

An Inextensible Model for the Robotic Manipulation of Textiles

Franco Coltraro^{a,*}, Jaume Amorós^b, Maria Alberich-Carramiñana^a, Carme Torras^a

^a*Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain.*

^b*Universitat Politècnica de Catalunya, Barcelona, Spain.*

Abstract

We introduce a new isometric strain model for the study of the dynamics of cloth garments in a moderate stress environment, such as robotic manipulation in the neighborhood of humans. This model treats textiles as surfaces that are inextensible, admitting only isometric motions. Inextensibility is derived in a continuous setting, prior to any discretization, which gives consistency with respect to remeshing and prevents the problem of locking even with coarse meshes. The simulations of robotic manipulation using the model are compared to the actual manipulation in the real world, finding that the difference between the simulated and the real position of each point in the garment is lower than 1cm in average even when a coarse mesh is used. Aerodynamic contributions to motion are incorporated to the model through the virtual uncoupling of the inertial and gravitational mass of the garment. This approach results in an accurate, when compared to the recorded dynamics of real textiles, description of cloth motion incorporating aerodynamic effects by using only two parameters.

Keywords:

inextensible cloth, FEM simulation, physical modeling, experimental validation, robotic manipulation

2010 MSC: 74K20, 53Z05

1. Introduction

Robotic manipulation of cloth in a domestic environment, i.e. not in its manufacturing plant, is an increasingly relevant problem because of the ubiquity and versatility of textiles in human lives and activities. Nevertheless, textiles are difficult to model from a material viewpoint. They can be naturally treated as thin plates or shells, but they show an extreme dichotomy: they initially stretch as an elastic solid in the tangent directions, but bend almost freely in the out-of-plane ones. This combination means that textiles are prone to buckling under the slightest compression. Moreover, they are usually non-isotropic because of their weaved inner structure, and nonlinear responses to stresses

*Corresponding author

Email address: fcoltraro@iri.upc.edu (Franco Coltraro)

appear very soon [1]. An additional difficulty for the modeling of the motion of cloth in everyday uses is posed by aerodynamics: most textiles are light, so they are considerably affected by the resistance of air to their movement. One can try simple drag-and-lift models of this resistance in the surface of a cloth away from its edges [2], but those edges bend easily and thus interact with flow turbulences around them in a more complicated way than a rigid solid, even a vibrating one [3].

The purpose of this work is to introduce and discuss a model of cloth intended for its control under robotic manipulation in a human environment, which means that the textiles will be subjected to moderate to low stresses [4]. Because of its control purpose, the authors sought a model with which simulations of the motion of cloth could be computed fast, but with a small margin of error (of the order of 1 cm) in the position of every point of the cloth when compared to its position in the real cloth subjected to the same manipulation by the robot. Thus, the main aims of this model are shape realism and speed, while dramatic effects and aesthetic considerations will be disregarded.

The achievement of computational speed taking advantage of the allowed cm margin of error, led the authors to the most basic hypothesis in the model: we will assume that textiles are *inextensible*, that is, the surfaces that represent them only deform isometrically through space, aiming at preserving not only the area but also both dimensions of each piece of the cloth. This assumption simplifies the model by removing all considerations of elasticity in it, at a cost of introducing errors in the mm range in the position of the cloth under the low stresses at which domestic robots operate. We will model inextensibility as a set of constraints, and since they will be derived from continuous conditions, the resulting model becomes very stable under different resolutions of the meshed fabric, allowing its use in coarse-mesh simulations.

Since the publication of the seminal paper [5] a lot of attention has been given to the simulation of inextensible and quasi-inextensible textiles. The main difficulty of inextensible cloth simulation is *locking*, i.e. the meshed garment becoming very rigid or facing a spurious resistance to bending. With limited exceptions such as creasing along straight lines, this is an unavoidable consequence of applying elasticity or spring models to a triangulated fabric and then tuning them for full inextensibility: in the inelastic limit, the lengths of all edges in the mesh become fixed, and often the only deformations preserving all these lengths are those in which the polyhedral surface moves as a rigid body. The underlying reason is the classical Cauchy Rigidity Theorem for polytopes, extended by Gluck [6], who showed that generic closed polyhedral surfaces in space are rigid, i.e. they lock once one fixes the length of their edges.

To overcome this difficulty, two approaches have been proposed in literature: in [5] a discrete model is presented where a quad-mesh discretized cloth is made inextensible in the horizontal and vertical directions but still elastic diagonally, i.e. allowing shearing but not stretching. For triangle meshes, [7] uses non-conforming triangles to impose full inextensibility, albeit sacrificing continuity of the polyhedral surface at the edges. One may wonder then, why is another inextensible model necessary? We give several reasons for this:

1. To our knowledge, all inextensible or quasi-inextensible models in literature are discrete in nature, i.e. they impose inextensibility constraining areas, edge lengths, etc. This may create mesh related artifices [8] and often impedes convergence as the mesh is refined [9].

2. The other family of models, which are continuous, are all based on elasticity theory, with the drawback that, in the inelastic limit when Poisson’s ratio tends to zero, exhibit locking (see [10] and [11]).
3. The development of a continuous model that performs well with coarse meshes is useful, particularly in the context of robotic manipulation of deformable objects, where speed is of critical importance.

In this work, we avoid the problem of locking by using an inextensible cloth model that is continuous and based on Differential Geometry. This in turn causes all the edges of the meshed garment not to have fixed length and allows the cloth to deform freely but at the same time conserving the total area of the cloth within an error of less than 1% (see Section 8.3).

1.1. Contributions of this work

- A locking-free continuous model for the simulation of inextensible textiles based on natural geometrical constraints is presented.
- A novel and efficient discretization of the model using the Finite Element Method (FEM) is presented in full detail. Our approach results in a unified treatment of triangles and quadrilaterals.
- We propose a framework for the empirical validation of any cloth model using a robot and a simple way of accounting for the full aerodynamic contribution to dynamics in our cloth model.

2. Related work

The mechanic behavior of cloth has been extensively studied from the engineering and computer graphics viewpoints. The engineering focus has usually been on cloth manufacture and its mechanical resistance, using elasticity theory to examine local properties of the material. The aim in computer graphics has always been the simulation and representation of cloth motion, using mostly mass-spring models for speed of computation and paying attention to global problems such as nontrivial cloth topology and collisions. In both fields, the textiles are usually modeled as two-dimensional surfaces. Since the pioneering work of [12] dozens of different models have been proposed for cloth simulation. Our approach retakes the original idea of that paper: understanding cloth’s internal dynamics as the preservation of the first fundamental form of the surface. For surveys and research problems see [8, 9, 13]. We now review some of the most popular methods for simulating the internal dynamics of cloth, noting that there is not a clear cut separation between some of them. We deliberately only reference lines of research that model internal dynamics of cloth in an essentially different *physical* manner. Approaches that develop novel numerical methods to integrate existing physical models are not considered, since that is not the contribution of our paper.

Textile engineering: The modeling of fabrics from a Structural Mechanics viewpoint has been a topic of interest in Textile Engineering since the start of industrial manufacture of cloth. See [1] for a survey, and completion in many respects, of this

theory, or [14] for updates. This modeling has been hampered by the complexity of fabric as a material: it is inhomogeneous, even discontinuous, already at a relatively sizable scale; highly anisotropic and capable of a nonlinear response even to relatively modest strains; prone to buckling under very small compression.

Because of these complexities, investigations of the explicit mathematical expression of the stress-strain relationships of fabrics start from the elasticity theory of solids, and typically consider fabric as an elastic thin plate, more rarely as a shell. Ignoring its thickness, the resulting surface has 6 strain fields that determine its displacement and deformation, and a 6×6 symmetric stiffness matrix giving the constitutive equations that relate stress and strain ([14] p.10). Particular properties of fabric such as orthotropy reduce the number of stiffness parameters to take into account and lead to simplifications of this general elastic model (e.g. [1] and [15]), or to the introduction of nonlinear responses in the model [16].

The development of these models has been based foremost on experimentation, using procedures such as the Kawabata Evaluation System (KES) [17], which accurately measures the tensile, shear, pure bending, compression and friction behavior of a cloth sample under stresses that are typical in the cloth manufacturing process, in order to establish the stiffness parameters of the model. The increase in available computing power has made viable, although not for real time computing, sophistications of this basic model to take into account the fiber structure of yarns and fabrics (see [18] and [19]).

The computational complexity of these models opened the way to the development of simpler, descriptive models that would be later become of common use in Graphic Computer Science (as we explain below), treating fabrics as a viscoelastic material that can be modeled as a mass-spring system [20]. While such descriptive models have been extensively used in fabric simulation, their tenuous connection to reality has made them of limited use for the industrial handling of cloth. Here the thin-plate with linear elasticity models prevail.

Elastic models: these models, e.g. [21, 22] and [23], derive the internal forces of the garment from elasticity theory. They are the practical realization of the textile engineering models previously discussed, aimed at graphical simulations rather than at the study of physical properties of cloth. They are all continuous in nature and have the advantage of being physically based, stable under different meshing of the garment and convergent when the mesh is refined [13]. Mostly, they use finite elements to discretize the equations of motion [24]. They can be expensive to evaluate (especially when using non-linear elasticity or the co-rotational method, necessary for ensuring rotational invariance [22]) and may exhibit locking of triangular meshes if elasticity is heavily reduced: see [7] and [25] and more in general [10] and [11] for a more detailed discussion.

Mass-spring systems: these models, e.g. [26] and [27], derive internal forces from spring-like energies. They are cheap to evaluate and very intuitive, but less physically sound: they are mesh dependent, have a lot of (non-physical) parameters to be tuned (see [28] for an evolutionary algorithm to tune them) and do not show a convergent behaviour when the mesh is refined [9].

Constrained dynamics: these models derive internal forces from explicit conditions that the cloth must satisfy [8]. Most of the methods use some kind of Lagrange multipliers to impose the constraints. They mostly differ in what the conditions are and the algorithm used to impose the constraints. Our method fits in this category. Sometimes they are used as velocity filters that complement the previous methods. There are mainly two kinds:

1. *Continuous:* in this case the constraints are continuous in nature, e.g. imposing bounds to the strain tensor. Examples of this approach are [29, 30, 31] and [32]. To our knowledge all methods use elasticity theory to some extent, and therefore in the limit, when elasticity is reduced, face the same problems commented previously. Our method lies in this category with the important difference that it does not use elasticity, but differential geometry of the surface to derive the constraints.
2. *Discrete:* in this case constraints are discrete in nature, e.g. preserving the length of the edges of the meshed garment or the area of the elements. They are derived concretely for the mesh at hand. Examples are [5, 7, 25, 33]. Their main drawback is the possible lack of convergence of the methods (as opposed to the continuous case) and the possible introduction of mesh related artifacts [31]. Nevertheless they can be fast and handle better the inelastic scenario than the elasticity-based continuous models.

Others: There are two different recent trends: modeling woven cloth at the fiber level (e.g. [18] and [34]) as opposed to the macroscopic level (i.e. as a surface) and considering cloth as a non-Newtonian fluid using the Material Point Method (MPM) adapted to co-dimensional elasticity [19]. The first method is computationally very intensive and the second while being more efficient depicts cloth as very elastic.

The definitive test for a model of cloth is its comparison to reality. Textile engineering models are focused on such comparison, to the point of developing specialized testing equipment. But the object of their study are local properties of cloth, such as elasticity parameters, which are tested in static scenarios (e.g. [14, 35, 36, 37]). To the knowledge of the authors, none of the models developed in the area of Computer Graphics for the simulation of textiles has been able to compare its results with the motion of cloth dynamically. Being oriented to robotic manipulation of cloth, our model is ideally placed for such experimental validation as discussed in Section 9.

3. Overview

Our main assumption will be to consider cloth as a *continuous* and *inextensible* two-dimensional *surface*. Formulating the model at the continuous level is important because it avoids as much as possible mesh dependencies: i.e. without changing the physical parameters of the model (e.g. stiffness, damping, etc.) the results are mostly independent of mesh topology (this will be checked in our experiments). This is not the case when using mass-spring systems [9]. On the other hand, since most cloth materials do not stretch under their own weight, inextensibility is a very reasonable approximation for most textiles (specially in a robotics context, where fine details such as wrinkles are

not needed). We will show this comparing our model to real-life experimental data (see Section 9). Moreover, this assumption reduces greatly the amount of tuneable parameters of the model (stretching and shearing [1, 23] are no longer present).

Definition 1 (Inextensibility). A smooth surface is said to be inextensible if over time its metric is preserved. This is equivalent to requiring, that at all times the length of any given curve inside the surface remains constant (see Figure 1).

Given a parametrization $\varphi = \varphi(\xi, \eta) \in \mathbb{R}^3$ of a smooth surface $S \subset \mathbb{R}^3$ its *metric* (first fundamental form) is given by:

$$d\varphi^\top \cdot d\varphi = \begin{bmatrix} \langle \varphi_\xi, \varphi_\xi \rangle & \langle \varphi_\xi, \varphi_\eta \rangle \\ \langle \varphi_\eta, \varphi_\xi \rangle & \langle \varphi_\eta, \varphi_\eta \rangle \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix},$$

where $\varphi_\xi = \partial_\xi \varphi$ denotes partial differentiation. The importance of this 2×2 matrix relies on the fact that it gives a unique way of measuring angles and distances intrinsically on the surface [38].

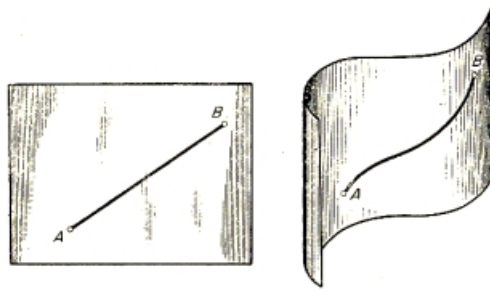


Figure 1: Isometric deformation of a surface: in both surfaces, the length of the depicted (geodesic) curve is the same. Image taken from <https://solitaryroad.com/c334.html>.

Remark 3.1 (Quasi-inextensibility). When the coefficients E and G of the metric are preserved, but not necessarily F , the surface is said to be *quasi-inextensible*. In physical terms this means that the surface shears but does not stretch.

3.1. Equations of inextensibility

Assume now that S is inextensible and it is moving through space; by virtue of the previous definition this means that we have a continuum of surfaces $\{S_t\}_{t \geq 0}$ such that $S_0 = S$ and for every $t > 0$ there exists a (smooth) isometry $H_t : S_0 \rightarrow S_t$. We think of the parameter $t \geq 0$ as time. If φ is any parametrization of S , then $\varphi(t) = H_t \circ \varphi$ is a parametrization of S_t and inextensibility means:

$$d\varphi(t)^\top \cdot d\varphi(t) = d\varphi^\top \cdot d\varphi \quad \text{for all } t \geq 0. \quad (1)$$

These are 3 independent conditions (depending on time and space):

$$\langle \varphi_\xi, \varphi_\xi \rangle(t) = E_0, \quad \langle \varphi_\xi, \varphi_\eta \rangle(t) = F_0, \quad \langle \varphi_\eta, \varphi_\eta \rangle(t) = G_0 \quad \text{for } t \geq 0. \quad (2)$$

Remark 3.2 (Rigidity invariance). Noting that $d\varphi(t) = \nabla H_t|_\varphi \cdot d\varphi$, we can rewrite (1) as

$$d\varphi^\top (\nabla H_t|_\varphi^\top \cdot \nabla H_t|_\varphi - I) \cdot d\varphi = 0 \quad (3)$$

where ∇H_t is the Jacobian of H_t and I is the identity 3×3 matrix. Therefore if $H_t(x) = Ax + b$ is any rigid transformation of \mathbb{R}^3 , we must have that A is an orthogonal matrix and hence it preserves the metric of S as expected. This rigidity invariance is not obtained when using linear elasticity models, e.g. big rotations can lead to different elastic restoring forces [22].

Remark 3.3 (Relationship to elasticity). In elasticity theory, the *Cauchy-Green strain tensor* is defined as: $E = \frac{1}{2} [\nabla H_t|_\varphi^\top \cdot \nabla H_t|_\varphi - I]$. Note that all deformations that have zero strain are isometric (see Equation (3)), but not all isometries are strain-free (e.g. bending along a line). This means that imposing (3) as a hard constraint (as we will do) is not equivalent to imposing zero strain deformations with classical elasticity theory.

3.2. Derivation of the model

The continuation of the paper is organized as follows: we explain how to impose conditions (2) as hard constraints using *Lagrange multipliers*. In order to treat computationally this, we employ finite elements [24] in the following way: we use them to discretize the Lagrangian of the mechanical system but not the equations of motion, obtaining this way as *Euler-Lagrange equations* an ordinary differential equation (ODE) system and not a (difficult to integrate, e.g. [12]) partial differential equation (PDE) system. We then discuss the computational complexity of evaluating the inextensibility constraints. These will be the most technical parts of the manuscript and this is due to the proposed model having aspects that are delicate and original from a mathematical viewpoint. Particularly important is how to constrain nodes lying at the boundaries (if not done properly this could lead to *locking*), and the parametrization of the surface, in order to allow arbitrary topologies. After this is done, we discuss how to integrate numerically the equations of motion, modifying the approach described in [5] to consider at the same time inextensibility and collisions against a fixed object. Bending is modeled using Willmore’s Energy as described in [39].

3.3. Performance evaluation:

We test the performance of the presented inextensible model in several challenging scenarios: first, we perform a simple quasi-static test to show the locking-free nature of our model, along with its independence with respect to different meshed topologies. We use different triangle and quadrilateral meshings to prove our point. Second, we show how to simulate Cusick’s test with our model. With this experiment we show the stability of our model when the mesh is refined. Lastly, we present a scenario with non-trivial cloth topology, where we simulate the motion of a tank-top and check to what extend our theoretical inextensibility assumptions are being met in practice, computing several area errors.

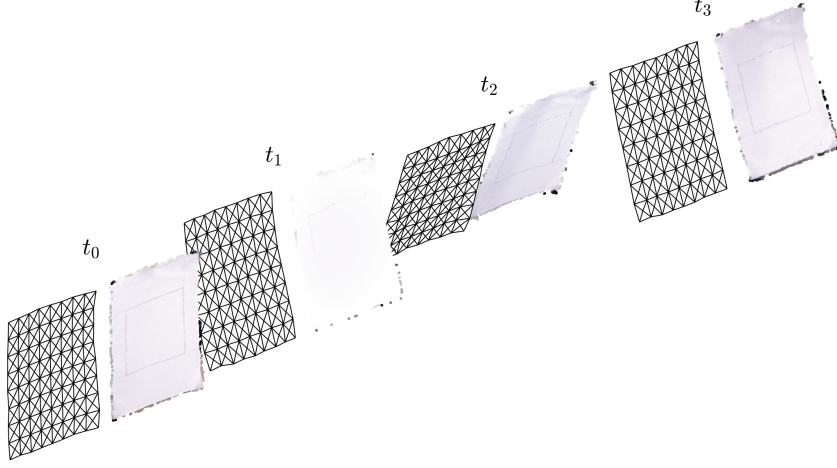


Figure 2: Comparison at four time instants of the recorded fast motion of paper (right) versus its simulation with the inextensible model (left) with $\delta = 0.37$ and $\alpha = 2.49$. The mean absolute error is 0.41 cm.

3.4. Experimental validation

Finally, we compare the inextensible model with reality (see Figure 2) and show how with as few as two parameters, we are able to model different types of textiles (such as cotton, wool and felt) under fast and slow motions. To perform this real-world validation we use a Barret robotic arm together with a depth camera to record the real motion of garments being shaken at different velocities (see Figure 3).

We implement oscillatory motions: a forward and backwards shaking (see Figure 2) at two different speeds. Afterwards, the resulting point-cloud recordings are de-noised, interpolated and meshed, so that we end up with the spatial trajectory of the vertices of a polyhedron. In order to keep the validation manageable, we focus on rectangular pieces of cloth for four different textiles: light cotton, wool, felt and paper. They are all of the same dimensions: 29.7×42 cm (DIN A3). We use a trivial rectangular topology in order to avoid occlusions in the point-cloud when using the depth camera to record the motions. Finally, we find the physical parameters of the model that best approximate the motion of the polyhedron using a least squares approximation and a minimization algorithm. We study the evolution of absolute errors and dispersion measures (i.e. standard deviations) of the simulated garments versus the real ones for all textiles and motions.

Before we proceed, for reference, we present a list of the most important symbols used in this paper. We will follow the following notational conventions:

1. Matrices, tensors and vector functions will be denoted by bold capital letters.
2. Vectors (except for points in \mathbb{R}^3) will be denoted by bold lower-case letters.
3. The rest (e.g. scalar parameters, points in \mathbb{R}^3 , etc.) will be denoted by capital and lower-case italics.



Figure 3: Experimental setup for the recording of real world data. On the left, the depth camera. On the right, the robotic arm with an affixed hanger.

List of Symbols

S	Surface used to model the cloth.
Ω_e	Elements (e.g. triangles) of the discretized surface.
$\varphi_e(\xi, \eta)$	Local parametrization of the element Ω_e of the surface.
$\partial_\xi \varphi_e, \partial_\eta \varphi_e$	Partial derivatives of the parametrizations.
n	Number of vertices/nodes of the discrete surface.
$p_i(t)$	Coordinates in \mathbb{R}^3 of the node i of the surface.
\mathcal{N}_i	Indicator functions, i.e. $\mathcal{N}_i(p_j) = \delta_{ij}$.
$\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)$	x -coordinates (resp. y and z) of all the nodes of the surface in \mathbb{R}^n .
$\boldsymbol{\varphi}(t)$	Position of all nodes together, i.e. $(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t))^T \in \mathbb{R}^{3n}$.
$E_k(t), F_k(t), G_k(t)$	Coefficients of the metric of the discrete surface at node k .
$\mathbf{C}(\boldsymbol{\varphi}(t))$	Constraint function, e.g. $C_i(\boldsymbol{\varphi}(t)) = E_k(t) - E_k(0)$.
$\boldsymbol{\lambda}(t)$	Lagrange multipliers associated to the constraints.
$\mathbf{T}_E, \mathbf{T}_F, \mathbf{T}_G$	Tensors used to compute $E_k(t), F_k(t), G_k(t)$.
$\dot{\boldsymbol{\varphi}}(t), \ddot{\boldsymbol{\varphi}}(t)$	Velocity and acceleration of all nodes.
$\mathbf{M}, \mathbf{K}, \mathbf{D}$	Mass, stiffness and damping matrices.
$dt > 0$	Time step used to discretize the equations of motion.
$\boldsymbol{\varphi}^m, \dot{\boldsymbol{\varphi}}^m$	Position and velocity of the nodes at each instant $t_m = m \cdot dt$.
ϕ^m	Nodes of the meshed depth-camera point-cloud at t_m .

4. Discretization of the cloth with FEM

For the rest of the paper we will assume that our surface S has been discretized, i.e., that we have a polyhedron consisting on an ensemble of *quadrilaterals or triangles* Ω_e for $e \in \{1, \dots, n_q\}$ that approximates our original surface: $S \simeq \cup_e \Omega_e$ (see Figure 4). For the sake of clarity we will assume the use of quadrilaterals, although everything works the same with triangles. Let us state precisely the choices in our discretization scheme. Each element will have its own local parametrization:

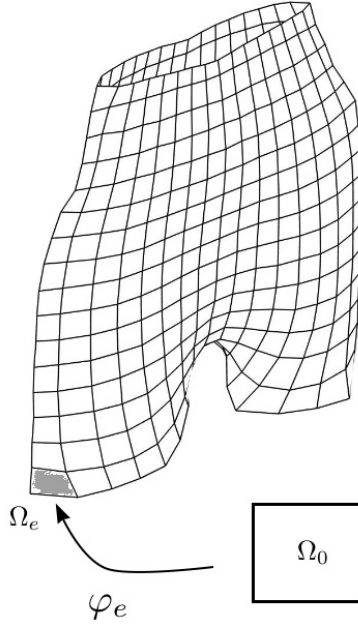


Figure 4: Bilinear parametrization of one of the elements of a quadrangulated surface (shorts).

$$\varphi_e : \Omega_0 := [-1, 1] \times [-1, 1] \rightarrow \Omega_e; \quad \varphi_e(\xi, \eta) = \sum_{i=1}^4 p_i^e N_i(\xi, \eta) \quad (4)$$

where the $p_i^e \in \Omega_e \subset \mathbb{R}^3$ are the 4 corners of the quadrilateral and the functions N_i are called the shape functions:

$$\begin{aligned} N_1 &= \frac{1}{4}(1 - \xi)(1 - \eta), & N_2 &= \frac{1}{4}(1 + \xi)(1 - \eta), \\ N_3 &= \frac{1}{4}(1 + \xi)(1 + \eta), & N_4 &= \frac{1}{4}(1 - \xi)(1 + \eta) \end{aligned}$$

and have the property that $N_i(q_j) = \delta_{ij}$ where δ_{ij} is Kronecker delta and the q_j 's are the four corners of Ω_0 starting at $(-1, -1)$ and going counter clockwise. Hence, $\varphi_e(q_i) = p_i^e$. If the surface is moving through space, the parametrizations $\varphi_e = \varphi_e(t)$ depend on time and this means that $p_i^e = p_i^e(t)$ vary with time.

Definition 2 (Nodes of the surface). Let us fix a *global* ordering of the n vertices of S . We will call them the nodes of the surface: $\text{Nodes}(S) = [p_1, \dots, p_n]$, where $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$. Then, we can construct the vector of positions of the textile for any time $t \geq 0$:

$$\begin{aligned}\boldsymbol{\varphi}(t) &:= (\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t))^\top \\ &= (x_1(t), \dots, x_n(t) | y_1(t), \dots, y_n(t) | z_1(t), \dots, z_n(t))^\top \in \mathbb{R}^{3n}.\end{aligned}$$

Definition 3 (Indicator functions). We can define global (but non-smooth) indicator functions $\mathcal{N}_i : \cup_e \Omega_e \rightarrow \mathbb{R}$ such that $\mathcal{N}_i(p_j) = \delta_{ij}$ as follows: given any element $\Omega_e \ni p_i$ we define $\mathcal{N}_i|_{\Omega_e} = N_k$ where $N_k(\varphi_e^{-1}(p_i)) = 1$. Thus, if $p = \varphi(t) \in S_t$ is any point of the moving surface, it can be written as

$$\varphi(t) = \sum_{i=1}^n p_i(t) \cdot \mathcal{N}_i(p), \quad (5)$$

and therefore $\partial_\xi \varphi(t) = \sum_{i=1}^n p_i(t) \partial_\xi \mathcal{N}_i$ and $\partial_\eta \varphi(t) = \sum_{i=1}^n p_i(t) \partial_\eta \mathcal{N}_i$.

Remark 4.1. Usually with FEM methodologies one parametrizes a region $R \subset \mathbb{R}^2$ where the values of a function φ are searched; however we parametrize each quadrilateral $\Omega_e \subset S$ independently. Therefore, with this approach S can have any given topology. Also, since $\varphi_e(0,0) = \frac{1}{4} \sum_{i=1}^4 p_i^e$, the mean of the nodes of each quadrilateral belongs to the element (even though the quadrilaterals are **not** flat, they are bi-linear), so when rendering we can get higher resolution by adding the middle point.

5. Constraint function enforcing inextensibility

We will construct a smooth and easy-to-evaluate constraint function

$$\mathbf{C} : \text{Nodes}(S_t) \simeq \mathbb{R}^{3n} \rightarrow \mathbb{R}^{n_c} \quad (6)$$

such that it is identically zero when all the n_c conditions ensuring inextensibility are satisfied. This function must depend only on the position vector of the nodes $\boldsymbol{\varphi}(t)$, and on nothing else. As an example, in the case of the isometric motion of a curve, we will preserve the lengths of all its edges, so for each edge k we add the constraint:

$$C_k(\boldsymbol{\varphi}(t)) = \|p_{k_1}(t) - p_{k_2}(t)\|_{\mathbb{R}^3}^2 - l_k^2, \quad (7)$$

where p_{k_1}, p_{k_2} are the two endpoints of edge k and $l_k > 0$ is the length of the edge at rest. We will apply these constraints to the boundary curves of our cloth S .

5.1. Weighted Galerkin residual method

On the other hand, we now explain how to discretize Equation (2) for constraining interior nodes. Substituting Equation (5) in the left hand side of the first equation of (2) we get:

$$\langle \varphi_\xi, \varphi_\xi \rangle(t) = \sum_{i,j=1}^n \langle p_i(t), p_j(t) \rangle \cdot \partial_\xi \mathcal{N}_i \partial_\xi \mathcal{N}_j.$$

The only problem with the previous equation is that it cannot be evaluated at the nodes because the derivative of the indicator functions is not defined there. Nevertheless, if we write:

$$\langle \varphi_\xi, \varphi_\xi \rangle(t) = \sum_{l=1}^n E_l(t) \cdot \mathcal{N}_l,$$

where $E_l(t)$ are the values of the first coefficient of the metric at the nodes, and then multiply by any indicator function \mathcal{N}_k and integrate over the surface, we obtain:

$$\sum_{l=1}^n E_l(t) \int_S \mathcal{N}_k \mathcal{N}_l dA = \sum_{i,j=1}^n \langle p_i(t), p_j(t) \rangle \cdot \int_S \mathcal{N}_k \partial_\xi \mathcal{N}_i \partial_\xi \mathcal{N}_j dA, \quad (8)$$

and therefore the coefficients $E_l(t)$ can be found by pre-multiplying the right hand side of (8) by the inverse of the *mass matrix*

$$M_{ij} = \int_{\cup_e \Omega_e} \mathcal{N}_i \mathcal{N}_j dA. \quad (9)$$

In practical implementations the mass matrix M is substituted by a diagonal matrix called the *lumped mass matrix* M_L defined as

$$(M_L)_{ii} = \sum_j M_{ij}. \quad (10)$$

We can imagine this process as collapsing all the mass of the surface to the nodes [24]. This is very convenient because it allows to compute the inverse of the mass matrix as the multiplication by the reciprocals of the constants $m_k := (M_L)_{kk}$.

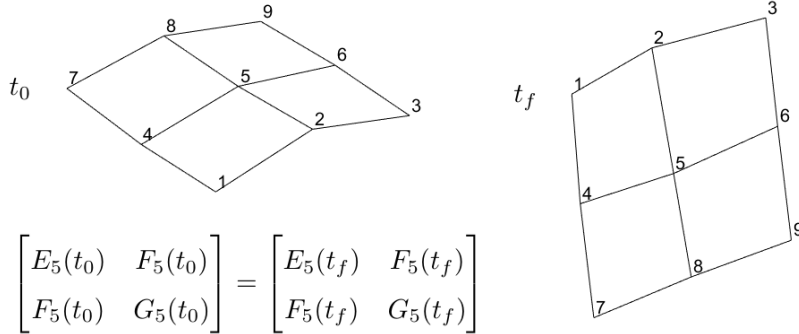


Figure 5: In order to have an isometry between the times t_0 and t_f , we impose the preservation of the first fundamental form at the central node 5, we constraint the length of the boundary edges $\vec{12}, \vec{23}, \dots, \vec{74}, \vec{41}$ and we enforce $F_k(t_0) = F_k(t_f)$ at the corners $k \in \{1, 3, 7, 9\}$.

Therefore

$$E_k(t) = \frac{1}{m_k} \sum_{i,j=1}^n \langle p_i(t), p_j(t) \rangle \cdot \int_S \mathcal{N}_k \partial_\xi \mathcal{N}_i \partial_\xi \mathcal{N}_j dA. \quad (11)$$

Naturally, similar equations are obtained when discretizing $\langle \varphi_\eta, \varphi_\eta \rangle(t)$ and $\langle \varphi_\xi, \varphi_\eta \rangle(t)$:

$$G_k(t) = \frac{1}{m_k} \sum_{i,j=1}^n \langle p_i(t), p_j(t) \rangle \cdot \int_S \mathcal{N}_k \partial_\eta \mathcal{N}_i \partial_\eta \mathcal{N}_j dA,$$

$$F_k(t) = \frac{1}{m_k} \sum_{i,j=1}^n \langle p_i(t), p_j(t) \rangle \cdot \int_S \mathcal{N}_k \partial_\xi \mathcal{N}_i \partial_\eta \mathcal{N}_j dA.$$

In summary, to impose constraints (2), for every interior node k and $t > 0$, we enforce $E_k(t) = E_k(0)$, $F_k(t) = F_k(0)$, $G_k(t) = G_k(0)$ (see Figure 5).

Algorithm 1 Computation of $\mathbf{T}_E, \mathbf{T}_F, \mathbf{T}_G$

```

1:  $m \leftarrow \# \text{nodes of each element}$   $\triangleright (m = 3 \text{ (triangles) or } m = 4 \text{ (quads)})$ 
2: for element  $\Omega_e$  do
3:    $\varphi_e := \sum_{i=1}^4 p_i^e N_i$   $\triangleright (\text{parametrization of } \Omega_e)$ 
4:    $I \leftarrow \text{Index}(\Omega_e)$   $\triangleright (\text{node's indices of } \Omega_e \text{ e.g. } [4, 48, 50, 13])$ 
5:   for Gauss point  $x_l$  and weight  $w_l$  do
6:      $E_l \leftarrow \varphi_\xi(x_l) \cdot \varphi_\xi(x_l)$ ;
7:      $F_l \leftarrow \varphi_\xi(x_l) \cdot \varphi_\eta(x_l)$ ;
8:      $G_l \leftarrow \varphi_\eta(x_l) \cdot \varphi_\eta(x_l)$ ;
9:      $dA_l \leftarrow \sqrt{|E_l \cdot G_l - F_l^2|} \cdot w_l$ ;
10:    for  $k, i, j = 1, \dots, m$  do
11:       $t_F^{kij} = N_k(x_l) \cdot \partial_\xi N_i(x_l) \cdot \partial_\xi N_j(x_l) \cdot dA_l$ ;
12:       $t_F^{kij} = \frac{1}{2} N_k(x_l) \cdot [\partial_\xi N_i(x_l) \cdot \partial_\eta N_j(x_l) + \partial_\eta N_i(x_l) \cdot \partial_\xi N_j(x_l)] \cdot dA_l$ ;
13:       $t_G^{kij} = N_k(x_l) \cdot \partial_\eta N_i(x_l) \cdot \partial_\eta N_j(x_l) \cdot dA_l$ ;
14:    end for
15:  end for
16:  for  $k, i, j = 1, \dots, m$  do  $\triangleright (\text{global assembly of the tensors})$ 
17:     $\mathbf{T}_E^{I(k)I(i)I(j)} += \frac{1}{m_{I(k)}} t_E^{kij}$ ;
18:     $\mathbf{T}_F^{I(k)I(i)I(j)} += \frac{1}{m_{I(k)}} t_F^{kij}$ ;
19:     $\mathbf{T}_G^{I(k)I(i)I(j)} += \frac{1}{m_{I(k)}} t_G^{kij}$ ;
20:  end for
21: end for

```

5.2. Efficient computation of the coefficients of the metric

We now discuss how to compute the right hand side of (11). Let $\text{Int}(S)$ denote the interior (non-boundary) nodes of S . We can define three *time independent* 3-tensors that will allow us to compute the 3 coefficients of the metric of S . First we have \mathbf{T}_E :

$$T_E^{kij} = \frac{1}{m_k} \int_{\cup_e \Omega_e} \mathcal{N}_k \partial_\xi \mathcal{N}_i \partial_\xi \mathcal{N}_j dA,$$

where $k \in \text{Int}(S)$ and $i, j \in \text{Nodes}(S)$. Similarly, \mathbf{T}_G is

$$T_G^{kij} = \frac{1}{m_k} \int_{\cup_e \Omega_e} \mathcal{N}_k \partial_\eta \mathcal{N}_i \partial_\eta \mathcal{N}_j dA,$$

and lastly \mathbf{T}_F , where $k \in \text{Int}(S) \cup \text{Corners}(S)$, is

$$T_F^{kij} = \frac{1}{2m_k} \int_{\cup_e \Omega_e} \mathcal{N}_k (\partial_\xi \mathcal{N}_i \partial_\eta \mathcal{N}_j + \partial_\eta \mathcal{N}_i \partial_\xi \mathcal{N}_j) dA.$$

For a pseudo-algorithm for the computation of the 3 tensors, see Algorithm 1.

Remark 5.1 (Symmetry of \mathbf{T}_F). This choice of the \mathbf{T}_F tensor is so that it is symmetric (as the other two tensors) in the i, j indexes and hence computing the Jacobian of \mathbf{C} will be simpler.

Remark 5.2 (Treatment of corners). On the other hand, in the presence of corners special care must be taken in order to avoid shearing at those points; that is why we include the corner's indices in the definition of \mathbf{T}_F (see Figure 5).

5.3. Constraints evaluation

This way, evaluating all the n_c constraints simultaneously reduces to computing the dot product matrix $\mathbf{P}(t) \in \mathbb{R}^{n \times n}$ defined by $P_{ij}(t) = \langle p_i(t), p_j(t) \rangle$ and then doing a tensor product (see Equation 11):

$$\mathbf{C}(\varphi(t)) = \mathbf{T}_{E,F,G} \otimes \mathbf{P}(t) - \mathbf{C}_0 \quad (12)$$

where we have grouped the previous tensors in just one $\mathbf{T}_{E,F,G} = [\mathbf{T}_E; \mathbf{T}_F; \mathbf{T}_G] \in \mathbb{R}^{n_c \times n \times n}$, and of course $\mathbf{C}_0 = \mathbf{T}_{E,F,G} \otimes \mathbf{P}(0)$. With this definition, computing the Jacobian matrix $\nabla \mathbf{C} \in \mathbb{R}^{n_c \times 3n}$ is very easy (because the tensors are symmetric in the last two indexes):

$$\nabla \mathbf{C}(\varphi(t)) = 2 \cdot [\mathbf{T}_{E,F,G} \otimes \mathbf{x}(t), \mathbf{T}_{E,F,G} \otimes \mathbf{y}(t), \mathbf{T}_{E,F,G} \otimes \mathbf{z}(t)], \quad (13)$$

recall that $\varphi(t) = (\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t))^\top \in \mathbb{R}^{3n}$.

5.4. Practical implementation

We now list some considerations to take into account in a practical implementation of our method:

- As mentioned before, the constraints (7) are added to \mathbf{C} to impose that every edge of all boundary curves are preserved. Also, we remark again that the tensor constraints are only applied for interior nodes, except in the presence of corners, where we include their indices in the definition of \mathbf{T}_F in order to avoid shearing there.
- The tensor $\mathbf{T}_{E,F,G}$ needs to be computed only once at the beginning of the simulation because it is time independent. All the integrals involved are evaluated exactly using standard *Gaussian quadratures* (see [24] for details) as shown in Algorithm 1.
- In order to be efficient we first calculate the Jacobian (according to Equation 13) and then put $\mathbf{C}(\varphi) = \frac{1}{2} \nabla \mathbf{C}(\varphi) \cdot \varphi - \mathbf{C}_0$, so the matrix $\mathbf{P}(t)$ is actually never computed.
- The tensor $\mathbf{T}_{E,F,G}$ is highly sparse due to the fact that most products of the 3 indicator functions (Equation 11) are zero. Nevertheless, it is not difficult to see that it has of the order of $\sim 150n$ non-zero elements.

6. Equations of motion of the cloth

Finally we can write down the Lagrangian (kinetic minus potential energies) of our discretized surface:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\varphi}(t), \dot{\boldsymbol{\varphi}}(t)) &= \frac{\rho}{2} \dot{\boldsymbol{\varphi}}(t)^\top \cdot \mathbf{M} \cdot \dot{\boldsymbol{\varphi}}(t) - \rho \mathbf{g}^\top \cdot \mathbf{M} \cdot \boldsymbol{\varphi}(t) \\ &\quad - \frac{\kappa}{2} \boldsymbol{\varphi}(t)^\top \cdot \mathbf{K} \cdot \boldsymbol{\varphi}(t) - \boldsymbol{\lambda}(t)^\top \cdot \mathbf{C}(\boldsymbol{\varphi}(t)),\end{aligned}$$

where

1. $\rho > 0$ is the density of the cloth (assuming homogeneous mass), \mathbf{M} is the augmented mass matrix and $\mathbf{g} = (0, \dots, 0 | 0, \dots, 0 | g, \dots, g)^\top$ where $g = 9.8m/s^2$ is gravity,
2. the stiffness matrix (we are using the isometric bending model described in [39]) is $\mathbf{K} = \mathbf{L}^\top \mathbf{M} \mathbf{L}$ where \mathbf{L} is an approximation of the point-wise Laplacian and $\kappa > 0$ is a bending constant,
3. and finally $\boldsymbol{\lambda}(t)$ are the Lagrange multipliers ensuring inextensibility (at the interior nodes and boundaries) and other possible positional constraints (e.g. the corners of the textiles to be manipulated) included.

Thus, we get as Euler-Lagrange equations [40] the following ODE system:

$$\begin{cases} \rho \mathbf{M} \ddot{\boldsymbol{\varphi}} = \mathbf{f}_\rho - \kappa \mathbf{K} \boldsymbol{\varphi} - \mathbf{D} \dot{\boldsymbol{\varphi}} - \nabla \mathbf{C}(\boldsymbol{\varphi})^\top \boldsymbol{\lambda} \\ \mathbf{C}(\boldsymbol{\varphi}) = 0 \end{cases} \quad (14)$$

where $\mathbf{f}_\rho = -\rho \mathbf{M} \mathbf{g}$ is the force of gravity and we have added Rayleigh damping: $\mathbf{D} = \alpha \mathbf{M} + \beta \mathbf{K}$, where α and β are positive parameters [41].

Notice how our inextensibility assumption reduces greatly the number of physical parameters of the model (we do not have shearing or stretching parameters and their respective dampings). Nevertheless, cloth dynamics can be very complicated and there is one important factor we are not accounting for in the previous equations: air resistance. Although there exist some simplified models [2], aerodynamic forces of a deformable object (i.e. cloth) submerged in a fluid (i.e. air) are difficult to model (see [42] and [43]), especially near the boundaries of cloth, because turbulences appear and the nonrigid response of cloth to such turbulences is way more unpredictable than that of a rigid, even vibrating, body (see [3]).

6.1. Modeling of aerodynamics through virtual mass

The equivalence principle states that for any object its inertial mass is equal to its gravitational mass, it means that an object in vacuum of inertial mass m would fall freely under the action of a force of magnitude mg , where g is gravity. Nevertheless, experiments show that in the presence of air the free falling velocities depend on the (shape of the) object at hand. In our experiments we have found that aerodynamic effects can be modeled without explicitly including them by allowing inertial and gravitational masses to be different. Although this is not true in the physical world, the consideration of the two masses as virtual in our model allows us to account for all aerodynamic contributions to cloth dynamics (drag, lift, turbulences at the boundaries) with great accuracy. Hence, we put $\mathbf{f}_\delta = -\delta \mathbf{M} \mathbf{g}$, setting δ as a new parameter to be fitted. In Table 1 we summarize the meaning of all the parameters of our model.

Parameter	Meaning
ρ	Density (inertial mass)
δ	Virtual (gravitational) mass
κ	Bending/stiffness
α	Damping of slow oscillations
β	Damping of fast oscillations

Table 1: Physical parameters of the model and their meaning.

7. Discretization of the equations of motion

As usual, we approximate $\varphi(t)$ and $\dot{\varphi}(t)$ with $\{\varphi^0, \varphi^1, \dots\}$ and $\{\dot{\varphi}^0, \dot{\varphi}^1, \dots\}$, where φ^n and $\dot{\varphi}^n$ are the position and velocities of the nodes of the mesh at time $t_n = n \cdot dt$ and $dt > 0$ is the size of the time step. Using an implicit Euler scheme to integrate Equations (14) we obtain:

$$\begin{cases} \varphi^{n+1} = \varphi_0^{n+1} - \mathbf{M}^{-1} \nabla \mathbf{C}(\varphi^{n+1})^\top \boldsymbol{\lambda}^{n+1} \\ \mathbf{C}(\varphi^{n+1}) = 0, \end{cases} \quad (15)$$

where φ_0^{n+1} is the unconstrained step (which depends on φ^n and $\dot{\varphi}^n$). Note that at time $t_n = n \cdot dt$ the only unknown in previous equations is φ^{n+1} (and $\boldsymbol{\lambda}^{n+1}$).

7.1. Fast projection algorithm

Finding φ^{n+1} can be done solving the following constrained optimization problem:

$$\begin{cases} \min_{\varphi^{n+1}} & (\varphi^{n+1} - \varphi_0^{n+1})^\top \cdot \mathbf{M} \cdot (\varphi^{n+1} - \varphi_0^{n+1}) \\ \text{s.t.} & \mathbf{C}(\varphi^{n+1}) = 0, \end{cases} \quad (16)$$

because Equations (15) are the critical (stationary) points of the optimization problem. This is a quadratic program with quadratic constraints which can be solved with Newton's method. Nevertheless, in order to make it computationally tractable in front of difficulties such as indefinite system matrices, it is approximated by a *sequence* of quadratic programs with linear constraints. Write $\varphi_{j+1} = \varphi_j + \Delta\varphi_{j+1}$ and make the approximation $\mathbf{C}(\varphi_{j+1}) = \mathbf{C}(\varphi_j + \Delta\varphi_{j+1}) \simeq \mathbf{C}(\varphi_j) + \nabla \mathbf{C}(\varphi_j) \Delta\varphi_{j+1}$, then the sequence is:

$$\begin{cases} \min_{\Delta\varphi_{j+1}} & \Delta\varphi_{j+1}^\top \cdot \mathbf{M} \cdot \Delta\varphi_{j+1} \\ \text{s.t.} & \mathbf{C}(\varphi_j) + \nabla \mathbf{C}(\varphi_j) \cdot \Delta\varphi_{j+1} = 0, \end{cases}$$

the initial point being $\varphi_0 = \varphi_0^{n+1}$. This is called the *fast projection algorithm* [5]. Each of these quadratic programs can be reduced to solving a linear system and we iterate until the constraints are satisfied to a given relative tolerance (usually 0.1%). We have found this algorithm very stable and fast enough (since we are simulating mainly coarse meshes) for our purposes.

7.2. Contact and collisions with an obstacle

For its application in the real world, we need to include into our model collisions of the cloth with a fixed object (e.g. a table). We will assume this obstacle is given by

an implicit equation $\mathbf{H}(\boldsymbol{\varphi}) = 0$ with a well defined outwards normal $\nabla\mathbf{H}(\boldsymbol{\varphi})$ (almost everywhere). We can model collisions by including new (non-smooth) forces in every iteration of the *fast projection algorithm* described earlier. Signorini's contact model then reads (see [44] and [45]):

$$\begin{cases} \mathbf{M}\ddot{\boldsymbol{\varphi}} = \mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) - \nabla\mathbf{C}(\boldsymbol{\varphi})^\top \boldsymbol{\lambda} + \nabla\mathbf{H}(\boldsymbol{\varphi})^\top \boldsymbol{\gamma}, \\ \mathbf{C}(\boldsymbol{\varphi}) = 0, \\ \mathbf{H}(\boldsymbol{\varphi}) \geq 0, \quad \boldsymbol{\gamma} \geq 0, \quad \boldsymbol{\gamma}^\top \cdot \mathbf{H}(\boldsymbol{\varphi}) = 0, \end{cases} \quad (17)$$

where we have grouped in the force term $\mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})$ damping, gravity, etc. Now the system is non-smooth, that is why we need to use a first-order scheme [44]. To integrate it numerically, we put again $\boldsymbol{\varphi}_{j+1} = \boldsymbol{\varphi}_j + \Delta\boldsymbol{\varphi}_{j+1}$ where the initial point is the unconstrained step $\boldsymbol{\varphi}_0 = \boldsymbol{\varphi}_0^{n+1}$ given by an implicit Euler scheme. Also, we write:

$$\mathbf{H}(\boldsymbol{\varphi}_{j+1}) = \mathbf{H}(\boldsymbol{\varphi}_j + \Delta\boldsymbol{\varphi}_{j+1}) \simeq \mathbf{H}(\boldsymbol{\varphi}_j) + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1},$$

and then solve iteratively the following sequence of quadratic programs with linear equality and inequality constraints:

$$\begin{cases} \min_{\Delta\boldsymbol{\varphi}_{j+1}} \frac{1}{2} \Delta\boldsymbol{\varphi}_{j+1}^\top \cdot \mathbf{M} \cdot \Delta\boldsymbol{\varphi}_{j+1} \\ \mathbf{C}(\boldsymbol{\varphi}_j) + \nabla\mathbf{C}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = 0, \\ \mathbf{H}(\boldsymbol{\varphi}_j) + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} \geq 0. \end{cases}$$

Remark 7.1. Note that the critical points (compare to (17)) of the previous quadratic problems are:

$$\begin{cases} \mathbf{M} \cdot \Delta\boldsymbol{\varphi}_{j+1} = -\nabla\mathbf{C}(\boldsymbol{\varphi}_j)^\top \Delta\boldsymbol{\lambda}_{j+1} + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)^\top \Delta\boldsymbol{\gamma}_{j+1}, \\ \mathbf{C}(\boldsymbol{\varphi}_j) + \nabla\mathbf{C}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = 0, \\ \mathbf{H}(\boldsymbol{\varphi}_j) + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} \geq 0, \\ \Delta\boldsymbol{\gamma}_{j+1} \geq 0, \quad \Delta\boldsymbol{\gamma}_{j+1}^\top \cdot [\mathbf{H}(\boldsymbol{\varphi}_j) + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1}] = 0. \end{cases}$$

8. Evaluation and results

8.1. Locking test

With this experiment we intend to show the locking-free nature of our model, and its stability with respect to mesh topology. We fix three corners of a flat sheet of cloth (with added random noise of standard deviation 3mm) of 1m by 1m, and let the fourth corner fall freely. We use four different meshings of the cloth: two with triangles and two with quadrilaterals (see Figure 6), but we keep the physical parameters fixed.

The visual results of the experiment can be seen in Figure 7. The four textiles fold diagonally without any locking artifact.

In Table 2 we compute the euclidean distance of the bottom corner of every textile with respect to each other. Note how, doubling the number of nodes, or changing from irregular triangles to regular quadrilaterals, does not alter the result within a margin of error of a few centimeters (recall that the sheets are 1m×1m).

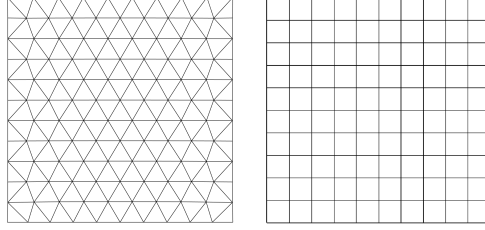


Figure 6: Different meshes used for the locking test. The triangular meshes (left) are irregular whereas the quadrilateral ones (right) are uniform.

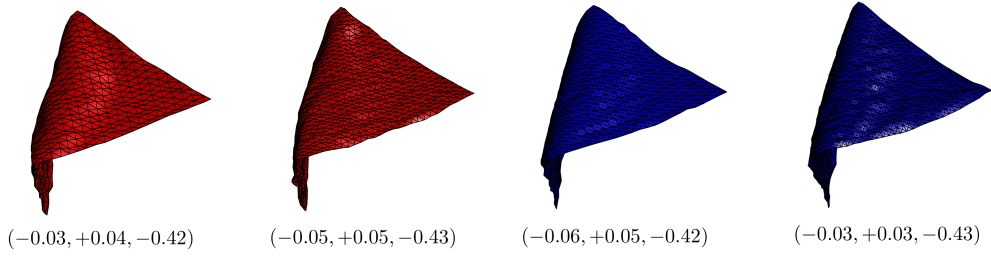


Figure 7: Four different meshes used for the experiment: the red ones are made of triangles and the blue ones of quadrilaterals. From left to right the number of nodes is: 472, 941, 529, 900. On the bottom of each cloth, we display the euclidean position of the free corner.

Nodes / Element	472 / \triangle	941 / \triangle	529 / \square	900 / \square
472 / \triangle	0	2.45 cm	3.16 cm	1.41 cm
941 / \triangle	-	0	1.41 cm	2.83 cm
529 / \square	-	-	0	3.74 cm
900 / \square	-	-	-	0

Table 2: Distances (cm) between the bottom corner of the cloths (of dimensions $1\text{m} \times 1\text{m}$).

8.2. Cusick's test

In this section we explain how to simulate Cusick's test [14] using our simulator. This subsection will demonstrate the stability of our model under refinements of the mesh. Cusick's test consists in letting a circular cloth of radius 15cm drape on top of an also circular table of radius 9cm (see Figure 8) with their centers aligned.



Figure 8: Photo of a circular cloth draping on top of a circular table. Image taken from [46].

Then, one computes the area of the plane projection of the draped cloth C and divides it by the area of the flat ring A of width 6cm (see Figure 9).

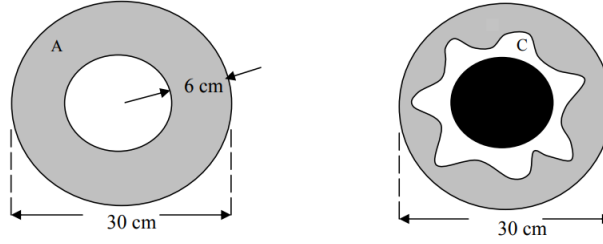


Figure 9: After the cloth has draped on top of the table, the drape coefficient is calculated as the ratio between the area of the plane projection of the cloth C , divided by the area of the flat ring A of width 6cm. Image adapted from [46].

This is called the **drape coefficient** of the cloth:

$$\text{DC \%} = 100 \times \frac{C}{A}. \quad (18)$$

It is known that this coefficient depends heavily on the stiffness of the cloth [14]. This can be understood intuitively: a completely rigid sheet of metal would not bend and would have DC equal to 100%, whereas a really light material (e.g. silk) has a DC pretty close to 0. The measurement of this coefficient is not trivial: it usually requires a dedicated machine (Cusick's Drape Tester [17]); and variation of the measured coefficient for the same cloth is common, so several specimens of the same textile have to be employed and then their DC's averaged [46].

Naturally, this has caused the creation of alternative simulation methods (e.g. [47]). In order to simulate Cusick's test with our model, we quadrangulate the ring A of Figure 9, we fix in space its inner boundary nodes and let gravity act (see Figure 10). In

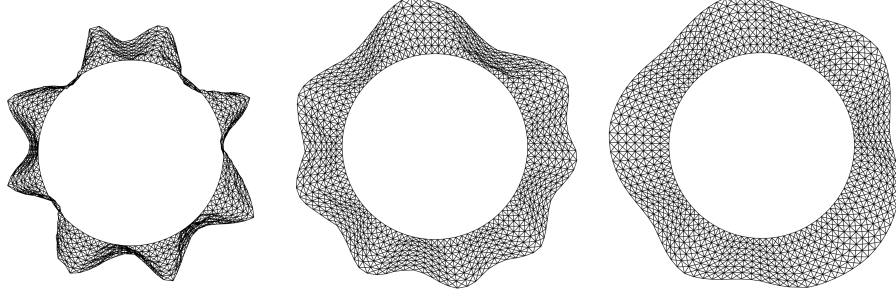


Figure 10: Simulation of Cusick’s test with the inextensible model and a mesh of 768 nodes. From left to right we vary the stiffness of the cloth and get respectively DCs of 26.1%, 54.9%, 77.2%. They correspond to *peach-skin* polyester (low stiffness), *imitation* wool (medium) and *tussore* cotton (high).

order to study the dependency of the DC with respect to mesh resolution, in Figure 11 we plot the DC for three stiffness values κ (low: κ_0 , medium: $5\kappa_0$ and high: $10\kappa_0$; where $\kappa_0 = 0.005$) and 24 different meshes of the ring A ranging from 250 nodes to 1300. The three mean values for the computed DCs are 23.2%, 52.4%, 77.3% and they approximately correspond to *peach-skin* polyester (low stiffness), *imitation* wool (medium stiffness) and *tussore* cotton (high stiffness) (see the Appendix of [46]: IDs: 38, 37, 22). The figure shows that the computation of the DC with our model is very stable (especially from 700 nodes on), and hence the model has a robust behavior with respect to mesh resolution. As we already said, this is very relevant for robotic applications, where the use of coarse meshes becomes a necessity because of performance constraints.

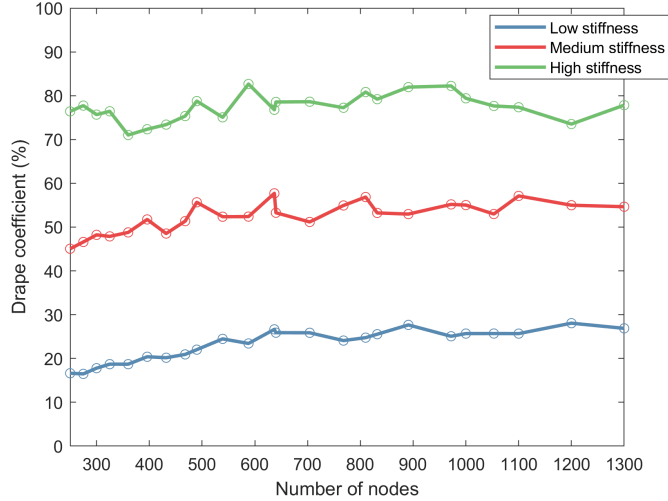


Figure 11: Computation of the drape coefficient for 3 stiffness values (low, medium and high) and 24 different meshes of cloth from 250 to 1300 nodes. The three mean values for the computed DCs are 23.2%, 52.4%, 77.3%.

8.3. Tank-top simulation

With this experiment we aim to show the ability of our model to simulate complex topologies (e.g. a tank-top shirt, see Figure 12 and the Supplementary Video 1) and study different area errors. We track the variation of area of the garments through time. We think area is a good measure since it is a physical property of cloth and not a mesh-dependent metric (e.g. stretching of the edges of the elements). We compute the area of each element of the discretized cloth using its local parametrization (4).

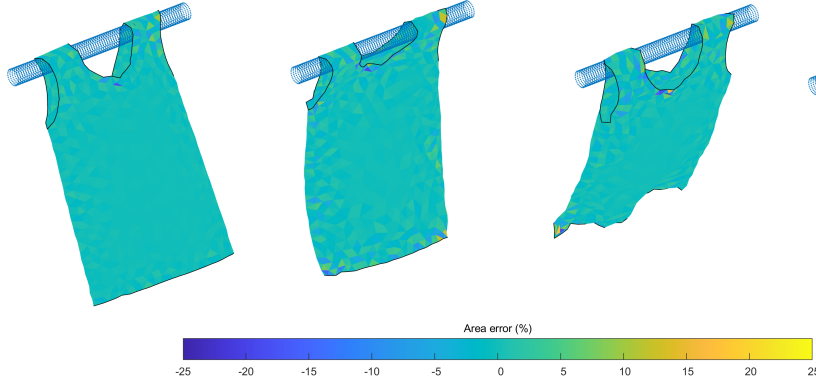


Figure 12: Simulation with the inextensible model of the shaking of a triangulated tank-top. At each time instant ($t = 1.1s, 1.75s, 2.5s, 3.5s$) we plot the area error with sign (22) of each individual triangle.

We use three area errors:

1. *Total area error*:

$$e_t(t) = \frac{|A(t) - A_0|}{A_0}, \quad (19)$$

where A_0 is the total area of the garment at time $t = 0$ and $A(t)$ is its area at time $t > 0$.

2. *Mean element error*:

$$e_m(t) = \frac{1}{n_e} \sum_e \frac{|a_e(t) - a_e(0)|}{a_e(0)}, \quad (20)$$

where $a_e(t)$ is the area of element Ω_e at time $t \geq 0$.

3. *Dispersion*:

$$d_a(t) = e_m(t) + 2\sqrt{\text{Var}_e \left(\frac{|a_e(t) - a_e(0)|}{a_e(0)} \right)}. \quad (21)$$

We simulate the shaking with a *hanger* of a meshed tank-top with 1676 triangles (see Figure 12) during 4.5s. There, we also plot the area error with sign of each individual triangle for each time instant:

$$e(t) = \frac{a_e(t) - a_e(0)}{a_e(0)}. \quad (22)$$

It is interesting to notice the concentration of error near the boundaries of the cloth and at the points where the tank-top makes contact with the moving cylinder. Nevertheless, note that when a triangle is stretched (color yellow), next to it appears a triangle that contracts (color blue). This makes the total area error to be very small (see Figure 13). Finally in Figure 13, we plot the three area errors (19),(20) and (21). The total area error is very low during all the simulation, almost zero. The mean area error is higher, but still very low (it peaks around $t = 1.75$ s, the second frame in Figure 12), keeping itself around 1%. The dispersion error on the other hand, tells us that if the distribution of the errors were normal, the error of 95% of the mesh triangles would be under 7% during all the simulation.

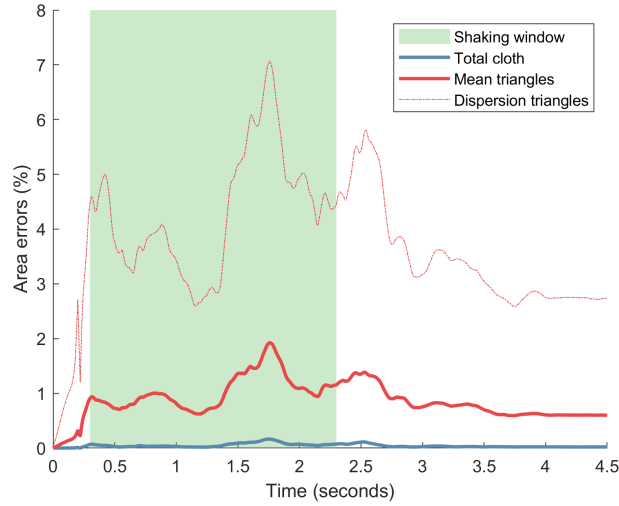


Figure 13: Plot of the time-dependent errors: total area (19), mean element (20) and dispersion (21). Note how the errors increase during the complex parts of the simulation (from $t = 0.5$ s to $t = 3$ s) and decrease afterwards.

9. Experimental validation

9.1. Camera data

Comparison with reality is performed by recording in the laboratory the motion of a piece of cloth when subjected to exactly the same movements of the robotic arm controlling it as specified in the simulations with our model. The real motion is captured by a depth camera Kinect XB360-607 (see Figure 14). The garment is a rectangular piece of cloth, with its two upper corners fixed to a rigid hanger, which is affixed itself to a robotic arm Barrett WAM. To keep the garment completely in view of the camera, the robot moves it along the depth axis of the camera (axis x in our reference) following a curve of equation:

$$(x(t), y(t), z(t)) = (A \cos(2\pi ft) + c, y_0, z_0). \quad (23)$$

The two corners of the garment fixed to the robot arm follow the same oscillation, with different but constant values y_1, y_2 instead of y_0 .

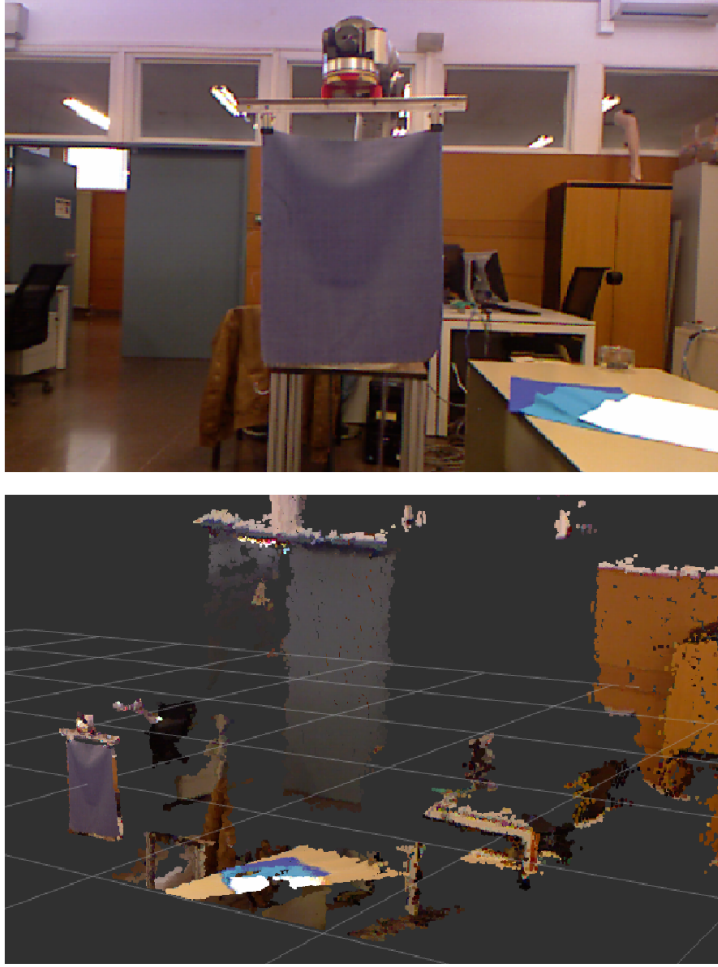


Figure 14: Point-cloud obtained using a depth camera (bottom) and RGB image (top). Note that the camera only gets the depth of the objects visible to it, all the rest (e.g. the robot behind the cloth) is occluded.

Two motions are recorded: the *slow* one has $A = 0.15\text{m}$ and $f = 0.3\text{Hz}$, while the *fast* one has $A = 0.075\text{m}$ and $f = 0.6\text{Hz}$. Surrounding objects are filtered using planes for each recorded frame, and outlying points are detected on the basis of distance and removed. Only garments with a homogeneous color are tested, so we can remove further noise through use of a color filter. After the point cloud has been thus filtered, it is meshed using the algorithm described in [48] (see Figure 15). The model that we are validating is continuous, producing simulations that are very stable under remeshing. Because of this, it suffices to select as a baseline for all the experiments a coarse 9×9 mesh.

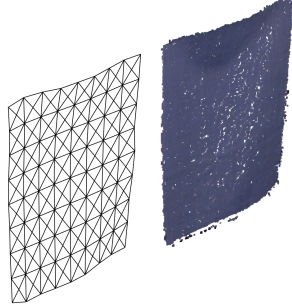


Figure 15: Quadrilateral meshing (left) of one of the frames of the filtered and de-noised point-cloud (right). Each quadrilateral is divided into triangles for plotting purposes.

The robotic arm follows the oscillation of Equation (23) for 10 seconds. The camera starts recording 1 second before the start, and finishes 3 seconds after the end of the robotic motion, for a total of 14 seconds of recorded garment motion. The original recording has its frames at irregular time steps. We replace them through linear interpolation to obtain the meshed cloth’s position each $dt = 0.01$ seconds.

9.2. Parameter fitting

We will be adjusting only 2 parameters: the damping parameter α and the (virtual) gravitational mass δ . The other parameters in Table 1 are set to 0 except for $\rho = 1$, since these two parameters are the most relevant for the motions performed in this experimental validation (see [49]). In Section 9.1 we explained how to obtain a sequence of positions of the nodes of the real cloth $\{\phi^0, \phi^1, \dots, \phi^m\}$. If we integrate numerically Equation (14) using the same trajectories (23) for the two upper corners, we get a sequence $\{\varphi^0(\delta, \alpha), \varphi^1(\delta, \alpha), \dots, \varphi^m(\delta, \alpha)\}$ (where $\varphi^0 = \phi^0$) for each value of the two parameters. Hence, a natural metric is:

$$L(\delta, \alpha) = \sum_{i=1}^{\lfloor \frac{m}{2} \rfloor} \|\varphi^i(\delta, \alpha) - \phi^i\|_{\mathbf{M}}^2, \quad (24)$$

where $\|\cdot\|_{\mathbf{M}}$ is the L^2 norm with respect to the matrix \mathbf{M} (i.e. $\|\mathbf{x}\|_{\mathbf{M}}^2 = \mathbf{x}^\top \cdot \mathbf{M} \cdot \mathbf{x}$), and we only use the first half (7 seconds) of the recorded frames to perform the fitting. We call this the *training window*. Finally, we minimize L using a derivative-free algorithm (the Nelder-Mead Simplex Method) and find the optimal values of δ, α . The resulting parameters are shown in Table 3. We will analyze the evolution of the time-dependent absolute error:

$$e_i(\delta, \alpha) = \sqrt{\|\varphi^i(\delta, \alpha) - \phi^i\|_{\mathbf{M}}^2}. \quad (25)$$

In the last two columns of Table 3 we show the average values of e_i over the *testing window* (i.e. the last 7 seconds of the motion). Moreover, we use two times the standard deviation of the error at each node j as a dispersion measure:

Material	δ_{slow}	δ_{fast}	α_{slow}	α_{fast}	\bar{e}_{slow}	\bar{e}_{fast}
Paper	0.32	0.37	1.40	2.49	0.45 cm	0.41 cm
Cotton	0.52	0.52	1.29	2.69	0.41 cm	0.31 cm
Felt	0.46	0.61	1.05	2.65	0.39 cm	0.30 cm
Wool	0.47	0.50	1.19	2.52	0.37 cm	0.34 cm

Table 3: Estimated parameters and mean absolute errors (cm).

$$d_i(\delta, \alpha) = e_i(\delta, \alpha) + 2\sqrt{\text{Var}_{j \in \text{Nodes}(S)} (\|\varphi_j^i(\delta, \alpha) - \phi_j^i\|_{\mathbb{R}^3})}. \quad (26)$$

9.3. Sensitivity analysis

In order to show how robust our model is when away from the optimal values found in Table 3 we perform a sensitivity analysis. We carry out 100 different simulations with different values of the parameters $\alpha \geq 0$ and $\delta > 0$ (with an upper limit of two times their optimal value), and compute the mean of the absolute error (25) on the testing window formed by the last 7 seconds of the movement. The results for the fast motion of paper (see Figure 2) are shown in Figure 16.

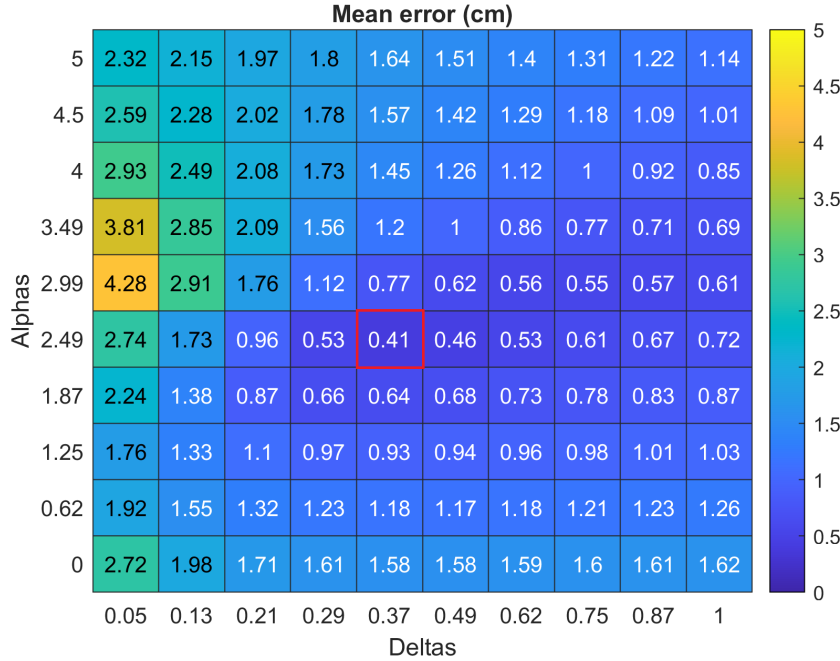


Figure 16: Mean of the absolute error (25) for the fast motion of paper using different values of the parameters of the model (damping α and virtual mass δ). In red we highlight the error with the optimal value of the parameters.

First of all, note that for the discrete set of values considered, the optimal values found are in fact a global minimum. It is interesting to note that the model is very robust with respect to the virtual mass δ , and in fact even when $\delta = 1$ (no aerodynamics), there is a value of $\alpha = 2.99$ that gives a very low overall error (0.61 cm). Other interesting limit cases are $\alpha = 0$ (no damping) and $\alpha = 5$ (over-damped), but they give higher errors. Nevertheless, we do not consider $\delta = 0$ (no gravity), since it is very unrealistic, and that is confirmed in the picture since the largest errors occur around $\delta = 0.05$.

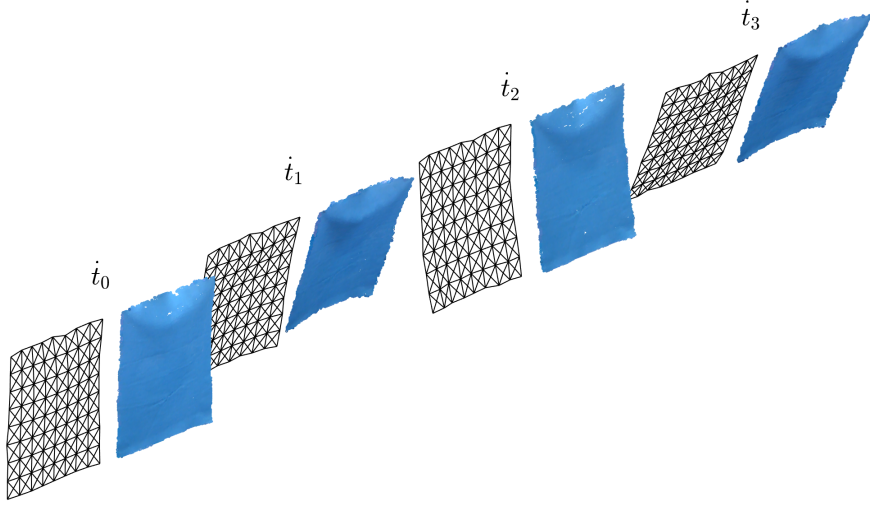


Figure 17: Comparison at 4 time instants of the fast motion of light cotton (right) versus its simulation with the inextensible model (left) with $\delta = 0.52$ and $\alpha = 2.69$. The mean absolute error is 0.31 cm.

9.4. Comparison with other models

We compare our method with three other models for the fast motion of cotton (see Figure 17). We study the evolution in time of the absolute error (25) and the dispersion error (26); the curves are shown in Figure 18. Moreover, we look at 4 different metrics: the mean of the absolute error (25), the maximum of the dispersion error (26), the total simulation time (in seconds) and, when applicable, the mean number of iterations per step of the *fast projection algorithm* described in Section 7.1. In Table 4 we display the metrics, the first two only computed over the last 7 seconds of the movement. The models considered are:

1. Quasi-inextensible model [5]: this is a discrete model in which stretching is not allowed because all the edges of the quadrangulated cloth are constrained to maintain their length, and only shearing is permitted by using non-stiff diagonal springs.
2. Continuum-elastic model [23]: this is a very popular elastic continuous model. Stretching and shearing are both permitted by using elastic energies instead of constraints.
3. Mass-spring system e.g. [26] and [27]: this is arguably the most widespread cloth model, used by popular animation and robotics software such as Blender [50] and

MuJoCo [51]. Stretching and shearing are both allowed by using springs to connect the nodes of the cloth. There are no constraints.

We fit the relevant parameters of each of the models (shearing, stretching, damping, etc.) using the same framework applied to find the optimal parameters of our model (Section 9.2). All comparisons are performed using a Intel Core i7-8700K with 12 cores of 3.70 GHz.

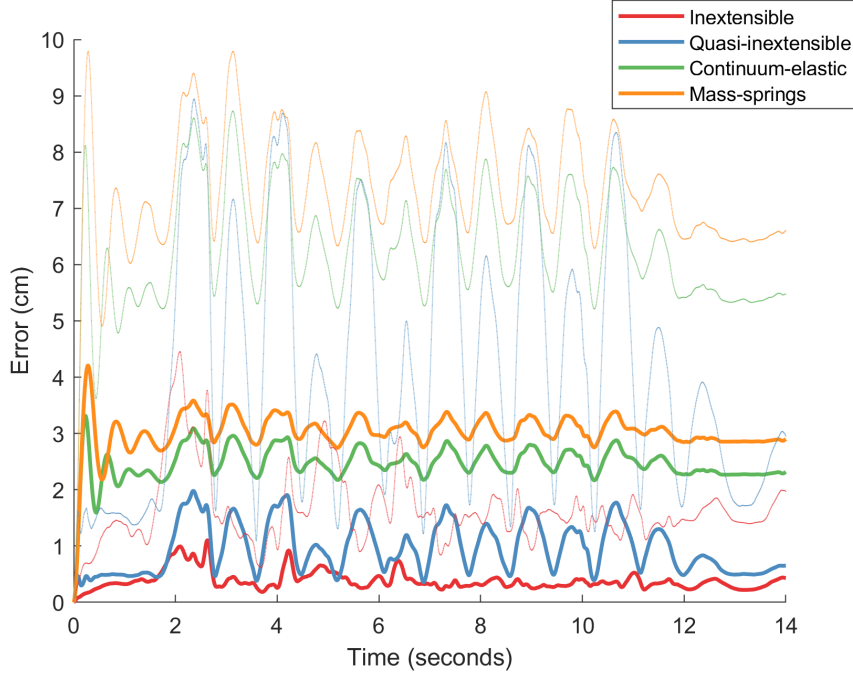


Figure 18: Comparison of the fast motion of light cotton between the inextensible and the other 3 models’ errors (bottom curves) and dispersions (top curves), with respect to the recorded motion (using a 9×9 meshing). The mean absolute error for the inextensible model is 0.31cm.

The first thing to remark is that the inextensible model has overall the lowest mean absolute error, being 3 times smaller than the quasi-inextensible one, and almost 10 times smaller than the elastic models (see Table 4). This supports the idea that inextensibility is a very realistic assumption. Moreover the dispersion curves in Figure 18 tell us even more: in the second half of the motion there are nodes that have absolute errors over 7cm with the other models, whereas with ours, they are at most 2cm apart from the recorded motion. On the other hand, the two elastic models have the smallest simulation times, and this is easily explainable by the fact that they do not have constraints and hence the numerical integration is simpler. Nevertheless, our model simulates 14 seconds of real time in only 2.10, which implies that we get a speed up of seven times with respect to real time. Finally, the inextensible model requires less iterations of the *fast projection algorithm* than the quasi-inextensible one, but each iteration is more expensive, so in the end they have similar total simulation times.

Model	\bar{e}	$\max(d)$	Total time	Iterations
Inextensible	0.31 cm	2.02 cm	2.10 sec	1.60
Quasi-inextensible	0.95 cm	8.35 cm	2.20 sec	4.51
Continuum-elastic	2.46 cm	7.87 cm	1.16 sec	-
Mass-springs	3.03 cm	9.07 cm	1.30 sec	-

Table 4: Comparison of the 4 models. The first column is the mean of the absolute error (25) of the simulated cloth with respect to the recorded motion.

9.5. Discussion of the results

As shown in Figure 19, the error (25) for all four textiles and both motions are very low (its mean is under 5mm, see Table 3). For a visual comparison of the results, see Figures 2, 17, 20 and the Supplementary Video 2. A phenomenon that is noticeable is the concentration of error (clearly seen in the dispersion curves) at the beginning of the movement; likely due to the fact that air turbulences are more complicated at this stage. In regards to the parameter values, we note their discrepancy when comparing the fast and the slow motions (using the same textile), showing that they actually account for the aerodynamics of the motion. While this approach is not standard, note that using only 2 parameters we are able to predict cloth dynamics accurately.

It is relevant to emphasize that these results are obtained using a 9×9 mesh (with which our simulations are 7 times faster than real time) for a piece of cloth of size A3 (29.7×42 cm). Keeping the obtained optimal parameters, we also computed the error (25) for other three mesh resolutions 9×17 , 17×9 and 17×17 , getting respectively 0.37, 0.34 and 0.35cm as the mean of the absolute error (25) for the fast motion of light cotton (see Figure 17; with the 9×9 mesh the error is 0.31cm). This shows again (see Section 8.2) the robustness of the model with respect to mesh resolution.

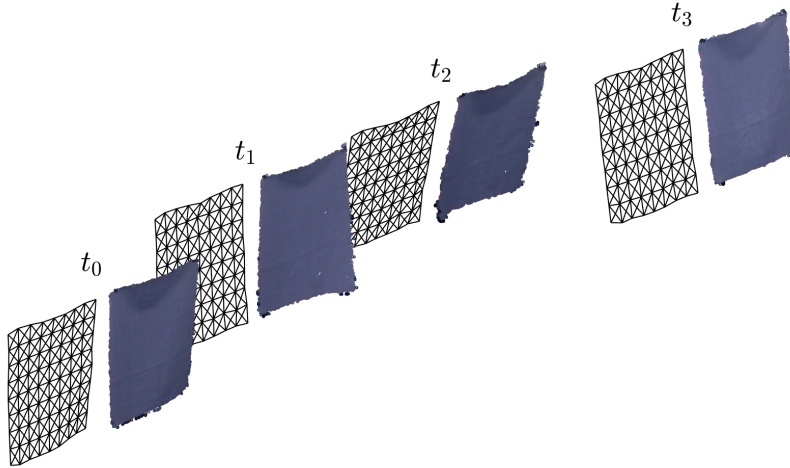


Figure 20: Comparison at 4 time instants of the slow motion of wool (right) versus its simulation with the inextensible model (left) with $\delta = 0.47$ and $\alpha = 1.19$. The mean absolute error is 0.37 cm.

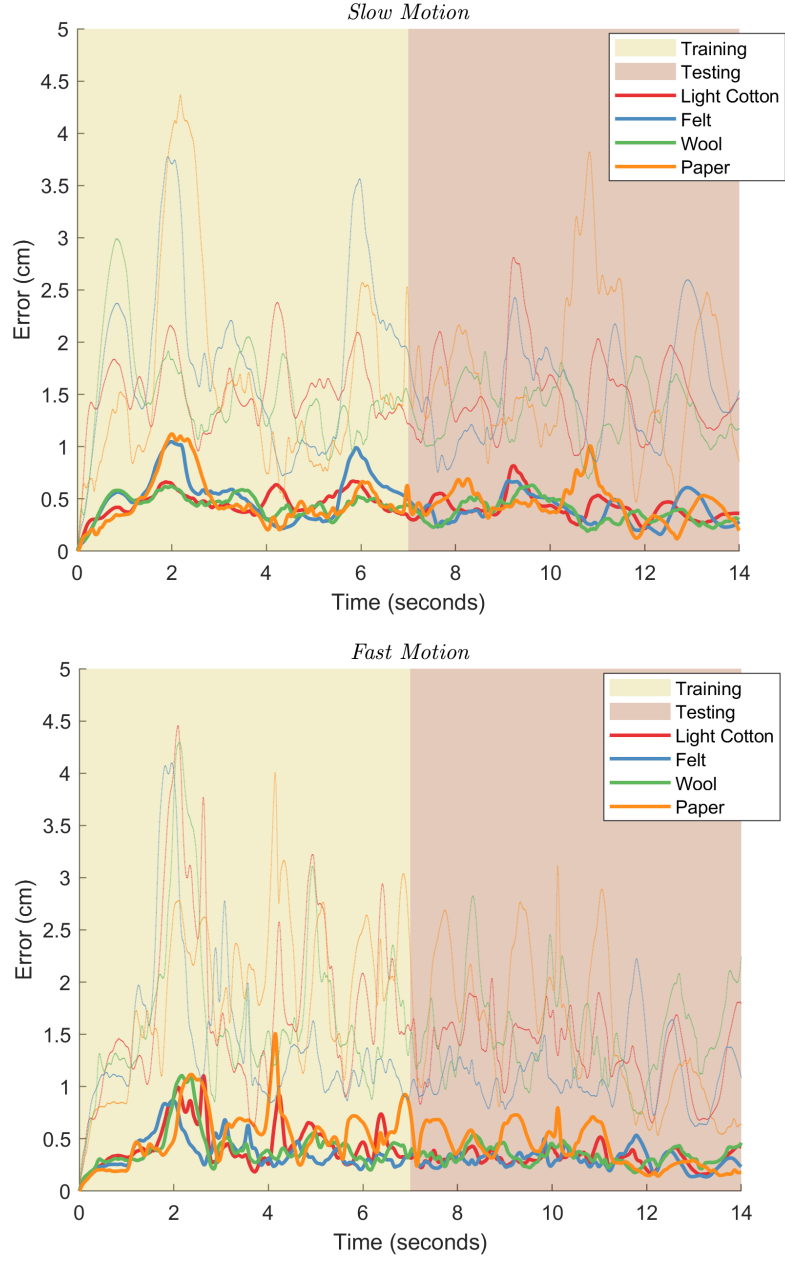


Figure 19: Absolute error (bottom curves) and dispersion (top curves) of the position of the simulated textile vs. the real motion for the slow movement (top) and the fast one (bottom) using the inextensible model.

In the application of our model for control of robotic manipulation purposes, it may be necessary to simulate the motion of the garment prior to any possibility of calibration of its two relevant parameters. In such case, estimates of these parameters should be made from a prior study of the model and the involved fabric. This topic will be the subject of future work, but we have found that a simple, interim solution is already accurate to within our desired error bounds: as a prediction of motion regardless of cloth speed one may take as estimate of each parameter the arithmetic average of the values of its calibrations found in Table 3 for a comparable fabric. Performing the simulations of both the slow and fast motions for light cotton using these a priori estimates instead of calibration yielded 0.51cm and 0.59cm respectively as the mean of error (25) (compared to 0.41cm and 0.31cm with the optimal parameters). This is in agreement with the stability of the model with respect to its parameters shown in the sensitivity analysis previously performed.

10. Conclusions and Further work

In this work we have presented a continuous model for the simulation of inextensible textiles. Continuity is important because it guarantees that the model is very stable under different resolutions, allowing coarse meshes to be used (e.g. in robotic applications) soundly. On the other hand, inextensibility as implemented here permits the conservation of area within an error of less than 1% without exhibiting *locking*. We have also illustrated a framework for the empirical validation of this model. The results are accurate: even with a coarse 9×9 mesh the mean error stays under 5mm for different textiles and both fast and slow motions. Besides, the calibration depends only on two physical parameters. In a novel, and remarkably simple way, those two parameters take into account even the aerodynamic perturbations to motion, which for cloth are harder to model than for rigid, even vibrating, bodies.

These properties make our model ideal for planning and control of cloth manipulation by robots. The authors intend to experiment further with varied types of cloth, fabrics, and motion, to learn about the meaning of the model parameters α, δ and their relation to known factors that govern cloth behavior (such as the KES parameters) and aerodynamics (permeability, velocity of motion, geometry). This study should allow the use of the model as a simulator replacing the calibration of the parameters with an a priori forecast of their value at a very modest cost in terms of accuracy. We have already shown that with a crude forecast of their value our model already matches real motion within a 1cm error margin.

Currently we are working on the modeling of friction and self-collisions specifically tailored to the simulation of inextensible cloth. Moreover, another direction to work on is the empirical validation of our model with more complex garments, such as a shirt, and movements (e.g. rotations). This irremediably involves occlusions. When parts of the cloth are not visible some of the time, we need to apply an identification scheme to our laboratory recordings to identify with accuracy these regions when they appear once again in the limelight. This reidentification scheme, which must have an error smaller than that of our model (i.e. in the mm range), is still under development.

11. Acknowledgements

This project was developed in the context of the project CLOTHILDE ("CLOTH manIpulation Learning from DEmonstrations") which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 741930) and is also supported by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656).

References

- [1] J. Hearle, P. Grosberg, S. Backer, Structural mechanics of fibers, yarns, and fabrics, Vol. 1, Wiley-Interscience, 1969.
- [2] T. Mao, S. Xia, Z. Wang, Evaluating simplified air force models for cloth simulation, in: 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, 2009, pp. 81–86. doi:10.1109/CADCG.2009.5246929.
- [3] M. Shelley, J. Zhang, Flapping and bending bodies interacting with fluid flows, Annual Review of Fluid Mechanics 43 (2011) 449–465. doi:10.1146/annurev-fluid-121108-145456.
- [4] J. Borràs, G. Alenyà, C. Torras, A grasping-centered analysis for cloth manipulation, IEEE Transactions on Robotics 36 (3) (2020) 924–936. doi:10.1109/TR0.2020.2986921.
- [5] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, E. Grinspun, Efficient simulation of inextensible cloth, Vol. 26, ACM, New York, NY, USA, 2007.
URL <http://doi.acm.org/10.1145/1276377.1276438>
- [6] H. Gluck, Almost all simply connected closed surfaces are rigid, in: L. C. Glaser, T. B. Rushing (Eds.), Geometric Topology, Springer Berlin Heidelberg, Berlin, Heidelberg, 1975, pp. 225–239.
- [7] E. English, R. Bridson, Animating developable surfaces using nonconforming elements, ACM TOG 27 (3) (2008) 66:1–66:5.
URL <http://doi.acm.org/10.1145/1360612.1360665>
- [8] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, M. Macklin, A survey on position-based simulation methods in computer graphics, Computer Graphics Forum 33 (6) (2014) 228–251.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12346>
- [9] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson, Physically based deformable models in computer graphics, Computer Graphics Forum 25 (4) (2006) 809–836.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2006.01000.x>
- [10] I. Babuška, M. Suri, Locking effects in the finite element approximation of elasticity problems, Numerische Mathematik 62 (1) (1992) 439–463. doi:10.1007/BF01396238.
URL <https://doi.org/10.1007/BF01396238>
- [11] P. J. Phillips, M. F. Wheeler, Overcoming the problem of locking in linear elasticity and poroelasticity: an heuristic approach, Computational Geosciences 13 (1) (2009) 5–12. doi:10.1007/s10596-008-9114-x.
URL <https://doi.org/10.1007/s10596-008-9114-x>
- [12] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, Elastically deformable models, SIGGRAPH Comput. Graph. 21 (4) (1987) 205–214.
URL <http://doi.acm.org/10.1145/37402.37427>
- [13] K.-J. Choi, H.-S. Ko, Research problems in clothing simulation, Comput. Aided Des. 37 (6) (2005) 585–592.
URL <http://dx.doi.org/10.1016/j.cad.2004.11.002>
- [14] J. Hu, Structure and mechanics of woven fabrics., The Textile Institute and Woodhead Publishing Ltd (UK), 2004.
- [15] Z. Wu, C. Au, M. Yuen, Mechanical properties of fabric materials for draping simulation, International Journal of Clothing Science and Technology 15 (2003) 56–68. doi:10.1108/09556220310461169.
- [16] D. W. Lloyd, The analysis of complex deformations, Mechanics of Flexible Fiber Assemblies (1980) 311–342.
URL <https://ci.nii.ac.jp/naid/10003779259/en/>
- [17] S. Kawabata, The development of the objective measurement of fabric handle, Eurographics Association, 1982.

- [18] G. Cirio, J. Lopez-Moreno, D. Miraut, M. A. Otaduy, Yarn-level simulation of woven cloth, *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH Asia)* 33 (6).
URL <http://www.gmr.v.es/Publications/2014/CLM014>
- [19] C. Jiang, T. Gast, J. Teran, Anisotropic elastoplasticity for cloth, knit and hair frictional contact, Vol. 36, ACM, New York, NY, USA, 2017, pp. 152:1–152:14.
URL <http://doi.acm.org/10.1145/3072959.3073623>
- [20] S. A. Paipetis, Mathematical modelling of composites, *Developments in Composite Material-2-stress Analysis Holister* (1981) 1–29.
- [21] O. Eitzmuss, M. Keckeisen, W. Strasser, A fast finite element solution for cloth modelling, in: 11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings., 2003, pp. 244–251. doi:10.1109/PCCGA.2003.1238266.
- [22] J. Rodriguez-Navarro, A. Susin, Non structured meshes for Cloth GPU simulation using FEM, in: C. Mendoza, I. Navazo (Eds.), *Vriphys: 3rd Workshop in Virtual Reality, Interactions, and Physical Simulation*, The Eurographics Association, 2006. doi:10.2312/PE/vriphys/vriphys06/001-007.
- [23] D. Baraff, A. Witkin, Large steps in cloth simulation, in: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, ACM, New York, NY, USA, 1998, pp. 43–54.
URL <http://doi.acm.org/10.1145/280814.280821>
- [24] D. Braess, *Finite elements. Theory, fast solvers and applications in elasticity theory*, Berlin: Springer., 2007.
- [25] N. Jin, W. Lu, Z. Geng, R. P. Fedkiw, Inequality cloth, in: *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '17*, ACM, New York, NY, USA, 2017, pp. 16:1–16:10.
URL <http://doi.acm.org/10.1145/3099564.3099568>
- [26] X. Provot, Deformation constraints in a mass-spring model to describe rigid cloth behavior, *Graphics Interface* 23(19).
- [27] K.-J. Choi, H.-S. Ko, Stable but responsive cloth, *ACM Trans. Graph.* 21 (3) (2002) 604–611.
URL <http://doi.acm.org/10.1145/566654.566624>
- [28] D. Mongus, B. Repnik, M. Mernik, B. Alik, A hybrid evolutionary algorithm for tuning a cloth-simulation model, *Appl. Soft Comput.* 12 (1) (2012) 266–273.
URL <http://dx.doi.org/10.1016/j.asoc.2011.08.047>
- [29] B. Thomaszewski, S. Pabst, W. Strasser, Continuum-based strain limiting, *Computer Graphics Forum* 28 (2) (2009) 569–576.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01397.x>
- [30] R. Narain, A. Samii, J. F. O'Brien, Adaptive anisotropic remeshing for cloth simulation, *ACM Trans. Graph.* 31 (6) (2012) 152:1–152:10.
URL <http://doi.acm.org/10.1145/2366145.2366171>
- [31] G. Ma, J. Ye, J. Li, X. Zhang, Anisotropic strain limiting for quadrilateral and triangular cloth meshes, *Computer Graphics Forum* 35.
- [32] Z. Wang, T. Wang, M. Tang, R. Tong, Efficient and robust strain limiting and treatment of simultaneous collisions with semidefinite programming, *Computational Visual Media* 2 (2) (2016) 119–130. doi:10.1007/s41095-016-0042-8.
URL <https://doi.org/10.1007/s41095-016-0042-8>
- [33] D. Han, Area Preserving Strain Limiting, in: F. Jalliet, F. Zara, G. Zachmann (Eds.), *Workshop on Virtual Reality Interaction and Physical Simulation*, The Eurographics Association, 2015. doi:10.2312/vriphys.20151340.
- [34] P. F. Antonietti, P. Biscari, A. Tavakoli, M. Verani, M. Vianello, Theoretical study and numerical simulation of textiles, *Applied Mathematical Modelling* 35 (6) (2011) 2669–2681. doi:<https://doi.org/10.1016/j.apm.2010.11.062>.
URL <https://www.sciencedirect.com/science/article/pii/S0307904X10004725>
- [35] H. Wang, R. Ramamoorthi, J. F. O'Brien, Data-driven elastic models for cloth: Modeling and measurement, *ACM Transactions on Graphics* 30 (4) (2011) 71:1–11, proceedings of ACM SIGGRAPH 2011, Vancouver, BC Canada.
URL <http://graphics.berkeley.edu/papers/Wang-DDE-2011-08/>
- [36] E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. Otaduy, S. Marschner, Data-driven estimation of cloth simulation models, *Computer Graphics Forum* 31 (2012) 519–528. doi:10.1111/j.1467-8659.2012.03031.x.
- [37] D. Clyde, J. Teran, R. Tamstorf, Modeling and data-driven parameter estimation for woven fabrics, in: *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*,

- SCA '17, Association for Computing Machinery, New York, NY, USA, 2017. doi:10.1145/3099564.3099577.
URL <https://doi.org/10.1145/3099564.3099577>
- [38] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976.
 - [39] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, E. Grinspun, A quadratic bending model for inextensible surfaces, in: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, Eurographics Association, 2006, pp. 227–230.
URL <http://dl.acm.org/citation.cfm?id=1281957.1281987>
 - [40] J. R. Taylor, *Classical Mechanics*, University Science Books, 2005.
 - [41] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, Sixth Edition, Butterworth-Heinemann, 2005.
 - [42] M. Ghalandari, S. Bornassi, S. Shamshirband, A. Mosavi, K. W. Chau, Investigation of submerged structures' flexibility on sloshing frequency using a boundary element method and finite element analysis, *Engineering Applications of Computational Fluid Mechanics* 13 (1) (2019) 519–528. arXiv: <https://doi.org/10.1080/19942060.2019.1619197>, doi:10.1080/19942060.2019.1619197.
URL <https://doi.org/10.1080/19942060.2019.1619197>
 - [43] X. Liu, L. Liu, An immersed transitional interface finite element method for fluid interacting with rigid/deformable solid, *Engineering Applications of Computational Fluid Mechanics* 13 (1) (2019) 337–358. arXiv: <https://doi.org/10.1080/19942060.2019.1586774>, doi:10.1080/19942060.2019.1586774.
URL <https://doi.org/10.1080/19942060.2019.1586774>
 - [44] M. Kunze, *Non-Smooth Dynamical Systems*, Vol. 1744, Springer, 2000. doi:10.1007/BFb0103852.
 - [45] Z. Zhong, *Finite Element Procedures for Contact-Impact Problems*, Oxford U.P., 1993.
 - [46] A. Gider, *An Online Fabric Database to Link Fabric Drape and End-Use Properties*, LSU Master's Thesis. 3902, 2004.
URL https://digitalcommons.lsu.edu/gradschool_theses/3902
 - [47] L. Gan, N. Ly, G. Steven, A study of fabric deformation using nonlinear finite elements, *Textile Research Journal* 65 (11) (1995) 660–668. arXiv: <https://doi.org/10.1177/004051759506501106>, doi:10.1177/004051759506501106.
URL <https://doi.org/10.1177/004051759506501106>
 - [48] K. Hormann, G. Greiner, Quadrilateral remeshing, in: *Proceedings of Vision, Modeling and Visualization*, 2000, pp. 153–162.
 - [49] R. Aranyó Llach, *Validación del modelo de tela del proyecto Clothlide mediante la simulación y el cálculo numérico*, Universitat Politècnica de Catalunya, Degree Thesis, 2020.
URL <http://hdl.handle.net/2117/330056>
 - [50] Blender, <https://www.blender.org/>.
 - [51] Mujoco (multi-joint dynamics with contact), <http://www.mujoco.org/index.html>.