



HAL
open science

An Implicit Frictional Contact Solver for Adaptive Cloth Simulation

Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George Brown, Laurence Boissieux

► **To cite this version:**

Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, et al.. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. ACM Transactions on Graphics, 2018, Proceedings SIGGRAPH 2018, 37 (4), pp.1-15. 10.1145/3197517.3201308 . hal-01834705

HAL Id: hal-01834705

<https://inria.hal.science/hal-01834705>

Submitted on 10 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Implicit Frictional Contact Solver for Adaptive Cloth Simulation

JIE LI, University of Minnesota, USA

GILLES DAVIET, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France and Weta Digital, New Zealand

RAHUL NARAIN, University of Minnesota, USA and Indian Institute of Technology Delhi, India

FLORENCE BERTAILS-DESCOUBES, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France

MATTHEW OVERBY and GEORGE E. BROWN, University of Minnesota, USA

LAURENCE BOISSIEUX, Univ. Grenoble Alpes, Inria, France



Fig. 1. Our implicit solver simultaneously resolves cloth elasticity, non-penetration and exact Coulomb friction constraints at every body-cloth and cloth-cloth contact, largely improving physical realism over previous methods. This new solver especially allows us to simulate accurately the effect of a varying friction coefficient μ , capturing a diversity of cloth sliding motions and folding patterns as shown in this batwing dress example (from left to right, $\mu = 0$, $\mu = 0.1$, $\mu = 0.3$, and $\mu = 0.6$). In this example featuring 2,600 contact points on average, our solver converges at each time step ($dt = 2ms$) to a high precision in a few hundred milliseconds only.

Cloth dynamics plays an important role in the visual appearance of moving characters. Properly accounting for contact and friction is of utmost importance to avoid cloth-body and cloth-cloth penetration and to capture typical folding and stick-slip behavior due to dry friction. We present here the first method able to account for cloth contact with exact Coulomb friction, treating both cloth self-contacts and contacts occurring between the cloth and an underlying character. Our key contribution is to observe that for a nodal system like cloth, the frictional contact problem may be formulated based on velocities as primary variables, without having to compute the costly Delaunay operator. Then, by reversing the roles classically played by the velocities

Authors' addresses: Jie Li, lixx4611@umn.edu, University of Minnesota, USA; Gilles Daviet, gdaviet@wetafx.co.nz, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France, Weta Digital, New Zealand; Rahul Narain, narain@cse.iitd.ac.in, University of Minnesota, USA, Indian Institute of Technology Delhi, India; Florence Bertails-Descoubes, florence.descoubes@inria.fr, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France; Matthew Overby, over0219@umn.edu; George E. Brown, brow2327@umn.edu, University of Minnesota, USA; Laurence Boissieux, laurence.boissieux@inria.fr, Univ. Grenoble Alpes, Inria, 38000, Grenoble, France.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2018 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

0730-0301/2018/8-ART52 \$15.00

<https://doi.org/10.1145/3197517.3201308>

and the contact impulses, conical complementarity solvers of the literature can be adapted to solve for compatible velocities at nodes. To handle the full complexity of cloth dynamics scenarios, we have extended this base algorithm in two ways: first, towards the accurate treatment of frictional contact at any location of the cloth, through an adaptive node refinement strategy; second, towards the handling of multiple constraints at each node, through the duplication of constrained nodes and the adding of pin constraints between duplicata. Our method allows us to handle the complex cloth-cloth and cloth-body interactions in full-size garments with an unprecedented level of realism compared to former methods, while maintaining reasonable computational timings.

CCS Concepts: • **Computing methodologies** → **Animation; Physical simulation**;

Additional Key Words and Phrases: cloth simulation, dry frictional contact, implicit solver

ACM Reference Format:

Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George E. Brown, and Laurence Boissieux. 2018. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Trans. Graph.* 37, 4, Article 52 (August 2018), 15 pages. <https://doi.org/10.1145/3197517.3201308>

1 INTRODUCTION

Cloth simulation has been a very active research area over the last decades, yielding compelling animations of cloth in feature film movies that are less and less distinguishable from reality. Relying upon mature cloth simulators, a myriad of new applications has

appeared in the latest years, such as non-invasive parameter estimation [Miguel et al. 2012], inverse design [Bartle et al. 2016; Casati et al. 2016], perceptual evaluation for artistic input [Sigal et al. 2015], sound generation [An et al. 2012], or even manipulation by controlled avatars [Clegg et al. 2015]. In computer graphics, the current background and set of tools for simulating the dynamics of cloth also gives the hope for multiple synergies to come with other disciplines: with the textile engineering community for instance, by merging virtual prototyping and real manufacturing of garments [Bingham 2012; Konaković et al. 2016]; or with the robotic and medical assistance fields, by helping measure the feeling of a person when touching and grasping fabric [Erickson et al. 2017].

However, despite such exciting advances, it is striking to note that all emerging applications suffer from a major limitation inherent to current cloth simulators, namely, an *overly simplified treatment of frictional contact*. Standard models for contact and friction in cloth date back to the 2000's [Bridson et al. 2002; Harmon et al. 2008], and have seldom been questioned since then. So far they indeed proved satisfactory enough for the movie industry, as they were filling the important request of removing any visual artifact such as cloth-body and cloth-cloth interpenetration. Nowadays, the computer graphics community is facing some more demanding challenges, where contact and friction forces should not only produce artifact-free states, but reflect the richness of physics more accurately at the macroscopic scale, while being parameterized intuitively. For instance, beyond what achieved today [Sigal et al. 2015], an artist should be able to tune a friction coefficient finely so as to explore wider ranges of cloth materials, and accurately experience stick-slip transitions during motion. Likewise, an avatar should be able to grasp a garment thanks to Coulomb friction only, without having to resort to some artificial sticking model at pinch [Clegg et al. 2015] nor to the enforcement of static friction at controlled vertices [Bai et al. 2016]. Applications at the frontier with nearby disciplines like engineering, robotics or medicine also call for much more accurate models for frictional contact than what are currently used. In our view, the time has thus come to revisit standard algorithms for contact and friction in cloth, and incorporate more physics into them.

We present here the first method able to account for cloth contact with exact Coulomb friction, treating both cloth self-contacts and contacts occurring between the cloth and an underlying character. We use a constrained optimization formulation, as in [Ostaduy et al. 2009], albeit relying upon the exact Coulomb friction cone instead of a pyramidal approximation. Furthermore, compared to previous exact Coulomb friction solvers, for instance developed for granular systems [Jean 1999] or hair dynamics [Daviet et al. 2011], a key difference is that we use a simplified though equivalent formulation that fully leverages the fact that cloth is a *nodal* system. More precisely, we make a key observation for a nodal system subject to at most one contacting constraint per node (either an external or self-contact): the frictional contact problem may be formulated based on *velocities* as primary variables instead of forces (or impulses), without having to compute the costly Delassus operator. Then, by reversing the roles classically played by the velocities and the contact impulses, conical complementarity solvers of the literature can be adapted to solve for compatible velocities at nodes. To handle

the full complexity of cloth dynamics scenarios, we have extended this base algorithm in two ways: first, through an *adaptive* node refinement strategy to allow the accurate treatment of frictional contact at any location of the cloth; second, through the duplication of constrained nodes with *pin constraints* between them to support multiple contact constraints at each node.

Our method greatly improves the realism of body-cloth and cloth-cloth interactions compared to former methods, as the resulting contact forces exactly satisfy the Coulomb constraints at each time step. As a consequence, the *friction coefficient* becomes a meaningful parameter of our simulator that can be finely tuned, and whose value is shown to have a major impact on the visual output, at the macroscopic scale (see, e.g., Figure 1). Meanwhile, this gain in realism does not go along with a major computational overhead, thanks to some careful modeling choices. Computational timings are thus kept reasonable with respect to the state of the art, ranging between 30 seconds and 2 minutes per frame for a moving garment discretized into 6,000 nodes and subject to 2,500 contact points on average. Our resulting software, named ARGUS (after the merging between ArcSim [Narain et al. 2012] and So-BOGUS [Daviet et al. 2011]), is to be freely distributed as open source.

2 RELATED WORK

Realistic cloth modeling and simulation has a rich history in computer graphics and computer-aided design. Within less than three decades, practicable solutions have been proposed to design complex garments and to animate them with an impressive level of detail. While a few recent works have considered representing woven cloth at the fiber level [Cirio et al. 2014], most approaches — including ours — model cloth at the macroscopic scale, using an assembly of thin elastic sheets which are seamed together [Baraff and Witkin 1998; Bridson et al. 2002]. In order to model full-size garments, those patches are furthermore assumed to interact with themselves and with the body through contact and dry (Coulomb) friction.

Simulating cloth elasticity. How to simulate the dynamics of thin sheets which are mainly prone to out-of-plane deformations (namely *bending*) and much less to in-plane deformations (namely *stretching* and *shearing*) has been a central research topic in the latest decades. The work by Baraff and Witkin [1998] was the first to allow for the stable simulation of full-size garments in reasonable timings. The key was to leverage first-order implicit discretization in order to deal with the inherent stiffness of cloth dynamics, even when using large time steps. Subsequent works have focussed on better formulations for bending [Bridson et al. 2003; Grinspun et al. 2003], stretch limiting [English and Bridson 2008; Goldenthal et al. 2007; Thomaszewski et al. 2009], adaptive cloth remeshing for faster simulation [Narain et al. 2012], and the fitting of numerical models with real material properties [Miguel et al. 2012; Wang et al. 2011].

Capturing contact and friction in cloth. A major concern when dealing with cloth simulation is the proper handling of frictional contact. Classically, contact and friction are viewed as an interaction model between geometrically distinct solid bodies, a strategy named *discrete element* modeling. A notable exception to that streamline is the recent work by Jiang et al. [2017] where cloth is modelled as a non-Newtonian fluid using a Material Point Method (MPM) adapted

to codimensional elasticity. A great advantage of this mixed Eulerian-Lagrangian approach is that frictional contact is automatically taken into account through a nonsmooth constitutive law, similarly to the Drucker-Prager law for granular materials but adapted to surfaces. The method however suffers from excessive stretching. Moreover, the use of an explicit time integrator imposes the using of very small timesteps. These two features are in contrast with our method where soft to very stiff surfaces can be coupled with frictional contact using moderate time steps, thanks to implicit time integration. In the remainder of this section, we focus on methods targeted at discrete element modeling.

Penalty-based contact solvers. Most popular approaches in computer graphics and mechanical engineering consist in assuming that the objects in contact are locally *compliant*, allowing them to slightly penetrate each other. This is the principle of *penalty-based methods*, which consists in adding mutual repulsive forces of the form $k f(\delta)$, where δ is the penetration depth detected at current time step [Cundall 1971; Moore and Wilhelms 1988]. Though simple to implement and computationally efficient, the penalty-based method often fails to prevent excessive penetration of the contacting objects, which may prove fatal in the case of thin objects like cloth as those may just end up traversing each other. One solution might be to set the stiffness factor k to a large enough value, however this causes the introduction of parasitical high frequencies and calls for very small integration steps [Baraff 1989]. To better account for the many different temporal scales involved in a scene with multiple impacts, Harmon et al. [2009] have investigated a new event-driven scheme based on asynchronous contact integration. Unlike classical penalty-based methods, their resulting scheme is robust without the need for excessive parameter tuning, yet at the price of a prominent computational cost.

Probably the most famous scheme that strictly ensures non-penetration in cloth simulation in a both robust and efficient manner is Bridson et al.'s method [2002]. Their hybrid algorithm combines repulsion forces with an "impact zone" failsafe first proposed by Provot [1997], rigidifying zones where recalcitrant collisions are detected through continuous collision detection. Harmon et al. [2008] have subsequently improved the failsafe by allowing some sliding motion of the incriminated vertices. Although the resulting Bridson/Harmon scheme has proved indeed capable of preventing self-penetration of cloth even in challenging scenarios, the treatment of Coulomb friction remains dubious as it gives no guarantee of consistency between the states of neighboring vertices. For instance, a high normal force for one vertex may impose inappropriately high frictional forces on other vertices within an impact zone.

As acknowledged by Harmon et al. [2008], a constraint-based method which would simultaneously and implicitly resolve impacts, contacts, and friction, would provide a more realistic cloth simulator. Constraint-based solvers are discussed below.

Constrained-based contact solvers. An alternative to locally compliant contacts is to consider a *rigid* contact model coupled to the exact (multivalued) Coulomb friction, namely the *Signorini-Coulomb* model, which enforces strict non-penetration while integrating the effects of frictional contact at the macroscopic scale. For stability and consistency purposes, such a nonsmooth model requires implicit

time-stepping schemes for being simulated numerically [Moreau 1988]. The resulting discrete frictional contact problem then takes the form of algebraic equations subject to *complementarity* set-valued constraints, which are better understood and manipulated using convex analysis tools, the latter having been mainly developed in the early 60's by Jean-Jacques Moreau and R. Tyrrel Rockafellar.

Interestingly, nonsmooth contact mechanics had been introduced to computer graphics by David Baraff as soon as in the early 90's (see e.g. [Baraff 1991, 1994]). However, it had not been followed up by the community at that time, in contrast to the computational mechanics community where many subsequent works were conducted in that area (see a recent review by Acary and Brogliato [2008]). Later on, such methods have slowly regained popularity in computer graphics in the context of rigid body dynamics, notably with the work of Kaufman and colleagues [2005; 2008].

The classical way to solve the discrete frictional contact problem is to first eliminate generalized velocities \mathbf{v} (primal variables) through the computation of the *Delassus* operator $\mathbf{W} = \mathbf{J}^T \mathbf{A}^{-1} \mathbf{J}$, where \mathbf{A} is the discrete inertia matrix and \mathbf{J} the gradient matrix relating local relative velocities at contact points \mathbf{u} to generalized velocities \mathbf{v} . Then, one solves the system in local contact forces \mathbf{r} (dual variables), and finally compute \mathbf{v} from \mathbf{r} using the linear momentum equation. In the case of a hair system, Daviet et al. [2011] introduced an efficient solver for this dual problem, able to take exact Coulomb friction into account. For cloth however, which features a large number of degrees of freedom interconnected together, such a method is computationally inefficient. Indeed, the matrix \mathbf{A} is costly to invert for large systems and its inverse is dense. To the best of our knowledge, Otaduy and colleagues [2009] were the first authors who attempted to alleviate the cost of the Delassus operator in the case of cloth dynamics. Their strategy consists in decomposing the computation of velocities from forces thanks to iterative relaxation. Injecting this decomposition in their one-step problem yields two nested loops, the inner loop boiling down to a sparse Linear Complementarity Problem (LCP) to solve. They further interlace frictional contact iterations with normal contact iteration using a pyramidal approximation of the Coulomb friction cone, which allows them to preserve a LCP structure for their inner problem. Modeling the frictional contact problem as an LCP is a classical and tempting approximation [Kaufman et al. 2008; Klarbring 1987; Stewart and Trinkle 1996], as it benefits from the large body of theoretical results and solvers available in the literature [Cottle et al. 2009]. However, as noted by Acary and Brogliato [2008], such an approximation yields some artificial anisotropic motion of the contacting objects. In our results, we also highlight the fact that linearized friction may cause some undesirable behavior of cloth at the macroscopic scale.

Towards a nodal contact solver. The constraint-based methods discussed above are generic, in the sense that they can be applied to any Lagrangian system. As such they are not optimized to exploit the *peculiar* structure of cloth dynamics. Despite the profusion of simulation models for cloth modelled with thin elastic sheets, it is indeed noteworthy that all of them share a similar *nodal* representation. That is, cloth is always modeled as a mesh whose vertices (3D positions) serve as degrees of freedom for the corresponding dynamical system.

In their seminal work, Baraff and Witkin [1998] have leveraged the nodal cloth representation for treating some particular bilateral constraints at nodes, such as a fixed or sliding constraint. In their paper the authors argue against Lagrange multiplier methods based on the KKT form, due the non-positiveness of the KKT matrix. The Schur complement form, which in contrast would allow them to recover positive-definiteness (see, e.g., [Blumentals et al. 2016, Section 2]), is not mentioned. Instead, an efficient filtering scheme, namely *mass modification*, is proposed. Thanks to that clever scheme, some arbitrary velocity change can be applied to any particle of the cloth; for instance, a particle can be enforced to slide on a certain direction that is prescribed in advance. Interestingly, this filtering operation boils down to an orthogonal projection of the dynamics [Ascher and Boxerman 2003], and turns out to be equivalent to a Lagrange multiplier method for bilateral constraints that can be written as orthogonal projection operators. This simple scheme is however not extended to a proper treatment of unilateral contact; instead, interpenetrations are resolved with artificial sticking constraints only. Lastly, the authors disregard friction in this paper.

Despite its limitations, this method was very inspiring to us, since it gave the premises of a constraint-based method tailored to nodal models, for which constraints are simpler to express compared to non-nodal models.

3 OVERVIEW

In this work, we consider implicit integration of frictional contact dynamics in cloth and thin sheets, using the exact Signorini–Coulomb model for frictional contact. Our work is based on the motivating observation that the primal formulation of the discrete frictional contact problem exactly preserves the block sparsity of the discrete inertia matrix \mathbf{A} , if the contacts only occur between mesh nodes and obstacles. Self-contact and layered contact require some additional extensions, but can still be handled without significantly degrading the sparsity. We present a new frictional contact solver that can efficiently and robustly solve nodal contact problems in this form.

Of course, contacts between bodies discretized as polygonal meshes can occur not just at nodes but also at edges and faces. In our approach, we simply relocate all such contact points to nodes, and use an adaptive refinement strategy to limit the error incurred by this process. Adaptive refinement ensures that regions near contact have enough mesh resolution to represent contacts solely through nodal constraints. This allows our method to handle general contact scenarios efficiently through the use of a nodal contact solver alone.

In practice, our adaptive solver proves to converge well in a vast majority of situations. In the rare events where it fails to converge to the requested precision and causes some interpenetrations, we remove them using the intersection contour minimization (ICM) algorithm of Volino and Magnenat-Thalmann [2006]. Our results show that our solver alone successfully handles all penetrations in 95% to 100% of the timesteps. In the worst case (very large friction coefficient), less than 5% of the timesteps need to apply ICM, so the overall accuracy of the simulation is still maintained.

Overall algorithm. At each time step, we carry out the following steps in order. First, we perform adaptive remeshing based on the algorithm of Narain et al. [2012], modified to ensure the presence

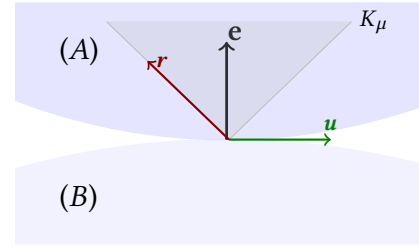


Fig. 2. A body (A) colliding a body (B), with contact normal \mathbf{e} . Relative velocity \mathbf{u} and contact force \mathbf{r} in the case when the contact is sliding.

of sufficiently many nodes in regions near contact (Section 5). Proximity and collision queries are carried out on the refined mesh, and the results are processed to obtain purely nodal contact constraints (Section 5.2). If multiple constraints are incident on a single mesh node, we make the constraints independent by creating duplicate nodes coupled with pin constraints (Section 6). Finally, the resulting constraints, together with the linear system for time-stepping the external and elastic forces, are passed into our nodal contact solver to compute the new state at the end of the time step (Sections 4 and 6). If some interpenetrations remain in the new state, we remove them using ICM. As the nodal solver for frictional contact is the core of our algorithm, we discuss it first below.

4 DERIVING A SIMPLE PRIMAL FORMULATION FOR FRICTIONAL CONTACT IN NODAL SYSTEMS

The idea of deriving a primal formulation for frictional contact tailored to nodal systems was briefly communicated in a prepublication [Daviet et al. 2015], in the case of a mesh with fixed resolution colliding a moving obstacle. Here we largely extend these preliminary results by providing an extension to self-contact as well as a new and faster primal solver.

4.1 Background

We consider a simulation with one or more cloth sheets, each discretized as triangle meshes. For simplicity, we consider all the m vertices of the cloth mesh(es) as a single system, and define a vector of vertex positions $\mathbf{x} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_m^\top]^\top \in \mathbb{R}^{3m}$ and velocities $\mathbf{v} = d\mathbf{x}/dt = [\mathbf{v}_1^\top, \dots, \mathbf{v}_m^\top]^\top \in \mathbb{R}^{3m}$. Without contact, the system evolves under the equation of motion

$$M \frac{d\mathbf{v}}{dt} = \mathbf{f}(t, \mathbf{x}, \mathbf{v}), \quad (1)$$

where M is the diagonal mass matrix, and \mathbf{f} includes both external forces such as gravity, and internal forces such as elasticity and damping.

If the system is subject to contacts $i = 1, \dots, n$, we can define for each contact pair the relative velocity of the two contacting points, $\mathbf{u}^i \in \mathbb{R}^3$, and the force in the local contact basis, $\mathbf{r}^i \in \mathbb{R}^3$, as shown in Figure 2. The relative velocity and contact force are subject to the Signorini–Coulomb law with friction coefficient $\mu^i \geq 0$, which we denote in an abstract manner as $(\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu^i}$,

$$\begin{aligned}
 (\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i} &\iff \\
 \left\{ \begin{array}{l} \mathbf{r}^i = 0 \text{ and } u_N^i \geq 0, \\ \text{or } \|\mathbf{r}_T^i\| \leq \mu^i r_N^i \text{ and } \mathbf{u}^i = 0, \\ \text{or } \|\mathbf{r}_T^i\| = \mu^i r_N^i, u_N^i = 0, \text{ and } \exists \alpha^i > 0 : \mathbf{r}_T^i = -\alpha^i \mathbf{u}_T^i. \end{array} \right. & \begin{array}{l} \text{(taking off)} \\ \text{(sticking)} \\ \text{(sliding)} \end{array}
 \end{aligned}$$

Here we have denoted the normal and tangential components of any vector $\mathbf{y} \in \mathbb{R}^3$ by $y_N = \mathbf{y} \cdot \mathbf{e} \in \mathbb{R}$ and $\mathbf{y}_T = \mathbf{y} - y_N \mathbf{e} \in \mathbb{R}^3$ respectively, \mathbf{e} being the normal vector at contact. (By convention, scalars are denoted by plain letters while vectors of size greater than one are denoted by bold letters).

The Signorini–Coulomb law is presented in detail in [Daviet et al. 2011], along with different equivalent formulations. We illustrate the case of sliding friction in Figure 2: the friction force lies on the boundary of the cone, and the relative velocity lies on the surface of contact while remaining aligned with the tangential component of the force, but in the opposite direction. In the sticking case, the relative velocity vanishes and the frictional contact force can lie anywhere inside the friction cone.

To simplify notation, we now collect all the vectors \mathbf{u}^i and \mathbf{r}^i into $\mathbf{u} \in \mathbb{R}^{3n}$ and $\mathbf{r} \in \mathbb{R}^{3n}$ respectively, and define the deformation gradient matrix at contact $J = \frac{\partial \mathbf{u}}{\partial \mathbf{v}}$. The relative velocities \mathbf{u} and contact forces \mathbf{r} can be related to the generalized velocities \mathbf{v} and generalized contact forces \mathbf{f}^c as

$$\mathbf{u} = J\mathbf{v} + \mathbf{u}_f, \quad (2)$$

$$\mathbf{f}^c = J^T \mathbf{r}, \quad (3)$$

where \mathbf{u}_f is the value of \mathbf{u} when $\mathbf{v} = 0$, for example, in the presence of moving obstacles which do not appear in \mathbf{v} . Later, we will see that \mathbf{u}_f also helps encode position-level non-penetration constraints in discrete time stepping. The nonsmooth system including frictional contact forces \mathbf{r} can be written as

$$\left\{ \begin{array}{l} M \frac{d\mathbf{v}}{dt} = \mathbf{f}(t, \mathbf{x}, \mathbf{v}) + \underbrace{J^T \mathbf{r}}_{\mathbf{f}^c} \\ \mathbf{u} = J\mathbf{v} + \mathbf{u}_f \\ \forall i = 1 \dots n, (\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i}. \end{array} \right. \quad (4)$$

The notation $\frac{d\mathbf{v}}{dt}$ is actually misused as the inclusion of contacts implies that we might observe jumps in the generalized velocities \mathbf{v} . To circumvent this difficulty, we follow Moreau’s time-stepping scheme and integrate the equations over an arbitrary timestep dt [Moreau 1988]. We thus get a formulation over mechanical impulses instead of forces, and a velocity jump instead of an acceleration.

From now on, upright letters will stand for *discrete* variables. Let us set $\mathbf{r} := \int_{\delta t} \mathbf{r} dt$, and let \mathbf{v} and \mathbf{u} be the discrete approximations of the generalized velocity and the relative velocity, respectively, at the end of the timestep. The constraint-free dynamics (1) is discretized using a θ -scheme and can be written at a given timestep as $\mathbf{A}\mathbf{v} + \mathbf{b} = 0$. We further assume that J and \mathbf{u}_f are almost constant during the timestep, and approximate them with J and \mathbf{u}_f , respectively. The discretized velocity/impulse formulation of (4), with unknowns $(\mathbf{v}, \mathbf{u}, \mathbf{r}) \in \mathbb{R}^{3m} \times \mathbb{R}^{3n} \times \mathbb{R}^{3n}$, then reads

$$\left\{ \begin{array}{l} \mathbf{A}\mathbf{v} + \mathbf{b} = J^T \mathbf{r} \\ \mathbf{u} = J\mathbf{v} + \mathbf{u}_f \\ \forall i = 1 \dots n, (\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i}. \end{array} \right. \quad (5)$$

Problem (5), referred to as the *discrete frictional contact problem (DFCP)* is the core problem we wish to solve here. A common way to tackle (5) is to eliminate \mathbf{v} by introducing the *Delassus operator* $\mathbf{W} := J\mathbf{A}^{-1}J^T$ and by setting $\mathbf{d} := \mathbf{u}_f - J\mathbf{A}^{-1}\mathbf{b}$, leading to the following reduced *DFCP*,

$$\left\{ \begin{array}{l} \mathbf{W}\mathbf{r} + \mathbf{d} = \mathbf{u} \\ \forall i = 1 \dots n, (\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i}, \end{array} \right. \quad (6)$$

which may again be compacted as a force (or impulse) based problem, namely a *dual* formulation,

$$\forall i = 1 \dots n, (\mathbf{r}^i, (\mathbf{W}\mathbf{r})^i + \mathbf{d}^i) \in C_{\mu_i}. \quad (7)$$

For cloth however, where primal variables are 3D velocities, such a method is computationally inefficient. Indeed, the matrix \mathbf{A} is costly to invert for large systems and its inverse is dense. To make the problem tractable, Otaduy and colleagues [2009] have proposed an iterative relaxation algorithm where each inner problem retrieves sparsity. We in turn depart from the dual formulation (7) and totally avoid computing the Delassus operator. Another major difference between our method and Otaduy et al.’s is that we consider the true second-order friction cone instead of a polyhedral approximation. As a result, the dynamics combined to the Signorini–Coulomb law C_{μ_i} cannot be cast in the form of an LCP, but reads as a nonlinear and asymmetric conical complementarity.

4.2 A new primal formulation tailored to nodal systems

We now introduce our new, primal formulation of the *DFCP*, albeit in a restricted set of cases for the moment. Contacts are assumed to occur exactly at mesh vertices, and no two contacts share a vertex. Under these assumptions, each contact pair involves either one or two cloth vertices, depending on whether it is an instance of body-cloth contact or cloth self-contact. Further, each cloth vertex is associated with at most one contact pair. We will remove these restrictions in later sections.

Let us first consider the case where there are only body-cloth contacts. A key observation is that each nonzero block of J related to a vertex i in contact is simply a rotation matrix corresponding to the inverse (transpose) of the local contact basis E_i (we have $\mathbf{u}^i = E_i^T \mathbf{v}^i$). Our idea is then to build the square block-diagonal matrix G with G_{ii} equal to E_i^T if vertex i is in contact, and to the 3×3 identity matrix otherwise. G can thus be inverted trivially as $G^{-1} = G^T$. Augmenting \mathbf{u} , \mathbf{u}_f , and \mathbf{r} so that their size matches that of \mathbf{v} ($m \geq n$), we can write (5) as

$$\left\{ \begin{array}{l} \mathbf{G}\mathbf{A}\mathbf{G}^T \mathbf{u} + \mathbf{G}\mathbf{b} - \mathbf{G}\mathbf{A}\mathbf{G}^T \mathbf{u}_f = \mathbf{r} \\ \mathbf{v} + \mathbf{G}^T \mathbf{u}_f = \mathbf{G}^T \mathbf{u} \\ (\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i} \quad \text{if } i \text{ in contact} \quad \text{and } \mathbf{r}^i = 0 \text{ otherwise.} \end{array} \right.$$

Denoting $\widehat{\mathbf{W}} = \mathbf{G}\mathbf{A}\mathbf{G}^T$ and $\widehat{\mathbf{d}} = \mathbf{G}\mathbf{b} - \mathbf{G}\mathbf{A}\mathbf{G}^T \mathbf{u}_f$, we obtain a system that is very close to the dual formulation (6),

$$\left\{ \begin{array}{l} \widehat{\mathbf{W}}\mathbf{u} + \widehat{\mathbf{d}} = \mathbf{r} \\ \forall i = 1 \dots n, (\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i}, \end{array} \right. \quad (8)$$

except that \mathbf{u} and \mathbf{r} have reversed roles in the linear equation. Another notable difference is that the new operator $\widehat{\mathbf{W}}$ involves the matrix \mathbf{A} directly, and not its inverse. This nice feature will be key in designing an efficient solver (see Section 4.3).

Self-contact can be included in the same framework with little effort. Let us first notice that contact between two cloth vertices j and k is treated by constraining their *relative velocity*, $E_{jk}^T (\mathbf{v}^j - \mathbf{v}^k)$, which is stored by convention in the variable \mathbf{u}^j . The rotation matrix E_{jk} is the local contact basis defined by the surface of contact between the two vertices j and k (with the normal pointing from k to j). The linear relationship above is not enough to build an orthogonal system. We thus complete it with a second linear relationship, orthogonal to the first one, which is encoded as $\mathbf{u}^k = E_{jk}^T (\mathbf{v}^j + \mathbf{v}^k)$. Unlike \mathbf{u}^j , which is imposed the Coulomb constraint $(\mathbf{r}^j, \mathbf{u}^j) \in C_{\mu_j}$, the quantity \mathbf{u}^k is to be left unconstrained, that is, we have to set $\mathbf{r}^k = 0$. After normalizing the two equations above, we obtain a rotation matrix \mathbf{G} dedicated to self-contact (with size doubled compared to body-cloth contact), whose associated blocks are defined as

$$\begin{bmatrix} \mathbf{G}_{jj} & \mathbf{G}_{jk} \\ \mathbf{G}_{kj} & \mathbf{G}_{kk} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} E_{jk}^T & -E_{jk}^T \\ E_{jk}^T & E_{jk}^T \end{bmatrix}. \quad (9)$$

It is noteworthy that the introduced factor $1/\sqrt{2}$ has no influence on the friction law imposed on \mathbf{u}^j , since conical complementarity is insensitive to positive scaling.

Finally, we note that in the discrete-time case, the velocity condition $u_N^i \geq 0$ encoded by the Coulomb law C_{μ_i} is not equivalent to a non-penetration constraint on vertex positions. For example, if a vertex is initially at a distance z^i away from an obstacle, non-penetration at the end of the time step requires only the weaker condition $u_N^i \geq -z^i/\delta t$; further, if a nonzero contact thickness h is desired, we require $u_N^i \geq (h - z^i)/\delta t$ instead. An analogous condition holds for self-contact as well. To account for this, for each contact pair we compute the separation z^i in the normal direction at the beginning of the time step, and add a term $(z^i - h)/\delta t$ to the normal component of \mathbf{u}_T^i .

So far, we have defined a simple formulation for frictional contact in cloth, subject to the limitations that contacts may only occur at cloth vertices, and any two contacts must not share vertices. In the following section, we show how the above formulation can be solved efficiently. Afterwards, we will show how the two key limitations of the formulation can be lifted, while retaining the same efficient solver, in Sections 5 and 6.

4.3 Fast and robust primal solver

Solving (8) using the dual solver of Daviet et al. [2011], which is dedicated to solving (6), is not immediate, at least in the non-trivial case where $\mu^i \neq 0$. Indeed, in this case, since the C_{μ_i} law is not symmetric w.r.t. the contact force \mathbf{r}^i and the relative velocity \mathbf{u}^i , the two variables cannot be swapped in the frictional contact law. Hence they are not interchangeable in the **DFCP**.

In [Daviet et al. 2015], the frictional contact law is *convexified* by applying the change of variable initially proposed by De Saxcé and Feng [1998],

$$\tilde{\mathbf{u}}^i = \mathbf{u}^i + \mathbf{s}^i(\mathbf{u}^i) \quad \text{with} \quad \mathbf{s}^i(\mathbf{u}^i) = \mu^i \|\mathbf{u}_T^i\| \mathbf{e}^i \quad (10)$$

where \mathbf{u}_T^i is the tangential relative velocity at contact i . The modified relative velocity $\tilde{\mathbf{u}}^i$ is thus orthogonal to \mathbf{r}^i , and belongs to the *dual* Coulomb cone. For a fixed value of $\mathbf{s} = \{\mathbf{s}^i\}_i$, (8) can be

identified as the KKT conditions of a convex quadratic optimization problem [Cadoux 2009]. Each convex problem is solved as a subcase of Daviet et al. [2011]’s method, and the full (non-convex) problem is resolved by iteratively updating \mathbf{s} using a fixed point algorithm.

To improve efficiency even more, we propose here to avoid this global fixed-point loop. Instead, we start back from the original Gauss–Seidel algorithm of Daviet et al. [2011], targeted at the full (non-convex) **DFCP** in its dual form. In this method, contacts are resolved individually in an iterative manner, relying on a robust local solver for the dual one-contact Coulomb friction problem. As our main variable now is the relative velocity \mathbf{u} instead of the force \mathbf{r} , we must construct a local solver dedicated to the primal one-contact Coulomb friction problem,

$$\begin{cases} \mathbf{r}^i = \widehat{\mathbf{W}}_{ii} \mathbf{u}^i + \bar{\mathbf{d}}^i, & \bar{\mathbf{d}}^i = \widehat{\mathbf{d}}^i + \sum_{i \neq j} \widehat{\mathbf{W}}_{ij} \mathbf{u}^j \\ (\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i}. \end{cases} \quad (11)$$

To do this, we can simply *adapt* the Fischer–Burmeister based solver proposed by Daviet et al. [2011], as explained below.

We omit the trivial case where $\mu^i = 0$ (the **DFCP** boiling down to an **LCP**), and consider $\mu^i > 0$. Following Daviet et al. [2011], we first build a *symmetric* frictional contact law in $(\hat{\mathbf{r}}^i, \hat{\mathbf{u}}^i)$ by applying a new change of variable,

$$\hat{\mathbf{r}}^i = (\mu^i r_N^i, \mathbf{r}_T^i) \quad \text{and} \quad \hat{\mathbf{u}}^i = (\tilde{u}_N^i, \mu^i \tilde{\mathbf{u}}_T^i), \quad (12)$$

which maps \mathbf{r}^i and \mathbf{u}^i to the self-dual cone (i.e., the cone corresponding to a frictional coefficient $\mu = 1$) while preserving their orthogonality. Let f^{FB} be the so-called *modified Fischer–Burmeister* function, first introduced by Fukushima et al. [2002], which extends the classical Fischer–Burmeister function (applicable to real complementarity problems) to conical complementarity problems. Then, as $(\mathbf{r}^i, \mathbf{u}^i) \in C_{\mu_i}$ if and only if $f^{FB}(\hat{\mathbf{u}}^i, \hat{\mathbf{r}}^i) = 0$, we can directly use a Newton algorithm to find the roots of our new function $g^{FB,i} : \mathbf{u}^i \mapsto f^{FB}(\hat{\mathbf{u}}^i(\mathbf{u}^i), \hat{\mathbf{r}}^i(\mathbf{u}^i))$, where $\hat{\mathbf{u}}^i(\mathbf{u}^i)$ and $\hat{\mathbf{r}}^i(\mathbf{u}^i)$ are deduced by combining (12), (11) and (10). The only difficult quantities to compute for applying the Newton algorithm are the derivatives of the f^{FB} function, which are provided in [Daviet et al. 2011]. Then derivatives of $g^{FB,i}$ are simply obtained by using the chain rule.

Daviet et al. [2011] advocate using an hybrid solver for the dual one-contact problem, which combines an analytical solver with the above Newton solver. While a similar quartic polynomial could be derived for the primal case, we found the pure Newton method to converge well-enough for our cloth simulations – which could hint at the primal matrix $\widehat{\mathbf{W}}$ being better conditioned than the Delassus operator. The remainder of the Gauss–Seidel algorithm can be left mostly unchanged, but for the sake of self-containment we recall it in Algorithm 1. Note that to be meaningful, the sleeping criterion, which intends to freeze steady or small contacts forces for a few iterations, must still be computed on the local force and not on the local relative velocity.

In practice, since the primal operator $\widehat{\mathbf{W}}$ is easy to assemble and sparse, our primal strategy turns out to be orders of magnitude faster compared to the dual strategy. For instance, for a dress example featuring 1,100 contact points, we have observed a speed-up by a factor of 1,200 (see [Daviet 2016, Section 4.3.2]). This computational gain allowed us to address challenging cloth problems in reasonable timings, as shown in Section 7.

ALGORITHM 1: Primal Gauss–Seidel algorithm with sleeping heuristics

```

// Initialize v and u with unconstrained velocity
v ← -A-1d;
u0 ← Gv - uf;
for k ← 1 to maxIters do // GS iteration
    for i ← 1 to m do // Loop over vertices
        if SkipTab [i] > 0 then // Sleeping contact
            SkipTab [i] ← SkipTab [i] - 1;
            continue;
        end
        di ← di + ∑i≠j Wijuj;
        if not in contact then
            ui ← -Wii-1di; // Solve ri = 0
        else
            ui,prev ← ui;
            Update ui using Newton's method applied to gFB,i(ui) = 0;
            if did not converge then ui ← ½ (ui + ui,prev) // Dampen;
        end
        ri ← Wiiui + di;
        if ||ri||2 < εsmall or ||ri - ri,prev||2 < εconverged then
            SkipTab [i] ← nSkip; // Put contact to sleep
        end
    end
    if k mod nEvalEvery = 0 then
        // Evaluate residual
        r ← Wu + d;
        if maxi ||fFB(ui, ri)||2 < tol then break;
    end
end
// Compute global velocity
v ← GT(u - uf);
    
```

5 ADAPTIVE REFINEMENT

The nodal formulation above requires all contacts to occur at mesh nodes. However, for surfaces represented as polygon meshes, contacts can also occur at points that lie on mesh edges or faces, and naively ignoring them can lead to interpenetrations (Figure 3). In this section, we discuss our approach to reconciling these conflicting requirements.

We start from the following key idea: If the mesh is refined appropriately relative to the contact geometry, then nodal constraints are sufficient to prevent interpenetrations. The amount of refinement need not be excessive; for example, for a cloth sheet in contact with a flat plane, no refinement is necessary. For a more representative test problem, consider a triangulated cloth sheet whose vertices are constrained to remain at least a distance h away from a sphere of radius r . It can be shown that no face or edge of the cloth mesh can penetrate the sphere as long as the edge lengths are $O(\sqrt{rh})$. Thus, for smoothly curved objects and moderate collision distances, the necessary mesh resolution is typically quite acceptable.

Motivated by this idea, we address the issue of general collision constraints by ensuring that the mesh is refined adequately relative to the curvature of the contact surface. We do so using a “proximity

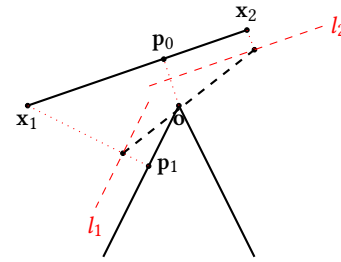


Fig. 3. A 2D demonstration of contact issues when the mesh is too coarse. With only nodal constraints, non-penetration between the edge x_1x_2 and the sharp corner at o cannot be guaranteed, even if the edge-corner constraint l_2 is imposed at one of the vertices. However, if the edge was refined, this intersection could be prevented.

metric” that generalizes the obstacle distance heuristic originally proposed by Narain et al. [2012].

A note on edge and face constraints. As an alternative, one could directly incorporate the edges and faces involved in contact constraints directly into the blocks of the matrix G . Similar to the self-contact formulation, this only requires constructing an orthonormal basis for the degrees of freedom involved in the constraint. However, with this approach, each vertex will be involved in constraints arising from all its adjacent edges and faces, which must be resolved by the pin constraint algorithm described in Section 6. For dense contact scenarios, this results in excessive vertex duplication and a greatly increased number of degrees of freedom, and we found that the computation time was far from competitive.

5.1 Dynamic remeshing and the proximity metric

For adapting the resolution of the mesh, we build on the remeshing scheme of Narain et al. [2012]. This method dynamically refines the mesh where necessary to resolve detail in the cloth, and coarsens it in smooth flat areas. The algorithm has two steps. First, we compute a *sizing field* \mathbf{M} , which is a metric tensor field that encodes the desired resolution of the mesh. This sizing field is determined using various refinement criteria which measure the current or anticipated future detail in the cloth, such as surface curvature, velocity gradient, and in-plane compression. Each criterion defines its own sizing metric $\mathbf{M}_1, \mathbf{M}_2, \dots$, which are then combined into the final sizing field $\mathbf{M} = \text{combine}(\mathbf{M}_1, \mathbf{M}_2, \dots)$ via the procedure described by Narain et al. [2013]. Using the sizing field, we define the *size* s_{ij} of an edge between vertices i and j as

$$s_{ij}^2 = \bar{\mathbf{x}}_{ij}^T \mathbf{M} \bar{\mathbf{x}}_{ij}, \quad (13)$$

where $\bar{\mathbf{x}}_{ij}$ is the vector from vertex i to vertex j in the underlying reference space. (Reference-space coordinates were denoted \mathbf{u} by Narain et al. [2013, 2012]; we switch to $\bar{\mathbf{x}}$ since we are using \mathbf{u} for the relative contact velocity.) In the second step of the algorithm, we perform local remeshing operations until all mesh edges satisfy $s_{ij} \leq 1$, and the mesh is close to Delaunay in the normalized space defined by the sizing metric [Bossen and Heckbert 1996].

To add contact-based refinement to this approach, we only need to define an additional refinement metric \mathbf{M}_{prox} based on proximity

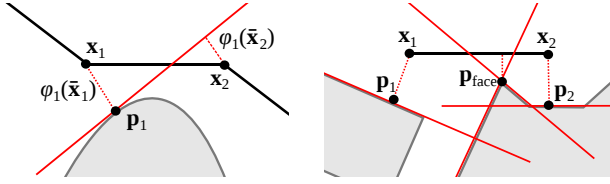


Fig. 4. (Left) The geometry defining the obstacle metric of Narain et al. [2012]. As \mathbf{x}_1 and \mathbf{x}_2 are on the different side of the tangential plane, this edge needs to be split. (Right) An illustration of the tangent planes used in defining our improved proximity metric.

to contacts, such that $\bar{\mathbf{x}}_{ij}^\top \mathbf{M}_{\text{prox}} \bar{\mathbf{x}}_{ij} \leq 1$ only if edge ij is refined enough to adequately resolve the contact. Narain et al. [2012] defined an “obstacle metric” \mathbf{M}_{obs} with a similar motivation, based on the requirement that an edge must not be long enough to cross nearby tangent planes of an obstacle, as illustrated in Figure 4 (left). Formally, consider a mesh triangle with vertices $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$. For each vertex i , we find its closest point \mathbf{p}_i among all obstacles, and the associated tangent plane, i.e. the plane through \mathbf{p}_i normal to $\mathbf{x}_i - \mathbf{p}_i$. Let φ_i be the signed distance function of this tangent plane, viewed as a function of reference coordinates $\bar{\mathbf{x}}$ on the cloth surface. The obstacle-related sizing tensor for vertex i is defined as

$$\mathbf{M}_{\text{obs}}^i = \frac{\nabla \varphi_i \nabla \varphi_i^\top}{\varphi_i(\bar{\mathbf{x}}_i)^2}, \quad (14)$$

and compute \mathbf{M}_{obs} for the face as the average of the three such tensors obtained from its vertices.

This definition has a number of attractive properties. First, it guarantees that for a convex obstacle, any face or edge that satisfies the associated metric cannot intersect the obstacle. Second, under this metric, the mesh resolution increases smoothly with decreasing distance to the obstacle, preserving the temporal coherence of the animation. Third, it only requires finding, for each vertex, the nearest point on the obstacle and its tangent plane, and so can be implemented easily for various obstacle representations, such as signed distance fields (trivially) or polygon meshes (via BVH-based proximity queries). However, it has the drawback that obstacle proximities are only evaluated at mesh vertices, which can lead to missed collisions with nonconvex shapes. We also observe this definition to lead to unnecessary refinement when the cloth and obstacle surfaces are not parallel.

We define an improved *proximity metric* \mathbf{M}_{prox} that retains the benefits of the original obstacle metric while overcoming its drawbacks. Consider again a single mesh triangle. We perform proximity queries to find the closest point for each vertex, for each edge, and for the face itself. For each obtained point \mathbf{p} , we define a *collection* of tangent planes: if \mathbf{p} lies on a face, we take the plane of the face itself; if on an edge or a vertex, we take the planes of all the adjacent faces. This process gives us a set of tangent planes characterizing potential collisions for this face, as shown in Figure 4 (right). Since the tangent planes are no longer in one-to-one correspondence with the face vertices, we use the maximum vertex distance in the denominator; this also allows us to avoid excessive refinement. Finally, we take the the signed distance functions $\varphi_1, \varphi_2, \dots$ of all the tangent

planes, and define the proximity metric as

$$\mathbf{M}_{\text{obs}}^j = \frac{\nabla \varphi_j \nabla \varphi_j^\top}{\max_{i=1,2,3} \varphi_j(\bar{\mathbf{x}}_i)^2}, \quad (15)$$

$$\mathbf{M}_{\text{prox}} = \text{combine}(\mathbf{M}_{\text{obs}}^1, \mathbf{M}_{\text{obs}}^2, \dots), \quad (16)$$

again using the combination procedure of Narain et al. [2013]. To account for self-contact and contact with other cloth sheets, we include both obstacle and cloth surfaces when looking for nearest points to a mesh primitive. When including self-contact, care must be taken not to include adjacent mesh primitives in the proximity search; we prune these extraneous detections by requiring that the ratio of world distance to material-space distance be less than a threshold, which in our implementation was 0.1.

5.2 Collision detection and constraint construction

After remeshing, we perform proximity and collision tests to obtain the contact constraints that will be input to the nodal contact solver.

In the proximity tests, we find all pairs of elements whose distance is less than a detection threshold (typically 1.5 times the contact thickness h described in Section 4.2). For additional robustness in the presence of fast motions, we also perform continuous collision detection: here we extrapolate the future positions of all vertices using their current velocities, and collect contact pairs for all primitives that are predicted to intersect over the time step. Both proximity and collision tests are computed efficiently using a bounding volume hierarchy.

Taking the union of the results from both steps, we obtain a collection of contact pairs, which may involve points in the interior of faces and edges. We relocate every such contact point to the nearest vertex on the same primitive, yielding a set of purely *nodal* contact constraints. Since our remeshing step has already refined mesh regions near contact, the amount of relocation required is typically very small. We also perform contact pruning to remove nearly redundant constraints: for any two contacts that involve the same set of cloth vertices and whose normals are within 10° of each other, we retain only the one with the higher priority, namely the pair with the deepest predicted penetration at the end of the time step. The effect of contact pruning is quantified in Section 7.1. Finally, to handle multiple constraints acting on a cloth vertex, we apply the pin constraint algorithm described in the following section.

6 PIN CONSTRAINTS

Whether due to layered contact or relocation of contacts to vertices, it is inevitable that some vertices may be subject to multiple contact constraints. We incorporate such constraints into the nodal solver by simply duplicating any vertex with $p > 1$ contacts into p copies, each associated with one contact, and adding a “pin constraint” that requires them to be coincident.

For ease of exposition, we first consider the case of duplicating a single vertex. We start from the discretized constraint-free dynamics equations, $\mathbf{A}\mathbf{v} + \mathbf{b} = 0$. Suppose we wish to split the i th vertex into p vertices connected by the constraint $\mathbf{v}_{i,1} = \mathbf{v}_{i,2} = \dots = \mathbf{v}_{i,p}$, and extend \mathbf{A} and \mathbf{b} such that the solution is merely replicated, that is, all $\mathbf{v}_{i,1}, \dots, \mathbf{v}_{i,p}$ in the extended solution equal \mathbf{v}^i in the original solution if no additional constraints are imposed. This is readily

achieved by dividing the blocks associated with i in \mathbf{A} and \mathbf{b} into p equal parts, as follows.

After duplicating the vertex i , the concatenated velocity vector is now $\tilde{\mathbf{v}} = [\mathbf{v}_1^\top, \dots, \mathbf{v}_{i-1}^\top, \mathbf{v}_{i,1}^\top, \dots, \mathbf{v}_{i,p}^\top, \mathbf{v}_{i+1}^\top, \dots, \mathbf{v}_m^\top]^\top \in \mathbb{R}^{3(m-1+p)}$. Let I denote the indices associated with $\mathbf{v}_{i,1}, \dots, \mathbf{v}_{i,p}$, and let j range over the remaining vertices $j \neq i$. The pin constraint $\mathbf{v}_{i,1} = \mathbf{v}_{i,2} = \dots = \mathbf{v}_{i,p}$ is equivalent to $\mathbf{C}\tilde{\mathbf{v}} = \mathbf{0}$, where

$$\begin{aligned} \mathbf{C}_{II} &= \mathbb{I}_{3p} - \frac{1}{p} \begin{bmatrix} \mathbb{I}_3 & \cdots & \mathbb{I}_3 \\ \vdots & \ddots & \vdots \\ \mathbb{I}_3 & \cdots & \mathbb{I}_3 \end{bmatrix}, & \mathbf{C}_{Ij} &= \mathbf{0}, \\ \mathbf{C}_{jI} &= \mathbf{0}, & \mathbf{C}_{jj} &= \mathbf{0}. \end{aligned}$$

We denote the feasible set of the constraint as $\mathcal{P} = \text{Nul } \mathbf{C} = \{\tilde{\mathbf{v}} : \mathbf{v}_{i,1} = \dots = \mathbf{v}_{i,p}\}$. Observe that \mathbf{C} is an orthogonal projection, so $(\mathbb{I} - \mathbf{C}^\top)\mathbf{y} \in \mathcal{P}$ for any vector \mathbf{y} .

Duplicating the corresponding blocks of \mathbf{A} and \mathbf{b} , we replace the i th diagonal block of \mathbf{A} with a block diagonal matrix containing p copies of $\frac{1}{p}\mathbf{A}_{ii}$. The off-diagonal blocks of \mathbf{A} and the i th block of \mathbf{b} are similarly split into p copies each scaled by $\frac{1}{p}$, giving

$$\begin{aligned} \hat{\mathbf{A}}_{II} &= \frac{1}{p} \begin{bmatrix} \mathbf{A}_{ii} & & \\ & \ddots & \\ & & \mathbf{A}_{ii} \end{bmatrix}, & \hat{\mathbf{A}}_{Ij} &= \frac{1}{p} \begin{bmatrix} \mathbf{A}_{ij} \\ \vdots \\ \mathbf{A}_{ij} \end{bmatrix}, \\ \hat{\mathbf{A}}_{jI} &= \frac{1}{p} [\mathbf{A}_{ji} \quad \cdots \quad \mathbf{A}_{ji}], & \hat{\mathbf{A}}_{jj} &= \mathbf{A}_{jj}, \\ \tilde{\mathbf{b}}_I &= \frac{1}{p} [\mathbf{b}_i^\top \quad \cdots \quad \mathbf{b}_i^\top]^\top, & \tilde{\mathbf{b}}_j &= \mathbf{b}_j. \end{aligned}$$

This duplication procedure can be applied recursively to deal with multiple duplicated vertices. Ultimately, we obtain an enlarged system with a velocity vector $\tilde{\mathbf{v}} \in \mathbb{R}^{3d}$, where d is the total number of vertices after duplication, and with a constraint matrix \mathbf{C} containing a nonzero block for every duplicated vertex. We continue to denote the extended matrix and force vector by $\hat{\mathbf{A}}$ and $\tilde{\mathbf{b}}$ respectively. Under the pin constraints, the dynamics equation becomes

$$\hat{\mathbf{A}}\tilde{\mathbf{v}} + \tilde{\mathbf{b}} = \mathbf{C}^\top \boldsymbol{\lambda}, \quad (17)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{3d}$ is the Lagrange multiplier vector associated with the pin constraint forces. The components of $\boldsymbol{\lambda}$ for non-duplicated vertices have no influence and are taken to be zero.

Putting back the frictional contact constraints (now i ranges from 1 to d), we obtain

$$\begin{cases} \hat{\mathbf{A}}\tilde{\mathbf{v}} + \tilde{\mathbf{b}} = \mathbf{G}^\top \mathbf{r} + \mathbf{C}^\top \boldsymbol{\lambda} \\ \mathbf{C}\tilde{\mathbf{v}} = \mathbf{0}; \\ \mathbf{u} = \mathbf{G}\tilde{\mathbf{v}} + \mathbf{u}_f; \\ (\mathbf{u}^i, \mathbf{r}^i) \in C_{\mu_i} \text{ if } i \text{ is a contact point} \\ \mathbf{r}^i = \mathbf{0} \text{ if } i \text{ is not a contact point.} \end{cases}$$

By construction of $\hat{\mathbf{A}}$ and $\tilde{\mathbf{b}}$, if the right-hand side satisfies the constraint, then so will the velocity vector $\tilde{\mathbf{v}}$. To satisfy the pin constraint, it is therefore sufficient to ensure that $\mathbf{G}^\top \mathbf{r} + \mathbf{C}^\top \boldsymbol{\lambda} \in \mathcal{P}$. This can be achieved by choosing $\boldsymbol{\lambda} = -\mathbf{G}^\top \mathbf{r}$; then we have $\mathbf{G}^\top \mathbf{r} + \mathbf{C}^\top \boldsymbol{\lambda} = (\mathbb{I} - \mathbf{C}^\top)\mathbf{G}^\top \mathbf{r} \in \mathcal{P}$, because \mathbf{C} is an orthogonal projection.

Eliminating $\tilde{\mathbf{v}}$ and $\boldsymbol{\lambda}$ via the equations $\mathbf{u} = \mathbf{G}\tilde{\mathbf{v}} + \mathbf{u}_f$ and $\boldsymbol{\lambda} = -\mathbf{G}^\top \mathbf{r}$, the system to be solved becomes

$$\begin{cases} \mathbf{r} - \mathbf{GCG}^\top \mathbf{r} = \hat{\mathbf{G}}\hat{\mathbf{A}}\mathbf{G}\mathbf{u} + \hat{\mathbf{d}} \\ (\mathbf{u}^i, \mathbf{r}^i) \in C_{\mu_i} \text{ if } i \text{ is a contact point} \\ \mathbf{r}^i = \mathbf{0} \text{ if } i \text{ is not a contact point,} \end{cases}$$

with $\hat{\mathbf{d}} = \hat{\mathbf{G}}\hat{\mathbf{A}}\mathbf{G}^\top \mathbf{u}_f - \hat{\mathbf{G}}\tilde{\mathbf{b}}$ as before.

Solution algorithm. The new force-dependent term in the right-hand side, $\mathbf{GCG}^\top \mathbf{r}$, is dealt with using operator splitting,

$$\mathbf{r} = \hat{\mathbf{G}}\hat{\mathbf{A}}\mathbf{G}^\top \mathbf{u} + \underbrace{\mathbf{GCG}^\top \mathbf{r}}_{\mathbf{c}} + \hat{\mathbf{d}}.$$

This approach is integrated in the main contact-solving Gauss-Seidel loop of Algorithm 1 as follows:

- Initialize $\mathbf{c} \leftarrow \hat{\mathbf{d}}$;
- For each Gauss-Seidel iteration:
 - Sequentially solve each vertex as in the case without pin constraints but using \mathbf{c} instead of $\hat{\mathbf{d}}$, yielding new values for \mathbf{u} and \mathbf{r} ;
 - Update $\mathbf{c} \leftarrow \hat{\mathbf{d}} + \mathbf{GCG}^\top \mathbf{r}$.

Note that this algorithm does not require any explicit linear-system solve for the pin constraints; only one additional sparse matrix-vector product is performed at each iteration.

7 EVALUATION AND RESULTS

To illustrate the benefit of our approach, we first provide some challenging benchmarks and compare the quality of our results to that of Harmon et al. [2008] and Otaduy et al. [2009], based on our own implementation of these two methods. We also compare our new proximity metric with the original obstacle metric [Narain et al. 2012]. Then we present our results on full size garments, which showcase the practicality of our approach. Performance and convergence results are then analysed before discussing the limitations of our method. All our simulation results are included in our accompanying video published as supplementary material.

7.1 Benchmarks and comparisons with existing work

To begin with, we validate our method by comparison with the analytical solution of a simple example. We set up a sheet of cloth parallel to an inclined plane, falling onto the plane and sliding down on it, as shown in Figure 5 (top). Since the cloth is parallel to the plane during the entire motion of falling and sliding, the velocities and displacements of all vertices should be the same. We recorded the magnitude of the average velocity and the average y -coordinate of all vertices to perform the comparison, and show the results in Figure 5 (bottom). Both velocity and position are perfectly consistent with the analytical results.

Due to the implicit nature of our solver, we can robustly handle phenomena involving strong coupling between elastic forces and frictional contact, such as the house of cards shown in Figure 6. When the friction coefficient is sufficiently high ($\mu = 0.6$), the majority of the structure remains standing even when the top two cards fall, while for a smaller friction coefficient ($\mu = 0.2$) the structure collapses realistically. For comparison, we reimplemented

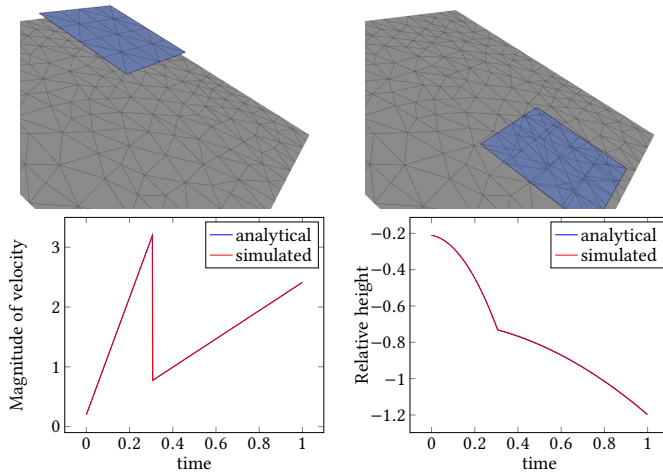


Fig. 5. Comparison of the simulated results with the analytical results. Top: the scenario used for comparison. A cloth parallel to an inclined plane falls on the plane and slides down. The motion lasts for 1 second, with $\mu = 0.2$. Bottom left: the magnitude of the velocity over time. Bottom right: the relative height of the mass center of the cloth over time

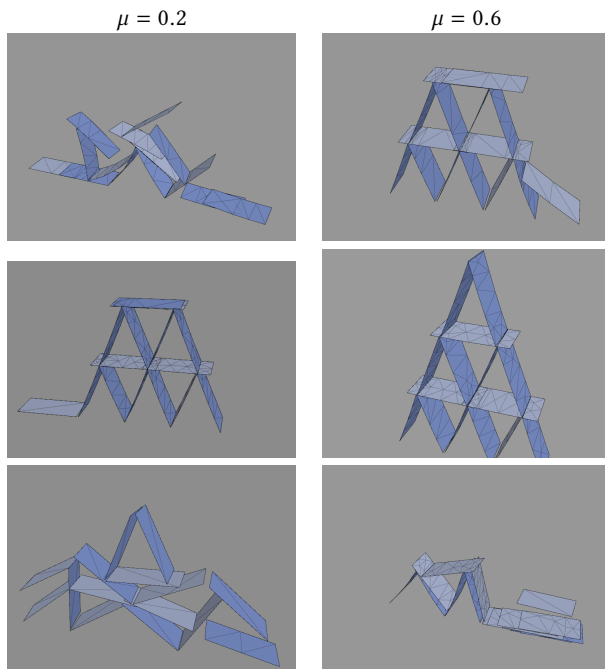


Fig. 6. A house of cards simulated with different methods. *Top row*: our method with $\Delta t = 2$ ms. *Middle row*: the Bridson/Harmon scheme with $\Delta t = 2$ ms. *Bottom row*: the Bridson/Harmon scheme with $\Delta t = 0.2$ ms.

the traditional collision processing scheme proposed by Bridson et al. [2002], with the inelastic projection of Harmon et al. [2008] replacing the geometric collision response and rigid impact zones. Using the same adaptive refinement settings, we were not able to

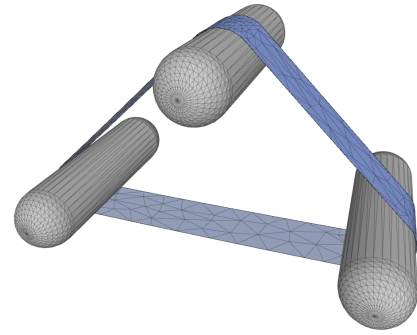


Fig. 7. A belt stretched between three cylinders.

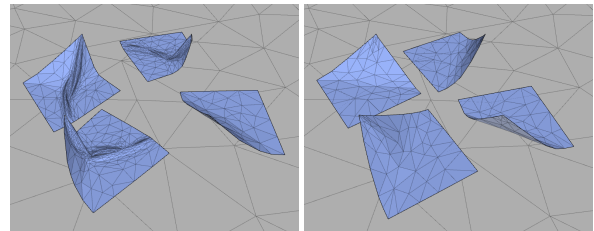


Fig. 8. A sheet of cloth is pulled in a circle, starting from right and going clockwise. The method of Otaduy et al. (*left*) uses a pyramidal approximation aligned with the current relative velocity, creating anisotropic wrinkles and a noticeable bias in the motion of the cloth. In contrast, our method (*right*) uses the exact Coulomb friction cone, resulting in a smooth circular motion.

obtain stable behavior with this scheme. Increasing the minimum triangle size gave stable results but with excessive friction creating a sticking effect, especially apparent in Figure 6 with $\Delta t = 2$ ms and $\mu = 0.2$ (left column, middle row). While many of the collisions are resolved through repulsion forces, the failsafe is still triggered every frame. Excessive friction is likely caused by these impact zones, with a large normal force increasing the friction forces on other vertices in the zone. We found that decreasing the time step by a factor of 10 removed this artifact entirely.

The coupling between elastic and contact forces is also important in materials under high tension, such as the belt shown in Figure 7. The internal tension in the belt leads to strong frictional contact forces with the cylinders, which in turn causes the belt to be pulled along when the cylinders rotate. Similar situations arise in a flexible strap that supports a heavy object, such as a backpack or handbag. We were not able to simulate this problem successfully with the Bridson/Harmon method. Due to the strong normal forces created by the high tension in the belt, impact zones were formed over the entire contact surfaces with the cylinders, leading to very large and extremely poorly conditioned systems in the inelastic projection step.

The more recent methods proposed by Kaufman et al. [2008] and Otaduy et al. [2009] handle elasticity and contact together in an implicit manner, and thus do not suffer from the limitations of the traditional decoupled approach. However, both these methods linearize the Coulomb friction cone using a polyhedral approximation,

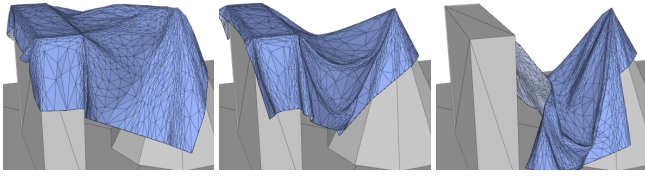


Fig. 9. Adaptive refinement allows our method to resolve contacts involving sharp edges and corners, such as this example with a box edge and a conical obstacle.

creating artificial anisotropy in the frictional behavior. An example demonstrating the undesirable effects of this anisotropy is shown in Figure 8. One vertex of a cloth sheet is constrained to move in a circle, while the rest of the cloth drags behind it while experiencing frictional contact with the floor. The method of Otaduy et al. [2009] approximates the friction cone with a square pyramid aligned with the current relative velocity, and we observe a significant bias for the contact to continue in the same direction. As a result, the cloth exhibits an unrealistic squarish trajectory, only turning when the external forces change significantly in direction. Our method, which does not linearize the cone, is free of these directional artifacts.

Thanks to our adaptive remeshing step, our method is capable of resolving contacts involving sharp obstacle edges and corners, even though our constraints are applied only on mesh vertices. Figure 9 shows an example involving a box edge and a conical obstacle. The cloth slides smoothly off the box and eventually hangs stably on the cone. A non-adaptive cloth simulation is prone to exhibit artifacts in such cases due to vertices “snagging” on sharp features, unless the mesh resolution is extremely high. By adaptively refining the mesh only near the sharp features, our method successfully computes a realistic frictional contact response in this situation.

We illustrate the advantage of our proximity metric over the existing obstacle metric with two simple examples. In Figure 10 (top), we show a simplified house of cards with two cards leaning against each other. Near the bottom of the cards, the previous metric (left) results in more unnecessary refinement, while it fails to refine the mesh near the top where the cards are in contact. Figure 10 (bottom) shows a cloth falling on a sharp point and an edge. The previous metric does not cause enough refinement at the contacting areas, and exhibits bumpy artifacts as a result.

Pruning redundant contacts with nearly the same normal vectors, as described in Section 5.2, simplifies the problem by reducing the number of constraints and the amount of vertex duplication. To quantify its effect on solver behavior and simulation quality, we ran the table and cone example (Figure 9) multiple times, setting the pruning thresholds to 0, 0.17, 0.34, and 0.5 radians (roughly 0° , 10° , 19° , and 29°). Figure 11 shows the number of contacts and the number of iterations for each frame obtained from the simulation results. With more pruning, fewer redundant constraints are kept and less vertex duplication is necessary, while the iteration count is largely unaffected; consequently, the computation becomes faster. As can be seen in the supplementary video, this operation does not introduce noticeable artifacts as long as the pruning threshold is not too large. We found that 0.17 radians (10°) was a safe value for all

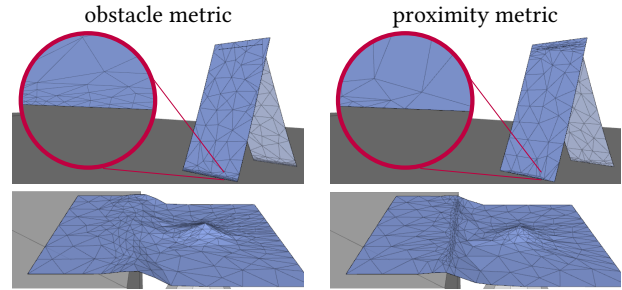


Fig. 10. Comparison between the existing obstacle metric and our proximity metric. Top row: two stiff cards lean against each other. Bottom row: a square sheet of cloth falling onto an edge and a sharp point.

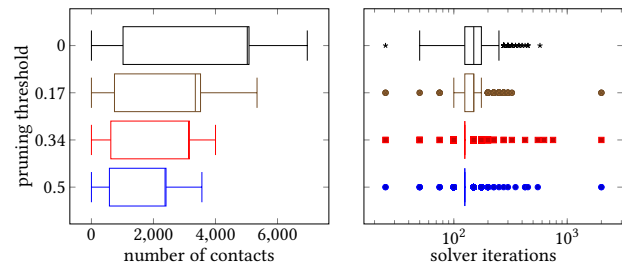


Fig. 11. The number of contact constraints and number of solver iterations for the table and cone example with different pruning thresholds (in radians), visualized as boxplots.

our examples, and reduced the running time considerably compared to no pruning.

7.2 Garment simulation

We have run our method on four different character animations with various types of garments. All character animations and garments were designed under the Autodesk 3ds max software, which was also used for producing the final rendering of our results. The first character motion, named **Twist**, is shown in Figure 1, and the three others, named **Arabesque**, **Clubbing**, and **HipHop** respectively, are displayed in Figure 12. For each garment involved we chose some material parameters using the Berkeley garment database [Wang et al. 2011], as described in Table 1.

To comply with friction coefficients measured in real textiles [Inoue 2011], we chose μ in the range 0.0 – 0.6. Exact friction coefficients taken for each motion are given in Table 2.

Table 1. Material parameters for our simulated garments, taken from the Berkeley garment database [Wang et al. 2011].

Animation	Garment	Material	Damping
Twist	batwing dress	tango-red-jet-set	0.001
Arabesque	'60s dress	gray-interlock	0.001
Clubbing	'60s dress	gray-interlock	0.001
Hip-Hop	hoodie	pink-ribbon-brown	0.002
Shawl	shawl	gray-interlock	0



Fig. 12. Three garment simulations, namely **Arabesque**, **Clubbing**, and **HipHop**, with (left) $\mu = 0$ and (right) $\mu = 0.3$.

Note that for a given simulation, we chose a uniform friction coefficient for all cloth-cloth and cloth-body contact pairs for the sake of simplicity. But of course, our method allows the user to specify a different friction coefficient at each contact, and for instance to make a distinction between cloth-cloth friction and cloth-body friction.

Finally, here again, all simulations were run with the same timestep of 2 ms, with similar remeshing parameters, and with a solver tolerance equal to 10^{-8} m/s.

The **Twist** sequence shown in Figure 1 yields a wide range of cloth sliding motions and folding patterns enabled by varying the friction coefficient on a batwing dress featuring a large neckline and a tight waist. For low friction parameters ($\mu \leq 0.1$), sleeves slide along the shoulders during the twisting of the torso, revealing a nude shoulder of the character. In contrast, higher friction coefficients ($\mu \geq 0.3$) limit the sliding of the garment over the shoulders and instead create wrinkles along the neckline. Compared to a large friction coefficient ($\mu = 0.3$), a very large friction coefficient ($\mu = 0.6$) furthermore amplifies the number of folds appearing on the torso (see accompanying video), due to increased sticking.

The **Arabesque** and **Clubbing** animations show two different dance motions: the first featuring slow and gradual movements, the second rapid and energetic. In both sequences, our method accurately resolves the contact interactions between the character and the garments, even in the presence of shocks caused by rapid character motions throughout **Clubbing** and at the end of **Arabesque**. The importance of frictional contact can be seen in the behavior of the dress in the raised-leg pose as well as the subtle patterns of wrinkles over the torso.

The **HipHop** animation in the last row of Figure 12 further demonstrates the importance of friction for realistic garment behavior. Here, a character wearing a hooded pullover performs both slow and rapid movements of the head and upper body. The action of friction is essential for the hood to move convincingly with the head of the character. As can be seen in the supplementary video, without friction the hood slides around unrealistically over the head, and eventually falls off at the end of the animation.



Fig. 13. Shawl demo, with $\mu = 0.6$. The character grasps the shawl thanks to contact and Coulomb friction, and then throws it around her left shoulder. Finally the shawl slightly rolls over the shoulder due to gravity and friction. With a lower friction coefficient, the character is unable to grasp the shawl as it is too slippery.

In the **Shawl** animation (Figure 13), we show that our friction solver can be used to model manual manipulation of cloth, in this case, throwing a shawl over one shoulder. Previous work on dexterous manipulation [Bai et al. 2016; Clegg et al. 2015] has required artificial sticking models or additional constraints due to inadequacies with existing contact models for cloth. In this animation, we manually animated only the motion of the character's arm and hand, while the shawl was simulated using our method. After its initial, flat state, the shawl falls on the right shoulder and can only rest

on it for very large friction coefficients ($\mu > 0.4$). Otherwise, it slips over the shoulder and falls down on the ground (which also yields a challenging scenario as many shocks and self-contacts are involved). For $\mu = 0.6$, the hand manages to grasp the cloth and to throw it on the left shoulder, and the shawl finally rests on top of the shoulder after some slight rolling motion. Note that no artificial constraints between the character’s hand and the shawl were added. Of course, in our case the character motion and especially the finger displacement had to be carefully designed for the specific initial conditions, and does not work if the dynamical properties of the cloth change; automatic control of such motions is still an open challenge. We show this result to demonstrate the potential of our approach to facilitate future work in this direction.

7.3 Performance

Our method was implemented in C/C++ with a multi-threaded architecture, with OpenMP applied on basic parallel tasks such as matrix assembly; recall that the Gauss–Seidel algorithm for our primal solver is inherently sequential. Our simulations were all run on the same PC featuring 4 dual-core Intel i7-5600U processors running at 2.60GHz. Detailed performance of our algorithm on all our simulation examples are provided in Table 2.

One can first notice that, in general, the computational cost is mainly due to the primal frictional contact solver, while other steps (collision detection, remeshing, merging) have marginal costs. Roughly, small and short demos like **House of cards** took a few minutes to complete, whereas large and long demos like **Arabesque** took around 20 hours.

The performance data indicates that the convergence of the solver is related to two different factors. On the one hand, the convergence deteriorates as the friction coefficient increases. On the other hand, the solver has trouble converging when body self-penetrations occur. We have especially noticed that in the **HipHop** motion, there are a hundred frames where the left arm is colliding with the chest near the armpit. This zone precisely corresponds to the place where the solver does not manage to reach the tolerance, even for a small friction coefficient. For $\mu = 0.0$, one timestep even diverges but the solver manages to recover at the next timesteps. These two convergence issues are further discussed in Section 7.4. Overall, the solver converges to a decent precision without having to tune any parameters. It is noteworthy that the recourse to geometrical intersection removal is rare: at worst, less than 0.1% contacts are affected by interpenetrations on average, and these occur in less than 5% of the timesteps.

Finally, although it is difficult to compare our performance with previous methods due to differences in implementation, we have not noticed any significant workload due to our algorithm. The run time performance of the Bridson/Harmon scheme is highly influenced by how the contacts are resolved. We used Eigen’s sparse conjugate gradient solver for the inelastic projection, with LDLT as a failsafe if it failed to converge. For **House of cards**, Bridson/Harmon was approximately 10 times faster than our method per time step with $\Delta t = 2$ ms, however yielding much less accurate results: for the same quality of results, the timestep had to be decreased by a factor 10. For **Circular Dragging**, we compared performance against our

implementation of Otaduy et al.’s nested relaxation solver. On the same hardware, our timings were comparable to those of their solver, while the quality of the results was improved.

7.4 Discussion and limitations

Nonexistence of solutions. Our method seeks to exactly resolve the non-penetration and Coulomb constraints at the end of each time step. As such, it fails to converge when the cloth is trapped between self-intersecting parts of the character or other obstacle geometry, as mentioned before. Of course, in the presence of obstacle self-intersections, there is no solution to the frictional contact problem as defined. Nevertheless, as such situations arise frequently in character animation, it would be desirable for the solver to handle them gracefully without compromising physical accuracy in the rest of the mesh. One possible solution would be to simply live with interpenetration, by relying for instance on the flypapering method of Baraff et al. [2003] for handling vertices that are trapped in the self-interpenetrating regions.

Our method also exhibits slower convergence for larger coefficients of friction ($\mu \geq 0.3$). This is not surprising: theoretical results predict that the frictional contact problem is more likely to possess a solution when the friction coefficient is low, both in the continuous [Blumentals et al. 2016] and discrete case [Cadoux 2009]. A well-known example of non-existence of solution due to a high friction coefficient is the Painlevé paradox. It is thus expected that the frictional contact problem is all the more difficult to solve numerically as the friction coefficient is high. The non-existence of solution is one consequence of choosing a rigid frictional contact model. One interesting research direction would be to relax the Signorini–Coulomb law when it fails to yield a solution, for instance by temporarily allowing for some local compliancy of the contacting objects at the troublesome locations.

On theoretical guarantees of robustness. While our method works well on the wide range of examples we have shown, we do not have theoretical guarantees of intersection-free solutions in all cases. In particular, when our solver does not find a solution to the frictional contact problem, we apply the ICM algorithm of Volino and Magnenat-Thalmann [2006] to remove any interpenetrations. ICM is not truly a failsafe and is known to fail on highly tangled input, although we have observed it to be sufficient in all of our experiments. If theoretical guarantees of intersection-freeness are required, one could perform intersection tests within the remeshing process [Brochu and Bridson 2009]. An impact zone failsafe [Bridson et al. 2002; Harmon et al. 2008; Provot 1997] could then be employed after our contact solver, guaranteeing intersection resolution in the presence of static or rigidly moving obstacles.

Further, in our implementation, we only linearize the contact constraints at the beginning of each time step, which may lead to inaccuracies or missed contacts under large time steps. For greater robustness, one could incorporate the contact manifold refinement (CMR) algorithm proposed by Otaduy et al. [2009], which adds an outer loop in which contact constraints are added and the contact solve is repeated until all interpenetrations are resolved. In our preliminary experiments with such an approach, we did not find it to be beneficial as our time steps were already small.

Table 2. Performance of our solver for all our examples

Example	μ	\bar{n}_v ¹	\bar{n}_c ¹	\bar{n}_i ²	\bar{n}_{ir} ²	\bar{t}_p ³	\bar{t}_r ³	\bar{t}_{cd} ³	\bar{t}_m ³	\bar{t}_s ³	\bar{T} ⁴	\bar{e} ⁵	\bar{n}_{iter} ⁵
Box and Cone	0.2	4372	2364	0.02	0	1.66	0.30	0.11	0.11	0.54	26.6	6.3e-9	129
High-tension belt	0.4	502	476	0.00	0	2.48	0.03	0.01	2e-3	2.42	39.7	5.7e-9	261
House of cards	0.2	75	176	5e-4	0	0.09	2e-3	9e-4	4e-4	0.09	1.4	2.6e-12	387
	0.6	218	119	0.00	0	0.29	6e-3	2e-3	3e-4	0.28	4.6	3.9e-11	1778
Circular dragging	0.6	59	26	0.00	0	0.04	2e-3	6e-5	0.00	0.04	0.6	1.5e-10	25
Twist	0.0	5874	2557	0.02	0	2.10	0.64	0.21	0.09	0.59	33.6	7.6e-9	192
	0.1	5827	2521	0.01	0	2.08	0.61	0.20	0.09	0.66	33.3	7.4e-9	180
	0.3	5801	2647	0.51	0	6.22	0.67	0.20	0.09	4.58	99.5	1.0e-4	378
	0.6	6052	2663	1.63	0	8.12	0.90	0.20	0.09	5.91	129.9	1.3e-3	484
Arabesque	0.0	5224	2845	0.03	0	3.45	0.69	0.19	0.14	1.67	55.2	6.8e-8	262
	0.3	5532	2996	0.85	0	4.37	0.70	0.19	0.15	2.50	69.9	1.1e-3	294
Clubbing	0.0	7008	2803	0.02	0	2.36	0.71	0.21	0.13	0.68	37.8	7.7e-9	254
	0.1	6907	2833	0.04	0	2.41	0.69	0.21	0.13	0.77	38.6	4.7e-9	199
	0.3	7153	2827	0.03	0	4.75	0.75	0.18	0.14	2.95	76.0	2.3e-8	242
HipHop	0.0	5577	2597	0.05	0	11.01	0.57	0.18	0.09	9.59	176.2	8.8e-9 [*]	353
	0.3	5500	2508	0.09	0	6.64	0.81	0.23	0.11	4.52	106.2	1.e-4	558
Shawl	0.3	6795	3416	1.18	0	3.25	0.69	0.24	0.28	0.86	52.0	1.8e-3	273
	0.6	3733	1571	1.79	0	2.18	0.46	0.16	0.03	0.89	34.9	4.e-4	219

¹ Average number of vertices (n_p) and contact points (n_c)

² Average number of remaining intersections after our frictional contact solver (n_i), and after ICM (n_{ir})

³ Average simulation time in seconds per timestep (dt = 2 ms) for a full timestep (\bar{t}_p), including the time for remeshing (\bar{t}_r), collision detection (\bar{t}_{cd}), merging (\bar{t}_m), and the primal frictional contact solver (\bar{t}_s). Remeshing time (\bar{t}_r) includes tangent plane computation and ICM.

⁴ Average simulation time in seconds per rendering frame ($\bar{T} = 16 \times \bar{t}_p$)

⁵ Average error of the solver in m/s (\bar{e}), with $tol = 1e^{-8}$ everywhere except for **House of cards** where $tol = 1e^{-11}$, and average number of iterations (\bar{n}_{iter}), with $n_{itermax} = 2000$

^{*} Result after discarding one single outlier timestep due to body self-penetration in **HipHop**

Accuracy of nodal constraints. A certain amount of inaccuracy is introduced by our strategy of moving contact constraints to the nearest mesh vertices. In particular, contact forces may not conserve angular momentum, as the force vectors are no longer collinear. This error is mitigated by the fact that our remeshing strategy refines the mesh near contact points, ensuring that the amount of displacement for each contact constraint is typically small. We have not observed any visual artifacts or nonphysical behavior resulting from this step, although we expect that it would become noticeable if the maximum level of refinement was decreased significantly.

Future generalizations. While our motivating applications and example results focus on cloth simulation, our nodal contact solver is not per se limited to simulating cloth and other low-dimensional objects. In principle, the solver can be applied to any deformable object with a nodal discretization, including volumetric elastic bodies, although remeshing volumetric models is more challenging. Our method also requires forces to be linearized over the time step, so may not work well in cases where this is a poor approximation, for example when modeling strongly nonlinear elasticity, or strain limiting constraints. Finally, we would also like to extend our method to more complex friction models, such as anisotropic and asymmetric friction [Pabst et al. 2009]. We leave the exploration of these directions for future work.

8 CONCLUSION

We have presented a novel technique for simulating frictional contact in cloth simulation both accurately and robustly. Our technique relies on a nodal solver that efficiently solves the implicit frictional contact problem, combined with the use of adaptive refinement which permits the solver to be applied in general contact scenarios with complex geometry. In contrast to existing methods, our solver does not decouple elastic forces from frictional contact, does not require the inverse of a large system matrix, and works with the exact Coulomb cone without linearization. We have demonstrated our method on many complex scenes, showing its benefits over existing methods and its effectiveness for realistic garment simulation on animated characters.

ACKNOWLEDGMENTS

The authors would like to thank Liam Toran for his early thoughts as an intern on a new quartic solve adapted to the primal formulation, David Harmon for sharing reference code for inelastic projection, and Mickaël Ly for help with creating renderings and the supplementary video. We are also grateful to the anonymous reviewers for their valuable comments. This work was supported in part by the NSF grant #1657089, the ERC grant GEM (StG-2014-639139), and by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissement d’Avenir.

REFERENCES

- V. Acary and B. Brogliato. 2008. *Numerical methods for nonsmooth dynamical systems*. Lecture Notes in Computational and Applied Mechanics, Vol. 35. Springer.
- S. An, D. James, and S. Marschner. 2012. Motion-driven Concatenative Synthesis of Cloth Sounds. *ACM Trans. Graph.* 31, 4, Article 102 (July 2012), 10 pages. <https://doi.org/10.1145/2185520.2185598>
- U. Ascher and E. Boxerman. 2003. On the modified conjugate gradient method in cloth simulation. *The Visual Computer* 19, 7-8 (2003), 526–531. <https://doi.org/10.1007/s00371-003-0220-4>
- Y. Bai, W. Yu, and K. Liu. 2016. Dexterous Manipulation of Cloth. *Computer Graphics Forum* 35, 2 (2016), 523–532. <https://doi.org/10.1111/cgf.12852>
- D. Baraff. 1989. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Computer Graphics Proceedings*. ACM, New York, NY, USA, 223–232.
- D. Baraff. 1991. Coping with friction for non-penetrating rigid body simulation. In *Computer Graphics Proceedings*. ACM, 31–40.
- D. Baraff. 1994. Fast contact force computation for nonpenetrating rigid bodies. In *Computer Graphics Proceedings*. ACM, New York, NY, USA, 23–34.
- D. Baraff and A. Witkin. 1998. Large Steps in Cloth Simulation. In *Computer Graphics Proceedings*. 43–54. <http://www.cs.cmu.edu/~baraff/papers/sig98.pdf>
- D. Baraff, A. Witkin, and M. Kass. 2003. Untangling Cloth. *ACM Transactions on Graphics* 22, 3 (2003), 862–870.
- A. Bartle, A. Sheffer, V. Kim, D. Kaufman, N. Vining, and F. Berthouzoz. 2016. Physics-driven Pattern Adjustment for Direct 3D Garment Editing. *ACM Trans. Graph.* 35, 4, Article 50 (July 2016), 11 pages. <https://doi.org/10.1145/2897824.2925896>
- G. Bingham. 2012. Efficient Three Dimensional Modelling of Additive Manufactured Textile Structures.
- A. Blumentals, B. Brogliato, and F. Bertails-Descoubes. 2016. The contact problem in Lagrangian systems subject to bilateral and unilateral constraints, with or without sliding Coulomb's friction: a tutorial. *Multibody System Dynamics* 38, 1 (2016), 43–76. <https://doi.org/10.1007/s11044-016-9527-6>
- F. Bossen and P. Heckbert. 1996. A Pliant Method for Anisotropic Mesh Generation. In *Proc. 5th International Meshing Roundtable*. 63–76.
- R. Bridson, R. Fedkiw, and R. Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.* 21, 3 (2002), 594–603. <http://www.cs.ubc.ca/~rbridson/docs/cloth2002.pdf>
- R. Bridson, S. Marino, and R. Fedkiw. 2003. Simulation of Clothing with Folds and Wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 28–36. <http://dl.acm.org/citation.cfm?id=846276.846281>
- T. Brochu and R. Bridson. 2009. Robust Topological Operations for Dynamic Explicit Surfaces. *SIAM Journal on Scientific Computing* 31, 4 (2009), 2472–2493. <https://doi.org/10.1137/080737617> arXiv:<http://dx.doi.org/10.1137/080737617>
- F. Cadoux. 2009. *Méthodes d'optimisation pour la dynamique non-régulière*. Ph.D. Dissertation. Université Joseph Fourier.
- R. Casati, G. Daviet, and F. Bertails-Descoubes. 2016. *Inverse elastic cloth design with contact and friction*. Research Report. Inria Grenoble Rhône-Alpes, Université de Grenoble. <https://hal.archives-ouvertes.fr/hal-01309617>
- G. Cirio, J. Lopez-Moreno, D. Miraut, and M. Otaduy. 2014. Yarn-level Simulation of Woven Cloth. *ACM Trans. Graph.* 33, 6, Article 207 (Nov. 2014), 11 pages. <https://doi.org/10.1145/2661229.2661279>
- A. Clegg, J. Tan, G. Turk, and K. Liu. 2015. Animating Human Dressing. *ACM Trans. Graph.* 34, 4, Article 116 (July 2015), 9 pages. <https://doi.org/10.1145/2766986>
- R. Cottle, J.-S. Pang, and R. Stone. 2009. *Numerical methods for nonsmooth dynamical systems*. Siam.
- P. Cundall. 1971. A computer model for simulating progressive large scale movements of blocky rock systems. In Proceedings of the Symposium of the International Society of Rock Mechanics. In *Proceedings of the Symposium of the International Society of Rock Mechanics*, Vol. 1. 132–150.
- G. Daviet. 2016. *Modeling and simulating complex materials subject to frictional contact: Application to fibrous and granular media*. Ph.D. Dissertation. Université Grenoble Alpes. <https://tel.archives-ouvertes.fr/tel-01684673>
- G. Daviet, F. Bertails-Descoubes, and L. Boissieux. 2011. A hybrid iterative solver for robustly capturing Coulomb friction in hair dynamics. *ACM Trans. Graph.* 30 (2011), 139:1–139:12. Issue 6. <http://www.inrialpes.fr/bipop/people/bertails/Papiers/hybridIterativeSolverHairDynamicsSiggraphAsia2011.html>
- G. Daviet, F. Bertails-Descoubes, and R. Casati. 2015. Fast Cloth Simulation with Implicit Contact and Exact Coulomb Friction. ACM SIGGRAPH / Eurographics Symposium on Computer Animation. (Aug. 2015). <https://hal.inria.fr/hal-01180756> Poster.
- G. De Saxcé and Z.-Q. Feng. 1998. The bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. *Math. Comput. Modelling* 28, 4-8 (1998), 225–245.
- E. English and R. Bridson. 2008. Animating developable surfaces using nonconforming elements. *ACM Trans. Graph.* 27, 3 (2008), 1–5. <https://doi.org/10.1145/1360612.1360665>
- Z. Erickson, A. Clegg, W. Yu, G. Turk, C. K. Liu, and C. C. Kemp. 2017. What does the person feel? Learning to infer applied forces during robot-assisted dressing. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 6058–6065. <https://doi.org/10.1109/ICRA.2017.7989718>
- M. Fukushima, Z.-Q. Luo, and P. Tseng. 2002. Smoothing Functions for Second-Order-Cone Complementarity Problems. *SIAM J. on Optimization* 12 (February 2002), 436–460. Issue 2. <https://doi.org/10.1137/S1052623400380365>
- R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. 2007. Efficient simulation of inextensible cloth. In *ACM Trans. Graph. (SIGGRAPH '07)*. ACM, New York, NY, USA, Article 49. <https://doi.org/10.1145/1275808.1276438>
- E. Grinspun, A. Hirani, M. Desbrun, and P. Schröder. 2003. Discrete Shells. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'03)*. ACM-EG SCA, 62–67. <http://www.multires.caltech.edu/pubs/ds.pdf>
- D. Harmon, E. Vouga, B. Smith, R. Tamstorf, and E. Grinspun. 2009. Asynchronous Contact Mechanics. *ACM Trans. Graph.* 28, 3, Article 87 (July 2009), 12 pages. <https://doi.org/10.1145/1531326.1531393>
- D. Harmon, E. Vouga, R. Tamstorf, and E. Grinspun. 2008. Robust Treatment of Simultaneous Collisions. *ACM Trans. Graph.* 27, 3, Article 23 (Aug. 2008), 4 pages. <https://doi.org/10.1145/1360612.1360622>
- M. Inoue. 2011. Surface Friction Properties of Fabrics and Human Skin, in *New Tribological Ways. In Croatia: InTech*. 265–272. <http://cdn.intechweb.org/pdfs/15598.pdf>
- M. Jean. 1999. The Non Smooth Contact Dynamics Method. *Computer Methods in Applied Mechanics and Engineering* 177 (1999), 235–257. Special issue on computational modeling of contact and friction, J.A.C. Martins and A. Klarbring, editors.
- C. Jiang, T. Gast, and J. Teran. 2017. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact. *ACM Trans. Graph.* 36, 4, Article 152 (July 2017), 14 pages. <https://doi.org/10.1145/3072959.3073623>
- D. Kaufman, T. Edmunds, and D. Pai. 2005. Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.* 24, 3 (2005), 946–956.
- D. Kaufman, S. Sueda, D. James, and D. Pai. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Trans. Graph.* 27, 5, Article 164 (Dec. 2008), 11 pages. <https://doi.org/10.1145/1409060.1409117>
- A. Klarbring. 1987. Contact Problems with Friction by Linear Complementarity. *Unilateral Problems in Structural Analysis - 2* (1987), 197–219. https://doi.org/10.1007/978-3-7091-2967-8_11
- M. Konaković, K. Crane, B. Deng, S. Bouaziz, D. Piker, and M. Pauly. 2016. Beyond Developable: Computational Design and Fabrication with Auxetic Materials. *ACM Trans. Graph.* 35, 4, Article 89 (July 2016), 11 pages. <https://doi.org/10.1145/2897824.2925944>
- E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. Otaduy, and S. Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Computer Graphics Forum* 31, 2 (may 2012). <http://www.gmrsv.com/Publications/2012/MBTBMOM12>
- M. Moore and J. Wilhelms. 1988. Collision detection and response for computer animation. In *Computer Graphics Proceedings*. 289–298.
- J.-J. Moreau. 1988. Unilateral contact and dry friction in finite freedom dynamics. Nonsmooth mechanics and applications, CISM Courses Lect. 302, 1-82 (1988).. (1988).
- R. Narain, T. Pfaff, and J. O'Brien. 2013. Folding and Crumpling Adaptive Sheets. *ACM Trans. Graph.* 32, 4 (2013), 51.
- R. Narain, A. Samii, and J. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph.* 31, 6, Article 152 (Nov. 2012), 10 pages. <https://doi.org/10.1145/2366145.2366171>
- M. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. 2009. Implicit Contact Handling for Deformable Objects. *Computer Graphics Forum* 28, 2 (2009), 559–568. <https://doi.org/10.1111/j.1467-8659.2009.01396.x>
- S. Pabst, B. Thomaszewski, and W. Straßer. 2009. Anisotropic Friction for Deformable Surfaces and Solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '09)*. ACM, New York, NY, USA, 149–154. <https://doi.org/10.1145/1599470.1599490>
- X. Provot. 1997. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation '97*, Daniel Thalmann and Michiel van de Panne (Eds.). Springer Vienna, Vienna, 177–189.
- L. Sigal, M. Mahler, S. Diaz, K. McIntosh, E. Carter, T. Richards, and J. Hodgins. 2015. A Perceptual Control Space for Garment Simulation. *ACM Trans. Graph.* 34, 4, Article 117 (July 2015), 10 pages. <https://doi.org/10.1145/2766971>
- D. Stewart and J. Trinkle. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *Internat. J. Numer. Methods Engrg.* 39, 15 (1996).
- B. Thomaszewski, S. Pabst, and W. Strasser. 2009. Continuum-based Strain Limiting. *Computer Graphics Forum* 28, 2 (apr 2009).
- P. Volino and N. Magnenat-Thalmann. 2006. Resolving Surface Collisions Through Intersection Contour Minimization. *ACM Trans. Graph.* 25, 3 (July 2006), 1154–1159. <https://doi.org/10.1145/1141911.1142007>
- H. Wang, R. Ramamoorthi, and J. O'Brien. 2011. Data-driven elastic models for cloth: modeling and measurement. *ACM Trans. Graph.* 30, 4, Article 71 (Aug. 2011), 12 pages. <https://doi.org/10.1145/2010324.1964966>