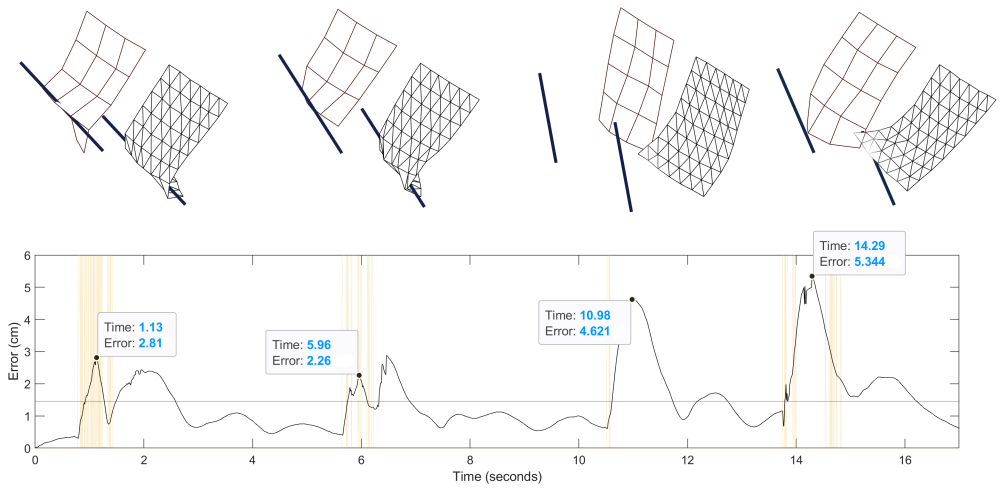
Graphical Abstract

A novel collision model for inextensible textiles and its experimental validation

Franco Coltraro, Jaume Amorós, Maria Alberich-Carramiñana, Carme Torras



In this work, we present a novel collision model for inextensible textiles and validate it experimentally through a comparison with real textiles as recorded by a *Motion Capture System*. In the figure we can see four frames comparing the recorded hitting with a stick of a DIN A2 (42 x 59.4 cm) polyester sheet (left) with its inextensible simulation (right); its average error being 1.44 cm. The recordings are obtained by attaching 20 reflective markers of diameter 3 mm and weight of 0.013 g to the real textile and following their trajectory. On the bottom, we show a full plot of the mean absolute error of the position of each marker with respect to its simulation and with yellow lines we highlight the moments in which the object is in contact with the cloth. Notice that the biggest errors appear not during the hits but just after because of aerodynamic effects. The above simulations, with a 7×9 mesh, are two times faster than real-time. Our custom active-set solver is three times faster than a standard interior-point method using the same mesh resolution.

A novel collision model for inextensible textiles and its experimental validation

Franco Coltraro[a,b,*], Jaume Amorós[b], Maria Alberich-Carramiñana[a,b], Carme Torras[a]

[a] *Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain.*
[b] *Universitat Politècnica de Catalunya, Barcelona, Spain.*

Abstract

In this work, we introduce a collision model specifically tailored for the simulation of inextensible textiles. The model considers friction, contacts, and inextensibility constraints all at the same time without any decoupling. Self-collisions are modeled in a natural way that allows considering the thickness of cloth without introducing unwanted oscillations. The discretization of the equations of motion leads naturally to a sequence of quadratic problems with inequality and equality constraints. In order to solve these problems efficiently, we develop a novel active-set algorithm that takes into account past active constraints to accelerate the resolution of unresolved contacts. We put to a test the developed collision procedure with diverse scenarios involving static and dynamic friction, sharp objects, and complex-topology folding sequences. Finally, we perform an experimental validation of the collision model by comparing simulations with recordings of real textiles as given by a *Motion Capture System*. The results are very accurate, having errors around 1 cm for DIN A2 textiles (42 x 59.4 cm) even in difficult scenarios involving fast and strong hits with a rigid object.

Keywords: inextensible cloth, collision modeling, Coloumb friction, experimental validation, constrained optimization
2010 MSC: 70F35, 68U20, 90C20

1. Introduction and related work

Textile objects are important and omnipresent in many relevant scenarios of everyday life, e.g. domestic, healthcare, or industrial contexts. However, as opposed to rigid objects, whose position in space is determined by position and orientation (i.e. six degrees of freedom), textile objects are challenging to handle in an automated fashion (e.g. by robots) because they change shape under contact and motion, resulting in an infinite-dimensional configuration space (when considered as continuous surfaces in 3D space). This huge dimensional jump makes existing perception and manipulation methods difficult to apply to textiles. Recent reviews on cloth manipulation, like [1, 2], agree on the need to find simple, but realistic, models that enable more powerful learning methods to solve different problems related to cloth manipulation.

*Corresponding author
Email address: franco.coltraro@upc.edu (Franco Coltraro)

In [3] a physical model of cloth intended for its control under robotic manipulation in a human environment –meaning that the textiles are subjected to moderate to low stresses [4]– was introduced and discussed. Because of its control purpose, a model was developed with which simulations of the motion of cloth could be computed fast, but with a small margin of error 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. This led to the most basic hypothesis of the cloth model, i.e. to 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, which simplifies the model by removing all considerations of elasticity in it, was shown to be very realistic for several materials and motions in [3]. Furthermore, since inextensibility is modeled with continuous constraints, the model is stable under different mesh resolutions, allowing coarse meshes to be used (e.g. in robotic applications) soundly.

Once we have a working inextensible cloth simulator that describes the internal dynamics of cloth; a related problem that always arises with inextensibility simulations is how to conciliate them with collision and friction forces (i.e. contacts with an obstacle or self-collisions), since both inextensibility and contact forces are hard constraints that the cloth must satisfy at the same time (moreover, friction forces depend on the magnitude of the contact force). In the conclusions of [5], it is stated:

"(...) there is no longer an efficient way to perfectly enforce both ideal inextensibility and ideal collision handling, since one filter must execute before the other, and both ideals correspond to sharp constraints. To enforce both perfectly would require combining them in a single pass, an elegant and exciting prospect from the standpoint of theory, but one which is likely to introduce considerable complexity and convergence challenges."

To our knowledge, the vast majority of models in literature decouple friction, contacts, and strain limiting (i.e. inextensibility), with all the possible artifacts that this introduces. In this work, we develop a novel collision model, its discretization and an active set-solver that can be seen as an extension of the *fast projection algorithm* of [5], in order to incorporate contacts, friction and inextensibility in a single pass.

1.1. Contributions

In the following, we list the most important contributions of this article:

- We present a novel collision model for cloth which results in a non-decoupled resolution of friction, strain limiting and contact constraints in a single pass.
- Self-collisions are modeled and detected in a way that allows considering the thickness of cloth without introducing unwanted oscillations.
- An efficient active-set solver for the integration of the system is developed, which –as opposed to traditional methods– can start from any non-necessarily feasible point.
- We present an empirical validation of the model, achieving errors of around 1 cm in challenging scenarios involving low and high-friction surfaces and high-velocity collisions with a rigid object.

1.2. Related Work

There is a rich history of research on contact and collisions for cloth simulation; in the following, we will review what we consider the most relevant methods, with an emphasis on newer works. We focus especially on articles that model collisions in physically different manners or use novel numerical algorithms to resolve them (and not so much on performance, e.g. GPU implementations of existing methods). Most of these works come from the Computer Graphics (CG) community and not so much from the Textile Engineering fields. This is due to the fact that in CG applications a core concern has always been simulating dressed moving mannequins (e.g. for movies and video games), whereas for textile engineers the focus has been on measuring intrinsic fabric properties. Furthermore, it is also relevant to mention the Applied Mechanics community, which has made major contributions to the problem of contact modeling, especially for rigid bodies and collections of them. Since the seminal work of J. J. Moreau (see for example [6] for a description of his now standard *sweeping process*), hundreds of papers have appeared in literature. For detailed references and the main ideas of the field, we refer the reader to the following books [7–9] where a general introduction to non-smooth mechanics and dynamics and some applications can be found.

Now, even though most cloth collision models assume that the garment has been meshed (e.g. triangulated), the constituting elements (i.e. triangles) are not completely rigid in any cloth model: treating the piece of cloth as an assembly of rigid elements has to be avoided, as it leads to poor forecasts on its behavior (i.e. it causes problems such as *locking*, in which the model makes the cloth behave as a single rigid body, see [3] for a discussion and many relevant references). On top of that, inextensible cloth's equations of motion usually lead to a large (more than one hundred variables) system of *stiff* ordinary differential equations (ODEs), which must be integrated implicitly in order to avoid unwanted oscillations and achieve stability. All of these reasons make the direct application to cloth simulation of the algorithms found in the aforementioned books [7–9] very challenging. Nevertheless, throughout this work, some of the ideas of non-smooth mechanics will be discussed and incorporated into our contact model for cloth.

With that said, to our knowledge there are 3 main types of collision-response methods specifically developed for cloth simulation which we describe next. We note that they correspond approximately with the classification given in the introduction of [10].

Penalty-based methods: these include very stiff spring-like forces of the form $kf(\epsilon)$ (where ϵ is the detected penetration depth) into the dynamical system when a penetration is detected. They are easy to implement and can work for simple cases, but are not physically accurate (e.g. they do not conserve momentum during the collision, see [11]) and introduce a lot of stiffness into the system when k is large (making it harder to integrate numerically). For example, Provot [12] is one of the first to propose a penalty-based approach to solve collisions for cloth modeled as a mass-spring system. However, his method has no theoretical guarantees when there is more than one simultaneous collision. That is why, when many collisions accumulate during the same time step, he must resort to a fail-safe consisting of rigidifying zones of the cloth. On the other hand, he is among the first to give a formula to detect continuous-time collisions (i.e. when two moving edges or a node and triangle intersect, see Section 3). Finally, one of the main theoretical problems with penalty-based methods is that there is always a fast or strong enough collision where they fail because the spring force is not strong enough (although there are sophistications with more guarantees such as [13]). Despite their mentioned limitations, penalty methods are still widely used because of their ease of implementation (see, e.g. [14]).

Impulse-based methods: these methods include impulse forces (mostly based on rigid-body mechanical ideas, see [15]) which are then used to modify velocities (and thus positions) instantaneously. They work well for individual collisions and are fast to compute, but run into problems for multiple simultaneous collisions when simulating cloth using time-stepping methods. Impulse-based forces for simultaneous collisions can be included soundly in event-based schemes for collections of rigid bodies, see the Liu-Zhao-Brogliato (LZB) approach in [16]. However, event-based schemes can be very challenging to implement for textiles, because being cloth a very deformable body, collisions are many and moreover accumulate over a time-step, causing these event-based algorithms to be computationally very intensive. In this line, the work by Bridson et al. [17] is considered to be by the CG community the first truly robust method for handling collisions, contact and friction for cloth simulation. More than a unified physical model, their method consists of a list of procedures used to get a state of cloth that is collision-free but not necessarily physically realistic. They first apply penalty forces as a prevention method and then impulse forces for detected collisions in continuous time. When there are simultaneous collisions in the same time step, after a fixed number of iterations where they average impulses over the cloth surface, they also resort to rigid impact zones (but with corrected formulas with respect to [12]). Some alternatives exist to avoid the rigidification of areas of the cloth, e.g. [18] and [19]. These two methods derive impulse forces for the case of simultaneous collisions with the aid of constraints that are being violated by the detected penetrations. Their main problem is that in order to be efficient they derive the impulses from equality constraints, and this introduces *sticking* artifacts into the simulation (some nodes are forced to stay in contact, when they otherwise would depart). Moreover, another issue with impulse-based methods is that since one is modifying positions instantaneously, strain-limiting procedures must be performed prior to (or after) collision response, and thus constraints such as inextensibility cannot be maintained exactly. These type of models have somehow fallen out of fashion in recent times (at least in the research literature for cloth simulation) in favor of constraint-based approaches.

Constraint-based methods: with the increase of computing power in the last decades these methods have flourished from a research viewpoint. Their idea is simple: once a collision is detected, a constraint is defined (which is being violated because of the collision) and an optimization problem should be solved with all the detected constraints. Most methods vary mainly in how the optimization problem is solved and how friction is modeled (going from exact Coloumb models to linearized ones). These restrictions can be imposed as equalities or inequalities. As already said, imposing the constraints as equalities can be very efficient but one runs into *sticking artifacts* (since some constraints can pull from others) and thus it is better to consider them inequalities. This is in turn known as the Signorini-contact model, which can handle robustly simultaneous collisions in time-stepping schemes. Otaduy et al. [20] were among the first to propose a physically sound constrained dynamics formulation for cloth simulation and contact. They employ Signorini's contact model and add to it a linearized Coulomb's friction model. The optimization program is stated formally as a linear complementary problem (LCP) and friction and contacts are afterwards decoupled in order to be solved numerically.

Years later, Li et al. [21] implemented exact Coulomb friction for cloth simulation using adaptive meshes. Their constraint-based solver (released later as an open-source simulator called ARGUS) is costly to run but treats contacts (and friction) simultaneously and implicitly. Recently, Ly et al. [22] proposed an alternative numerical algorithm based on Projective Dynamics that accelerates by an order of magnitude results obtained with ARGUS (see also [23] for a differentiable cloth model based on Projective Dynamics). Their main drawback is that they inherit

the limitations of Projective Dynamics, in particular, the lack of a simple rule to ensure convergence. Moreover, they do not consider strain limiting and collisions simultaneously and hence depict cloth as a very elastic material. On a related note, using ideas from [20, 21], Daviet proposes in [24] an approach for the numerical simulation of hard contacts with nonlinear Coulomb friction for a large class of dynamical objects including cloth. One of the main limitations of all the previous methods is the difficulty of integrating at the same time strain limiting (e.g. inextensibility) with the collision handling algorithm. In this line, Li et al. [25] have developed a method that integrates strain limiting and collisions in a single pass with the extensive use of barrier functions. They also propose a benchmark set of challenging tests. Finally, we mention that, naturally, constraint-based methods also have some drawbacks: most of the time they need dedicated solvers and thus can be cumbersome to implement and since several optimization problems must be solved, they are in general slower than impulse or penalty-based methods.

1.3. Overview

The model we will derive in this work lies in the category of constraint-based methods and hence can handle efficiently simultaneous collisions. We will solve a quadratic problem with inequality constraints and therefore we will be employing the physically accurate model of Signorini. In order to do so, we will develop a novel active-set solver to resolve collisions efficiently. Moreover, we will derive a simple friction model that allows us to integrate all forces and constraints in a simple pass without the need to decouple contact and friction forces like it has been traditionally done for rigid body contacts (see [26]). Finally, our method considers strain limiting (inextensibility) and contact at the same time, unlike most current methods. Our algorithm can be seen as an extension of the *fast projection algorithm* (see [5]) developed in order to incorporate contacts, friction and inextensibility in a single pass.

1.4. Organization

The rest of this paper is organized as follows: in Section 2 we explain how to introduce contacts and friction into the equations of motion in a physically sound manner. Next, in Section 3 we delve into the problem of detecting self-collisions and resolving them, including how to take into account the thickness of the cloth during this process. In Section 4 we explain how to discretize the model in order to integrate it numerically without decoupling contacts, friction and inextensibility constraints, as it is usually done. In Section 5 we present a novel algorithm to solve efficiently the quadratic problems that arise from the previous discretization. Finally in Sections 6 and 7 we evaluate the presented model, including the simulation of challenging scenarios and performing an empirical validation of the collision model, by comparing it with real data.

1.5. Notation

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

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

List of most important symbols

S	Surface in $\mathbb{R}^3$ used to model the cloth.
N	Number of vertices/nodes of the discretized surface.
$p_i(t) = (x_i(t), y_i(t), z_i(t))$	Coordinates in $\mathbb{R}^3$ of the node i of the surface S at time t, $p_i = p_i(0) = (x_i, y_i, z_i)$.
$\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)$	x-coordinates (resp. y and z) of all the nodes of the discretized surface in $\mathbb{R}^N$ at time t.
$\boldsymbol{\varphi} = \boldsymbol{\varphi}(t) \in \mathbb{R}^{3N}$	Position of all nodes together at time t, i.e. $(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t))^\intercal \in \mathbb{R}^{3N}$.
$\mathbf{C}(\boldsymbol{\varphi}) = \mathbf{0}$	Set of n_C inextensibility equality constraints evaluated on $\boldsymbol{\varphi}$.
$\mathbf{H}(\boldsymbol{\varphi}) \geq \mathbf{0}$	Set of n_H collision inequality constraints evaluated on $\boldsymbol{\varphi}$.
$\boldsymbol{\lambda}(t) \in \mathbb{R}^{n_C}$	Lagrange multipliers associated to the inextensibility constraints at time t.
$\boldsymbol{\gamma}(t) \in \mathbb{R}^{n_H}$	Lagrange multipliers associated to the contact constraints at time t.
$\|\cdot\|$	Usual Euclidean norm in $\mathbb{R}^{3N}$.
$\dot{\boldsymbol{\varphi}}(t), \ddot{\boldsymbol{\varphi}}(t) \in \mathbb{R}^{3N}$	Velocity and acceleration of all nodes together at time t.
$\mathbf{F}_\mu(\dot{\boldsymbol{\varphi}}) \in \mathbb{R}^{3N}$	Friction force.
$\mathbf{V}(\dot{\boldsymbol{\varphi}}) \in \mathbb{R}^{n_H \times 3N}$	$(n_H \times 3N)$ matrix of unit relative tangent velocities at the points of contact.
$\Delta t > 0$	Time step used to discretize the equations of motion.
$\boldsymbol{\varphi}^m, \dot{\boldsymbol{\varphi}}^m \in \mathbb{R}^{3N}$	Position and velocity of all nodes at $t_m = m \cdot \Delta t$.
$\mathcal{W}, \mathcal{O}$	Working and observation set of the active-set solver.
$\boldsymbol{\phi}^m$	Nodes of the meshed recording of real textiles at t_m.
e_m	Time-dependent mean absolute error in cm at t_m.
s_m	Time-dependent spatial standard deviation in cm at t_m.

2. Modeling of contacts and friction

Assuming that $S \subset \mathbb{R}^3$ is a smooth inextensible surface moving through space (i.e. we have a family of surfaces $\{S_t\}_{t\geq 0}$ isometric to $S = S_0$) and $\varphi^t : R \to S_t$ is a smooth parametrization of S_t, where R is an open set of the plane and $(\xi, \eta) \in R$, in [3] the following time- and space-dependent constraints were derived

$$\langle \varphi_\xi, \varphi_\xi \rangle(t) = E_0,\ \langle \varphi_\xi, \varphi_\eta \rangle(t) = F_0,\ \langle \varphi_\eta, \varphi_\eta \rangle(t) = G_0 \text{ for } t \geq 0; \tag{1}$$

where $t \geq 0$ represents time, E_0, F_0, G_0 are the coefficients of the first fundamental form or metric of S that are constant in time (but not necessarily in R), and $\varphi_\xi(t) = \partial_\xi \varphi^t$ denotes partial differentiation with respect to the variable ξ.

Remark 2.1 (Discretization)**.** For the remaining of this work we will assume that the cloth S has been discretized into a triangular or quadrilateral mesh and the position of all its N vertices or nodes (denoted by $p_i(t) = (x_i(t), y_i(t), z_i(t)) \in \mathbb{R}^3$) at time $t \geq 0$ is given by $\boldsymbol{\varphi}(t) = (\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t))^\intercal \in \mathbb{R}^{3N}$, where $\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)$ denote the x-coordinates (resp. y and z) of all the nodes of the discrete surface in $\mathbb{R}^N$ at time t. Moreover, later when integrating numerically the equations of motion of the cloth (see Equation (2)), as usual, we will approximate $\boldsymbol{\varphi}(t)$ and $\dot{\boldsymbol{\varphi}}(t)$ by $\{\boldsymbol{\varphi}^0, \boldsymbol{\varphi}^1, \dots\}$ and $\{\dot{\boldsymbol{\varphi}}^0, \dot{\boldsymbol{\varphi}}^1, \dots\}$, where $\boldsymbol{\varphi}^n$ and $\dot{\boldsymbol{\varphi}}^n$ are the position and velocities of all

the nodes of the mesh at time $t_n = n \cdot \Delta t$ and $\Delta t > 0$ is the size of the chosen fixed time step (unless otherwise stated we set $\Delta t = 0.01$).

Then, fixing any discretization of the cloth surface S, it is possible to efficiently discretize in space the system of partial differential equations (1) with the aid of finite elements, such that we obtain a smooth function $\mathbf{C} : \mathbb{R}^{3N} \to \mathbb{R}^{n_C}$ and the system (1) reduces to $\mathbf{C}(\boldsymbol{\varphi}) = \mathbf{0}$. Each constraint $C_i(\boldsymbol{\varphi}(t)) = 0,\ i = 1, \dots, n_C$ is in fact a quadratic function of its argument and we have n_C of them, depending on the number of nodes N of the discretization. Indeed, for each interior (non-boundary) node index $k \in \mathrm{Int}(S)$ of the discretized surface S, each of the three equations of system (1) gives rise to a constraint given by $E_k(t) - E_k(0) = 0$ or $F_k(t) - F_k(0) = 0$ or $G_k(t) - G_k(0) = 0$, where

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

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

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

Here $m_k > 0$ is one third the sum of the areas of all incident triangles (one fourth in the case of quadrilaterals) to the kth node p_k of the surface and $\mathcal{N}_i : S \to \mathbb{R}$ are piece-wise smooth continuous indicator functions (also called a Lagrange basis or hat functions) such that $\mathcal{N}_i(p_j) = \delta_{ij}$ is the Kronecker delta. Finally, the inextensibility of the boundary curves of S is maintained by preserving the length of each edge lying on the boundary. See [3] for more details and an efficient algorithm to evaluate these constraints.

The inextensibility constraints $\mathbf{C}(\boldsymbol{\varphi}) = \mathbf{0}$ model what is usually called the *internal dynamics* of cloth; furthermore, for its application in the real world, we also need to include in any cloth model collisions of the cloth with an object (e.g. a table) and with itself (see Section 3). We will model this by enforcing a set $H_i(\boldsymbol{\varphi}(t)) \geq 0$, $i = 1, \dots, n_H$ of n_H inequality constraints $\mathbf{H}(\boldsymbol{\varphi}) \geq \mathbf{0}$, such that the differentiable function $\mathbf{H} : \mathbb{R}^{3N} \to \mathbb{R}^{n_H}$ satisfies that for each $i = 1, \dots, n_H$ we have a non-zero outwards normal $\nabla H_i(\boldsymbol{\varphi}) \neq \mathbf{0}$. We can then model collisions by including non-smooth (see [8] and [27]) forces into the equations of motion of the cloth (this is known as Signorini's contact model, see [28]):

$$\begin{cases} \mathbf{M}\ddot{\boldsymbol{\varphi}} = \mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) - \nabla \mathbf{C}(\boldsymbol{\varphi})^\intercal \boldsymbol{\lambda} + \nabla \mathbf{H}(\boldsymbol{\varphi})^\intercal \boldsymbol{\gamma}, \\ \mathbf{C}(\boldsymbol{\varphi}) = \mathbf{0}, \\ \mathbf{H}(\boldsymbol{\varphi}) \geq \mathbf{0}, \quad \boldsymbol{\gamma} \geq \mathbf{0}, \quad \boldsymbol{\gamma}^\intercal \cdot \mathbf{H}(\boldsymbol{\varphi}) = 0, \end{cases} \tag{2}$$

where $\mathbf{M} \in \mathbb{R}^{3N \times 3N}$ is the mass matrix obtained from the discretization with finite elements of S, $\mathbf{C}(\boldsymbol{\varphi}) = \mathbf{0}$ are as already explained the discretization of the inextensibility constraints (1) and $\boldsymbol{\lambda} = \boldsymbol{\lambda}(t) \in \mathbb{R}^{n_C}$ are its associated Lagrange multipliers, $\boldsymbol{\gamma} = \boldsymbol{\gamma}(t) \in \mathbb{R}^{n_H}$ are n_H non-negative contact Lagrange multipliers and we have grouped in the force term $\mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) \in \mathbb{R}^{3N}$ damping, bending, gravity and aerodynamic forces. Now the system is non-smooth, which is why we will need to use a first-order (implicit) integration scheme [29]. A simple model for friction can be introduced if we add to the first equation of (2) yet another force of the form:

$$\mathbf{F}_\mu(\dot{\boldsymbol{\varphi}}) = -\mu \mathbf{V}(\dot{\boldsymbol{\varphi}})^\intercal \boldsymbol{\beta} \tag{3}$$

where $\mu > 0$ is a friction constant, $\boldsymbol{\beta} = \boldsymbol{\beta}(t) \in \mathbb{R}^{n_H}$ are new multipliers (one for each contact constraint) satisfying that they belong to the friction's cone, i.e. they satisfy that for each $i = 1, \ldots, n_H$ we have $0 \leq \beta_i \leq \|\gamma_i \nabla H_i(\boldsymbol{\varphi})^\intercal\|$ (unless otherwise denoted $\|\cdot\|$ represents the usual Euclidean norm in $\mathbb{R}^{3N}$), and $\mathbf{V}(\dot{\boldsymbol{\varphi}})$ is a $(n_H \times 3N)$ matrix consisting of unit (relative) tangent velocities at the points of contact, i.e. for the case of a collision with a static obstacle, each row of $\mathbf{V}(\dot{\boldsymbol{\varphi}}) \in \mathbb{R}^{n_H \times 3N}$ would be:

$$kV_i(\dot{\boldsymbol{\varphi}})^\intercal = \dot{\boldsymbol{\varphi}} - \langle \dot{\boldsymbol{\varphi}}, \mathbf{n}_i \rangle \cdot \mathbf{n}_i,$$

where $\mathbf{n}_i = \frac{\nabla H_i(\boldsymbol{\varphi})}{\|\nabla H_i(\boldsymbol{\varphi})\|}$ and k is a normalization constant. For theoretical details and more sophisticated models for friction, see [29].

Remark 2.2. We now list some assumptions we are making in stating the collision model as in Equation (2):

1. When $h(x, y, z) = 0$ defines an implicit surface in $\mathbb{R}^3$ (e.g. a plane $z = 0$ or a sphere $x^2 + y^2 + z^2 = 1$), the conditions $\mathbf{H}(\boldsymbol{\varphi}) \geq \mathbf{0}$ mean that for each node p_i of the cloth's mesh we impose

   $$H_i(\boldsymbol{\varphi}) := h(p_i(t)) \geq 0.$$

   This only forces the vertices of the cloth to be outside the obstacle (but there could be some penetrations of the faces). When the mesh is fine enough this is not really a problem, in the case of coarse meshes one can add yet another constraint for the middle point of each face of the triangulation.

2. Signorini's condition implies that when there is no contact taking place, i.e. $H_i(\boldsymbol{\varphi}) > 0$, then there is no repulsive force acting, i.e. $\gamma_i \nabla H_i(\boldsymbol{\varphi})^\intercal = \mathbf{0}$. Therefore there is also no friction force acting, i.e. $\beta_i = 0$.

3. Without any other condition the multipliers $\boldsymbol{\beta}$ are not uniquely defined. A common approach is to assume that these multipliers cause *maximal dissipation* (see [11, 26, 28]). This amounts to solving a (constrained) linear optimization problem. In practice, we will assume that $\beta_i = \|\gamma_i \nabla H_i(\boldsymbol{\varphi})^\intercal\|$ (which is anyways always the case when the tangent velocity is nonzero: this is called sliding friction in literature, see [30]).

4. This model assumes that the collision is inelastic (there is no bouncing). This is a reasonable assumption for cloth; we will corroborate this in Section 7 when we perform the empirical validation of the collision model.

3. Modeling and detection of self-collisions

In this section, we explain how to detect and include self-collision constraints under the framework presented in Section 2. Particularly important for efficiency and to avoid unwanted oscillations is how to take into account the thickness of cloth.

The goal is to define the constraints H_k that account for modeling self-collisions of the cloth inside the set of inequality constraints $\mathbf{H}(\boldsymbol{\varphi}) \geq \mathbf{0}$. To do this, in principle we need to integrate numerically the equations of motion (2) and advance the simulation from $\boldsymbol{\varphi}^n$ to $\boldsymbol{\varphi}^{n+1} = \boldsymbol{\varphi}^n +$

$\Delta t \cdot \dot{\boldsymbol{\varphi}}^{n+1}$, then check if in the process self-collisions took place, and in case they did, add new constraints H_k to the system and repeat the numerical integration. Notice that the detection of self-collisions and its response will be done between the discrete time steps t_n and t_{n+1} and assuming that the nodes move at constant velocity. This process must be repeated until no new collisions are found. In practice this is costly and thus we will develop a more efficient procedure that takes advantage of the way we integrate numerically the equations of motion (see Section 4.1). For the time being, assume we have both $\boldsymbol{\varphi}^n$ and $\boldsymbol{\varphi}^{n+1}$ (and their velocities) available to make computations.

3.1. Detection of self-collisions

In general, we assume that the cloth is triangulated (in case of a quadrangulation we can always divide the quads in two triangles); then in case of collision of two faces, there are only two stable (i.e. detectable) possibilities: an edge-edge collision and a node-face collision (see e.g.[12]). In both cases, we have four nodes involved which at some instant of time belong to the same plane. We must then only check if two co-planar segments intersect or if a point lies inside a triangle. These two problems are readily solved using barycentric coordinates.

Now we describe in more detail the process: in order to save computational time, we only check if a collision has happened for pairs of edges (or nodes and faces) that at time t_n or t_{n+1} are *sufficiently* close. To obtain this list of sufficiently close (up to some tolerance) pairs, there are several methods: one of the most widely used are *hierarchical methods* [12], where the mesh is divided into large regions, which are in turn also divided in smaller regions, up to a certain number of times (the number of hierarchies). Then one only checks if two pairs are sufficiently close when all of the larger regions containing these are close enough (e.g. by computing the distance between their center of masses), otherwise, they are discarded. Since our meshes are fairly coarse we compare directly the center of masses of our edge-edge or node-face pairs in order to detect those that are candidates for collision. Next, denoting by $p_1, p_2, p_3, p_4 \in \mathbb{R}^3$ the position of the four *candidate* nodes of the cloth at time t_n and by v_1, v_2, v_3, v_4 their velocities at t_{n+1} (we are using that $\boldsymbol{\varphi}^{n+1} = \boldsymbol{\varphi}^n + \Delta t \cdot \dot{\boldsymbol{\varphi}}^{n+1}$), we must check if for some $t \in [0, \Delta t]$ the four nodes are co-planar assuming they move with constant velocity. This is equivalent to checking if the three vectors $\tilde{p}_i = p_i - p_4$, moving linearly with velocities $\tilde{v}_i = v_i - v_4$ are linearly dependent for some $t \in [0, \Delta t]$. Hence:

$$\det(\tilde{p}_1 + t \cdot \tilde{v}_1, \tilde{p}_2 + t \cdot \tilde{v}_2, \tilde{p}_3 + t \cdot \tilde{v}_3) = 0 \tag{4}$$

This is a cubic equation $a_3 t^3 + a_2 t^2 + a_1 t + a_0 = 0$ whose coefficients are found expanding the determinant, i.e.

$$\begin{cases} a_3 = \det(\tilde{v}_1, \tilde{v}_2, \tilde{v}_3), \\ a_2 = \det(\tilde{p}_1, \tilde{v}_2, \tilde{v}_3) + \det(\tilde{v}_1, \tilde{p}_2, \tilde{v}_3) + \det(\tilde{v}_1, \tilde{v}_2, \tilde{p}_3), \\ a_1 = \det(\tilde{p}_1, \tilde{p}_2, \tilde{v}_3) + \det(\tilde{p}_1, \tilde{v}_2, \tilde{p}_3) + \det(\tilde{v}_1, \tilde{p}_2, \tilde{p}_3), \\ a_0 = \det(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3). \end{cases}$$

When $t \ll \Delta t$ is small, the previous cubic polynomial can be approximated by the linear equation $a_1 t + a_0 = 0$, with solution $-\frac{a_0}{a_1}$. In any case, after finding numerically the zeros of the polynomial, for every real root $t_c \in [0, \Delta t]$ found, we must then do two different calculations with the four co-planar points $q_i = p_i + t_c \cdot v_i$ in order to see if a collision has occurred. Namely: in

the edge-edge case we check if the two co-planar segments intersect and in the node-face case check if the node is inside the (triangular) face. In both cases we follow standard procedures from Computational Geometry (see Ch. 7 of [31]).

3.2. Constraint definition for self-collisions

Once we have detected a self-collision, we now describe the computation of the response constraint H_k. It will be linear in $\boldsymbol{\varphi}$ and naturally have slightly different forms depending on our two cases:

1. Edge-edge case:
$$H_k(\boldsymbol{\varphi}) := \langle \pi_\alpha(p_1, p_2) - \pi_\beta(p_3, p_4), \nu \rangle \geq 0,$$
where p_i are the four endpoints of the two edges, $\pi_\alpha(p_1, p_2) = (1-\alpha)p_1 + \alpha p_2$ and $\pi_\beta(p_3, p_4) = (1-\beta)p_3 + \beta p_4$ are the closest points between the two segments and ν is the normal vector to both edges. In general, the values ν, α, β vary with time. We will nevertheless assume that they are constant during the time-step, and compute them with the positions of the segments given at the collision time t_c, that is using the four endpoints $q_i = p_i + t_c \cdot v_i$. The normal vector ν is oriented such that when we evaluate $H_k(\cdot)$ at $\boldsymbol{\varphi}^n$ with the computed fixed values ν, α, β the constraint is not violated, that is $H_k(\boldsymbol{\varphi}^n) \geq 0$.

2. Node-face case:
$$H_k(\boldsymbol{\varphi}) := \langle p_4 - \pi(p_1, p_2, p_3), \nu \rangle \geq 0,$$
where p_4 is the node, p_i are the three corners of the triangle, again $\pi(p_1, p_2, p_3) = up_1 + vp_2 + wp_3$ is the closest point inside the face to the node and ν is the normal vector to the triangle. In general, the values ν, u, v, w vary with time. We will again assume that they are constant in time, and compute them with the positions given at the collision time. The normal vector ν is again oriented such that $H_k(\boldsymbol{\varphi}^n) \geq 0$.

Remark 3.1. Notice that:

1. By construction $H_k(\boldsymbol{\varphi}^{n+1}) < 0$. Indeed, consider, e.g. for the edge-edge case, the function $h(t) = \langle c_1(t) - c_2(t), \nu \rangle$, where $c_1(t) = (1-\alpha)p_1(t) + \alpha p_2(t)$, $c_2(t) = (1-\beta)p_3(t) + \beta p_4(t)$, $p_i(t) = p_i + tv_i$ and α, β, ν are fixed to its computed values at t_c as explained. Then, in the non-degenerate case $h(0) > 0$, we have that $h(t_c) = 0$ (since the 4 points are co-planar) and $\dot{h}(t)$ is constant and non-zero, hence $\dot{h}(t) < 0$ and therefore $H_k(\boldsymbol{\varphi}^{n+1}) = h(\Delta t) < 0$.

2. The constraint H_k is an approximation of the signed distance between the pairs edge-edge and node-face since ν and the barycentric coefficients are fixed, taking their value at the collision time $t_c \in [0, \Delta t]$.

3. Since in practice cloth has thickness, say τ_0, the constraint we actually must impose is $H_k(\boldsymbol{\varphi}) \geq \tau_0$.

3.3. Proximity constraints and cloth thickness

Adding the constraints we have just defined is enough to correct all present self-intersections. Nevertheless, there are two main drawbacks:

1. Efficiency: most cloth self-intersections can be avoided before they happen by adding preventive constraints.

2. Vibrations: since we are assuming that the cloth has a thickness $\tau_0 > 0$, when we integrate the system again and go from $H_k(\boldsymbol{\varphi}^n) < 0$ to $H_k(\boldsymbol{\varphi}^{n+1}) \geq \tau_0$, the change between the position of the nodes can be too large, and since our cloth is inextensible, this could create unwanted oscillations.

In order to avoid these two problems, we apply the detection procedure previously explained in 3.1 with one small difference: during the detection phase we move the pairs (edge-edge or face-node) closer, using their normal vectors and taking into account the thickness of the cloth, so that pairs that are too close and/or are approaching each other, are kept at a minimum distance of τ_0 before they actually intersect. Since the restrictions we are considering are inequalities, we can add these to the system because they only affect the dynamics of cloth in case the constraint will actually get violated. In symbols, this means that we compute Equation (4) of the third degree polynomial using the altered positions given by $\hat{p}_i = p_i \pm \omega\tau_0 \nu$ (instead of p_i), where ν is the unit normal vector (the cross product for the edge-edge case and the normal to the triangle for node-face case), $\omega \approx 0.5$ is what we will call a *proximity parameter* and the sign $\pm$ is chosen so that the pairs approach each other. Afterwards the normals and the barycentric coordinates for the response constraint H_k are also calculated with the altered positions $\hat{p}_i$.

Although there are algorithms that estimate time-to-contact based on trajectories, update these estimates with a frequency dependent on the velocities of the involved pairs, and perform collision tests only for segments that are close to contact taking into account the thickness of the simulated material, we found that these methods did not perform well when considering inextensible cloth, and added unwanted oscillation not present when using the method previously described using the proximity parameter ω.

Remark 3.2. It is usually enough to use the altered positions $\hat{p}_i = p_i \pm \omega\tau_0 \nu$, where $\omega \approx 0.5$ to detect all self-collisions, nevertheless some of them can sometimes be missed because the nodes were moved too much. In that case, we enter an iterative process reducing gradually the value of ω until all are resolved. Notice that for $\omega = 0$ no self-collision can be missed since the altered positions $\hat{p}_i = p_i \pm \omega\tau_0 \nu$ coincide with p_i. See Section 4.1 for details on how this iterative process is coupled with our numerical integration scheme.

Definition 1. (Self-collision constraints). We will denote by

$$C = \text{Collisions}_\omega \left(\boldsymbol{\varphi}^n \rightarrow_{\Delta t} \boldsymbol{\varphi}^{n+1} \right)$$

the set of self-collision constraints detected when moving from the state $\boldsymbol{\varphi}^n$ to the state $\boldsymbol{\varphi}^{n+1}$ with proximity parameter $\frac{1}{2} > \omega \geq 0$.

4. Numerical integration of the system

In this section, a novel numerical discretization is presented in order to integrate implicitly the equations of motion of the cloth. This discretization leads naturally to a sequence of quadratic problems with inequality constraints. We explain in detail how to include self-collision constraints under this scheme.

The friction force and the contact constraints introduced in Section 2 are in general highly non-linear and stiff and thus must be integrated implicitly. The same situation applies to the inextensibility constraints, see [3]. To integrate the system numerically from time t_n to t_{n+1} (i.e.

to advance the simulation from $\boldsymbol{\varphi}^n$ to $\boldsymbol{\varphi}^{n+1}$), we perform as in [3] an iterative process $\boldsymbol{\varphi}_{j+1} = \boldsymbol{\varphi}_j + \Delta\boldsymbol{\varphi}_{j+1}$ where the initial point is the unconstrained step $\boldsymbol{\varphi}_0 = \boldsymbol{\varphi}_0^{n+1}(\boldsymbol{\varphi}^n, \dot{\boldsymbol{\varphi}}^n)$ given by applying an implicit Euler scheme to the unconstrained equations of motion (i.e. to $\mathbf{M}\ddot{\boldsymbol{\varphi}} = \mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})$, see Equations (2)). 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 similarly

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

and then solve iteratively for $\Delta\boldsymbol{\varphi}_{j+1}$ the following sequence of quadratic programs with linear equality and inequality constraints:

$$\begin{cases} \min_{\Delta\boldsymbol{\varphi}_{j+1}} \frac{1}{2}\Delta\boldsymbol{\varphi}_{j+1}^{\intercal} \cdot \mathbf{M} \cdot \Delta\boldsymbol{\varphi}_{j+1} - \Delta\boldsymbol{\varphi}_{j+1}^{\intercal} \cdot \mathbf{F}_\mu(\dot{\boldsymbol{\varphi}}_j) \\ \mathbf{C}(\boldsymbol{\varphi}_j) + \nabla\mathbf{C}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = \mathbf{0}, \\ \mathbf{H}(\boldsymbol{\varphi}_j) + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} \geq \mathbf{0}, \end{cases} \tag{5}$$

where

1. $\dot{\boldsymbol{\varphi}}_j = \frac{\boldsymbol{\varphi}_j - \boldsymbol{\varphi}^n}{\Delta t}$ is the approximation of $\dot{\boldsymbol{\varphi}}^{n+1}$ at iteration j,
2. $\mathbf{F}_\mu(\dot{\boldsymbol{\varphi}}_j) = -\mu\mathbf{V}(\dot{\boldsymbol{\varphi}}_j)^{\intercal}\Delta\boldsymbol{\beta}_j$ is the friction force at iteration j,
3. $\mathbf{V}(\dot{\boldsymbol{\varphi}}_j) \in \mathbb{R}^{n_H \times 3N}$ are the relative unit tangent velocities, where each row $i = 1, \dots n_H$ is computed by removing from $\dot{\boldsymbol{\varphi}}_j$ its normal component in the direction of $\nabla H_i(\boldsymbol{\varphi}_j)$ and then normalizing,
4. $(\Delta\beta_j)_i = \|(\Delta\gamma_j)_i \nabla H_i(\boldsymbol{\varphi}_j)^{\intercal}\|$ is the magnitude of the contact forces at iteration j, where $(\Delta\beta_j)_i$ is the i-th coordinate of the vector $\Delta\boldsymbol{\beta}_j \in \mathbb{R}^{n_H}$,
5. $\Delta\boldsymbol{\gamma}_j \geq \mathbf{0}$ are the multipliers associated with the contact (inequality) constraints,
6. and $\Delta\boldsymbol{\gamma}_0 = \mathbf{0}$ and $\Delta\boldsymbol{\beta}_0 = \mathbf{0}$ are both initialized to $\mathbf{0}$.

We iterate until the equality and inequality constraints are satisfied up to some given tolerances $\epsilon_0, \epsilon_1 > 0$ and the infinity norm of the increment $\Delta\boldsymbol{\varphi}_j$ is also bellow some $\epsilon_2 > 0$ (this ensures that the friction force has stabilized). In symbols this means:

$$\|\mathbf{C}(\boldsymbol{\varphi}_j)\|_\infty < \epsilon_0, \quad \ell(\mathbf{H}(\boldsymbol{\varphi}_j)) \geq -\epsilon_1, \quad \|\Delta\boldsymbol{\varphi}_j\|_\infty < \epsilon_2, \tag{6}$$

where for any vector $\mathbf{z} = (z_1, \dots, z_m) \in \mathbb{R}^m$ we have $\|\mathbf{z}\|_\infty = \max(|z_1|, \dots, |z_m|)$ and $\ell(\mathbf{z}) = \min(z_1, \dots, z_m)$.

Note that the critical points of the previous quadratic problems (5) are:

$$\begin{cases} \mathbf{M} \cdot \Delta\boldsymbol{\varphi}_{j+1} = -\nabla\mathbf{C}(\boldsymbol{\varphi}_j)^{\intercal}\Delta\boldsymbol{\lambda}_{j+1} + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)^{\intercal}\Delta\boldsymbol{\gamma}_{j+1} - \mu\mathbf{V}(\dot{\boldsymbol{\varphi}}_j)^{\intercal}\Delta\boldsymbol{\beta}_j, \\ \mathbf{C}(\boldsymbol{\varphi}_j) + \nabla\mathbf{C}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = \mathbf{0}, \\ \mathbf{H}(\boldsymbol{\varphi}_j) + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} \geq \mathbf{0}, \\ \Delta\boldsymbol{\gamma}_{j+1} \geq \mathbf{0}, \quad \Delta\boldsymbol{\gamma}_{j+1}^{\intercal} \cdot \left[\mathbf{H}(\boldsymbol{\varphi}_j) + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1}\right] = 0, \\ (\Delta\beta_j)_i = \|(\Delta\gamma_j)_i \nabla H_i(\boldsymbol{\varphi}_j)^{\intercal}\|, \ i = 1, \dots n_H. \end{cases} \tag{7}$$

These are the Karush–Kuhn–Tucker (KKT) conditions (see e.g. [32]) of the optimization problem (5), and since this problem is quadratic and the equality and inequality constraints are linear, satisfying these conditions is necessary and sufficient for optimality for each $j \geq 0$.

Remark 4.1. In order to integrate numerically the friction force, we have dropped the gradient of $\mathbf{F}_\mu$ that would normally appear in the first equation of (7) if we had applied Newton's method to an implicit first-order scheme for the equations of motion (2). This is what is also done with the gradient of the constraint forces, see [5] for more details.

4.1. Addition of self-collision constraints

Instead of checking and generating all self-collision constraints only with the states $\boldsymbol{\varphi}^n$ and $\boldsymbol{\varphi}^{n+1}$, i.e. computing $C = \text{Collisions}_\omega\left(\boldsymbol{\varphi}^n \to_{\Delta t} \boldsymbol{\varphi}^{n+1}\right)$, we take advantage of the fact that we perform an iterative process. We now explain how we introduce self-collisions into the sequence of problems (5) for every iteration j: we check for self-collisions (see Section 3.1) taking into account the thickness of the cloth (Section 3.3) between the states $\boldsymbol{\varphi}^n$ and $\boldsymbol{\varphi}_j$ (using of course that $\dot{\boldsymbol{\varphi}}_j = \frac{\boldsymbol{\varphi}_j - \boldsymbol{\varphi}^n}{\Delta t}$) and generate the corresponding constraints (Section 3.2) in order to resolve them in the next iteration. In symbols this means that all the constraints $C_j = \text{Collisions}_\omega\left(\boldsymbol{\varphi}^n \to_{\Delta t} \boldsymbol{\varphi}_j\right)$ for $j \geq 0$ and $\omega \approx 0.5$ are added to the system for the next iteration $j+1$. In the rare case that the same collision is found in two different iterations $j_1 < j_2$ we only keep the constraint defined by the later iteration j_2. Then, when we find a state $\boldsymbol{\varphi}_{j^*}$ that satisfies the stopping criteria (6), we check for self-collisions with $\omega = 0$ between the states $\boldsymbol{\varphi}^n$ and $\boldsymbol{\varphi}_{j^*}$, and in case no self-collision is detected, we put $\boldsymbol{\varphi}^{n+1} = \boldsymbol{\varphi}_{j^*}$. Otherwise, we repeat the whole iteration process from $\boldsymbol{\varphi}_0$ with a new smaller value of omega: $\omega_{\text{new}} \leftarrow 0.9\omega_{\text{old}}$. This process is guaranteed to end since for $\omega = 0$ all self-collisions are detected and resolved (see Remark 3.2).

5. Efficient solution of the quadratic problems

In this section, we study how to solve efficiently the sequence of quadratic problems defined before. We present a novel *active-set* method tailored to our problem. A detailed procedure is laid out in pseudo-code in Algorithm 1.

Definition 2 (Active constraint)**.** In a constrained optimization problem (such as (5)), we say that an inequality constraint $G(\boldsymbol{\varphi}) \geq 0$, $G : \mathbb{R}^{3N} \to \mathbb{R}$ is *active* at a feasible point $\hat{\boldsymbol{\varphi}} \in G^{-1}([0, +\infty)) \subset \mathbb{R}^{3N}$ if $G(\hat{\boldsymbol{\varphi}}) = 0$. Otherwise, when $G(\hat{\boldsymbol{\varphi}}) > 0$, we say it is *inactive*. Note that all equality (in our case inextensibility) constraints are always active.

In order to solve the sequence of problems (5) we could employ any quadratic problem solver, but we would not be taking advantage of the structure of our problem. That is, if in one of the iterations j one of the contact constraints $H_i \geq 0$ is active (see the previous definition), then it is likely that it will be active again at the next iteration. Physically, this means that nodes of the cloth that are in contact with an obstacle (or among themselves) at some iteration, are likely to remain in contact. This suggests the use of active-set-methods [32] to solve the quadratic problems. We will develop a novel *active-set* algorithm in the following pages. Although we could use one of the many existing ones, they always require that one begins the solution to each problem with a feasible (albeit not optimal) solution, i.e. in our case this would mean having to compute for each $j \geq 0$ a (non-optimal) feasible increment $\Delta\hat{\boldsymbol{\varphi}}_{j+1}$ satisfying the second and third equations of (5). Our method will not have this requirement.

The main idea of active-set methods is to find all the constraints that are *active* at the optimal solution, because then, once known, the problem can be solved by ignoring inactive constraints, and assuming that all active inequality constraints are equality constraints. Recall that solving quadratic problems with equality constraints is very cheap and can be done by solving a linear system (see [5]). This will be precisely what we will do for every iteration of the sequence (5). In order to find the active set, one splits the constraints in two sets:

The working set, $\mathcal{W}$: these are the constraints believed to be active ($G = 0$) and therefore are imposed as equality constraints when one solves the optimization problem. This can be initialized as the set consisting only of equality constraints.

The observation set, $\mathcal{O}$: these are the constraints believed to be inactive ($G > 0$) and therefore are not imposed as equality constraints. Since they are not included in the problem one must be careful that they do not become violated.

Then our method will proceed as follows:

1) solve the equality problem defined by the working set;
2) compute the Lagrange multipliers of the working set for the inequality constraints;
3) send some subset of the constraints with negative Lagrange multipliers to the observation set and remove them from the working set;
4) if all multipliers are positive, check if all constraints in the observation set remain feasible;
5) send some subset of the infeasible constraints to the working set and remove them from the observation set;
6) repeat until all multipliers for the inequality constraints are positive and all constraints in the observation set are satisfied.

Then, going back to our problem, this means that when we find an increment $\Delta\boldsymbol{\varphi}_{j+1}$ such that all contact inequality constraints in the working set have positive Lagrange multipliers $\Delta\boldsymbol{\gamma}_{j+1} \geq \mathbf{0}$ (see Equation (7)) and all constraints in the observation set are not violated, we have found the optimal solution to (5) (see [32]) and we can make the update $\boldsymbol{\varphi}_{j+1} = \boldsymbol{\varphi}_j + \Delta\boldsymbol{\varphi}_{j+1}$. The following proposition ensures that we do not enter a never-ending cycle:

Proposition 1 (Entry and exit of constraints)**.** Consider the system of equations arising as the critical points of the quadratic function $\frac{1}{2}\Delta\boldsymbol{\varphi}^\intercal\mathbf{M}\Delta\boldsymbol{\varphi}$ with equality constraints $\mathbf{H}(\boldsymbol{\varphi}) + \nabla\mathbf{H}(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} = \mathbf{0}$:

$$\begin{cases} \mathbf{M}\Delta\boldsymbol{\varphi} = \nabla\mathbf{H}(\boldsymbol{\varphi})^\intercal\Delta\boldsymbol{\gamma}, \\ \mathbf{H}(\boldsymbol{\varphi}) + \nabla\mathbf{H}(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} = \mathbf{0}, \end{cases} \tag{8}$$

where $\Delta\boldsymbol{\gamma}$ are the Lagrange multipliers associated to the equality constraints; and similarly, the system (with a different solution $\Delta\tilde{\boldsymbol{\varphi}}$ and associated multipliers $\Delta\tilde{\boldsymbol{\gamma}}$)

$$\begin{cases} \mathbf{M}\Delta\tilde{\boldsymbol{\varphi}} = \nabla\mathbf{H}^{-k}(\boldsymbol{\varphi})^\intercal\Delta\tilde{\boldsymbol{\gamma}}, \\ \mathbf{H}^{-k}(\boldsymbol{\varphi}) + \nabla\mathbf{H}^{-k}(\boldsymbol{\varphi})\Delta\tilde{\boldsymbol{\varphi}} = \mathbf{0}, \end{cases} \tag{9}$$

arising as the critical points of the quadratic function $\frac{1}{2}\Delta\boldsymbol{\varphi}^\intercal \mathbf{M}\Delta\boldsymbol{\varphi}$ but with constraints $\mathbf{H}^{-k}(\boldsymbol{\varphi}) + \nabla\mathbf{H}^{-k}(\boldsymbol{\varphi})\Delta\tilde{\boldsymbol{\varphi}} = \mathbf{0}$, which are the same from the first system but removing the single constraint $H_k(\boldsymbol{\varphi}) + \nabla H_k(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} = 0$. Then it holds that

$$\Delta\gamma_k \cdot (H_k(\boldsymbol{\varphi}) + \nabla H_k(\boldsymbol{\varphi})\Delta\tilde{\boldsymbol{\varphi}}) \leq 0, \tag{10}$$

where $\Delta\gamma_k$ is the Lagrange multiplier of the removed constraint k.

Proof. Subtracting the first two equations of the systems, we get:

$$\mathbf{M}(\Delta\tilde{\boldsymbol{\varphi}} - \Delta\boldsymbol{\varphi}) = \nabla\mathbf{H}^{-k}(\boldsymbol{\varphi})^\intercal(\Delta\tilde{\boldsymbol{\gamma}} - \Delta\boldsymbol{\gamma}^{-k}) - \nabla H_k(\boldsymbol{\varphi})^\intercal\Delta\gamma_k.$$

Then, multiplying both sides by $(\Delta\tilde{\boldsymbol{\varphi}} - \Delta\boldsymbol{\varphi})^\intercal$, we deduce that

$$0 \leq (\Delta\tilde{\boldsymbol{\varphi}} - \Delta\boldsymbol{\varphi})^\intercal \cdot \mathbf{M} \cdot (\Delta\tilde{\boldsymbol{\varphi}} - \Delta\boldsymbol{\varphi}) = 0 - (\Delta\tilde{\boldsymbol{\varphi}} - \Delta\boldsymbol{\varphi})^\intercal \cdot \nabla H_k(\boldsymbol{\varphi})^\intercal\Delta\gamma_k,$$

since $(\Delta\tilde{\boldsymbol{\varphi}} - \Delta\boldsymbol{\varphi})^\intercal \cdot \nabla\mathbf{H}^{-k}(\boldsymbol{\varphi})^\intercal = \mathbf{H}^{-k}(\boldsymbol{\varphi})^\intercal - \mathbf{H}^{-k}(\boldsymbol{\varphi})^\intercal = \mathbf{0}$. Finally, using that $\nabla H_k(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} = -H_k(\boldsymbol{\varphi})$, and rearranging terms we get

$$0 \leq -\Delta\tilde{\boldsymbol{\varphi}}^\intercal\nabla H_k(\boldsymbol{\varphi})^\intercal\Delta\gamma_k - H_k(\boldsymbol{\varphi})\Delta\gamma_k.$$

From here (10) follows easily. □

Corollary 1. If a constraint in the working set has a negative Lagrange multiplier, when it is taken out of the system and put in the observation set, it becomes feasible. Conversely, when a constraint in the observation set is infeasible and we send it to the active set, its associated Lagrange multiplier is positive.

Remark 5.1. The heuristic that is usually followed to decide which constraint to remove or to add is: delete from the working set the constraint with the most negative Lagrange multiplier and add to the working set the constraint from the observational set that is being most violated (the most negative one).

To finish this section we study the case of linearly dependent constraints. This is relevant since in general, we do not want to introduce linearly dependent constraints into the system because they give rise to (near) singular matrices. Assume that we start with all constraints in the working set having linearly independent gradients.

Lemma 1. If a constraint G in the observation set can be written as a linear combination of constraints of the working set $H_i(\boldsymbol{\varphi}) + \nabla H_i(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} = 0$, i.e. $G(\boldsymbol{\varphi}) = \sum \alpha_i H_i(\boldsymbol{\varphi})$, then the linearized constraint is feasible $G(\boldsymbol{\varphi}) + \nabla G(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} \geq 0$.

Proof.

$$G(\boldsymbol{\varphi}) + \nabla G(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} = G(\boldsymbol{\varphi}) + \sum \alpha_i \nabla H_i(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} = G(\boldsymbol{\varphi}) - \sum \alpha_i H_i(\boldsymbol{\varphi}) = 0.$$

□

Remark 5.2. The previous lemma ensures that in general, we do not send redundant constraints from the observation set to the working set. Nevertheless, it is possible to have a degenerate case where the constraints themselves are not linear combinations of each other but their gradients are. In symbols, this would mean that a constraint in the observation set satisfies $G(\boldsymbol{\varphi}) + \nabla G(\boldsymbol{\varphi})\Delta\boldsymbol{\varphi} \leq 0$ and moreover $\nabla G(\boldsymbol{\varphi}) = \sum \alpha_i \nabla H_i(\boldsymbol{\varphi})$ but $G(\boldsymbol{\varphi}) \neq \sum \alpha_i H_i(\boldsymbol{\varphi})$. What we do then is to introduce G in the working set while removing the H_i with the greatest $\alpha_i \neq 0$ in absolute value. The constraints in this new working set have linearly independent gradients since otherwise it would contradict the assumption that the original working set without G had linearly independent gradients.

5.1. Factorization of the matrix system

Every time that a constraint goes from the working set to the observation set (or vice versa), i.e. when the index sets $\mathcal{W}$ and O are updated, the Lagrange multipliers must be recomputed, i.e. a linear system must be solved in order to find the solution of (11).

$$\begin{cases} \mathbf{M} \cdot \Delta\boldsymbol{\varphi}_{j+1} = -\nabla\mathbf{C}(\boldsymbol{\varphi}_j)^\intercal \Delta\boldsymbol{\lambda}_{j+1} + \nabla\mathbf{H}(\boldsymbol{\varphi}_j)^\intercal \Delta\boldsymbol{\gamma}_{j+1} - \mu\mathbf{V}(\dot{\boldsymbol{\varphi}}_j)^\intercal \Delta\boldsymbol{\beta}_j, \\ \mathbf{C}(\boldsymbol{\varphi}_j) + \nabla\mathbf{C}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = \mathbf{0}, \\ H_i(\boldsymbol{\varphi}_j) + \nabla H_i(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = \mathbf{0} \text{ for } i \in \mathcal{W}. \end{cases} \tag{11}$$

Notice that (11) is the system of equations obtained as the critical points of the quadratic problem given by:

$$\begin{cases} \min_{\Delta\boldsymbol{\varphi}_{j+1}} \frac{1}{2}\Delta\boldsymbol{\varphi}_{j+1}^\intercal \cdot \mathbf{M} \cdot \Delta\boldsymbol{\varphi}_{j+1} - \Delta\boldsymbol{\varphi}_{j+1}^\intercal \cdot \mathbf{F}_\mu(\dot{\boldsymbol{\varphi}}_j) \\ \mathbf{C}(\boldsymbol{\varphi}_j) + \nabla\mathbf{C}(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = \mathbf{0}, \\ H_i(\boldsymbol{\varphi}_j) + \nabla H_i(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} = \mathbf{0} \text{ for } i \in \mathcal{W}. \end{cases} \tag{12}$$

In order to ease readability we will include inextensibility constraints and the contact constraints of the working set in only one function denoted by $\mathbf{G} : \mathbb{R}^{3N} \to \mathbb{R}^{n_C + n'_H}$, where $n'_H \leq n_H$ is the number of inequality constraints in the working set. Now, since in general only one constraint will be entering or exiting at the time, the linear systems that we have to solve are almost identical with the exception of a few rows and columns. That is why, the use of factorizations becomes an important tool to achieve efficiency. The linear system we need to solve to find the multipliers of (11) is:

$$\left(\nabla\mathbf{G}(\boldsymbol{\varphi}_j)\mathbf{M}^{-1}\nabla\mathbf{G}(\boldsymbol{\varphi}_j)^\intercal\right)\Delta\boldsymbol{\zeta}_{j+1} = -\mathbf{G}(\boldsymbol{\varphi}_j) - \nabla\mathbf{G}(\boldsymbol{\varphi}_j)\mathbf{M}^{-1}\mathbf{F}_\mu(\boldsymbol{\varphi}_j), \tag{13}$$

where $\Delta\boldsymbol{\zeta}_{j+1}^\intercal = [\Delta\boldsymbol{\lambda}_{j+1}^\intercal, \Delta\boldsymbol{\gamma}_{j+1}^\intercal]$. Hence the system matrix (called in the literature the Delassus' matrix [33]) is positive definite (since $\mathbf{M}$ is positive definite because it is the mass matrix). Therefore we can use the Cholesky decomposition [34] of the matrix $\nabla\mathbf{G}(\boldsymbol{\varphi}_j)\mathbf{M}^{-1}\nabla\mathbf{G}(\boldsymbol{\varphi}_j)^\intercal$, provided our constraints have linearly independent gradients (see again Lemma 1 and Remark 5.2). Every time a constraint enters or exits the working set, the Cholesky decomposition of the system matrix can be efficiently updated without recomputing the factorization from scratch (see, e.g. [35, 36])

5.2. Detailed algorithm for collisions

To finish this section we give a detailed description of the full numerical algorithm written in pseudo-code in Algorithm 1.

Remark 5.3. We now make some comments about Algorithm 1:

1 The working set is always initialized at least with the inextensibility constraints, but we can also add the active contact constraints from the previous time-step t_{n-1}.

2 We have not explicitly written the friction force, but it obviously comes up in the computation of the multipliers and the increment (lines 7 and 9, see Equations (11)).

3 Note that after successfully finding the active set of constraints at $\Delta\boldsymbol{\varphi}_{j+1}$ (line 10), for the next step $j+1$, we do not change the working set $\mathcal{W}$. Only the observation set O is updated with the possible new self-collision constraints found (line 12).

Algorithm 1 Collisions active-set algorithm

Require: $\boldsymbol{\varphi}^n, \dot{\boldsymbol{\varphi}}^n$

1: $\boldsymbol{\varphi}_0 \leftarrow \text{unconstrained}(\boldsymbol{\varphi}^n, \dot{\boldsymbol{\varphi}}^n, \dots),\ j \leftarrow 0$ ▷ (*Implicit Euler integration step*)
2: $C_0 = \text{Collisions}_\omega\,(\boldsymbol{\varphi}^n \rightarrow_{\Delta t} \boldsymbol{\varphi}_0)$ ▷ (*Self-collision constraints*)
3: $\mathcal{W} \leftarrow \{i : G_i(\boldsymbol{\varphi}^n) = 0\},\ O \leftarrow \mathcal{W}^c$ ▷ (*Working and observation sets*)
4: $\boldsymbol{J} \leftarrow \text{gradient}(\boldsymbol{\varphi}_0, C_0, \mathcal{W}, \dots)$ ▷ (i.e. $\nabla G_i(\boldsymbol{\varphi}_0),\ i \in \mathcal{W}$)
5: $\boldsymbol{L} \leftarrow \text{cholesky}\left(\mathbf{J} \cdot \mathbf{M}^{-1} \cdot \mathbf{J}^\intercal\right)$
6: **while** $\|\mathbf{C}(\boldsymbol{\varphi}_j)\|_\infty \geq \epsilon_0$ or $\ell(\mathbf{H}(\boldsymbol{\varphi}_j)) \leq -\epsilon_1$ or $\|\Delta\boldsymbol{\varphi}_j\|_\infty \geq \epsilon_2$ **do**
7: $[\Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\gamma}] = \text{multipliers}(\boldsymbol{\varphi}_j, \mathbf{L})$ ▷ (Equation (13))
8: **if** $\ell(\Delta\boldsymbol{\gamma}) \geq 0$ **then**
9: $\Delta\boldsymbol{\varphi}_{j+1} \leftarrow \text{increment}(\Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\gamma}, \mathbf{J})$ ▷ (Equation (11))
10: **if** $H_i(\boldsymbol{\varphi}_j) + \nabla H_i(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1} \geq 0$ for $i \in O$ **then**
11: $\boldsymbol{\varphi}_{j+1} \leftarrow \boldsymbol{\varphi}_j + \Delta\boldsymbol{\varphi}_{j+1}$
12: $C_{j+1} = \text{Collisions}_\omega\left(\boldsymbol{\varphi}^n \rightarrow_{\Delta t} \boldsymbol{\varphi}_{j+1}\right),\ O \leftarrow O \cup C_{j+1}$
13: $\boldsymbol{J} \leftarrow \text{gradient}(\boldsymbol{\varphi}_{j+1}, \cup_k C_k, \mathcal{W}, \dots)$
14: $\boldsymbol{L} \leftarrow \text{cholesky}\left(\mathbf{J} \cdot \mathbf{M}^{-1} \cdot \mathbf{J}^\intercal\right),\ j \leftarrow j + 1$
15: **else**
16: $i_{in} \leftarrow \text{indmin}_{i \in O}(H_i(\boldsymbol{\varphi}_j) + \nabla H_i(\boldsymbol{\varphi}_j)\Delta\boldsymbol{\varphi}_{j+1})$
17: $O \leftarrow O \setminus i_{in},\ \mathcal{W} \leftarrow \mathcal{W} \cup i_{in}$
18: $\mathbf{L} \leftarrow \text{update}(\mathbf{L}, i_{in})$
19: **end if**
20: **else**
21: $i_{out} \leftarrow \text{indmin}_{i \in \mathcal{W}}(\Delta\boldsymbol{\gamma})$
22: $O \leftarrow O \cup i_{out},\ \mathcal{W} \leftarrow \mathcal{W} \setminus i_{out}$
23: $\mathbf{L} \leftarrow \text{update}(\mathbf{L}, i_{out})$
24: **end if**
25: **end while**
26: $\boldsymbol{\varphi}^{n+1} \leftarrow \boldsymbol{\varphi}_{j^*},\ \dot{\boldsymbol{\varphi}}^{n+1} \leftarrow \frac{\boldsymbol{\varphi}^{n+1} - \boldsymbol{\varphi}^n}{\Delta t}$
27: **return** $\boldsymbol{\varphi}^{n+1}, \dot{\boldsymbol{\varphi}}^{n+1}$

4 When there is a negative multiplier (line 20), note that we take out from the system the constraint with the smallest (most negative) multiplier. Likewise, when one of the constraints in the observation set must be introduced (line 15), we choose the one that is being the most violated.

5 Since every time we take a step, a sequence of quadratic optimization problems must be solved and we must detect and generate all self-collision constraints, we set a fixed time step $\Delta t > 0$ as opposed to alternative adaptative methods so as to be efficient.

5.3. Similarities and differences with common active-set methods

Now that we have presented the full algorithm we use to solve the quadratic problems, we can talk in more detail about how it compares to standard active-set methods like the one described in [32]. The main difference was already mentioned: to our knowledge, all active-set methods require that one begins at a feasible point and then iterates from there. This has the disadvantage that one must find a feasible point to begin with, e.g. solving a linear optimization problem with

equality and inequality constraints. On the other hand, those classic methods have the advantage that all the constraints in the observation set are kept non-violated, and this allows one to take smaller steps towards the solution when all the multipliers are positive (potentially causing the algorithm to converge faster). Since we are solving so many relatively large sparse quadratic problems in a row, we have found that the requirement of starting at a feasible point is way more expensive than employing the novel algorithm here presented, where we allow constraints in the observation set to be violated. This is the case because low-rank updates of sparse Cholesky decompositions can be carried out very efficiently.

6. Qualitative evaluation of the collision model

In this section, we present several experiments to test our collision model. They are qualitative in nature, i.e. we show that our simulator is capable of dealing with such scenarios. We show that our modelization of friction is effective in static (cylinder experiment, Section 6.1) and dynamic (rotating sphere experiment, Section 6.2) settings, that we can easily include collisions with sharp objects (Section 6.3) and that we can simulate complicated folding sequences of cloth with non-trivial topologies (shorts experiment, Section 6.4). The second and the third experiments are scenarios suggested by [25] as challenging tests for a robust cloth collision model.

6.1. Frictional cylinder

This is the most basic of the four experiments: a flat sheet of cloth falls on top of a frictional cylinder during 2 seconds. In Figure 1 we show the final configuration of the textile for $t = 2$. All the physical parameters are kept constant but friction, which varies among $\mu \in \{0.2, 0.4, 0.55\}$. The cylinder is 30 cm off-center (with respect to the center of mass of the cloth whose measures are 130 cm $\times$ 130 cm) and the textile is slightly rotated (20 degrees with respect to the z-axis). This means that in the absence of friction (or with a small friction coefficient), the cloth collides with the cylinder and then falls to the floor. We show with this scenario that our implementation of friction is effective and can handle scenarios with persistent contact. In the first image of

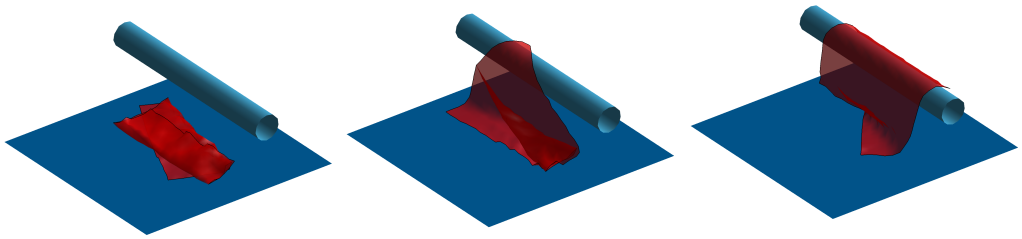


Figure 1: Final frame ($t = 2$ seconds) of 3 separate simulations of the fall of a sheet of cloth on top of an off-center cylinder. All the physical parameters are kept constant but friction, which varies among $\mu \in \{0.2, 0.4, 0.55\}$.

Figure 1, we have that $\mu = 0.2$ is too small and therefore the cloth falls onto the floor. In the second image, the friction $\mu = 0.4$ is somehow bigger and the cloth can be seen still in the process of falling but at a later stage, which shows that the friction forces have acted and delayed the fall. Finally, in the third image with $\mu = 0.55$, the friction is high enough so that the sheet lies stably on top of the cylinder. For a video of the three simulations, see the Supplementary Video 1.

6.2. Rotating sphere

In this second experiment, we simulate the collision of a sheet of cloth with a frictional sphere and the floor. The cloth measures 190 cm $\times$ 190 cm whereas the sphere has a radius of 35 cm. One second after the textile has fallen, the sphere performs half a rotation along the z-axis during one second. The discrepancy in size is intentional so that after the fall the textile is also in contact with the floor and can then wrap around the sphere. In Figure 2 we can see the final frame of the simulation at $t = 2.5$ seconds. For a full video of the simulation, see the Supplementary Video 2.

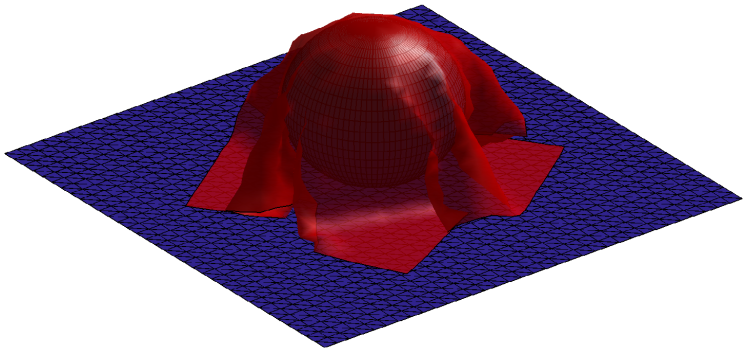


Figure 2: Final frame ($t = 2.5$ seconds) of the collision of a sheet of cloth with a frictional sphere and the floor. One second after the textile has fallen, the sphere performs half a rotation along the z-axis during one second.

In order to simulate properly the sliding contacts between the cloth and the sphere, we must take into account the speed of the rotation for the points of the mesh that are in contact with the sphere. In practice this means that when computing at every iteration j of our solver the tangent velocities at points of contact, we must subtract the speed of the sphere at the current time step (as before the tangent velocities are afterwards normalized).

In order to obtain an interesting behavior of the simulation, it is important to calibrate carefully the interplay between the friction with the floor and with the sphere. We select $\mu_{sphere} = 0.5$ and $\mu_{floor} = 0.4$, so that the cloth follows the rotation of the sphere but with considerable resistance from the floor.

6.3. Collision with a sharp obstacle

In this third experiment, we simulate the collision of a piece of cloth with a collection of needle-like obstacles. They are given by the implicit surface:

$$h(x, y, z) = c_1 c_2 z - \sin(c_1 x) \sin(c_1 y) = 0, \tag{14}$$

where we take $c_1 = 20$ and $c_2 = 0.075$ (see Figure 3).

The interest of this scenario lies in the fact that it is not enough to impose the previous equation (14) as a hard constraint for every node of the meshed cloth (as we did with the sphere and the cylinder, see Remark 2.2); but that we need in addition to take into account the *peaks* of the surface. These difficulties are typical for most physical simulators and they arise when the obstacles we are simulating present characteristics of lower dimensional objects (e.g. a really

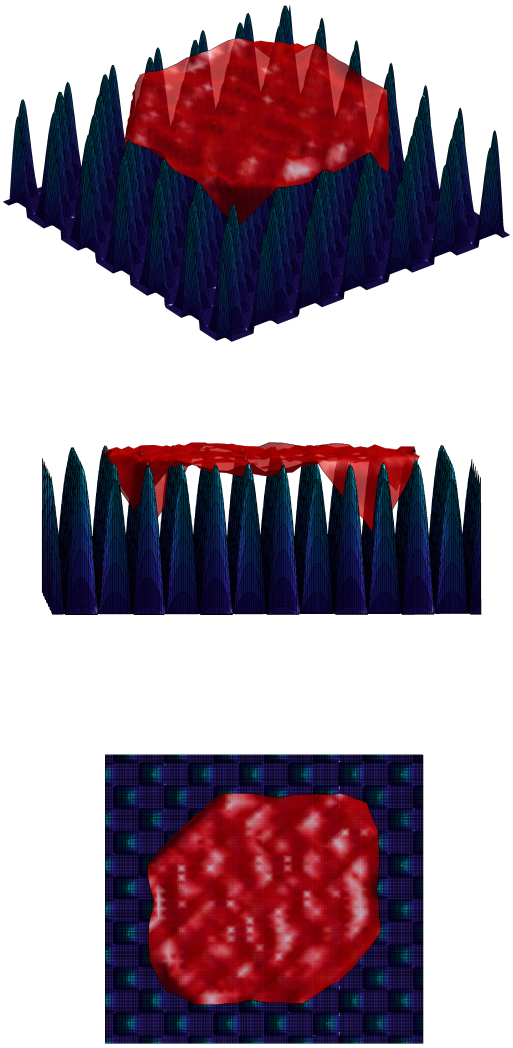


Figure 3: Simulated final frame of cloth's collision with a collection of needle-like obstacles seen from 3 different angles. The peaks must be taken into account separately from the rest of the surface and are treated with the same algorithm we treat self-collisions.

thin cylinder or the peaks in this case). It is easy to see that the peaks (maxima of h) are given by:

$$x = \frac{2\pi m \pm \frac{\pi}{2}}{c_1}, \quad y = \frac{2\pi m \pm \frac{\pi}{2}}{c_1}, \quad z = \frac{1}{c_1 c_2}, \tag{15}$$

where $m \in \mathbb{Z}$.

Let us denote a subset of the peaks in a region where the cloth moves by $\{o_1, \ldots, o_f\}$. Then, for every iteration j of the solver (5), similarly like we do with self-collisions (see Section 4.1), we must check if a collision occurred during the motion from $\boldsymbol{\varphi}^n$ to $\boldsymbol{\varphi}_j$ between the peaks $\{o_1, \ldots, o_f\}$ and the triangular faces of our meshed cloth. This means that for every detected collision, in the next iteration $j+1$ we must add a constraint of the form:

$$H_i(\boldsymbol{\varphi}_{j+1}) = \langle o_i - \pi(p_1, p_2, p_3), \nu \rangle \geq 0,$$

where o_i is the corresponding peak, p_i are the 3 corners of the triangle, $\pi(p_2, p_3, p_4) = up_1 + vp_2 + wp_3$ is the closest point between the face and o_i, and ν is the normal vector to the triangle. The values ν, u, v, w are (like in the case of self-collisions) constant in time, and are computed with the positions given at the moment of collision. The normal vector ν is oriented as always so that $H_i(\boldsymbol{\varphi}^n) \geq 0$ (see Section 3.2).

Remark 6.1. As with self-collisions we consider cloth's thickness in practice by imposing $H_i(\boldsymbol{\varphi}_{j+1}) \geq \tau_0 > 0$. Moreover, as before this thickness is taken into account in the detection process (see Section 3.3).

In Figure 3 we can observe the result of the simulation from three different viewpoints. The cloth lies stably on top of the peaks without any noticeable artifacts. For a video of the simulation, see the Supplementary Video 3.

6.4. Folding sequence of short pants

In this final experiment, we simulate the dynamical folding of a pair of shorts on top of a table. In order to do so, we control two nodes at the top of the shorts. The first part of the motion is performed fast enough so that the shorts have sufficient momentum to lay partially flat on top of the table after lowering them. Finally, the fold is completed by dropping the top two corners on top of the leg loops.

In Figure 4 we depict six frames of the simulation. Notice how crucial is the well-functioning of the self-collisions algorithm for a realist outlook of the whole folding sequence. For a full video of the simulation, see the Supplementary Video 4.

7. Experimental validation of the collision model

The definitive test for a model of cloth is its comparison to experimental data coming from real scenarios. Textile engineers have focused on such tasks, to the point of developing specialized testing equipment. But the goal of their study has always been local properties of cloth, such as elasticity parameters, which are tested in static scenarios (e.g. see [37–40]). Other, more recent lines of research such as [41, 42] focus on estimating friction coefficients using non-intrusive video images. To the knowledge of the authors, none of the models previously mentioned has been able to compare its results with dynamic motions of textiles involving collisions.

To perform this set of experiments, we use a *Motion Capture System*, and record two collision scenarios of four textiles. A system of 5 cameras detects and tracks 20 reflective markers that are hooked on the cloth. These markers reflect infrared light, so the cameras are able to follow their motion through space. For more details on the recording setting see Appendix A.

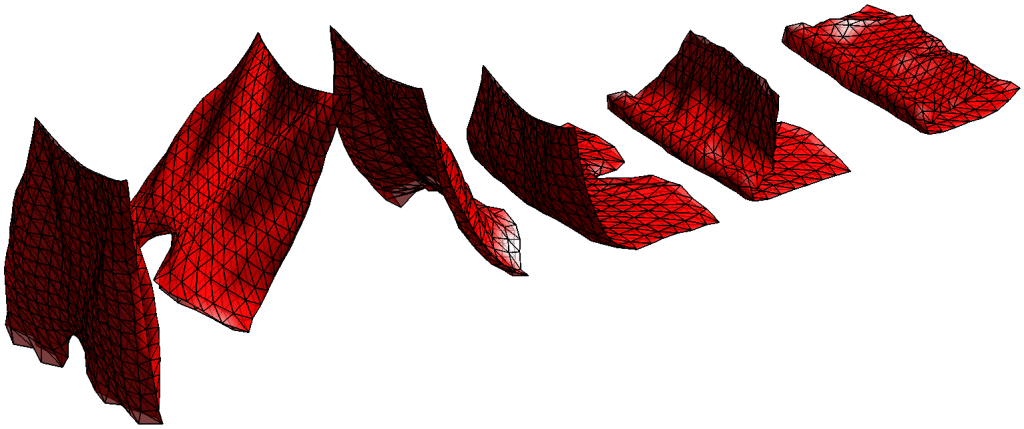


Figure 4: Simulated sequence of the dynamical folding of a pair of shorts. The first part of the motion (frames two and three) is performed fast enough so that the shorts lay partially flat on top of the table after a lowering phase (frame four). The final fold is completed by dropping the top two corners on top of the leg loops (frames five and six).

In one of the scenarios, the fabrics are laid dynamically on top of a table in a putting-a-tablecloth fashion. In the other, they are hit by a cylindrical stick four times at various places and with different strengths. The goal is to assess the accuracy of the collision and friction model previously developed. For the first scenario, we find the optimal friction parameter for both a high and a low friction case and study the stability of the model with respect to this parameter. For the hitting experiment, we find again the optimal parameters of the model and also check computational times comparing our active-set solver with a standard interior-point method.

7.1. Cloth's materials and sizes

For the experiments in this validation, we employ four cloth materials of size DIN A2 (0.42 x 0.594 m). Before performing the experiments they were ironed as to remove all considerations of plasticity from the validation process. In Table 1 we can see the density and type of all the fabrics and some typical examples of garments made from them.

Fabric	**Density** ($\mathbf{kg \cdot m^{-2}}$)	**Examples**
Polyester	0.1042	Silk-like
Wool	0.1804	Formal suit
Denim	0.3046	Jeans
Stiff-cotton	0.3046	Sack

Table 1: Density and examples of all the materials used in all the experiments.

7.2. Parameter fitting

We denote the sequence of positions of the recorded fabric's nodes given by the motion capture system every $\Delta t > 0$ seconds by $\{\boldsymbol{\phi}^0, \boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^m\}$ and the simulated sequence obtained

by using Algorithm 1 by $\{\boldsymbol{\varphi}^0, \boldsymbol{\varphi}^1, \ldots, \boldsymbol{\varphi}^m\}$. This sequence is obtained by taking $\boldsymbol{\varphi}^0 = \boldsymbol{\phi}^0$ and simulating the cloth using the recorded trajectories of the two upper corners with the same $\Delta t > 0$. In order to validate the realism of our collision model, we fit three parameters: α (Rayleigh damping, this damps long oscillations) and δ (virtual mass, this models aerodynamics), and μ (friction coefficient). The first two parameters were introduced in [3] as an accurate way to describe the dynamics of inextensible sheets of cloth without collisions, and μ was introduced in Section 2. In order to obtain their optimal value, we minimize the mean along time of the absolute error:

$$\sum_{n=1}^{m} e_n(\delta, \alpha, \mu) = \sum_{n=1}^{m} \sqrt{\|\boldsymbol{\varphi}^n(\delta, \alpha, \mu) - \boldsymbol{\phi}^n\|^2_{\mathbf{M}}}, \tag{16}$$

where $\|\cdot\|_{\mathbf{M}}$ is the norm induced by the matrix $\mathbf{M}$, i.e. $\|\mathbf{x}\|^2_{\mathbf{M}} = \mathbf{x}^\intercal \cdot \mathbf{M} \cdot \mathbf{x}$. The use of the mass matrix $\mathbf{M}$ gives a greater weight to error in nodes limiting larger elements, and a smaller weight to error in nodes limiting smaller elements. This is appropriate in an error measure that is global in the domain, and tries to be as independent of the mesh as possible. All other physical parameters (e.g. bending) are set to 0 except for the cloth density ρ which is set to its corresponding value of Table 1. For a justification of this choice of parameters, see [3]. For the simulations we consider a refinement of the initial mesh 4×5 given by the markers, that is, we employ a 7×9 resolution.

As metrics to evaluate the fitting of the model, we use the absolute error:

$$e_n(\delta, \alpha, \mu) = \sqrt{\|\boldsymbol{\varphi}^n(\delta, \alpha, \mu) - \boldsymbol{\phi}^n\|^2_{\mathbf{M}}}, \tag{17}$$

and the following *spatial* standard deviation:

$$s_n(\delta, \alpha, \mu) = \sqrt{\mathrm{Var}\left(\|p_i^n(\delta, \alpha, \mu) - \hat{p}_i^n\|_{\mathbb{R}^3}\right)}, \tag{18}$$

where $p_i^n(\delta, \alpha, \mu)$ (resp. $\hat{p}_i^n$) is the position at time $t_n = n\Delta t$ of the i-th node of the simulated cloth (resp. recorded textile) and the variance is taken along all the nodes $i = 1, \ldots, N$ of the mesh. This metric gives us an idea of the distribution of the errors on the mesh.

Remark 7.1. The errors are only computed at the recorded nodes. Furthermore, as mentioned before, some of the markers disappear for small periods of time, in those cases, they are simply excluded from the computation of the errors (no interpolation is performed).

The experiments are performed by a human (with bare hands) and consist of two scenarios which we describe in the following two subsections.

7.3. Tablecloth scenario

The textile starts suspended at about 10 cm of height and is afterwards laid dynamically (only partially, so that half of the cloth is still suspended) onto the table (see Figure 5). Each motion lasts approximately 4 seconds (with a frame every $\Delta t = 0.01$ seconds) and it is performed with two different surfaces acting as the table, one with *low* friction (a raw polished table) and one with *high* friction (a table with a tablecloth). The goal here is to estimate the friction coefficient μ (see Equation (7)) for the two different surfaces and to study the sensitivity of the model with respect to friction.

In Table 2 we can see the optimal values of the friction coefficients along with their computed errors and deviations for the low and high friction scenarios. The optimal friction coefficients

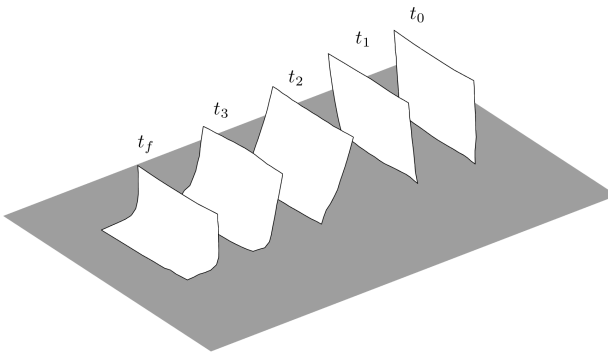


Figure 5: Putting a tablecloth motion sequence (right to left): the cloth starts suspended and it is afterwards laid dynamically (only partially) onto the table.

for the low friction case (raw polished table) were all smaller than 10^{-3} and that is why they were rounded up to zero on the table. For a visual comparison of the results see Figure 6 and the Supplementary Video 5.

Material	$\mu_{\textbf{low}}$	$\bar{e}_{\textbf{low}}$ **(cm)**	$\bar{s}_{\textbf{low}}$ **(cm)**	$\mu_{\textbf{high}}$	$\bar{e}_{\textbf{high}}$ **(cm)**	$\bar{s}_{\textbf{high}}$ **(cm)**
Polyester	0	0.95	1.20	1	0.84	1.03
Wool	0	0.58	0.73	2	0.52	0.75
Stiff-cotton	0	0.60	0.86	2	0.58	0.77
Denim	0	0.77	1.11	1.6	0.61	0.80

Table 2: Optimal values of the friction coefficients ($\mu_{\text{low}}, \mu_{\text{high}}$) along with the mean of the absolute error ($\bar{e}_{\text{low}}, \bar{e}_{\text{high}}$), see (17), and the mean of the spatial standard deviation ($\bar{s}_{\text{low}}, \bar{s}_{\text{high}}$), see (18), for the low and high friction scenarios.

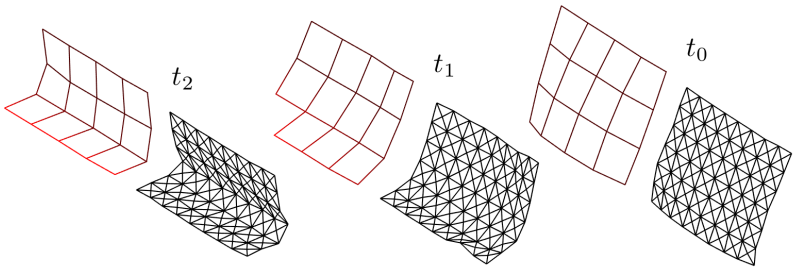


Figure 6: Three frames comparing the recorded tablecloth low friction scenario of A2 wool (left) with its inextensible simulation (right). The error at the three depicted frames from right to left is 0.80, 1.17, and 0.76 cm respectively; being the average error of the whole simulation 0.58 cm.

In order to understand how friction influences the dynamics of the textiles we perform a sensitivity analysis for the high friction case, i.e. we vary the value of μ (keeping all the other parameters fixed), and compute the mean of the absolute error (17). The results can be seen in

the heat-map depicted in Figure 7. Notice that in general, the model is quite stable with respect to the optimal friction value.

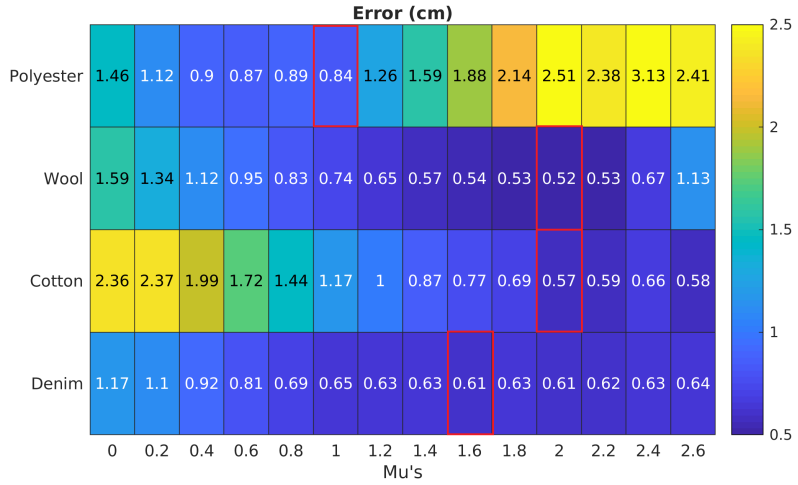


Figure 7: Sensitivity analysis for the high friction case, i.e. we vary the value of μ and compute the absolute error (17) for the four A2 fabrics. In red we encircle the error found with the optimal parameter of μ.

7.4. Hitting scenario

In this second scenario, the fabrics are held suspended in the air (with the long sides perpendicular to the floor) and hit repeatedly with a long stick. The hits are aimed at various locations of the cloth with varied strengths and speeds (see Figure 8). In order to simulate the hits, the stick is subdivided into small edges and we employ a procedure similar to the one used for self-collisions of the cloth in the case of an edge-edge collision (see Section 3).

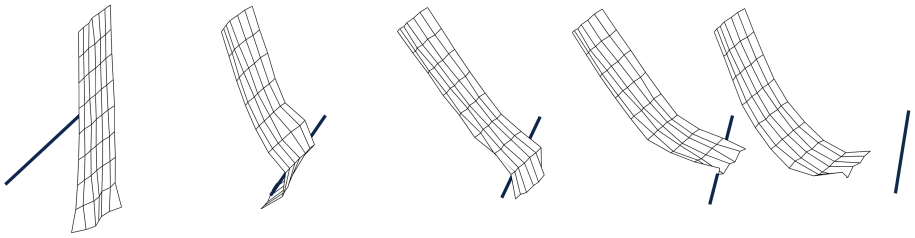


Figure 8: Long-stick hits sequence (left to right): the cloth is held by its two upper corners and then is hit repeatedly with the cylindrical stick. The hits are aimed at different locations with varied intensities.

Let us denote by $\{a_1(t), \dots, a_m(t)\}$ the endpoints of the edges of the stick. Then, for every iteration j of the iterative process (5), as we do with self-collisions (see Section 4.1), we must check if a collision occurred during the (linear) motion between the two stick-edges states

$$\{a_1(t_n), \dots, a_m(t_n)\} \text{ and } \{a_1(t_{n+1}), \dots, a_m(t_{n+1})\}$$

and the motion of the edges of our triangulated cloth between the states $\boldsymbol{\varphi}^n$ and $\boldsymbol{\varphi}_j$. This means that for every detected collision, in the next iteration $j+1$ of the sequence of quadratic problems (5) we must add a constraint of the form:

$$H(\boldsymbol{\varphi}_{j+1}) = \langle \pi_{\alpha'}(a_1, a_2) - \pi_{\beta'}(p_1, p_2), \nu \rangle \geq 0,$$

where a_1, a_2 are the two endpoints of the corresponding edge of the stick at t_{n+1}, p_1, p_2 are likewise the two endpoints of the cloth's edge, $\pi_{\alpha'}(a_1, a_2) = (1-\alpha')a_1 + \alpha' a_2$ and $\pi_{\beta'}(p_1, p_2) = (1-\beta')p_3 + \beta' p_4$ are the closest points between the two segments and ν is the normal vector to both edges. The values ν, α', β' are assumed to be constant in time, and are computed as in the case of self-collisions with the positions given by the moment of collision. The sign of ν is choosen as always so that $H(\boldsymbol{\varphi}^n) \geq 0$ (see Section 3.2).

On the other hand, the real long stick has a length of 75 cm and a diameter of 1.5 cm. Two markers with a diameter of 1.5 cm are put at both ends of the stick to record its trajectory.

Remark 7.2. We consider the stick's thickness by imposing $H(\boldsymbol{\varphi}_{j+1}) \geq \tau_0$, where $\tau_0 = 0.75$cm is the radius of the stick. Moreover, this thickness is taken into account in the detection process (see Section 3.3).

The stick is made of polished plastic and hence we consider friction between the cloth and the stick to be negligible (moreover, since the cloth is held firmly by the two upper corners the small amount of friction that could exist is always overcome by the stick). Each textile is hit four times with recordings varying between 12 and 18 seconds (as usual with a frame every $\Delta t = 0.01$ seconds). On top of fitting as usual the damping parameter α and the (virtual) gravitational mass δ, the goal of this scenario is to assess the realism of our collision algorithm when modeling the hits.

Remark 7.3. By the nature of this collision experiment, some movements of the textiles are very abrupt and therefore, as mentioned before, the markers disappear some of the time. This problem is also present in the recording of the trajectory of the stick; since the stick is rigid, to solve this issue we have found that it is enough to interpolate linearly its missing positions.

In this experiment, we also study the performance of the **active-set** collision algorithm described in Section 5. We compare it with a standard **interior-point** algorithm implemented to solve large sparse quadratic problems (e.g. as implemented in the MATLAB function `quadprog`, see Section 16.7 of [32]). We tried to compare our novel algorithm with a standard active-set method for quadratic optimization, but we were unable to find a working implementation in any programming language for large and sparse problems. All comparisons are performed using an Intel Core i7-8700K with 12 cores of 3.70 GHz. Since the four recordings have different durations, we compute the quotient

$$q = \frac{T_{\mathbf{sim}}}{T_{\mathbf{rec}}}, \tag{19}$$

where $T_{\mathbf{rec}}$ is the duration of the recording and $T_{\mathbf{sim}}$ is the amount of time it takes to simulate it. Hence $q \approx 1$ would mean that the simulations work in real-time, $q \approx 0.5$ means that they are twice as fast, etc. In Table 3 we can see the value of the absolute error and standard deviation with the optimal value of the parameters α and δ (not shown).

Material	$T_{\mathbf{rec}}$ **(sec)**	$\bar{e}$ **(cm)**	$\bar{s}$ **(cm)**	**Active-set (q)**	**Interior-point (q)**
Polyester	17	1.44	2.13	0.456	1.344
Wool	13	1.39	2.23	0.437	1.298
Denim	14	0.98	1.86	0.425	1.235
Stiff-cotton	12	1.07	1.85	0.510	1.576

Table 3: Mean absolute error $\bar{e}$ and mean spatial standard deviation $\bar{s}$ (see Equations (17) and (18)) computed with the optimal value of the parameters α and δ (not shown). In the two last columns, we display the quotient (19), i.e. $q = \frac{T_{\mathbf{sim}}}{T_{\mathbf{rec}}}$, for our active-set collision algorithm and a standard interior-point method.

For a visual comparison of the results, together with a plot of how the absolute error varies with time for the four textiles, see Figure 9 (stiff-cotton), the graphical abstract at the beginning of the paper (polyester) and the Supplementary Video 6 (for all four materials, including denim and wool). With yellow lines, we highlight the moments in which the stick is in contact with the cloth (during the simulations). Notice that it is precisely in those instants where more missing data is found. In the figures we see clearly that the error concentrates after the hit and not during it, showing that the collision model is very realistic but afterwards the aerodynamics become dominant and the errors increase. Overall the fitting is quite good, being the average error across the four fabrics 1.22 cm, see Table 3; with mean errors approximately half a centimeter bigger than those found in Section 7.3 (being 0.73 cm the average of the second column of Table 2). Finally, our active-set algorithm is found to be almost 3 times faster than a standard interior-point method, with simulations going on average 2 times faster than real-time for a 7×9 mesh (see Table 3).

8. Conclusions and further work

In this work, we delved into the problem of modeling collisions, including the response, for inextensible cloth. We explained how to incorporate contacts with an object (using Signorini's conditions), self-collisions, and Coloumb friction into the equations of motion in such a way that the model integrates all constraints (inextensibility and contacts) and friction forces at the same time without any decoupling. We developed a novel numerical discretization of contact and friction forces that can be seen as a natural extension of the *fast projection algorithm* for the efficient simulation of inextensible cloth, in order to include inextensibility, contacts, and friction in a single pass. This discretization led naturally to a sequence of quadratic problems with inequality and equality constraints. We presented a novel *active-set* method tailored to our problem which takes into account past active constraints to accelerate the resolution of unresolved contacts. The main advantage of this new algorithm with respect to standard *active-set* methods is its ability to start from any point and not necessarily from a feasible one. Moreover, we showed with different simulations that our model of friction is effective in static and dynamic settings, that collisions with sharp objects can be easily included and that complex folding sequences of cloth with non-trivial topologies (a pair of shorts) can be performed.

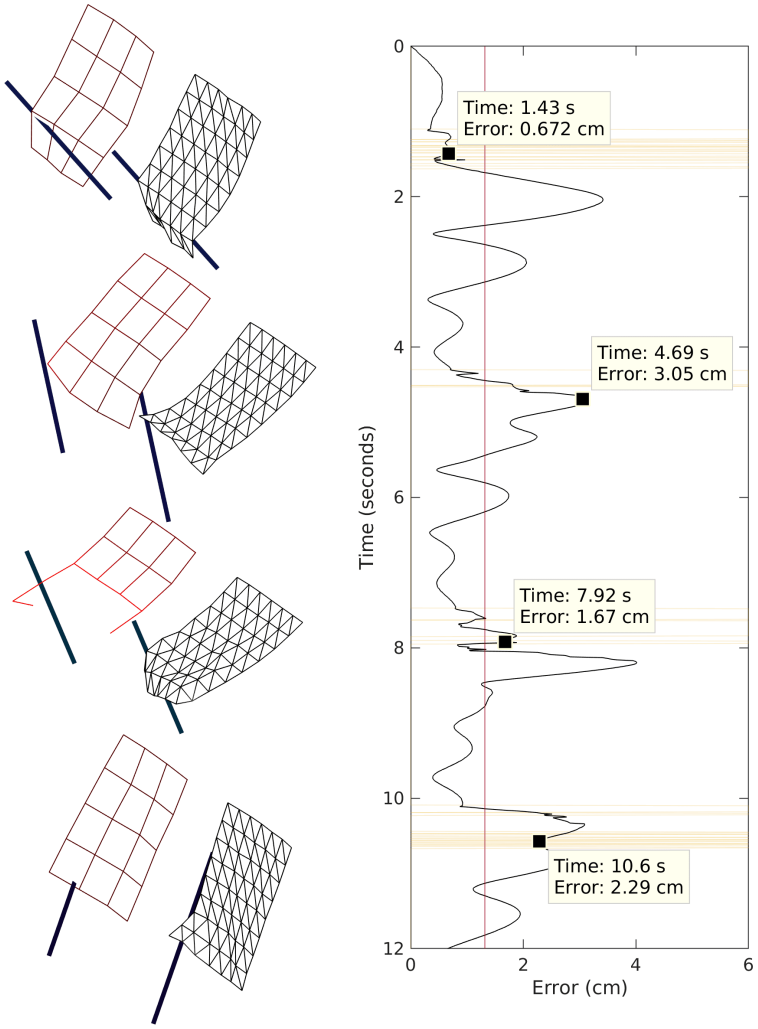


Figure 9: Four frames comparing the recorded hitting of A2 stiff-cotton (left) with its inextensible simulation (right); its average error being 1.07 cm. On the right, we show a full plot (vertically) of the absolute error and with yellow lines, we highlight the moments in which the stick is in contact with the cloth.

Finally, with the aid of a *Motion Capture System*, we embarked ourselves on the empirical validation of the developed model. We validated two different but related aspects of the collision model: its ability to simulate properly friction and to model the dynamics of fast and strong hits with a rigid object. We found the optimal friction parameters for both a high and a low friction case (see Table 2), with absolute errors under 1 cm for four DIN A2 textiles. We showed that the simulations are very stable with respect to friction by performing a sensitivity analysis. Furthermore, using only two parameters, we were able to model the most challenging scenario of this paper: four size DIN A2 cloths were held by their two upper corners and then hit repeatedly

at different locations and varied intensities with a long stick. The average errors are all around 1 cm and we were able to properly simulate the hits, appearing the biggest errors not during the hits but just after because of aerodynamic effects. The simulations on a desktop computer are two times faster than real-time (for the hitting scenario with a 7×9 mesh), being our novel active-set solver three times faster than a standard interior-point method using the same mesh resolution.

The work presented here essentially completes the task initiated with the development of the inextensible cloth model (see Equations (1)): the implementation of a mechanical model for cloth able to simulate common tasks in a human environment, such as folding, in real-time or faster, in a way that is faithful to the real behavior of cloth, with a margin of error of order 1 cm for a typical garment size. Realism, rather than *spectacularity* of the model is crucial for its usefulness in the training of Machine Learning algorithms, such as Neural Networks, to physically perform these tasks using robotic arms. The extensive work reported here for its validation has been undertaken to make our model capable of replacing actual physical manipulations of cloth with a robot and thus speeding up the training of such algorithms.

Because of this reason, the natural continuation of this work will be the application of Control Theory to the robotic control of cloth manipulation, both through a classical, deterministic approach and also using Deep and Reinforcement Learning methods. As a first step in this direction, the authors are currently integrating an implementation of the here presented model in a Virtual Reality environment to make data collection faster and easier for this application to Robotics.

9. Acknowledgments

This work 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 programme (grant agreement No. 741930). M. Alberich-Carramiñana is also with the Barcelona Graduate School of Mathematics (BGSMath) and the Institut de Matemàtiques de la UPC-BarcelonaTech (IMTech), and she and J. Amorós are partially supported by the Spanish State Research Agency AEI/10.13039/501100011033 grant PID2019-103849GB-I00 and by the AGAUR project 2021 SGR 00603 Geometry of Manifolds and Applications, GEOMVAP.

10. CRediT author statement

Franco Coltraro: Methodology, Software, Investigation, Writing - Original Draft; **Jaume Amorós:** Supervision, Formal analysis, Writing - Review and Editing; **Maria Alberich-Carramiñana:** Supervision, Project administration, Validation, Writing - Review and Editing; **Carme Torras:** Funding acquisition, Writing - Review and Editing.

References

[1] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, Y. Mezouar, Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey, Int. J. Robotic Res. 37 (2018) 688–716.

[2] H. Yin, A. Varava, D. Kragic, Modeling, learning, perception, and control methods for deformable object manipulation, Sci. Robotics 6 (2021) eabd8803.

[3] F. Coltraro, J. Amorós, M. Alberich-Carramiñana, C. Torras, An inextensible model for the robotic manipulation of textiles, Appl. Math. Model. 101 (2022) 832–858.

[4] J. Borràs, G. Alenyà, C. Torras, A grasping-centered analysis for cloth manipulation, IEEE Trans. on Robotics 36 (2020) 924–936.
[5] R. Goldenthal, et al., Efficient simulation of inextensible cloth, volume 26, ACM SIGGRAPH 2007 Papers, New York, NY, USA, 2007.
[6] J. J. Moreau, Numerical aspects of the sweeping process, Comput. methods applied mechanics engineering 177 (1999) 329–349.
[7] F. Pfeiffer, C. Glocker, Multibody dynamics with unilateral contacts, volume 421, Springer Science & Business Media, 2000.
[8] M. Kunze, Non-Smooth Dynamical Systems, volume 1744, Springer, 2000.
[9] B. Brogliato, Nonsmooth mechanics. Models, Dynamics and Control: Third Edition, volume 3, Springer, 2016.
[10] N. S. Nguyen, B. Brogliato, Multiple impacts in dissipative granular chains, Springer, 2019.
[11] B. Smith, D. M. Kaufman, E. Vouga, R. Tamstorf, E. Grinspun, Reflections on simultaneous impact, ACM Trans. on Graph. 31 (2012) 1 – 12.
[12] X. Provot, Collision and self-collision handling in cloth model dedicated to design garments, in: Computer Animation and Simulation '97. Eurographics., 1997.
[13] D. Harmon, E. Vouga, B. Smith, R. Tamstorf, E. Grinspun, Asynchronous contact mechanics, Commun. ACM 55 (2012) 102–109.
[14] M. Geilinger, et al., Add: Analytically differentiable dynamics for multi-body systems with frictional contact, ACM Trans. on Graph. 39 (2020).
[15] D. House, J. C. Keyser, Foundations of physically based modeling and animation, CRC Press, 2016.
[16] C. Liu, Z. Zhao, B. Brogliato, Frictionless multiple impacts in multibody systems. i. theoretical framework, Proc. Royal Soc. A: Math. Phys. Eng. Sci. 464 (2008) 3193–3211.
[17] R. Bridson, R. Fedkiw, J. Anderson, Robust treatment of collisions, contact and friction for cloth animation, ACM Trans. Graph. 21 (2002) 594–603.
[18] P. Volino, N. Magnenat-Thalmann, Accurate collision response on polygonal meshes, Proc. Comput. Animat. 2000 (2000) 154–163.
[19] D. Harmon, E. Vouga, R. Tamstorf, E. Grinspun, Robust treatment of simultaneous collisions, ACM Trans. Graph. 27 (2008) 1–4.
[20] M. A. Otaduy, R. Tamstorf, D. Steinemann, M. H. Gross, Implicit contact handling for deformable objects, Comput. Graph. Forum 28 (2009) 559–568.
[21] J. Li, et al., An implicit frictional contact solver for adaptive cloth simulation, ACM Trans. on Graph. 37 (2018) 1 – 15.
[22] M. Ly, J. louis Jouve, L. Boissieux, F. Bertails-Descoubes, Projective dynamics with dry frictional contact, ACM Trans. on Graph. 39 (2020) 57:1 – 57:8.
[23] Y. Li, T. Du, K. Wu, J. Xu, W. Matusik, Diffcloth: Differentiable cloth simulation with dry frictional contact, ACM Trans. on Graph. 42 (2022).
[24] G. Daviet, Simple and scalable frictional contacts for thin nodal objects, ACM Trans. Graph. 39 (2020).
[25] M. Li, D. M. Kaufman, C. Jiang, Codimensional incremental potential contact, ACM Trans. on Graph. 40 (2021) 1 – 24.
[26] D. M. Kaufman, S. Sueda, D. L. James, D. K. Pai, Staggered projections for frictional contact in multibody systems, ACM Trans. on Graph. 27 (2008) 1–11.
[27] Z. Zhong, Finite Element Procedures for Contact-Impact Problems, Oxford U.P., 1993.
[28] M. Jean, The non-smooth contact dynamics method, Comput. Methods Appl. Mech. Eng. 177 (1999) 235–257.
[29] V. Acary, B. Brogliato, Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics, Springer Science & Business Media, 2008.
[30] A. Blumentals, B. Brogliato, F. Bertails-Descoubes, The contact problem in lagrangian systems subject to bilateral and unilateral constraints, with or without sliding coulomb's friction: A tutorial, Multibody Syst. Dyn. 38 (2016).
[31] F. P. Preparata, M. I. Shamos, Computational geometry: an introduction, Springer Science & Business Media, 2012.
[32] J. Nocedal, S. J. Wright, Numerical Optimization, 2e ed., Springer, New York, NY, USA, 2006.
[33] B. Brogliato, Inertial couplings between unilateral and bilateral holonomic constraints in frictionless lagrangian systems, Multibody Syst. Dyn. 29 (2013).
[34] G. H. Golub, Matrix computations, John Hopkins University Press, 1983.
[35] M. W. Seeger, Low rank updates for the cholesky decomposition, Tech. Rep. (2004).
[36] T. A. Davis, W. W. Hager, Dynamic supernodes in sparse cholesky update/downdate and triangular solves, ACM Trans. on Math. Softw. 35 (2009) 27:1–27:23.
[37] J. Hu, Structure and mechanics of woven fabrics., The Textile Institute and Woodhead Publishing Ltd (UK), 2004.
[38] H. Wang, R. Ramamoorthi, J. F. O'Brien, Data-driven elastic models for cloth: Modeling and measurement, ACM Trans. on Graph. 30 (2011) 71:1–11.

[39] E. Miguel, et al., Data-driven estimation of cloth simulation models, Comput. Graph. Forum 31 (2012) 519–528.
[40] D. Clyde, J. Teran, R. Tamstorf, Modeling and data-driven parameter estimation for woven fabrics, Proc. ACM SIGGRAPH / Eurographics Symp. on Comput. Animat. (2017).
[41] A. H. Rasheed, et al., Learning to measure the static friction coefficient in cloth contact, 2020 IEEE/CVF Conf. on Comput. Vis. Pattern Recognit. (CVPR) (2020) 9909–9918.
[42] A. H. Rasheed, et al., A visual approach to measure cloth-body and cloth-cloth friction., IEEE Trans. on Pattern Anal. Mach. Intell. 44 (2022) 6683–6694.

Appendix A. Recording setting

To record the motion of the textiles a system of cameras detects and tracks reflective markers that are hooked on the cloth (see Figure A.10).

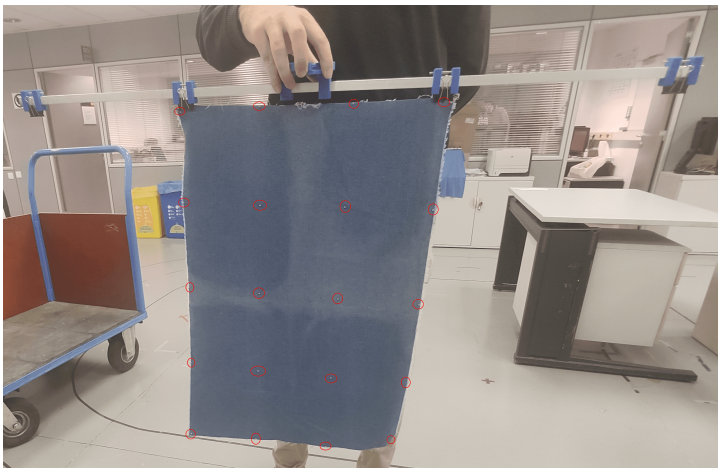


Figure A.10: Reflective markers attached to the denim sample (encircled in red). The markers are very small, with a diameter of 3 mm and a weight of 0.013 g. We use 20 reflective markers.

These markers, with a diameter of 3 mm and a weight of 0.013 g, reflect infrared light, so the cameras are able to follow their motion through space. We use hardware and software from the manufacturer *NaturalPoint Inc*: five *Optitrack Flex 13* cameras surround the scene we wish to record (see Figure A.11) and afterwards the recordings are processed with the software *Motive* such that we obtain a recorded frame every $\Delta t = 0.01$ seconds. This combination of software and hardware offers sub-millimeter marker precision, in most applications less than 0.10 mm according to the manufacturers.

This technology has been extensively used to track the motion of rigid and articulated bodies (e.g. human movements by following the trajectories of all joints). Nevertheless, its use for deformable objects has been less common since the weight of the markers could affect the dynamics of the object. This does not happen in our case since the markers we use are so light and small that account for less than 1% of cloth's weight even for the lightest materials.

We use 20 reflective markers, which are placed equidistantly in order to obtain a faithful representation of the dynamics of the fabrics. In contrast to the experiments done in [3], this time the motions are performed by a human. This introduces more uncertainty, since every movement has its own unique variabilities.

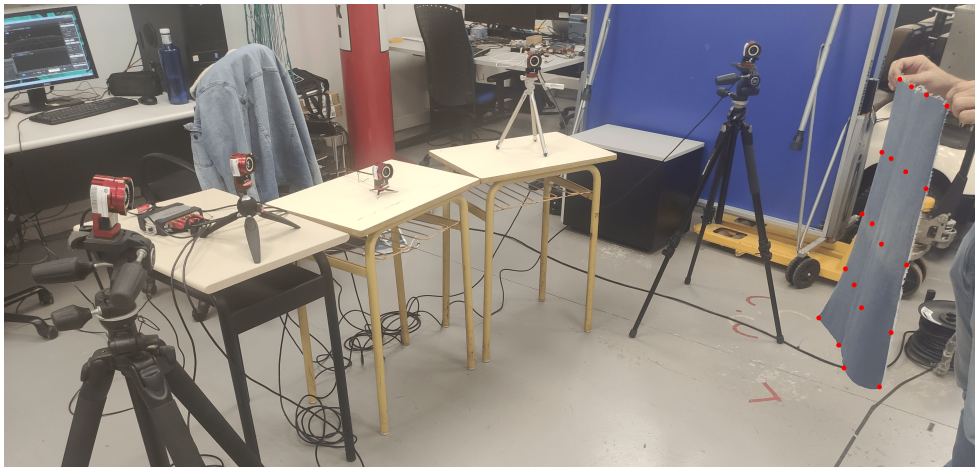


Figure A.11: Setup used to record the motion of the textiles: 5 cameras surround the scene so that every marker (highlighted in red in the photo) is visible to at least 2 cameras at the same time. This ensures that the system can be certain of the 3D position of the marker.

Since we have 5 cameras surrounding a scene, we can record more varied and faster movements without losing track of the textiles, as opposed to, e.g. using just one depth camera. Nevertheless, some markers are lost some of the time (especially with fast or abrupt movements), for instance when the textiles deform so much that the corners are no longer visible to the cameras. We have taken care that in our recordings these disappearances only happen for short periods of time.