Frictional Contact on Smooth Elastic Solids

EGOR LARIONOV, University of British Columbia, Canada
YE FAN, Vital Mechanics, Canada
DINESH K. PAI, University of British Columbia and Vital Mechanics Research, Canada

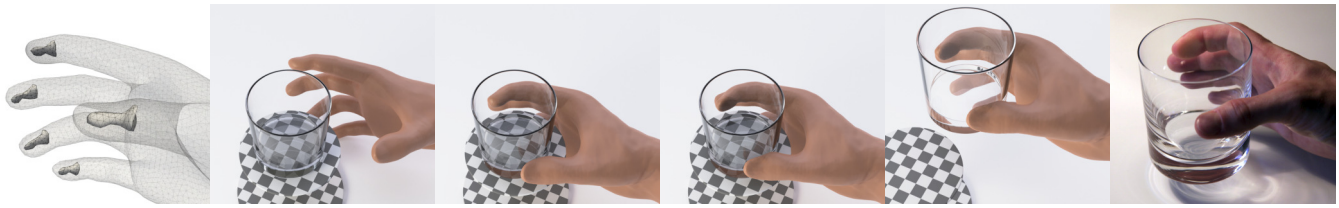


Fig. 1. A rigid whiskey glass is pinched between the index finger and thumb of an animated hand model and lifted up. The first frame shows the internal geometry of the distal phalanges whose vertices drive the finger tips of the tetrahedral simulation mesh of the hand. The remaining hand bones and tendons (not shown) are used to determine other interior animated vertices. The following frames show a selection of frames from the resulting simulation showing the grasp, lift and hold of the glass. The last image shows a photo of a similar scenario for reference. The collision surface of the hand is represented by an implicit function approximating a smoothed distance potential, while the glass surface is sampled using discrete points. Our method produces realistic deformation at the point of contact between the fingers and the rigid object.

Frictional contact between deformable elastic objects remains a difficult simulation problem in computer graphics. Traditionally, contact has been resolved using sophisticated collision detection schemes and methods that build on the assumption that contact happens between polygons. While polygonal surfaces are an efficient representation for solids, they lack some intrinsic properties that are important for contact resolution. Generally, polygonal surfaces are not equipped with an intrinsic inside and outside partitioning or a smooth distance field close to the surface.

Here we propose a new method for resolving frictional contacts against deforming implicit surface representations that addresses these problems. We augment a moving least squares (MLS) implicit surface formulation with a local kernel for resolving contacts, and develop a simple parallel transport approximation to enable transfer of frictional impulses. Our variational formulation of dynamics and elasticity enables us to naturally include contact constraints, which are resolved as one Newton-Raphson solve with linear inequality constraints. We extend this formulation by forwarding friction impulses from one time step to the next, used as external forces in the elasticity solve. This maintains the decoupling of friction from elasticity thus allowing for different solvers to be used in each step. In addition, we develop a variation of staggered projections, that relies solely on a non-linear optimization without constraints and does not require a discretization of the friction cone. Our results compare favorably to a popular industrial elasticity solver (used for visual effects), as well as recent academic work in frictional contact, both of which rely on polygons for contact resolution. We present examples of coupling between rigid bodies, cloth and elastic solids.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; • **Applied computing** → *Physics*.

Additional Key Words and Phrases: dry friction, contact, elasticity

Authors' addresses: Egor Larionov, Computer Science Department, University of British Columbia, Vancouver, BC, Canada, egor@cs.ubc.ca; Ye Fan, Vital Mechanics, Vancouver, BC, Canada, yefan@vitalmechanics.ca; Dinesh K. Pai, Computer Science Department, University of British Columbia and Vital Mechanics Research, Vancouver, BC, Canada, pai@cs.ubc.ca.

ACM Reference Format:
Egor Larionov, Ye Fan, and Dinesh K. Pai. 2021. Frictional Contact on Smooth Elastic Solids. *ACM Trans. Graph.* 40, 2, Article 15 (April 2021), 17 pages. https://doi.org/10.1145/3446663

1 INTRODUCTION

Frictional contact between elastic solids is an integral part of any simulation pipeline. It is essential for many real behaviors including grasping, walking, hitting a ball with racket and holding yarns of fabric together. However, simulating frictional contact remains challenging despite more than 30 years of research in computer graphics (see Section 2), and more than 300 years of research in physics and engineering.

A key reason for the difficulty is that the standard laws governing frictional contact behavior are non-smooth. They require integrating smooth differential equations with inequality constraints due to contact and friction, leading to non-smooth jumps in forces and velocities. This has been the major focus of previous research. There has been significant progress in understanding this aspect of frictional contact, especially following the seminal work of Moreau [1973]. It is now understood that introducing non-smoothness is essential to model friction's ability to make things stick, and to stop motion in finite time.

However, this makes frictional forces extraordinarily sensitive to non-smoothness in representing geometry, especially deforming geometry. Widely used polygonal surface representations, if directly used for simulating frictional contact, can result in simulations whose behavior depends sensitively on mesh resolution (see Figure 6 for a preview). A high resolution simulation may behave in a completely different way from a low resolution pre-visualization. Non-smoothness in polyhedral representations of the friction cone can also result in resolution-dependent artifacts. Thus, while non-smoothness in frictional forces is essential, artificially introduced non-smoothness in previous work can produce incorrect behavior.

Here we show how these problems can be effectively solved. We represent surfaces as smooth implicit surfaces computed from standard polygonal mesh representations. In contrast to parameteric representations of smooth surfaces (e.g., subdivision surfaces, NURBS), implicit surfaces provide a natural way to define the contact problem as an optimization with inequality constraints defined using a smooth potential function.

Many efficient solutions for frictional contact problems involve the linearization of the friction cone, producing a polyhedral approximation. Unfortunately this causes the artificial anisotropy of friction responses. Recent work in graphics present reformulations of the friction problem in various ways, which sidestep this issue. However, all of these methods require specialized optimization solvers. To maintain the flexibility of the method, we chose to improve on the simpler formulation of Kaufman et al. [2008] by reparametrizing the cone into polar coordinates, which removes the aforementioned artifacts.

We focus on methods that integrate frictional contact forces implicitly. This ensures that Coulomb's law is satisfied at the *end* of the time step exactly, and our sliding solid will eventually come to a complete stop. The need for fully implicit integration of frictional contact is further motivated by the observation that contact and friction impulses are generally not orthogonal (especially for stiff solids) [Erdmann 1994], and in fact may reinforce one another.

Modern solutions for implicit time integration of frictional contact typically involve explicit mesh representations [Daviet et al. 2011; Erleben 2017; Kaufman et al. 2008; Li et al. 2018]. To the best of our knowledge, fully implicit time integration of frictional contact has not yet been applied to solids represented by implicit surfaces. Some preliminary work has addressed contact [Desbrun and Cani 1994; Gascuel 1993] as well as explicit friction [Salisbury and Tarr 1997] on implicit surfaces, however these works target interactive applications and suffer from the aforementioned drawbacks of explicit integration of friction forces.

Here we propose a new method for modelling implicit frictional contact with smooth non-linear elastic solids. Our main contributions are:

- An implicit surface representation for modelling deformable solids. This approach admits a simple formulation for resolving non-linear contact using non-penetration constraints without the need for additional collision detection schemes.
- An algorithm for resolving non-linear frictional contact on implicit surfaces. Without having to linearize the friction cone at each point of contact, we are able to produce high fidelity frictional contact responses that are suitable for modelling real world contact problems.
- An enhanced time-splitting mechanism that propagates friction impulses into the constrained elasticity solve, while resolving contacts exactly. This permits using larger time steps in simulation with fine grained meshes.

In the following section we give a more comprehensive review of related works. Section 3 establishes the context for dynamic simulation, followed by a description of our implicit surface model in Section 4. We describe the details of our friction formulation in Section 5 and provide results for various simulation scenarios in Section 6. Finally, Section 7 explains the limitations of our method, and Section 8 suggests topics for future work and concludes with a summary of our contributions.

2 RELATED WORK

Simulation of realistic elastic objects and contact between them received much attention from the computer graphics community over decades. A number of approaches have been explored for modelling, animating and simulating the physics of rigid and elastic solids as well as flexible shells like cloth. Some methods employ particles to represent the deformable media, from smoothed particle hydrodynamics (SPH) [Becker et al. 2009] to material point method (MPM) [Guo et al. 2018; Han et al. 2019] with full frictional contact formulations as well as position based dynamics (PBD) [Müller et al. 2007] and derivatives, for real-time applications. The most popular method for simulating elastic bodies — perhaps due to the sheer volume of available literature — remains the finite element method (FEM). Some methods employ discontinuous discretizations [English and Bridson 2008; Kaufmann et al. 2008] while others rely on continuously linked meshes [Sifakis and Barbic 2012] to represent deformable media. The latter, being the most popular is the discretization we choose, although our contact formulation does not limit us to a particular style of discretization since we require only a sampling of points and normals to produce a contact surface.

Smooth contact has previously been explored in the context of rigid-bodies [Baraff 1990; Kry and Pai 2003] using parametric surfaces. Smoothness becomes even more critical when the contacting bodies are allowed to deform and slide against each other. We chose to represent the contacting surface using an implicit surface formulation to leverage the intrinsic potential surrounding such a surface for anticipating potential contacts.

Implicit surfaces. Implicit surfaces have been widely used to represent 3D geometry in literature. From modelling and animation [Turk and O'Brien 2002; Vaillant et al. 2014] to simulating contact between deformable solids [Cani 1993; McAdams et al. 2011]. A popular Eulerian method for using implicit surfaces for developable media was introduced by Oscher et al. [2004], although Lagrangian variations exist [Hieber and Koumoutsakos 2005]. In contrast, we rely on FEM to compute elasticity and dynamics equations, while using the implicit surface solely for contact resolution and friction. McAdams et al. [2011] relied on a signed distance function representation of the geometry for collision detection. Vaillant et al. [2013; 2014] used Hermite radial basis functions (HRBF) to improve skinning. HRBFs employ a collection of points with normals to reconstruct the sampled surface, however it relies on a global solve including all involved sample points. Shen et al. [2004] used a *moving least squares* (MLS) implicit surface to approximate geometry, similarly involving all points per query, however no linear solve is required. We use the same MLS approach but with a compactly supported kernel [Morse et al. 2005; Ohtake et al. 2003; Öztireli et al. 2009].

Frictional contact. We review frictional contact models in the general case of rigid or deformable solids. We are particularly interested in the trade-offs between accuracy, simplicity and scalability

for these models. At a high level, frictional contact models can be grouped into penalty-based methods, and constraint-based methods.

The idea behind penalty-based contact handling is that of applying a restorative force between colliding objects in the direction that resolves the collision. Typically objects are allowed to intersect during simulation at which point the restorative penalty force is applied as a function of the amount of interpenetration [Kikuchi and Oden 1988; Moore and Wilhelms 1988] to emulate elasticity. This method is particularly efficient and hence popular in real-time computer graphics, however it suffers from stiffness and stability issues. Similarly, friction can be modelled by a viscous force proportional to tangential velocity between touching surfaces. This approach is especially useful for resolving friction within deformable solids [Baraff and Witkin 1998]. Unfortunately, viscosity based friction lacks the ability to model the biphasic nature of dry friction.

Constraint-based methods take a fundamentally different approach to frictional contact. Instead of guessing what penalty forces would lead to a collision free state (and with the correct friction behavior), we may chose the desired configuration directly [Baraff and Witkin 1998; Müller et al. 2007] or pick the exact forces that produce the target positions and/or velocities [Spillmann and Teschner 2008]. This corresponds to imposing explicit constraints on the configuration space. Although this approach is very efficient, it poses problems in the context of complex simultaneous collisions where no clear change in velocity produces a collision free state. Furthermore this technique is ill suited to more sophisticated domain discretizations like reduced coordinates or implicit surfaces.

Lastly, to circumvent the issues mentioned above, we may consider velocity as unknown and solve for frictional contact simultaneously with dynamics. This is the only way of enforcing the desired conditions at the end of each time step. Solving for contact and Coulomb friction simultaneously remains a challenging problem because it poses non-linear and non-smooth constraints on the overall problem. As a result, most methods in computer graphics tend to approximate Coulomb friction rather than solving for it exactly. For instance, Tresca-like methods model the sliding threshold as independent of the normal force [Lötstedt 1982]. An artifact of this approximation is that two solids with different weights will start sliding at different times on a gradually increasing slope. Kaufman et al. [2005] proposed to use the normal component of the reaction force without friction to determine tangential friction force. This violates the coupling between the frictional forces and normal forces as highlighted by the Painlevé paradox [Painlevé 1895]. Kaufman et al. [2008] then improved the technique by solving for tangential and normal components of the frictional contact using staggered projections. This approach lacks fidelity and reproducibility of certain behaviors since it requires a discretization of the friction cone that imposes a discretization dependent anisotropy on the sliding direction as noted by Acary and Brogliato [2008]. Later, Bertails-Descoubes et al. [2011] used the formulation by Alart and Curnier [1991] for frictional contact to produce a non-smooth Newton-Raphson root finding technique for hair friction. This was then extended by Daviet et al. [2011] to improve scalability using a Gauss-Seidel approach in conjunction with an alternative, Fischer-Burmeister formulation of frictional contact. The work by Daviet et al. was subsequently used to further improve efficiency using adaptive non-linearity [Kaufman et al. 2014]. Erleben [2017] abstracted the Alart-Curnier approach into a general proximal algorithm allowing for other iterative methods and friction models such as Coulomb-Contensou. Li et al. [2018] extended these ideas to adaptive cloth simulation based on nodal constraints. Macklin et al. [2019] produce an efficient solver for resolving frictional contact between rigid and deformable in interactive simulations. Finally Verschoor et al. [2019] extend the conjugate residual method to resolve frictional contact simultaneously with elasticity. In contrast to previous work, we maintain separate elasticity and frictional contact steps, allowing our method to fit into existing simulation pipelines more easily. Our work mainly focuses on contact fidelity and method extensibility.

3 BACKGROUND

In this section we will establish the context for handling frictional contact. Namely we will develop the dynamics partial differential equation for hyperelastic materials subject to traction on the surface. Then we describe how frictional contact can be resolved using this formulation in the sections to follow.

3.1 Generalized Coordinates

Consider a system of solids represented with m generalized coordinates $\mathbf{q}(t) \in \mathbb{R}^m$ with generalized velocities $\dot{\mathbf{q}}(t) \in \mathbb{R}^m$ at some time t. The generalized mass, denoted by $\mathbf{M}$, is an $m \times m$ square matrix. Generalized coordinates serve as an abstraction for a discrete space of coordinates that uniquely identify the configuration of a system of solid bodies. The position of a point i on the solid in physical space is given by $\boldsymbol{x}_i(\mathbf{q}) \in \mathbb{R}^3$. That is, $\boldsymbol{x}_i$ is parameterized by generalized coordinates $\mathbf{q}$. The velocity of this point is similarly represented by $\boldsymbol{v}_i(\dot{\mathbf{q}}) \in \mathbb{R}^3$. In general, we will use the italics to mark quantities in physical space and roman font face for quantities in configuration space.

3.2 Equations of Motion

The equations of motion can be written in configuration space as

$$\mathbf{M}\frac{d\dot{\mathbf{q}}}{dt} = \mathbf{b}(t, \mathbf{q}, \dot{\mathbf{q}}), \tag{1}$$

where $\mathbf{b}$ is the total generalized force. Forces generated by collisions can be very large and can happen on a very small time scale, especially between stiff solids. This can be quite problematic for numerical simulation. To mitigate this shortcoming we chose to work with force *impulses* rather than forces directly [Mirtich and Canny 1994; Moreau 1973]. Using backward Euler time integration, we integrate the equations of motion over an arbitrary time step Δt. To simplify notation, we use superscripts to denote the time variable:

$$\mathbf{M}\Delta\dot{\mathbf{q}} = \mathbf{f}^{t+\Delta t}, \tag{2}$$

where $\Delta\dot{\mathbf{q}} = \dot{\mathbf{q}}^{t+\Delta t} - \dot{\mathbf{q}}^{t}$ and $\mathbf{f}^{t+\Delta t}$ is the total generalized force impulse at the next time step:

$$\mathbf{f}^{t+\Delta t} = \int_t^{t+\Delta t} \mathbf{b}(s, \cdot, \cdot)\, ds.$$

The total impulse $\mathbf{f}$ is composed of two parts:

$$\mathbf{f} = \mathbf{f}_e + \mathbf{r}$$

where $\mathbf{f}_e$ is the elastic impulse and $\mathbf{r}$ is the generalized frictional contact impulse. By rearranging (2), we can form an implicit system for $\dot{\mathbf{q}}^{t+\Delta t}$ as follows

$$\dot{\mathbf{q}}^{t+\Delta t} = \dot{\mathbf{q}}^{t} + \mathbf{M}^{-1}(\mathbf{f}_e^{t+\Delta t} + \mathbf{r}^{t+\Delta t}). \tag{3}$$

To complete the simulation loop, we apply backward Euler time integration on the configuration variable to get

$$\mathbf{q}^{t+\Delta t} := \mathbf{q}^{t} + \Delta t \dot{\mathbf{q}}^{t+\Delta t}. \tag{4}$$

3.3 Contact Mechanics

In order to accurately represent frictional contacts, it is essential to establish a mapping from a generalized configuration space to the physical space where contacts occur. We define the Jacobian for a point i on a solid as

$$\boldsymbol{J}_i := \frac{\partial \boldsymbol{x}_i}{\partial \mathbf{q}}, \tag{5}$$

which maps configuration vectors to vectors in physical space:

$$\dot{\boldsymbol{x}}_i := \boldsymbol{v}_i = \boldsymbol{J}_i \dot{\mathbf{q}}. \tag{6}$$

Similarly, impulses in physical space can be mapped into the generalized configuration space. Suppose $\boldsymbol{f}_i$ is a force impulse at the point i on a solid in physical space, then from the principle of virtual work, $\boldsymbol{J}_i^\top \boldsymbol{f}_i$ corresponds to the same impulse in configuration space.

This is sufficient to model contacts with objects outside the system governed by $\mathbf{q}$; however for internal contacts extra care is required. Suppose a contact occurs between points i and j on two solids (possibly the same). Then the relative velocity between the two points is given by

$$\boldsymbol{v}_i - \boldsymbol{v}_j = (\boldsymbol{J}_i - \boldsymbol{J}_j)\dot{\mathbf{q}}.$$

In this scenario, by Newton's third law, the force impulse on point i is equal in magnitude and opposite in direction to the force impulse on point j:

$$\boldsymbol{f}_j = -\boldsymbol{f}_i.$$

Then the generalized force impulse on the whole system can be written as

$$\mathbf{f} = \boldsymbol{J}_i^\top \boldsymbol{f}_i + \boldsymbol{J}_j^\top \boldsymbol{f}_j = (\boldsymbol{J}_i - \boldsymbol{J}_j)^\top \boldsymbol{f}_i.$$

This produces a global *contact Jacobian* matrix $\boldsymbol{J}$ consisting of triplets of rows for each contact between points i and j defined by $\boldsymbol{J}_i - \boldsymbol{J}_j$. For contacts against external objects (not governed by $\mathbf{q}$), $\boldsymbol{J}_j$ is taken to be zero. Thus $\boldsymbol{J}$ is a $3n \times m$ matrix where n is the total number of contacts.

Finally we can rewrite (3) in terms of the stacked frictional contact impules $\boldsymbol{r} \in \mathbb{R}^{3n}$ as follows:

$$\dot{\mathbf{q}}^{t+\Delta t} = \dot{\mathbf{q}}^{t} + \mathbf{M}^{-1}(\mathbf{f}_e^{t+\Delta t} + \boldsymbol{J}^\top \boldsymbol{r}^{t+\Delta t}). \tag{7}$$

To disambiguate solving for $\boldsymbol{f}_i$ or $\boldsymbol{f}_j$ (and choosing the relative velocity direction) on internal contacts, we rely on an intrinsic property of our contact handling mechanism — we use a vertex based mesh to represent solid deformation and a smooth implicit surface to model its collision boundary. Collisions between two objects are only allowed between vertices on one object and an implicit surface representation of the other. Thus for each collision between two objects, the point i is always taken to be the point on the implicit surface, while j is a point corresponding to a mesh vertex.

Let us assume for the remainder of this section that the frictional contact impulses $\boldsymbol{r}^{t+\Delta t}$ are known. Then equation (7) can be solved from a variational principle:

$$\dot{\mathbf{q}}^{t+\Delta t} = \operatorname*{argmin}_{\mathbf{u}} \frac{1}{2} \left\| \dot{\mathbf{q}}^{t} - \mathbf{u} \right\|_{\mathbf{M}}^2 + W(\mathbf{q}^{t+\Delta t}) - \mathbf{u}^\top \boldsymbol{J}^\top \boldsymbol{r}^{t+\Delta t},$$

where the mass induced norm is defined by $\|\mathbf{u}\|_{\mathbf{M}}^2 := \mathbf{u}^\top \mathbf{M}\mathbf{u}$ for all $\mathbf{u} \in \mathbb{R}^m$, and $W(\mathbf{q}^{t+\Delta t})$ is the total elastic strain energy at the end of the time step. The negative derivative of strain energy $-\partial W / \partial \mathbf{q}$ effectively produces the desired elastic forces. We use the standard non-linear neo-Hookean strain energy [Sifakis and Barbic 2012] or its variation called Stable neo-Hookean [Smith et al. 2018], although other choices also work. Finally, with (4) we get the optimization problem

$$\dot{\mathbf{q}}^{t+\Delta t} = \operatorname*{argmin}_{\mathbf{u}} \frac{1}{2} \left\| \dot{\mathbf{q}}^{t} - \mathbf{u} \right\|_{\mathbf{M}}^2 + W(\mathbf{q}^{t} + \Delta t\, \mathbf{u}) - \mathbf{u}^\top \boldsymbol{J}^\top \boldsymbol{r}^{t+\Delta t}. \tag{8}$$

The goal of this paper is to build a method for solving Equation (8) for elastic solids with boundaries represented by implicit surfaces where $\boldsymbol{r}^{t+\Delta t}$ is constrained to satisfy the Coulomb friction conditions.

Note that the frictional contact impulse $\boldsymbol{r}^{t+\Delta t}$ is dissipative and cannot be derived from a single energy potential. In Section 5 we will build an algorithm for solving for $\boldsymbol{r}^{t+\Delta t}$ by iteratively solving a slightly modified version of equation (8).

4 IMPLICIT SURFACES

Handling contact on smooth solids can be problematic when they are represented by polygons. For instance, artifacts can arise when two smooth objects are sliding if contact is resolved on a per polygon basis, especially with coarse resolutions as shown in Figure 2. Furthermore non-smooth discretizations limit the ways general purpose optimization solvers can be applied for contact resolution. This forces the development of complex iterative schemes for collision handling and resolution [Bertails-Descoubes et al. 2011; Daviet et al. 2011; Li et al. 2018]. For these reasons we aim to design a surface representation that is ideal for smooth frictional contact handling and resolution. In particular we are looking for the following properties:

- The surface is smooth and supports a surrounding potential field. This enables using general purpose smooth non-linear optimizers that can handle inequality constraints for contact resolution.
- This field must be differentiable with respect to deformation of the solid in order to produce smooth sliding during deformation.
- To prevent dense derivatives, the potential must have a locally compact area of influence: changes at some local patch of the surface will not affect the potential in a far away region.
- The surface supports a differentiable interpolation between the degrees of freedom (generalized coordinates) and individual contact points. This is required for computing the contact Jacobian $\boldsymbol{J}$.

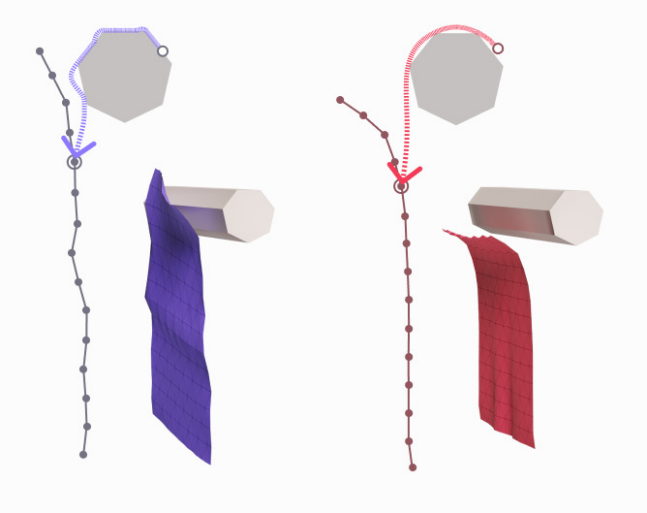


Fig. 2. *Smooth sliding on implicit surface.* This is a cross section of the coarsest sample simulation of Figure 6c listed in Section 6 after the cloth has slipped off the cylinder. Here, Houdini's cloth simulation is on the left, and ours is on the right. A vertex (circled) on the cloth in both examples is selected and traced from its original position as it slides along the surface. On the left, the polygonal contact produces an irregular contact pattern that causes unwanted vibrations in the cloth, while our method on the right produces a more natural smooth trajectory for a sliding vertex.

Previous work has employed embedded signed distance fields (SDFs) for collision handling [McAdams et al. 2011], however distance fields defined on polygonal surfaces are non-differentiable and can pose serious convergence issues when used as constraints in optimization solvers. Thus methods that use distance fields for implicit contact handling require some form of mollification [Li et al. 2020]. This is further illustrated in Appendix A.

It is worth noting that the desired potential field need not correspond to or approximate a signed distance to the polygonal surface. It is sufficient for the potential to be signed and smooth at the surface, whereas its scaling can be adjusted for optimal convergence. Nevertheless, the implicit surface must closely approximate the object being simulated. We measure the error of approximation using a Hausdorff distance.

In the remainder of this section we propose a collision surface representation that meets the above criteria.

4.1 Local Moving Least Squares Potential

Suppose we have a solid domain $\Omega(\mathbf{q}) \subset \mathbb{R}^3$ parametrized by $\mathbf{q}$ with a smooth boundary $\partial\Omega$. We omit the parameter $\mathbf{q}$ in this section for brevity, but it will become significant in Section 5.

Note that $\partial\Omega$ has a smooth outward facing unit normal field. Let $\mathcal{S}$ be a finite set of sample points on $\partial\Omega$. Then each $\boldsymbol{s} \in \mathcal{S}$ is endowed with a well defined outward facing unit normal $\boldsymbol{n}_{\boldsymbol{s}}$.

We would like to compute a global potential $\Psi : \mathbb{R}^3 \to \mathbb{R}$ whose zero iso-surface approximates the surface $\partial\Omega$. Ψ should have C^1 continuity at the zero iso-surface such that it can be included as a smooth inequality constraint in a dynamics simulation to resolve contacts. We define the approximate solid domain implicitly using Ψ:

$$\tilde{\Omega} := \{\, \boldsymbol{x} \in \mathbb{R}^3 \,:\, \Psi(\boldsymbol{x}) \leq 0 \,\}.$$

As before, we require locality. To achieve this, neighboring local potentials are blended using barycentric weights.

At each sample point $\boldsymbol{s} \in \mathcal{S}$, we assign a local potential field $\psi(\,\cdot\,;\boldsymbol{s}) : \mathbb{R}^3 \to \mathbb{R}$, which can be chosen arbitrarily but must be vanishing at the sample point and have a gradient aligned with $\boldsymbol{n}_{\boldsymbol{s}}$. An example of such a field is

$$\psi(\boldsymbol{x};\boldsymbol{s}) = \boldsymbol{n}_{\boldsymbol{s}}^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{s}).$$

Then for any $\boldsymbol{x} \in \mathbb{R}^3$ we identify a neighborhood of samples close to $\boldsymbol{x}$ and blend the corresponding local potential to form a C^1 continuous global potential field Ψ. In particular, let $\mathcal{N}(\boldsymbol{x}) \subset \mathcal{S}$ be a neighborhood of samples near $\boldsymbol{x}$, then define

$$\Psi(\boldsymbol{x}) = \sum_{\boldsymbol{s}\in\mathcal{N}(\boldsymbol{x})} w(\boldsymbol{x};\boldsymbol{s})\psi(\boldsymbol{x};\boldsymbol{s}), \quad \text{where} \quad \sum_{\boldsymbol{s}\in\mathcal{N}(\boldsymbol{x})} w(\boldsymbol{x};\boldsymbol{s}) = 1.$$

The partition-of-unity criterion allows Ψ to be independent of the size of $\mathcal{N}(\boldsymbol{x})$.

It remains to define the neighborhood $\mathcal{N}$ and the barycentric weight function w. This framework allows us to define a local potential field with the desired properties, and there are many possible options for defining $\mathcal{N}$ and w. Furthermore, this formulation is a specialization of the more general moving least squares (MLS) framework for interpolating and approximating implicit surfaces [Shen et al. 2004]. We use a *locally compact* weight function [Morse et al. 2005; Ohtake et al. 2003; Öztireli et al. 2009], which is critical for our application because we require sparse derivatives.

4.1.1 Weight Function. There are three main properties we require from the weight function:

Locality: $w(\boldsymbol{x};\boldsymbol{s})$ must vanish when $\boldsymbol{x}$ and $\boldsymbol{s}$ are sufficiently far away.

Smoothness: w should have continuous derivatives for the resulting potential to be smooth at the zero iso-surface.

Interpolation: The zero iso-surface of Ψ must pass through (or be close to) the sample points $\mathcal{S}$.

We repurposed the weight function [Most and Bucher 2005], which was originally used for interpolation in element-free Galerkin methods. This weight function is defined as

$$w_R(\boldsymbol{x};\boldsymbol{s}) := \frac{\tilde{w}_R(\|\boldsymbol{x}-\boldsymbol{s}\|_2)}{\sum_{\boldsymbol{s}\in\mathcal{N}_R(\boldsymbol{x})} \tilde{w}_R(\|\boldsymbol{x}-\boldsymbol{s}\|_2)} \tag{9}$$

$$\tilde{w}_R(r) := \frac{\left((r/R)^2+\epsilon\right)^{-2} - (1+\epsilon)^{-2}}{\epsilon^{-2} - (1+\epsilon)^{-2}}, \tag{10}$$

where $\epsilon \ll 1$ determines how closely the potential will approximate the sample points and R is a radius of influence that determines the neighborhood:

$$\mathcal{N}_R(\boldsymbol{x}) := \{\, \boldsymbol{s} \in \mathcal{S} \,:\, \|\boldsymbol{x}-\boldsymbol{s}\|_2 < R \,\}.$$

This weight function is useful when R is large, because ϵ can be reduced to improve the interpolation of the samples in $\mathcal{S}$. However, when R is relatively small, we can use a simpler cubic weight

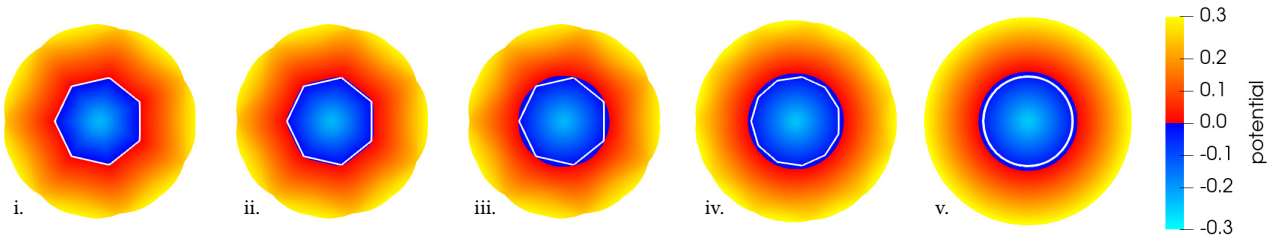


Fig. 3. A cross section of the potential field $\Psi(\boldsymbol{x})$ generated by the cylinders from Figure 6 at various resolutions, with different kernels and parameters. The shape of the cylinder cross section which is used to generate the potential is outlined in white in each of the examples. Starting with the heptagonal cylinder, the potential fields generated by $\tilde{w}_R$ with $\epsilon = 10^{-5}$ (i.), $\tilde{w}_R$ with $\epsilon = 0.1$ (ii.) and finally the potential generated by $\tilde{w}_R^{\text{cubic}}$ (iii.) are shown. Points outside the colored area have empty neighborhoods $\mathcal{N}_R$ and are omitted from the contact solve. When using the $\tilde{w}_R^{\text{cubic}}$ kernel, we can see that increasing the resolution of the cylinder produces a similarly smooth contact surface (iv. and v.).

function given by

$$\tilde{w}_R^{\text{cubic}}(r) := 1 - 3\left(\frac{r}{R}\right)^2 + 2\left(\frac{r}{R}\right)^3.$$

This function is generally smoother than $\tilde{w}_R$ and is faster to compute. However, it is not guaranteed to interpolate the provided samples and it tends to exaggerate large curvature regions — concavities in the samples become deeper and convexities become more protruding. We found that $\tilde{w}_R$ produces better results when local deformations are larger, requiring larger values for R, but for smaller deformations, $\tilde{w}_R^{\text{cubic}}$ is sufficient. Figure 3 shows the potential generated by each kernel for a heptagonal cylinder.

4.1.2 Discretization. As the surface deforms, we need to know how the samples and their corresponding normals change. In other words, we need to be able to compute their derivatives. This calls for a more concrete discretization scheme for the solid domain. Because we use the standard tetrahedral finite element method to model elasticity, we chose to sample the same mesh on the surface to generate our implicit collision surface. We sample the tetrahedral mesh at the centroids of all surface triangles[1]. The sample normals are computed from each triangle $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3)$ as

$$\boldsymbol{n}_{\boldsymbol{s}} = \frac{(\boldsymbol{p}_1 - \boldsymbol{p}_2) \times (\boldsymbol{p}_1 - \boldsymbol{p}_3)}{\|(\boldsymbol{p}_1 - \boldsymbol{p}_2) \times (\boldsymbol{p}_1 - \boldsymbol{p}_3)\|_2}.$$

4.1.3 Contact Jacobian. For frictionless contact problems, the formulation above is sufficient. However, when friction is involved, we must also map tangential force impulses to the corresponding degrees of freedom.

Our generalized coordinates correspond to vertex positions on a tetrahedral mesh. We use this fact to construct the contact Jacobian.

Let $\boldsymbol{Q}_{\boldsymbol{s}}(\boldsymbol{x}) \in \mathbb{R}^{3\times3}$ be a minimal rotation matrix from the coordinate frame at sample point $\boldsymbol{s}$ to the coordinate frame at a contact point $\boldsymbol{x}$ such that

$$\boldsymbol{Q}_{\boldsymbol{s}}(\boldsymbol{x})\boldsymbol{n}_{\boldsymbol{s}} = \nabla_{\boldsymbol{x}}\Psi(\boldsymbol{x}).$$

[1]We could have chosen to sample the mesh at the vertices and use an averaging scheme to compute vertex normals, however this is more complex and would generate denser Jacobian and Hessian matrices.

In words, the matrix $\boldsymbol{Q}_{\boldsymbol{s}}(\boldsymbol{x})$ rotates normals at sample points to align with the gradient at contact points. We can compute $\boldsymbol{Q}_{\boldsymbol{s}}(\boldsymbol{x})$ as

$$\boldsymbol{Q}_{\boldsymbol{s}}(\boldsymbol{x}) = \boldsymbol{I} + [\boldsymbol{z}]_\times + [\boldsymbol{z}]_\times^2 \frac{1}{1 + \boldsymbol{n}_{\boldsymbol{s}} \cdot \nabla_{\boldsymbol{x}}\Psi(\boldsymbol{x})}.$$

where $\boldsymbol{I}$ is the 3×3 identity, $\boldsymbol{z} = \boldsymbol{n}_{\boldsymbol{s}} \times \nabla_{\boldsymbol{x}}\Psi(\boldsymbol{x})$ and $[\boldsymbol{z}]_\times$ is its skew-symmetric matrix form. This matrix allows us to transport any vector at a site $\boldsymbol{s}$ into the coordinate frame of the contact at $\boldsymbol{x}$ such that normal and tangent vectors at $\boldsymbol{s}$ remain normal and tangent respectively at $\boldsymbol{x}$.

Now if $\bar{\mathbf{s}} \in \mathbb{R}^{3|\mathcal{S}|}$ is a stacked vector of sample points and $\bar{\boldsymbol{p}}$ is the stacked vector of mesh surface vertex positions, then we can write the contact Jacobian at each contact i as

$$\boldsymbol{J}_i = \frac{\partial \boldsymbol{x}_i}{\partial \mathbf{q}} = \frac{\partial \boldsymbol{x}_i}{\partial \bar{\boldsymbol{p}}}\frac{\partial \bar{\boldsymbol{p}}}{\partial \mathbf{q}} = \frac{\partial \boldsymbol{x}_i}{\partial \bar{\mathbf{s}}}\frac{\partial \bar{\mathbf{s}}}{\partial \bar{\boldsymbol{p}}}\frac{\partial \bar{\boldsymbol{p}}}{\partial \mathbf{q}},$$

which allows us to use the kernel weights to intepolate velocity contributions from mesh vertices. Since $\mathbf{q}$ corresponds to mesh vertices and $\bar{\boldsymbol{p}}$ is a subset of those on the surface, we have that $\frac{\partial \bar{\boldsymbol{p}}}{\partial \mathbf{q}}$ is a simple selection matrix. In our case, sample points are located at surface triangle centroids, which means that $\frac{\partial \boldsymbol{s}}{\partial \bar{\boldsymbol{p}}}$ is the identity matrix scaled by $1/3$ if $\boldsymbol{p}$ is a vertex of the triangle for sample $\boldsymbol{s}$ and zero otherwise. It follows that $\frac{\partial \bar{\mathbf{s}}}{\partial \bar{\boldsymbol{p}}}$ is a sparse block matrix of scaled 3×3 identity matrices.

Finally we can define

$$\frac{\partial \boldsymbol{x}_i}{\partial \bar{\mathbf{s}}} = \begin{cases} w_R(\boldsymbol{x}_i; \boldsymbol{s})\boldsymbol{Q}_{\boldsymbol{s}}(\boldsymbol{x}_i) & \text{if } \boldsymbol{s} \in \mathcal{N}_R(\boldsymbol{x}_i) \\ \mathbf{0} & \text{otherwise} \end{cases},$$

which is a block matrix of scaled rotations. The sparsity of $\frac{\partial \boldsymbol{x}_i}{\partial \bar{\mathbf{s}}}$ is determined by the choice of $\mathcal{N}_R$.

4.1.4 Choosing R. Note that the discontinuity at $r = R$ is only problematic where the potential is negative (inside the solid), since it produces a non-smooth contact boundary. The optimization is not affected by discontinuities at positive potential values away from the boundary. Thus we must ensure that the MLS surface has no holes to prevent points from contacting the non-smooth boundary inside the solid. Generally we enforce this manually by setting the radius R to be larger than half of the largest triangle diameter in the input mesh. However, an adaptive scheme can produce a variable

R depending on local feature size and maximum expected relative velocities.

5 FRICTIONAL CONTACT

In this section we will prescribe the necessary constraints to produce a frictional contact response satisfying the Coulomb friction model. We focus on Coulomb friction since it is an integral part of many other friction models like Stribeck, Dahl, LuGre, and elastoplastic friction [Liu et al. 2015].

5.1 Contact Space

Let Γ_c denote the index set of all points of contact. For any vector $\boldsymbol{y} \in \mathbb{R}^3$ in physical space associated with a contact point from Γ_c, we define y_N and $\boldsymbol{y}_\mathrm{T}$ to be the normal and tangential components of $\boldsymbol{y}$ with respect to the contact surface. We call y_N and $\boldsymbol{y}_\mathrm{T}$ contact space coordinates. In general we use italic sans-serif font for quantities in contact space coordinates, which implies a unique orthogonal change of basis for each contact point. This notation is used consistently in the following sections to describe velocities and impulses at the point of contact.

To make this precise we designate a change of basis matrix $\boldsymbol{B} = [\boldsymbol{B}_\mathrm{N} \,|\, \boldsymbol{B}_\mathrm{T}]$ where $\boldsymbol{B}_\mathrm{N}$ and $\boldsymbol{B}_\mathrm{T}$ are block diagonal matrices with:

$$[\boldsymbol{B}_\mathrm{N}]_{i,i} := \boldsymbol{n}_i \quad \text{and} \quad [\boldsymbol{B}_\mathrm{T}]_{i,i} := [\boldsymbol{t}_i \,|\, \boldsymbol{n}_i \times \boldsymbol{t}_i]$$

where $\boldsymbol{n}_i$ is the unit normal and $\boldsymbol{t}_i$ is a tangent direction at contact point $i \in \Gamma_c$. This allows us to write the normal and tangential components of the stacked vector $\boldsymbol{y}$ as $y_\mathrm{N} := \boldsymbol{B}_\mathrm{N}^\top \boldsymbol{y}$ and $\boldsymbol{y}_\mathrm{T} := \boldsymbol{B}_\mathrm{T}^\top \boldsymbol{y}$ respectively.

5.2 The Coulomb Friction Model

We start with a disjunctive description of the Coulomb friction model [Bertails-Descoubes et al. 2011; Daviet et al. 2011]. The *friction cone*, which confines the frictional contact impulse on the solid is defined by

$$K_\mu := \left\{ \boldsymbol{r} = (r_\mathrm{N}, \boldsymbol{r}_\mathrm{T}) \in \mathbb{R} \times \mathbb{R}^2 \,:\, r_\mathrm{N} \geq 0,\, \boldsymbol{r}_\mathrm{T} \in \mu r_\mathrm{N} D \right\},$$

where $D := \{\boldsymbol{y} \in \mathbb{R}^2 : \|\boldsymbol{y}\| < 1\}$ is the open disc centered about the origin. To formalize the Coulomb friction model, consider a

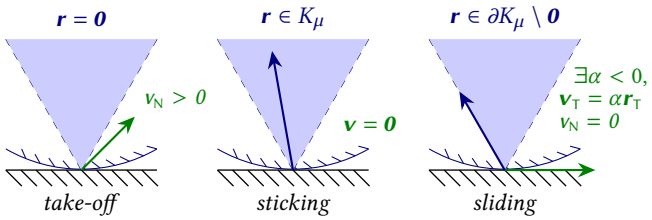


Fig. 4. *Coulomb friction model.* A stationary object (bottom surface in black) is in contact with a moving object (curved surface in blue). The conditions on the friction impulse $\boldsymbol{r}$ and velocity $\boldsymbol{v}$ can be described as a disjunction of three cases as illustrated where the normal is assumed to be pointing up.

single point of contact where the deforming solid is above and a fixed surface is below. In this scenario as illustrated in Figure 4, the Coulomb friction law can be defined by a disjunction of three cases:

$$\boldsymbol{r} = \boldsymbol{0} \text{ and } v_\mathrm{N} > 0 \qquad (\textit{take-off})$$
$$\boldsymbol{r} \in K_\mu \text{ and } \boldsymbol{v} = \boldsymbol{0} \text{ and} \qquad (\textit{sticking})$$
$$\boldsymbol{r} \in \partial K_\mu \setminus \boldsymbol{0},\ v_\mathrm{N} = 0 \text{ and } \boldsymbol{v}_\mathrm{T} \in \{\alpha \boldsymbol{r}_\mathrm{T} : \alpha < 0\}. \qquad (\textit{sliding})$$

To be precise we will refer to $\boldsymbol{r}_\mathrm{T}$ as the frictional impulse and r_N as the contact impulse in the following sections.

5.3 Contact

If for a moment, we ignore the force of friction, (assume $\mu = 0$), then the cases (*take-off*), (*sticking*), and (*sliding*) can be distilled into $\boldsymbol{r}_\mathrm{T} = \boldsymbol{0}$ and the complementarity condition

$$0 \leq r_\mathrm{N} \perp v_\mathrm{N} \geq 0 \quad \text{on } \Gamma_c, \qquad \text{(SFC)}$$

which indicates that only one of r_N and v_N may be non-zero at any given time. This is called the Signorini-Fichera Condition (SFC) [Fichera 1964; Signorini 1933].

Assuming that $\boldsymbol{r}_\mathrm{T}^{\mathrm{t}+\Delta\mathrm{t}} = \boldsymbol{0}$, equation (7) with (SFC) become the Karush-Kuhn-Tucker (KKT) conditions for the following constrained optimization problem

$$\begin{aligned} \min_{\mathbf{u}} \quad & \gamma(\mathbf{u}) \\ \text{subject to} \quad & \boldsymbol{v}_\mathrm{N} \geq \boldsymbol{0}, \end{aligned} \qquad (11)$$

where $\gamma(\mathbf{u}) := \frac{1}{2}\left\|\dot{\mathbf{q}}^{\mathrm{t}} - \mathbf{u}\right\|_{\mathbf{M}}^2 + W(\mathbf{q}^{\mathrm{t}} + \Delta\mathrm{t}\,\mathbf{u})$ and $\boldsymbol{v}_\mathrm{N} = \boldsymbol{B}_\mathrm{N}^\top \boldsymbol{J}\mathbf{u}$. The dual form of equation (11) reveals the role of the contact impulse $\boldsymbol{r}_\mathrm{N}$ as the Lagrange multiplier:

$$(\boldsymbol{r}_\mathrm{N}^{\mathrm{t}+\Delta\mathrm{t}}, \dot{\mathbf{q}}^{\mathrm{t}+\Delta\mathrm{t}}) = \arg\max_{\boldsymbol{\lambda} \geq \boldsymbol{0}} \min_{\mathbf{u}} \gamma(\mathbf{u}) - \boldsymbol{\lambda}^\top \boldsymbol{v}_\mathrm{N}, \qquad (12)$$

where $\boldsymbol{r}_\mathrm{N}^{\mathrm{t}+\Delta\mathrm{t}}$ is the optimal value of the Lagrange multiplier $\boldsymbol{\lambda}$ and $\dot{\mathbf{q}}^{\mathrm{t}+\Delta\mathrm{t}}$ is the optimal value of the generalized velocity $\mathbf{u}$ as before. The dual form in (12) is equivalent to (8) with a maximization over positive contact impulses.

We use the dual form in our exposition since it explicitly lists inputs and outputs of the optimization problem, however in our implementation we aim to solve a constrained minimization.

5.3.1 Non-Penetration Constraint. To enforce impenetrability, we constrain a selection of vertices on one simulation mesh k to lie outside the collision surface of all meshes except mesh k itself represented by $\Psi[\mathbf{q}_{\bar{k}}]$ (defined in Section 4) where $\mathbf{q}_{\bar{k}}$ are all the coordinates in $\mathbf{q}$ that do *not* affect the deformation of mesh k. We limit ourselves to contact between disconnected meshes to ensure this mutual exclusion. Let $\mathcal{V}_k$ be the set of all vertices of mesh k to be constrained. Then the non-penetration constraint can be written as

$$\Psi[\mathbf{q}_{\bar{k}}](\boldsymbol{p}) \geq 0 \qquad \forall \boldsymbol{p} \in \mathcal{V}_k. \qquad (13)$$

To simplify the following sections we group all constraints into one vector $\boldsymbol{\phi}(\mathbf{q})$ with the following component-wise definition:

$$\boldsymbol{\phi}_{k,i}(\mathbf{q}) := \Psi[\mathbf{q}_{\bar{k}}](\boldsymbol{p}_i) \qquad \forall \boldsymbol{p}_i \in \mathcal{V}_k. \qquad (14)$$

This allows us to write the constraint simply as

$$\boldsymbol{\phi}(\mathbf{q}) \geq \mathbf{0}. \qquad (15)$$

The union of all constrained vertex sets $\mathcal{V}_k$ forms the complete set of contact points in Γ_c. Finally with the non-penetration constraint

substituting (SFC), the problem becomes

$$(\boldsymbol{r}_{\mathrm{N}}^{\mathrm{t}+\Delta\mathrm{t}}, \dot{\mathbf{q}}^{\mathrm{t}+\Delta\mathrm{t}}) = \arg\max_{\boldsymbol{\lambda}\geq\mathbf{0}} \min_{\mathbf{u}} \gamma(\mathbf{u}) - \frac{1}{\Delta\mathrm{t}}\boldsymbol{\lambda}^{\top}\boldsymbol{\phi}(\mathbf{q}^{\mathrm{t}} + \Delta\mathrm{t}\,\mathbf{u}), \tag{16}$$

where we apply backward Euler to $\boldsymbol{\phi}$ to be consistent with the elastic energy computation in $\gamma(\mathbf{u})$.

5.4 Friction

Now suppose that $\mu \geq 0$. This means that the frictional contact impulse must satisfy the complete Coulomb law we introduced in Section 5.2. As we mentioned before, $\boldsymbol{r}^{\mathrm{t}+\Delta\mathrm{t}}$ cannot be derived from a single potential but it can be derived by minimizing two coupled potentials [De Saxcé and Feng 1998]. We can use the result we developed in the previous section but reintroduce the unknown tangential impulse $\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}+\Delta\mathrm{t}}$ into the non-penetration problem (16):

$$\begin{aligned}(\boldsymbol{r}_{\mathrm{N}}^{\mathrm{t}+\Delta\mathrm{t}}, \dot{\mathbf{q}}^{\mathrm{t}+\Delta\mathrm{t}}) = &\arg\max_{\boldsymbol{\lambda}\geq\mathbf{0}} \min_{\mathbf{u}} \gamma(\mathbf{u}) \\ &- \frac{1}{\Delta\mathrm{t}}\boldsymbol{\lambda}^{\top}\boldsymbol{\phi}(\mathbf{q}^{\mathrm{t}} + \Delta\mathrm{t}\,\mathbf{u}) - \boldsymbol{v}_{\mathrm{T}}^{\top}\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}+\Delta\mathrm{t}},\end{aligned} \tag{17}$$

where $\boldsymbol{v}_{\mathrm{T}} = \boldsymbol{B}_{\mathrm{T}}^{\top}\boldsymbol{J}\mathbf{u}$. The remaining potential is responsible for resolving the friction impulse $\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}+\Delta\mathrm{t}}$.

5.4.1 Maximal Dissipation Principle. For single point contact conditions, frictional impulses for sliding can be derived from the *Maximal Dissipation Principle* (MDP) [Moreau 1973]. MDP states that the frictional impulse at a point of contact maximizes negative work:

$$\boldsymbol{r}_{\mathrm{T}} = \operatorname*{argmax}_{\boldsymbol{y}\in\mu r_{\mathrm{N}}\bar{D}} -\boldsymbol{y}\cdot\boldsymbol{v}_{\mathrm{T}} = \operatorname*{argmin}_{\boldsymbol{y}\in\mu r_{\mathrm{N}}\bar{D}} \boldsymbol{y}\cdot\boldsymbol{v}_{\mathrm{T}}, \tag{18}$$

where $\bar{D}$ is the closed unit disc and $\boldsymbol{v}_{\mathrm{T}}$ is the tangential relative velocity vector. The minimization in (18) is trivially solved by

$$\boldsymbol{r}_{\mathrm{T}} = -\mu r_{\mathrm{N}}\frac{\boldsymbol{v}_{\mathrm{T}}}{\|\boldsymbol{v}_{\mathrm{T}}\|} \tag{19}$$

on the disc when $\boldsymbol{v}_{\mathrm{T}} \neq \mathbf{0}$.

Equation (19) is in fact exactly the equation we get from the (*sliding*) condition. To see this, note that since $\boldsymbol{v}_{\mathrm{T}} = \alpha\boldsymbol{r}_{\mathrm{T}}$ for some $\alpha < 0$, then it must be that $\alpha = -\|\boldsymbol{v}_{\mathrm{T}}\|/\|\boldsymbol{r}_{\mathrm{T}}\|$. Then substituting in the $\|\boldsymbol{r}_{\mathrm{T}}\| = \mu r_{\mathrm{N}}$ condition from $\boldsymbol{r} \in \partial K_{\mu}$, we get exactly (19). In addition, if $\boldsymbol{v}_{\mathrm{T}} = \mathbf{0}$, then all of $\mu r_{\mathrm{N}}\bar{D}$ solves equation (18), which corresponds to (*sticking*). Thus the (MDP) condition along with (SFC), which handles (*take-off*), corresponds to the whole Coulomb model introduced in Section 5.2. This observation connects the Coulomb friction formulation using (SFC) and (MDP) [Erleben 2017; Kaufman et al. 2008] to the disjunctive formulation using (*take-off*), (*sticking*) and (*sliding*) [Bertails-Descoubes et al. 2011; Daviet et al. 2011; Li et al. 2018].

We can now extend (18) to all the contacts in Γ_c. First, let $n := |\Gamma_c|$ denote the total number contact points in the system. We define $\bar{D}_{\Gamma_c} : \mathbb{R}^n \to \mathbb{R}^{2n}$ to be the map from stacked contact impulses to a cartesian product of closed discs in the tangent space of all contact points in Γ_c:

$$\bar{D}_{\Gamma_c}(\boldsymbol{y}) := \prod_{i\in\Gamma_c} \boldsymbol{y}_i\bar{D} \subset \mathbb{R}^{2n}$$

where $\boldsymbol{y}_i \in \mathbb{R}$ is the ith component of $\boldsymbol{y}$. Then (18) can be rewritten for all contact points and in terms of generalized velocities as

$$\boldsymbol{r}_{\mathrm{T}} = \operatorname*{argmin}_{\boldsymbol{y}\in\bar{D}_{\Gamma_c}(\mu\boldsymbol{r}_{\mathrm{N}})} \boldsymbol{y}^{\top}\boldsymbol{B}_{\mathrm{T}}^{\top}\boldsymbol{J}\dot{\mathbf{q}}. \tag{MDP}$$

5.5 Frictional Contact Solutions

The remaining question is how to simultaneously solve the contact and friction problems in (17) and (MDP) respectively. It is well known that finding a global optimum for both optimization problems is NP-Hard [Kaufman et al. 2008], however we can focus on finding an approximation to the global optimum.

5.5.1 Time-Splitting. Elastic impulses move through a typical soft solid at a much slower rate than frictional contact impulses [Bertails-Descoubes et al. 2011; Daviet et al. 2011; Kaufman et al. 2008; Li et al. 2018]. For instance, consider a rubber ball bouncing against the solid ground. During the time step when contact initially occurs between the ball and the ground, the contact impulse is felt immediately by the ball, causing it to deform to avoid penetrating through the floor. Given that the ball is soft enough, it takes multiple time steps for elasticity to reverse the velocity field on every point on the ball such that it bounces back. This may justify splitting the solve for elasticity from frictional contact in certain scenarios.

However, implicit time integration allows us to take large time steps, within which elastic and frictional contact impulses may be comparable in magnitude. This is further exacerbated when the hyperelastic material is stiff.

Furthermore, as previously mentioned in Section 2, friction and contact must be resolved simultaneously.

These observations motivate a novel *predictor-corrector* scheme (otherwise known as *time-splitting*) for solving the coupled variational problem in (17) and (MDP). Instead of splitting the entire frictional contact solve completely from the elasticity optimization [Bertails-Descoubes et al. 2011; Kaufman et al. 2008], we opt to split only the incremental update to the friction impulse. We call this mechanism *friction forwarding* since it effectively forwards the computed friction impulses from one time step to the next as an external force. In other words, we propose a three step algorithm:

1. Compute the intermediate (predictor) velocity $\dot{\mathbf{q}}^{*}$ and contact impulse $\boldsymbol{r}_{\mathrm{N}}^{*}$ by solving the elasticity problem subject to contact constraints as well as friction impulses from the previous time step:

$$(\boldsymbol{r}_{\mathrm{N}}^{*}, \dot{\mathbf{q}}^{*}) = \arg\max_{\boldsymbol{\lambda}\geq\mathbf{0}} \min_{\mathbf{u}} \gamma(\mathbf{u}) - \frac{1}{\Delta\mathrm{t}}\boldsymbol{\lambda}^{\top}\boldsymbol{\phi}(\mathbf{q}^{\mathrm{t}} + \Delta\mathrm{t}\,\mathbf{u}) - \boldsymbol{v}_{\mathrm{T}}^{\top}\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}}. \tag{20}$$

2. Using $\boldsymbol{r}_{\mathrm{N}}^{*}$ as the initial guess for the contact impulse, solve the modified staggered projections as described in §5.5.2, to obtain the full frictional contact impulse

$$\boldsymbol{r}^{\mathrm{t}+\Delta\mathrm{t}} = (\boldsymbol{B}_{\mathrm{T}}\boldsymbol{r}_{\mathrm{N}}^{\mathrm{t}+\Delta\mathrm{t}}, \boldsymbol{B}_{\mathrm{N}}\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}+\Delta\mathrm{t}}), \tag{21}$$

3. Update predictor velocity with the friction impulse corrector:

$$\dot{\mathbf{q}}^{\mathrm{t}+\Delta\mathrm{t}} = \dot{\mathbf{q}}^{*} + \boldsymbol{M}^{-1}\boldsymbol{J}^{\top}(\boldsymbol{r}^{\mathrm{t}+\Delta\mathrm{t}} - \boldsymbol{r}^{\mathrm{t}}) \tag{22}$$

To solve the first step as written, we would require a non-linear optimization solver capable of enforcing non-linear inequality constraints. This requirement can be relaxed by linearizing the contact

constraint, converting (20) into

$$(\boldsymbol{r}_{\mathrm{N}}^{*}, \dot{\mathbf{q}}^{*}) = \arg\max_{\boldsymbol{\lambda}\geq\boldsymbol{0}} \min_{\mathbf{u}} \gamma(\mathbf{u}) - \boldsymbol{\lambda}^{\top}\left(\frac{\partial \boldsymbol{\phi}^{\mathrm{t}}}{\partial \mathbf{q}}\mathbf{u} + \frac{1}{\Delta \mathrm{t}}\boldsymbol{\phi}^{\mathrm{t}}\right) - \boldsymbol{v}_{\mathrm{T}}^{\top}\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}}, \quad (23)$$

where $\boldsymbol{\phi}^{\mathrm{t}} = \boldsymbol{\phi}(\mathbf{q}^{\mathrm{t}})$.

Note that friction forwarding is distinct from warm-start techniques [Kaufman et al. 2008]. Our method uses the friction from the previous time step to compute the velocity predictor $\dot{\mathbf{q}}^{*}$, whereas Kaufman et al. [2008] use the friction from the previous step only as a starting guess for the staggered projections algorithm.

5.5.2 Modified Staggered Projections. In this section we will describe how staggered projections can be solved efficiently in contact space without linearizing the friction cone defined by $\bar{D}_{\Gamma_c}$.

First, let us define the projection operator $P_{\mathrm{T}} : \mathbb{R}^{n} \times \mathbb{R}^{3n} \to \mathbb{R}^{2n}$ of stacked impulses in physical space onto the set of contact tangent discs as follows

$$P_{\mathrm{T}}(\boldsymbol{r}_{\mathrm{N}}; \boldsymbol{\xi}) := \operatorname*{argmin}_{\boldsymbol{y}\in\bar{D}_{\Gamma_c}(\mu\boldsymbol{r}_{\mathrm{N}})} \frac{1}{2}\|\boldsymbol{B}_{\mathrm{T}}\boldsymbol{y} - \boldsymbol{\xi}\|^{2}_{\boldsymbol{M}_e^{-1}}. \quad (24)$$

where $\boldsymbol{M}_e^{-1} := \boldsymbol{J}\mathbf{M}^{-1}\boldsymbol{J}^{\top}$ is the inverse of the *effective mass* (Delassus operator) at contact points. Similarly we can define the projection of impulses onto the set of contact normals:

$$P_{\mathrm{N}}(\boldsymbol{\xi}) := \operatorname*{argmin}_{\boldsymbol{y}\geq\boldsymbol{0}} \frac{1}{2}\|\boldsymbol{B}_{\mathrm{N}}\boldsymbol{y} - \boldsymbol{\xi}\|^{2}_{\boldsymbol{M}_e^{-1}}. \quad (25)$$

Given these two projection operators we can write down the staggered projections scheme as

$$\boldsymbol{r}_{\mathrm{T}}^{k} \leftarrow P_{\mathrm{T}}\left(\boldsymbol{r}_{\mathrm{N}}^{k-1}; \boldsymbol{z} - \boldsymbol{B}_{\mathrm{N}}\boldsymbol{r}_{\mathrm{N}}^{k-1}\right) \quad (26a)$$

$$\boldsymbol{r}_{\mathrm{N}}^{k} \leftarrow P_{\mathrm{N}}\left(\boldsymbol{z} - \boldsymbol{B}_{\mathrm{T}}\boldsymbol{r}_{\mathrm{T}}^{k}\right) \quad (26b)$$

starting with $\boldsymbol{r}_{\mathrm{N}}^{0} = \boldsymbol{r}_{\mathrm{N}}^{*}$ and a constant predictor

$$\boldsymbol{z} = \boldsymbol{B}_{\mathrm{N}}\boldsymbol{r}_{\mathrm{N}}^{*} + \boldsymbol{B}_{\mathrm{T}}\left(\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}} - \boldsymbol{B}_{\mathrm{T}}^{\top}\boldsymbol{M}_e\boldsymbol{J}\dot{\mathbf{q}}^{*}\right). \quad (27)$$

Our predictor is different from standard staggered projections. We construct it from the normal and tangential components. The contact impulse $\boldsymbol{r}_{\mathrm{N}}^{*}$ is computed in (20) as a Lagrange multiplier and can be used directly as the normal component of the predictor. The normal component of $-\boldsymbol{M}_e\boldsymbol{J}\dot{\mathbf{q}}^{*}$ is ignored since the non-penetration is already solved by the contact constraint. Finally any tangential frictional impulses from the previous step must be added to $-\boldsymbol{B}_{\mathrm{T}}\boldsymbol{M}_e\boldsymbol{J}\dot{\mathbf{q}}^{*}$ to ensure that the predictor contains the full frictional impulse.

Unfortunately as written, (24) is a quadratic problem subject to non-linear inequality constraints. As such, popular methods for friction simulation [Kaufman et al. 2008; Otaduy et al. 2009] often discretize the contact tangent discs to linearize inequality constraints. This approach works, but can produce visible artifacts when the resolution of the disc discretization is low [Li et al. 2018]. To maintain both speed and accuracy, we choose to reparametrize the contact space into cylindrical coordinates instead, producing a non-linear problem with bounds constraints. This allows us to rewrite (24) as

$$P_{\mathrm{T}}(\boldsymbol{r}_{\mathrm{N}}; \boldsymbol{\xi}) := \operatorname*{argmin}_{\substack{\boldsymbol{\theta}\in[0,2\pi]^{n} \\ \boldsymbol{0}\leq\boldsymbol{\alpha}\leq\mu\boldsymbol{r}_{\mathrm{N}}}} \frac{1}{2}\|\boldsymbol{B}_{\mathrm{T}}\boldsymbol{R}(\boldsymbol{\theta}, \boldsymbol{\alpha}) - \boldsymbol{\xi}\|^{2}_{\boldsymbol{M}_e^{-1}}, \quad (28)$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ are stacked vectors of angles and radii in the tangent plane of each contact respectively. Then $\boldsymbol{R} : [0, 2\pi]^{n} \times [0, \infty)^{n} \to \mathbb{R}^{2n}$ is a non-linear reparametrization operator that can be defined per contact as $\boldsymbol{R}(\theta_i, \alpha_i)_i = (\alpha_i \cos(\theta_i), \alpha_i \sin(\theta_i))$ for all $1 \leq i \leq n$.

The complete procedure is shown in Algorithm 1.

ALGORITHM 1: Velocity Step

Input:

$k_{max} \leftarrow$ maximum number of friction solve iterations allowed
$\epsilon \leftarrow$ tolerance for relative friction error
$\Delta \mathrm{t} \leftarrow$ current time step
$\boldsymbol{M}_e \leftarrow$ effective mass (Delassus operator)
$\mathbf{M} \leftarrow$ generalized mass matrix
$\mathbf{q}^{\mathrm{t}}, \dot{\mathbf{q}}^{\mathrm{t}} \leftarrow$ previous generalized positions and velocities
$\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}} \leftarrow$ previous friction impulse

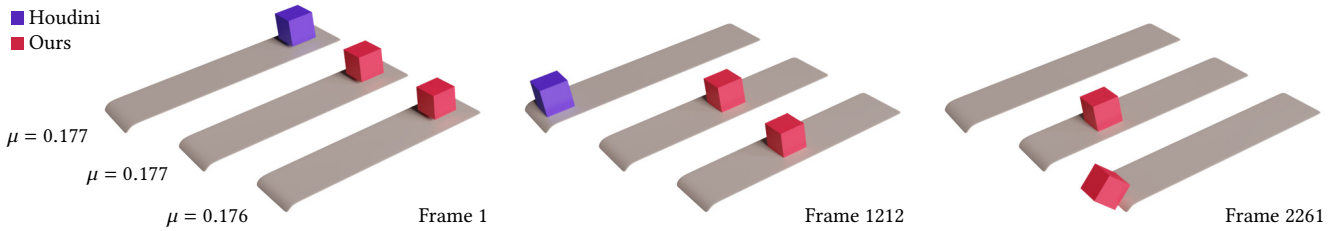
Output: $\dot{\mathbf{q}}^{\mathrm{t}+\Delta \mathrm{t}} \leftarrow$ generalized velocity for the next time step

begin Solve the constrained optimization problem (23):
  $(\boldsymbol{r}_{\mathrm{N}}^{*}, \dot{\mathbf{q}}^{*}) \leftarrow \begin{cases} \arg\min_{\mathbf{u}} & \gamma(\mathbf{u}) - \boldsymbol{v}_{\mathrm{T}}^{\top}\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}} \\ \text{s.t.} & \frac{\partial \boldsymbol{\phi}^{\mathrm{t}}}{\partial \mathbf{q}}\mathbf{u} + \frac{1}{\Delta \mathrm{t}}\boldsymbol{\phi}^{\mathrm{t}} \geq \mathbf{0} \end{cases}$
end
$\boldsymbol{z} \leftarrow \boldsymbol{B}_{\mathrm{N}}\boldsymbol{r}_{\mathrm{N}}^{*} + \boldsymbol{B}_{\mathrm{T}}\left(\boldsymbol{r}_{\mathrm{T}}^{\mathrm{t}} - \boldsymbol{B}_{\mathrm{T}}^{\top}\boldsymbol{M}_e\boldsymbol{J}\dot{\mathbf{q}}^{*}\right)$ `/* predictor impulse (27) */`
begin Solve friction problem to convergence:
  $\boldsymbol{r}_{\mathrm{N}}^{0} \leftarrow \boldsymbol{r}_{\mathrm{N}}^{*}$
  for $k \leftarrow 1$ **to** k_{max} **do**
    $\boldsymbol{r}_{\mathrm{T}}^{k} \leftarrow P_{\mathrm{T}}\left(\boldsymbol{r}_{\mathrm{N}}^{k-1}; \boldsymbol{z} - \boldsymbol{B}_{\mathrm{N}}\boldsymbol{r}_{\mathrm{N}}^{k-1}\right)$
    $\boldsymbol{r}_{\mathrm{N}}^{k} \leftarrow P_{\mathrm{N}}\left(\boldsymbol{z} - \boldsymbol{B}_{\mathrm{T}}\boldsymbol{r}_{\mathrm{T}}^{k}\right)$
    $\boldsymbol{r}_{\mathrm{T}}^{k} \leftarrow \boldsymbol{B}_{\mathrm{T}}\boldsymbol{r}_{\mathrm{T}}^{k}$
    $err \leftarrow \frac{\|\boldsymbol{r}_{\mathrm{T}}^{k} - \boldsymbol{r}_{\mathrm{T}}^{k-1}\|^{2}_{\boldsymbol{M}_e^{-1}}}{\|\boldsymbol{r}_{\mathrm{T}}^{k-1}\|^{2}_{\boldsymbol{M}_e^{-1}}}$
    if $err < \epsilon$ **then**
      break
    end
  end
end
$\boldsymbol{r}^{\mathrm{t}+\Delta \mathrm{t}} \leftarrow \boldsymbol{B}_{\mathrm{T}}\boldsymbol{r}_{\mathrm{T}}^{k} + \boldsymbol{B}_{\mathrm{N}}(\boldsymbol{r}_{\mathrm{N}}^{k} - \boldsymbol{r}_{\mathrm{N}}^{*})$ `/* (21) */`
$\dot{\mathbf{q}}^{\mathrm{t}+\Delta \mathrm{t}} \leftarrow \dot{\mathbf{q}}^{*} + \mathbf{M}^{-1}\boldsymbol{J}^{\top}(\boldsymbol{r}^{\mathrm{t}+\Delta \mathrm{t}} - \boldsymbol{r}^{\mathrm{t}})$ `/* (22) */`

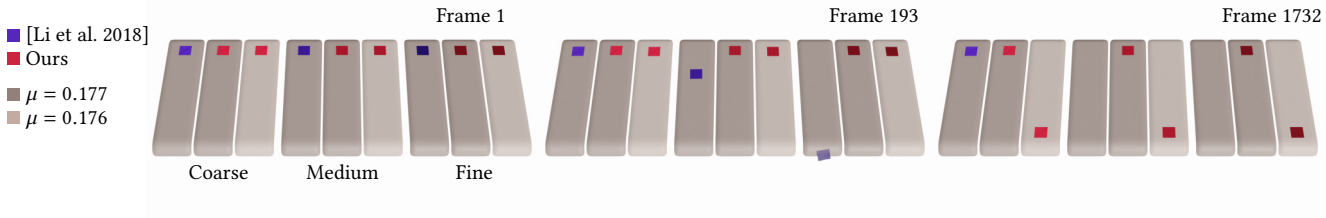
6 RESULTS

The following examples are generated using our Houdini 18 [SideFX 2018] plugin written mainly in the Rust programming language [Matsakis and Klock 2014] and backed by the interior point optimization package, Ipopt [Wächter and Biegler 2006]. Rust provided us with additional confidence in our numerical results due to its strong memory-safety guarantees without sacrificing performance. Ipopt allowed us to focus on the core method without the need for a custom constrained optimization solver. We used Intel MKL with the PARDISO direct linear solver as a backend for Ipopt.

For nearest neighbor lookup, we used an off-the-shelf R*-tree [Altmayer 2020] with bulk loading. The tree is rebuilt for each iteration where the implicit surface samples are updated.



(a) *Block Slide.* A simulation of a soft FEM block sliding across the ramp. Houdini's simulation fails to produce adequate friction forces to stop the block from sliding off at $\mu = 0.177$ (top). Our method successfully stops the block from sliding off at $\mu = 0.177$ (middle) and expectedly allows it to slide off at $\mu = 0.176$ (bottom).



(b) *Cloth Slide.* A square piece of cloth with varying resolutions slides down the ramp with varying coefficients of friction as shown. The lightly colored ramp has a coefficient of friction of $\mu = 0.176$, while the darker ramps have a higher coefficient at $\mu = 0.177$. While [Li et al. 2018] produces stopping at the coarsest level (left-most ramp), the simulation fails to produce adequate friction forces for higher resolutions. (Not shown) On the $\mu = 0.176$ ramp, the method by [Li et al. 2018] results in slipping at all resolutions albeit with significantly different accelerations. Our method (middle and right-most ramp) produces a consistent friction behavior regardless of resolution as shown.

Fig. 5. *Ramp Slide.* Simulations of objects sliding on a rigid ramp inclined at exactly 10 degrees with the horizontal. The physical minimum friction coefficient required for sticking as computed from the ramp incline is $\mu = \tan(\pi/18) \approx 0.176326$.

Some of our results are compared with Houdini's FEM cloth solver to demonstrate how our method measures up to a popular industrial FEM implementation, which could be used to solve similar problems. It is worth noting that Houdini's contact model uses penalty forces. To establish a point of reference with academic works, we compare also against the method by Li et al. [2018] where we disable cloth remeshing to produce an identical cloth mesh between the two methods. We plan to release our simulator as open source software to encourage further development and improve reproducibility of our results.

All examples were run on the AMD Ryzen Threadripper 1920X CPU with 12 cores, 24 threads at 3.7 GHz boost clock and 32GB RAM. We used Blender 2.8 [2019] and ParaView 5.7 [Ahrens et al. 2005] for all generated images and videos.

We used $\epsilon = 10^{-4}$ in all simulations. We pruned contacts with a potential value (divided by bounding box size) of greater than 10^{-4} from the friction solve.

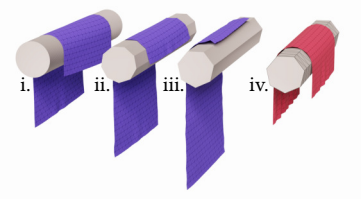
Projections (25) and (28) are solved to convergence with Ipopt using a residual tolerance of 10^{-10}. Elasticity and dynamics in (23) is also solved with Ipopt with a tolerance of 10^{-9}. We rescaled variables and objective functions such that all first order derivatives are close to 1 in magnitude.

In examples involving cloth, it is always the volumetric solid that produces the implicit contact field and the cloth collides at cloth vertices.

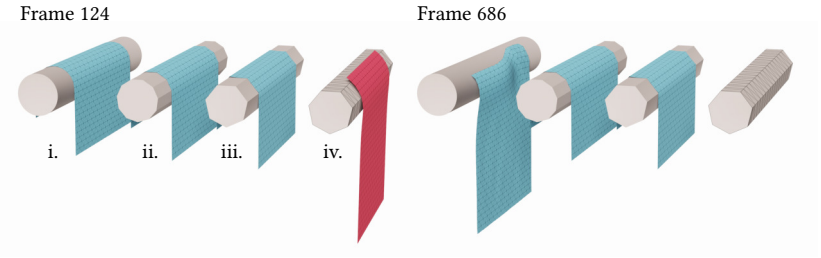
Ramp slide. A soft block made from 320 tetrahedra slides down a slope at $\theta = 10$ degrees from the horizontal with varying coefficients of friction in Figure 5a. This example demonstrates the ability of our method of producing precise expected friction response. Our method is able to stop the cube from sliding off the ramp with a value of μ within 0.001 of the expected stopping coefficient $\mu = \tan(\theta\pi/180)$.

A similar experiment is performed with cloth to compare our method to [Li et al. 2018] using their implementation. While stopping is observed at $\mu = 0.177$ with their method in certain configurations, even modest scaling ratios can produce vastly different results at larger time steps. For instance, we compare how a square piece of cloth of varying resolutions slides down a long ramp in Figure 5b. In contrast to [Li et al. 2018] we are able to produce consistent friction behavior at all resolutions.

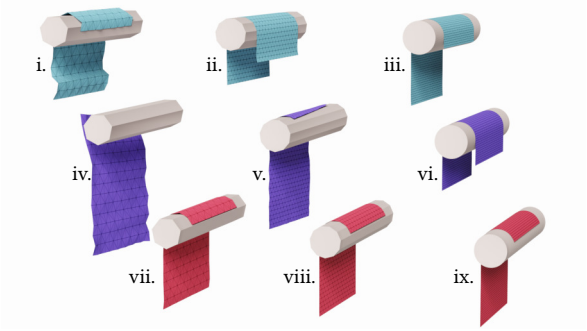
Cloth on a cylinder. In Figure 6 we drape a patch of cloth on a cylinder to demonstrate the contact fidelity of our method and compare it against [Li et al. 2018] and Houdini's FEM cloth simulation. Figure 6a demonstrates the artifacts produced by standard polygonal collision detection and response schemes, and shows how our method overcomes these limitations. Polygon based contact schemes require fine resolutions to improve contact fidelity, whereas our approach naturally produces smooth contact surfaces even with coarse discretizations. In this example, our method allows the cloth to stick to the cylinder. While Houdini's simulations



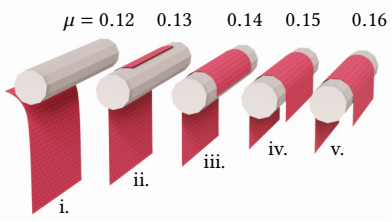
(a) *Sticking.* First three cylinders from the left show Houdini's cloth simulation at different cylinder discretizations. The last cylinder demonstrates our cloth simulation. This example uses $\mu = 0.2$, and the cloth is initially offset with 0.8m vertical distance between the ends. Our method allows the cloth to stick, while Houdini's fails to stick even at higher resolutions. Note that the mesh must be sampled uniformly in order to produce an implicit surface without bumps, as shown by the uniform triangulation of the last cylinder. The actual resulting contact surface is smooth as shown in Figure 3.



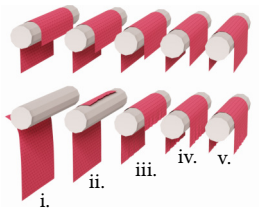
(b) *Reliability.* First three cylinders from the left show cloth simulation from [Li et al. 2018] at different cylinder discretizations. The last cylinder demonstrates our cloth simulation at the coarsest sampling, which shows symmetric sliding without artifacts. This example uses $\mu = 0.12$ and $\Delta t = 0.02$, with the same configuration as in Figure 6a. In this example, the middle two cylinders demonstrate how additional artificial friction can be introduced due to polygon-based contact. Collision detection schemes can also be susceptible to artifacts as seen on the left-most cylinder.



(c) *Resolution independence.* [Li et al. 2018] (top row) and Houdini's cloth simulation (middle row) is compared with our method (bottom row) for different cylinder and cloth resolutions with $\mu = 0.13$. This example shows that our contact model produces similar results regardless of resolution, while a polygon based contact resolution such as Houdini's or [Li et al. 2018] may produce vastly different behaviors.



(d) *Coefficient of friction.* From left to right, the coefficient of friction μ is varied from 0.12 (leftmost) to 0.16 (rightmost) by increments of 0.01. The initial draped cloth is offset by 0.4m from being balanced (i.e. the vertical distance between the two sides is 0.8m). Slipping is consistently decreased as the coefficient of friction is increased from left to right with correct stopping at $\mu > 0.14$ as analytically computed in Appendix B. The material properties were chosen to prevent excessive stretching, which allows the simulation to reproduce the analytic stopping coefficient.



(e) *Sliding consistency.* Here the cloth is draped with an increasing offset from being perfectly balanced. From right to left, the initial offset distance is varied from 0m to 0.64m in increments of 0.16m. This example demonstrates a consistent response to the increase in force difference between the two sides.

Fig. 6. *Cloth on a cylinder.* A patch of cloth draped over a static cylinder 6m in length with radius of 0.5m. For all examples here, we used the cubic kernel to generate the contact field around the cylinder. Here we used $\Delta t = 0.01$s (unless otherwise stated) and all Houdini's cloth simulations are run with 4 collision and simulation substeps. Material parameters between different simulators are matched on a best effort basis.

improve the friction response with higher resolutions it ultimately fails to stick[2].

In Figure 6b we use a similar setup to compare against the non-smooth friction solver from [Li et al. 2018]. This demonstrates that the sliding artifacts are not merely caused by a penalty based friction solve used in Houdini but really from polygon-based contact. Note that under mesh and timestep refinement, our method tends towards the same solution as in [Li et al. 2018], although exact correspondence would require identical material models.

Figure 6c demonstrates how an implicit surface contact model allows us to produce consistent sliding behavior irrespective of resolution. In contrast, polygonal contact models like [Li et al. 2018] or Houdini's can produce vastly different contact behaviors depending on mesh resolution.

We are able to demonstrate precise control over slipping behavior by changing the friction coefficient μ in Figure 6d. Additionally in Figure 6e we show that adjusting the configuration of the cloth will cause it to slip when μ is held constant.

Performance numbers shown in Table 1 indicate that our method is comparable to existing implementations for modest resolutions. In order to produce better scaling with resolution, we recommend using an iterative linear solver.

Ball spin. In Figure 7 a spinning tennis ball with a hollow core (as shown in the inset figure) is dropped onto a slanted ramp to produce a bounce commonly observed in racket sports and golf. We experiment with μ being 0.01, 0.5, and 1.0 to produce a deflected bounce

[2]Sticking is not achieved even after increasing substeps and collision passes to 20, as well as increasing μ to 1.

Example		# Elements	Method	μ	Time	Seconds / Frame
Sticking	i.	2K	Houdini	0.2	1:36	0.19
	ii.				1:26	0.17
	iii.				1:25	0.17
	iv.	2K	Ours	0.2	1:32	0.19
Reliability	i.	2K	[Li et al. 2018]	0.12	3:12	0.38
	ii.				2:38	0.32
	iii.				2:45	0.33
	iv.	2K	Ours	0.12	0:09	0.09
Resolution independence	i.	392	[Li et al. 2018]	0.13	0:35	0.07
	ii.	2K			2:30	0.30
	iii.	7K			7:17	0.87
	iv.	392	Houdini	0.13	1:52	0.22
	v.	2K			2:19	0.28
	vi.	7K			5:54	0.71
	vii.	392	Ours	0.13	0:17	0.03
	viii.	2K			0:38	0.08
	ix.	7K			3:56	0.47
Coefficient of friction	i.	2K	Ours	0.12	0:26	0.05
	ii.			0.13	0:33	0.07
	iii.			0.14	0:35	0.07
	iv.			0.15	1:10	0.14
	v.			0.16	1:17	0.15
Sliding consistency	i.	2K	Ours	0.15	1:13	0.15
	ii.				1:25	0.17
	iii.				1:19	0.16
	iv.				0:54	0.11
	v.				0:51	0.10

Table 1. Timing results (m:ss) over the length of entire corresponding simulations for the *Cloth on a cylinder* example from Figure 6.

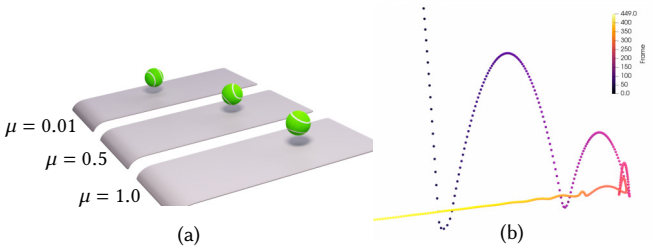
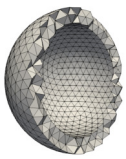


Fig. 7. *Ball spin.* (a) Three identical hollowed balls with different coefficients of friction (as shown) are dropped on a ramp sloping down to the left. (b) The center-of-mass trajectory of the ball with the largest coefficient of friction ($\mu = 1$) is plotted. After the first two bounces up the slope caused by the spin of the ball, friction on the ball causes it to change spin directions temporarily before the ball looses enough energy and starts to roll down the slope.



with varying intensity. The spin is 100 degrees per second in all 3 examples. Our method correctly produces larger bounce deflections for larger coefficients of friction. We are also able to reproduce the subtle behavior of the ball bouncing up the slope (to the right) after it has already started moving down the slope (to the left) due to tangential elasticity forces at the point of contact, which convert the elastic potential into rotational energy. This phenomenon can easily be observed in bouncy rubber balls (SuperBalls).

Fig. 8. *Stick-slip chatter.* A stool slides down a ramp demonstrating stick-slip chatter of the stool legs.

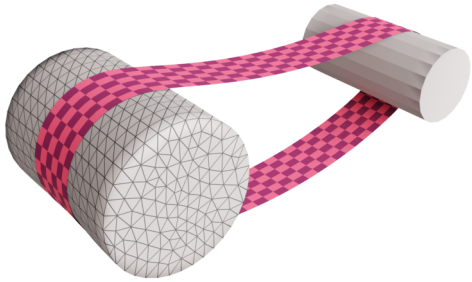
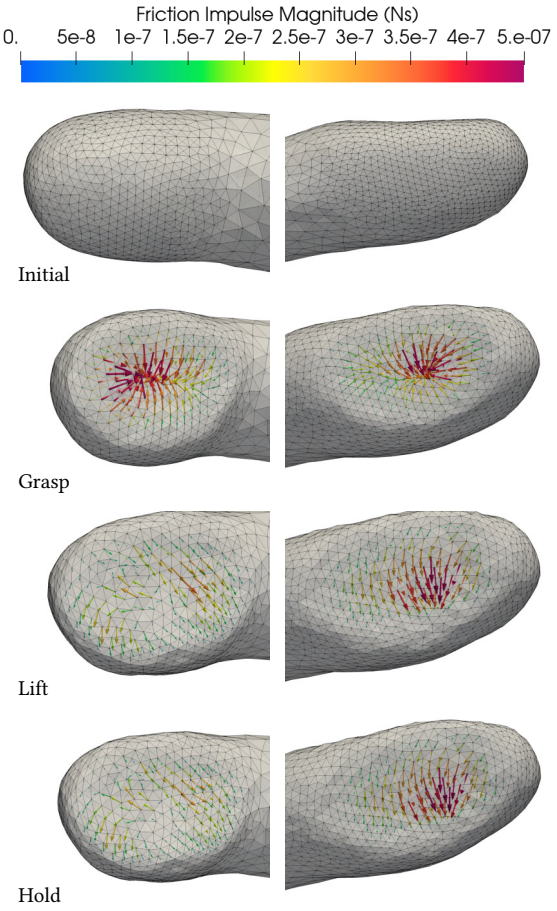


Fig. 9. *Belt drive.* The small driver cylinder is rotated at an accelerated rate, and abruptly stopped to show slipping and sticking for different values of μ.

Stool slide. Our method is able to produce the stick-slip chatter phenomenon on the legs of a stool as it slides down a subtle 10 degree slope as show in Figure 8. As the stool slides down the ramp, each leg of the stool oscillates between sticking and slipping friction modes at the points of contact with the ground. This occurs due to the elasticity of the stool, which periodically shifts the pressure between the front and hind legs of the stool. As with the other ramp slide examples, we use the ramp to produce the implicit field.

Belt drive. As shown in Figure 9, we simulate a belt drive mechanism where a smaller rigid cylinder drives a larger FEM cylinder connected with a belt loop. The friction coefficient between the belt and the small driver cylinder is kept constant at $\mu = 0.5$. Setting the friction coefficient between the large soft cylinder and the belt to $\mu = 0.2$ induces sticking behavior, which successfully drives the soft cylinder around its axis and stops it when the driver abruptly stops. When the coefficient of friction is reduced to $\mu = 0.02$, the soft cylinder is accelerated as before, however it slips through the belt when the driver stops.

Glass pinch. Rigid body simulation is usually insufficient to solve various control problems like picking up a rigid object [Balasubramanian and Santos 2014; Kry and Pai 2006]. We simulate a rigid whiskey glass being pinched and lifted between the index finger and the thumb of an animated soft hand model as shown in Figure 1. Our method produces accurate deformation at the contact patch on the tip of each digit. The shape of the contact patch remains consistent after the grasp as observed on real human digits. We show a detailed friction vector field on the tip of each digit for the grasp, lift and hold stages of the simulation in Figure 10. In the grasp stage, friction is pointing towards the middle of the contact patch to oppose the spreading of flesh around the contact surface as expected. In the lift and hold stages, an interesting pattern emerges on the index finger pad caused by a subtle tilt of the glass. Since the contact pivots are



(a) Index finger. (b) Thumb.

Fig. 10. The friction impulse (in N·s) vector field on the index finger and the thumb from the glass pinch simulation in Figure 1. Four stages of the simulation are captured for each digit: initial, grasp, lift and hold. The initial configuration is the frame captured immediately before any contact has occured. In the grasp stage, the finger and thumb first come into contact with the glass. In the lift stage the hand is moving up. Finally in the hold stage the hand is stationary.

not perfectly aligned between the index finger and the thumb, the glass tilts slightly towards the index finger and into the hand. The tilt towards the index finger transfers the load from the tangential component to the normal on the finger pad, which is why friction forces are larger on the thumb. Finally, the tilt towards the hand applies a torque onto the index finger pad causing the spiral pattern of the friction vector field. In this example, the hand mesh generates the implicit contact field, and the glass collides at vertices, although the opposite configuration would work as well.

Ball on a trampoline. To generate a more complex interaction between soft objects, we drop a hollowed ball onto a trampoline which is then contracted and stretched as shown in Figure 11. This example further demonstrates how a stretched thin sheet can reliably interact with another soft object represented by an implicit surface. The animation clearly demonstrates the tangential two way coupling between the ball and the trampoline as the ball rolls through the well of the trampoline and finally rests near the center as expected.

Armadillo tank top fitting. We fit a sleeveless shirt on an animated armadillo model to demonstrate an application of our method for simulated soft tissue cloth fitting. This application necessitates an accurate friction model to achieve a realistic fit.

We compare a frictionless fit against one with $\mu = 0.2$ against a *rigid* armadillo in Figure 12a. Here we can see that friction is required to keep the straps from sliding off the armadillo's shoulders.

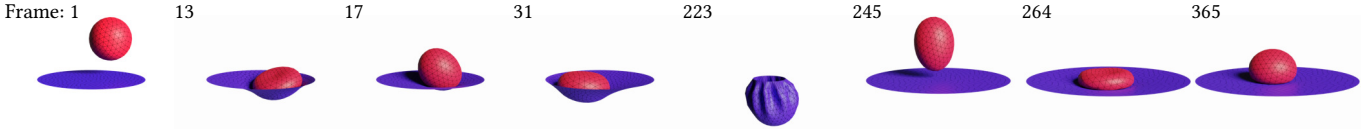
To gain more realism we simulate the *soft* tissue of the armadillo, again with and without friction in Figure 12b. In the soft fit, the animation reveals additional effects of frictional contact, namely the transfer of dynamic motion between the body and cloth at the point of contact. This scenario demonstrates how our method can be used for predicting the utility of active wear that controls soft tissue displacement during dynamic movement.

7 LIMITATIONS

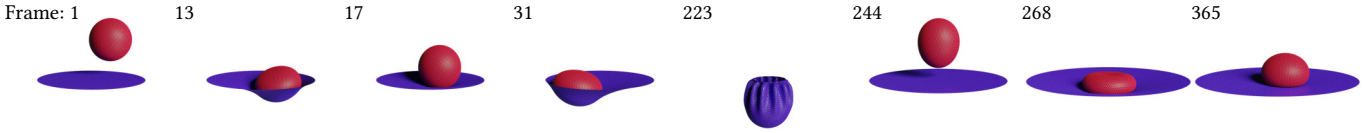
Self-contact. A limitation of our method is self-contact, which is excluded from our formulation. Self-contacts impose an additional difficulty of pruning neighborhoods $\mathcal{N}_i$ for each contact i since a mesh vertex should not collide against the surface generated by its neighbors. However this restriction can be more complicated in creases where colliding surfaces may be in the same neighborhood. We leave this important extension as future work.

Performance and scalability. Choosing an off-the-shelf non-linear solver like Ipopt allowed us to side-step the arduous and error-prone process of implementing a robust Newton-Raphson solver capable of enforcing linear inequality constraints. However, the use of a direct linear solver has some impact on the scalability of our method to larger meshes. In addition, our method relies on the explicit construction of the Delassus operator, which imposes scalability limitations in the total number of contacts, however, the Delassus operator has been used in contexts with a much larger contact count, such as hair [Daviet et al. 2011]. We aim to investigate other performance bottlenecks in the future.

Sampling. In contrast to polygon based methods for resolving contacts, our method may require a finer sampling of the surface to produce a sufficiently accurate approximation (e.g. in the *Belt drive* example, the sharp corners of the cylinder raise the Hausdorff error up to 1.4%). Good quality finite element meshes tend to be locally uniform on the surface, which is why we chose the finite element mesh to produce the implicit field directly. However, when used on a static collision mesh (like with the *cloth on cylinder* example), our method can impose additional sampling steps. Thus we recommend that MLS surfaces are used on deformable organic objects, which tend to have locally uniform tesselations and few naturally sharp corners or thin features. For this reason our method doesn't directly deal with sharp corners, but it can do so if extended with additional remeshing and optimization techniques [Li et al. 2018; Macklin et al. 2020].



(a) Low resolution ball on a trampoline.



(b) High resolution ball on a trampoline.

Fig. 11. *Ball on a trampoline.* A hollowed ball is released onto a stretched trampoline in frames 1 to 150. The trampoline is then contracted and stretched even more to launch the ball into the air in the remaining frames. This example demonstrates reliable frictional coupling between the trampoline closely conforming to the ball. In this example we used $\mu = 0.2$, and the cubic kernel for generating a smooth implicit contact surface.

Example	# Elements	# Vertices	Δt (s)	Frames	h_{err}	μ	n	$k_{\text{tmax}}/k_{\text{tavg}}$	$i_{\text{kmax}}/i_{\text{kavg}}$	Time	$\frac{\text{Seconds}}{\text{Frame}}$
Ball spin	9K	2K	0.005	1500	0.03%	0.01	867	3/0.41	12/6.6	0:07:32	0.30
						0.5	867	3/0.74	12/6.7	0:08:55	0.36
						1.0	860	3/0.76	13/6.8	0:08:52	0.35
Stick-slip chatter	11K	3K	$0.000\bar{3}$	18000	<0.01%	0.2	479	3/1.94	17/14.3	2:40:58	0.54
Belt drive	8K	2K	0.005	2500	1.4%	0.02	610	6/4.1	53/13.3	0:24:03	0.58
						0.2	620	6/4.0	43/13.5	0:21:23	0.51
Glass pinch	45K	9K	0.0001	10000	0.2%	0.15	989	3/1.0	38/15.0	8:31:57	3.07
Ball on a trampoline	8K	2K	0.01	1500	0.2%	0.2	548	4/2.1	48/13.6	0:10:48	0.43
	58K	15K	0.005	3000	0.02%	0.2	2963	8/2.4	182/27.5	3:56:30	4.73
Armadillo rigid fit	5K	3K	0.004	1000	0.2%	0	2542	3/1.5	14/12.1	0:08:09	0.49
						0.8	2559			0:12:26	0.75
Armadillo soft fit	88K	22K	0.001	4000		0	2608			5:57:17	5.36
						0.8	2609	5/1.6	53/17.2	8:05:47	7.29

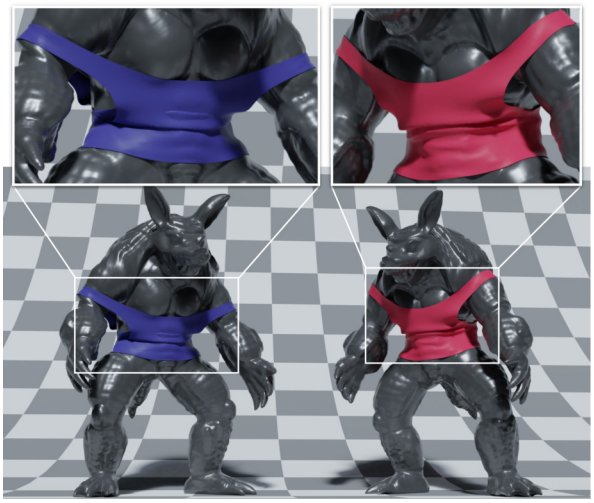
Table 2. Timing results (h:mm:ss) over the length of entire corresponding simulations. The *# Elements* column counts over triangles and tetrahedra with at least one free vertex. n is the maximum number of contact constraints. k_{tmax} and k_{tavg} are the maximum and average numbers of friction steps resp. taken per timestep respectively where the friciton solve converged. i_{kmax} and i_{kavg} are the maximum and average numbers of Ipopt iterations resp. taken per friction projection step (28). h_{err} is the initial Hausdorff distance of the MLS surface to the surface of the tetrahedral mesh divided by the maximum dimension of the mesh's bounding box.

Friction Convergence. In all our examples, the friction solve converged to within $\epsilon = 10^{-3}$, however a few instances (less than 30 frames in total from all examples) it failed to converge below $\epsilon = 10^{-4}$. We believe that convergence can be further improved by exploring other friction solvers such as the Fischer-Burmeister formulation [Daviet et al. 2011; Kaufman et al. 2014; Macklin et al. 2019]. However, non-convergence did not seem to have any noticeable effect on contact fidelity in our examples.

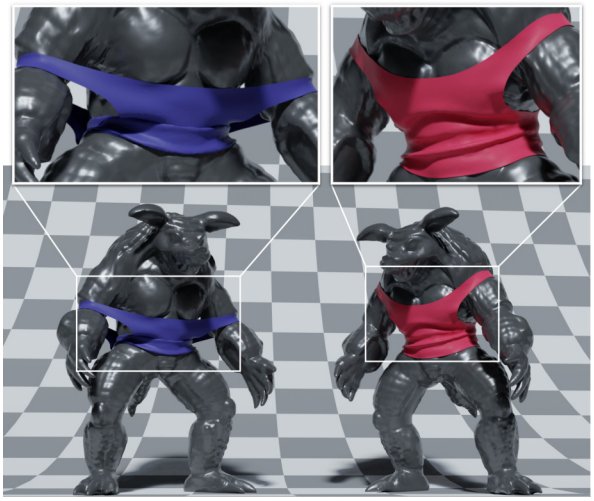
8 DISCUSSION AND CONCLUSIONS

Friction Forwarding. The choice of time splitting in our approach is critical to producing robust frictional responses. Figure 13 shows the effect of forwarding the friction impulse to the next time step. This technique allows friction to affect the elasticity solve, and as a result produces a more accurate prediction for how friction propagates through the rest of the elastic body. Figure 14 shows the artifacts produced when friction forwarding is disabled in the *glass pinch* example, by removing $\boldsymbol{r}^{\text{t}}$ from (20) and (22).

Flexibility and Extensibility. Because our method does not rely on polygonal collision detection schemes, it can be used on a variety of surface representations. The only required data is points with associated normals. This means our frictional contact model can be used with meshless methods and with raw point clouds generated by 3D scanners. Furthermore, our method is agnostic to the choice of linear solvers and non-linear constrained optimization algorithms, which simplifies future improvements and use of third-party libraries with better performance characteristics.



(a) Rigid fit.



(b) Soft fit.

Fig. 12. *Armadillo tank top fit.* A tank top is fit onto an animated Armadillo model. (a) shows the garment on a rigid armadillo without friction (left) and with $\mu = 0.8$ (right). A more realistic scenario (b) demonstrates the garment being fit on a soft armadillo model without friction (left) and with $\mu = 0.8$ (right).

Conclusions. We have introduced a novel approach for solving frictional contact problems with smooth elastic solids. Our solution is presented in a variational form, which allows for easy modifications with additional constraints. This formulation maintains loose coupling between the frictional contact solve and elasticity via *friction forwarding*, which permits different methods to be used in each step. We introduced a novel approach for contact resolution using smooth local implicit surfaces, which avoids additional collision detection schemes. Finally, our implicit surface formulation permits tangential force feedback using a parallel transport approximation.

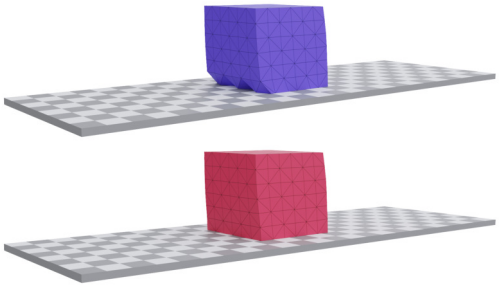


Fig. 13. An FEM block of tetrahedra is dragged from right to left from the top vertices. The block slides along a horizontal plane while slightly compressed. The top frame shows the simulation without friction forwarding (i.e. $\boldsymbol{r}^{\mathrm{t}} = \boldsymbol{0}$ in (20) and (22)). The bottom frame shows the same simulation with friction forwarding. The vertices near the front of the cube are better aligned because friction is forwarded to the elasticity solve (20). As a result, friction is felt by vertices not directly in contact with the surface within a single time step.

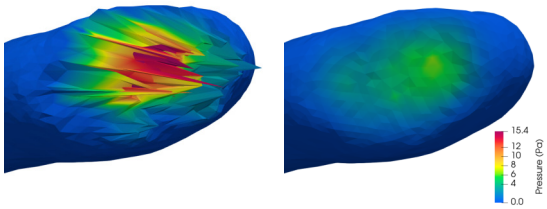


Fig. 14. A closeup of the thumb shortly after the grasp phase of the glass pinch example from Figure 1. The left image shows the thumb surface *without* friction forwarding , while the right shows the same simulation *with* friction forwarding. Note that in both cases friction was solved to convergence.

Our method is validated visually and by comparison with a solution used widely in the visual effects industry.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, Vital Mechanics Research, and the Canada Research Chairs Program.

REFERENCES

Vincent Acary and Bernard Brogliato. 2008. *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics.* Springer Science & Business Media.

James Ahrens, Berk Geveci, and Charles Law. 2005. Paraview: An end-user tool for large data visualization. *The visualization handbook* 717, 8 (2005).

Pierre P. 'Alart and Alain R. Curnier. 1991. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Computer Methods in Applied Mechanics and Engineering* 92, 3 (1991), 353–375. https://doi.org/10.1016/0045-7825(91)90022-X

Stefan Altmayer. 2020. rstar. https://github.com/Stoeoef/rstar

Ravi Balasubramanian and Veronica J Santos. 2014. *The human hand as an inspiration for robot hand development.* Vol. 95. Springer.

David Baraff. 1990. Curved surfaces and coherence for non-penetrating rigid body simulation. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques (SIGGRAPH '90).* Association for Computing Machinery, Dallas, TX, USA, 19–28. https://doi.org/10.1145/97879.97881

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

Markus Becker, Markus Ihmsen, and Matthias Teschner. 2009. Corotated SPH for Deformable Solids.. In *NPH*. Citeseer, 27–34.

Florence Bertails-Descoubes, Florent Cadoux, Gilles Daviet, and Vincent Acary. 2011. A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Transactions on Graphics (TOG)* 30, 1 (2011), 6.

Marie-Paule Cani. 1993. An implicit formulation for precise contact modeling between flexible solids. In *20th annual conference on Computer graphics and interactive techniques (SIGGRAPH'93))*. ACM SIGGRAPH, 313–320.

Blender Online Community. 2019. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 139.

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. *Mathematical and Computer Modelling* 28, 4 (Aug. 1998), 225–245. https://doi.org/10.1016/S0895-7177(98)00119-8

Mathieu Desbrun and Marie-Paule Cani. 1994. Highly deformable material for animation and collision processing. In *5th Eurographics Workshop on Computer Animation and Simulation (EGCAS'94)*. 89–102.

Elliot English and Robert Bridson. 2008. Animating developable surfaces using nonconforming elements. *ACM Trans. Graph.* 27 (08 2008). https://doi.org/10.1145/1360612.1360665

Michael Erdmann. 1994. On a Representation of Friction in Configuration Space. *The International Journal of Robotics Research* 13, 3 (1994), 240–271. https://doi.org/10.1177/027836499401300306 arXiv:https://doi.org/10.1177/027836499401300306

Kenny Erleben. 2017. Rigid Body Contact Problems Using Proximal Operators. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. ACM, New York, NY, USA, Article 13, 12 pages. https://doi.org/10.1145/3099564.3099575

Gaetano Fichera. 1964. *Problemi elastostatici con vincoli unilaterali: il problema di Signorini con ambigue condizioni al contorno*. Accademia nazionale dei Lincei.

Marie-Paule Gascuel. 1993. An implicit formulation for precise contact modeling between flexible solids. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 313–320.

Qi Guo, Xuchen Han, Chuyuan Fu, Theodore Gast, Rasmus Tamstorf, and Joseph Teran. 2018. A Material Point Method for Thin Shells with Frictional Contact. *ACM Trans. Graph.* 37, 4, Article 147 (July 2018), 15 pages. https://doi.org/10.1145/3197517.3201346

Xuchen Han, Theodore F. Gast, Qi Guo, Stephanie Wang, Chenfanfu Jiang, and Joseph Teran. 2019. A Hybrid Material Point Method for Frictional Contact with Diverse Materials. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 17 (July 2019), 24 pages. https://doi.org/10.1145/3340258

Simone E. Hieber and Petros Koumoutsakos. 2005. A Lagrangian particle level set method. *J. Comput. Phys.* 210, 1 (2005), 342–367. https://doi.org/10.1016/j.jcp.2005.04.013

Danny M. Kaufman, Timothy Edmunds, and Dinesh K Pai. 2005. Fast frictional dynamics for rigid bodies. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 946–956.

Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Transactions on Graphics (SIGGRAPH Asia 2008)* 27, 5 (2008), 164:1–164:11.

Danny M. Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. 2014. Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 123.

Peter Kaufmann, Sebastian Martin, Mario Botsch, and Markus Gross. 2008. Flexible Simulation of Deformable Models Using Discontinuous Galerkin FEM. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland) *(SCA '08)*. Eurographics Association, Goslar, DEU, 105–115.

Noboru Kikuchi and John Tinsley Oden. 1988. *Contact problems in elasticity: a study of variational inequalities and finite element methods*. Vol. 8. siam.

Paul G. Kry and Dinesh K. Pai. 2003. Continuous contact simulation for smooth surfaces. *ACM Transactions on Graphics* 22, 1 (Jan. 2003), 106–129. https://doi.org/10.1145/588272.588280

Paul G. Kry and Dinesh K. Pai. 2006. Interaction capture and synthesis. *ACM Transactions on Graphics* 25, 3 (July 2006), 872–880. https://doi.org/10.1145/1141911.1141969

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 Transactions on Graphics (TOG)* 37, 4 (2018), 52.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Transactions on Graphics* 39, 4 (2020).

Y. F. Liu, J. Li, Z. M. Zhang, X. H. Hu, and W. J. Zhang. 2015. Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system. *Mechanical Sciences* 6, 1 (2015), 15–28. https://doi.org/10.5194/ms-6-15-2015

Per Lötstedt. 1982. Time-dependent contact problems in rigid body mechanics. In *Nondifferential and Variational Techniques in Optimization*. Springer, 103–110.

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Zach Corse. 2020. Local Optimization for Robust Signed Distance Field Collision. *Proc. ACM Comput. Graph. Interact. Tech.* 3, 1, Article 8 (April 2020), 17 pages. https://doi.org/10.1145/3384538

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-smooth Newton Methods for Deformable Multi-body Dynamics. *ACM Transactions on Graphics* 38, 5 (Oct 2019), 1–20. https://doi.org/10.1145/3338695

Nicholas D. Matsakis and Felix S. Klock, II. 2014. The Rust Language. *Ada Lett.* 34, 3 (Oct. 2014), 103–104. https://doi.org/10.1145/2692956.2663188

Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient elasticity for character skinning with contact and collisions. In *ACM SIGGRAPH 2011 papers (SIGGRAPH '11)*. Association for Computing Machinery, Vancouver, British Columbia, Canada, 1–12. https://doi.org/10.1145/1964921.1964932

Brian Mirtich and John Canny. 1994. *Impulse-based dynamic simulation*. Computer Science Division (EECS), University of California California.

Matthew Moore and Jane Wilhelms. 1988. Collision detection and response for computer animation. In *ACM Siggraph Computer Graphics*, Vol. 22. ACM, 289–298.

Jean Jacques Moreau. 1973. *On Unilateral Constraints, Friction and Plasticity*. Springer Berlin Heidelberg, Berlin, Heidelberg, 171–322. https://doi.org/10.1007/978-3-642-10960-7_7

Bryan S. Morse, Terry S. Yoo, Penny Rheingans, David T. Chen, and K. R. Subramanian. 2005. Interpolating Implicit Surfaces from Scattered Surface Data Using Compactly Supported Radial Basis Functions. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) *(SIGGRAPH '05)*. Association for Computing Machinery, New York, NY, USA, 78–es. https://doi.org/10.1145/1198555.1198645

Thomas Most and Christian Bucher. 2005. A Moving Least Squares weighting function for the Element-free Galerkin Method which almost fulfills essential boundary conditions. *Structural Engineering and Mechanics* 21 (10 2005). https://doi.org/10.12989/sem.2005.21.3.315

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.

Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. 2003. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *2003 Shape Modeling International.* IEEE, 153–161.

Stanley Osher, Ronald Fedkiw, and K Piechor. 2004. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.* 57, 3 (2004), B15–B15.

Miguel A. Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus 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 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01396.x

Paul Painlevé. 1895. Sur les loi du frottement de glissement. In *Comptes Rendu des Séances de l'Academie des Sciences*. Paris, Article 121, 112–115 pages.

Kenneth Salisbury and Christopher Tarr. 1997. Haptic rendering of surfaces defined by implicit functions. In *ASME Dynamic Systems and Control Division*, Vol. 61. 61–67.

Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. 2004. Interpolating and Approximating Implicit Surfaces from Polygon Soup. In *ACM SIGGRAPH 2004 Papers* (Los Angeles, California) *(SIGGRAPH '04)*. ACM, New York, NY, USA, 896–904. https://doi.org/10.1145/1186562.1015816

SideFX. 2018. Houdini.

Eftychios Sifakis and Jernej Barbic. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California) *(SIGGRAPH '12)*. ACM, New York, NY, USA, Article 20, 50 pages. https://doi.org/10.1145/2343483.2343501

Antonio Signorini. 1933. Sopra alcune questioni di elastostatica. *Atti della Societa Italiana per il Progresso delle Scienze* 21, II (1933), 143–148.

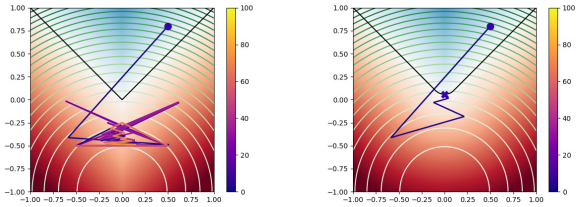
Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (March 2018), 15 pages. https://doi.org/10.1145/3180491

Jonas Spillmann and Matthias Teschner. 2008. An adaptive contact model for the robust simulation of knots. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 497–506.

Greg Turk and James F. O'Brien. 2002. Modelling with Implicit Surfaces That Interpolate. *ACM Trans. Graph.* 21, 4 (Oct. 2002), 855––873. https://doi.org/10.1145/571647.571650

Rodolphe Vaillant, Loïc Barthe, Gaël Guennebaud, Marie-Paule Cani, Damien Rohmer, Brian Wyvill, Olivier Gourmel, and Mathias Paulin. 2013. Implicit Skinning: Real-Time Skin Deformation with Contact Modeling. *ACM Trans. Graph.* 32, 4, Article 125 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461960

Rodolphe Vaillant, Gäel Guennebaud, Loïc Barthe, Brian Wyvill, and Marie-Paule Cani. 2014. Robust Iso-surface Tracking for Interactive Character Skinning. *ACM Trans. Graph.* 33, 6, Article 189 (Nov. 2014), 11 pages. https://doi.org/10.1145/2661229.



(a) *Non-smooth contact.* (b) *Smoothed contact.*

Fig. 15. This illustration depicts the initial configuration of simplification for a common contact scenario. A single vertex marked by the blue dot is placed inside a quadratic potential (with a minimum is at (0,-1)), and subjected to lay in the blue area bordered by the contact surface in black. In (a), the solution (final vertex position) to the elasticity problem subject to contact constraints lies at a non-differentiable point on the implicit contact surface, which causes typical gradient-based optimizers to diverge. In (b), the contact surface is smoothed producing a differentiable constraint function which allows gradient-based iterations to converge at the minimum located at the bottom of the smoothed cusp. The color bar on the side of each plot indicates the iteration number at each point on the path taken by the optimizer.

2661264

Mickeal Verschoor and Andrei C. Jalba. 2019. Efficient and Accurate Collision Response for Elastically Deformable Models. *ACM Trans. Graph.* 38, 2, Article 17 (March 2019), 20 pages. https://doi.org/10.1145/3209887

Andreas Wächter and Lorenz T. Biegler. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* 106, 1 (01 Mar 2006), 25–57. https://doi.org/10.1007/s10107-004-0559-y

A. C. Öztireli, G. Guennebaud, and M. Gross. 2009. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. *Computer Graphics Forum* 28, 2 (2009), 493–501. https://doi.org/10.1111/j.1467-8659.2009.01388.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01388.x

A SMOOTH VERSUS NON-SMOOTH IMPLICIT CONTACT

In this section we motivate the use of a custom smooth implicit surface over the standard signed distance field for use in constrained optimization for contact resolution as presented in Section 4.

Consider a crude simplification of the contact problem in the context of polygonal contact, where a single vertex on a larger mesh at position $\boldsymbol{x}_0$ is deformed to lay in a crevice between two polygons. Suppose that we use the signed distance field (SDF) of the polygonal surface as the constraint in our elasticity minimization problem for contact resolution. In Figure 15a we illustrate a scenario where a non-smooth implicit surface causes gradient-based optimizers to diverge. In Figure 15b, we illustrate the same simplified example, but with a smooth surface approximation at the cusp. Here, the derivative is well-defined at the minimum, which allows gradient-based methods to converge. Recent work has improved signed distance contact handling by smoothing the SDF specifically for edge-edge contact pairs [Li et al. 2020]. In our work, we smooth the entire surface using MLS, rather than smoothing polygon cusps directly, which maintains all of the artificial polygonal facets. These facets may be undesirable when modelling organic objects. Note that this issue affects any line-search method, which uses the constraint Jacobian for computing the descent direction.

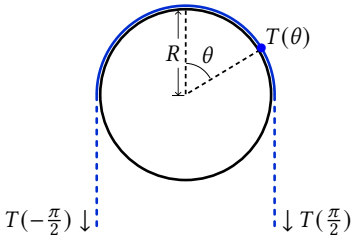


Fig. 16. *Cloth on cylinder.* This diagram shows a cross-section of our cloth on cylinder configuration for the capstan equation derived in Appendix B. The cloth (in blue) rests on the cylinder (in black), which has radius R. $T(\theta)$ is the tension on the cloth at each point on the top semi-circle.

B CLOTH ON CYLINDER STOPPING FRICTION

To compute the correct stopping coefficient for the cloth on cylinder example, we will start with a derivation for the capstan equation:

$$\frac{dT}{d\theta} = \mu T$$

where T is the tension on the cloth along each point of contact with the cylinder and θ is the angle of the contact point from the vertical as shown in Figure 16. When the tension caused by the weight of the cloth at every contact point is included, we get

$$\frac{dT}{d\theta} = \mu T + \mu Rb\cos(\theta) - Rb\sin(\theta) \tag{29}$$

where the additional two terms describe the contribution of the normal and tangential components of the gravitational pull at every contact point. Here R is the radius of the cylinder, and $b = \rho g w$ where ρ is the cloth mass density, g is gravitational acceleration and w is the width of the cloth (size along the length of the cylinder). The general solution to (29) is

$$T(\theta) = Ce^{\theta\mu} + \frac{-Rb\mu^2\cos(\theta) + 2Rb\mu\sin(\theta) + Rb\cos(\theta)}{\mu^2+1}$$

where C is the constant of integration. Since we know the weight of the cloth at each end, $T(-\frac{\pi}{2})$ and $T(\frac{\pi}{2})$, we can determine C to be

$$C = -\frac{b(\mu R - \ell)}{2\cosh(\frac{\pi\mu}{2})}$$

where ℓ is the total length of the cloth. Now by equating the weight at with $T(-\frac{\pi}{2})$, we will get

$$\ell_{-\pi/2} = -\frac{(\mu R - \ell)e^{-\frac{\pi\mu}{2}}}{2\cosh(\frac{\pi\mu}{2})} - \frac{2R\mu}{\mu^2+1} \tag{30}$$

We can now plug in all known lengths and solve for μ. For the example in Figure 6d and assuming that $\ell_{-\pi/2}$ represents the shorter end, we have

$$\ell_{-\pi/2} = 0.8, \quad \ell = 4, \quad R = \frac{1.6}{\pi} \tag{31}$$

where all values are given in meters. Finally, the root of (30) given (31) is

$$\mu \approx 0.14015161654962588\ldots$$