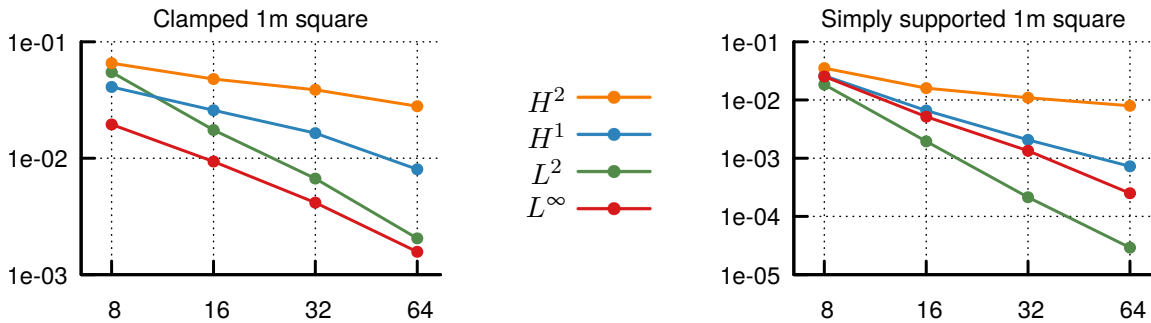


Figure 2.9: Convergence of the clamped and simply supported plate experiments in L^2 , H^1 , H^2 (energy), and L^∞ norms as the meshes are refined. L^∞ norms are measured in meters; the remaining norms are scaled to fit the graph.

§2.6: boundary faces of the control cages do not correspond to a surface patch, and Dirichlet constraints must be imposed as linear constraints with nontrivial stencils. These boundary mesh faces with no limit surface counterpart are commonly referred to as ghost faces.

These analyses both use a strain linearized with respect to the cloth-normal position ξ^3 and also a linear stress-strain relationship. To test the degree of impact of our nonlinear implementation, convergence analyses were implemented based on two commonly used thin plate experiments as shown in Figure 2.8. In the clamped boundary test, an initially flat cloth square is allowed to reach equilibrium subject to position and normal constraints at all boundary points. The simply supported boundary test is similar, but the cloth normals are not constrained. Both tests use an evenly spaced grid as the undeformed cloth mesh, with orthotropy directions \mathbf{d}_1 and \mathbf{d}_2 parallel to the edges of the cloth. The model used is the nonlinear representation of denim as will be determined in Chapter Three. For refinement analysis, each problem is solved using a variety of mesh sizes. An approximate error for each mesh except the densest is then computed by treating the densest mesh as the exact solution. Log-log plots of the resulting error estimates as shown in Figure 2.9 yield approximate convergence orders as the slope of a best-fit line. The errors are computed in four norms: L^2 , H^1 , H^2 , and L^∞ .

The experimental results for several variations of the two experiments are summarized

Test	Cloth size	L^2 order	H^1 order	H^2 order	L^∞ order
Clamped	25cm \times 25cm	1.87	1.78	1.11	1.81
Clamped	1m \times 1m	1.56	0.77	0.40	1.21
Clamped	4m \times 4m	1.06	0.47	0.20	0.96
Simply supported	25cm \times 25cm	3.65	2.48	1.50	2.79
Simply supported	1m \times 1m	3.11	1.72	0.70	2.19
Simply supported	4m \times 4m	3.12	1.64	0.62	2.08
Simply supported, linear	1m \times 1m	4.03	*	*	*
Simply supported, rotated	1m \times 1m	2.91	1.97	1.09	2.09
Simply supported, non-ghost	1m \times 1m	1.73	*	*	*

Table 2.1: Summary of convergence under refinement results. Fields marked (*) were not tested as a result of different functionality in an older version of our simulator.

in Table 2.1. The results show stronger convergence for the simply supported example than for the clamped example, but the observed convergence rates in H^2 do not replicate the high order convergence of [GT04], instead proving slightly worse than the $O(\Delta x^2)$ rate suggested by [Cia02] and [COS00]. Both experiments were repeated for three sizes of cloth squares, based on the intuition that a smaller square will experience less deformation and thus stay closer to the better-behaved linear regime. Indeed, the smaller cloth squares give better convergence rates in all cases. As further confirmation of this effect, a second model was implemented using a linearized strain tensor and stress-strain relationship, resulting in an L^2 convergence order of 4.03 for the simply supported experiment. As is typical for finite element convergence rates, each additional derivative in the Sobolev norm tends to subtract approximately one order from the convergence accuracy. Overall the data suggest that our implementation provides an H^2 convergence rate of $O(\Delta x^2)$ for the linear problem, with gradually worsened convergence as the problems move farther from the linear regime.

Two additional versions of the simply supported experiment were executed to confirm specific hypotheses. A rotated test using warp and weft directions offset from the cloth edge directions by 45° was used to confirm that the convergence rate does not depend on the orthotropy directions. Also, the effect of the ghost face boundary treatment of Green as opposed to the method of Cirak was measured by implementing and running the same convergence experiment with the non-ghost boundary definition. Use of the ghost face boundary improved the convergence rate by approximately two orders, agreeing with the finding of

Green. All of our experiments used a regular grid to mesh the cloth object. However, a full convergence study should include a variety of irregular meshes as demonstrated in [GGR06] for triangle meshes. The method is expected to converge in such a test, but the convergence rate may be slightly lower.

Regardless of order, convergence under mesh refinement is an important advantage of subdivision surface cloth simulation over triangle mesh based solvers. The bending models of [GHD03, GGW07] are neither consistent nor convergent due to reliance on a discrete shape operator which is heavily dependent on the mesh edge orientations. [GGR06] provides a modified model which provides convergence in certain weak senses as well as L^∞ convergence of unspecified order, but does not show convergence in functional norms.

CHAPTER 3

Data Driven Modeling for Woven Materials

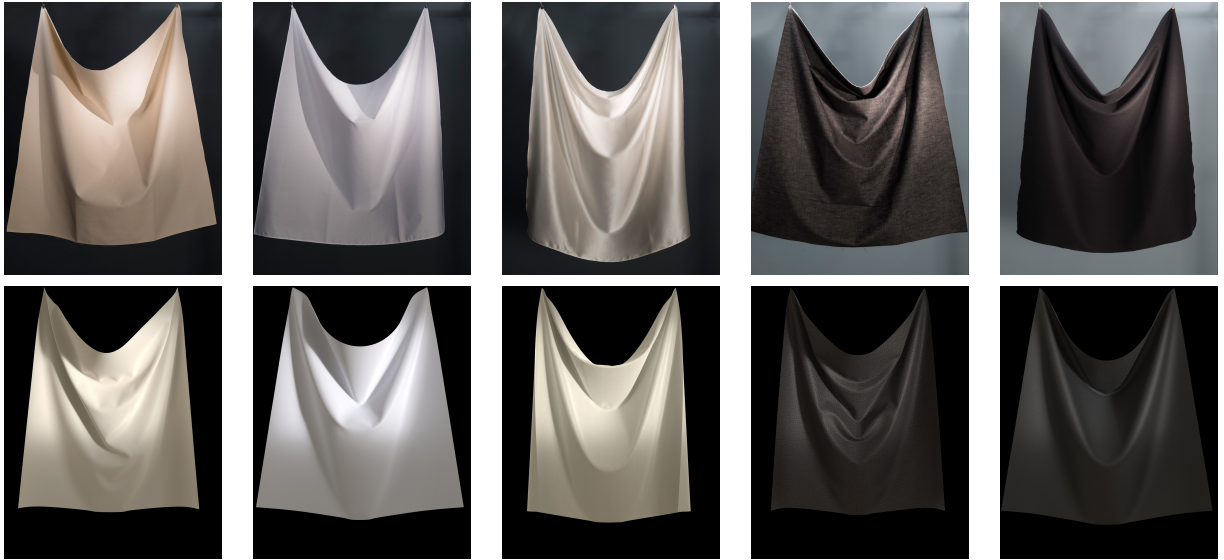


Figure 3.1: *Photographs of five fabrics (canvas, cotton poplin, silk charmeuse, denim and wool coating) along with the corresponding simulation after parameter estimation (bottom row). Each piece of fabric is 1m \times 1m, and the top corners are held 60cm apart. The real fabrics had multiple stable equilibria for the specified constraints, so exact matches cannot be expected.*

3.1 Experimental design

Our experimental approach is chosen to exercise all important deformation modes without sacrificing ease of obtainability. Standard, commercially-available procedures are strongly preferred over any test that requires physical construction of a complex apparatus. For this reason we focus on the ASTM testing standards. In section §3.6 we present results for

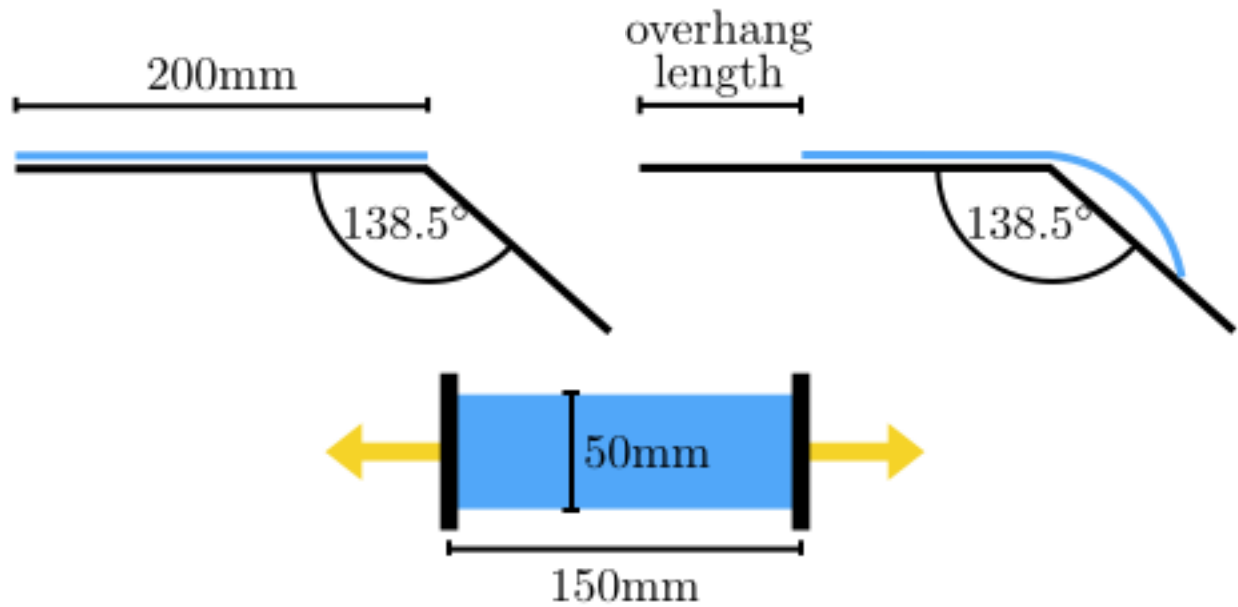


Figure 3.2: Top row: side view of the bending test ASTM D1388 before deformation begins (left) and at conclusion (right). Bottom: setup for tensile test ASTM D5035.

canvas, cotton, silk, denim, and wool. The protocol for denim and wool is different than the remaining materials as noted below.¹ To measure in-plane stretch and shear response, we use slight modifications of the tensile test ASTM D5035. The modified test uses cloth strips cut along the warp (90°), weft (0°) and bias (45°) direction. The strips used are 75 mm long (in the main direction) and 50 mm wide. For the denim and wool the strips are 150 mm long. The strips are clamped at both ends and gradually stretched until the cloth tears apart, with the clamps moving apart at a rate of 300 mm/min (200 mm/min for denim and wool). The stretch distance and the clamp forces along the stretch axis are recorded at least once per 0.1 second throughout the test for canvas, cotton and silk and every 2 milliseconds for denim and wool. For future experiments we recommend the highest possible sampling rate. For comparison purposes, each direction is tested five times for canvas, cotton and silk using a new cloth strip each time. We additionally run this test with cloth strips cut at 22.5° and 67.5° relative to the warp direction for model validation. For denim and wool we only run two repeats of each test and omit the tests for 22.5° and 67.5°. Specifically for

¹These differences are partially based on lessons learned from the first set of materials, and partially based on financial constraints.

this project, all of the above experiments were performed using an Instron 5569 tensile test machine.

Many different methods have been devised for measuring bend resistance of fabrics. One of the earliest methods based on a simple cantilever principle dates back to [Pei30]. Yet, this is still the method being used in the ASTM D1388 method as well as the FAST system. The basic idea behind the Kawabata system was first published by [LO64], and both approaches are reviewed by [GZ03]. We refer to the latter for diagrams illustrating these tests. To measure bending resistance, we use ASTM D1388. A 25 mm \times 200 mm cloth strip is slowly extended off the edge of a fixed plane and allowed to drape toward a second plane which is inclined at 41.5°. The recorded data is the overhang length of the cloth when it first makes contact with the inclined plane. This test is repeated for both warp- and weft-oriented cloth strips. We obtain the mass density of each material based on ASTM D3776, and we acquire a measured thickness for each material from ASTM D1777. However, the thickness of a woven fabric is not easy to rigorously define or measure; thus we treat the measured thickness as an initial guess as opposed to a guarantee of the correct value to be used in our model. As a result of treating thickness as an unknown in our fitting process, the measured mass density must be viewed as a mass per area instead of per volume. To compensate, the quadrature rule for the mass matrix in Equation (2.26) is adjusted to integrate only over the 2D midsurface, not through the full hyperelastic solid.

Our dataset contains little or no information about response to negative stretch strains, i.e. in-plane compression. We do not know of a test procedure for that strain regime because of cloth’s typical tendency to buckle instead of undergoing any significant compression. It should also be noted that there is no standard for measuring the shear properties of fabrics. The in-plane stretch test along the bias direction is known as the bias extension test and is well-known [Coo63], but is typically used with simplifying assumptions that are not valid for large strains, [DDB11]. However, due to our data fitting process we do not have to rely on those assumptions.

Lastly, we note that our dataset does not provide data to fit an area preservation or

Poisson type response in the cloth. The Poisson effect in fabrics is primarily due to crimp interchange which is most notable with unbalanced weaves, [SPP05]. The proposed methods in the literature for capturing data for this term are based on digital image correlation (DIC), [HRR09]. This is commercially available in so called video extensometers, and would conceptually be easy to add to our fitting method. However, we have found this term to have very little effect on the forces in our other experiments since the fabrics considered here have very little crimp to begin with. The more noticeable transversal contractions due to shearing, as seen in the bias extension test, are captured accurately with our data.

3.2 Constitutive model

We model woven cloth as an orthotropic hyperelastic solid subject to the Kirchhoff-Love kinematic assumptions. Based on the theory developed in §2.2.1, choice of a constitutive model simply amounts to choosing a strain energy density $\psi = \psi^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22})$ where $\tilde{\mathbf{E}}$ is a reduced Green-Lagrange strain matrix with entries \tilde{E}_{11} , \tilde{E}_{12} , and \tilde{E}_{22} measuring strain due to warp stretch, in-plane shearing, and weft stretch respectively. Choosing a form for $\psi^{\tilde{\mathbf{E}}}$ requires some care: the model must be sufficiently complex to capture the nonlinear stress-strain relationship exhibited in our experimental datasets, yet simple enough to facilitate a computational parameter fitting process.

Our model is defined in terms of scalar material parameters a_{11} , a_{12} , a_{22} , G_{12} as well as four scalar functions η_1 , η_2 , η_3 , η_4 to separate energy penalties for the distinct deformation modes:

$$\psi^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22}) = \frac{a_{11}}{2}\eta_1(\tilde{E}_{11}^2) + a_{12}\eta_2(\tilde{E}_{11}\tilde{E}_{22}) + \frac{a_{22}}{2}\eta_3(\tilde{E}_{22}^2) + G_{12}\eta_4(\tilde{E}_{12}^2) \quad (3.1)$$

The functions η_j are arbitrary except for the constraints $\eta_j(0) = 0$ and $\eta_j'(0) = 1$. The first constraint enforces a zero-energy, zero-stress rest configuration, while the second constraint allows a natural correspondence between the parameters $a_{\alpha\beta}$ and G_{12} and linear elasticity at infinitesimal strain. With this convention, $a_{\alpha\beta}$ and G_{12} should be interpreted as describing

the cloth’s small-strain behavior while the curves η_j describe the nonlinear response to larger strains. Indeed, comparison to Equation (2.9) shows we can replicate the orthotropic St. Venant-Kirchhoff model [BIE00] by choosing $\eta_j(x) = x$ for all j . Even with general choices of η_j , the linearization of our model around the rest configuration will always simply be the St. Venant-Kirchhoff model. We do not directly enforce positivity of the functions η_j ; this maintains the model’s expressiveness but requires some additional measures to make the method robust, as will be discussed in §3.2.3.

3.2.1 Parameterization and derivatives

The scalar functions η_j in Equation (3.1) must be expressive enough to fit the nonlinear stress-strain behavior of various material types. Simple spline representations are sufficiently general and easily adjustable by hand. However, we choose a form which is more suited to manipulation by an iterative inverse solver to facilitate subsequent data fitting. Inspired by [Its01] we use the notion of integer ‘degree’ $d_j \geq 1$ together with scalar parameters $\alpha_{j1}, \dots, \alpha_{jd_j}$ and $\mu_{j1}, \dots, \mu_{jd_j}$ to define η_j as

$$\eta_j(x) = \sum_{i=1}^{d_j} \frac{\mu_{ji}}{\alpha_{ji}} ((x+1)^{\alpha_{ji}} - 1).$$

This is akin to the constitutive model by Ogden, [Ogd72], but expressed in terms of the orthotropic functional basis $\{\tilde{E}_{11}, \tilde{E}_{22}, \tilde{E}_{12}^2\}$ instead of the principal stretches. Satisfaction of the constraint $\eta_j(0) = 0$ is automatic, while the constraint $\eta'_j(0) = 1$ is equivalent to requiring $\sum_{i=1}^{d_j} \mu_{ji} = 1$. We eliminate this constraint by removing μ_{j1} as a free parameter, setting $\mu_{j1} = 1 - \sum_{i=2}^{d_j} \mu_{ji}$.

The energy computations of §2.9 require the derivatives $\frac{\partial \psi^{\tilde{\mathbf{E}}}}{\partial \mathbf{E}}$ and $\frac{\partial^2 \psi^{\tilde{\mathbf{E}}}}{\partial \mathbf{E} \partial \mathbf{E}}$. These are com-

puted by the chain rule from Equation (3.1) in terms of the derivatives of η_j :

$$\begin{aligned}\eta_j'(x) &= \sum_{i=1}^{d_j} \mu_{ji} (x+1)^{\alpha_{ji}-1} \\ \eta_j''(x) &= \sum_{i=1}^{d_j} \mu_{ji} (\alpha_{ji} - 1) (x+1)^{\alpha_{ji}-2}\end{aligned}\tag{3.2}$$

3.2.2 Comparison to an existing model

To provide intuition for the choice of constitutive model in Equation (3.1), we compare against a nonlinear orthotropic model previously published by Itskov in [Its01]. For brevity we present a simplified version of Itskov's model which remains true to the original paper in the respects we wish to compare. Itskov's model can be defined in terms of tensor functions $\Theta_1, \Theta_2 : \mathbb{R}^{2 \times 2} \rightarrow \mathbb{R}^{2 \times 2}$ of the form

$$\Theta_j(\tilde{\mathbf{E}}) = \sum_{i=1}^{d_j} \frac{\mu_{ji}}{\alpha_{ji}} \mathbf{E}^{\alpha_{ji}}\tag{3.3}$$

as the sum of four distinct, non-negative energy terms:

$$\begin{aligned}\tilde{\psi}^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22}) &= \frac{a_{11}}{2} [\Theta_1(\tilde{\mathbf{E}}) : \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}]^2 + a_{12} [\Theta_1(\tilde{\mathbf{E}}) : \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}] [\Theta_2(\tilde{\mathbf{E}}) : \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}] \\ &+ \frac{a_{22}}{2} [\Theta_2(\tilde{\mathbf{E}}) : \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}]^2 + \frac{G_{12}}{4} [\Theta_1(\tilde{\mathbf{E}}) : \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}] [\Theta_2(\tilde{\mathbf{E}}) : \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}].\end{aligned}\tag{3.4}$$

The model parameters are positive scalar multipliers a_{11} , a_{12} , a_{22} , and G_{12} , together with integer degrees $d_j \geq 1$ and arbitrary scalar parameters $\alpha_{j1}, \dots, \alpha_{jd_j}$ and $\mu_{j1}, \dots, \mu_{jd_j}$. Non-integer powers of $\tilde{\mathbf{E}}$ arising in Equation (3.3) are computed via the matrix exponential.

The analysis in §2.2.2 provides valuable intuition for the interaction of Itskov's model with our experimental dataset. The warp stretch experiment provides approximately sparse strain tensors $\tilde{\mathbf{E}} \approx \begin{pmatrix} ? & 0 \\ 0 & 0 \end{pmatrix}$ and thus $\Theta_j(\tilde{\mathbf{E}})$ shares the same form $\Theta_j(\tilde{\mathbf{E}}) \approx \begin{pmatrix} ? & 0 \\ 0 & 0 \end{pmatrix}$. Only the a_{11} energy term is activated, creating a simplified fitting problem, and the warp stretch data is sufficient to fully determine a_{11} and the curve Θ_1 . Likewise, the weft stretch experiment determines a_{22} and Θ_2 . The bias stretch creates mostly shear strain, with $\tilde{\mathbf{E}} \approx \begin{pmatrix} 0 & ? \\ ? & 0 \end{pmatrix}$, but the matrix

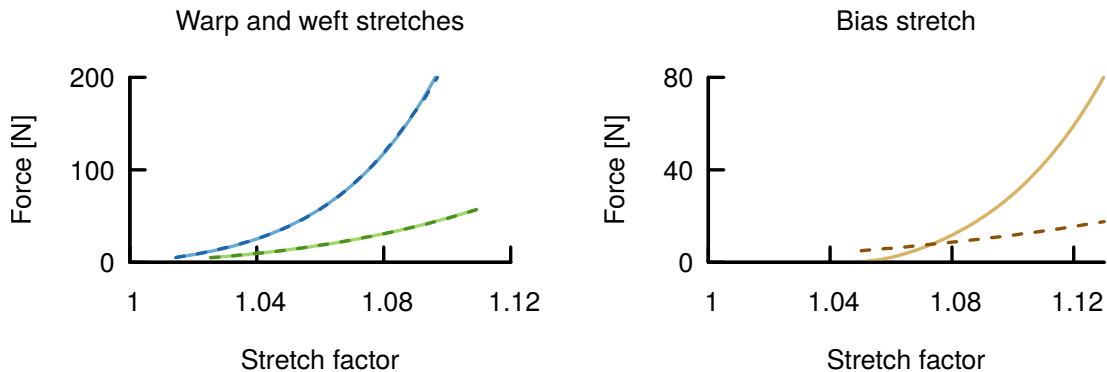


Figure 3.3: When using the model of [Its01] to fit data from cloth experiments, the warp (green) and weft (blue) datasets can be fit accurately. However, even when we set remaining energy contributions to zero, the resulting modelled bias response is far too strong at medium and large strains. Experimental data is indicated with dashed lines; solid lines show the model’s result.

exponentials in Equation (3.3) mean there is no (approximate) sparsity in the resulting matrices $\Theta_j(\tilde{\mathbf{E}})$. Therefore all four energy terms are activated. However, the a_{11} and a_{22} terms have already been determined based on warp and weft stretches, which generally offer heavy resistance in woven fabric. Shearing deformations should incur a far smaller energy penalty, but the a_{11} and a_{22} terms threaten to provide overpowering contributions even without an additional penalty from the a_{12} or G_{12} terms.

The above is only a heuristic analysis based on approximations of the deformation strains, but the conclusion is nevertheless borne out by numerical fitting attempts using real data² as displayed in Figure 3.3. Thus Itskov’s model cannot be directly applied to the cloth fitting problem. Our model in Equation (3.1) draws inspiration from Itskov’s, but our four energy summands are almost entirely decoupled. Pure warp stretch, weft stretch, and in-plane shear are each penalized by exactly one energy term, thus avoiding the interference described above.

²Figure 3.3 uses data from a partial dataset which predates the main experimental procedures described in §3.1 and which makes no appearance elsewhere in this project. The tensile experiment is similar to ours but does not gather small-strain data. The fabric tested is cotton denim.

3.2.3 Extrapolation

Our convention for η_j is effective at describing the experimental strain regime, but can give unpredictable results for strains outside the fitting dataset. The data covers strains up to the breaking point of the cloth, but we still need a plausible extrapolation to regimes where the real-life cloth would have torn apart. Although such configurations should not be observed in practice, they can temporarily occur in the iterative solvers used for fitting and simulation.

To address this, for each entry \tilde{E}_{ij} of $\tilde{\mathbf{E}}$ we define corresponding scalar ‘‘strain cutoffs’’ $\tilde{E}_{ij}^{\min} \leq 0$ and $\tilde{E}_{ij}^{\max} \geq 0$ based on the boundaries of our collected data. If all three of the strain entries satisfy $\tilde{E}_{ij}^{\min} \leq \tilde{E}_{ij} \leq \tilde{E}_{ij}^{\max}$, then the usual energy evaluation proceeds as described above. If some strain entries lie outside the valid regime, we instead evaluate our energy via a two term Taylor series approximation expanded around the closest ‘valid’ strain. For example, if $\tilde{E}_{12} < \tilde{E}_{12}^{\min}$ and $\tilde{E}_{\alpha\alpha}^{\min} \leq \tilde{E}_{\alpha\alpha} \leq \tilde{E}_{\alpha\alpha}^{\max}$, then the choice $\psi^{\tilde{\mathbf{E}}}$ in Equation (3.1) is replaced by

$$\begin{aligned} \hat{\psi}^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22}) &= \psi^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}^{\min}, \tilde{E}_{22}) + \frac{\partial \psi^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}^{\min}, \tilde{E}_{22})}{\partial \tilde{E}_{12}} \Bigg|_{\tilde{E}_{12}=\tilde{E}_{12}^{\min}} (\tilde{E}_{12} - \tilde{E}_{12}^{\min}) \\ &\quad + \frac{\partial^2 \psi^{\tilde{\mathbf{E}}}(\tilde{E}_{11}, \tilde{E}_{12}, \tilde{E}_{22})}{\partial \tilde{E}_{12} \partial \tilde{E}_{12}} \Bigg|_{\tilde{E}_{12}=\tilde{E}_{12}^{\min}} (\tilde{E}_{12} - \tilde{E}_{12}^{\min})^2 \\ &= \frac{a_{11}}{2} \eta_1(\tilde{E}_{11}^2) + a_{12} \eta_2(\tilde{E}_{11} \tilde{E}_{22}) + \frac{a_{22}}{2} \eta_3(\tilde{E}_{22}^2) \\ &\quad + G_{12} \eta_4((\tilde{E}_{12}^{\min})^2) + 2G_{12} \tilde{E}_{12}^{\min} \eta_4'((\tilde{E}_{12}^{\min})^2) (\tilde{E}_{12} - \tilde{E}_{12}^{\min}) \\ &\quad + \left[2G_{12} \eta_4'((\tilde{E}_{12}^{\min})^2) + 4G_{12} (\tilde{E}_{12}^{\min})^2 \eta_4''((\tilde{E}_{12}^{\min})^2) \right] (\tilde{E}_{12} - \tilde{E}_{12}^{\min})^2 \end{aligned}$$

with analogous substitutions made for other combinations of valid vs. invalid strain entries.

The choice of exactly two terms in the power series expansion is the simplest possible while still preserving the C^2 nature of the extrapolated energy density necessary for fitting and for implicit simulation. To maintain orthotropic symmetry we must use $\tilde{E}_{12}^{\min} = -\tilde{E}_{12}^{\max}$; thus we do not consider \tilde{E}_{12}^{\min} as an independent model parameter.

Table 3.1 provides the full list of parameters in our model.

Parameters	Units	Meaning
ρ	g/m ²	Mass density
τ	mm	Thickness
$a_{11}, a_{12}, a_{22}, G_{12}$	MPa	Infinitesimal strain parameters
$\mu_{1i}, \alpha_{1j}, 2 \leq i \leq d_1, 1 \leq j \leq d_1$	–	Nonlinear warp stretch response
$\mu_{2i}, \alpha_{2j}, 2 \leq i \leq d_2, 1 \leq j \leq d_2$	–	(Roughly) nonlinear area preservation
$\mu_{3i}, \alpha_{3j}, 2 \leq i \leq d_3, 1 \leq j \leq d_3$	–	Nonlinear weft stretch response
$\mu_{4i}, \alpha_{4j}, 2 \leq i \leq d_4, 1 \leq j \leq d_4$	–	Nonlinear shear response
$\tilde{E}_{ij}^{\max}, \tilde{E}_{ii}^{\min}, 1 \leq i, j \leq 2$	–	Transition point to extrapolation

Table 3.1: Complete list of parameters for determining cloth behavior in our model.

3.2.4 Alternative discretization using triangle mesh

Use of subdivision surface finite elements comes with two main downsides. The first of these is difficulty with collision handling. For triangle mesh surfaces, collision schemes based on [BFA02] are known to provide reliable simulation of collisions. Methods applying to subdivision or NURBS surfaces are comparatively less studied although some previous work does exist: [LZ14] attempts to adapt the standard triangle mesh collisions approach, while [DWH14] reviews several more possible methods.

The second potential downside of subdivision surface cloth is ease of adoption. Most existing cloth simulators rely on a triangle mesh cloth discretization. Adding the ability to simulate using subdivision surfaces represents a nontrivial investment of implementation time. For both of these reasons, it is valuable to note that our constitutive model (and the corresponding fitted parameter sets) can be applied to other spatial discretizations. As a demonstration, we have created an alternative implementation of our model using piecewise linear finite elements on triangle mesh surfaces. New approximation and discretization steps are required because the triangle mesh finite elements are only H^1 -smooth.

The Green-Lagrange strain \mathbf{E} for Kirchhoff-Love kinematics was written in §2.1.1 as a function $\mathbf{E}(\mathbf{z}, \xi^3)$. The subdivision surface discretization presented in Chapter Two uses the full Kirchhoff-Love strain. To obtain a version computable on triangle meshes, we follow the more traditional approach [GSH04, ZPA17, GHD03, GGW07] of expanding $\mathbf{E}(\mathbf{z}, \xi^3)$ in

a Taylor series around $\xi^3 = 0$, obtaining

$$\mathbf{E}(\mathbf{z}, \xi^3) = \mathbf{E}_m(\mathbf{z}) + \xi^3 \mathbf{E}_b(\mathbf{z}) + O((\xi^3)^2)$$

in which \mathbf{E}_m and \mathbf{E}_b are respectively termed the membrane and bending strains. These two strain components will be penalized by separate energy contributions in our triangle mesh implementation; the lower-order term $O((\xi^3)^2)$ will simply be ignored.

The ξ^3 -constant term $\mathbf{E}_m(\mathbf{z})$ is just $\mathbf{E}(\mathbf{z}, 0)$, the midsurface strain. To penalize this strain using our constitutive model, we simply plug \mathbf{E}_m into our model as if it were the full strain \mathbf{E} , and then integrate the resulting energy density over the surface in the usual way. Equivalently, one can think of using the same energy computation from the subdivision surface approach, except integrating on the midsurface only - essentially using one point quadrature in the thickness direction instead of our usual three point Simpson's rule. Inspection of Equation (2.2) and Equation (2.4) reveals that the midsurface \mathbf{E} depends only on \mathbf{a}_1 and \mathbf{a}_2 (first derivatives of \mathbf{x}), as opposed to the general Kirchhoff-Love \mathbf{E} which also depends on $\mathbf{a}_{1,1}$, $\mathbf{a}_{1,2}$, and $\mathbf{a}_{2,2}$ (second derivatives of \mathbf{x}). Thus the midsurface energy density is well defined even using the less regular H^1 triangle mesh finite elements.

To produce a discretized bending energy based on the bend strain \mathbf{E}_b , the orthotropic discrete shells approach [GHD03, GGW07] uses a hinge-based discrete shape operator to discretize the bending strain's contribution based on an orthotropic St. Venant-Kirchhoff model. By parameterizing the underlying orthotropic St. Venant-Kirchhoff model to match with the linearization of our fitted model around the rest configuration, we can build a discrete shells model that approximates our model's bending response. In the notation of [GGW07], we set the pre-discretization St. Venant-Kirchhoff parameters Y^0 , Y^1 , Y^{01} , and G^{01} (respectively representing Young's moduli in warp and weft directions, an optional area

preservation term, and the shear modulus) according to

$$\begin{aligned}
 Y^0 &= \left(\frac{a_{11}a_{22} - a_{12}^2}{a_{11}a_{22}} \right) a_{11} \\
 Y^1 &= \left(\frac{a_{11}a_{22} - a_{12}^2}{a_{11}a_{22}} \right) a_{22} \\
 Y^{01} &= \left(\frac{a_{11}a_{22} - a_{12}^2}{a_{11}a_{22}} \right) a_{12} \\
 G^{01} &= G_{12}.
 \end{aligned}$$

Combining these membrane and bending energies yields an approximation of our energy suitable for use on triangle mesh cloth. Thus even though the fitting process used subdivision surfaces, the final parameter sets can be used with minimal effort in existing triangle mesh based cloth simulators. Unfortunately the triangle mesh version of our model does not converge under mesh refinement due to inherent limitations of the discrete shell bending model. Despite this, the triangle mesh implementation produces visually acceptable simulations, and can even be combined with a standard triangle mesh collision handling implementation to run simulations involving contact. A simple plane-on-ball collision simulation is demonstrated in Figure 3.12.

3.3 Fitting strategy

We now present the methodology for fitting our model to the experimental data, with the following organization. First we construct a nonlinear objective function to minimize in §3.3.1. To facilitate the optimization we fit the parameters in multiple stages which we describe in §3.3.2. The choice of degree parameters d_j is explained in §3.3.3. Due to the complexity of the objective function it is beneficial to start when possible with good initial guesses, which we obtain through a bootstrapping method derived in §3.4. Finally we discuss the main fitting solver and several attendant implementation details in §3.5.

3.3.1 Optimization objective

The experimental observations consist of force-displacement data points for each of warp, weft, and bias stretch as well as bend test data for each of the warp and weft directions. Let \mathbf{k} denote the vector of all parameters of the constitutive model. For a given displacement with corresponding measured force \check{f} , we can recreate the same stretch distance using hard constraints to simulate clamps in our finite element solver and compute the resulting equilibrium. This gives a simulated force $\hat{f} = \hat{f}(\mathbf{k})$. Similarly, the bend test data states that some specified cloth overhang length drapes to an inclination angle of exactly $\check{\theta} = 41.5^\circ$. Again, we can recreate the test within our finite element solver by allowing a cloth strip of the same length to drape under gravity and measuring the equilibrium drape angle $\hat{\theta} = \hat{\theta}(\mathbf{k})$.

We denote list of measurements to be fit as $\{\check{p}_i : i \in I\}$ where I is an appropriate set of indices; thus each \check{p}_i is either a tensile force or a bend angle. We define the quality of a proposed fit via the following optimization objective function ε :

$$\varepsilon(\mathbf{k}) = \sum_{i \in I} \frac{(\hat{p}_i(\mathbf{k}) - \check{p}_i)^2}{\check{p}_i^2} \quad (3.5)$$

It is important to prioritize *relative* error in order to accurately match the shape of the stress-strain curve through all experimental regimes. Our datasets cover a wide variety of strain levels, so a method based on absolute error would over prioritize accuracy in the highest-strain experiments without accurately fitting the small strain data.

3.3.1.1 Interpretation of data from repeated tests

Each tensile experiment in our datasets is repeated multiple times, using a freshly cut cloth strip for each iteration. Comparing the resulting force-displacement data frequently shows relative differences of up to 20% for a fixed elongation. The cloth strip lengths and force measurements in our tensile tests are very precise due to the nature of the machinery used. We believe the major source of experimental error is imprecision in lining up the target warp-

vs-stretch-direction angles. Targeting a certain stretch angle can be a very difficult task, especially for extremely light or stretchy materials. This task is in general easiest for warp and weft directions; we generally expect the angle error to be larger for the off-angle stretches in the 45° (bias), 22.5° , and 67.5° directions. For warp or weft experiments, a strip stretched along the target direction will produce larger forces than a strip stretched along a direction offset by a few degrees. Thus, given data from multiple test executions we choose the curve with *largest* forces for fitting purposes, assuming that the corresponding experiment came closest to using the target stretch direction. There is no such simple heuristic to determine which bias cloth sample was most accurately aligned; thus we randomly select one sample for fitting. The data points from the other four experimental repetitions are omitted from the summation in ε .

3.3.2 Fitting in stages

We design our fitting approach around the effects of the various parameters \mathbf{k} . In particular, many of the test data points are almost unaffected by certain parameters leading to decoupling (or weak coupling) of some parameters. In particular, the warp stretch results depend almost exclusively on the parameters τ , a_{11} , μ_{1j} , and α_{1j} (for all $1 \leq j \leq d$). Similarly, the weft stretch results depend almost exclusively on τ , a_{22} , μ_{3j} , and α_{3j} . The bias stretch forces depend on all parameters relevant for warp or weft stretch, plus the parameters G_{12} , μ_{4j} and α_{4j} . Finally, the bend tests depend on the thickness τ together with the parameters relevant for a stretch along that same material direction.

In light of these dependencies, we design the following fitting strategy which relies on a series of minimizations of the objective function in Equation (3.5) over subsets of the experimental data. The procedure is designed such that each optimization step will provide fits for some additional parameter(s), while having no significant effect on the simulated test results fitted in the preceding steps.

1. Select an initial guess parameter set \mathbf{k}^0 . The extrapolation cutoffs \tilde{E}_{ij}^{\max} , \tilde{E}_{ii}^{\min} are ini-

tially turned off completely. The remaining parameters' guesses are chosen as described in §3.4 below.

2. Minimize the objective summed over warp stretch data points only, by solving for a_{11} , μ_{1j} , and α_{1j} with all other parameters held fixed at their values from \mathbf{k}^0 .
3. Minimize the objective summed over weft stretch data points only, by solving for a_{22} , μ_{3j} , and α_{3j} with all other parameters held fixed at their values from \mathbf{k}^0 .
4. Choose the strain cutoffs \tilde{E}_{11}^{\max} and \tilde{E}_{22}^{\max} to be 95% of the largest strain present in the corresponding tensile tests. (Note that \tilde{E}_{11}^{\min} and \tilde{E}_{22}^{\min} should always be 0 for our datasets, since we have no experimental data measuring response to warp or weft compression.)
5. Minimize the objective summed over bias stretch data points only, by solving for G_{12} , μ_{4j} , and α_{4j} with all the preceding steps' parameters fixed at their solved-for values and all remaining parameters fixed at their values from \mathbf{k}^0 .
6. Choose the strain cutoff \tilde{E}_{12}^{\max} to equal 95% of the largest strain \tilde{E}_{12} present in the bias tensile test.
7. Finally, minimize the objective summed over the warp and weft bend data points for the thickness τ . As τ is updated in the iterative process we modify the other small strain parameters using $a_{\alpha\beta}^{\text{new}} = \frac{\tau^0}{\tau} a_{\alpha\beta}$ and $G_{12}^{\text{new}} = \frac{\tau^0}{\tau} G_{12}$. All tested tensile forces scale very nearly linearly with both τ and the small-strain parameter set $\{a_{11}, a_{12}, a_{22}, G_{12}\}$; thus the parameter modifications in this step do not impact the fits of tensile data.

The parameters μ_{ij} and α_{ij} are not necessarily restricted to be positive, and in several cases the solver's final fit does include negative parameters. $a_{\alpha\alpha}$ and G_{12} are also not directly constrained, but since these are just the St. Venant-Kirchhoff parameters for infinitesimal strain they will always be positive when fitting the experimental data. The lack of explicit constraints on the parameters means the constitutive model is capable of representing non-viable energy functions; for example, the energy may not be bounded below. A successful

parameter fitting process will necessarily choose well-behaved energies in the experimental deformation regime due to the requirement of matching the measured data, but by default there is no reason to expect stable results for nonphysical stretches. Our introduction of the strain cutoffs \tilde{E}_{ij}^{\min} and \tilde{E}_{ij}^{\max} is specifically intended to ensure that the model remains stable even outside the physical regime. This is necessary for robust simulation because our implicit time integration occasionally requires evaluation at partially-converged position iterates which are greatly deformed.

Our data does not exercise the energy term $a_{12}\eta_2(\tilde{E}_{11}\tilde{E}_{22})$ sufficiently to allow fitting. In the linearization at zero strain, a_{12} determines the orthotropic Poisson's ratios via $\nu_{12} = \frac{a_{12}}{a_{11}}$ and $\nu_{21} = \frac{a_{12}}{a_{22}}$. In light of this, we use the St. Venant-Kirchhoff form $\eta_2(x) = x$ (i.e. $d_2 = 1$, $\mu_{21} = 1$, $\alpha_{21} = 1$) and set $a_{12} = \frac{1}{2} \min\{a_{11}, a_{22}\}$ for simplicity, which amounts to setting the larger Poisson's ratio to $\frac{1}{2}$.

3.3.3 Choice of degree parameters

In the list of fitting stages provided in §3.3.2, steps 2, 3, and 5 require fitting the curves η_1 , η_3 , and η_4 respectively. An iterative method is suitable for determining the parameters α_{jk} and μ_{jk} , but the integer degree parameters d_j must be determined by other means. The choice of d_j strongly affects the difficulty of the fitting process. If d_j is too small, η_j may not be sufficiently expressive to model the materials' nonlinear behavior under large deformations. If d_j is too large, we introduce near-nullspaces in the fitting process, which will complicate the optimization problem. In practice, the ideal d_j depends on the specific material being fitted, and thus we choose d_j on a per-solve basis.

To find the simplest possible model that fits the data, each solve stage is attempted first with a degree 1 curve. The degree is then incremented until a satisfactory fit is obtained. In warp or weft fitting, we compute initial guesses for each new curve degree based on a procedure which will be described in §3.4. For bias fits with degree 1 curves, the optimization is simple enough to be reliably solved regardless of the initial guess chosen. When running

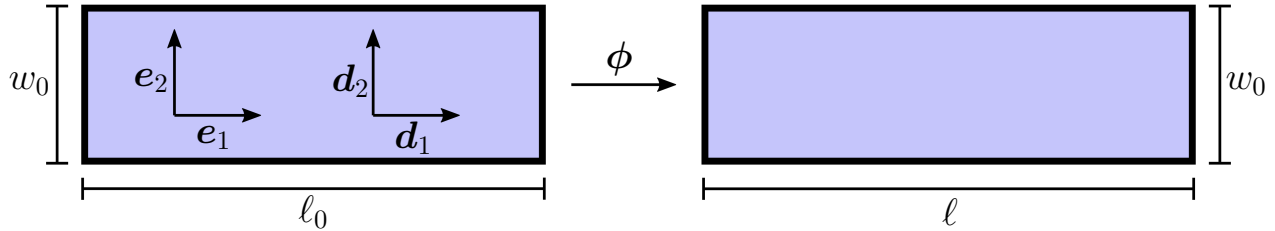


Figure 3.4: *The warp stretch equilibrium is approximated by assuming constant strain across the entire cloth object as the strip is stretched from initial dimensions $\ell_0 \times w_0$ to deformed dimensions $\ell \times w_0$. The weft stretch approximation takes the same form with the roles of \mathbf{d}_1 and \mathbf{d}_2 reversed.*

higher degree bias fits, the previous lower-degree curve is used as an initial guess. This approach takes advantage of the fact that a degree d_j curve η_j can also be written as a degree $d_j + 1$ curve, simply by choosing $\mu_{j(d_j+1)} = 0$. The corresponding exponent $\alpha_{j(d_j+1)}$ can be set to any value without affecting the initial energy; we choose the generic initial exponent $\alpha_{j(d_j+1)} = 1$.

3.4 Initial guesses for fitting solves

The minimization of Equation (3.5) is computationally expensive and benefits from good initial values in \mathbf{k}^0 . To obtain such initial guesses, we can first solve an approximated optimization problem created by assuming very simple equilibrium configurations. Such an approximated problem can be solved far more quickly because computing equilibrium configurations is normally the main runtime expense when evaluating the objective. If the approximated objective is sufficiently close to the original function, then the minimizing parameter set from the approximate problem can be chosen as \mathbf{k}^0 .

For the warp and weft stretches, we approximate the equilibrium by assuming that the entire cloth undergoes constant strain as shown in Figure 3.4. For the bias stretch we use a piecewise-constant strain assumption as discussed in [LPC08, DDB11]. The strip is broken into several components as shown in Figure 3.5, and the components marked \overline{A}_j are assumed to remain undeformed while the components \overline{B}_j and \overline{C} each undergo a constant

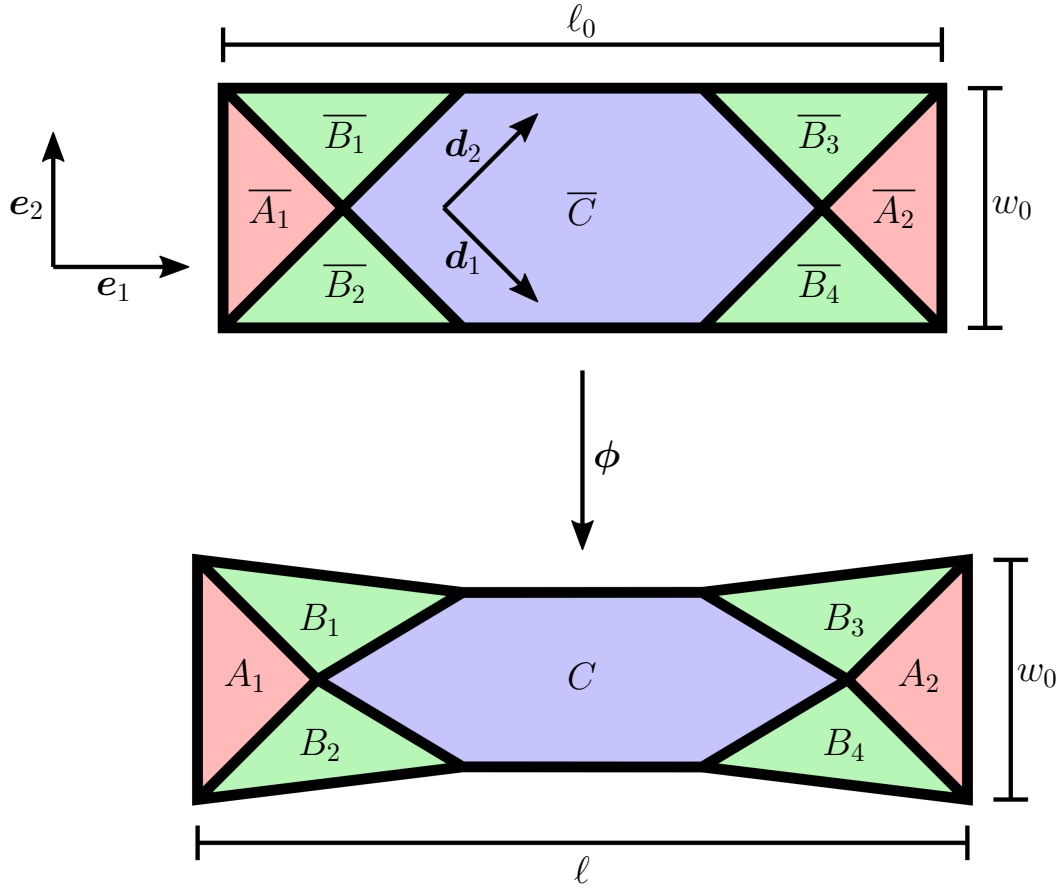


Figure 3.5: *The approximate equilibrium for the bias stretch test assumes piecewise constant strain defined per region in the diagram. When the strip as a whole is stretched from length ℓ_0 to length ℓ , regions \bar{A}_1 and \bar{A}_2 remain undeformed, while each other marked region undergoes a pure shear with respect to the warp and weft directions.*

pure shear with respect to the warp and weft directions. In both cases we assume the stretch direction is \mathbf{e}_1 and the stretch-perpendicular in-plane direction is \mathbf{e}_2 . The in-plane material directions are $\mathbf{d}_1 = \mathbf{e}_1$ and $\mathbf{d}_2 = \mathbf{e}_2$ for warp stretch, $\mathbf{d}_1 = \mathbf{e}_2$ and $\mathbf{d}_2 = \mathbf{e}_1$ for weft stretch, and $\mathbf{d}_1 = \frac{1}{\sqrt{2}}(\mathbf{e}_1 + \mathbf{e}_2)$ and $\mathbf{d}_2 = \frac{1}{\sqrt{2}}(\mathbf{e}_1 - \mathbf{e}_2)$ for bias stretch. For all tests we use the cloth normal direction $\mathbf{d}_3 = \mathbf{e}_3$. The initial and deformed strip lengths along the \mathbf{e}_1 direction are denoted by ℓ_0 and ℓ respectively; the initial strip width along the \mathbf{e}_2 direction is denoted by w_0 .

Denote the stretch factor by $\sigma = \frac{\ell}{\ell_0}$. Then the deformed configurations can be viewed simply as a function of σ . To emphasize this property, the notation of §2.1 and §2.3 is

extended: The deformation map becomes $\phi(\bar{\mathbf{p}}) = \phi(\bar{\mathbf{p}}; \sigma)$, the full and reduced Green-Lagrange strains become $\mathbf{E}(\bar{\mathbf{p}}) = \mathbf{E}(\bar{\mathbf{p}}; \sigma)$ and $\tilde{\mathbf{E}}(\bar{\mathbf{p}}) = \tilde{\mathbf{E}}(\bar{\mathbf{p}}; \sigma)$ respectively. We further define the total hyperelastic potential of the cloth object by $\Psi = \Psi(\sigma)$.

At the deformation level $\sigma = 1 + \frac{w_0}{\ell_0}(\sqrt{2}-1)$, the bias stretch kinematic assumption predicts a singular deformation gradient in the central region C . Thus we restrict our analysis to stretches bounded away from that critical level, i.e. $\sigma < 1 + \frac{w_0}{\ell_0}(\sqrt{2}-1) - \varepsilon$.

3.4.1 Power balance

Recall that the tensile experimental data only measures the clamp force component along the main stretch axis. Denote this measured force magnitude by \hat{f}^c ; the other components of clamp force are not relevant for this project. To build an approximate optimization objective, we must use the approximate kinematics to compute a model-predicted clamp force magnitude \bar{f}^c . This predicted force is computed by equating internal and external power as the clamp moves with a given speed $\dot{\sigma}$. The experimental data is obtained using very slow extension rates; thus for the following analysis we assume $\dot{\sigma}$ is a constant with $\dot{\sigma} \ll 1$.

The external power applied to the clamp by the testing apparatus is

$$\begin{aligned} P^{\text{ext}} &= (\text{force}) \cdot (\text{velocity}) \\ &= \begin{pmatrix} \bar{f}^c \\ ? \\ ? \end{pmatrix} \cdot \begin{pmatrix} \dot{\ell} \\ 0 \\ 0 \end{pmatrix} \\ &= \bar{f}^c \dot{\ell}. \end{aligned} \tag{3.6}$$

The internal power applied by the cloth strip is expressed in terms of the cloth potential energy V and kinetic energy T as

$$P^{\text{int}} = \dot{V} + \dot{T}. \tag{3.7}$$

We assume V equals the hyperelastic potential Ψ , ignoring the comparatively insignificant

gravity contribution. Then the potential energy term in Equation (3.7) is

$$\dot{V} = \frac{\partial \Psi^\sigma(\sigma)}{\partial \sigma} \dot{\sigma}. \quad (3.8)$$

To analyze the kinetic energy term \dot{T} we begin from Equation (2.10) and compute:

$$\begin{aligned} T &= \frac{1}{2} \int_{\bar{\Omega}^\tau} \bar{\rho}(\bar{\mathbf{p}}) \|\dot{\phi}(\bar{\mathbf{p}}; \sigma)\|^2 d\bar{\mathbf{p}} \\ &= \frac{\dot{\sigma}^2}{2} \int_{\bar{\Omega}^\tau} \bar{\rho}(\bar{\mathbf{p}}) \left\| \frac{\partial \phi}{\partial \sigma}(\bar{\mathbf{p}}; \sigma) \right\|^2 d\bar{\mathbf{p}}. \end{aligned} \quad (3.9)$$

Let $I(\sigma)$ denote the integral quantity in Equation (3.9), so that $T = \frac{\dot{\sigma}^2}{2} I(\sigma)$. It may be verified for either set of kinematic assumptions (in Figures 3.4 or 3.5) that $I(\sigma)$ is a differentiable function of σ with bounded derivative; the straightforward proof amounts to writing out explicit expressions for $I(\sigma)$ and is omitted here. As a result, the time derivative \dot{T} satisfies

$$\dot{T} = \dot{\sigma} \ddot{\sigma} I(\sigma) + \frac{\dot{\sigma}^3}{2} \frac{\partial I(\sigma)}{\partial \sigma}.$$

The first term is zero due to the assumption $\ddot{\sigma} = 0$. The second term lies in $O(\dot{\sigma}^3)$ and is thus insignificant compared to the $\dot{\sigma}$ -linear term in Equation (3.8) because of the assumption $\dot{\sigma} \ll 1$. Therefore the kinetic energy term \dot{T} can be safely ignored, so the power balance condition is obtained from Equation (3.6) and Equation (3.8):

$$\begin{aligned} \ell_0 \bar{f}^c \dot{\sigma} &= \frac{\partial \Psi^\sigma(\sigma)}{\partial \sigma} \dot{\sigma} \\ \bar{f}^c &= \frac{1}{\ell_0} \frac{\partial \Psi^\sigma(\sigma)}{\partial \sigma}. \end{aligned} \quad (3.10)$$

To make use of this expression, explicit expressions for $\Psi(\sigma)$ are necessary; these will depend on the specific kinematic assumptions chosen.

3.4.2 Potential energy as a function of stretch

In the approximate equilibrium for warp stretch, the deformation gradient at any undeformed position is $\mathbf{F} = \begin{pmatrix} \sigma & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, which produces the reduced strain $\tilde{\mathbf{E}} = \begin{pmatrix} \frac{\sigma^2-1}{2} & 0 \\ 0 & 0 \end{pmatrix}$. Thus in terms of our model in Equation (3.1), the hyperelastic potential is

$$\begin{aligned} \Psi(\sigma) &= \int_{\bar{\Omega}^\tau} \psi^{\mathbf{E}}(\mathbf{E}(\bar{\mathbf{p}}; \sigma)) d\bar{\mathbf{p}} \\ &= \text{vol}(\bar{\Omega}^\tau) \psi^{\tilde{\mathbf{E}}} \left(\frac{\sigma^2-1}{2}, 0, 0 \right) \\ &= \ell_0 w_0 \tau \frac{a_{11}}{2} \eta_1 \left(\left(\frac{\sigma^2-1}{2} \right)^2 \right). \end{aligned} \quad (3.11)$$

The computation for weft stretch is analogous and yields

$$\Psi(\sigma) = \tau \ell_0 w_0 \frac{a_{22}}{2} \eta_3 \left(\left(\frac{\sigma^2-1}{2} \right)^2 \right). \quad (3.12)$$

For the approximate bias stretch equilibrium, the piecewise-constant strain necessitates a separate term in the energy computation per strain region. In all regions, the deformation gradients map $\mathbf{e}_3 \mapsto \mathbf{e}_3$. In regions \bar{A}_1 and \bar{A}_2 , the deformation gradient is just an identity matrix and so there is no energy contribution. For the region \bar{C} , inspection of a central horizontal line before and after deformation shows that the deformation gradient $\mathbf{F}^C = \mathbf{F}^C(\sigma)$ maps $\mathbf{e}_1 \mapsto \frac{\ell-w_0}{\ell_0-w_0} \mathbf{e}_1 = \frac{\ell_0\sigma-w_0}{\ell_0-w_0} \mathbf{e}_1$. The kinematic assumptions require that every region undergoes a pure shear with respect to warp and weft, so $\|\mathbf{F}^C \mathbf{d}_1\| = 1$. By vertical symmetry we also know $\mathbf{F}^C \mathbf{e}_2$ must be a multiple of \mathbf{e}_2 . These facts are sufficient to uniquely determine \mathbf{F}^C :

$$\mathbf{F}^C(\sigma) = \begin{pmatrix} \frac{\ell_0\sigma-w_0}{\ell_0-w_0} & 0 & 0 \\ 0 & \sqrt{2 - \left(\frac{\ell_0\sigma-w_0}{\ell_0-w_0} \right)^2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The resulting reduced strain $\tilde{\mathbf{E}}^C$ can be simplified as

$$\begin{aligned}\tilde{\mathbf{E}}^C(\sigma) &= \begin{pmatrix} 0 & \tilde{E}_{12}^C(\sigma) \\ \tilde{E}_{12}^C(\sigma) & 0 \end{pmatrix} \\ \tilde{E}_{12}^C(\sigma) &= \frac{1}{2} \left(\left(\frac{\ell_0 \sigma - w_0}{\ell_0 - w_0} \right)^2 - 1 \right).\end{aligned}\tag{3.13}$$

The deformation gradient \mathbf{F}^{B_1} for region $\overline{B_1}$ is determined using continuity across the shared boundaries with regions $\overline{A_1}$ and \overline{C} , which requires $\mathbf{F}^{B_1}(\mathbf{e}_1 - \mathbf{e}_2) = \mathbf{e}_1 - \mathbf{e}_2$ and $\mathbf{F}^{B_1}(\mathbf{e}_1 + \mathbf{e}_2) = \mathbf{F}^C(\mathbf{e}_1 + \mathbf{e}_2)$. These requirements fully determine \mathbf{F}^{B_1} , and the resulting reduced strain is

$$\begin{aligned}\tilde{\mathbf{E}}^{B_1}(\sigma) &= \begin{pmatrix} 0 & \tilde{E}_{12}^{B_1}(\sigma) \\ \tilde{E}_{12}^{B_1}(\sigma) & 0 \end{pmatrix} \\ \tilde{E}_{12}^{B_1}(\sigma) &= \frac{1}{4} \left(\frac{\ell_0 \sigma - w_0}{\ell_0 - w_0} - \sqrt{2 - \left(\frac{\ell_0 \sigma - w_0}{\ell_0 - w_0} \right)^2} \right).\end{aligned}\tag{3.14}$$

Due to vertical symmetry, the regions $\overline{B_2}$, $\overline{B_3}$, and $\overline{B_4}$ share the same reduced strain, heretofore written simply as $\tilde{\mathbf{E}}^B$ with nonzero entry \tilde{E}_{12}^B ; the combined region is notated $\overline{B} = \cup_j \overline{B_j}$. The hyperelastic potential can now be constructed in terms of Equation (3.13) and Equation (3.14):

$$\begin{aligned}\Psi(\sigma) &= \int_{\overline{\Omega}^\tau} \psi^{\mathbf{E}}(\mathbf{E}(\overline{\mathbf{p}}; \sigma)) d\overline{\mathbf{p}} \\ &= \text{vol}(\overline{B}) \psi^{\tilde{\mathbf{E}}} (0, \tilde{E}_{12}^B(\sigma), 0) + \text{vol}(\overline{C}) \psi^{\tilde{\mathbf{E}}} (0, \tilde{E}_{12}^C(\sigma), 0) \\ &= \tau w_0^2 G_{12} \eta_4 (\tilde{E}_{12}^B(\sigma))^2 + \tau (\ell_0 w_0 - \frac{3}{2} w_0^2) G_{12} \eta_4 (\tilde{E}_{12}^C(\sigma))^2\end{aligned}\tag{3.15}$$

3.4.3 Implementation and predictive accuracy

The final model force \bar{f}^c is constructed by substituting into the power balance Equation (3.10) using $\Psi(\sigma)$ from Equation (3.11) (for warp experiments), Equation (3.12) (for weft), or Equation (3.15) (for bias). An approximated version of Equation (3.5) is then created by

substituting the approximate model forces \bar{f}^c in place of the model forces from finite element analysis \check{f}^c . The resulting minimization problem is solved numerically by a Wolfram Mathematica [Inc16] routine using Newton’s method with a line search. The simplicity of the approximated objective makes the solver runtime negligible even with a generic initial guess, while the symbolic algebra capabilities of Mathematica mean that the formulas we have provided for $\Psi(\sigma)$ are sufficient to implement the solver without requiring analytic formulas for $\frac{\partial\Psi(\sigma)}{\partial\sigma}$.

The usefulness of the approximated problem’s solution as an initial guess \mathbf{k}^0 for the full optimization problem depends on the accuracy of the assumed equilibria. Figure 3.6 provides sample quasistatic equilibria computed in the finite element solver for intuition purposes. Of course, the truly important factor is not visual similarity but rather the difference between forces in the two equilibria.

Comparison of Figures 3.4 and 3.6 reveals that the warp stretch equilibrium almost perfectly matches the assumed kinematics. In fact, for warp and weft stretches the forces computed using approximate equilibria typically come within 1% of the forces obtained using a full finite element quasistatic solve; the worst observed errors rise up near 3%. This means our approximate optimization problem is a very good model for the full solve, and therefore the initial guesses \mathbf{k}^0 we obtain are very close to the final fitted parameters \mathbf{k} . These initial guesses greatly reduce solver runtime for the phases which fit warp and weft stretch data.

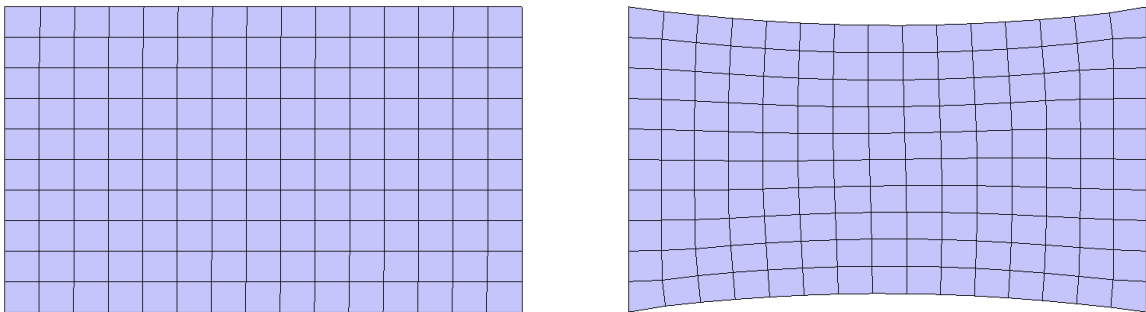


Figure 3.6: *Finite element equilibria for warp (left) and bias (right) tensile tests at a deformation level of 15%. The approximate kinematics (Figs. 3.4, 3.5) predict the warp equilibrium almost exactly but are inaccurate for the bias equilibrium.*

The bias stretch equilibria in Figures 3.5 and 3.6 are visually similar but clearly different. In our experiments, the forces from the approximate kinematics are frequently stronger than the finite element version by a factor of 1.25 or more. Due to this discrepancy, the approximate problem’s solution is not beneficial as an initial guess for the main solver, and in fact often provides worse performance than starting from generic starting parameters. [DDB11] observe the same inaccuracy and suggest multiplying the approximate equilibrium’s forces \bar{f}^c by an ad-hoc scalar to better approximate the true equilibrium forces \check{f}^c . However, outside of the small strain regime our tests show the nonlinear behaviors of \bar{f}^c and \check{f}^c cannot be related by a simple linear multiplier. As a result of these difficulties, we are forced to discard the approximate equilibrium approach for the bias stretch test, accepting a lengthier runtime cost for the corresponding solver phase.

The approximate bias stretch equilibrium described in this section only makes sense if the initial strip aspect ratio satisfies $\frac{\ell_0}{w_0} \geq 2$. This criterion is violated for our canvas, cotton, and silk datasets. [WPP98] provides a different version of the equilibrium for aspect ratios $\frac{\ell_0}{w_0} < 2$. We have not tested initial guesses created using that modified assumption but we expect that approach would encounter similar difficulties.

We are not aware of any plausible approximate equilibrium for the bending experiment. Instead, we choose the thickness guess to be the experimentally measured thickness. With this guess, the solver still incurs a significant runtime cost but performance is improved over choosing the same generic thickness guess for all materials.

3.5 Main fitting solver

For the final optimization steps in §3.3.2 we explored using both BFGS and Gauss-Newton methods, together with a line search protocol which guarantees satisfaction of the strong Wolfe conditions. Here we discuss the steps needed in computing the functional $\varepsilon(\mathbf{k})$ and its derivatives as needed in these approaches, as well as various subtle points which arise in building an efficient and robust solver for this optimization problem. The resulting iterative

minimization routine is summarized in Algorithm 3.1.

3.5.1 Equilibrium configuration

Given a mesh configuration \mathbf{q} , the corresponding clamp forces (for tensile tests) or drape angle (for bend test) can be computed as a simple function of \mathbf{q} . However, our optimization objective function $\varepsilon = \varepsilon(\mathbf{k})$ depends on various simulated *equilibrium* measurements $\hat{p}_i(\mathbf{k})$. In other words, to evaluate the terms $\hat{p}_i(\mathbf{k})$ appearing in the definition of ε (at each new solver step or line search iteration) we must compute the associated mesh equilibrium configurations.

We use $\mathbf{q}^e = \mathbf{q}^e(\mathbf{k})$ and $\boldsymbol{\lambda}^e = \boldsymbol{\lambda}^e(\mathbf{k})$ to denote the equilibrium configuration and Lagrange multipliers determined by the parameters \mathbf{k} . These are obtained by solving the force balance in the absence of inertia and damping (see Chapter Two for derivation of the force balance system) while satisfying the linear boundary condition constraints. The resulting discretized nonlinear system is

$$\begin{aligned} \mathbf{f}(\mathbf{q}^e; \mathbf{k}) + \mathbf{B}^T \boldsymbol{\lambda}^e &= 0 \\ \mathbf{B} \mathbf{q}^e &= \mathbf{b}. \end{aligned} \tag{3.16}$$

As defined in §2.5, $\mathbf{f} = \mathbf{f}(\mathbf{q})$ denotes the combination of elastic and gravity forces; here, we write $\mathbf{f}(\mathbf{q}; \mathbf{k})$ to emphasize that \mathbf{f} also depends on the model parameters. We adapt the notation $\psi^{\mathbf{q}}(\mathbf{q})$ to $\psi^{\mathbf{q}}(\mathbf{q}; \mathbf{k})$ for the same reason. \mathbf{B} and \mathbf{b} express the boundary conditions. For a tensile experiment, these boundary conditions describe the clamped edges held fixed at the appropriate stretch distance. For a bend experiment, a clamped constraint is chosen along one edge of the cloth to simulate draping over the edge of a fixed plane without relying on cloth collisions.

3.5.1.1 Quasistatic solver implementation

The fitting process requires solving many quasistatics problems of the form in Equation (3.16) for each BFGS iteration. For example, if a fitting solve attempts to match 16 data points

Algorithm 3.1 Structure of the BFGS fitting solver

Require: function `quasistatic_equilibria(\mathbf{k}, \mathbf{q})` returning equilibria \mathbf{q}^e for all solver meshes (§3.5.1)

Require: function `evaluate_objective_and_gradient(\mathbf{k}, \mathbf{q})` returning ε and $\frac{\partial \varepsilon}{\partial \mathbf{k}}$ (§3.5.2)

Input: initial guess \mathbf{k}^0 (from §3.4, or from a previous BFGS solve as per §3.5.3)

```
1:  $\mathbf{q}^b \leftarrow$  basic configurations for all meshes (§3.5.6)
2:  $n \leftarrow 0$ 
3: bfgsNeedsReset  $\leftarrow$  true
4: stepsSinceBfgsReset  $\leftarrow 0$ 
5: loop // main BFGS loop
6: // Set starting configuration for this BFGS iteration's quasistatic solves. (§3.5.6)
7: if bfgsNeedsReset then
8:    $\mathbf{q}^n \leftarrow$  quasistatic_equilibria( $\mathbf{k}^n, \mathbf{q}^b$ )
9: else
10:   $\mathbf{q}^n \leftarrow$  quasistatic_equilibria( $\mathbf{k}^n, \mathbf{q}^{n-1}$ )
11: end if
12:
13: // Set approximate inverse Hessian  $\mathbf{Q}$ . (§3.5.5)
14:  $(\varepsilon, \frac{\partial \varepsilon}{\partial \mathbf{k}}) \leftarrow$  evaluate_objective_and_gradient( $\mathbf{k}^n, \mathbf{q}^n$ )
15: if bfgsNeedsReset then
16:    $\mathbf{Q} \leftarrow$  inverse of Gauss-Newton's approximate Hessian
17:   bfgsNeedsReset  $\leftarrow$  false
18: else
19:   Update  $\mathbf{Q}$  based on the most recent step using BFGS rules.
20: end if
21:
22: // Compute and take the next step.
23:  $\Delta \mathbf{k} \leftarrow -\mathbf{Q} \frac{\partial \varepsilon}{\partial \mathbf{k}}$ 
24:  $\alpha \leftarrow$  step scalar from line search (§3.5.7)
25:  $\mathbf{k}^{n+1} \leftarrow \mathbf{k}^n + \alpha \Delta \mathbf{k}$ 
26: stepsSinceBfgsReset  $\leftarrow$  stepsSinceBfgsReset + 1
27:
28: // Check for sufficient progress and halting criterion. (§3.5.5)
29: if stepsSinceBfgsReset  $\geq 5$  and last 5 steps give total reduction  $< 0.1\%$  then
30:   if stepsSinceBfgsReset = 5 then
31:     halt execution and return  $\mathbf{k}^n$ .
32:   else
33:     bfgsNeedsReset  $\leftarrow$  true
34:     stepsSinceBfgsReset  $\leftarrow 0$ 
35:   end if
36: end if
37:  $n \leftarrow n + 1$ 
38: end loop
```

from a bias stretch experiment, then 16 distinct cloth meshes must be maintained because each data point corresponds to a distinct stretch level and thus requires computation of a different equilibrium configuration. Then at each objective value or gradient evaluation, the cloth meshes for every data point must all be evolved to equilibrium states based on the current iterate for \mathbf{k} . As a result, the position vectors \mathbf{q}^n , \mathbf{q}^b , \mathbf{q}^e , \mathbf{q}^{curr} , and \mathbf{q} in Algorithms 3.1 and 3.2 should all be understood as referring to the combined list of positions for all relevant meshes.

Each nonlinear quasistatic problem is solved using Newton’s method to minimize the total of hyperelastic and gravitational potential cloth energy subject to the given linear constraints. As described for dynamic simulation problems in §2.7, the Newton solver is augmented by a line search procedure which ensures each step satisfies the strong Wolfe conditions. The linearization of Equation (3.16) is a KKT system; as in §2.8, these symmetric indefinite systems are solved using the PARDISO routine in Intel’s MKL. Unfortunately, the linear systems resulting from the quasistatic problem often exhibit condition numbers on the order of 10^{10} , which easily can lead to inaccurate solutions. As a result the Newton step can sometimes be an ascent direction for the energy or otherwise fail to produce significant progress, even when the current iterate is visibly not a local energy minimizer. The solver escapes from such situations by taking several backward Euler time steps before returning to the Newton iteration process. Comparison of Equation (2.35) with Equation (3.18) shows that the linear systems in backward Euler steps benefit from a reduced condition number since they include the additional positive definite, diagonally dominant term $h^{-2}\mathbf{M}$. This use of backward Euler simulation as a failsafe is triggered anytime three consecutive Newton steps give a combined energy reduction of 0.1% or less, or immediately if Newton chooses an ascent direction for the cloth energy. A potential alternative is to leverage the method in [VM07].

As mentioned above, cloth quasistatics problems tend to be extremely ill-conditioned. Thus it is critical to retain as much precision as possible. We address this in three steps. First, we upgrade OpenSubdiv from single to double precision arithmetic. Second, we use iterative

refinement in our linear solves, using quad precision to store the residual. Finally, we are careful to choose a sufficient number of Gaussian quadrature points. We find 4×4 quadrature points to be sufficient, based on the heuristic discussion in §2.5 and also on numerical testing based on a quasistatic solver implementation with arbitrary Gauss quadrature order. Despite these considerations, there still exist fairly simple quasistatics problems on which our solver fails to make progress. However, our solver is robust enough to handle all problems which arise during the course of the model fitting procedure.

3.5.2 Differentiating the equilibrium configuration

To minimize $\varepsilon(\mathbf{k})$, both BFGS and Gauss-Newton require differentiation of each term in the sum in Equation (3.5). That is, for a given experimental measurement \check{p} , we must compute

$$\frac{\partial}{\partial \mathbf{k}} \left(\frac{(\hat{p}(\mathbf{k}) - \check{p})^2}{\check{p}^2} \right). \quad (3.17)$$

Each simulated force $\hat{f}(\mathbf{k})$ is a linear function of the corresponding equilibrium Lagrange multipliers $\boldsymbol{\lambda}^e$. Simulated angles $\hat{\theta}(\mathbf{k})$ can be written as a linear function of \mathbf{q}^e (which specifies the horizontal and vertical distance between two ends of the cloth) composed with an inverse trigonometric function. In either case, the main difficulty of evaluating Equation (3.17) lies in computation of $\frac{\partial \mathbf{q}^e}{\partial \mathbf{k}}$.

The function $\mathbf{q}^e(\mathbf{k})$ is defined implicitly via the system in Equation (3.16). To compute its derivative we must differentiate the entire equilibrium system. Note that the term $\mathbf{f}(\mathbf{q}^e; \mathbf{k})$ depends on \mathbf{k} both through $\mathbf{q}^e = \mathbf{q}^e(\mathbf{k})$ and through the usual dependence of elastic forces on the model parameters. The differentiated system is

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}^e}{\partial \mathbf{k}} + \frac{\partial \mathbf{f}}{\partial \mathbf{k}} + \mathbf{B}^T \frac{\partial \boldsymbol{\lambda}^e}{\partial \mathbf{k}} &= 0 \\ \mathbf{B} \frac{\partial \mathbf{q}^e}{\partial \mathbf{k}} &= 0. \end{aligned}$$

Here $\frac{\partial \mathbf{f}}{\partial \mathbf{q}} = -\mathbf{K}$ is the negative Hessian of the elastic potential V , and $\frac{\partial \mathbf{f}}{\partial \mathbf{k}}$ can be computed

directly. Then the unknowns $\frac{\partial \mathbf{q}^e}{\partial \mathbf{k}}$ and $\frac{\partial \lambda^e}{\partial \mathbf{k}}$ can be obtained by solving the linear system

$$\begin{pmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \frac{\partial \mathbf{q}^e}{\partial \mathbf{k}} \\ -\frac{\partial \lambda^e}{\partial \mathbf{k}} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial \mathbf{k}} \\ \mathbf{0} \end{pmatrix}. \quad (3.18)$$

The matrices $\frac{\partial \mathbf{q}^e}{\partial \mathbf{k}}$ and $\frac{\partial f}{\partial \mathbf{k}}$ each have $|\mathbf{k}|$ columns. Thus Equation (3.18) really represents $|\mathbf{k}|$ linear systems, each with the same system matrix but a different right-hand side. Once again, the KKT systems are addressed using Intel's PARDISO routine (see §2.8), which operates by explicit matrix factorization followed by back-substitution. As a result consecutive solves of the same system are much faster, since only the back-substitution phase must be repeated. Each step in §3.3.2 requires at most $2d_j$ free parameters in \mathbf{k} , where d_j depends on the degree of curve η_j being fitted. In this project all final fitted degrees satisfy $d_j \leq 5$, so there are at most 10 distinct columns of $\frac{\partial \mathbf{q}^e}{\partial \mathbf{k}}$. As a result of these factors, the total solve time for the $|\mathbf{k}|$ linear systems in Equation (3.18) is only slightly more costly than a single linear solve, and overall is dwarfed by the much more expensive computations of \mathbf{q}^e .

For meshes associated with a force measurement $\hat{f}(\mathbf{k})$, the linear solution process could be optimized slightly using the adjoint method [GP00]. Since $\hat{f}(\mathbf{k})$ is a linear function of λ^e , one need not fully evaluate $\frac{\partial \mathbf{q}^e}{\partial \mathbf{k}}$ and $\frac{\partial \lambda^e}{\partial \mathbf{k}}$ if the product $\frac{\partial \hat{f}}{\partial \lambda^e} \frac{\partial \lambda^e}{\partial \mathbf{k}}$ can be computed directly. Indeed, the desired product can be reformulated starting from

$$\frac{\partial \hat{f}}{\partial \lambda^e} \frac{\partial \lambda^e}{\partial \mathbf{k}} = - \begin{pmatrix} \mathbf{0} & \frac{\partial \hat{f}}{\partial \lambda^e} \end{pmatrix} \begin{pmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial f}{\partial \mathbf{k}} \\ \mathbf{0} \end{pmatrix}$$

by transposing both sides to obtain

$$\left(\frac{\partial \hat{f}}{\partial \lambda^e} \frac{\partial \lambda^e}{\partial \mathbf{k}} \right)^T = - \begin{pmatrix} \frac{\partial \hat{f}}{\partial \lambda^e}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ \frac{\partial f}{\partial \lambda^e}^T \end{pmatrix}.$$

The transformed right-hand side can be evaluated with just one KKT matrix solve since $\frac{\partial \hat{f}}{\partial \lambda^e}^T$ is a single column vector. The adjoint method can also be applied to the bending tests

by a similar analysis. The simulated drape angle can be computed from the horizontal and vertical distance between two ends of the cloth, which are both scalar linear functions of \mathbf{q}^e ; thus the adjoint method requires 2 linear solves instead of $|\mathbf{k}|$. However, as discussed in §3.3.2, the bend test fitting phase solves for only one parameter, so the adjoint method is actually slower in this case. In fact, we do not implement the adjoint approach even for tensile tests, simply because the runtime of $|\mathbf{k}|$ linear solves with the same system is insignificant to the overall fitting solver runtime.

3.5.2.1 Parameter derivatives of force

Computation of the parameter derivatives $\frac{\partial f}{\partial \mathbf{k}}$ requires a method for constructing $\frac{\partial^2 \psi^{\mathbf{q}}(\mathbf{q}; \mathbf{k})}{\partial \mathbf{q} \partial \mathbf{k}}$, which will then be used in a quadrature integral approximation based on Equation (2.24). To obtain the necessary energy density derivatives, we differentiate Equation (2.36) with respect to \mathbf{k} , obtaining

$$\frac{\partial^2 \psi^{\mathbf{q}}}{\partial \mathbf{q} \partial \mathbf{k}} = \frac{\partial^2 \psi^{\mathbf{E}}}{\partial \mathbf{E} \partial \mathbf{k}} \frac{\partial \mathbf{E}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{q}}$$

having noted that $\frac{\partial \mathbf{E}}{\partial \mathbf{z}}$ and $\frac{\partial \mathbf{z}}{\partial \mathbf{q}}$ are kinematics-based quantities with no dependence on the chosen constitutive model or its parameters \mathbf{k} . The \mathbf{k} -derivative of Equation (2.40) reveals

$$\frac{\partial^2 \psi^{\mathbf{E}}}{\partial E_{ab} \partial \mathbf{k}} = D_{qa} \frac{\partial^2 \psi^{\tilde{\mathbf{E}}}}{\partial \tilde{E}_{qr} \partial \mathbf{k}} D_{rb}.$$

Finally, the derivatives $\frac{\partial^2 \psi^{\tilde{\mathbf{E}}}}{\partial \tilde{E}_{qr} \partial \mathbf{k}}$ are computed by the chain rule from Equation (3.1) together with formulas for the \mathbf{k} -derivative of Equation (3.2):

$$\begin{aligned} \frac{\partial \eta'_j(x)}{\partial \mu_{ji}} &= (x+1)^{\alpha_{ji}-1} \\ \frac{\partial \eta'_j(x)}{\partial \alpha_{ji}} &= \mu_{ji} (x+1)^{\alpha_{ji}-1} \log(x+1) \end{aligned}$$

The implementation details for combining the various chain rule steps are analogous to the computation for $\frac{\partial^2 \Psi}{\partial \mathbf{q} \partial \mathbf{q}}$, and are thus incorporated into the same implemented proce-

dures. Adding the computation of $\frac{\partial^2 \psi}{\partial \mathbf{q} \partial \mathbf{k}}$ incurs a significant runtime cost as compared to just computing $\frac{\partial^2 \psi}{\partial \mathbf{q} \partial \mathbf{q}}$, and most energy Hessian evaluations do not require a corresponding evaluation of $\frac{\partial \mathbf{f}}{\partial \mathbf{k}}$. After all, $\frac{\partial \mathbf{f}}{\partial \mathbf{k}}$ is only computed once per quasistatic equilibrium obtained during a fitting solve, and is never needed for everyday simulation purposes, while the energy Hessian \mathbf{K} must be updated at every Newton step during both quasistatic solves and simulations. Thus similarly to the Hessian-free energy computation described in §2.11.5, the energy computation accepts a boolean flag which controls whether the force derivatives must be computed, and skips the corresponding computations and memory allocations if possible. In this way, the same energy computation routines are able to efficiently handle any combination of energy derivatives which might need to be computed.

3.5.3 Multiresolution meshing and curve sampling

Cloth simulation based on subdivision surface finite elements has the advantage of higher-order convergence under mesh refinement as compared to simpler finite elements or approximations thereof. We use this feature to speed up our fitting process. In particular, we run each fitting solve using a comparatively coarse mesh resolution to describe each simulated experiment. The resulting parameters are then used as an initial guess for a higher resolution fitting solve. This process may be repeated as many times as desired to obtain satisfactory convergence of high-resolution fitting solves more quickly than would be possible without such a means of generating initial guesses. In practice, we use this technique only for the solves involving tensile data. For canvas, cotton and silk, we use an 8×12 grid of finite elements for our coarsest meshes. Then we progress upward through resolutions 16×24 and finally 32×48 to refine the fit. For denim and wool we use 8×24 , 16×48 , and 32×96 grids.

Due to variance in the sampling rate during tensile experiments as well as different final stretches sustained before material failure, the experimental curves comprise numbers of data points ranging from 80 to 250. According to the optimization objective ε , each of these points should be weighted equally in the fitting process. However, as previously mentioned the BFGS solver runtime per objective evaluation is dominated by the cost of

computing quasistatic equilibria, and thus the cost is nearly linear in the number of data points being fit. To take advantage of this phenomenon, we first run initial fitting solves at each mesh resolution using only 16 uniformly sampled points from the experimental curve. The resulting parameters are then used as an initial guess for a refinement fit at the highest mesh resolution which includes all experimental data points. The number 16 is chosen purely by experimenting to make the initial fits proceed at a reasonable speed while also ensuring that the resulting parameters are sufficiently accurate to make the refinement fit with all data points converge within a few BFGS steps. A potential future improvement could be to automate the process of fitting at several sampling densities. More testing would be required to decide whether fitting many sample densities in sequence provides runtime benefits, or whether the current method of jumping from 16 points directly to the full dataset already reaps most of the available efficiency benefits.

3.5.4 BFGS vs. Gauss-Newton

The complexity of the fitting solves depends heavily on the choice of the degree d_j for each function η_j in the energy definition. For degree 1, either optimization method is reliable. However, Gauss-Newton frequently stalls in fitting curves of degree 2 or larger, repeatedly choosing search directions that allow virtually no progress. BFGS is robust in solving degree 2 and 3 problems, and successfully handles problems of degrees 4 and 5 when using the initial guesses specified in §3.4. We generally find degree 2 or 3 to be the minimum requirement for accurate fitting of our datasets. Thus the improved performance of BFGS as compared to Gauss-Newton is crucial to building a successful solver.

3.5.5 BFGS implementation

The BFGS algorithm requires an initial approximation of the objective function's inverse Hessian as an input. Since we implement both BFGS and Gauss-Newton methods, we simply compute and explicitly invert the Gauss-Newton approximate Hessian. Each BFGS

execution solves for 10 or fewer unknown parameters, so the runtime cost of inverting the Hessian is immaterial. Alternatively, we obtain successful convergence by choosing the initial Hessian to be the $|\mathbf{k}| \times |\mathbf{k}|$ identity matrix. However, the identity guess generally leads to very slow progress for the first few iterations, and so the Gauss-Newton approximation is recommended for efficiency reasons.

The accumulation of the approximate inverse Hessian is allowed to proceed as long as the solver continues to make progress. Once the optimization function fails to decrease by 0.1% over five consecutive steps, the inverse Hessian is reset to the Gauss-Newton initial guess. When no significant progress is made within the first five steps after a BFGS reset, the solve halts and if applicable restarts with the next higher mesh resolution. Experimentation has shown that when BFGS no longer makes progress, adding random perturbations to the parameters as suggested in [WOR11] in hopes of escaping from a local minimum is rarely beneficial. The fitted models often include very large - and thus sensitive - parameters α_{ij} and μ_{ij} , so blindly tweaking the parameters can yield huge energies from which the solver has a difficult time recovering. Tiny perturbations can avoid this effect, but also are usually insufficient for escaping from the offending local minimum. As a result, we do not use this perturbation idea except on the final dense mesh in the multiresolution approach.

3.5.6 Multi-valued objective

The objective function ε is written in Equation (3.5) as a function $\varepsilon(\mathbf{k})$, but in reality this is a slight abuse of notation. Due to the dependence on computation of equilibria, the objective value is also affected by the initial positions chosen for those quasistatic solves, since many of the data points have multiple local energy minima which could serve as equilibrium positions. To see this, consider the bias stretch test, which commonly induces out-of-plane buckling. Quasistatic problems featuring buckling almost universally possess at least two possible equilibria, since the vertical reflection of a local energy minimizer is generally very close to another minimum. Fortunately, our tests indicate that for experiments in our datasets, the various possible equilibria nearly always exhibit force measurements within 0.5%. This

provides some justification for designing the fitting method as if ε is a single-valued function, but caution must still be taken.

For example, assuming ε is single-valued suggests a runtime optimization for the quasistatic solves. Suppose the objective function was recently evaluated at \mathbf{k} , so that for each data point an equilibrium mesh $\mathbf{q}^e(\mathbf{k})$ has been computed. Then for the next objective function evaluation at $\tilde{\mathbf{k}}$, the solver can use $\mathbf{q}^e(\mathbf{k})$ as an initial guess when computing $\mathbf{q}^e(\tilde{\mathbf{k}})$. The resulting guesses prove quite successful in improving runtime because the parameters usually do not change drastically between consecutive evaluations.

However, this initial guess method means that evaluating ε at the same \mathbf{k} twice will yield different results, especially if a drastically different parameter set $\tilde{\mathbf{k}}$ has been evaluated in between. This creates problems during the line search algorithm, which is carefully designed to narrow down certain ranges which are guaranteed to contain a step size satisfying the strong Wolfe conditions. The BFGS solver occasionally experiences iterations with small progress, and during such steps the multi-valued nature of the computed equilibria can cause the line search to restrict its domain to a region with no valid step sizes.

At the other extreme, the problem with multi-valued ε can be avoided entirely by simply choosing one basic configuration for each mesh at the start of the solve – for example, the constant strain configuration from Figure 3.4 – and using that configuration as the initial guess every time that mesh must be brought to equilibrium. In this way, ε is made truly single-valued, since the computed equilibria no longer depend on previously computed values \mathbf{k} . The resulting method does not suffer from the aforementioned problems during line search, but a heavy cost is paid in runtime: evaluating at several nearby values \mathbf{k} requires solution of extremely similar quasistatic problems, using the same poor initial guess each time.

Fortunately, a middle ground is available which provides fast solves and also robust line search. The multi-valued nature of ε only causes problems through inconsistent values computed during the line search procedure. Our solution is to choose a consistent initial guess \mathbf{q}^n at the start of each BFGS iteration; thus if the current BFGS iterate is \mathbf{k}^n and we wish to compute the next step \mathbf{k}^{n+1} , all line search evaluations of ε will use the initial

guess configuration \mathbf{q}^n . In this way we ensure that each line search deals with a single-valued function while still allowing the choice of a useful initial guess.

In general, \mathbf{q}^n is chosen as the quasistatic solution obtained by starting from positions \mathbf{q}^{n-1} and using parameters \mathbf{k}^n . However, a different procedure is used on the very first BFGS step as well as immediately after the BFGS approximate inverse Hessian is reset. For those cases we first define a basic configuration \mathbf{q}^b for each mesh, namely a constant strain configuration for tensile tests and a flat (undeformed) configuration for bending tests. Then \mathbf{q}^n is selected as the quasistatic equilibrium computed by starting from \mathbf{q}^b with parameters \mathbf{k}^n . The effect of this periodic reversion to the basic configuration is to prevent an undesirable local energy minimizer from propagating through the entire BFGS solve.

3.5.7 Step size management

The objective function can be very sensitive to slight changes in the parameters, and especially in the parameters μ_{ij} and α_{ij} which control the shape of the nonlinear functions η_i . As a result, evaluations of ε may simply fail if the line search step attempted is too large. For example, the nature of our parameterization for η_i allows unbounded negative energies to be represented, and ε can certainly not be evaluated if the associated quasistatic problems reach such poorly behaved configurations.

This problem is addressed in two parts. First, the line search is modified to detect failing quasistatic solves and fail gracefully when necessary. If failure is encountered, the solver considers the successful evaluations already performed during the current line search, and returns whichever step size gave the largest objective function reduction. If no evaluation performed thus far gave a reduction in ε (for example, if failure was encountered on the very first evaluation of this line search) then the search direction is multiplied by 0.1 and the line search is restarted. Sufficiently small steps will always allow evaluation of ε , and even if this process sometimes leads to a small step in \mathbf{k} , BFGS tends to recover within a few iterations and make significant progress once again.

Algorithm 3.2 Specialized line search for BFGS fitting solver

Input: starting configuration \mathbf{q}^n , current parameters \mathbf{k}^n

Input: search direction $\Delta\mathbf{k}$

- 1: Scale down $\Delta\mathbf{k}$ if necessary so no parameter changes by more than 25%.
 - 2: Run Wolfe conditions line search as described in [NW06].
 - 3: **for each** evaluation site \mathbf{k}^{curr} required by line search **do**
 - 4: $\mathbf{q}^{\text{curr}} \leftarrow \text{quasistatic_equilibria}(\mathbf{k}^{\text{curr}}, \mathbf{q}^n)$
 - 5: $(\varepsilon, \frac{\partial\varepsilon}{\partial\mathbf{k}}) \leftarrow \text{evaluate_objective_and_gradient}(\mathbf{k}^{\text{curr}}, \mathbf{q}^{\text{curr}})$
 - 6: Compute derivative along search direction from $\frac{\partial\varepsilon}{\partial\mathbf{k}}$.
 - 7: **end for**
 - 8: **if** some line search evaluation cannot be completed **then**
 - 9: **if** at least one objective-reducing \mathbf{k}^{curr} has been found during this line search **then**
 - 10: **return** the step size which produced the largest reduction found thus far.
 - 11: **else**
 - 12: Multiply $\Delta\mathbf{k}$ by 0.1 and restart this line search algorithm.
 - 13: **end if**
 - 14: **else** // Line search completed successfully.
 - 15: **return** the computed step size.
 - 16: **end if**
-

The second part of our fix is search direction clamping. The search direction initially passed into line search is not allowed to change any parameter by more than 25%; if necessary, the entire search direction is rescaled to obey this clamping criterion. This protects the solver from running lots of failed evaluations in case of a wildly oversized BFGS step, which can sometimes occur in the first few iterations of a fitting solve. Our line search implementation uses the passed-in search direction as its first guess, but is capable of returning longer step sizes if necessary; in fact, the search interval is extended by doubling as necessary to ensure efficiency even when the optimal step is many times the size of the search direction. Thus in cases where the clamping is more restrictive than necessary, the line search still quickly produces an acceptable step, which is often 30 or more times the clamped size. Of course, if the doubling step sizes eventually jump too far and an evaluation of ε fails, the graceful failure approach described above will still allow selection of a fairly large successful step. The choice of 25% as the relative step size for clamping is based on hand-tuning over the course of several fitting solves, weighing the cost of failed ε evaluations due to underclamping against the cost of additional line search evaluations due to clamping too severely.

Fabric	Density [g/m ²]	Thickness [mm]	Thread count [ends/” × picks/”]	Weave
Canvas	294.0	0.53	108 × 56	Basket
Cotton	103.6	0.18	140 × 86	Plain
Silk	83.0	0.18	653 × 123	Satin
Wool	480.6	1.28	32 × 28	Twill
Denim	400.0	0.66	70 × 47	Twill

Table 3.2: *Measured properties of the test materials.*

These augmentations of the Wolfe conditions line search are summarized in Algorithm 3.2.

3.6 Results

The material model and fitting procedure are demonstrated by fitting parameters for five materials: canvas, cotton poplin, silk charmeuse, wool coating, and denim. Table 3.2 summarizes measurable properties of the materials, and Table 3.3 provides the final fitted parameter sets. Our target criterion for a successful fit is for the magnitude of fitting error to be smaller than the unavoidable error sources inherent in our framework. The most significant of these is experimental error: the relative difference between 5 executions of the same tensile experiment is often 15-20% for a fixed elongation. Smaller errors are introduced through the numerical discretization, nonlinear solver residual, Kirchhoff-Love hypotheses, and interpretation of data from slowly-moving tensile experiments as representing quasistatic equilibria.

We provide plots comparing experimental vs modeled forces in Figure 3.7. Quantitatively, the accuracy of fitted curves is judged by computing the average relative error over the domain. This is just a rescaling of ℓ^1 error to view the result as a percentage. The fitting process produces average relative error under 5% in most cases; see Table 3.4. Thus our fits are well within the level of experimental error in the data.

Parameter	Canvas	Cotton	Silk	Wool	Denim
τ	0.370	0.253	0.239	0.815	0.565
a_{11}	5.366	15.557	4.300	2.290	4.793
d_1	2	3	2	5	5
μ_{12}	-1880.386	4.952	-60.769	14.589	-1482.055
μ_{13}		348.919		-169.472	276.426
μ_{14}				8.812	2407.671
μ_{15}				274.830	3416.789
α_{11}	22.619	-63.206	-2.230	-85.275	-7.809
α_{12}	22.082	-550.784	-115.578	-0.777	10.804
α_{13}		-54.525		-79.055	18.952
α_{14}				-294.608	-12.887
α_{15}				-79.058	1.690
a_{12}	2.683	7.779	4.971	1.145	4.515
d_2	1	1	1	1	1
α_{21}	1	1	1	1	1
a_{22}	19.804	25.004	9.942	2.219	9.029
d_3	3	3	2	5	5
μ_{32}	70.532	-285.733	-12.460	9.062	0.658
μ_{33}	-41.905	2873.813		0.422	-185.695
μ_{34}				9295.356	-45.815
μ_{35}				16171.138	146.977
α_{31}	212.682	109.194	-2.612	-63.857	12.714
α_{32}	149.080	102.720	-195.533	-0.0174	-12802.045
α_{33}	69.532	108.583		-18575.181	17.907
α_{34}				-63.857	30.067
α_{35}				-63.884	24.683
G_{12}	2.009	1.076	0.570	0.371	2.448
d_4	3	2	3	4	3
μ_{42}	-1.805	-7.362	-3503.817	-1386.919	0.798
μ_{43}	-16.679		291.591	1083.167	11.082
μ_{44}				-2.252	
α_{41}	2.100	5.973	-2.926	-9.327	2.949
α_{42}	0.869	3.496	-3.205	-6.781	-2823.863
α_{43}	1.132		-6.401	-6.018	4.298
α_{44}				2.040	
\tilde{E}_{11}^{\min}	0	0	0	0	0
\tilde{E}_{11}^{\max}	0.11	0.14	0.41	0.50	0.28
\tilde{E}_{22}^{\min}	0	0	0	0	0
\tilde{E}_{22}^{\max}	0.067	0.14	0.34	0.62	0.28
\tilde{E}_{12}^{\max}	0.059	0.063	0.11	0.12	0.050

Table 3.3: Final fitted parameter sets for the five test materials. Thickness (τ) is measured in mm; parameters a_{11} , a_{12} , a_{22} , and G_{12} use MPa; all other parameters are dimensionless.

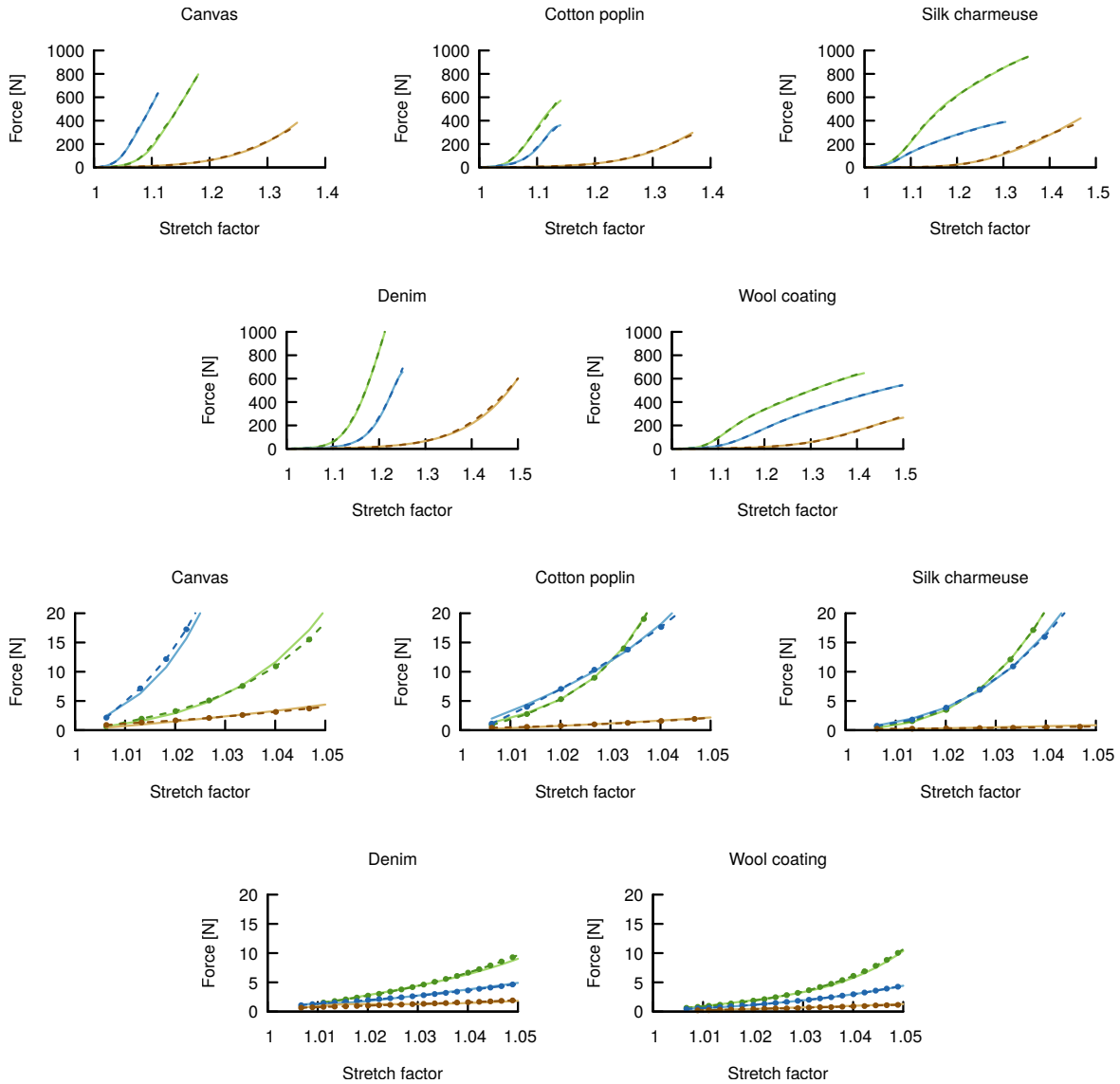


Figure 3.7: Force-elongation curves for 5 cm wide strips of canvas, cotton, silk, denim, and wool. Measured data is shown with dashed lines while fitted curves are shown in solid. The top set of plots shows the full strain domain while the bottom set shows a closeup for the small-strain regime. Each plot shows curves for warp (green), weft (blue) and bias (brown). Dots show individual data points, which are only shown in the small strain plots for legibility.

Fabric	Warp	Weft	Bias
Canvas	1.9%	2.0%	5.1%
Cotton poplin	2.9%	3.2%	2.6%
Silk charmeuse	1.9%	1.7%	4.5%
Wool coating	1.1%	0.8%	3.4%
Denim	1.6%	2.3%	4.7%

Table 3.4: Average relative error of the fitted models vs. the experimental data.

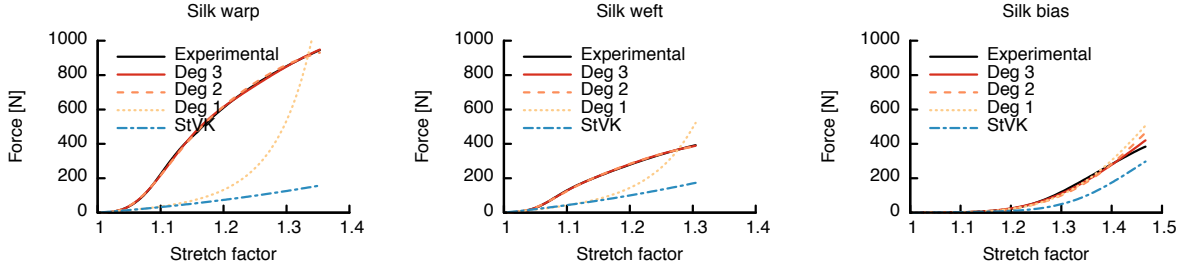


Figure 3.8: *Demonstration of the improvement of fit quality as the model complexity increases for silk. For warp and weft, degree 2 is sufficient to fit the experimental data well (average relative error under 2%). For bias, the transition from degree 2 to 3 still makes significant progress (average relative error reduces from 6% to 4.5%). In all cases, an orthotropic St. Venant-Kirchhoff model provides a poor fit. Due to our minimization of relative error, the small strain regime is prioritized even when using models too simple to match the full curve.*

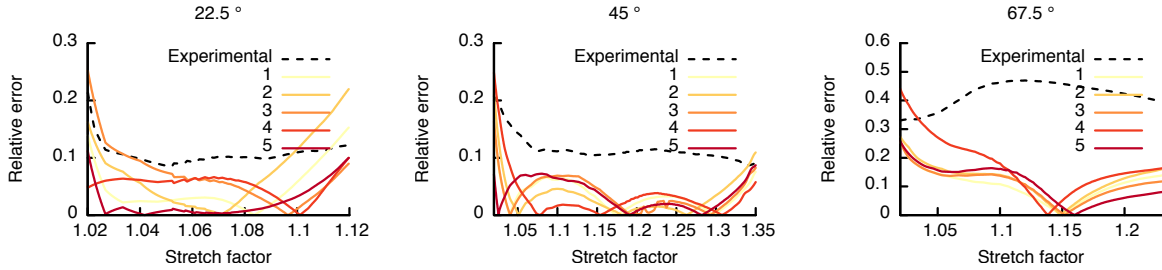


Figure 3.9: *Relative error plots for tensile tests at 22.5°, 45°, and 67.5° after adjusting for accidental cloth sample misalignments. The black curve shows the relative difference between the largest and the smallest unadjusted force measurements. Each of the remaining curves shows the relative error of the model fit compared to the experimental data for one cloth specimen. With very few exceptions the fitting error is well below the experimental error.*

Figure 3.8 demonstrates the progression of fit quality as the degrees d_j are increased and the resulting choice of degrees to fit the data accurately while minimizing model complexity. Table 3.3 gives the final list of curve degrees chosen for each material. The bending tests match less accurately. Recall that by design of the bending experiment, all cloth samples should drape downward at an angle of 41.5°. However, our fitted models produce a wide range of drape angles for some materials; see Table 3.5.

Fabric	Warp drape	Weft drape
Canvas	54.6°	23.7°
Cotton poplin	43.5°	39.2°
Silk charmeuse	52.8°	26.2°
Wool coating	42.2°	40.7°
Denim	50.6°	30.0°

Table 3.5: Bend test drape angles for the fitted models. To match experimental data, all drape angles should equal 41.5°.

For validation, the modeled forces are compared against tensile experiments at stretch directions 22.5° and 67.5° from weft. Each validation experiment is executed 5 times. To allow for the difficulty of accurately cutting strips at specific angles, we solve for an unknown angle deviation $\Delta\theta$ in each experimental run; thus the model forces used for comparison assume the strip is aligned at 22.5+ $\Delta\theta$ or 67.5+ $\Delta\theta$ degrees. With this adjustment, the relative errors between model and experimental forces are generally much smaller than experimental error. The validation test is run only for canvas, cotton, and silk. The relative errors for all three fabrics are summarized in Table 3.6. While these errors are larger than the fitting errors, they are consistently below the variation in the experimental data. The relative errors from the canvas validation are shown in Figure 3.9 as a function of the elongation.

Fabric	Model		Experiment	
	22.5°	67.5°	22.5°	67.5°
Canvas	5.8%	12.8%	11.0%	42.0%
Cotton poplin	7.3%	11.1%	22.4%	30.6%
Silk charmeuse	9.8%	5.6%	36.2%	22.3%

Table 3.6: Average relative error of the fitted models vs. the experimental data for the validation tests. Each number is the average over 5 experiments.

The qualitative look of the fitted materials is demonstrated in a series of simulations. The drape test (Figure 3.1) uses 1m × 1m squares of each material suspended from two corners held 60cm apart. This is similar to the setup used by [WOR11] but with a different distance between the corners. The simulated results may be compared against photographs

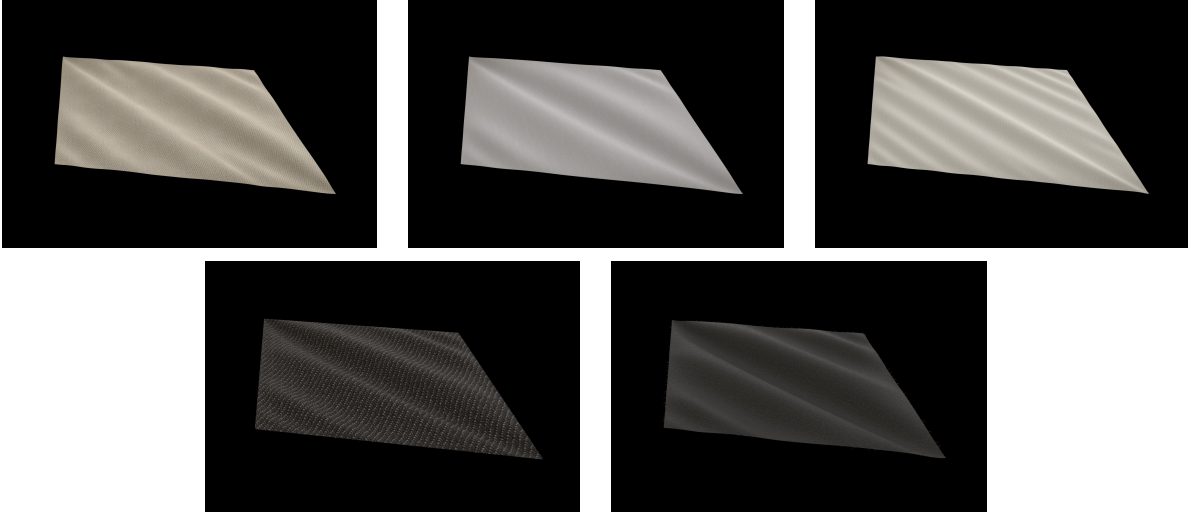


Figure 3.10: *The picture frame test constrains a $10\text{cm} \times 10\text{cm}$ cloth square along all four edges to introduce shearing and out-of-plane buckling. The materials from left to right are canvas, cotton, and silk (top row) and denim and wool (bottom row).*

of the same test run with real fabric samples. The picture frame test (Figure 3.10) shows deformation of a $10\text{cm} \times 10\text{cm}$ cloth square. The cloth is held along all edges and the constraints are moved to induce large shearing deformations and out-of-plane buckling. The corner pull test (Figure 3.11) also deforms a $10\text{cm} \times 10\text{cm}$ cloth square, this time using point constraints at the corners to create large diagonal stretches. In the plane-on-ball test (Figure 3.12), a $1\text{m} \times 1\text{m}$ fabric square is dropped onto an immovable ball of radius 25cm from an initial height of 50cm to produce collisions and complex wrinkling patterns. Collision handling is obtained from a triangle mesh discretization of our fitted models as discussed in §3.2.4.

All data fitting and simulations were run on an Intel Xeon E5-2698 v4 CPU with 64 GB RAM. Fitting each new material requires approximately 4 hours of compute time, together with some operator intervention to decide which degrees d_j produce the desired level of fitting accuracy.

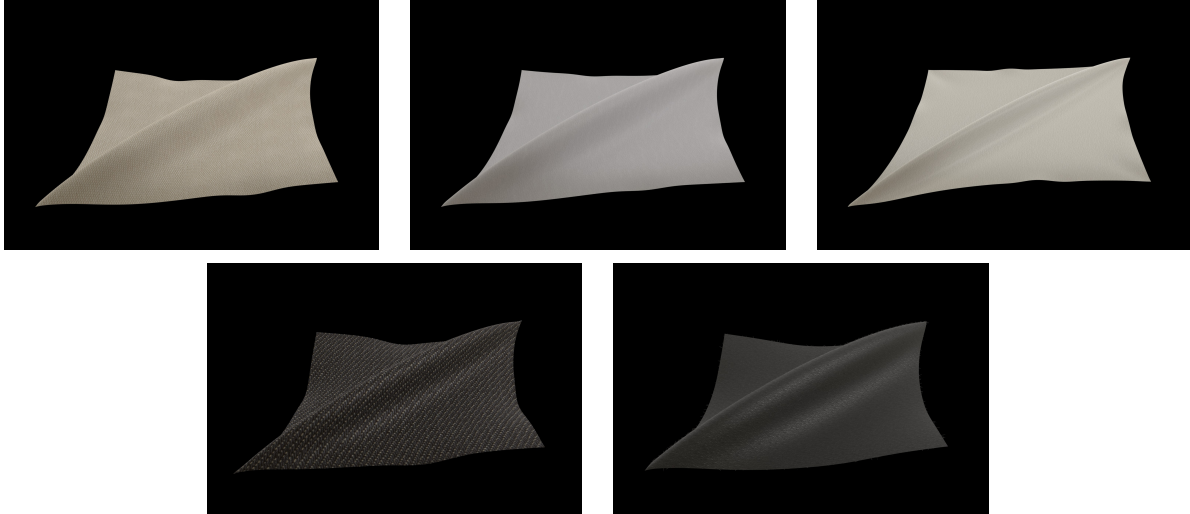


Figure 3.11: *A simulated corner pull test with the fitted materials. The $10\text{cm} \times 10\text{cm}$ cloth square is constrained at the corners and pulled diagonally, creating large stretch deformation and buckling. The materials from left to right are canvas, cotton, and silk (top row) and denim and wool (bottom row).*

3.7 Limitations and future work

Under the Kirchhoff-Love hypotheses, bending resistance is determined by the response to small strains ($< 3\%$ in our examples). The current tensile datasets exhibit large relative error between repeated tests in this regime. We believe this causes the discrepancies in our fitting of bend data; thus the bend fits could be improved using data from a tensile experiment which provides increased accuracy in small strain. Alternatively, the Kirchhoff-Love assumptions might need modification to capture the bending response for some types of cloth. That said, the popular discrete shells bending model, as used in [MMO16] and slightly modified for [WOR11] for similar cloth fitting problems, can also be derived as a discretization of the Kirchhoff-Love kinematic assumptions as discussed in §3.2.4.

Our results do not provide a fit for the Poisson effect term $a_{12}\eta_1(\tilde{E}_{11}\tilde{E}_{22})$ in the energy because this term has virtually no effect on the current experimental dataset. As discussed in §3.1, there exist experimental methods in the literature which could augment our dataset to exercise this term. Given such data we believe our approach could be extended to also fit this term - likely by fitting the new data after all tensile test data, between steps 6 and 7 of



Figure 3.12: *A $1m \times 1m$ cloth square falls onto a ball of radius $25cm$ to demonstrate contact handling and complex wrinkle patterns. The collision algorithm relies on the triangle mesh cloth discretization of our model. The materials from left to right are canvas, cotton, and silk (top row) and denim and wool (bottom row).*

the model fitting process in §3.3.2.

Our current optimization objective in Equation (3.5) penalizes relative error in all captured data points with equal weight. However, the tensile test machinery provides slightly varying sample density across various regimes of the strain domain. Further, it is unclear what distribution of sampled strains is desirable. This can be addressed by some choice of constant weights for each term in Equation (3.5), or by interpolating an experimental curve and then sampling points according to some chosen target measurement density (e.g., uniform). Such a step could improve the replicability of the results, since repeated tests tend to record samples at different points.

In fact, the structure of the fitting solver ignores another weighting choice which is potentially more important. Due to breaking the fit into a separate phase for each experiment, the question of relative weighting on e.g. warp stretch data versus bias stretch data is never addressed. Instead, the data is implicitly prioritized by phase order: warp and weft stretches

are highest priority, followed by bias stretch, and finally by bending. The main justification for breaking the solve down is that none of the phases affect the model’s fit of the data from previous steps and each step uses enough data to fully determine the parameters and curve shapes being computed. These properties ensure that if the model is capable of expressing the complete dataset then the phase-based solver will attain such a fit. The drawback lies in the choice of tradeoffs when the model cannot fit all input data at once. For example, it would be easy to fit the bending data perfectly by sacrificing the warp or weft fit. Further investigation could help produce minor compromises on a few experiments’ fits instead of the current larger errors in bending. Of course, a different model could avoid the problem simply by being capable of matching all of the experimental data simultaneously.

We do not attempt to model hysteresis from internal friction in the cloth. However, this is an important factor for the behavior of real cloth. Previously, this has been investigated by [MTB13] and we hope that our fitting method can be extended to also estimate the internal friction parameters. Similar to hysteresis, we have ignored any dependence in the deformation behavior on strain rate. Additional experimentation is needed to determine whether this is reasonable.

When the ultimate goal is to create pleasing and/or accurate cloth simulations, the interaction between the fabric and its environment can be just as important as the behavior of the cloth itself. Especially for light weight fabrics such as chiffon, the coupling between the cloth and the fluid flow of the surrounding air is essential.

Our fitting software implementation is tied to the Catmull-Clark subdivision finite element method detailed in Chapter Two. However, as discussed in §3.2.4 our constitutive model can be discretized to produce an approximate version computable on a triangle mesh, which we currently implement only for simulation purposes. Future work should investigate the viability and convergence properties of fitting using this triangle mesh model. Of course, the multiresolution optimization of fitting first with coarse meshes and gradually refining the discretization will become less useful because the triangle mesh model does not converge under refinement.

Another discretization-based avenue for study is the Reissner-Mindlin theory of thick shells, which provides an alternative to the Kirchhoff-Love theory by allowing midsurface-normal segments to undergo out of plane shearing. Finite elements for Reissner-Mindlin shells have been explored, for example in [AF89, Xia94]. The finite element basis functions require only H^1 regularity; however, the out of plane shear must be tracked as an additional unknown function. Also, specialized measures must be taken to avoid locking when Reissner-Mindlin finite elements are used for thinner objects such as cloth. Adaptation of our constitutive model and fitting procedure to a Reissner-Mindlin discretization is mathematically straightforward, but determining the numerical performance of such a method remains a useful future experiment.

REFERENCES

- [AF89] Douglas N Arnold and Richard S Falk. “A uniformly accurate finite element method for the Reissner–Mindlin plate.” *SIAM Journal on Numerical Analysis*, **26**(6):1276–1290, 1989. 96
- [Ben05] Benzi, Michele and Golub, Gene H. and Liesen, Jörg. “Numerical Solution of Saddle Point Problems.” **14**:1–137, 05 2005. 35
- [BFA02] Robert Bridson, Ronald Fedkiw, and John Anderson. “Robust treatment of collisions, contact and friction for cloth animation.” *ACM Transactions on Graphics (ToG)*, **21**(3):594–603, 2002. 60
- [BIE00] Y. Başar, M. Itskov, and A. Eckstein. “Composite laminates: nonlinear interlaminar stress analysis by multi-layer shell elements.” *Computer Methods in Applied Mechanics and Engineering*, **185**(2-4):367–397, 2000. 15, 56
- [BMF03] Robert Bridson, Sebastian Marino, and Ronald Fedkiw. “Simulation of clothing with folds and wrinkles.” In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 28–36. Eurographics Association, 2003. 5
- [BSL85] Ted Belytschko, Henryk Stolarski, Wing Kam Liu, Nicholas Carpenter, and Jame SJ Ong. “Stress projection for membrane and shear locking in shell finite elements.” *Computer Methods in Applied Mechanics and Engineering*, **51**(1-3):221–258, 1985. 47
- [BTH03] Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz. “Estimating Cloth Simulation Parameters from Video.” In *Proc 2003 ACM SIGGRAPH/Eurographics Symp Comp Anim, SCA '03*, pp. 37–51, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 6
- [CC78] E. Catmull and J. Clark. “Recursively generated B-spline surfaces on arbitrary topological meshes.” *Computer-Aided Design*, **10**(6):350 – 355, 1978. 24
- [Cia88] Philippe G Ciarlet. *Three-dimensional elasticity*, volume 20. Elsevier, 1988. 15
- [Cia02] Philippe G. Ciarlet. *Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. 48, 50
- [CL96] Philippe G. Ciarlet and Véronique Lods. “Asymptotic analysis of linearly elastic shells. III. Justification of Koiter’s shell equations.” *Archive for Rational Mechanics and Analysis*, **136**(2):191–200, Dec 1996. 9

- [CLM14] Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. “Yarn-level Simulation of Woven Cloth.” *ACM Trans. Graph.*, **33**(6):207:1–207:11, November 2014. [5](#)
- [CO01] Fehmi Cirak and Michael Ortiz. “Fully C^1 -conforming subdivision elements for finite deformation thin-shell analysis.” *International Journal for Numerical Methods in Engineering*, **51**(7):813–833, 2001. [2](#), [5](#), [16](#), [20](#), [26](#), [30](#)
- [Coo63] D. N. E. Cooper. “A Bias Extension Test.” *Textile Research Journal*, **33**(4):315–317, 1963. [54](#)
- [Coo97] R. Cools. “Constructing cubature formulae: the science behind the art.” *Acta Numerica*, **6**:1–54, 1 1997. [30](#)
- [COS00] Fehmi Cirak, Michael Ortiz, and Peter Schröder. “Subdivision surfaces: A new paradigm for thin-shell finite-element analysis.” *International Journal for Numerical Methods in Engineering*, **47**(12):2039–2072, 2000. [2](#), [5](#), [24](#), [30](#), [47](#), [48](#), [50](#)
- [CTT17] David Clyde, Joseph Teran, and Rasmus Tamstorf. “Modeling and Data-driven Parameter Estimation for Woven Fabrics.” In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA ’17, pp. 17:1–17:11, New York, NY, USA, 2017. ACM. [x](#)
- [DDB11] Samia Dridi, Abdelwaheb Dogui, and Philipp Boisse. “Finite element analysis of bias extension test using an orthotropic hyperelastic continuum model for woven fabric.” *The Journal of The Textile Institute*, **102**(9):781–789, 2011. [54](#), [67](#), [74](#)
- [DWH14] Laura De Lorenzis, Peter Wriggers, and Thomas J.R. Hughes. “Isogeometric contact: a review.” *GAMM-Mitteilungen*, **37**(1):85–123, 2014. [60](#)
- [GGR06] Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. “Computing discrete shape operators on general meshes.” *Computer Graphics Forum*, **25**(3):547–556, 2006. [2](#), [51](#)
- [GGW07] Akash Garg, Eitan Grinspun, Max Wardetzky, and Denis Zorin. “Cubic Shells.” In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’07, pp. 91–98, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. [51](#), [60](#), [61](#)
- [GHD03] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. “Discrete shells.” In *SCA ’03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 62–67. Eurographics Association, 2003. [5](#), [51](#), [60](#), [61](#)
- [GP00] Michael B. Giles and Niles A. Pierce. “An Introduction to the Adjoint Approach to Design.” *Flow, Turbulence and Combustion*, **65**(3):393–415, 2000. [79](#)

- [Gre03] Seth Green. *Multilevel, subdivision-based, thin shell finite elements: Development and an application to red blood cell modeling*. PhD thesis, University of Washington, 2003. [2](#)
- [GS14] Theodore F. Gast and Craig Schroeder. “Optimization Integrator for Large Time Steps.” In Vladlen Koltun and Eftychios Sifakis, editors, *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*. The Eurographics Association, 2014. [32](#), [33](#), [35](#)
- [GSH04] Yotam Gingold, Adrian Secord, Jefferson Y. Han, Eitan Grinspun, and Denis Zorin. “A Discrete Model for Inelastic Deformation of Thin Shells.” 2004. [60](#)
- [GT04] Seth Green and George Turkiyyah. “Second-order accurate constraint formulation for subdivision finite element simulation of thin shells.” *International Journal for Numerical Methods in Engineering*, **61**(3):380–405, 2004. [31](#), [47](#), [48](#), [50](#)
- [GZ03] Tushar K. Ghosh and Naiyue Zhou. “Characterization of fabric bending behavior: A review of measurement principles.” *Indian Journal of Fibre & Textile Research*, **28**(4):471–476, 2003. [54](#)
- [HCB05] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement.” *Computer Methods in Applied Mechanics and Engineering*, **194**(39-41):4135–4195, 2005. [2](#), [5](#), [27](#)
- [HRR09] A. Hursa, T. Rolich, and S. Ercegović Ražić. “Determining Pseudo Poisson’s Ratio of Woven Fabric with a Digital Image Correlation Method.” *Textile Research Journal*, **79**(17):1588–1598, 2009. [55](#)
- [HWS12] Jeffrey Lee Hellrung, Luming Wang, Eftychios Sifakis, and Joseph M Teran. “A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions.” *Journal of Computational Physics*, **231**(4):2015–2048, 2012. [36](#)
- [Inc16] Wolfram Research, Inc. “Mathematica, Version 11.0.”, 2016. Champaign, IL. [73](#)
- [Its01] Mikhail Itskov. “A generalized orthotropic hyperelastic material model with application to incompressible shells.” *International Journal for Numerical Methods in Engineering*, **50**(8):1777–1799, 2001. [56](#), [57](#), [58](#)
- [JMP16] Bert Jüttler, Angelos Mantzaflaris, Ricardo Perl, and Martin Rumpf. “On numerical integration in isogeometric subdivision methods for PDEs on surfaces.” *Computer Methods in Applied Mechanics and Engineering*, **302**:131–146, 2016. [30](#)
- [Kaw80] Suetō Kawabata. *The Standardization and Analysis of Hand Evaluation*. The Textile Machinery Society of Japan, Osaka, Japan, 2nd edition edition, 1980. [4](#)

- [KHW15] Josef Kiendl, Ming-Chen Hsu, Michael C.H. Wu, and Alessandro Reali. “Iso-geometric Kirchhoff-Love shell formulations for general hyperelastic materials.” *Computer Methods in Applied Mechanics and Engineering*, **291**:280–303, 2015. [6](#), [20](#)
- [KJM08] Jonathan M. Kaldor, Doug L. James, and Steve Marschner. “Simulating Knitted Cloth at the Yarn Level.” *ACM Trans. Graph.*, **27**(3):65:1–65:9, August 2008. [5](#)
- [KJS05] M. J. King, P. Jearanaisilawong, and S. Socrate. “A continuum constitutive model for the mechanical behavior of woven fabrics.” *International Journal of Solids and Structures*, **42**(13):3867–3896, 2005. [5](#)
- [Koi] WT Koiter. “On the mathematical foundation of shell theory.” [47](#), [48](#)
- [KYT06] L. Kharevych, Weiwei Yang, Y. Tong, E. Kanso, J. E. Marsden, P. Schröder, and M. Desbrun. “Geometric, variational integrators for computer animation.” In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA ’06, pp. 43–51. Eurographics Association, 2006. [33](#)
- [LB14] Yijing Li and Jernej Barbič. “Stable Orthotropic Materials.” In Vladlen Koltun and Eftychios Sifakis, editors, *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA ’14, pp. 41–46. The Eurographics Association, 2014. [7](#)
- [LBC12] Quan Long, P. Burkhard Bornemann, and Fehmi Cirak. “Shear-flexible subdivision shells.” *International Journal for Numerical Methods in Engineering*, **90**(13):1549–1577, 2012. [6](#)
- [LM08] Christiane Lubbe and Nadia Magnenat-Thalmann. “The Simulation of Cloth Using Accurate Physical Parameters.” In *Proceedings of the Tenth IASTED International Conference on Computer Graphics and Imaging*, CGIM ’08, pp. 123–128, Anaheim, CA, USA, 2008. ACTA Press. [6](#)
- [LO64] R. G. Livesey and J. D. Owen. “Cloth Stiffness and Hysteresis in Bending.” *Journal of the Textile Institute Transactions*, **55**(10):T516–T530, 1964. [54](#)
- [LPC08] W. Lee, J. Padvoiskis, J. Cao, E. de Luycker, P. Boisse, F. Morestin, J. Chen, and J. Sherwood. “Bias-extension of woven composite fabrics.” *International Journal of Material Forming*, **1**(1):895–898, 2008. [67](#)
- [LSN09] Charles Loop, Scott Schaefer, Tianyun Ni, and Ignacio Castaño. “Approximating Subdivision Surfaces with Gregory Patches for Hardware Tessellation.” *ACM Trans. Graph.*, **28**(5):151:1–151:9, December 2009. [41](#)
- [LZ14] Jia Lu and Chao Zheng. “Dynamic cloth simulation by isogeometric analysis.” *Computer Methods in Applied Mechanics and Engineering*, **268**:475 – 493, 2014. [60](#)

- [Mat94] Pontus Matstoms. “Sparse QR factorization in MATLAB.” *ACM Transactions on Mathematical Software (TOMS)*, **20**(1):136–159, 1994. [37](#)
- [MBT12] E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. “Data-Driven Estimation of Cloth Simulation Models.” *Computer Graphics Forum*, **31**(2pt2):519–528, May 2012. [4](#), [7](#)
- [Min95] Pier Giorgio Minazio. “FAST - Fabric Assurance by Simple Testing.” *International Journal of Clothing Science and Technology*, **7**(2/3):43–48, 1995. [4](#)
- [MLV07] Nadia Magnenat-Thalmann, Christiane Luble, Pascal Volino, and Etienne Lyard. “From Measured Fabric to the Simulation of Cloth.” In *Computer-Aided Design and Computer Graphics, 2007 10th IEEE International Conference on*, pp. 7–18, Oct 2007. [6](#)
- [MMO16] Eder Miguel, David Miraut, and Miguel A. Otaduy. “Modeling and Estimation of Energy-Based Hyperelastic Objects.” *Computer Graphics Forum*, **35**(2), 2016. [7](#), [93](#)
- [MTB13] Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C. Schwartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, and Miguel A. Otaduy. “Modeling and Estimation of Internal Friction in Cloth.” *ACM Trans. Graph.*, **32**(6):212:1–212:10, November 2013. [6](#), [95](#)
- [NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006. [34](#), [86](#)
- [Ogd72] R. W. Ogden. “Large Deformation Isotropic Elasticity: On the Correlation of Theory and Experiment for Compressible Rubberlike Solids.” *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, **328**(1575):567–583, 1972. [56](#)
- [Pei30] F. T. Peirce. “The “Handle” of Cloth as a Measurable Quantity.” *Journal of the Textile Institute Transactions*, **21**(9):T377–T416, 1930. [54](#)
- [Pow13] Jess Power. “Fabric objective measurements for commercial 3D virtual garment simulation.” *International Journal of Clothing Science and Technology*, **25**(6):423–439, 2013. [6](#)
- [PT12] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012. [26](#)
- [RCY07] Hansun Ryou, Kwansoo Chung, and Woong-Ryeol Yu. “Constitutive modeling of woven composites considering asymmetric/anisotropic, rate dependent, and nonlinear behavior.” *Composites Part A: Applied Science and Manufacturing*, **38**(12):2500–2510, 2007. [6](#)

- [RO99] R. Radovitzky and M. Ortiz. “Error estimation and adaptive meshing in strongly nonlinear dynamic problems.” *Computer Methods in Applied Mechanics and Engineering*, **172**(1-4):203–240, 1999. 33
- [SFH08] Ruben Sevilla, Sonia Fernández-Méndez, and Antonio Huerta. “NURBS-enhanced finite element method (NEFEM).” *International Journal for Numerical Methods in Engineering*, **76**(1):56–83, 2008. 27
- [SG06] Olaf Schenk and Klaus Gärtner. “On fast factorization pivoting methods for sparse symmetric indefinite systems.” *ETNA. Electronic Transactions on Numerical Analysis [electronic only]*, **23**:158–179, 2006. 35
- [SPP05] Huiyu Sun, Ning Pan, and Ron Postle. “On the Poisson’s ratios of a woven fabric.” *Composite Structures*, **68**(4):505–510, 2005. 55
- [Sta98] Jos Stam. “Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values.” In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 395–404. ACM, 1998. 24
- [TWS06] Bernhard Thomaszewski, Markus Wacker, and Wolfgang Straßer. “A Consistent Bending Model for Cloth Simulation with Corotational Subdivision Finite Elements.” In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’06, pp. 107–116, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. 6
- [VM07] Pascal Volino and Nadia Magnenat-Thalmann. “Stop-and-go cloth draping.” *The Visual Computer*, **23**(9):669–677, 2007. 77
- [VMF09] Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. “A simple approach to nonlinear tensile stiffness for accurate cloth simulation.” *ACM Trans. Graph.*, **28**:105:1–105:16, September 2009. 6
- [VSJ13] Roman Vetter, Norbert Stoop, Thomas Jenni, Falk K. Wittel, and Hans J. Herrmann. “Subdivision shell elements with anisotropic growth.” *International Journal for Numerical Methods in Engineering*, **95**(9):791–810, 2013. 6
- [WHP11] Anna Wawrzinek, Klaus Hildebrandt, and Konrad Polthier. “Koiter’s Thin Shells on Catmull-Clark Limit Surfaces.” In Peter Eisert, Joachim Hornegger, and Konrad Polthier, editors, *Vision, Modeling, and Visualization (2011)*. The Eurographics Association, 2011. 2, 6
- [Wil10] Robert W. Williams. *Measuring and modeling the anisotropic, nonlinear and hysteretic behavior of woven fabrics*. PhD thesis, University of Iowa, December 2010. 6
- [WLD08] X. Wang, X. Liu, and C.Hurren Deakin. “Physical and mechanical testing of textiles.” In Jinlian Hu, editor, *Fabric Testing*, Woodhead Publishing Series in Textiles, chapter 4, pp. 90–124. Woodhead Publishing, 2008. 6

- [WOR11] Huamin Wang, James F. O’Brien, and Ravi Ramamoorthi. “Data-Driven Elastic Models for Cloth: Modeling and Measurement.” *ACM Transactions on Graphics (SIGGRAPH 2011)*, **30**(4):71:1–71:12, August 2011. [4](#), [7](#), [83](#), [91](#), [93](#)
- [WP64] A. S. Wineman and A. C. Pipkin. “Material symmetry restrictions on constitutive equations.” *Archive for Rational Mechanics and Analysis*, **17**(3):184–214, Jan 1964. [15](#)
- [WP16] Anna Wawrzinek and Konrad Polthier. “Integration of generalized B-spline functions on Catmull-Clark surfaces at singularities.” *Computer-Aided Design*, **78**:60–70, 2016. {SPM} 2016. [30](#)
- [WPP98] J Wang, JR Page, and R Paton. “Experimental investigation of the draping properties of reinforcement fabrics.” *Composites Science and Technology*, **58**(2):229–237, 1998. [74](#)
- [Xia94] Cheng Xiao-liang. “A simple finite element method for the Reissner-Mindlin plate.” *Journal of Computational Mathematics*, pp. 46–54, 1994. [96](#)
- [XSZ15] Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbič. “Nonlinear Material Design Using Principal Stretches.” *ACM Trans. Graph.*, **34**(4):75:1–75:11, July 2015. [7](#)
- [Zor05] Denis Zorin. “Curvature-based energy for simulation and variational modeling.” In *Shape Modeling and Applications, 2005 International Conference*, pp. 196–204. IEEE, 2005. [2](#)
- [ZPA17] Qingyu Zhao, Stephen M. Pizer, Ron Alterovitz, Marc Niethammer, and Julian G. Rosenman. “Orthotropic Thin Shell Elasticity Estimation for Surface Registration.” In *IPMI*, 2017. [2](#), [60](#)