Constrained dynamics with frictional contact on smooth surfaces

by

Egor Larionov

B. Math., The University of Waterloo, 2013

M. Math., The University of Waterloo, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Computer Science)

The University of British Columbia

(Vancouver)

December 2022

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Constrained dynamics with frictional contact on smooth surfaces

submitted by **Egor Larionov** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Computer Science**.

Examining Committee:

Dinesh K. Pai, Professor, Computer Science, UBC Vancouver
Supervisor

Uri M. Ascher, Professor Emeritus, Computer Science, UBC Vancouver
Supervisory Committee Member

Michael Friedlander, Professor, Computer Science, UBC Vancouver
Supervisory Committee Member

Chen Greif, Professor, Computer Science, UBC Vancouver
University Examiner

Eldad Haber, Professor, Earth, Ocean and Atmospheric Sciences, UBC Vancouver
University Examiner

Abstract

Friction and contact pose a great challenge to efficient and accurate simulation of deformable objects for computer graphics and engineering applications. In contrast to many engineering applications, simulation software for graphics often permits larger approximation errors in favour of better predictability, controllability and efficiency. This dissertation explores modern methods for frictional contact resolution in computer graphics. In particular, the focus is on offline simulation of smooth elastic objects subject to contact with other elastic solids and cloth. We explore traditional non-smooth friction formulations as well as smoothed frictional contact, which lends itself well to differentiable simulation and analysis. We then explore a particular application of differentiable simulation to motivate the direction of research.

In graphics, even smooth objects are typically approximated using piecewise linear polyhedra, which exhibit sliding artifacts that can be interpreted as artificial friction making simulations less predictable. We develop a technique for improving fidelity of sliding contact between smooth objects.

Frictional contacts are traditionally resolved using non-smooth models, which are complex to analyse and difficult to compute to a desirable error estimate. We propose a unified description of the equations of motion subject to frictional contacts using a smooth model that converges to an accurate friction response. We further analyse the implications of this formulation and compare our results to state-of-the-art methods.

The smooth model uniquely resolves frictional contacts, while also being fully differentiable. This allows inverse problems using our formulation to be solved by gradient-based methods. We begin our exploration of differ-

entiable simulation applications with a parameter estimation task. Elastic parameters are estimated for a three distinct cloth materials using a novel capture, registration and estimation pipeline. Static equilibrium cloth configurations are efficiently estimated using a popular compliant constraint dynamics. In this work we address a common issue of bifurcation in cloth, which causes final configuration mismatches during estimation. Finally, we postulate an extension to compliant constraint dynamics using our friction model, to show how our previous work can be used in parameter estimation tasks involving contact and friction.

Lay Summary

The simulation of frictional contact between deformable elastic objects is a challenging problem in computer graphics. This dissertation explores modern methods for simulating soft objects subject to friction and contact forces. In particular, we focus on high fidelity simulation results used in visual effects, animation and fabrication applications. In contrast to many engineering applications, graphics permits larger approximation errors in favour of improved visual fidelity. In practice, this means that methods applied in graphics can be vastly different from methods in other domains. The specific methods explored enable further development of applications capable of automatically reproducing real-life behaviour in simulation. We present a pipeline for reproducing elastic behaviour of cloth to demonstrate one such application.

Preface

This dissertation describes results in one published research article in Chapter 3, which was presented at SIGGRAPH in 2021:

- 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. [78]

and two articles in the process of resubmission in Chapters 4 and 5 respectively:

- Egor Larionov, Andreas Longva, Uri M. Ascher, Jan Bender and Dinesh K. Pai. Fully implicit frictional dynamics with soft constraint. [80]

- Egor Larionov, Marie-Lena Eckert, Katja Wolff, Tuur Stuyck. Estimating cloth elasticity parameters using position-based simulation of compliant constrained dynamics. [79]

In all three projects the present author was the primary researcher and was the sole implementer of all algorithms presented as contributions in this manuscript. All co-authors provided advice and guidance in each of the three projects.

The first two projects were completed under the supervision of Prof. Pai and result in a novel end-to-end simulation software package written in Rust and published to GitHub under the Mozilla Public License v. 2.0. Ye Fan provided high-level help and advice during the first project. The second project was completed in collaboration with Prof. Uri M. Ascher as well as

Ph.D. candidate Andreas Longva and Prof. Jan Bender from the Computer Animation group at RWTH Aachen University.

The last of the three projects was implemented as part of the proprietary simulation system used by Meta Reality Labs. Dr. Stuyck, Dr. Eckert and Dr. Wolff provided guidance and support for this project during the author's internship with Meta Reality Labs in 2021.

Table of Contents

List of Tables

List of Figures

List of Acronyms

BDF backward differentiation formula.

BE backward Euler.

BiCGSTAB biconjugate gradient stabilized.

CCD continuous collision detection.

CGS centimetre-gram-second.

FEM finite element method.

FFT finite Fourier transform.

IMEX implicit-explicit.

IPC incremental potential contact.

KKT Karush-Kuhn-Tucker.

MDP maximum dissipation principle.

MLS moving least squares.

MPM material point method.

NR-ICP non-rigid iterative closest point.

NURBS non-uniform rational B-spline.

ODE ordinary differential equation.

PBD position based dynamics.

SDF signed distance field.

SDIRK singly diagonally implicit Runge-Kutta.

SFC Signorini-Fichera condition.

SPD symmetric positive definite.

SPH smoothed particle hydrodynamics.

StVK St. Venant-Kirchhoff.

TR trapezoid rule.

XPBD extended position based dynamics.

List of Symbols

δ contact distance tolerance measured in distance units.

ϵ friction curve smoothing tolerance measured in velocity units.

$\mathbf{J}$ residual Jacobian with respect to vertex velocities, $\partial \mathbf{r}/\partial \mathbf{v}$, where $\mathbf{r}$ is the residual or momentum balance function.

$\mathbf{J}_c$ contact Jacobian with respect to vertex positions, $\partial \mathbf{x}/\partial \mathbf{q}$ where $\mathbf{x}$ is the stacked vector of contact positions.

κ contact penalty stiffness for soft contact constraints.

κ_v isothermal compression coefficient.

$\mathbf{q}$ stacked vector of vertex positions.

$\mathbf{v}$ stacked vector of vertex velocities.

Acknowledgements

First and foremost I want to thank my academic advisor Prof. Dinesh Pai for his continued support and guidance. I would also like to thank Prof. Uri Ascher for the enlightening discussions, detailed notes on numerical analysis and for inspiring me to continue my research efforts. I am thankful to Prof. Michael Friedlander, for his teachings on convex optimization, invaluable feedback on my research and encouragement throughout my degree.

I am grateful to my intern manager, Tuur Stuyck, for his relentless support and encouragement during my internship at Meta Reality Labs. I want to thank Katja Wolff and Marie-Lena Eckert for their helpful feedback and discussions at Meta. The project on cloth parameter estimation would not be possible without Arkell Rasiah and Michelle Hill who set up the capture system and produced quality 3D reconstructions of real cloth swatches during our remote collaboration.

I am thankful to Andreas Longva for the deep and insightful discussions and collaboration as well as Prof. Jan Bender for the prompt and detailed feedback.

I would like to thank the faculty and staff at The University of British Columbia for providing the resources and support necessary to complete this research. I am especially grateful to the Department of Computer Science and the Natural Sciences and Engineering Research Council of Canada for providing funding for my research.

I want to thank my former colleagues Ye and Darcy as well as fellow lab mates Heon, Pearson, Edwin, Jan and many others for the helpful discussions, debates and collaborations.

Finally, I would like to thank my friends and family, especially my fiancée Dian, for their patience, love and support.

to my fiancée

Chapter 1

Introduction

Frictional contact is a prevalent phenomenon in the real world. It is responsible for a vast variety of behaviours exhibited by interacting objects. Frictional contact enables us to hold objects in our hands, to move while walking, to wear clothes made from threads held together by friction, and put topspin on a ball in racket sports. This makes friction and contact a particularly important research topic in computer graphics. However, simulating frictional contact remains challenging despite more than 30 years of research in computer graphics, and more than 300 years of research in physics and engineering.

Virtual representation of real world objects is fundamental in computer graphics. It involves many areas of research including visual rendering, animation and simulation. Rendering is responsible for how virtual things *look*. This is dictated by the interaction between light, materials and ultimately our eyes. Animation deals with how objects *move*. This often involves various tools and automations that allow one to control complex objects such as a human body in 3D. Finally, simulation addresses how objects *behave* and *interact* with one another in the virtual world. Graphics research has made great strides in advancing rendering and animation technology, making virtual worlds appear, at times, indistinguishable from reality. However, what often betrays virtual reality most, is the behaviour of objects we are most familiar with, and their mutual interaction. For instance, consider a

simple scenario depicted in Figure 1.1, where a hand grasps a whiskey glass between the thumb and finger. Visually, we expect to see the finger patch deform along the surface of the glass as it is lifted. When violated, any one of those phenomena will break the illusion of reality we seek. Although the motion in the scenario in question can be handcrafted with animation tools, it is impractical for high fidelity visuals. Physically-based simulation allows one to mimic reality automatically, which enables highly complex visuals.

This dissertation deals with physically-based simulation of soft objects and their interaction with one another. Specifically, we focus on simulating dynamics of hyperelastic objects subject to frictional contacts. Objects of interest are first discretized in space using the finite element method (FEM) and then in time using an appropriate time integration scheme. For cloth simulation, we also present a method for automatically determining simulation parameters from real-world captures, which further improves the effectiveness of simulation to represent real clothing.

Figure 1.1: A rigid whiskey glass is pinched between the index finger and thumb of an animated hand model and lifted. The first frame shows the internal geometry of the distal phalanges whose vertices drive the fingertips 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.

1.1 Overview

Frictional behaviour in simulation can be modelled using traditional non-smooth models like Coulomb friction, or smoothed models where some imperceptible slip is tolerated. Non-smooth models aim to find accurate sticking dynamics with no slip, although they require complex non-smooth methods with weak convergence guarantees or time-splitting techniques with inflexible discretization requirements. In contrast, smooth models allow some slip error, which can be limited with an explicit velocity tolerance. The two main benefits of smooth models are that they can be solved using methods with strong convergence guarantees and they allow differentiability of the entire simulation system. Differentiability makes simulators attractive tools for solving inverse problems where derivative-based methods can be used.

In Chapter 3 we address a common issue in contact handling of smooth objects discretized using piecewise linear surfaces. In effect, under different levels of discretization, piecewise linear contact surfaces exhibit distinct and unpredictable frictional sliding behaviour. We address this by resolving contacts against a smooth implicit surface. We show that this method enables consistent sliding behaviour under mesh refinement, since surfaces remain smooth under any resolution. A popular non-smooth friction method is then extended and applied to the new contact model. Friction behaviour is then verified using analytic results.

Chapter 4 digs deeper into accurate friction modelling. Here we address a known drawback of smooth friction models – accuracy. By expressing elastodynamics and frictional contact as a single system of equations, we allow friction to be integrated implicitly along with associated tangential transforms, which enable accurate friction responses on curved surfaces. This also allows us to analyse the friction problem in the context of elastodynamics more effectively. With this model we unlock the ability to easily apply higher-order time-integration schemes, where previous methods fail to do so.

In Chapter 5, we explore an application of differentiable simulation. In particular, we build a cloth parameter estimation pipeline for capturing real

world cloth deformation, registering simulation meshes to acquired captures and estimating the required material properties for a simulator to reproduce the desired result. This system forms a tool for automatically determining simulator parameters without the need for arduous manual parameter tuning. For gathering high resolution cloth wrinkle data, we use a fast compliant dynamics simulator to quickly determine equilibrium configurations. We further show how the chosen simulator can be extended with our frictional contact model, which would allow for estimating dynamic parameters like damping and friction.

Chapter 2

Background

In this chapter we establish the context for describing the simulation of hyperelastic objects subject to frictional contact. Namely, we specify the spatial discretization used to simulate elastodynamics problems. We then define all of the involved forces and specify the mechanism for applying traction and contact forces to the system. Finally we summarize the time discretization methods used to discretize the resulting system of ordinary differential equations (ODEs). This sets a common set of definitions and concepts to use throughout the remainder of the dissertation.

2.1 Generalized coordinates

Consider a system of solids represented with m generalized coordinates $\mathbf{q}(t) \in \mathbb{R}^m$ with generalized velocities $\mathbf{v}(t) := \dot{\mathbf{q}}(t) \in \mathbb{R}^m$ at some time $t \in \mathbb{R}$. The generalized mass, denoted by $\mathbf{M}$, is a constant sparse symmetric positive definite (SPD) $m \times m$ matrix. Generalized coordinates serve as an abstraction for a discrete space of coordinates that uniquely identify the configuration of a system of solid bodies. This lets us describe dynamics for soft solids or cloth where $\mathbf{q}$ can represent vertex positions, reduced coordinate systems or systems of rigid bodies.

In this work, we focus solely on dynamics of soft solids and cloth, which are represented by vertices connected by tetrahedra and triangles. This

means $m = 3n$ where n is the number of vertices and $\mathbf{q}$ and $\mathbf{v}$ are simply stacked vectors of vertex positions and velocities respectively. However, contact surfaces are represented using a different mechanism, which is introduced in Chapter 3.

2.2 Equations of Motion

It is not uncommon to begin with a variational principle to derive equations of motion for a dynamical system. Using this method one may lead to sophisticated variational time integrators with desirable properties like momentum and energy preservation, even in the presence of contacts [67] and friction [66]. However, we instead start directly with the equations of motion to maintain flexibility in the choice of time integration schemes.

The equations of motion for a non-conservative dynamical system can be written in configuration space as

$$\mathbf{M}\frac{d\mathbf{v}}{dt} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}), \tag{2.1}$$

where $\mathbf{f}(t, \mathbf{q}, \mathbf{v}) \in \mathbb{R}^m$ is the total generalized force. To illustrate the full complexity of solving dynamics in the graphics pipeline, we decompose the total force into all possible components, which can often require different integration schemes:

$$\mathbf{f}(t, \mathbf{q}, \mathbf{u}) = \mathbf{f}_g(t) + \mathbf{f}_d(\mathbf{q}(t), \mathbf{u}(t)) + \mathbf{f}_e(\mathbf{q}(t)) + \mathbf{f}_c(\mathbf{q}(t)) + \mathbf{f}_f(\mathbf{q}(t), \mathbf{u}(t)), \tag{2.2}$$

where $\mathbf{f}_g$ is the uniform external force (e.g. gravity), $\mathbf{f}_e$ gives the conservative elastic force, $\mathbf{f}_d$ is a dissipative (damping) force, $\mathbf{f}_c$ is the contact force and $\mathbf{f}_f$ gives the friction force. In the remaining sections, we omit the time parameter for brevity.

2.3 Elasticity and Damping

Elastic forces are typically derived from a configuration dependent energy potential $W(\mathbf{q})$ as

$$\mathbf{f}_e(\mathbf{q}) = -\frac{\partial}{\partial \mathbf{q}} W(\mathbf{q})$$

The elastic potential W can be defined by the classic linear, neo-Hookean, St. Venant-Kirchhoff (StVK) or Mooney-Rivlin models [33], or even a data driven model [142]. Here, we focus on neo-Hookean materials for both solids and cloth. The *stiffness matrix* $\mathbf{K}(\mathbf{q}) = -\frac{\partial}{\partial \mathbf{q}} \mathbf{f}_e(\mathbf{q})$ dictates how resistant an object is to deformation. For nonlinear models like neo-Hookean elasticity, $\mathbf{K}$ may be indefinite, which is important to know when picking an appropriate solver.

Damping forces are often defined by

$$\mathbf{f}_d(\mathbf{q}, \mathbf{v}) = -\mathbf{D}(\mathbf{q})\mathbf{v}$$

where $\mathbf{D}$ is square symmetric matrix that may be indefinite. A common choice for $\mathbf{D}$ is the Rayleigh damping model [120] where $\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$ for some constants $\alpha, \beta \geq 0$. There are also more sophisticated models that improve the damping behaviour for nonlinear elasticity [93]. An accurate damping model is important for lively dynamic animations, however further discussion falls outside the scope of this dissertation.

2.4 Contact mechanics

Contacts occur between two points on two distinct surfaces. In this work we focus specifically on point contacts, where force and velocity quantities are defined at localized points in space where individual contacts occur. Thus, the first thing we require is a mapping between the configuration space and the space where individual contacts occur, which may not coincide with vertex positions. We define the *contact Jacobian* of a point i on a solid in

contact with a fixed external point as

$$\mathbf{J}_{c,i} := \frac{\partial \mathbf{x}_i}{\partial \mathbf{q}}, \tag{2.3}$$

which maps configuration velocities to velocities in physical Euclidean space at the point of contact:

$$\mathbf{v}_i := \dot{\mathbf{x}}_i = \mathbf{J}_{c,i}\mathbf{v}. \tag{2.4}$$

Similarly, forces in physical space can be mapped into the configuration space. Suppose $\mathbf{f}_i$ is a force at the point i on a solid in physical space, then from the principle of virtual work, $\mathbf{J}_{c,i}^\top \mathbf{f}_i$ corresponds to the same force 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 contact i occurs between points a and b on two distinct surfaces (possibly belonging to the same object). Then the relative velocity between the two points is given by

$$\bar{\mathbf{v}}_i = \mathbf{v}_a - \mathbf{v}_b = (\mathbf{J}_{c,a} - \mathbf{J}_{c,b})\mathbf{v}.$$

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

$$\mathbf{f}_a = -\mathbf{f}_b.$$

Then the frictional contact force on the whole system can be written as

$$\mathbf{J}_{c,a}^\top \mathbf{f}_a + \mathbf{J}_{c,b}^\top \mathbf{f}_b = (\mathbf{J}_{c,a} - \mathbf{J}_{c,b})^\top \mathbf{f}_a.$$

This produces a global *contact Jacobian* matrix $\mathbf{J}_c$ consisting of triplets of rows for each contact between points a and b defined by $\mathbf{J}_{c,a} - \mathbf{J}_{c,b}$. Thus $\mathbf{J}_c$ is a $3n \times m$ matrix where n is the total number of contacts.

We can now write the combined frictional contact force $\mathbf{f}_{fc} = \mathbf{f}_f + \mathbf{f}_c$ in

terms of the stacked forces in physical space as $\mathbf{J}_c^\top \mathbf{f}_{fc}$.

As we move our focus towards time discretization, it is important to remember that $\mathbf{J}_c$ is a function of $\mathbf{q}$, which is time dependent.

2.5 Time integration

Here we introduce a variety of implicit time integration methods for discretizing Eq. (2.1) in time. Purely explicit integration schemes are omitted since they prohibitively restrict the admissible time step size in stiff problems — contact and friction can produce extremely large forces causing instability that is intrinsic to the problem we are trying to solve.

Using standard notation, we assume that at time t we know $\mathbf{q} = \mathbf{q}^t$ and $\mathbf{v} = \mathbf{v}^t$ and employ a step size h to proceed forward in time. The integration methods we consider can be expressed by the momentum balance equation

$$\mathbf{0} = \mathbf{r}(\mathbf{v}^{t+h}; h, \mathbf{f}, \mathbf{M}), \tag{2.5}$$

where we use superscripts to indicate time. Each integration scheme is characterized by one or more residual functions $\mathbf{r}$ used to determine the final velocity $\mathbf{v}^{t+h}$. Except for trapezoidal rule, all integrators we consider are L-stable, indicating that they dampen errors for stiff and highly oscillatory or unstable problems. Our smooth problem formulation can be unstable (see Section 4.5.2), however, we know that frictional contact is naturally dissipative, and so we expect solutions to behave stably. L-stability ensures that any additional stiffness present in the system will not destabilize the numerical solution. For further discussion on stability see [12].

2.5.1 Backward Euler

The simplest implicit scheme is backward Euler (BE), which is typically defined by

$$\mathbf{r}_{\text{BE}}(\mathbf{v}^{t+h}) = \mathbf{M}(\mathbf{v}^{t+h} - \mathbf{v}^t) - h\mathbf{f}(\mathbf{q}^t + h\mathbf{v}^{t+h}, \mathbf{v}^{t+h}) = \mathbf{0}. \tag{2.6}$$

This integrator is popular in computer graphics due to its supreme stability, however it suffers from time-step dependent numerical dissipation, which can make simulations unpredictable and highly dynamic simulations impractical.

2.5.2 Backward differentiation formula

Backward differentiation formula (BDF) is a family of integration schemes aimed at solving initial value ODEs of the form $\dot{u} = f(u)$. In general, it can be written as

$$\sum_{k=0}^{s} a_k u^{t+h(k-s+1)} = h\beta f(u^{t+h})$$

where the coefficients a_k and β are chosen so that the method achieves order s. A unique integrator of order s is named BDFs, where BDF1 for instance is backward Euler. We are particularly interested in BDF2, where $a_0 = 1/3$, $a_1 = 4/3$, $a_2 = 1$ and $\beta = 2/3$, which yields

$$\mathbf{r}_{\text{BDF2}}(\mathbf{v}^{t+h}) = \mathbf{M}\left(\mathbf{v}^{t+h} - \frac{4}{3}\mathbf{v}^{t} + \frac{1}{3}\mathbf{v}^{t-h}\right) - \frac{2}{3}h\mathbf{f}(\mathbf{q}^{t+h}, \mathbf{v}^{t+h}) = \mathbf{0}$$
$$\text{where } \mathbf{q}^{t+h} = \frac{4}{3}\mathbf{q}^{t} - \frac{1}{3}\mathbf{q}^{t-h} + \frac{2}{3}h\mathbf{v}^{t+h}.$$

This scheme has substantially better energy conservation properties when compared to BE, however it still suffers from time-step dependent numerical dissipation though to a lesser extent. Furthermore, unlike Runge-Kutta described below, BDF2 is a two-step method, which requires higher smoothness from the global solution.

2.5.3 Trapezoidal rule

A well known method for mixing explicitly and implicitly determined forces on the system is the trapezoid rule (TR). We define the TR residual by

$$\mathbf{r}_{\text{TR}}(\mathbf{v}^{t+h}) = \mathbf{M}(\mathbf{v}^{t+h} - \mathbf{v}^t) - \frac{h}{2}\left(\mathbf{f}(\mathbf{q}^t, \mathbf{v}^t) + \mathbf{f}(\mathbf{q}^{t+h}, \mathbf{v}^{t+h})\right) = \mathbf{0}$$
$$\text{where } \mathbf{q}^{t+h} = \mathbf{q}^t + \frac{h}{2}\left(\mathbf{v}^t + \mathbf{v}^{t+h}\right). \tag{2.7}$$

Notably, this method is equivalent to the most commonly used implicit Newmark-β [53, 63, 108] method with $\beta = 1/4$ and $\gamma = 1/2$. The frictional contact problem we address in Chapter 4 is not stable, and can generate large stiffnesses for high elastic moduli, large deformations or due to contact and friction. The smoothed frictional contact problem produces high frequency oscillations, which are exacerbated by TR, whereas ideally we want these to be damped away. See Section 4.5.2 for a concrete example. In spite of these flaws, TR is still used in practice, and often decoupled from the frictional contact problem [84]. In Section 4.6.1 we demonstrate how a properly coupled TR as defined in Eq. (2.7) can resolve some instabilities in practice. Fortunately, these issues disappear when a method with more numerical dissipation is interleaved with TR.

2.5.4 TR-BDF2

The two aforementioned methods can be interleaved producing a second order two stage method over one step from t to $t + h$. The method consists of solving the TR equation for a half step followed by a half BDF2 step. This method is known to be L-stable [13].

2.5.5 Singly diagonal implicit Runge-Kutta

Increasingly popular [92] is the singly diagonally implicit Runge-Kutta (SDIRK) family of methods categorized by the lower triangular structure of the corresponding Butcher tableau [9]. We are specifically interested in the two stage second order method, which we will refer to as SDIRK2. This method can be expressed in two stages. The first stage is BE applied for a

time step αh where $\alpha = 1 - 1/\sqrt{2}$, and the second can be written as

$$\begin{aligned}\mathbf{r}_{\text{SDIRK2}}(\mathbf{v}^{t+h}) &= \mathbf{M}(\mathbf{v}^{t+h} - \mathbf{v}^t) \\ &\quad - h\left((1-\alpha)\mathbf{f}(\mathbf{q}^{t+\alpha h}, \mathbf{v}^{t+\alpha h}) + \alpha \mathbf{f}(\mathbf{q}^{t+h}, \mathbf{v}^{t+h})\right) = \mathbf{0} \\ \text{where } \mathbf{q}^{t+h} &= \mathbf{q}^t + h\left((1-\alpha)\mathbf{v}^{t+\alpha h} + \alpha \mathbf{v}^{t+h}\right)\end{aligned}$$

Incidentally, it is not difficult to show that if $\mathbf{f}$ is linear, then SDIRK2 is equivalent to TR-BDF2 where the initial TR step is larger than 1/2. The numerical dissipation properties of SDIRK2 and TR-BDF2 are generally expected to be similar [13].

2.5.6 Summary

Here we have shown how to apply some of the more complex integrators to our system. Although many scenarios are highly dissipative, where BE is an adequate scheme, many cases require more lively animation to appear physical, for instance when simulating ball sports or interacting stiff objects. Such cases require stable higher-order time integration techniques with good dissipative properties, which we have shown can be easily applied in our framework. However, the specific visual phenomena unlocked related to friction in these cases remains to be investigated.

Chapter 3

Frictional contact on smooth elastic solids

3.1 Introduction

Two major issues plague frictional contact research. The first is that the laws governing macroscopic frictional contact behaviour are non-smooth. They require integrating smooth differential equations with strict inequality constraints due to contact and friction, leading to non-smooth jumps in forces and velocities. The second issue is precisely the coupling between forces and velocities at the point of contact that precludes a straightforward formulation of the frictional contact problem as energy minimization [40].

Non-smoothness has been a major focus of previous research. Following the seminal work of Moreau [105] there has been a substantial amount of progress towards understanding this aspect of frictional contact. Unsurprisingly, this topic has motivated the foundations of convex and variational analysis [121]. It is now well understood that non-smoothness is necessary for modelling absolute sticking behaviour in finite time.

Practical methods for applying friction forces involve "linearizing the friction cone", or discretizing the space of admissible friction forces, which creates artificial anisotropy in friction responses. Although these methods are efficient, they are known to be inaccurate. In this work we propose

a simple and efficient solution to computing isotropically accurate friction forces.

Frictional forces are also sensitive to non-smoothness in the representing geometry. While commonly used polygonal surfaces are convenient when simulating deformable or rigid objects, they can be problematic when used for frictional contact resolution. In particular, their piecewise linear nature makes frictional contact solutions highly dependent on resolution. Low resolution meshes can cause unwanted artifacts when used to represent otherwise smooth objects.

We address this problem by using a smooth implicit surface for representing the collision geometry. In contrast to other parametric representations of smooth surfaces (e.g., subdivision or non-uniform rational B-spline (NURBS) surfaces), implicit representations provide a natural way to define the contact problem via inequality constraints. Since our surface is smooth, this works reliably with gradient based solvers. In contrast, popular contact formulations using distance functions to piecewise linear surfaces require additional smoothing or specialized non-smooth solvers.

It has been established that in general friction and contact are not independent forces. Given a sufficiently stiff dynamical system, a contact point is in equilibrium (has zero relative velocity) when generalized friction and contact forces are exactly balanced [47, 69, 117]. This implies that an accurate method for resolving frictional contacts advances the system along a velocity field that balances friction and contact simultaneously at each time step. This observation implies that friction and contact forces must be integrated implicitly with respect to the assumed velocity field. This ensures accurate stopping behaviour at the end of each time step [21].

In this chapter we develop a non-smooth method for resolving frictional contacts between objects with intrinsically smooth surfaces (meaning surfaces with no or few sharp edges or corners). In summary, the main contributions proposed in this chapter 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 sophisticated 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 3.3 establishes the specific formulation of the problem. Section 3.4 describes of our implicit surface model. Our specific friction model is then Section 3.5.3. The results are presented in Section 3.6 and followed by a summary of limitations and concluding statements in Section 3.7.

3.2 Related work

Simulation of realistic elastic objects subject to frictional contacts 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, flexible shells like cloth and deformable rods like hair. Some methods employ particles to represent the deformable media, from smoothed particle hydrodynamics (SPH) [19] to material point methods (MPM) [56, 58] with full frictional contact formulations as well as position based dynamics (PBD) [107] and its generalizations like projective dynamics [24] and quasi-Newton methods [88]. The most popular method for simulating elastic bodies — perhaps due to the sheer volume of available literature — remains FEM. Some methods employ discontinuous discretizations [45, 71] while others rely on continuously linked meshes [126] 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 [17, 76] 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.

3.2.1 Implicit surfaces

Implicit surfaces have been widely used to represent 3D geometry in literature. From modelling and animation [135, 137] to simulating contact between deformable solids. A popular Eulerian method for using implicit surfaces for deformable media was introduced by Osher et al. [114]. This method is often augmented with particles for improving surface tracking, especially in rapidly deforming media with many topology changes like fluids [46]. Lagrangian variations also exist [61]. In contrast, we rely on FEM to compute elasticity and dynamics equations, while using the implicit surface solely for contact resolution and friction. Implicit functions have been widely used in computer graphics for resolving contact between soft bodies. Desbrun and Cani [28, 29, 41], developed a series of methods for resolving contacts between soft solids represented by radial functions defined around skeletal structures. These methods have very compact data representations and are very efficient, which makes them appropriate for real-time animation applications. However, they require extensive parameter tuning and generally lack the high fidelity deformation we seek here. Later, McAdams et al. [102] proposed a collision detection scheme using signed distance functions for representing collision geometry. This method was further extended to handle cloth collisions [97], which can likewise be applied in our formulation. Our formulation defers from [102] in that we do not require an embedded method for resolving collisions in material space, and unlike the signed distance function, our implicit surface is smooth, which improves solver conver-

gence. Vaillant et al. [136, 137] 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. Implicit functions have been widely used in surface reconstruction. Particularly the siminal work of Alexa et al. [8] popularized the moving least squares (MLS) methodology for surface reconstruction in computer graphics [32]. MLS uses an implicit surface to approximate geometry, similarly involving all points per query, however no global linear solve is required. MLS has difficulties representing sharp features [110]. Interestingly, this is a feature when it comes to formulating contact constraints, since smooth boundaries yield better convergence in constrained optimization involving constraint derivatives. Indeed, methods using distance functions for collision detection are often forced to explicitly smooth the constraint function [84]. This motivates our use of an inherently smooth, local MLS contact surface representation not unlike existing surface reconstruction methods [3].

3.2.2 Frictional contact

We review numerical methods for resolving frictional contacts in the general case of rigid or deformable solids. We are particularly interested in the trade-offs between accuracy, simplicity and scalability. At a high level, frictional contact methods 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 [74, 104] to emulate elasticity. This method is particularly efficient and hence popular in real-time computer graphics. Additionally, penalty-based methods are simple and allow various interacting physical systems to be solved by completely different software packages. As such, these methods are also popular in offline animation and visual effects

tools like Houdini [125], which uses penalties to model contact between various hair, cloth, rigid body and FEM solvers. Unfortunately, this method suffers from stiffness and stability issues, and often requires complex post-processing steps to resolve infeasible configurations. Penalties with low stiffness coefficients can often fail to resolve interpenetrations, however higher stiffnesses may require very small time steps to accurately resolve collisions [16].

Friction can also be modelled via an applied viscous force proportional to tangential predictor velocity between touching surfaces. This approach is especially useful for resolving friction within deformable solids [18]. Unfortunately, viscosity based friction lacks the ability to model the biphasic nature of dry friction.

Alternatively, instead of guessing what penalty forces would lead to a collision free state (and with the correct friction behaviour), we may choose the desired configuration directly [18, 107] or pick the exact forces that produce the target positions and/or velocities [129]. These methods are known as *explicit constraint-based* since they impose 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 where contact points are not linear with respect to the degrees of freedom.

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 [21]. Solving for contact and Coulomb friction simultaneously remains a challenging problem because it poses nonlinear 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 [90]. An artifact of this approximation is that two solids with different weights will start sliding at the same time on

an inclined surface. Kaufman et al. [68] 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 [117]. Kaufman et al. [69] 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 behaviours since it requires a linearization of the friction cone that imposes a discretization dependent anisotropy on the sliding direction as noted by Acary and Brogliato [1]. Later, Bertails-Descoubes et al. [21] used the Alart-Curnier [7] formulation for frictional contact to produce a non-smooth Newton-Raphson root finding technique for hair friction. This was then extended by Daviet et al. [39] to improve scalability using a Gauss-Seidel approach in conjunction with an alternative, smoother Fischer-Burmeister formulation of frictional contact. The work by Daviet et al. was subsequently used to further improve efficiency with adaptive non-linearity [70]. Erleben [48] 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. [83] extended these ideas to adaptive cloth simulation based on nodal constraints. Mazhar et al. [101] followed a slightly different direction to develop an accelerated projected gradient descent method to solve rigid body dynamics subject to frictional contacts. Macklin et al. [96] produce an efficient solver for resolving frictional contact between rigid and deformable objects in interactive simulations. Verschoor and Jalba [139] extend the conjugate residual method to resolve frictional contact simultaneously with elasticity. Li et al. [84] and Geilinger et al. [52] followed a common practice in mechanical engineering and relaxed the non-smooth Coulomb friction formulation into a smooth approximation.

While recent developments have drastically improved the scalability, accuracy and robustness of implicit frictional contact solutions, they rely on highly customized solvers, which makes them difficult to generalize. In contrast to previous work, here we aim to develop a method that fits into existing simulation pipelines more easily. In this chapter, we aim to reuse existing

optimization tools, thereby making implementation simpler and replicability easier.

In the area of frictional contact, our work mainly focuses on fidelity and method extensibility. For a thorough review of non-smooth methods for dynamical problems we refer the reader to [1].

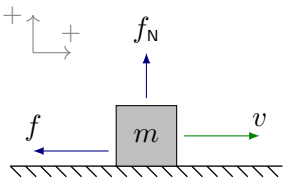
3.2.3 Friction models

An important aspect of numerical solutions for frictional contact is the friction model itself. By a wide margin the majority of the works mentioned above assume the Coulomb friction model or similar. However, friction is extremely difficult to model accurately even in simple scenarios, since the physics causing this phenomenon can be vastly different. Friction models can be split into two categories: static and dynamic [118]. Static models assume that during stiction, there is zero relative motion between the contacting bodies. In reality, a small amount of presliding displacement occurs, which is handled by dynamic models.

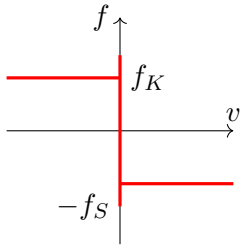
Coulomb friction [37], depicted in Figure 3.1b, is one of the oldest *static* models. It describes a phenomenon often referred to as dry friction, which is arguably the most common type of friction we typically encounter in everyday life. Coulomb friction postulates that friction is proportional to the normal load and opposite to the relative velocity of the contacting objects. A common extension to Coulomb friction is the Stribeck model shown in Figure 3.1c, which was developed in the early 20th century by Richard Stribeck and Mayo D. Hersey [60, 131, 132]. This model distinguishes static friction from kinetic, and adds a viscosity term to model damping due to lubricated contact. The static models have seen many adjustments [14, 75, 81, 140], which primarily aim at providing smooth approximations to the non-smooth friction models to improve and simplify the computational methods required to resolve friction forces. A stepwise friction model can be easily smoothed by the approximation $\operatorname{sgn}(x) \approx \frac{2}{\pi} \arctan(\epsilon x)$ with approximation parameter ϵ [140] as shown in Figure 3.1d.

Further advances in friction modelling were introduced by Dahl [38] in

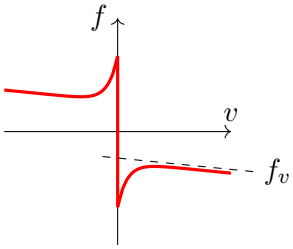
1968, which introduced a displacement dependency in presliding friction — a small elastic force that pulls contacts together before they break apart. The Dahl model has been extended to account for the Stribeck effect with the LuGre model [30]. Hysteretic effects have also been handled in the Leuven model [134]. Dynamic models typically propose to dynamically switch between the stick phase and macroscopic sliding, which can be too complex for many practical engineering purposes [145].



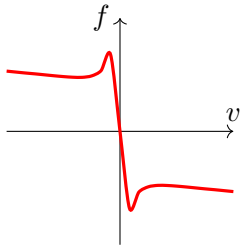
(a) A mass m is sliding with velocity v is subject to a friction force f and a normal force f_N. The force of gravity is implied.



(b) Coulomb friction. When the mass is stationary, the object may admit a force of static friction in the range $[-f_S, f_S]$. A moving object, however is subject to kinetic friction f_K if $v < 0$ or $-f_K$ if $v > 0$.



(c) Stribeck model. Static friction smoothly transitions into kinetic, which is positively dependent on relative velocity. For larger velocities, friction force can increase linearly with velocity as determined by the slope f_v.



(d) Smoothed Stribeck model. This model smoothes the Stribeck model at $v = 0$, by using the approximation $\frac{2}{\pi}\arctan(\epsilon v) \xrightarrow{\epsilon\to\infty} \operatorname{sgn}(v)$.

Figure 3.1: A variety of friction curves are plotted with relative velocity v on the horizontal axis and friction f on the vertical for a scenario depicted in (a).

Another important body of tribology research focuses on estimating the magnitudes of kinetic and static friction coefficients, as well as analysing their dependence on the load parameter. Indeed a general power law is typically used to monitor this dependency where $f = \alpha f_{\mathsf{N}}^{k}$ where k is called load index. Many experiments measuring friction on human skin found significant decrease in k when contacting surfaces are wetted [2] indicating that a significant portion of the friction force is due to adhesion. Prior work studying friction between skin and textiles concluded that

> Moisture on the skin is more important than the fiber type or the fabric construction parameters tested in these experiments in determining fabric-to-skin friction.
>
> - Kenins [73]

Other research studying textiles in isolation, found non-trivial friction forces between textiles with a negligible amount of normal force [57]. These findings generally support the idea that adhesion plays a significant role in friction phenomena, especially on textiles or moist surfaces like human skin.

To the best of our knowledge, adhesion in frictional contact methods has been largely unexplored in computer graphics. Mazhar et al. [101] formulate a cohesive frictional contact model. Although their focus is on scalability and performance, they take a step towards a more general contact model allowing negative normal forces to produce sticking. Adhesion in friction, however, is fundamentally different in that it relates the force of friction to the area of contact between contacting surfaces. Originally Bowden and Tabor [25] studied the friction between lubricated metals to find a two-term non-interacting friction model. This model was later adapted more generally for studying rolling friction for elastic lubricated surfaces [54], and more recently friction between human skin [2] and smooth rigid solids.

While adhesion is not explored in this dissertation, related works are included here for completeness, and we note that adhesion is a fruitful area for future research in computer graphics.

3.3 Formulation

In this section, we lay the groundwork for the algorithms introduced in this chapter.

3.3.1 Nonlinear equations as an optimization problem

As discussed in Section 3.1, frictional contact must be resolved implicitly to guarantee proper sticking behaviour. Here, we aim to develop an implicit algorithm with particular attention to friction and contact.

External and dissipative forces other than friction are integrated explicitly and are omitted from our exposition for brevity. In this chapter we use backward Euler time integration defined in Eq. (2.6). This scheme is unconditionally stable, and simple enough to allow us to focus on modelling contacts.

The nonlinear Eq. (2.5) with Eq. (2.6) form the necessary condition for the minimum of

$$\min_{\mathbf{v}} \frac{1}{2} \left\| \mathbf{v} - \mathbf{v}^t \right\|_{\mathbf{M}}^2 + W(\mathbf{q}^t + h\mathbf{v}) + W_{fc}(\boldsymbol{\lambda}, \mathbf{v}) \tag{3.1}$$

where W_{fc} is some frictional contact potential parametrized by some $\boldsymbol{\lambda}$ such that

$$\mathbf{f}_{fc} = -\frac{1}{h} \frac{\partial W_{fc}}{\partial \mathbf{v}}.$$

Note that in general, $\mathbf{f}_{fc}$ cannot be represented as a derivative of some potential, but it can here, since $\boldsymbol{\lambda}$ is fixed. The mass induced norm is defined by $\|\mathbf{z}\|_{\mathbf{M}}^2 := \mathbf{z}^\top \mathbf{M}\mathbf{z}$ for all $\mathbf{z} \in \mathbb{R}^m$, and $W(\mathbf{q}^t + h\mathbf{v})$ is the total implicit elastic strain energy. The negative derivative of strain energy produces the desired elastic forces

$$\mathbf{f}_e = -\frac{\partial W}{\partial \mathbf{q}}.$$

We use the standard nonlinear neo-Hookean strain energy [126] or its stable variation [128], although other choices also work. We haven't yet defined W_{fc}, however we have established that all implicit potentials and forces of interest are effectively functions of the velocity $\mathbf{v}$.

This formulation is convenient because it can easily be extended to include physical constraints forming a constrained optimization problem, which is arguably more extensively studied than constrained ODE problems. Furthermore, local minima in (3.1) correspond to valid physical configurations, whereas methods for solving nonlinear systems of equations have weaker global convergence guarantees [109].

3.3.2 Time splitting

In this section we will briefly present the time splitting technique. Suppose our momentum step is driven by some two implicit forces $\mathbf{f}_a$ and $\mathbf{f}_b$ as follows

$$\mathbf{v}^{t+h} = \mathbf{v}^t + h\mathbf{M}^{-1}\left(\mathbf{f}_a^{t+h} + \mathbf{f}_b^{t+h}\right). \tag{3.2}$$

We can split Eq. (3.2) by solving for the contributions of $\mathbf{f}_a$ and $\mathbf{f}_b$ separately:

$$\mathbf{v}^* = \mathbf{v}^t + h\mathbf{M}^{-1}\mathbf{f}_a^*, \tag{3.3a}$$

$$\mathbf{v}^{t+h} = \mathbf{v}^* + h\mathbf{M}^{-1}\mathbf{f}_b^{t+h}, \tag{3.3b}$$

This technique yields a first order integrator regardless of the accuracy in each solve, which is often sufficient for graphics applications. Time splitting is sometimes referred to as the predictor-corrector method [69] and is widely used to separate elasticity and friction solves [21, 39, 48].

Furthermore, if we maintain that the forces can be derived from an energy potential, then one or both of the Eqs. (3.3) can still be solved by an optimization like in (3.1).

Our goal here is to develop a method for solving (3.1) for elastic solids with boundaries represented by smooth implicit surfaces where $\mathbf{f}_{fc}^{t+h}$ is constrained to satisfy friction conditions of a static friction model like Coulomb friction.

Note that the frictional contact force $\mathbf{f}_{fc}$ is unknown and (possibly non-uniquely) determined by the configuration $(\mathbf{q}, \mathbf{v})$. Unfortunately, in general $\mathbf{f}_{fc}$ cannot be derived from a single energy potential, however it can be

constructed from two coupled potentials [40]. In Section 3.5.3 we propose an algorithm to solve for $\mathbf{f}_{fc}^{t+h}$ by iteratively solving a split version of (3.1).

3.3.3 Force forwarding

To mitigate some of the artifacts introduced by time splitting, we propose to forward the computed forces into the next time step. For instance, continuing with the example from the previous section, we observe that $\mathbf{f}_b^{t+h}$ is a known quantity at the end of the time step, which can be used to drive the next time step as follows

$$\mathbf{v}^* = \mathbf{v}^t + h\mathbf{M}^{-1}\left(\mathbf{f}_a^* + \mathbf{f}_b^t\right), \tag{3.4a}$$

$$\mathbf{v}^{t+h} = \mathbf{v}^* + h\mathbf{M}^{-1}\left(\mathbf{f}_b^{t+h} - \mathbf{f}_b^t\right). \tag{3.4b}$$

This produces a slightly different intermediate velocity $\mathbf{v}^*$ but is otherwise equivalent to Eq. (3.3). This facilitates closer coupling between $\mathbf{f}_a$ and $\mathbf{f}_b$ than the initial splitting in Eq. (3.3).

Alternatively, starting again with Eq. (3.3), we can reintroduce the predicted $\mathbf{f}_a^*$ from the intermediate step back into the final step to ensure that both $\mathbf{f}_a$ and $\mathbf{f}_b$ are solved implicitly for the end of the time step (for instance this is necessary for friction and contact impulses to generate correct sticking behaviour). This technique will then look like

$$\mathbf{v}^* = \mathbf{v}^t + h\mathbf{M}^{-1}\mathbf{f}_a^*, \tag{3.5a}$$

$$\mathbf{v}^{t+h} = \mathbf{v}^* + h\mathbf{M}^{-1}\left(\mathbf{f}_b^{t+h} + \mathbf{f}_a^{t+h} - \mathbf{f}_a^*\right), \tag{3.5b}$$

This change reduces to the original implicit system in Eq. (3.2) simply by eliminating $\mathbf{v}^*$. Although solving Eq. (3.5a) and then immediately subtracting the result in Eq. (3.5b) may seem redundant, it makes a difference if $\mathbf{f}_a$ is combined with another force in Eq. (3.5a), thus facilitating additional coupling between the two forces. To demonstrate this, we can apply the techniques above to the elasticity and frictional contact forces $\mathbf{f}_e$ and $\mathbf{f}_{fc}$ respectively. First, we split $\mathbf{f}_e$ from $\mathbf{f}_{fc}$ as in Eq. (3.3). Then after

splitting frictional contact force into its friction and contact constituents: $\mathbf{f}_{fc} = \mathbf{f}_c + \mathbf{f}_f$, we forward the friction force $\mathbf{f}_f$ as in Eq. (3.4), and reintroduce the intermediate contact force $\mathbf{f}_c$ into the final step as in Eq. (3.5). The final system is given by

$$\mathbf{v}^* = \mathbf{v}^t + \mathbf{M}^{-1}\left(\mathbf{f}_e^* + \mathbf{f}_c^* + \mathbf{f}_f^t\right), \tag{3.6a}$$

$$\mathbf{v}^{**} = \mathbf{v}^* - \mathbf{M}^{-1}\left(\mathbf{f}_c^* + \mathbf{f}_f^t\right), \tag{3.6b}$$

$$\mathbf{v}^{t+h} = \mathbf{v}^{**} + \mathbf{M}^{-1}\left(\mathbf{f}_c^{t+h} + \mathbf{f}_f^{t+h}\right), \tag{3.6c}$$

which effectively couples $\mathbf{f}_e$ with $\mathbf{f}_c$ in Eq. (3.6a) and $\mathbf{f}_c$ with $\mathbf{f}_f$ in Eq. (3.6c). This is precisely the splitting technique we apply to friction and contact forces in Section 3.5.3, which we call *friction forwarding*. In the sections to follow, we develop the necessary constraints on these forces, to solve the frictional contact problem.

3.4 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 3.2. 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 [21, 39, 83, 84]. 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.

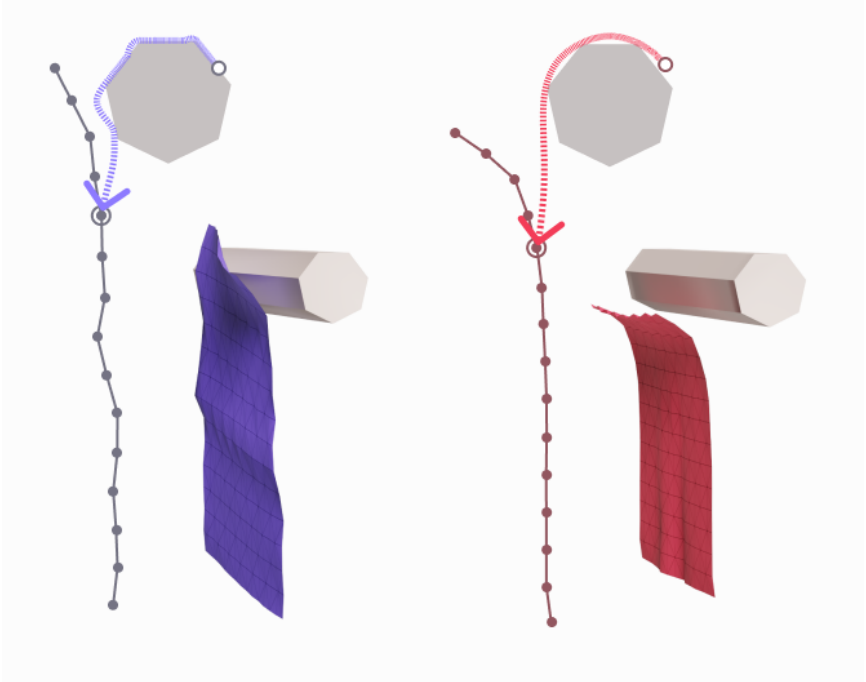


Figure 3.2: *Smooth sliding on implicit surface.* A piece of cloth is draped off-center onto a rigid cylinder. It then slips to the side as captured by this illustration. A vertex (circled) on the cloth in both examples is selected and traced from its original position as it slides along the surface of the cylinder. On the left, a polyhedral surface is used to represent the cylinder's contact surface, which produces an irregular contact pattern that causes unwanted vibrations in the cloth. In contrast, a smooth contact surface is used on the right, which produces a more natural smooth trajectory for a sliding vertex.

- To prevent dense derivatives, the potential must have a local 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 $\mathbf{J}_c$.

Some previous work has employed embedded signed distance fields for collision handling [102], however distance fields defined on polygonal surfaces are non-differentiable and can pose serious convergence issues when used as constraints in optimization solvers [84]. This is further illustrated in Appendix A.

A widely successful method proposed by Li et al. [84] used unsigned distance fields to enforce non-penetration. However, this approach requires accurate continuous collision detection, and cannot handle initially intersecting geometry, which is ubiquitous in animation pipelines. We propose to simplify the technique by using existing information about polygon orientations to construct a signed potential. In this way, our method fills a different niche, where initial interpenetrations are tolerated.

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. To show that our implicit surface stays sufficiently close to the polygonal surface, we measure the Hausdorff distance between the two surface representations.

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

3.4.1 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 3.5.3.

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 $\mathbf{s} \in \mathcal{S}$ is endowed with a well defined outward facing unit normal $\mathbf{n_s}$.

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:

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

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

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

$$\psi(\mathbf{x};\mathbf{s}) = \mathbf{n}_\mathbf{s}^\top(\mathbf{x} - \mathbf{s}).$$

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

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

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

It remains to define the neighbourhood $\mathcal{N}$ 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 framework for interpolating and approximating implicit surfaces [124]. However, in contrast to [124], we use a *local* weight function, which is critical for our application because we require sparse derivatives.

Weight function

There are three main properties we require from the weight function:

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

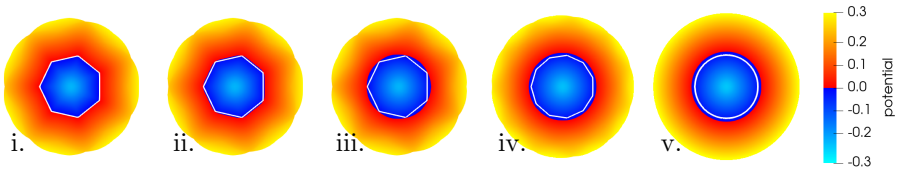


Figure 3.3: A cross section of the potential field $\Psi(\mathbf{x})$ generated by the cylinder from Figure 3.2 with different kernels, parameters and resolutions (last two images). The shape of the cylinder cross section which is used to generate the potential is outlined in white in each of the examples. 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. The radius R is set to be half of the largest triangle diameter in the input mesh. Points outside the colored area have empty neighbourhoods $\mathcal{N}_R$ 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.).

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 presented in [106], which was originally used for interpolation in element-free Galerkin methods. This weight function is defined as

$$w_R(\mathbf{x};\mathbf{s}) := \frac{\tilde{w}_R(\|\mathbf{x}-\mathbf{s}\|_2)}{\sum_{\mathbf{s}\in\mathcal{N}_R(\mathbf{x})} \tilde{w}_R(\|\mathbf{x}-\mathbf{s}\|_2)} \tag{3.7}$$

$$\tilde{w}_R(r) := \frac{\left((r/R)^2+\epsilon\right)^{-2} - (1+\epsilon)^{-2}}{\epsilon^{-2} - (1+\epsilon)^{-2}}, \tag{3.8}$$

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

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

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 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 3.3 shows the potential generated by each kernel for a heptagonal cylinder.

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 $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ as

$$\mathbf{n_s} = \frac{(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_3)}{\|(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_3)\|_2}.$$

To summarize, the equations of motion are solved with FEM on a tetrahedral mesh, where frictional contact forces at mesh vertices are mapped from the implicit surface, which is generated using oriented sample points at triangle centres on the surface of the tetrahedral mesh. While mapping normal

[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.

forces on the implicit surface to mesh vertices is straightforward, mapping tangential forces requires more work to ensure that purely tangential forces cause accurate torques on the FEM mesh. Mapping normal and tangential forces this way is given by the contact Jacobian defined in the next section.

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

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

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

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

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

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

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

which allows us to use the kernel weights to interpolate velocity contributions from mesh vertices. Since $\mathbf{q}$ corresponds to mesh vertices and $\bar{\mathbf{v}}$ is a subset

of those on the surface, we have that $\frac{\partial \bar{\mathbf{v}}}{\partial \mathbf{q}}$ is a simple selection matrix. In our case, sample points are located at surface triangle centroids, which means that $\frac{\partial \mathbf{s}}{\partial \mathbf{v}}$ is the identity matrix scaled by $1/3$ if $\mathbf{v}$ is a vertex of the triangle for sample $\mathbf{s}$ and zero otherwise. It follows that $\frac{\partial \bar{\mathbf{s}}}{\partial \bar{\mathbf{v}}}$ is a sparse block matrix of scaled 3×3 identity matrices.

Finally we can define

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

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

3.5 Frictional contact

In this section we will propose a set of conditions on $\mathbf{f}_{fc}$ to produce a force that satisfies the Coulomb friction model. We also propose a concrete algorithm for solving this problem. Here, the discussion of friction and contact is more convenient with impulses rather than forces, so for the remainder of this chapter, we will use $\mathbf{r} := h\mathbf{f}_{fc}$ to denote the *frictional contact impulse*.

The Coulomb friction model is a reasonable point of departure for formulating friction problems since other static models include Coulomb friction as a component [89]. However, some friction models smooth the friction curve, which can allow for a simpler formulation [52, 84]. In Chapter 4 we propose an alternative method for resolving smooth friction models on smooth implicit surfaces.

3.5.1 Contact space

To set the context for describing Coulomb friction, we start by defining the vector space of impulses and velocities on the set of all contact points $\mathcal{C}$ in our system. Hereafter let $\mathcal{C}$ denote the index set of all potential contact points at time t, with $|\mathcal{C}| = n$.

For any vector $\mathbf{y}_i \in \mathbb{R}^3$ in physical space associated with a contact point

$i \in \mathcal{C}$, we define $\mathsf{y}_{\mathsf{N},i}$ and $\mathsf{y}_{\mathsf{T},i}$ to be the normal and tangential components of $\mathbf{y}_i$ with respect to the contact surface. We call $\mathsf{y}_{\mathsf{N},i}$ and $\mathsf{y}_{\mathsf{T},i}$ contact space coordinates and $\mathsf{y}_i = (\mathsf{y}_{\mathsf{N},i}, \mathsf{y}_{\mathsf{T},i}) \in \mathbb{R}^3$ denotes the whole 3D vector in contact space. In general we use the 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 $\mathbf{B} = [\mathbf{B}_\mathsf{N} \,|\, \mathbf{B}_\mathsf{T}]$ where $\mathbf{B}_\mathsf{N}$ and $\mathbf{B}_\mathsf{T}$ are block diagonal matrices with:

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

where $\mathbf{n}_i$ is the unit normal and $\mathbf{t}_i$ is a tangent direction at contact point $i \in \mathcal{C}$. This allows us to write the normal and tangential components of the stacked vector $\mathbf{y}$ as

$$\mathsf{y}_\mathsf{N} := \mathbf{B}_\mathsf{N}^\top \mathbf{y} \quad \text{and} \quad \mathsf{y}_\mathsf{T} := \mathbf{B}_\mathsf{T}^\top \mathbf{y}, \quad \text{or just} \quad \mathsf{y} := \mathbf{B}^\top \mathbf{y}.$$

3.5.2 Contact

Assume for the moment that $\mathsf{r}_\mathsf{T} = 0$ (or more precisely $\mathbf{B}_\mathsf{T}^\top \mathbf{J}_c^\top \mathbf{r} = 0$). Then contact can be defined by the complementarity condition

$$0 \leq \mathsf{r}_\mathsf{N} \perp \mathsf{v}_\mathsf{N} \geq 0, \tag{SFC}$$

which indicates that only one of $\mathsf{r}_{\mathsf{N},i}$ and $\mathsf{v}_{\mathsf{N},i}$ may be non-zero at any given time for each $i \in \mathcal{C}$, although both must be non-negative at all times. This is called the Signorini-Fichera condition (SFC) [49, 127].

Assuming that $\mathsf{r}_\mathsf{T} = 0$, Eq. (2.6) with Eq. (SFC) become the Karush-Kuhn-Tucker (KKT) conditions for the following constrained optimization problem

$$\begin{aligned} \min_{\mathbf{v}} \quad & \gamma(\mathbf{v}) \\ \text{subject to} \quad & \mathsf{v}_\mathsf{N} \geq 0, \end{aligned} \tag{3.9}$$

abbreviated with $\gamma(\mathbf{v}) := \frac{1}{2}\left\|\mathbf{v}^t - \mathbf{v}\right\|_{\mathbf{M}}^2 + W(\mathbf{q}^t + h\mathbf{v})$ and $\mathsf{v}_\mathsf{N} = \mathbf{B}_\mathsf{N}^\top \mathbf{J}_c \mathbf{v}$ as usual. The Lagrange dual of Eq. (3.9) reveals the role of the contact impulse r_N as the Lagrange multiplier:

$$(\mathsf{r}_\mathsf{N}^{t+h}, \mathbf{v}^{t+h}) = \arg\max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{v}} \gamma(\mathbf{v}) - \boldsymbol{\lambda}^\top \mathsf{v}_\mathsf{N}, \tag{3.10}$$

where $\mathsf{r}_\mathsf{N}^{t+h}$ is the optimal value of the Lagrange multiplier $\boldsymbol{\lambda}$ and $\mathbf{v}^{t+h}$ is the optimal value of the generalized velocity $\mathbf{v}$ as before. The original problem in (3.1) is a subproblem of Eq. (3.10) maximized over $\boldsymbol{\lambda} \geq \mathsf{0}$ where $W_{fc}(\boldsymbol{\lambda}, \mathbf{v}) = -\boldsymbol{\lambda}^\top \mathbf{B}_\mathsf{N}^\top \mathbf{J}_c \mathbf{v}$.

We use the dual form in our exposition since it explicitly lists inputs and outputs of the optimization problem. In practice, we differ to an off-the-shelf nonlinear constrained optimization solver to find the optimal values.

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 3.4) where $\mathbf{q}_{\bar{k}}$ are *all* the coordinates in $\mathbf{q}$ that do *not* affect the deformation of mesh k (this effectively excludes the possibility of self-contact). We limit ourselves to contact between distinct objects[2] 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

$$\Psi[\mathbf{q}_{\bar{k}}](\mathbf{v}) \geq 0 \qquad \forall \mathbf{v} \in \mathcal{V}_k. \tag{3.11}$$

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

$$\phi_{k,i}(\mathbf{q}) := \Psi[\mathbf{q}_{\bar{k}}](\mathbf{v}_i) \qquad \forall \mathbf{v}_i \in \mathcal{V}_k,\ \forall k. \tag{3.12}$$

[2]Distinct objects are represented by meshes with no shared vertices or edges.

This allows us to write the constraint simply as

$$\phi(\mathbf{q}) \geq \mathbf{0}. \tag{3.13}$$

The union of all constrained vertex sets $\mathcal{V}_k$ forms the complete set of contact points in $\mathcal{C}$. Finally with the non-penetration constraint substituting (SFC), the problem becomes

$$(\mathsf{r}_\mathsf{N}^{t+h}, \mathbf{v}^{t+h}) = \arg\max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{v}} \gamma(\mathbf{v}) - \frac{1}{h}\boldsymbol{\lambda}^\top \phi(\mathbf{q}^t + h\,\mathbf{v}), \tag{3.14}$$

where we apply backward Euler to the argument of ϕ such that the constraint is enforced implicitly. Although solving Eq. (3.14) as stated can be expensive since ϕ can be very nonlinear, a common technique is to linearize this constraint about $\mathbf{q}^t$ to simplify the problem in order to achieve better performance at the cost of accuracy.

This formulation also does not preclude initial interpenetrations where a vertex of one object may be located inside the negative of another object's contact potential Ψ. Whether initial interpenetrations are valid or not is determined by the numerical method used to resolve the inequality constraint. As we will see by the end of Section 3.5.3, the final problem we target can be solved by a nonlinear programming package implementing filtered line search [50, 141], which allows initial constraint violations.

3.5.3 Friction

When r_T is allowed to be non-zero, the complete frictional contact impulse r must satisfy the complete Coulomb law. As we mentioned before, r cannot be derived from a single potential but it can be derived by minimizing two coupled potentials [40]. We can use the result we developed in the previous section but reintroduce the unknown tangential impulse r_T into the non-penetration problem in Eq. (3.14) as follows:

$$(\mathsf{r}_\mathsf{N}^{t+h}, \mathbf{v}^{t+h}) = \arg\max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{v}} \gamma(\mathbf{v}) - \frac{1}{h}\boldsymbol{\lambda}^\top \phi(\mathbf{q}^t + h\mathbf{v}) - \mathsf{v}_\mathsf{T}^\top \mathsf{r}_\mathsf{T}(\boldsymbol{\lambda}, \mathbf{v}), \tag{3.15}$$

where $\mathsf{v}_\mathsf{T} = \mathbf{B}_\mathsf{T}^\top \mathbf{J}_c \mathbf{v}$ as before and r_T is a function of the contact impulse $\boldsymbol{\lambda}$ and velocity $\mathbf{v}$, which will become evident in the next section. The original problem (3.1) can now be seen as a subproblem of Eq. (3.15) maximized over $\boldsymbol{\lambda} \geq \mathsf{0}$ as in Eq. (3.10), but this time the frictional contact potential is given by

$$W_{fc}(\boldsymbol{\lambda}, \mathbf{v}) = -\frac{1}{h}\boldsymbol{\lambda}^\top \phi(\mathbf{q}^t + h\mathbf{v}) - \mathbf{v}^\top \mathbf{J}_c^\top \mathbf{B}_\mathsf{T} \mathsf{r}_\mathsf{T}(\boldsymbol{\lambda}, \mathbf{v}).$$

Maximal dissipation principle

For single point contact conditions, frictional impulses for sliding can be derived from the *maximum dissipation principle (MDP)* as described in [105]. MDP states that the frictional impulse at a point of contact $i \in \mathcal{C}$ maximizes negative work:

$$\mathsf{r}_{\mathsf{T},i} = \underset{\mathsf{y} \in \mu \mathsf{r}_{\mathsf{N},i} D}{\operatorname{argmax}} -\mathsf{y}^\top \mathsf{v}_{\mathsf{T},i} = \underset{\mathsf{y} \in \mu \mathsf{r}_{\mathsf{N},i} D}{\operatorname{argmin}} \mathsf{y}^\top \mathsf{v}_{\mathsf{T},i}, \tag{3.16}$$

where $D \subset \mathbb{R}^2$ is the closed unit disc centered at the origin and $\mathsf{v}_{\mathsf{T},i}$ is the tangential relative velocity vector at contact i. The minimization in Eq. (3.16) is trivially solved by

$$\mathsf{r}_{\mathsf{T},i} = -\mu \mathsf{r}_{\mathsf{N},i} \frac{\mathsf{v}_{\mathsf{T},i}}{\|\mathsf{v}_{\mathsf{T},i}\|} \tag{3.17}$$

on the disc when $\mathsf{v}_{\mathsf{T},i} \neq \mathsf{0}$.

We can now extend Eq. (3.16) to all the contacts in $\mathcal{C}$. We define $D_\mathcal{C} : \mathbb{R}^n \to \mathbb{R}^{2n}$ to be the map from stacked contact impulses to a Cartesian product of closed discs in the tangent space of all contact points in $\mathcal{C}$:

$$D_\mathcal{C}(\mathsf{y}) := \prod_{i \in \mathcal{C}} \mathsf{y}_i D$$

where $\mathsf{y}_i \in \mathbb{R}$ is the ith component of y. Then Eq. (3.16) can be rewritten for

all contact points as a function of contact impulses and generalized velocities:

$$\mathsf{r}_\mathsf{T}(\boldsymbol{\lambda}, \mathbf{v}) = \underset{\mathsf{y} \in D_C(\mu\boldsymbol{\lambda})}{\operatorname{argmin}} \ \mathsf{y}^\top \mathbf{B}_\mathsf{T}^\top \mathbf{J}_c \mathbf{v}. \tag{MDP}$$

3.5.4 Frictional contact solutions

The remaining question is how to simultaneously solve the contact and friction problems in Eq. (3.15) and Eq. (MDP) respectively. It is well known that finding a global optimum for both optimization problems is NP-Hard [69], however we can focus on finding an approximation to the global optimum.

It is worth noting that although a closed form solution to Eq. (MDP) exists for non-zero velocities, r_T becomes set valued at zero velocity. In Chapter 4 we reformulate this problem as an inclusion (4.3), and use a smooth approximation technique [52, 84] to combine it with damping and elasticity in a single nonlinear system.

Time-splitting

An important consideration used in [21, 39, 69, 83] is that elastic impulses move through a typical soft solid at a much slower rate than frictional contact impulses. 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 3.2, and explained in

[69, §4], 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 Eqs. (3.15) and (4.2). Instead of splitting the entire frictional contact solve completely from the elasticity optimization as in [21, 69], we opt to split only the incremental update to the friction impulse as we have demonstrated in Section 3.3.3. 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 following Eq. (3.6):

1. Compute the intermediate (predictor) velocity $\mathbf{v}^*$ and contact impulse r_N^* by solving the elasticity problem subject to contact constraints as well as friction impulses from the previous time step:

$$(\mathsf{r}_\mathsf{N}^*, \mathbf{v}^*) = \arg\max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{v}} \gamma(\mathbf{v}) - \frac{1}{h}\boldsymbol{\lambda}^\top \phi(\mathbf{q}^t + h\mathbf{v}) - \mathsf{v}_\mathsf{T}^\top \mathsf{r}_\mathsf{T}^t. \tag{3.18}$$

2. Compute the predictor velocity

$$\mathbf{v}^{**} = \mathbf{v}^* - \mathbf{M}^{-1}\mathbf{J}_c^\top \left(\mathbf{B}_\mathsf{N}\mathsf{r}_\mathsf{N}^* + \mathbf{B}_\mathsf{T}\mathsf{r}_\mathsf{T}^t\right) \tag{3.19}$$

3. Using r_N^* as the initial guess for the contact impulse, and the predictor impulse computed from $\mathbf{v}^{**}$, solve the modified staggered projections as described in Section 3.5.4, to obtain the full friction and contact impulses $\mathsf{r}_\mathsf{T}^{t+h}$ and $\mathsf{r}_\mathsf{N}^{t+h}$ respectively. Then use them to update the velocity

$$\mathbf{v}^{t+h} = \mathbf{v}^{**} + \mathbf{M}^{-1}\mathbf{J}_c^\top \left(\mathbf{B}_\mathsf{N}\mathsf{r}_\mathsf{N}^{t+h} + \mathbf{B}_\mathsf{T}\mathsf{r}_\mathsf{T}^{t+h}\right) \tag{3.20}$$

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

ing Eq. (3.18) into

$$(\mathsf{r}_\mathsf{N}^*, \mathbf{v}^*) = \arg\max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{v}} \gamma(\mathbf{v}) - \boldsymbol{\lambda}^\top \left(\frac{\partial \phi^t}{\partial \mathbf{q}} \mathbf{v} + \frac{1}{h}\phi^t \right) - \mathsf{v}_\mathsf{T}^\top \mathsf{r}_\mathsf{T}^t, \tag{3.21}$$

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

Note that friction forwarding is distinct from warm-start as described in [69]. Our method uses the friction from the previous time step to compute the velocity predictor $\mathbf{v}^*$, whereas Kaufman et al. [69] use the friction from the previous step only as a starting guess for the staggered projections algorithm.

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 $D_\mathcal{C}$.

To summarize we aim to solve Eq. (3.20), subject to (SFC) and Eq. (4.2) as reproduced below

$$\mathbf{v} = \mathbf{v}^{**} + \mathbf{M}^{-1}\mathbf{J}_c^\top \left(\mathbf{B}_\mathsf{N}\mathsf{r}_\mathsf{N} + \mathbf{B}_\mathsf{T}\mathsf{r}_\mathsf{T} \right) \tag{3.22a}$$

$$0 \leq \mathsf{r}_\mathsf{N} \perp \mathbf{B}_\mathsf{N}^\top \mathbf{J}_c \mathbf{v} \geq 0, \tag{3.22b}$$

$$\mathsf{r}_\mathsf{T} = \operatorname*{argmin}_{\mathsf{y}\in D_\mathcal{C}(\mu \mathsf{r}_\mathsf{N})} \mathsf{y}^\top \mathbf{B}_\mathsf{T}^\top \mathbf{J}_c \mathbf{v}. \tag{3.22c}$$

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

$$P_\mathsf{T}(\mathsf{r}_\mathsf{N}; \xi) := \operatorname*{argmin}_{\mathsf{y}\in D_\mathcal{C}(\mu \mathsf{r}_\mathsf{N})} \frac{1}{2} \|\mathbf{B}_\mathsf{T}\mathsf{y} - \xi\|^2_{\mathbf{M}_e^{-1}}. \tag{3.23}$$

where $\mathbf{M}_e^{-1} := \mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^\top$ 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:

$$P_{\mathsf{N}}(\xi) := \underset{\mathsf{y}\geq 0}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{B}_{\mathsf{N}}\mathsf{y} - \xi\|^2_{\mathbf{M}_e^{-1}}. \tag{3.24}$$

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

$$\mathsf{r}_{\mathsf{T}}^k \leftarrow P_{\mathsf{T}}\left(\mathsf{r}_{\mathsf{N}}^{k-1}; \mathbf{z} - \mathbf{B}_{\mathsf{N}}\mathsf{r}_{\mathsf{N}}^{k-1}\right) \tag{3.25a}$$

$$\mathsf{r}_{\mathsf{N}}^k \leftarrow P_{\mathsf{N}}\left(\mathbf{z} - \mathbf{B}_{\mathsf{T}}\mathsf{r}_{\mathsf{T}}^k\right) \tag{3.25b}$$

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

$$\mathbf{p} = -\mathbf{M}_e\mathbf{J}_c\mathbf{v}^{**}. \tag{3.26}$$

Unfortunately as written, Eq. (3.23) is a quadratic problem subject to nonlinear inequality constraints. As such, popular methods for friction simulation [69, 115] 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 [83]. To maintain both speed and accuracy, we choose to reparametrize the contact space into cylindrical coordinates instead, producing a nonlinear problem with bounds constraints. This allows us to rewrite Eq. (3.23) as

$$P_{\mathsf{T}}(\mathsf{r}_{\mathsf{N}}; \xi) := \underset{\substack{\boldsymbol{\theta}\in[0,2\pi]^n \\ 0\leq\boldsymbol{\alpha}\leq\mu\mathsf{r}_{\mathsf{N}}}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{B}_{\mathsf{T}}\mathbf{R}(\boldsymbol{\theta}, \boldsymbol{\alpha}) - \xi\|^2_{\mathbf{M}_e^{-1}}, \tag{3.27}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ are stacked vectors of angles and radii in the tangent plane of each contact respectively. Then $\mathbf{R} : [0, 2\pi]^n \times [0, \infty)^n \to \mathbb{R}^{2n}$ is a nonlinear reparametrization operator that can be defined per contact as

$$\mathbf{R}(\theta_i, \alpha_i)_i = (\alpha_i \cos(\theta_i), \alpha_i \sin(\theta_i)) \quad \text{for all } 1 \leq i \leq n.$$

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
$h \leftarrow$ current time step
$\mathbf{M}_e \leftarrow$ effective mass (Delassus operator)
$\mathbf{M} \leftarrow$ generalized mass matrix
$\mathbf{q}^t, \mathbf{v}^t \leftarrow$ previous generalized positions and velocities
$\mathsf{r}_\mathsf{T}^t \leftarrow$ previous friction impulse

Output: $\mathbf{v}^{t+h} \leftarrow$ generalized velocity for the next time step

begin Solve the constrained optimization problem in Eq. (3.21):

$$(\mathsf{r}_\mathsf{N}^*, \mathbf{v}^*) \leftarrow \begin{cases} \arg\min_{\mathbf{v}} & \gamma(\mathbf{v}) - \mathsf{v}_\mathsf{T}^\top \mathsf{r}_\mathsf{T}^t \\ \text{s.t.} & \frac{\partial \phi^t}{\partial \mathbf{q}} \mathbf{v} + \frac{1}{h}\phi^t \geq \mathbf{0} \end{cases}$$

end

$\mathbf{v}^{**} \leftarrow \mathbf{v}^* - \mathbf{M}^{-1}\mathbf{J}_c^\top \left(\mathbf{B}_\mathsf{N}\mathsf{r}_\mathsf{N}^* + \mathbf{B}_\mathsf{T}\mathsf{r}_\mathsf{T}^t\right)$ /* predictor velocity in Eq. (3.19) */

$\mathbf{p} \leftarrow -\mathbf{M}_e\mathbf{J}_c\mathbf{v}^{**}$ /* predictor impulse in Eq. (3.26) */

begin Solve friction problem to convergence:

$\mathsf{r}_\mathsf{N}^0 \leftarrow \mathsf{r}_\mathsf{N}^*$

for $k \leftarrow 1$ **to** k_{max} **do**

$\mathsf{r}_\mathsf{T}^k \leftarrow P_\mathsf{T}\left(\mathsf{r}_\mathsf{N}^{k-1}; \mathbf{p} - \mathbf{B}_\mathsf{N}\mathsf{r}_\mathsf{N}^{k-1}\right)$

$\mathsf{r}_\mathsf{N}^k \leftarrow P_\mathsf{N}\left(\mathbf{p} - \mathbf{B}_\mathsf{T}\mathsf{r}_\mathsf{T}^k\right)$

$\mathbf{r}_\mathsf{T}^k \leftarrow \mathbf{B}_\mathsf{T}\mathsf{r}_\mathsf{T}^k$

$err \leftarrow \frac{\|\mathbf{r}_\mathsf{T}^k - \mathbf{r}_\mathsf{T}^{k-1}\|^2_{\mathbf{M}_e^{-1}}}{\|\mathbf{r}_\mathsf{T}^{k-1}\|^2_{\mathbf{M}_e^{-1}}}$

if $err < \epsilon$ **then**

break

end

end

end

$\mathbf{r}^{t+h} \leftarrow \mathbf{B}_\mathsf{T}\mathsf{r}_\mathsf{T}^k + \mathbf{B}_\mathsf{N}\mathsf{r}_\mathsf{N}^k$

$\mathbf{v}^{t+h} \leftarrow \mathbf{v}^{**} + \mathbf{M}^{-1}\mathbf{J}_c^\top \mathbf{r}^{t+h}$ /* Eq. (3.20) */

3.6 Results

The following examples are generated using our Houdini [125] plugin written in C++ and Rust [100] and backed by the interior point optimization

package, Ipopt [141]. Rust provided us with additional confidence in our numerical results due to its strong memory and thread 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 neighbour lookup, we used an off-the-shelf R*-tree [10] with bulk loading. The tree is rebuilt for each iteration where the implicit surface samples are updated.

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 Argus cloth simulator [83] where we disable cloth remeshing to produce an identical cloth mesh between the two methods. Our simulator is publicly available under a permissive open source license 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 [111] and ParaView 5.7 [6] 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 in Eqs. (3.24) and (3.27) are solved to convergence with Ipopt using a residual tolerance of 10^{-10}. Elasticity and dynamics in Eq. (3.21) 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

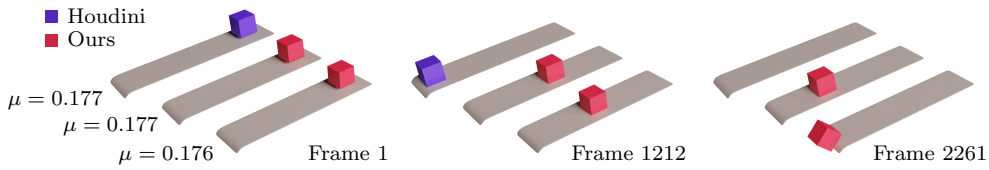
Example	Method	μ	# Elements	Time	$\frac{\text{Seconds}}{\text{Frame}}$	Stopped Sliding
Block Slide	Houdini	0.177	320	8:32	0.10	No
	Ours	0.177		3:21	0.04	Yes
	Ours	0.176		2:01	0.02	No
Cloth Slide	Argus	0.177	2	1:14	0.10	Yes
	Ours	0.177		3:21	0.11	Yes
	Ours	0.176		2:54	0.09	No
	Argus	0.177	722	1:55	0.15	No
	Ours	0.177		5:51	0.19	Yes
	Ours	0.176		5:21	0.17	No
	Argus	0.177	1322	6:52	0.55	No
	Ours	0.177		10:57	0.36	Yes
	Ours	0.176		11:21	0.37	No

Table 3.1: Timing results (m:ss) over the entire 5000 frames of the block slide and 750 frames of the cloth slide simulations. These results compare different simulators with comparable material properties. We set $\Delta t = 0.01$s for all simulations. The rightmost column indicates whether eventually stops sliding, which is expected only when $\mu > \tan(10°)$.

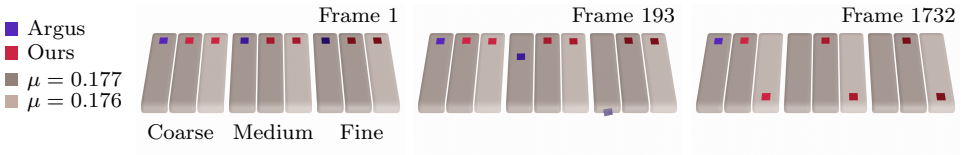
in Figure 3.4a. This example demonstrates the ability of our method of producing precise expected friction response. Our method is able to stop the cube from sliding off the ramp with a value of μ within 0.001 of the expected stopping coefficient $\mu = \tan(\theta\pi/180)$. Timing comparisons are presented in Table 3.1.

A similar experiment is performed with cloth to compare our method to Argus. 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 3.4b. In contrast, Argus we are able to produce consistent friction behaviour at all resolutions.

Cloth on a cylinder. In Figure 3.5 we drape a patch of cloth on a cylinder to demonstrate the contact fidelity of our method and compare it against



(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 Argus 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, Argus' simulation results in slipping at all resolutions albeit with significantly different accelerations. Our method (middle and right-most ramp) produces a consistent friction behaviour regardless of resolution as shown.

Figure 3.4: *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 $\mu = \tan(\pi/18) \approx 0.176326$.

Argus and Houdini's FEM cloth simulation. Figure 3.5a 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 improve the friction response with higher resolutions it ultimately fails to stick[3].

In Figure 3.5b we use a similar setup to compare against the non-smooth

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

friction solver from Argus. 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 Argus, although exact correspondence would require identical material models.

Figure 3.5c demonstrates how an implicit surface contact model allows us to produce consistent sliding behaviour irrespective of resolution. In contrast, polygonal contact models like Argus' or Houdini's can produce vastly different contact behaviours depending on mesh resolution.

We are able to demonstrate precise control over slipping behaviour by changing the friction coefficient μ in Figure 3.5d. Additionally in Figure 3.5e we show that adjusting the configuration of the cloth will cause it to slip when μ is held constant.

Performance numbers shown in Table 3.2 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 3.6 a spinning tennis ball with a hollow core (as shown in Figure 3.7) 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 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 behaviour 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).

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

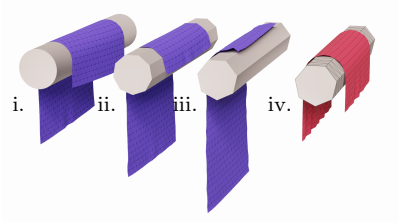
Example		# Elements	Method	μ	Time	$\frac{\text{Seconds}}{\text{Frame}}$
Sticking	i.	2K	Houdini	0.2	1:36	0.19
	ii.				1:26	0.17
	iii.				1:25	0.17
	iv.	2K	Ours	0.2	1:32	0.19
Reliability	i.	2K	Argus	0.12	3:12	0.38
	ii.				2:38	0.32
	iii.				2:45	0.33
	iv.	2K	Ours	0.12	0:09	0.09
Resolution independence	i.	392	Argus	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 3.2: Timing results (m:ss) over the length of entire corresponding simulations for the *Cloth on a cylinder* example from Figure 3.5.

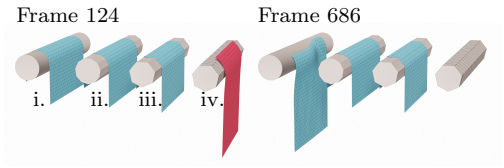
Figure 3.8. As the stool slides down the ramp, each leg of the stool oscillates between sticking and slipping friction modes at the points of contact with the ground. This occurs due to the elasticity of the stool, which periodically shifts the pressure between the front and hind legs of the stool. As with the other ramp slide examples, we use the ramp to produce the implicit field.

Belt drive. As shown in Figure 3.9, 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 behaviour, 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.

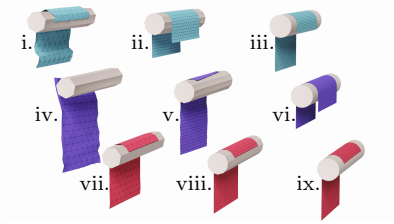
Glass pinch. Rigid body simulation is usually insufficient to solve various control problems like picking up a rigid object [15, 77]. We simulate a rigid whiskey glass being pinched and lifted between the index finger and the thumb of an animated soft hand model as shown in Figure 1.1. Our method produces reasonably 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 3.10. 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 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 also works.



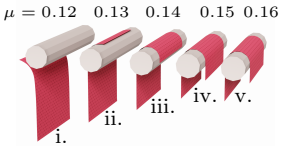
(a) *Sticking.* First three cylinders from the left show Houdini's cloth simulation at different discretizations. The last cylinder shows our method. Here we use $\mu = 0.2$, and the cloth is initially offset by 0.8 m vertically between the ends. With our method the cloth sticks, whereas with Houdini's it slips even at higher resolutions. The mesh must be sampled uniformly to produce a quality implicit surface, as shown by the uniform triangulation of the last cylinder. The contact surface is always smooth as shown in Figure 3.3.



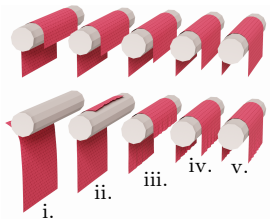
(b) *Reliability.* First three cylinders from the left show cloth simulation from Argus at different cylinder discretizations. The last cylinder demonstrates our cloth simulation at the coarsest sampling, which shows symmetric sliding without artifacts. This example uses $\mu = 0.12$ and $\Delta t = 0.02$, with the same configuration as in Figure 3.5a. 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.* Argus (top row) and Houdini's method (middle row) is compared with our method (bottom row) for different cylinder and cloth resolutions with $\mu = 0.13$. Here our contact model produces similar results regardless of resolution, while polygon based contact handling such as Houdini's or Argus' may produce vastly different behaviours.



(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. Initially, the draped cloth ends are offset vertically by 0.8 m. 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.



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

Figure 3.5: *Cloth on a cylinder.* A patch of cloth draped over a static cylinder 6 m in length with radius of 0.5 m. The cubic kernel was used to generate the contact field around the cylinder. Here, $\Delta t = 0.01$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.

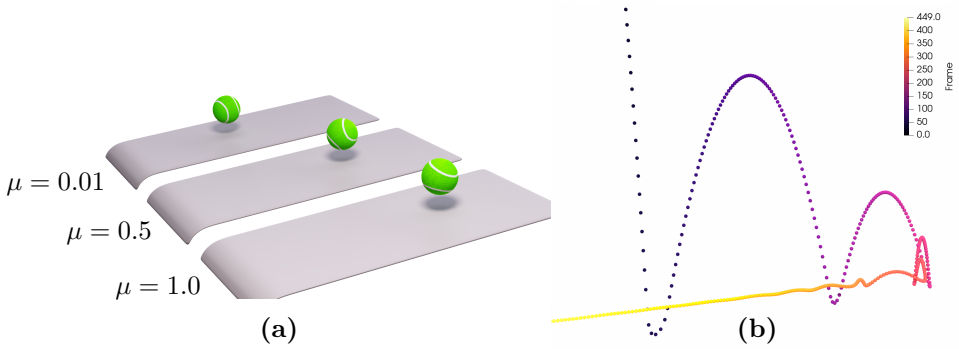


Figure 3.6: *Ball spin.* (a) Three identical hollowed balls with different coefficients of friction (as shown) are dropped on a ramp sloping down to the left. (b) The center-of-mass trajectory of the ball with the largest coefficient of friction ($\mu = 1$) is plotted. After the first two bounces up 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.

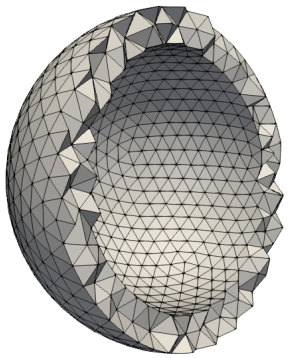


Figure 3.7: Hollow tennis ball mesh clipped to show the interior.

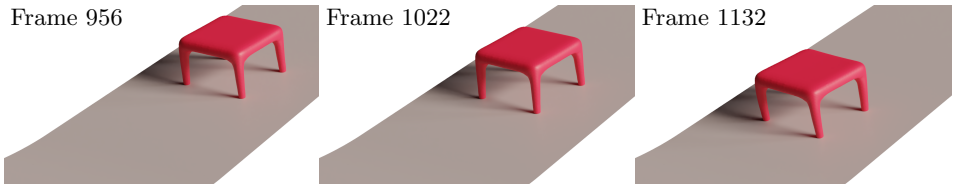


Figure 3.8: *Stick-slip chatter.* A stool slides down a ramp demonstrating stick-slip chatter of the stool legs.

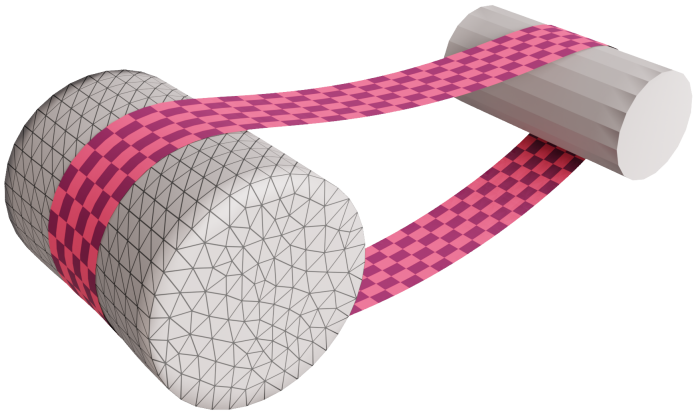
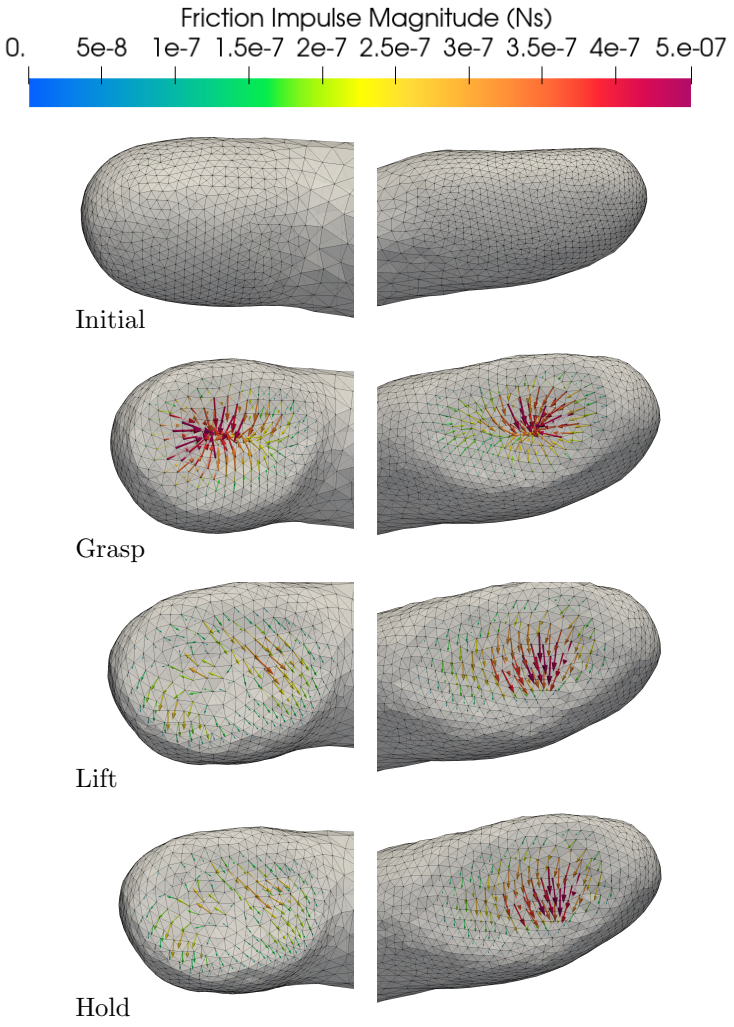
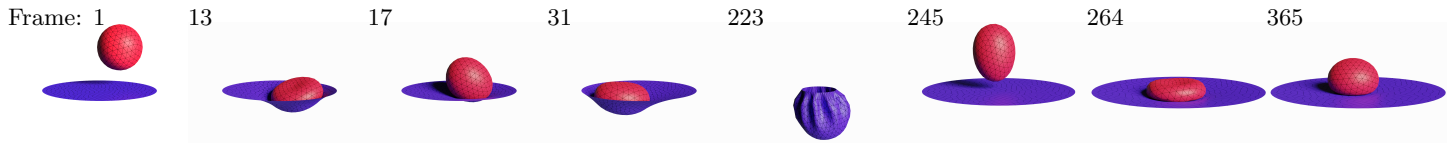


Figure 3.9: *Belt drive.* The small driver cylinder is rotated at an accelerated rate, and abruptly stopped to show slipping and sticking for different values of μ.

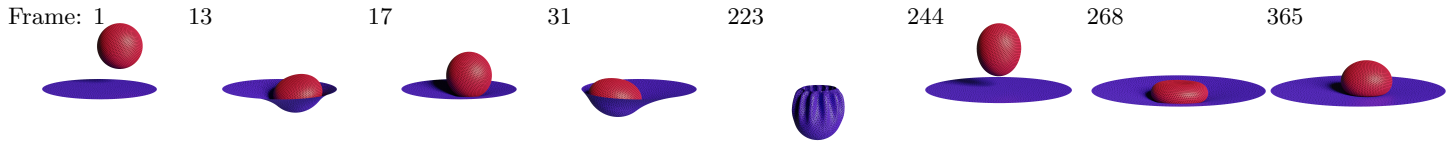


(a) Index finger. **(b)** Thumb.

Figure 3.10: The friction impulse (in N·s) vector field on the index finger and the thumb from the glass pinch simulation in Figure 1.1. 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 occurred. 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.



(a) Low resolution ball on a trampoline.



(b) High resolution ball on a trampoline.

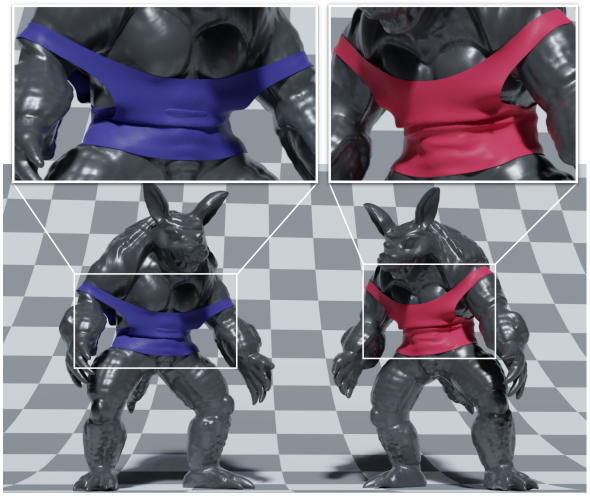
Figure 3.11: *Ball on a trampoline.* A hollowed ball is released onto a stretched trampoline in frames 1 to 150. The trampoline is then contracted and stretched even more to launch the ball into the air in the remaining frames. This example demonstrates reliable frictional coupling between the trampoline closely conforming to the ball. In this example we used $\mu = 0.2$, and the cubic kernel for generating a smooth implicit contact surface.

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 3.11. 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.

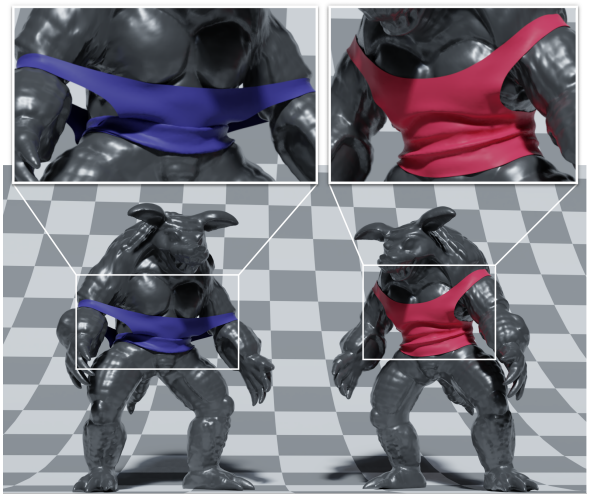
Armadillo tank top fitting. We fit a sleeveless shirt on an animated armadillo model to demonstrate an application of our method for simulated soft tissue cloth fitting. This application necessitates an accurate friction model to achieve a realistic fit.

We compare a frictionless fit against one with $\mu = 0.2$ against a *rigid* armadillo in Figure 3.12a. 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 3.12b. 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.



(a) Rigid fit.



(b) Soft fit.

Figure 3.12: *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).

Example	n_E	n_V	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			0:08:09	0.49
						0.8	2559	3/1.5	14/12.1	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 3.3: Timing results (h:mm:ss) over the length of entire corresponding simulations. The n_E column counts over triangles and tetrahedra with at least one free vertex. The n_V column counts the total number of simulated vertices. n is the maximum number of contact constraints. k_{tmax} and k_{tavg} are the maximum and average numbers of friction steps taken per timestep respectively where the friction solve converged. i_{kmax} and i_{kavg} are the maximum and average numbers of Ipopt iterations respectively taken per friction projection step Eq. (3.27). 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.

3.7 Discussion and conclusions

3.7.1 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 neighbourhoods $\mathcal{N}_i$ for each contact i since a mesh vertex should not collide against the surface generated by its neighbours. However this restriction can be more complicated in creases where colliding surfaces may be in the same neighbourhood. 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 [39]. 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 [83, 97].

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 [39, 70, 96]. However, non-convergence did not seem to have any noticeable effect on contact fidelity in our examples.

3.7.2 Discussion

Friction forwarding The choice of time splitting in our approach is critical to producing robust frictional responses. Figure 3.13 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 3.14 shows the artifacts produced when friction forwarding is disabled in the *glass pinch* example, by removing r^t from Eqs. (3.18) and (3.20).

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.

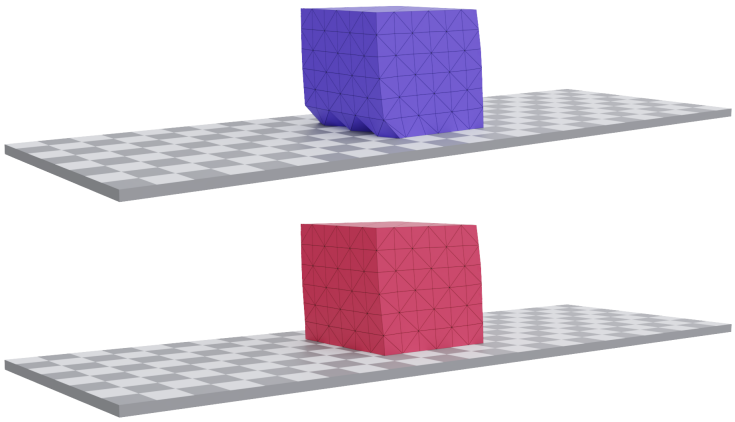


Figure 3.13: An FEM block of tetrahedra is dragged from right to left from the top vertices. The block slides along a horizontal plane while slightly compressed. The top frame shows the simulation without friction forwarding (i.e. $\mathsf{r}^t = \mathsf{0}$ in Eqs. (3.18) and (3.20)). The bottom frame 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 Eq. (3.18). As a result, friction is felt by vertices not directly in contact with the surface within a single time step.

3.7.3 Conclusion

We have introduced a novel approach for solving frictional contact problems with smooth elastic solids. Our method uses optimization, 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. Our method is validated visually and by comparison with a solution used widely in the visual effects industry.

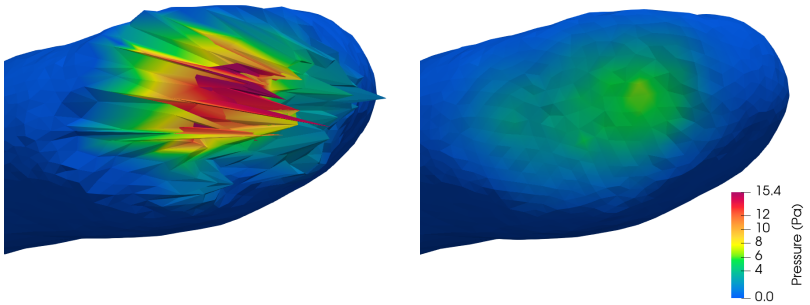


Figure 3.14: A closeup of the thumb shortly after the grasp phase of the glass pinch example from Figure 1.1. The left image shows the thumb surface *without* friction forwarding, while the right shows the same simulation *with* friction forwarding. Note that in both cases friction was solved to convergence.

Chapter 4

Nonlinear smooth dynamics

4.1 Introduction

Frictional contact problems do not lend themselves naturally to optimization formulations. For instance, in Chapter 3, we required iterating over 3 separate optimizations to arrive at the final result for each time step. Other popular methods [84] are forced to lag the friction computation to allow for an optimization-based solve.

In spite of this, energy minimization used for solving discretized ODEs over each time step has remained popular in graphics, due to its flexibility and robustness characteristics. Unfortunately, optimization based solvers used for dynamics equations require specialized algorithms for handling frictional contacts, which necessarily produces drawbacks in accuracy or robustness.

In this Chapter, we demonstrate the failure cases in popular optimization based frictional contact solvers and propose an alternative method for solving elastodynamic problems with frictional contacts that is simple to implement and accurate in comparison. By resolving the sliding frame and contact forces implicitly when computing friction, our method can produce more accurate friction behaviour, and it requires no additional iterations or specialized mechanisms for coupling friction, contact and elasticity. Furthermore, we show how approximate methods for handling friction such as

lagged friction [84] are insufficient for simulating accurate friction behaviour. In particular, we found that the lagged friction model produces inaccurate time-step dependent behaviour in scenarios near the stick-slip threshold, where stuck-together surfaces are close to slipping (e.g. tire rolling and handling delicate objects).

Frictional contact is traditionally modelled as a non-smooth problem requiring sophisticated tools. In particular, non-smooth integrators, root finding or optimization techniques are needed for handling inclusion terms in the mathematical model. This drastically complicates the problem and substantially limits the number of solution approaches. While non-smoothness is required to guarantee absolute sticking, it is not generally necessary if simulations are limited in time. In fact, when observed on a microscale, even dry friction responds continuously to changes in velocity [145]. Accordingly, we adopt a smooth friction formulation, and show that when applied correctly in dynamics equations it can produce predictable sticking. In contrast, popular friction models like lagged friction cause inaccurate and time-step dependent sticking behaviour.

To our knowledge this work is the first to demonstrate the importance of evaluating the sliding basis (defined in Section 4.3.2) and contact forces implicitly for accurate friction simulation.

Finally, to maintain smoothness of the entire problem we employ a penalty force for contact resolution and a smooth implicit surface model proposed in Chapter 3 (Larionov et al. [78]) for representing the contact surface. We also show how additional soft constraints can be added to the system for controlling the volume of an object.

An important consequence of having a smooth formulation, is that the entire simulation sequence can be differentiated with respect to the trajectory of the object. This property has become increasingly important in applications that aim to produce simulations that match real world data without the need for expensive higher-order methods or small time-steps. Differentiable forward models can be efficiently optimized over material properties, trajectories or initial conditions. Since forward simulation is inherently expensive, having gradient information is invaluable.

We include friction and contact as part of the equations of motion as proposed by Geilinger et al. [52]. This formulation lends itself to traditional analysis tools and methods for solving ODEs.

We further show how our system can be extended to handle other constraints by modelling volume preservation using a unified soft constraint introduced in Section 4.3.3.

In summary, we propose

- A simple fully implicit method for simulating hyperelastic objects producing realistic friction behaviour.
- Accurate high-order time integration applied to frictional contact problems for our fully implicit method as well as popular lagged friction formulations.
- An adaptive penalty stiffening strategy for effectively resolving interpenetrations with penalty-based contact methods.
- A physically-based volume change penalty for controlling compressibility in compressible and nearly incompressible regions.

Furthermore, to better characterize the instability of single point frictional contacts, we present an eigen-analysis of a 2D point contact subject to friction in Section 4.5.

4.2 Related work.

This section extends Section 3.2 with topics and works specific to this chapter.

Frictional Contact. Recently, lots of attention was brought towards modelling friction as a smoothly changing force at the stick-slip limit [52, 84]. This allows each simulation step to remain differentiable. Geilinger et al. [52] favoured a more traditional root-finding solver combining friction and contact forces with elastic equations of motion. In contrast, Li et al. [84] proposed a robust optimization framework to solve for contact and lagged

friction forces at every time step. Unfortunately, even with multiple iterations, the lagged friction approach does not converge to an accurate friction solution, which is especially noticeable in configurations near the stick-slip threshold. Here we demonstrate this shortcoming and propose a solution that favours friction accuracy at the cost of some robustness, while maintaining smoothness of the system.

Higher-order integrators for contact problems. Most contact formulations, especially those formulated in terms of constraints, intrinsically rely on a particular choice of time discretization, which is usually BE. However, the highly dissipative characteristics of BE have motivated the use of higher-order schemes like TR-BDF2 or SDIRK2, which preserve high energy dynamics while maintaining stability [13, 92]. A benefit of smooth contact models based on penalty or barrier functions is that both normal and friction forces are defined with explicit formulas, as opposed to implicitly defined through constraints. This makes it possible to apply higher-order integrators directly, as demonstrated by Geilinger et al. [52] for BDF2. Li et al. [84] applied TR, however it is only applied to non-contact forces, which causes stability issues. Brown et al. [26] focused on first-order methods and applied TR-BDF2 to a non-smooth optimization-based contact model with lagged friction, though this has unclear implications for high-accuracy second-order methods. In contrast to previous work, our approach is the first to incorporate contact, friction, elasticity and damping in a single fully implicit system evaluated using objects with deforming contact surfaces. We demonstrate the use of BDF2, TR-BDF2 and SDIRK2 applied to our formulation in Section 4.6.4.

Volume preservation. Many solids exhibit volume preserving behaviour. We focus primarily on inflated objects like tires, sports equipment (e.g. sports balls), as well as nearly incompressible objects like the human body. Inflated objects are typically simulated using soft constraints [20], where the volume change of an object is penalized. These methods are effective,

however, their physical accuracy is rarely questioned. Incompressible or nearly-incompressible materials are often modeled with stiff Poisson's ratios [128] or hard volume preservation constraints [123]. In contrast, we propose a unified physically-based penalty formulation for volume preservation that models both compressible and nearly-incompressible objects using a single penalty controlled by a physical compression coefficient.

4.3 Formulation

4.3.1 Contact

Traditionally contact constraints have been formulated with a positivity constraint (strict or not) on some "gap" function $\mathbf{d}(\mathbf{q})$ that roughly determines how far objects are away from each other. This function $\mathbf{d}$ may be closely related to a component-wise signed distance function, however, generally it merely needs to be continuous, monotonically increasing in the direction of separation, and constant at the surface. We define d_i for each potential contact point i such that $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ where n is the total number of potential contacts. Here we use the contact model of Chapter 3 (Larionov et al. [78]), where surface vertices of one object are constrained to have non-negative potential values when evaluated against a smooth implicit function $\mathbf{d}$ closely approximating the surface of another object. Interestingly, if we allow objects some small separation tolerance at equilibrium, we can reformulate this constraint as an *equality* constraint by using a soft-max [52] or a truncated log barrier [84]. These types of equality constraints greatly simplify the contact problem and have shown tremendous success in practice.

One disadvantage of log-barrier formulations is that the initial state must be free of collisions prior to the optimization step in order to avoid infinite energies and undefined derivatives. In the absence of thin features or risk of tunneling, it is sufficient to use a simple penalty function to resolve interpenetrating geometry. In this work we choose to use penalty-based contacts for simplicity, however, our formulation is fully compatible with a log-barrier method coupled with continuous collision detection (CCD) as proposed by

Li et al. [84]. The idea is to help our solver guide interpenetrating meshes out of intersecting configurations. We define a cubic contact penalty by

$$b(x;\delta,\kappa) = \kappa \begin{cases} -(x-\delta)^3 & \text{if } x < \delta \\ 0 & \text{otherwise,} \end{cases}$$

where $\delta > 0$ is the thickness tolerance and $\kappa > 0$ is a contact stiffness parameter that will need to be automatically increased to ensure that no surface vertices of one object are penetrating the implicit surface of another at the end of the time step. Here b corresponds to the first non-zero term in the Taylor expansion of the truncated log-barrier used by Li et al. [84], but unlike the log-barrier it is well-defined also for negative arguments. The penalty is applied to each contact point giving us an aggregate contact energy

$$W_c(\mathbf{d}) = \sum_{i=1}^{n} b(d_i;\delta,\kappa).$$

Now the contact force can be written simply as the negative energy derivative

$$\mathbf{f}_c(\mathbf{q})^\top = -\frac{\partial W_c}{\partial \mathbf{q}} = \boldsymbol{\lambda}(\mathbf{q})^\top \frac{\partial \mathbf{d}}{\partial \mathbf{q}}, \qquad \text{where} \qquad \boldsymbol{\lambda}(\mathbf{q})^\top = -\frac{\partial W_c}{\partial \mathbf{d}} \tag{4.1}$$

is the stacked vector of contact force magnitudes. In effect our contact formulation enforces the equality constraint $W_c(\mathbf{d}(\mathbf{q})) = 0$. Note that for $\boldsymbol{\lambda}$ to represent a true force magnitude, $\partial\mathbf{d}/\partial\mathbf{q}$ must be normalized if it does not coincide with a distance field Jacobian. In the interest of compute performance and simplicity, we leave $\partial\mathbf{d}/\partial\mathbf{q}$ as is, although the error is reduced with denser surface sampling.

4.3.2 Friction

We define the *contact Jacobian* $\mathbf{J}_c(\mathbf{q})$ and tangent basis $\mathbf{B}(\mathbf{q})$ over all potential contact points. Then $\mathbf{T}(\mathbf{q}) = \mathbf{J}_c(\mathbf{q})^\top \mathbf{B}(\mathbf{q})$ is the $m \times n$ matrix defining the *sliding basis* [84]. In short, this matrix maps forces in contact space to generalized forces in configuration space.

We can now derive the smoothed friction force [52, 84] from first principles. For each contact i, MDP postulates that friction force ought to maximally oppose relative velocity

$$\mathbf{f}_{f,i}(\mathbf{v};\mu) = \underset{\|\mathbf{y}\| \leq \mu\lambda_i}{\operatorname{argmax}}(-\bar{\mathbf{v}}_i^\top \mathbf{y}), \tag{4.2}$$

where μ is the coefficient of friction, which limits the friction force[1] and $\bar{\mathbf{v}}_i \in \mathbb{R}^2$ is the relative tangential velocity at contact point i. The contact force magnitude λ_i is the ith element of $\boldsymbol{\lambda}$ as defined in Eq. (4.1). We can solve Eq. (4.2) explicitly with an inclusion

$$\mathbf{f}_{f,i}(\mathbf{v};\mu) \in -\mu\lambda_i \begin{cases} \{\bar{\mathbf{v}}_i/\|\bar{\mathbf{v}}_i\|\} & \text{if } \|\bar{\mathbf{v}}_i\| > 0 \\ \{\mathbf{u} \in \mathbb{R}^2 \ : \ \|\mathbf{u}\| \leq 1\} & \text{otherwise.} \end{cases} \tag{4.3}$$

This is commonly referred to as Coulomb friction. Unfortunately, the non-smoothness around $\|\bar{\mathbf{v}}_i\| = 0$ calls for non-smooth optimization or root-finding techniques [21, 48, 69], making this problem numerically challenging. Another disadvantage of non-smoothness is that it greatly complicates differentiation of the solver, which can be critical for solving inverse problems efficiently. We opt to approximate Coulomb friction using a smoothed model [52, 84]. Since most animations call for relatively short time frames, we typically do not require absolute sticking. Interestingly, smooth friction has been proposed in older engineering literature [14, 75, 145] to model hysteretic behaviour and alleviate numerical difficulties. A simple smoothing of Eq. (4.3) can be written as

$$\mathbf{f}_{f,i}(\mathbf{v};\mu) \approx -\mu\lambda_i \boldsymbol{\eta}(\bar{\mathbf{v}}_i), \tag{4.4}$$

where $\boldsymbol{\eta} : \mathbb{R}^2 \to \mathbb{R}^2$ defines the per-contact nonlinearity

$$\boldsymbol{\eta}(\bar{\mathbf{v}}_i) = s(\|\bar{\mathbf{v}}_i\|) \begin{cases} \bar{\mathbf{v}}_i/\|\bar{\mathbf{v}}_i\| & \text{if } \|\bar{\mathbf{v}}_i\| > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

[1] Unless otherwise specified, $\|\cdot\|$ refers to the Euclidean norm.

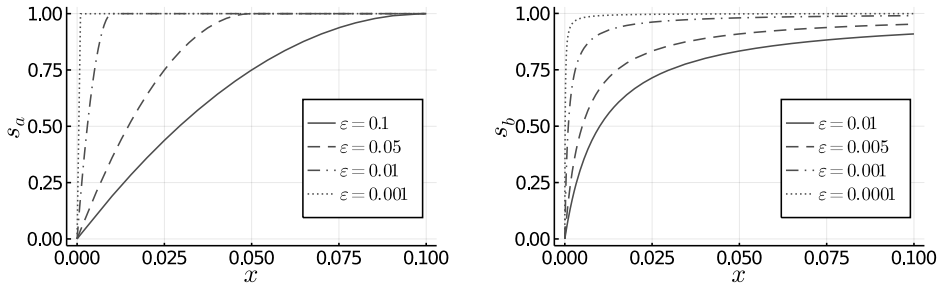


Figure 4.1: Examples of smoothing functions s_a (left) and s_b (right).

and the function s defines the pre-sliding transition. A suitable C^1 option for s is

$$s_a(x;\epsilon) = \begin{cases} \frac{2x}{\epsilon} - \frac{x^2}{\epsilon^2} & \text{if } x < \epsilon \\ 1 & \text{otherwise.} \end{cases} \tag{4.6}$$

For a C^∞ option, we can define

$$s_b(x;\epsilon) = \frac{x}{x+\epsilon}. \tag{4.7}$$

Figure 4.1 illustrates how s_a and s_b behave for different values of the stick-slip parameter ϵ. In our experiments, both functions produced reasonably accurate friction effects.

We can express the nonlinearity in Eq. (4.5) as a function over all (stacked) relative velocities $\bar{\mathbf{v}} \in \mathbb{R}^{2n}$ using a diagonal block matrix

$$\mathbf{H}(\bar{\mathbf{v}}) = \begin{bmatrix} \boldsymbol{\eta}(\bar{\mathbf{v}}_1) & & \\ & \ddots & \\ & & \boldsymbol{\eta}(\bar{\mathbf{v}}_n) \end{bmatrix}.$$

Then the total friction force can be written compactly as

$$\mathbf{f}_f(\mathbf{q}, \mathbf{v}) = -\mu \mathbf{T}(\mathbf{q}) \mathbf{H}(\mathbf{T}(\mathbf{q})^\top \mathbf{v}) \boldsymbol{\lambda}(\mathbf{q}), \tag{4.8}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^\top$ is the stacked vector of contact force magnitudes.

4.3.3 Volume change penalty

In soft tissue simulation, resistance to volume change is typically controlled by Poisson's ratio. This, however, assumes that the simulated body is homogeneous and void of internal structure. For more complex structures like the human body, a zonal constraint is a more suitable method to enforce incompressibility [123]. Compressible objects, however, require a different method altogether. In this section we propose a physically-based and stable model to represent compressible and nearly incompressible objects. In particular, we want to efficiently model inflatable objects like balloons, tires and sports balls, as well as nearly incompressible objects like the human body or other organic matter.

We start from the isothermal compression coefficient [99, Section 5.3] defined by

$$\kappa_v = -\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_T, \tag{4.9}$$

where V is the volume of interest[2], P is internal pressure and the T subscript indicates that temperature is held constant. For compressible continua like air in normal conditions, which behaves like an ideal gas, $\kappa_v = 1/P$. For nearly incompressible continua like water at room temperature, $\kappa_v \approx 4.6 \times 10^{-5}$ atm^{-1} is relatively constant. Assuming rest volume V_0 and initial pressure $P_0 = 1$ atm, we can derive the work W needed to change the volume of the container to V. For an ideal gas PV is constant, which gives

$$W_{ig}(V) = P_0\left(V - V_0\left(1 + \ln\frac{V}{V_0}\right)\right). \tag{4.10}$$

[2]For instance a region occupied by FEM elements or the volume of a watertight triangle mesh.

For a nearly incompressible continuum, κ_v is constant, which yields

$$W_{nif}(V) = \frac{1}{\kappa_v}\left(V_0 - V\left(1 - \ln\frac{V}{V_0}\right)\right). \tag{4.11}$$

For details of the derivation see Section 4.3.3.

Unfortunately, both models are undefined for negative volumes, which can easily lead to configurations with undefined penalty forces. Taking the second order approximation of Eq. (4.11) gives us

$$W_2(V) = \frac{(V - V_0)^2}{2V_0\kappa_v}, \tag{4.12}$$

which coincides with the second order approximation of Eq. (4.10) when $\kappa_v = 1$. Thus, our second-order model approximates both compressible and nearly incompressible continua well for small changes in volume as shown in Figure 4.2. For larger changes in volume, we recommend modeling Eq. (4.10) directly, since it also approximates Eq. (4.11) well and volume changes are not significant in nearly incompressible continua.

To alleviate the approximation error for scenarios that involve more compression (such as in Figure 4.10), we recommend decreasing κ_v to produce stronger restorative forces.

The penalty force is then given directly by the negative derivative of Eq. (4.12) and controlled by the compression parameter κ_v:

$$\mathbf{f}_v(\mathbf{q}) = -\frac{(V - V_0)}{V_0\kappa_v}\frac{\partial V}{\partial \mathbf{q}}. \tag{4.13}$$

This can then be added directly to Eq. (2.2). Incidentally, the Jacobian of Eq. (4.13) is dense, however, it can be approximated by the sparse term involving $\partial^2 V/\partial \mathbf{q}^2$, which expresses only local force changes. In matrix-free solvers where only matrix-vector products are required, the complete derivative can be computed without hindering performance since the full dense Jacobian is never stored in memory.

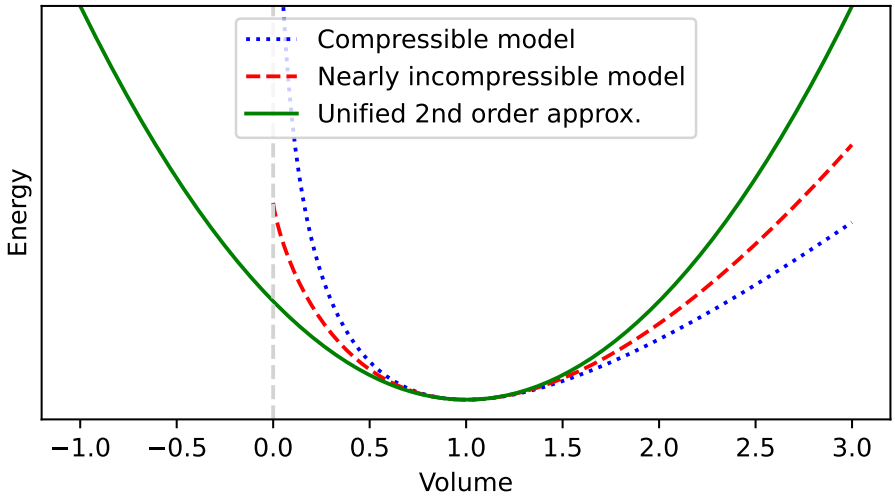


Figure 4.2: *Volume change energy.* The energy (negative of work) is plotted for the compressible model in Eq. (4.10) (dotted curve), the nearly incompressible model in Eq. (4.11) (dashed curve) and the 2nd order approximate model in Eq. (4.12) (solid curve). Here $V_0 = 1$ m^3, $\kappa_v = 1$ atm^{-1}, and $P_0 = 1$ atm. The quadratic model approximates both cases, but is ultimately too weak for excessive compression but too strong during large expansion. Depending on the use case, it may be necessary to model one of (4.10) or (4.11) directly.

Volume change penalty formulas

In this section we briefly describe how to derive the volume penalties for ideal gas and nearly incompressible continua given in Eqs. (4.10) and (4.11) respectively.

Assuming hydrostatic equilibrium, the mechanical potential of a body can be expressed as the work done on the system when being compressed or expanded from V to $V + dV$.

$$dW = -PdV,$$

where P is the pressure in the continuum. Since we simulate objects in an

atmosphere, the work done on a system is offset by the atmospheric pressure acting on the system. At the scale of our simulations, the atmospheric pressure is approximately constant, so the work that can affect the rest of the simulation is given by

$$dW = (-P + P_0)\, dV, \tag{4.14}$$

where $P_0 = 1$ atm is the atmospheric pressure.

Ideal gas. For an ideal gas, Boyle's law dictates that PV is constant, which implies that it can be computed from atmospheric pressure and rest volume $PV = P_0 V_0$. Thus, from Eq. (4.14), the energy of the system undergoing a change of volume from V_0 to V_1 can be written as

$$\begin{aligned} W(V_1) &= \int_{V_0}^{V_1} -P(V) + P_0\, dV = P_0 \int_{V_0}^{V_1} 1 - \frac{V_0}{V}\, dV \\ &= P_0 \left(V_1 - V_0 \left(1 + \ln \frac{V_1}{V_0} \right) \right). \end{aligned}$$

Nearly incompressible continua. Recall the compressibility coefficient κ_v defined by Eq. (4.9) for nearly incompressible continua. The pressure change of a bounded continuum caused by volume change from V_0 to V_1 can be expressed in terms of κ_v as

$$P(V_1) - P(V_0) = \int_{V_0}^{V_1} \frac{\partial P}{\partial V} dV = -\int_{V_0}^{V_1} \frac{dV}{\kappa_v V} = -\frac{1}{\kappa_v} \ln \frac{V_1}{V_0}.$$

With $P_0 = P(V_0)$, we can then derive the expression for energy directly as follows

$$\begin{aligned} W(V_1) &= \int_{V_0}^{V_1} -P(V) + P_0 \, dV = \int_{V_0}^{V_1} \frac{1}{\kappa_v} \ln \frac{V}{V_0} \, dV \\ &= \frac{1}{\kappa_v} \left(\int_{V_0}^{V_1} \ln V \, dV - (V_1 - V_0) \ln V_0 \right) \\ &= \frac{1}{\kappa_v} \left(V_0 - V_1 \left(1 - \ln \frac{V_1}{V_0} \right) \right). \end{aligned}$$

4.4 Numerical Methods

In this section we outline and motivate methods for solving the non-linear momentum balance system in Eq. (2.5).

4.4.1 Damped Newton

The momentum balance Eq. (2.5) can be solved efficiently by second-order root-finding methods like Newton. In the absence of constraints, this can be seen as a generalization of incremental potential optimization [66], when the merit function is set to be an energy potential[3] $W(\mathbf{v})$ such that $\partial W(\mathbf{v})/\partial \mathbf{v} = \mathbf{r}(\mathbf{v})$, albeit in that case to maintain a descent direction, $\partial \mathbf{r}/\partial \mathbf{v}$ must be appropriately modified to remain positive definite.

Since the presence of friction forces precludes a single potential W for minimization [40], many methods relying on incremental potentials build special workarounds to solve for exact Coulomb-based friction, including staggered-projections [69], fixed-point methods [48] and lagged-friction [84]. Others employ non-smooth Newton to find roots of a proxy function [21, 39, 70]. Our method is closest to the penalty-based frictional contact approach promoted by Geilinger et al. [52]. We extend this idea by using an implicit sliding basis, which makes our method fully implicit with respect to friction

[3] While the original incremental potential is intended to be optimized over positions, the velocity derivatives of all integrators we consider are a constant multiple of positional derivatives. Thus optimizing over velocity here is equivalent to optimizing over positions.

and contact. This unlocks the potential of using larger time steps to get results with good accuracy. For lower resolution examples we use the damped Newton algorithm as defined in Algorithm 2, where the problem Jacobian defined by $\mathbf{J} = \partial\mathbf{r}/\partial\mathbf{v}$ is a square non-symmetric matrix (see Section 4.5.1). Assuming that $\mathbf{J}$ is invertible in the neighbourhood of the root, for a sufficiently good initial estimate, damped Newton is guaranteed to converge[4] [109]. While singular Jacobians can cause problems, in our experiments they are rare, and often can be eliminated by decreasing the time step in dynamic simulations. Furthermore, in Section 4.5.1 we show that our method does not introduce singularities through coupling between elasticity, contact and friction on a single node.

ALGORITHM 2: DAMPEDNEWTON

Input:

$k_{\max} \leftarrow$ maximum number of Newton iterations
$\mathbf{v} \leftarrow$ previous velocities

Output: $\mathbf{v}_k \leftarrow$ velocity for the next time step

$\mathbf{v}_0 \leftarrow \mathbf{v}$ `/* Initialize velocity */`
for $k \leftarrow 0$ **to** k_{max} **do**
  if SHOULDSTOP$(\mathbf{r}(\mathbf{v}_k), \mathbf{v}_k)$ **then**
    break
  end
  $\mathbf{p}_k \leftarrow -\mathbf{J}(\mathbf{v}_k)^{-1}\mathbf{r}(\mathbf{v}_k)$ `/* Set search direction */`
  $\alpha \leftarrow$ LINESEARCH$(\mathbf{v}_k, \mathbf{p}_k)$
  $\mathbf{v}_{k+1} \leftarrow \mathbf{v}_k + \alpha\mathbf{p}_k$
end

4.4.2 Inexact damped Newton

For large scale problems, it is often preferable to use an iterative linear solver, which can outperform a direct solver when degrees of freedom are sufficiently abundant. We use *inexact Newton* to closely couple the iterative solver with our damped Newton's method.

Since friction forces produce a non-symmetric Jacobian, we chose the

[4] If $\mathbf{J}$ is also sufficiently regular then convergence is Q-quadratic.

biconjugate gradient stabilized (BiCGSTAB) algorithm [138] to find Newton search directions $\mathbf{p}_k$. For Jacobian $\mathbf{J}_k = \partial \mathbf{r}_k / \partial \mathbf{v}$ with $\mathbf{r}_k = \mathbf{r}(\mathbf{v}_k)$, the search direction is determined by

$$\|\mathbf{r}_k + \mathbf{J}_k \mathbf{p}_k\| \leq \sigma_k \|\mathbf{r}_k\|,$$

where $\sigma_k = \min(\|\mathbf{r}_k\|^{\varphi} / \|\mathbf{r}_{k-1}\|^{\varphi}, \sigma)$ and $\varphi = (1 + \sqrt{5})/2$ to maintain Q-quadratic convergence [44].

Using BiCGSTAB as the iterative solver additionally allows one to use forward automatic differentiation to efficiently compute products $\mathbf{Jp}$.

The final inexact Newton algorithm is presented in Algorithm 3.

ALGORITHM 3: InexactDampedNewton

Input:

$k_{\max} \leftarrow$ maximum number of Newton iterations
$\mathbf{v} \leftarrow$ previous velocities
$c_1 \leftarrow 10^{-4}$

Output: $\mathbf{v}_k \leftarrow$ velocity for the next time step

$\mathbf{v}_0 \leftarrow \mathbf{v}$ `/* Initialize velocity */`
for $k \leftarrow 0$ **to** k_{max} **do**
  if ShouldStop$(\mathbf{r}(\mathbf{v}_k), \mathbf{v}_k)$ **then**
    break
  end
  $\sigma_k \leftarrow \min(\|\mathbf{r}_k\|^{\varphi} / \|r_{k-1}\|^{\varphi}, \sigma)$
  Find $\mathbf{p}_k$ such that $\|\mathbf{r}_k + \mathbf{J}_k \mathbf{p}_k\| \leq \sigma_k \|\mathbf{r}_k\|$
  $\alpha \leftarrow 1$
  `/* Backtracking */`
  while $\|\mathbf{r}(\mathbf{v} + \alpha \mathbf{p})\| > (1 - c_1 \alpha (1 - \sigma_k)) \|\mathbf{r}(\mathbf{v}_k)\|$ **do**
    $\alpha \leftarrow \rho \alpha$
  end
  $\sigma_k \leftarrow 1 - \alpha(1 - \sigma_k)$
  $\mathbf{v}_{k+1} \leftarrow \mathbf{v}_k + \alpha \mathbf{p}_k$
end

For further details on the tradeoffs between inner convergence (of the linear solver) and outer convergence (of the nonlinear solver), which is managed by σ_k, as well as discussion about other Jacobian-free Newton-Krylov

(JFNK) techniques, we refer the reader to Kelley [72].

4.4.3 Contact

To ensure that no penetrations remain (i.e. $d_i > 0$ for each contact i) at the end of the time step we measure the deepest penetration depth $d_{\text{deepest}} = \min_i(\mathbf{d}_i(\mathbf{q}^{t+h}))$, and bump the contact stiffness parameter κ by the factor $\frac{b'(d_{\text{deepest}})}{b'(0.5\delta)}$ whenever $d_{\text{deepest}} < 0$, where b' is the scalar derivative of b. The same step is then repeated with the new κ. This scheme sets the optimal contact penalty value found by the Newton scheme to appear for contacts 0.5δ outside the contact surface. As such, in most cases one time step with an additional contact iteration is sufficient before all contacts are resolved. Furthermore, κ is never decreased so long as there are active contacts to avoid oscillations at the contact surface.

The downside of this technique is that it compromises the smoothness of the simulation. We postulate that in practice, this may not be problematic in a differentiable pipeline since κ is not changed frequently and subsequent differentiable iterations can carry forward the maximal κ to maintain smoothness.

4.4.4 Compatibility with lagged friction models

Li et al. [84] introduced a robust method to resolve contacts by minimizing incremental potentials with friction being evaluated using lagged positional estimates from the previous time step. Here we express their incremental potential contact (IPC) method as a nonlinear system, and propose a simple change that will establish more accurate frictional responses. The total force with lagged friction can be expressed as

$$\mathbf{f}_{lag}(\mathbf{q}^t, \mathbf{q}^{t+h}, \mathbf{v}^{t+h}) = \mathbf{f}_{edcg}(\mathbf{q}^{t+h}, \mathbf{v}^{t+h}) + \mathbf{f}_f(\mathbf{q}^t, \mathbf{v}^{t+h}), \tag{4.15}$$

where $\mathbf{f}_{edcg}$ is the sum of elastic, damping, contact and external forces and $\mathbf{f}_f$ is the friction force as before. This net force can then be integrated with an implicit-explicit (IMEX) style scheme. Specifically for BE and TR we

get

$$\mathbf{r}_{\text{BE,IPC}}(\mathbf{v}^{t+h}) = \mathbf{v}^{t+h} - \mathbf{v}^t - h\mathbf{M}^{-1}\mathbf{f}_{lag}(\mathbf{q}^t, \mathbf{q}^{t+h}, \mathbf{v}^{t+h}), \tag{4.16}$$

$$\begin{aligned}\mathbf{r}_{\text{TR,IPC}}(\mathbf{v}^{t+h}) &= \mathbf{v}^{t+h} - \mathbf{v}^t \\ &\quad -\frac{h}{2}\mathbf{M}^{-1}\left(\mathbf{f}_{lag}(\mathbf{q}^t, \mathbf{q}^{t+h}, \mathbf{v}^{t+h}) + \mathbf{f}_{lag}(\mathbf{q}^t, \mathbf{q}^t, \mathbf{v}^t)\right),\end{aligned} \tag{4.17}$$

respectively. This formulation has the advantage of having a well-defined antiderivative with respect to $\mathbf{v}^{t+h}$, which can be minimized using common optimization tools. In this view, IPC effectively solves Eq. (4.16) or Eq. (4.17) using the proposed log barrier potential as a merit function, CCD aided line search and Hessian projection. Although this can be done iteratively with better estimates for the lagged friction force, this approach has limitations as demonstrated in Section 4.6.1. Instead, we can establish more accurate friction in IPC, if we abandon the popular optimization view and instead solve the original momentum balance problem where friction forces are resolved implicitly (e.g. replace $\mathbf{f}_f(\mathbf{q}^t, \mathbf{v}^{t+h})$ with $\mathbf{f}_f(\mathbf{q}^{t+h}, \mathbf{v}^{t+h})$ for BE).

4.5 2D Analysis

To simplify our analysis and better illustrate the problem, consider a simple 2D example. A mass is attached by an idealized spring to the origin resting on a conveyor belt, which moves left at a constant velocity as illustrated in Figure 4.3. Using this example we analytically evaluate the conditioning of single point frictional contacts, and demonstrate potential issues that can arise in the formulation.

We evaluate our methods on the simple 2D example proposed in Section 4.1. We developed the following 2D simulations in the Julia programming language [22] using Jupyter notebooks. The 2D equations of motion (2.1) for this problem can be written as

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{K}(\mathbf{q})\mathbf{q} - \mathbf{D}(\mathbf{q})\mathbf{v} + \mu b'(q_y)\boldsymbol{\eta}(\mathbf{v} - \mathbf{v}_s) + \mathbf{f}_c(\mathbf{q}) + \mathbf{M}\mathbf{g},$$

where $\mathbf{q}$ and $\mathbf{v}$ are the 2D position and velocity of the free end respectively,

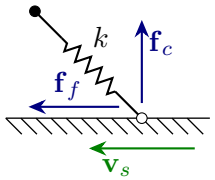


Figure 4.3: *Springy rod example in 2D.* A rod with spring constant k is fixed at the top (filled circle) and rests on the belt with the free end (hollow circle) experiencing a contact force $\mathbf{f}_c$. The belt moves to the left at a constant velocity $\mathbf{v}_s$ causing a friction force $\mathbf{f}_f$ on the free end of the rod.

$\mathbf{K} = -k(l_0 - \|\mathbf{q}\|)/\|\mathbf{q}\|$, l_0 is the rest length of the spring, $\mathbf{f}_c = (0, -b'(q_y))^\top$ and $\mathbf{g}$ is gravitational acceleration. The momentum balance for BE is then given by

$$
\begin{aligned}
\mathbf{r}_{\text{BE}} &= \mathbf{M}(\mathbf{v}^{t+h} - \mathbf{v}^t) && \text{(inertia)} \\
&\quad - h\left(\mathbf{K}(\mathbf{q}^{t+h})\mathbf{q}^{t+h} - \mathbf{D}(\mathbf{q}^{t+h})\mathbf{v}^{t+h}\right) && \text{(elasticity and damping)} \\
&\quad - h\left(\mu b'(q_y^{t+h})\boldsymbol{\eta}(\mathbf{v}^{t+h} - \mathbf{v}_s) + \mathbf{f}_c(\mathbf{q}^{t+h})\right) && \text{(friction and contact)} \\
&\quad - h\mathbf{M}\mathbf{g} && \text{(gravity)} \\
&\quad = \mathbf{0}, \\
&\quad \text{where} \;\; \mathbf{q}^{t+h} = \mathbf{q}^t + h\mathbf{v}^{t+h}
\end{aligned}
$$

In this case the gap function is the scalar $d(\mathbf{q}) = q_y$, and $\mathbf{T}$ is the projection onto the x-axis.

4.5.1 Jacobian conditioning

First we demonstrate the asymmetry of $\mathbf{J} = \partial\mathbf{r}/\partial\mathbf{v}$ for this problem to illustrate why we need solvers for non-symmetric linear systems. Here we will investigate backward Euler, however, the same findings should also apply for other integrators.

When the contact is not activated (i.e. $q_y \geq \delta$ in the 2D problem), both friction and contact forces are zero, reducing the problem to damped

elastodynamics. In this case it is well known that $\mathbf{J}$ is symmetric and often even positive definite, a thorough eigenanalysis of this case is explored by Smith et al. [128].

For the activated case (i.e. $q_y < \delta$), we will use Eq. (4.7) to analyze the 2D example. It models sticking, pre-sliding and sliding stages of friction with a single smooth function.

The derivative of Eq. (2.6) for our 2D problem is a 2×2 matrix

$$\mathbf{J}_{\text{BE}} = \mathbf{J}_{\text{BE}}^{\text{inactive}} + \begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix},$$

where $\mathbf{J}_{\text{BE}}^{\text{inactive}}$ identifies the part without friction or contact. When the contact is not active, we simply have $\mathbf{J}_{\text{BE}} = \mathbf{J}_{\text{BE}}^{\text{inactive}}$. For active contacts, we have

$$J_{xx} = -J_{yy}\frac{\mu\epsilon(q_y - \delta + 1)}{2h(\epsilon + |v_s|)^2}, \qquad J_{xy} = J_{yy}\frac{\mu v_s}{\epsilon + |v_s|},$$
$$J_{yx} = 0, \qquad J_{yy} = -\frac{6h^2\kappa}{m\delta}(q_y - \delta + 1),$$

assuming that the velocity at the free end is zero. Here m is the mass of the free end. Other quantities are as defined in previous sections. These expressions demonstrate a couple of interesting properties of frictional contact:

- Asymmetry of the Jacobian is due only to friction. Frictionless contact ($\mu = 0$) produces a symmetric system. Optimization based methods [78, 84] work around this asymmetry using time splitting or explicit integration, where friction is solved separately from the main elasticity equations. This is problematic because friction, contact and elasticity produce comparable impulses for large time steps, especially for large elastic moduli.

- When v_s vanishes (i.e. sticking), J_{xx} becomes very sensitive to the stick-slip parameter ϵ, which suggests potentially poor conditioning.

We can symbolically compute the singular values of $\mathbf{J}_{\text{BE}}$ to evaluate how the conditioning of this matrix depends on h, κ, δ, and ϵ parameters. As

mentioned before we are especially interested in the case where $v_s = 0$, the sticking case, where conditioning is poor. The following properties can be computed symbolically:

$$\lim_{\kappa \to \infty} \text{cond}(\mathbf{J}_{\text{BE}}) = \max\left(\frac{2h\epsilon}{\delta\mu}, \frac{\delta\mu}{2h\epsilon}\right) \tag{4.18}$$

$$\text{cond}(\mathbf{J}_{\text{BE}}) \in O(\min(\tfrac{1}{\delta}, \kappa)) \qquad \text{as } \delta \to 0 \text{ and } \kappa \to \infty \tag{4.19}$$

$$\text{cond}(\mathbf{J}_{\text{BE}}) \in O(\tfrac{1}{\epsilon}) \qquad \text{as } \epsilon \to 0 \tag{4.20}$$

In consequence, we know that:

1. For arbitrarily large κ, the conditioning of $\mathbf{J}_{\text{BE}}$ is bounded above by a function of δ by Eq. (4.18).

2. Conditioning will not degrade further when κ is increased if $\kappa > 1/\delta$ by Eq. (4.19).

3. $\text{cond}(\mathbf{J}_{\text{BE}})$ can grow linearly with $1/\epsilon$ by Eq. (4.20).

This analysis further expands on the tradeoffs introduced by the smoothed friction formulation. We can use these relationships to pick appropriate values of ϵ and δ for a particular application.

4.5.2 Stability

Our 2D problem can exhibit instabilities, two of which we will address here.

We can rewrite the balance Eq. (2.1) with $\mathbf{v} = \dot{\mathbf{q}}$ as a single system where $\mathbf{u} = (\mathbf{q}, \mathbf{v})^\top$ and

$$\dot{\mathbf{u}} = \mathbf{F}(\mathbf{u}) = \begin{bmatrix} \mathbf{v} \\ \mathbf{f}(\mathbf{q}, \mathbf{v}) \end{bmatrix}.$$

Although $\mathbf{F}$ is non-linear, we can evaluate the local stability of the system by analyzing the eigenvalues of $\partial\mathbf{F}/\partial\mathbf{u}$ [12, Ch. 2]. The presence of positive eigenvalues suggests local areas of instability.

To start, let's consider the case where $\mathbf{q} = (0, -1)^\top$, meaning the rod is compressed and vertical, $\mathbf{v} = (0, 0)^\top$, and the conveyor belt is still, $v_s = 0$.

This scenario demonstrates an unstable equilibrium or bifurcation, which is bound to generate a positive eigenvalue. We further assume no damping (i.e. $\alpha = \beta = 0$) to simplify analysis. The four eigenvalues of $\partial \mathbf{F}/\partial \mathbf{u}$ in this scenario are

$$\left(\frac{-3\delta\kappa\mu \pm \sqrt{(4\sqrt{2}-4)km\epsilon^2 + 9\delta^2\kappa^2\mu^2}}{2m\epsilon}, \pm\frac{i\sqrt{k+6\kappa}}{\sqrt{m}} \right). \tag{4.21}$$

Here we see that the right pair of eigenvalues are purely imaginary, while on the left, there is a single positive eigenvalue which vanishes as $\epsilon \to 0$. In this case smaller values of ϵ will improve the stability of the problem but will not completely stabilize it.

This type of scenario is not particularly special to our formulation but it demonstrates that our friction formulation will not completely eliminate this type of instability.

Now suppose that $\mathbf{q} = (-1, -1+\delta)^\top$, meaning the rod is on the left side and δ away from the contact surface. As before, $\mathbf{v} = (0,0)^\top$ and $v_s = 0$. In this case the contact is transitioning between active and inactive states. Assuming the contact is still active, we can compute the eigenvalues of $\partial \mathbf{F}/\partial \mathbf{u}$:

$$\left(\pm\sqrt{\frac{k}{m}\left(\sqrt{\frac{2}{(\delta-1)^2+1}} - 1\right)}, \pm i\sqrt{\frac{k}{m}} \right). \tag{4.22}$$

Here we notice that while the right pair of eigenvalues are purely imaginary as before, one of the left eigenvalues is positive for all $0 < \delta < 2$. This is a case of instability that is uniquely characteristic of the smoothed friction formulation.

Configuring the rod spring to start in the starting configuration giving eigenvalues (4.22) such that the free end repeatedly goes in and out of contact, we plot the eigenvalues of $\partial \mathbf{F}/\partial \mathbf{u}$ against the shaded unstable region for BE in Figure 4.4 with h = 4e-5 s. Increasing the time step, will reduce the number of eigenvalues in the unstable region for BE as errors are

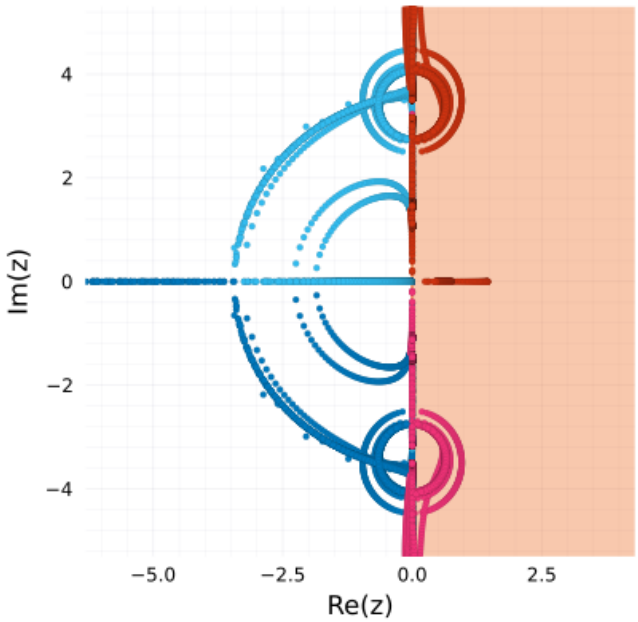


Figure 4.4: *Friction instability.* Eigenvalues of $\partial \mathbf{F}/\partial \mathbf{u}$ are plotted on the complex plane against the stability region of BE with $h =$ 4e-5 s. Each frame of the simulation corresponds to 4 eigenvalues, plotted here in 4 different colors, although some eigenvalues lie well outside the plotted region. Here $k = 10$ N/m, $m = 1$ kg, $\mu = 1.2$, $\kappa = 100$ N/m, $\delta = 0.4$ m and $\epsilon = 0.01$ m/s so that the free end of the rod repeatedly goes in and out of contact. This transition causes eigenvalues to fall into the unstable region, however it does not destabilize the overall simulation.

damped and the stability region expands, however, for non-L-stable integrators like TR, the errors are not damped with time step increase and so more eigenvalues remain in the unstable region throughout the simulation.

In the next instance we disable damping and set $v_s = -0.5$ m/s, $k = 15$ N/m, $m = 1$ kg, $\mu = 0.2$, $\kappa = 10^3$ N/m, $\delta = 0.01$ m, and $\epsilon = 0.01$ m/s to demonstrate how the 2D elastic rod behaves when subject to smooth frictional contacts. In Figure 4.5 we plot the kinetic energy of the system as h is decreased to reveal the high frequency oscillations that characterize smooth frictional contact. Consider the latter part of the animation in Figure 4.5,

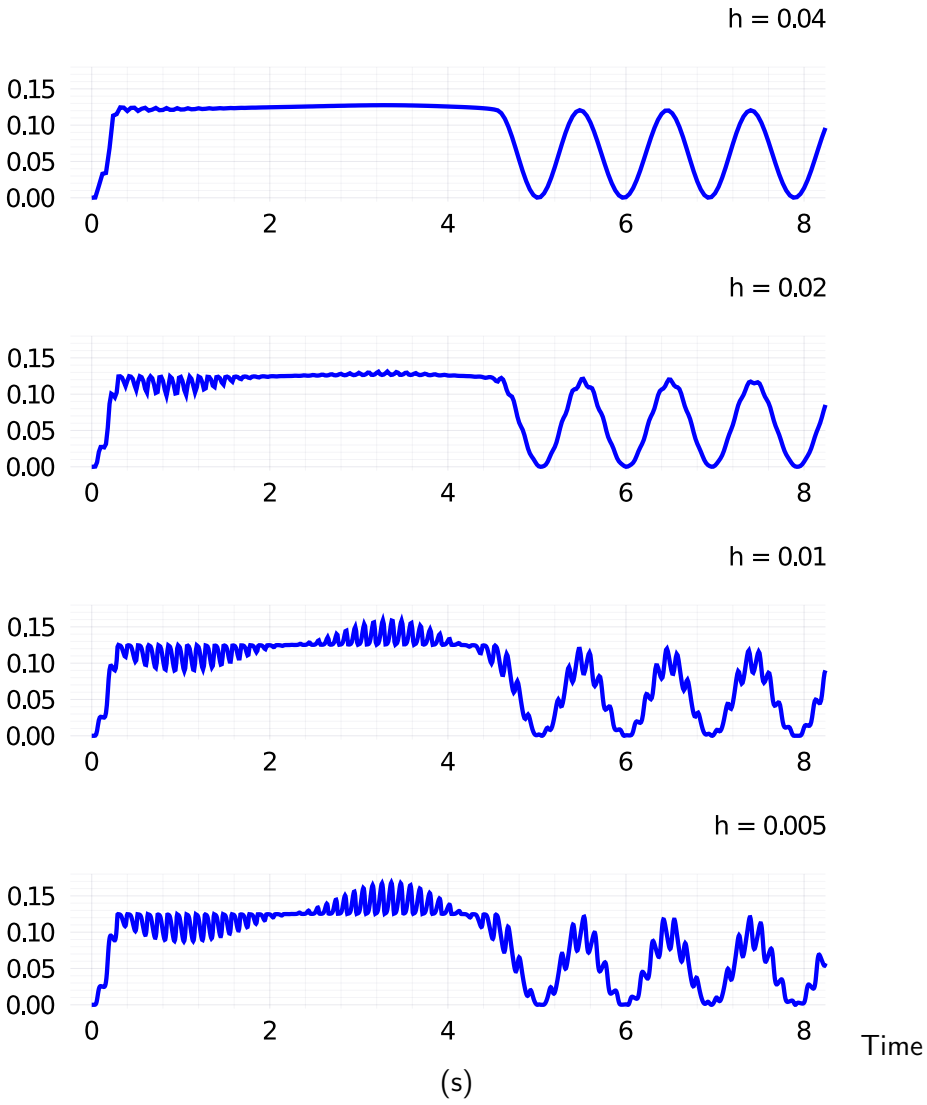


Figure 4.5: Kinetic energy of the 2D rod example is plotted for different time steps. High frequency oscillations emerge as time step h is decreased. We used TR-BDF2 here, although other converging integrators exhibit similar behaviour. TR will exhibit the high frequency oscillations even for $h = 0.04$ s, while BE requires time steps as low as $h =$ 1e-4 s before these artifacts become visible.

when the rod is dragged. As the conveyor belt pulls on the rod, the elastic force pulls the mass up, thus decreasing the normal force experienced. As a result, the friction force is decreased and eventually the rod accelerates in the positive direction. This causes the "large" oscillations in Figure 4.5. Now because our contact is enabled by a smooth but steep potential, the rod is effectively bouncing up and down at a very high frequency, which causes high frequency changes to the friction force. This causes the "small" oscillations.

4.6 Results

All examples were run on the AMD Ryzen Threadripper 1920X CPU with 12 cores, 24 threads at 3.7 GHz boost clock and 32 GB RAM. We used Blender 2.9 [112] and ParaView 5.7 [6] for all generated images and videos. For Algorithm 2 we used the Intel MKL sparse LU solver to solve the square non-symmetric linear system on line 6. In the following results Algorithms 2 and 3 are dubbed "Direct" and "Iterative", respectively, since the former uses a direct linear solver and the latter uses an iterative linear solver.

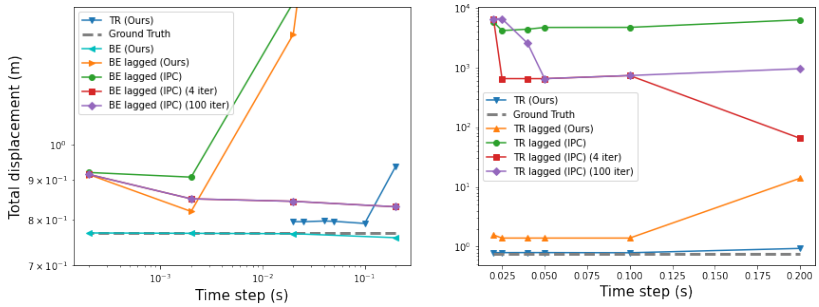
4.6.1 Friction accuracy

With the following examples we demonstrate two scenarios where lagged friction causes large deviations from an expected accurate and stable friction response.

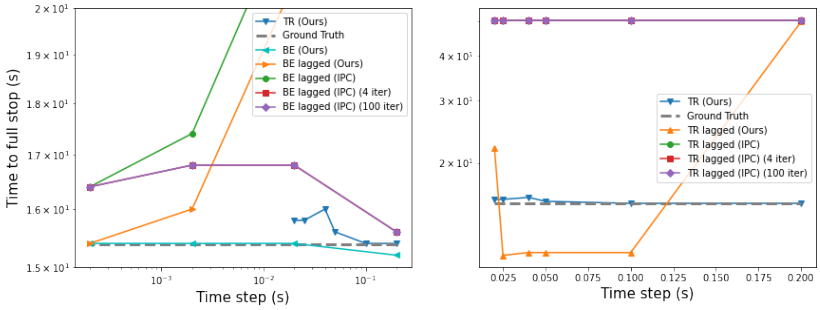
Block slide

In this example we let a stiff elastic block slide down a 10 degree slope expecting it to stop for $\mu = 0.177 > \tan(10^\circ)$ after sliding for a total of $x_T = 0.769$ m for $T = 15.38$ seconds (see supplemental document for details on the experiment).

In Figure 4.6 we demonstrate that our method produces consistent stopping across a variety of time step sizes using BE and TR time integration. We compare against a state-of-the-art smoothed friction method [84] using a lagged friction approach to show that it fails to establish consistent stopping with BE, and fails to stop with TR altogether after 50 seconds. We reproduce the lagged friction method in our simulator to demonstrate that TR can be used to generate reliable stopping if the equations of motion are correctly integrated as in Eq. (4.17). Our method produces a more accurate stopping distance using TR than IPC does using BE even after using multiple fixed point iterations.



(a) Total displacement travelled by the block before coming to a stop. With BE and multiple fixed point iterations the lagged friction approach can produce a reasonable approximation of the stopping behaviour, however, the approximation does not converge to the true solution. Our method produces a more accurate estimate of the true behaviour with BE, while TR produces a better estimate than lagged friction at $h = 0.1$ s and lower. With lagged friction, the TR method presented in Eq. (4.17) produces a more accurate result than IPC since contact is handled together with other implicit forces.



(b) Time taken by each block to come to a stop. With BE, our method produces a more accurate approximation to the true stopping time for all but the largest time step. In contrast, lagged friction fails to converge to the true stopping time under multiple fixed point iterations. With TR, the approximation is not as accurate as with BE, although still closer to the true value than lagged friction using BE. With TR, lagged friction does not produce a reliable stopping time, even under refinement. Using TR as implemented by IPC, the box does not stop after 50 s for any time step.

Figure 4.6: *Block slide.* Comparisons of analytic stopping conditions of a sliding block to numerical results.

Bowl grasp

Control over the friction coefficient is particularly important in grasping scenarios since grasped objects are often delicate. This means that friction forces involved in lifting are often close to the sliding threshold.

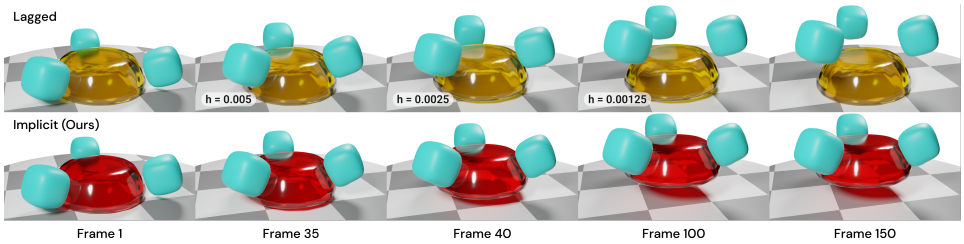
As shown in Figure 4.7, an upside down bowl is lifted using 3 soft pads to compare sticking stability of *lagged* friction proposed in Eq. (4.16) against a fully *implicit* method from Eq. (2.6). The bowl is successfully picked up and stuck to the pads for a range of time steps when using the *implicit* method, however it slips for different time step values with *lagged* friction. The height of the bowl is plotted in Figure 4.7b for each method and time step combination.

Ball in a box

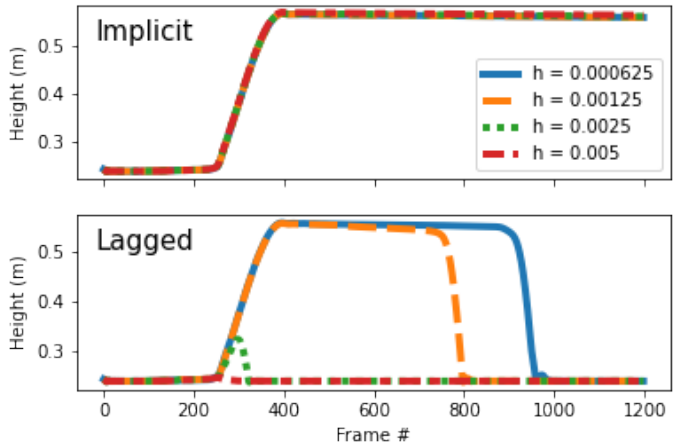
A rubber ball placed inside an elastic box has an initial spin of 4800 rotations per minute and an initial velocity set to $\vec{v}_0 = (-0.923, -0.385, 0)$. In Figure 4.8, we demonstrate how our fully coupled TR integrator produces more stable dynamic simulations when compared to the decoupled TR as proposed by Li et al. [84]. This scenario is simulated with both methods for 800 frames at $h = 0.01$ s with comparable damping parameters. The TR implementation used by IPC blows up, whereas in our formulation the energy is eventually dissipated as expected. In Figure 4.8b, we simulate the ball and box separately with each method using the same damping parameter as in the final simulation from Figure 4.8a. This shows that damping is comparable with both methods, which means that the reason for blowup in IPC is not due to integration of elasticity equations alone, but indeed due to the loose coupling between elastic and contact terms.

4.6.2 Performance

In this section, we show how various combinations of volume preservation and frictional contact constraints can affect the performance of the simulation. In addition, we compare Algorithms 2 and 3 in performance and memory usage.

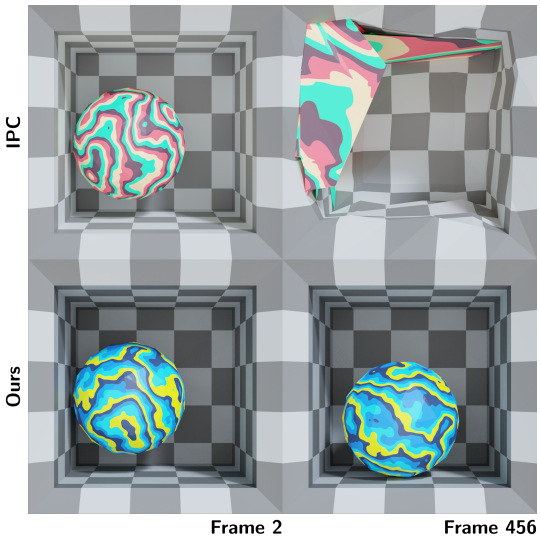


(a) The bowl is simulated using lagged friction from Eq. (4.16) (top row) and our fully implicit method (Eq. (2.6), bottom row). The chosen frames between 1 and 150, are frames 35, 40 and 100, at which point the bowl slips out for time steps $h = 0.005$ s, 0.0025 s, and 0.00125 s respectively for the lagged method. Using fully implicit integration, the bowl sticks even at the largest time step $h = 0.005$ s as shown.

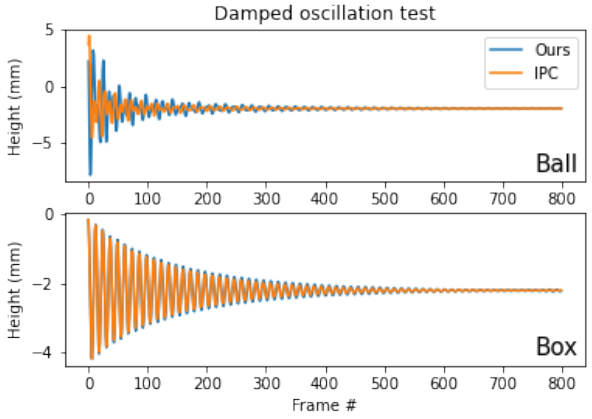


(b) The height of the bowl is plotted against frame number time step h is varied. Here the bowl slips for the lagged friction method, whereas the fully implicit method maintains stable sticking for every time step, hence all plotted lines overlap.

Figure 4.7: *Bowl grasp.* An upside down bowl is lifted up using 3 soft pads. Here $\rho = 1000$ kg/m^3, $E = 600$ KPa, and $\nu = 0.49$ for the pads and $\rho = 400$ kg/m^3, $E = 11000$ KPa and $\nu = 0.1$ for the bowl, with friction coefficient set to $\mu = 0.65$ between them.



(a) The TR formulation proposed in IPC is unstable (top row) whereas our method produces a dissipative scenario (bottom row) even with a slightly weaker damping as shown in Figure 4.8b.



(b) Damping ratio of 0.02 is selected for IPC, and damping parameter of 0.1 Hz is set for our model. In both cases Rayleigh damping is used. The ball (top) and the box (bottom) are simulated individually with both methods and centroid heights are plotted to ensure the that oscillation amplitudes with TR as implemented by IPC do not exceed those generated by our method.

Figure 4.8: *Ball in a box.* A spinning ball bounces inside an elastic box. Here $\rho = 10000$ kg/m^3, $E = 500$ KPa, and $\nu = 0.1$ for the box and $\rho = 500$ kg/m^3, $E = 50$ KPa and $\nu = 0.45$ for the ball, with friction coefficient set to $\mu = 0.1$ between them.

Tube cloth bend

The inexact Newton Algorithm 3 shines particularly in scenarios with numerous contacts, such as with tight-fitting garments, where the entire garment is in contact with a body. Here we simulate a simplified scenario of a tube cloth wrapped around a bending soft object resembling an elbow or knee as depicted in Figure 4.9. Table 4.1 shows the corresponding timing results, which indicate that inexact Newton performs much better than the damped Newton algorithm employing a direct solver. Furthermore, the performance gap becomes large when the number of elements is increased. Interestingly this data also indicates that larger friction coefficients cause a bigger bottleneck for the solve compared even to stiff volume change penalties (indicated by small κ_v).

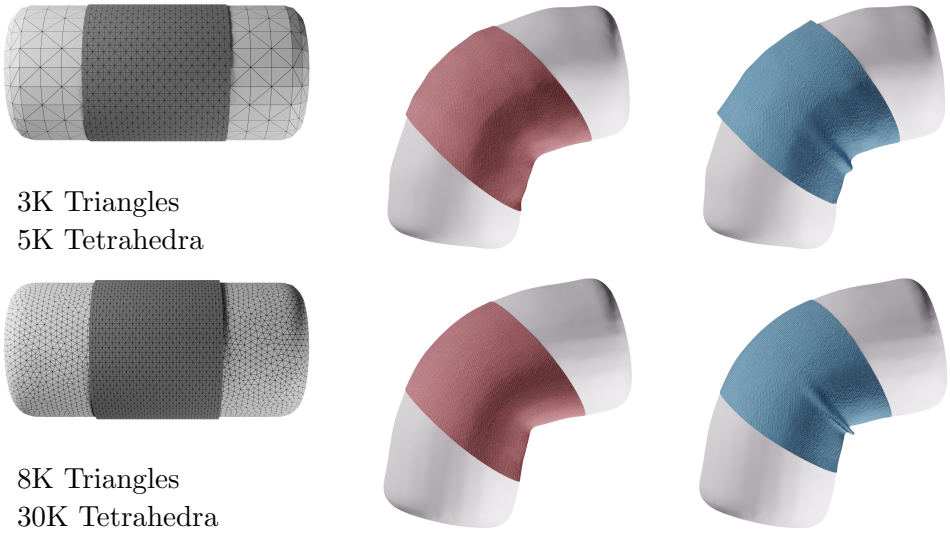


Figure 4.9: *Tube cloth bend.* A cylindrical garment is wrapped around a bending capsule. Lower resolution example is shown at the top row and higher resolution at the bottom row. The first column shows the initial configuration, second column shows the end result with low friction ($\mu = 0.2$), and the last column shows the end result with high friction ($\mu = 0.8$).

# Elements	μ	Solver Type	κ_v	Time	Memory	Volume Loss
3K Tris 5K Tets	0.2	Iterative	-	4.81	1.84 GB	0.962%
		Direct	-	22.1	6.51 GB	0.962%
	0.8	Iterative	-	12.7	1.91 GB	0.962%
		Direct	-	25.6	7.09 GB	0.962%
8K Tris 30K Tets	0.2	Iterative	-	8.59	830 MB	2.55%
		Iterative	4.6e-5	12.2	858 MB	2.44e-4%
	0.8	Iterative	-	51.6	625 MB	2.57%
		Iterative	4.6e-5	53.2	918 MB	2.44e-4%

Table 4.1: *Tube cloth bend performance data.* Time is measured in seconds per frame and memory refers to total memory growth during the simulation as reported by Houdini's performance monitor. Simulations without volume change penalty have κ_v set to $-$.

Ball squish

A hollow ball at various resolutions (1K, 55K and 160K elements) is pressed between two flat planes. As a result the ball experiences volume loss. To preserve some of the volume we simulate the compression with compression coefficients $\kappa_v = 1$ (e.g. a ball filled with air) and $\kappa_v = 0.01$ (e.g. a ball filled with water). In the latter case we expect significantly less volume loss, which is reflected in our experiments as shown in Table 4.2. Furthermore, we note from Table 4.2b that scenarios with small κ_v favour the "Iterative" method. In Table 4.2c we see that this is true whether $\mathbf{J}$ is sparsely approximated ("Inexact") or not ("Exact"). In contrast, stiffer scenarios prefer the "Direct" method due to worse system conditioning.

4.6.3 Real world phenomena

In the following example we show how our simulator can reproduce deformations captured in the real world.

# Elements	1K	55K	160K
Direct	0.354 (0.228)	15.9 (1.69)	61.0 (4.26)
Iterative	0.391 (0.081)	31.9 (0.719)	111 (1.92)
Volume loss	33.6%	27.8%	27.7%

(a) $E = 100$ KPa, $\kappa_v = 1$.

# Elements	1K	55K
Direct	1.08 (0.478)	89.1 (2.03)
Iterative	0.463 (0.214)	42.6 (0.884)
Volume loss	0.49%	0.36%

(b) $E = 100$ KPa, $\kappa_v = 0.01$.

$\kappa_v =$	1	0.01
Direct Exact **J**	0.832 (0.393)	0.871 (0.672)
Direct Inexact **J**	0.354 (0.228)	1.08 (0.478)
Iterative	0.391 (0.081)	0.463 (0.214)

(c) $E = 100$ KPa, for 1K elements.

# Elements	1K	55K	160K
Direct	0.312 (0.388)	8.28 (1.52)	25.5 (4.14)
Iterative	0.593 (0.097)	28.3 (0.732)	56.2 (2.62)
Volume loss	74.3%	74.6%	75.0%

(d) $E = 1000$ KPa with no volume preservation constraint.

Table 4.2: *Ball squish timings, memory usage and volume loss.* A hollow ball is squished between two flat rigid plates at time step $h = 0.001$ s. Timings are given in seconds per frame and averaged over 804 frames. The memory usage measured in GB over the entire simulation sequence is shown in parentheses. Volume loss is computed as the change in volume between frames 1 and 804 as a percentage of initial volume inside the ball. Each table specifies the number of elements, Young's modulus E and compression coefficient κ_v where applicable. Here we compare how the performance characteristics of our "Direct" and "Iterative" methods change when problem stiffness and κ_v is varied for different mesh resolutions.

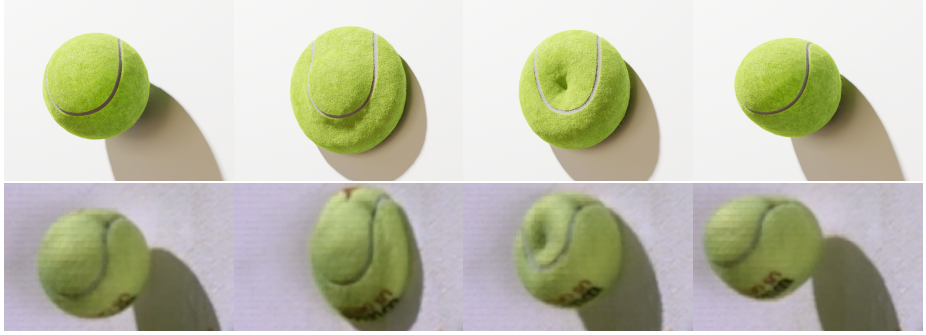


Figure 4.10: *Highspeed tennis ball collision.* Simulated ball (top row) is compared against a real ball video reference [11].

4.6.4 Tennis ball

Tennis ball dynamics is a prime example of all methods proposed in this chapter. We launch a tennis ball at a wall at 100 mph (44.704 m/s) to reproduce accurate slow motion deformation. Tennis balls are typically pressurized to approximately 1 atm above atmospheric pressure to increase longevity and to produce a livelier bounce during play. In Figure 4.10 we show how deformation changes when the ball is pressurized and compare the result with live footage. The simulation contains 100K tetrahedra and 92K vertices. The ball is hollow with a stiff inner core ($E = 6$ MPa, $\nu = 0.4995$ and $\rho = 934$ kg/m^3) and a light outer felt material ($E = 5.4$ MPa, $\nu = 0.3$ and $\rho = 4.69$ kg/m^3). The volume change penalty is applied to the interior of the ball. This 1000 frame simulation took 4.89 seconds per frame and a total of 11.33 GB in memory.

Next, a tennis ball is dropped from a 254 cm height to evaluate its bounce with and without pressurization. We show how pressurization and choice of integrator can drastically affect the height of the bounce in Figure 4.11.

4.6.5 Tire wrinkling

We model an inflated tire used by top fuel dragsters to show the folding phenomenon at the start of the race. The tires are deliberately inflated at a low pressure of 0.68 atm above atmospheric pressure, which allows them

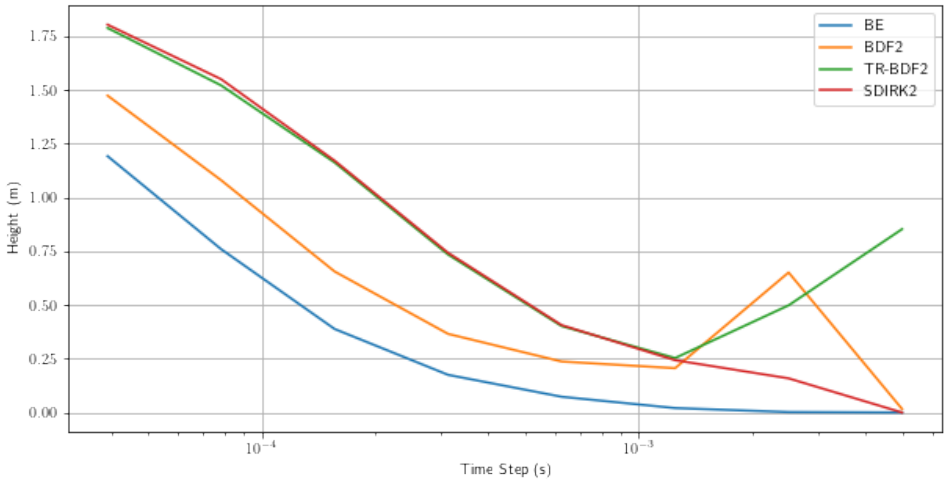


Figure 4.11: *Tennis ball drop.* A tennis ball dropped from a 254 cm height, is simulated using different integrators. Here, BE and BDF2 produce a much lower bounce than either TR-BDF2 or SDIRK2 methods across a range of different time steps. Higher-order integrators are defined in the supplemental document.

to better grip the asphalt for a better head start. As a result the soft tire tends to wrinkle as the wheels start to turn. This phenomenon allows for a larger contact patch between the tire and the ground for better traction, which translates to a larger acceleration. In Figure 4.12 we demonstrate this phenomenon in simulation with a shell model tire inflated using our volume change penalty. The outer side of the tire is initially stuck to the ground and then dragged while maintaining consistent contact. Accurate simulation of stick-slip transitions of the tire is critical in determining its performance since traction transfers torque into forward acceleration of the vehicle, which ultimately determines the outcome of a race. The tire is simulated using 18K triangles and 9K vertices. The tire mesh is split into a stiffer outer part that is in contact with the ground and a softer inner part where the label is printed. Here $\mu = 0.8$, and $\nu = 0.49$ everywhere, while $E = 400$ KN/m, $\rho = 200$ kg/m^2 and bending stiffness set at 0.1 on the outer part, and $E = 200$ KN/m, $\rho = 50$ kg/m^2, and bending stiffness set at 0.01

on the inner part. The simulation ran with $h = 0.00125$ s for 3.63 seconds per frame using the damped Newton solver.

Figure 4.12: *Dragster tire wrinkle.* A soft tire spins against the ground with a large friction coefficient, causing the rubber to wrinkle.

4.7 Conclusions and limitations

Smooth contact. An important detail of our formulation is that it requires a smooth contact surface representation in order to guarantee local convergence of Newton's method. This also implies that for each step the collision detection algorithm must capture all potential collisions within distance δ. Otherwise, a collision detected after it has become closer than δ away from the contact surface will cause a discontinuity in contact forces. This may occasionally cause oscillations in Newton's iterations if the root lies near the discontinuity.

Local minima. Solving for roots of nonlinear momentum equations allows one to resolve friction forces more accurately, however, this comes with a

trade-off. Optimization theory allows one to reliably find a descent direction even when the objective Hessian is indefinite via projection or filtering techniques. Although computing the descent direction for finding roots of nonlinear equations allows one to use the entire unfiltered Jacobian, global convergence can only be theoretically guaranteed when the Jacobian is bounded on the neighbourhood of the initial point. For stiff systems, this assumption can become problematic, although the practical implications are unclear.

Hydrostatic equilibrium. Our volume change penalty model expects hydrostatic equilibrium, which may not always be a good approximation given the rate of change of volume during a simulation. For quickly deforming objects like in our tennis ball and tire wrinkling example, some details of the deformation may be missing due to this approximation. This is because the object deforms faster than the air moves inside the volume, creating nonuniform pressure distribution throughout the volume. The comparison of our hydrostatic model to a fully dynamic fluid simulation remains as future work.

In conclusion, we presented a fully implicit method for simulating hyperelastic objects subject to frictional contacts. This method generalizes the popular optimization framework for simulating hyperelastics with contact and lagged friction potentials. We demonstrate how integrators like trapezoid rule or higher order integrators can be applied in our method as well as in IPC-style frameworks. Our method addresses the lack of friction convergence in lagged friction formulations by evaluating contacts, friction forces as well as tangential bases implicitly.

Furthermore, we propose a physically-based volume change penalty that can be used to simulate compressible as well as nearly incompressible solids in a single framework without additional complex constraint solvers.

Our system is entirely smooth, featuring unique gradients, which are ideal for differentiable simulation.

Chapter 5

Cloth parameter estimation

5.1 Introduction

Clothing plays an important part in human culture and self-expression. Traditional fashion houses are becoming more and more aware of the virtual landscape and are experimenting with virtual collections in the metaverse in addition to real-world fashion lines and virtual try-on. Virtual clothing items have reduced production costs and time to market. To reproduce familiar experiences and opportunities for self-expression in the virtual world, we need the ability to simulate a large variety of different types of clothing made from vastly different materials and structures. Unfortunately, creating realistic clothed human animations currently involves hours of tweaking simulation parameters and expert knowledge to obtain the desired clothing look.

In Chapter 3, we introduced a method for handling contact on smooth objects which allows for high fidelity sliding behaviour and permits differentiable contact constraint formulations. In Chapter 4, we further smoothed the friction curve making the whole simulation pipeline differentiable, and addressed the resulting accuracy limitations. This work produces a solid foundation for differentiable simulation of objects subject to high fidelity frictional contacts. However, the specific algorithms and material models we chose are not efficient enough for many inverse problems. Fortunately,

our chosen frictional contact formulation can, in theory, be solved by any algorithm for solving nonlinear systems, although the efficacy of specific algorithms remains to be demonstrated. In this chapter, we divert our attention towards an application of differentiable simulation to further motivate the direction of the preceding chapters, however we use a simpler and more efficient simulation algorithm. Here, we leave the extension to resolve frictional contacts as future work.

With this project, we aim to reduce the time and cost required to create realistic cloth simulations of real-world materials. While much of the effort in building simulators targets increasing numerical accuracy and minimizing computation time, we instead focus on improving the realism of an existing real-time cloth simulation method. To that end, we are the first to present a material estimation pipeline leveraging differentiable position-based simulation of compliant constrained dynamics [95] and easy to reproduce cloth captures. Capturing cloth samples is often difficult to control due to environmental, material shape memory, and hysteresis effects. Small perturbations can lead to significantly different cloth equilibrium states, which typically makes the optimization heavily biased towards the specific captured sample of the fabric. To overcome these limitations we introduce the following contributions:

- A novel robust objective function that operates in frequency space and is able to capture material-specific behaviour, independent of the current cloth wrinkle state.
- A simplified pipeline for estimating static cloth material properties that is decoupled from the real-world capture system through a template registration process.
- A real-world validation of XPBD cloth simulation with numerical comparisons of simulated results and captured scans.
- A set of material parameters for the standard StVK cloth material model for three distinct common materials.

5.2 Related work

Given the high relevance of virtual textile simulation both in research and manufacturing, material estimation has been a topic of investigation in the research community [91] and industry [27, 35, 113].

Material parameter estimation has been a focus point in computer graphics in recent years. While some techniques address the full scope of dynamic simulation parameters using video sequences [23], we focus instead on elasticity, which can be inferred from static captures of deformed cloth. Wang et al. [143] have proposed a non-standard piecewise material model and fit 39 parameters to a set of sparse correspondences. While sophisticated, their scheme can exhibit unlikely Poisson's ratios below zero or above one. Prior work focusing on mass-spring and diagonalized StVK models [103] produce promising parameter estimations, but at the cost of requiring a complex and expensive capture setup. Furthermore, their method omits the Poisson effect, which greatly limits the applicability of their results in modern cloth simulators. Both works employ nodal positions to estimate approximation error, which makes their methods sensitive to wrinkles in captured data. Clyde et al. [36] propose a more sophisticated non-linear material model to better fit existing standard ASTM measurements in large deformation scenarios. Some commercial hardware like Fabric Analyzer by Browzwear [27] or CLO Fabric Kit 2.0 [34] used for estimating fabric properties are often coupled with proprietary software to infer the correct parameters for the corresponding simulation software. Similar systems for estimating fabric parameters are also developed in academia, where additional cameras and custom clamps can allow for a wider range of materials [59]. In contrast, we focus on improving the optimization method to handle data coming from non-standard and potentially cheaper setups. A different line of work attempts to estimate cloth simulation parameters using neural networks from a static drape [65] or from videos [146]. In another line of work, the mismatch between simulation and observation is expressed using stochastic models to infer an estimate for simulation control parameters that can more closely resemble physical material properties [43, 122].

Differentiable simulation and system identification has been investigated in recent research [86, 119] to infer material parameters from observations [42, 62, 64, 130]. Another topic of focus has been bringing real objects into the virtual world using sparse observations and interaction of the object with its environment [31, 144].

Cloth simulation has been a topic of research in computer graphics for many decades. Starting from the earlier work [18], many improvements have been proposed to increase accuracy, stability and performance [24, 87, 95, 107, 116, 133].

In summary, the research field has made great progress while several issues still remain. Deeply intertwined pipelines of closely coupled capture and optimization systems complicate the combination of different capture and optimization techniques. Furthermore, multiple prior works use complex capture setups that are difficult to scale. The use of absolute positions in the objective function and combined bifurcation in bending produce numerous local minima, which leads to possibly various valid configurations for the same material. Moreover, many techniques are difficult to generalize to other solvers, because they employ non-standard material models [36, 143]. In this work, we leverage position-based simulation of compliant dynamics, also known as *extended position based dynamics (XPBD)* [95], see Section 5.3.3.

5.3 Method

Our goal is to estimate the material properties needed for the cloth simulator to represent particular cloth materials as realistically as possible. To achieve this, we develop a pipeline with three independent stages made from four components as visualized in Figure 5.1. Initially, a *capture system* gathers point or mesh data of a deformed rectangular cloth swatch in various configurations. Then, a template mesh is *registered* to the captured data using a landmark registration technique: the non-rigid iterative closest point (NR-ICP). Finally, the registered mesh is compared to a *simulation* of the template mesh, and an optimal set of parameters is found to align both as

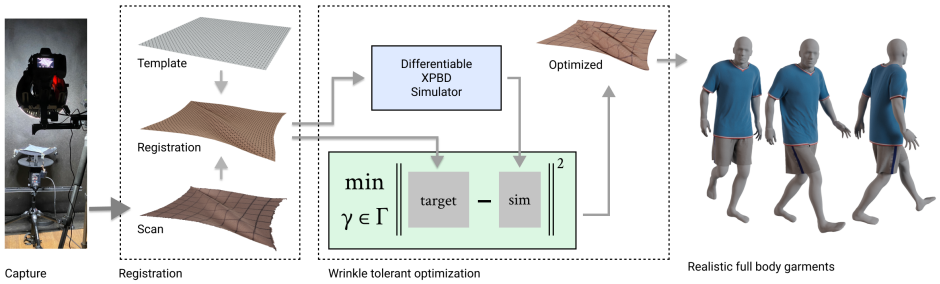


Figure 5.1: *Parameter estimation pipeline.* With our pipeline, we decouple the cloth capture (left) from the parameter optimization (middle-right) using NR-ICP mesh registration (middle-left). The optimization pass is able to handle wrinkled cloth, which greatly simplifies the capture process. With the optimized parameters γ, we generate realistic full-body cloth simulation (right) bypassing laborious manual parameter picking.

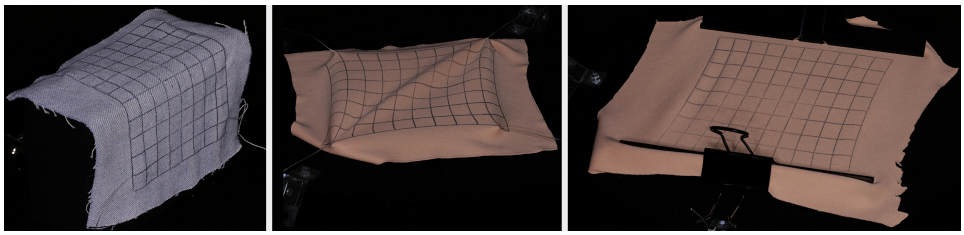


Figure 5.2: *Cloth capture.* Our simple cloth capture system records the cloth swatch under different force applications to drive the optimization.

closely as possible by using *least squares optimization*.

5.3.1 Capture system

We propose a lightweight cloth capture setup that can be easily reproduced with minimal cost. Cloth swatches are suspended using clamps that apply a controllable external force through weights, see Figure 5.2. The images are processed using *Agisoft Metashape* [5] to produce a high-resolution texture mapped 3D mesh. A free and open-source alternative is Meshroom [55]. The swatch is stamped with a regular grid pattern to provide landmarks for

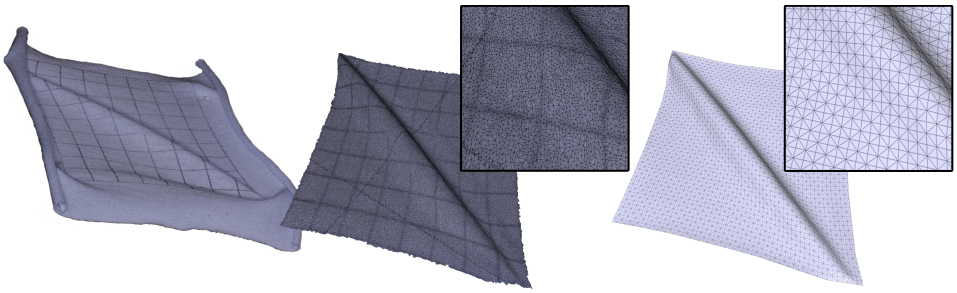


Figure 5.3: *Registration.* Capture setup (left), scanned mesh (middle), and registered mesh (right) with magnified regions in the insets.

the registration process described below.

5.3.2 Template registration

In contrast to prior art, our simulation targets are decoupled from the captured scans through mesh registration, see Figure 5.3. We create a triangulated regular grid mesh representing the simulated cloth swatch. This template is then registered to the scan by pairing vertices of the grid with landmarks on the scanned mesh via the NR-ICP method [82]. Vertices that are not aligned with the stamped grid pattern are projected along the mesh normal towards the surface of the scanned mesh to improve the position estimate in the normal direction. While more accurate methods have been developed for establishing dense correspondences between meshes, we stress that any stage of our pipeline – including template registration – is loosely coupled and can be replaced independently.

5.3.3 Simulation

XPBD is a recent constraint-based simulation algorithm that is often more stable and efficient when compared to expensive nonlinear solvers like Newton's method, which we used in previous chapters. It uses an iterative Gauss-Seidel solution for the linearized equations of motion. The method can be easily parallelized [51] and implemented on hardware such as multi-core

CPUs and GPUs, enabling interactive or real-time simulations on common modern hardware. While XPBD converges to the same result as Newton's method with spatial and temporal refinement, it can take significantly more iterations, especially for stiffer materials. Empirically, we found that stiffnesses exhibited in our experiments are sufficiently low such that XPBD can significantly outperform Newton solvers.

The method aims to solve Newton's equations of motion

$$\mathbf{M}\ddot{\mathbf{x}} = -\nabla U(\mathbf{x}), \tag{5.1}$$

where $\mathbf{x} \in \mathbb{R}^{3n}$ encodes n vertex positions (of the cloth mesh in this case) and $\mathbf{M}$ is the mass matrix computed from element volumes and constant material density ρ. The energy potential $U(\mathbf{x})$ needs to be specified in terms of a vector of constraint functions $\mathbf{C} = [C_1(x), C_2(x), \cdots, C_m(x)]^\top$ as

$$U(\mathbf{x}) = \frac{1}{2}\mathbf{C}(\mathbf{x})^\top \boldsymbol{\alpha}^{-1}\mathbf{C}(\mathbf{x}), \tag{5.2}$$

where $\boldsymbol{\alpha}$ is a block diagonal compliance matrix. Any energy that can be written this way is suitable for XPBD. Using implicit Euler time integration, the XPBD algorithm reduces to solving for the constraint multiplier updates $\Delta\boldsymbol{\lambda}$ with

$$(\nabla\mathbf{C}(\mathbf{x}_i)^\top \mathbf{M}^{-1}\nabla\mathbf{C}(\mathbf{x}_i) + \tilde{\boldsymbol{\alpha}})\Delta\boldsymbol{\lambda} = -\mathbf{C}(\mathbf{x_i}) - \tilde{\boldsymbol{\alpha}}\boldsymbol{\lambda}_i, \tag{5.3}$$

where $\tilde{\boldsymbol{\alpha}} = \frac{\boldsymbol{\alpha}}{\Delta t^2}$, followed by a position update

$$\Delta\mathbf{x} = \mathbf{M}^{-1}\nabla\mathbf{C}(\mathbf{x}_i)\Delta\boldsymbol{\lambda}. \tag{5.4}$$

The system in Eq. (5.3) is typically solved using Gauss-Seidel- or Jacobi-style updates.

Elasticity model

We employ an orthotropic StVK membrane energy model along with simple discrete bending [20] for modelling cloth. This model suggests a per-element

inverse compliance matrix of the form

$$\boldsymbol{\alpha}_{\triangle}^{-1} = A \begin{bmatrix} C_{00} & C_{01} & \\ C_{01} & C_{11} & \\ & & C_{22} \end{bmatrix},$$

where A is the area of a single triangle and C_{ij} are the compliance coefficients. The constraint function for each triangle is then defined to be the Green strain ϵ in Voigt notation

$$\mathbf{C}_{\triangle}(\mathbf{x}) = (\epsilon_{uu}, \epsilon_{vv}, 2\epsilon_{uv})^{\top}, \tag{5.5}$$

where subscripts u and v indicate warp and weft directions, respectively. With orthotropic Young's moduli E_u, E_v for modelling distinct warp and weft behaviour of the fabric, Poisson's ratios ν_{uv}, ν_{vu} and shear modulus μ, we have

$$\begin{bmatrix} C_{00} & C_{01} & \\ C_{01} & C_{11} & \\ & & C_{22} \end{bmatrix} = \frac{1}{1-\nu_{uv}\nu_{vu}} \begin{bmatrix} E_u & \nu_{vu}E_u & \\ \nu_{uv}E_v & E_v & \\ & & \mu(1-\nu_{uv}\nu_{vu}) \end{bmatrix}. \tag{5.6}$$

Note that this matrix is symmetric since $\nu_{vu}E_u = \nu_{uv}E_v$ and the Poisson's ratio ν_{vu} corresponds to a contraction in direction u when an extension is applied in direction v. In the following sections, we abbreviate $\nu_{uv} = \nu$ whereas ν_{vu} is computed from ν, E_u, and E_v.

The bending constraint [20] is defined for each pair of adjacent triangles ($\mathbf{x}_1$, $\mathbf{x}_3$, $\mathbf{x}_2$), ($\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_4$) as the angle strain

$$C_{\text{bend}} = \arccos\left(\frac{\mathbf{x}_{2,1} \times \mathbf{x}_{3,1}}{\|\mathbf{x}_{2,1} \times \mathbf{x}_{3,1}\|} \cdot \frac{\mathbf{x}_{2,1} \times \mathbf{x}_{4,1}}{\|\mathbf{x}_{2,1} \times \mathbf{x}_{4,1}\|}\right) - \phi_0,$$

where ϕ_0 is the rest dihedral angle and $\mathbf{x}_{i,j} = \mathbf{x}_i - \mathbf{x}_j$ are edge vectors between vertices i and j. The inverse compliance matrix is then given by the scalar bending stiffness: $\boldsymbol{\alpha}_{\text{bend}}^{-1} = [b]$.

Differentiable simulation

To use gradient-based optimization with this compliant constraint formulation, we compute the gradient with respect to the material parameters. Both bending energy and StVK membrane energy are differentiable. It suffices to compute the derivative of the position update in Eq. (5.4), which involves differentiating through Eq. (5.3). Derivatives are computed analytically and accumulated in tandem with the Gauss-Seidel or Jacobi iterates. We must compute $\partial \Delta \mathbf{x} / \partial \gamma$ where γ is the set of parameters we aim to recover, which we define in the next section.

5.3.4 Estimating elastic material parameters

In this section, we describe our method for finding a set of parameters γ required to reproduce a cloth shape captured under the influence of external forces. We focus on estimating static elasticity parameters that define the stress-strain relationship. We chose to use a dynamics simulator to find quasi-static solutions for two reasons. First, the inertia term in dynamics equations can be an effective method to speed up convergence to quasi-static solutions. Second, in future work we can easily extend this system to estimate dynamics parameters like friction or damping, which define how cloth behaves in motion.

Optimization problem

As output from the capture pipeline, we obtain a cloth mesh in a deformed configuration at rest. We introduce *shape descriptors* $\mathbf{s}$ that represent the cloth's shape in different ways as in Eq. (5.9). Below, $\mathbf{s}_{\text{target}}$ is used to describe the target shape. To obtain more information about the stress-strain relationship in the material and to resolve global scaling ambiguities, we additionally use force data from some of the boundary nodes. While other works have used full force vectors, we opt to capture only force magnitudes $\mathbf{f}_{\text{target}}$ to avoid the need for complex and costly setups necessary to collect accurate directional forces. We then match the corresponding shape $\mathbf{s}_{\text{sim}}$ and boundary forces $\mathbf{f}_{\text{sim}}$ of a simulation to the target by finding suitable

material properties. In general, we can stack as many meshes as needed into $\mathbf{s}$ and $\mathbf{f}$.

We optimize directly over the compliance coefficients and bending parameter by choosing the parameter set to be

$$\gamma := (C_{00}, C_{11}, C_{01}, C_{11}, b),\ \gamma \in \Gamma, \tag{5.7}$$

where Γ is a rectangular constraint set of feasible material parameter combinations. The chosen parameter set γ is a better candidate for optimization compared to Young's moduli and Poisson's ratio due to its good relative scaling. To penalize unrealistic Poisson's ratios ν above 1, we add the following penalty

$$W_\nu = \max(0, \nu - 1).$$

Putting everything together forms our final optimization problem

$$\min_{\gamma \in \Gamma} \|\mathbf{r}_{\text{sim}}(\gamma) - \mathbf{r}_{\text{target}}\|^2 + s_\nu W_\nu, \tag{5.8}$$

where $\mathbf{r}^\top = (\mathbf{s}^\top, \mathbf{f}^\top)$ is the vector of stacked shape descriptors $\mathbf{s}$ and boundary force magnitudes $\mathbf{f}$. Here, $\mathbf{r}_{\text{sim}}$ corresponds to a simulation after N timesteps of a quasi-static XPBD solve initialized to the target vertex positions, whereas $\mathbf{r}_{\text{target}}$ are the shape descriptors and measured force magnitudes of the registered mesh as described in Section 5.3.2. We weigh the Poisson penalty high with $s_\nu = 1e8$. To solve Eq. (5.8), we employ the Ceres non-linear least squares solver [4].

Shape descriptors

Many prior methods rely on precise control over cloth swatches to measure specific parameters [36, 103, 143]. In particular, it is convenient to produce deformations that are affected by just one or a few material parameters. For instance, the bending angle in a piece of cloth, as it hangs from the edge of a table, is determined by the bending parameter while being largely insensitive to changes in other stiffness parameters. Unfortunately, it is

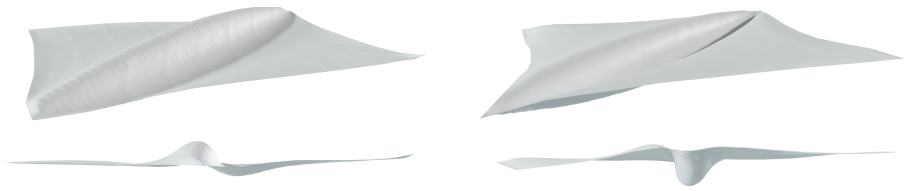


Figure 5.4: *Wrinkle bifurcation*: two valid cloth configurations with contrary wrinkle patterns shown from above (top) and side (bottom). The same cloth swatch is pulled with the same force at opposite corners.

difficult to design experiments that isolate material properties like shear stiffness and Poisson's ratio, since cloth tends to buckle in such scenarios producing wrinkles that inherently couple bending with other parameters.

Due to the bifurcation behaviour of wrinkling, deformed cloth can reach distinct equilibrium states depending on the deformation trajectory, initial conditions, and material shape memory effects. Consider Figure 5.4 with a square piece of cloth pulled at opposing corners for one example of such ambiguity in stable equilibrium states. Since real-world cloth can exhibit bias towards a particular wrinkle pattern, the example cloth swatch wrinkles in the same way independent of its orientation. When the cloth is flipped upside down, the pattern flips as well from a viewer perspective. However, this ambiguity poses a challenge for parameter optimization, because this effect is difficult to capture in simulation. In this case, simulated cloth will approach the same observed wrinkle pattern due to gravity. Here, naive position-based metrics will result in a very different value for both valid cloth configurations. Another example for ambiguity is visualized in Figure 5.6 where simulating the same material with varying initial conditions results in two distinct equilibrium states.

While many previous methods simply avoid wrinkles in the captured data [36, 103, 143], we instead propose a novel objective to handle such ambiguities and are able to process coupled-parameter captures, which in turn allows for simpler and more affordable capture pipelines.

As outlined above, the evaluation of the objective in Eq. (5.8) results in a

very different value for both equilibrium states in Figure 5.4 when choosing $\mathbf{s}$ to be a stacked locator of vertex positions $\mathbf{p}$, although both target swatches are generated with the same material properties. In general, positional loss encoding is sensitive to small differences in the captured cloth. Thus, we developed an objective function that reports a similar error for wrinkle patterns generated by the same material. We evaluate the following shape descriptors:

$$\begin{aligned} \mathbf{s}_{\text{pos}}(\mathbf{p}) &= \mathbf{p}, \\ \mathbf{s}_{\text{energy}}(\mathbf{p}) &= \mathbf{U}_{\text{triangle}}(\mathbf{p}) + \mathbf{U}_{\text{bend}}(\mathbf{p}), \\ \mathbf{s}_{\text{strain}}(\mathbf{p}) &= \mathbf{C}_{\text{triangle}}(\mathbf{p}) + \mathbf{C}_{\text{bend}}(\mathbf{p}), \\ \mathbf{s}_{\text{FFT}}(\mathbf{p}) &= \texttt{FFT}(\mathbf{p}), \end{aligned} \tag{5.9}$$

where $\mathbf{U}$ is either a triangle or bending *energy* and $\mathbf{C}$ is the *strain* as defined in Section 5.3.3. The position descriptor $\mathbf{s}_{\text{pos}}$ represents the swatch shape in terms of vertex positions, leading to evaluating the absolute position of wrinkles. In contrast, $\mathbf{s}_{\text{energy}}$ and $\mathbf{s}_{\text{strain}}$ measure the per-element energies and strain, respectively, which determine how much the cloth is stretched. They differ in $\mathbf{U}$ usually being a squared strain scaled by the material parameters. This implies that $\mathbf{s}_{\text{target}}$ depends on γ, which complicates the computation of the corresponding derivatives. The finite Fourier transform (FFT) descriptor $\mathbf{s}_{\text{FFT}}$ applies the normalized 2D FFT to grid vertices. The normalization removes phase information, leading to small metric differences when comparing distinct wrinkle phases and larger discrepancies for varying wrinkle amplitude or frequency.

Our experiments show that the energy and strain descriptors $\mathbf{s}_{\text{energy}}$ and $\mathbf{s}_{\text{strain}}$ work well when bifurcations are symmetric as in Figure 5.4. However, for fine wrinkles as in Figure 5.5, only the FFT descriptor $\mathbf{s}_{\text{FFT}}$ performs robustly as we elaborate in the next section.

5.4 Evaluation

In this work, we use the centimetre-gram-second (CGS) system of units for material parameters. While our optimization results in optimal parameters

γ as defined in Eq. (5.7), we often report the traditional, more intuitive parameters E_u, E_v, μ, ν instead of C_{00}, C_{11}, C_{01}, and C_{22} as defined in Eq. (5.6). In the following sections, we use γ to represent parameters in both forms depending on context.

A popular experiment for cloth is the *picture frame shear*. Here, a square piece of cloth is fixed on all sides onto a rigid "picture frame", which is then deformed at the hinges. For validation purposes, we hand-pick fictional properties for three synthetic materials, see Table 5.1 (top), that we call cotton, denim, and silk as they produce visually similar shapes to these fabrics. We then simulate a picture frame experiment using two different initial conditions for all three materials. For this experiment we use a square cloth swatch with a side length of 15 cm. The two initial conditions are perturbed vertex positions and a flat alignment of all vertices. The simulations with varying initial conditions converge to significantly different equilibrium states for each material as shown in Figure 5.6, which demonstrates the ambiguity mentioned in Section 5.3.4.

5.4.1 Choice of objective function

To evaluate the effectiveness of the proposed shape descriptors $\mathbf{s}$, we generate a synthetic cotton picture frame experiment with our XPBD simulator and γ_{cotton}, see target shape in Figure 5.5. Starting with perturbed parameters $\gamma_{\text{initial,c}}$ from Table 5.1 (middle), we optimize for material parameters with each shape descriptor from Eq. (5.9). We then run four simulations with the found parameter sets, all initialized to the target cloth shape. As demonstrated in Figure 5.5, the optimization using the parameters retrieved through the FFT descriptor $\mathbf{s}_{\text{FFT}}$ converges to the exact target cloth shape while $\mathbf{s}_{\text{energy}}$ and $\mathbf{s}_{\text{strain}}$ produce the largest differences to the target.

Furthermore, we assess the ability of $\mathbf{s}_{\text{pos}}$ and $\mathbf{s}_{\text{FFT}}$ to discriminate between different materials. Thereby, we evaluate both descriptors on simulations of the same material with varying initial conditions and on simulations of different materials. We want the descriptors to be similar for the first case, but divergent for the second. We evaluate cloth simulations for

three materials $m \in \{\text{cotton, denim, silk}\}$ with two varying initial conditions $\{\mathbf{p}_{\text{initial1}}, \mathbf{p}_{\text{initial2}}\}$ on a cloth swatch as described in Figure 5.6. The descriptor difference $\Delta\mathbf{s}$ within one material, here for FFT, is

$$\Delta s_{FFT,m} = \|\mathbf{s}_{FFT,m}(\mathbf{p}_{\text{initial1}}) - \mathbf{s}_{FFT,m}(\mathbf{p}_{\text{initial2}})\|_2.$$

The delta across two different materials m, $m2$ with
$m2 \in \{\{\text{cotton, denim, silk}\} \setminus m\}$ is defined as

$$\Delta s_{FFT,m,m2,i,j} = \|\mathbf{s}_{FFT,m}(\mathbf{p}_i) - \mathbf{s}_{FFT,m2}(\mathbf{p}_j)\|_2,$$

with $i, j \in \{\text{initial1, initial2}\}$. Since these absolute values are not necessarily comparable across different descriptors, we instead compare relative differences:

$$\Delta \hat{s}_{FFT,m,m2,i,j} = \Delta s_{FFT,m,m2,i,j} - \Delta s_{FFT,m},$$

where positive values indicate that the descriptor reports larger shape differences in different materials than varying cloth configurations for one material, as desired. With $m2$ and i, j, we obtain eight instances of $\Delta \hat{s}_{FFT,m,m2,i,j}$ for each material m. Figure 5.7 shows that the FFT shape descriptor correctly produces higher differences for varying materials compared to distinct configurations of the same material. Hence, we choose the FFT shape descriptor for our objective function.

5.4.2 Validation

To analyze the accuracy of our novel parameter estimation, we generate four synthetic cotton simulations with our XPBD simulator: two picture frame deformations and two stretch experiments as visualized in Figure 5.9. To allow for precise comparison of the retrieved and target parameter sets, we use the same XPBD simulator as we employ in the optimization. Starting with the same perturbed parameter set $\gamma_{\text{initial,c}}$ as in the previous section, our optimizer recovers the target parameters very well after 13 iterations as shown in Table 5.1 (bottom) – although only four target cloth shapes are

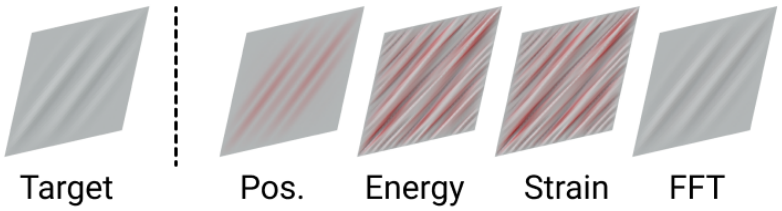


Figure 5.5: *Shape descriptor evaluation.* Target cloth is simulated with XPBD (left) followed by optimization results for each descriptor (right). Euclidean vertex distances to the target are color-coded with a maximum of 1.8 mm.

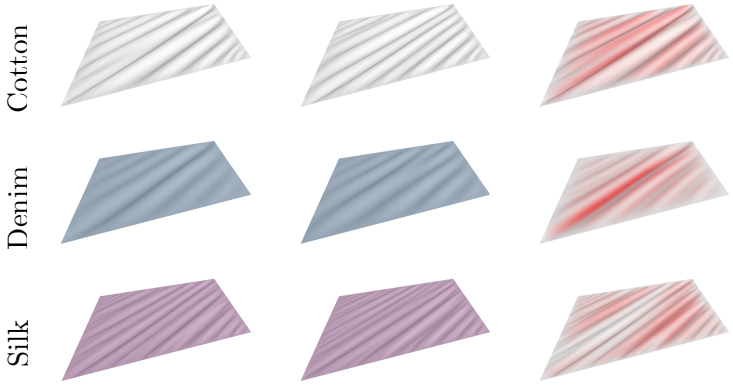


Figure 5.6: *Distinct equilibrium states for the same material.* Two simulations for each material with different initial conditions: perturbed vertices (left), flat configuration (middle), and color-coded Euclidean distance between both meshes in red (right), with a maximum distance of 4.3 mm.

used. Remarkably, the bending parameter is recovered within a 4% error in spite of there being no explicit bending examples in the target set. The membrane parameters are recovered within 0.5% error.

In Figure 5.8 we further compare the shapes of unseen cloth configurations re-simulated with our retrieved parameter set $\gamma_{\mathrm{FFT,c}}$ to target simulations created with γ_{cotton}. The error in the recovered parameters causes a difference in vertex positions below 0.28 mm, which is largely imperceptible.

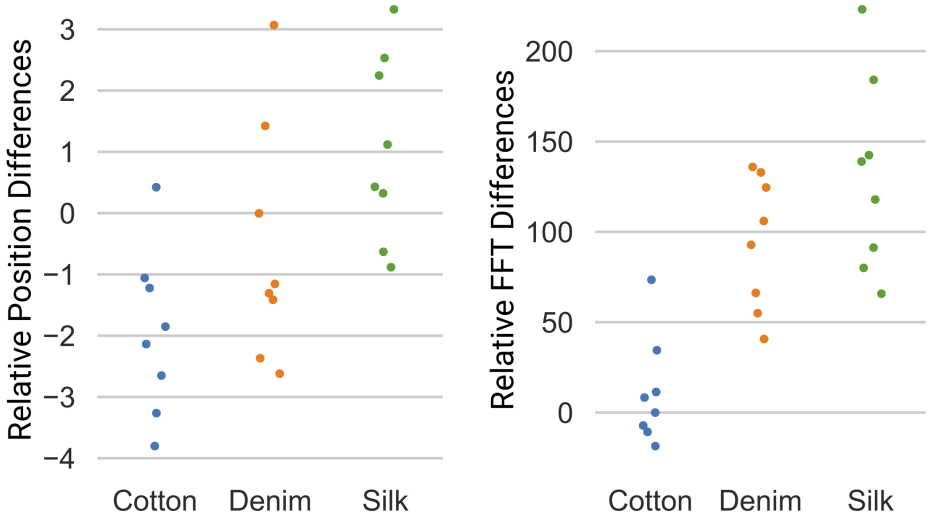


Figure 5.7: *Relative metric differences.* Values of $\Delta\hat{s}_{\text{pos},m,m2,i,j}$ (left) and $\Delta\hat{s}_{\text{FFT},m,m2,i,j}$ (right) where m is the x-axis: $\mathbf{s}_{\text{pos}}$ doesn't necessarily produce higher metric differences across materials compared to varying cloth configurations for the same material, as shown by the negative and close-to-zero values (left), while $\mathbf{s}_{\text{FFT}}$ does as observed in the mostly positive values (right).

5.5 Results

Synthetic Experiments We show that our method is capable of reproducing cloth simulations from third-party software. First, we generate a set of square swatch targets in Houdini [125] with different material presets; the targets for silk are depicted in Figure 5.10. Then, we estimate the bending parameter b using the bending scenarios of Figure 5.10 (right). The membrane parameters $(C_{00}, C_{11}, C_{01}, C_{22})$ are estimated together in a separate pass by pulling the corners of the suspended cloth swatch as in Figure 5.10 (left). This process is repeated thrice to ensure that the coupling between bending and membrane stiffnesses is not lost. Furthermore, to avoid getting stuck in local minima, we run the same optimization from three different randomized starting points.

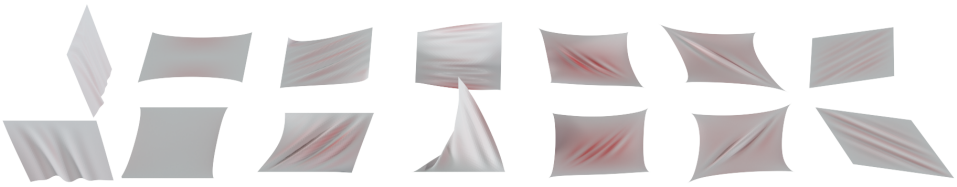


Figure 5.8: *Re-simulation of various cloth configurations with $\gamma_{FFT,c}$ estimated from XPBD targets with γ_{cotton}.* The vertex displacement error is color-coded in red, with a small maximum error of 0.28 mm. The second and last columns are targets while the rest are unseen by the optimization.

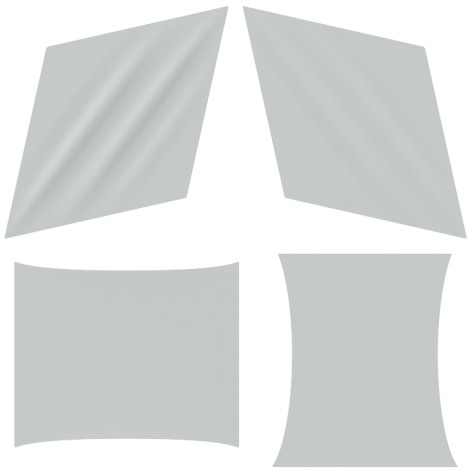


Figure 5.9: *XPBD validation targets.*

This decoupling technique produces a lower objective, i.e., a better fit, compared to optimizing all parameters at once. In the future, we aim to further eliminate the need for separate passes for bending and membrane parameters to produce a more streamlined pipeline. The comparison to Houdini silk targets is illustrated in Figure 5.11. For all deformations apart from the second column, the local vertex displacement is below 2.4 mm. In the second column, the error is clustered at large strain regions, which indicates that our StVK model deviates from Houdini's material model for large strain deformations.

The retrieved parameters are listed in Table 5.2. By optimizing for these parameters, we avoid manual hand-picking of material parameters as in Section 5.4, but instead, are able to reproduce the cloth behaviour of the physically more accurate and computationally more expensive Houdini FEM simulator as demonstrated in Figure 5.12. Here, we re-simulate the Houdini experiment with our XPBD simulator by using the same initial and boundary conditions as in Houdini. We thereby demonstrate that our pipeline is capable of effectively reproducing the aesthetics of different cloth materials generated by third-party simulators.

Real-world experiments. With real-world captures of cloth subject to various force applications and deformations, we generate six targets each for three different materials as shown in Figure 5.13 using a 9.6 cm cloth swatch. The first two targets are generated by applying a 200 g weight on opposing sides of the cloth using wide paper clamps as shown in Figure 5.2 (right). The third target is generated by applying a 200 g weight on all corners while the fourth is created by applying 20 g at two diagonally opposite corners and 100 g on the remaining corners. The last two targets are generated by letting the cloth drape from a horizontal ledge.

We optimize for the parameters of each material, which are shown in Table 5.3. Then, we re-simulate the target with the estimated parameters by setting the initial vertex positions to the target shapes, see Figure 5.14. Our results demonstrate that our method is able to closely match the captured targets within 3.7 mm of maximum vertex displacement. Our optimization correctly determines higher Young's moduli for a stiff material like denim

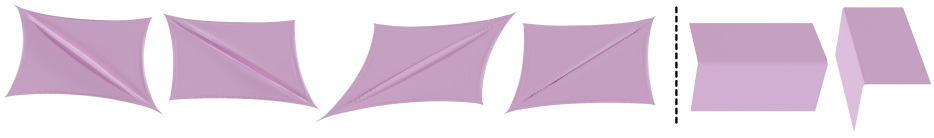


Figure 5.10: *Synthetic silk targets generated with Houdini,* used for estimating membrane stiffnesses (left) and bending stiffnesses (right).

Figure 5.11: *Re-simulation with the estimation of Houdini's silk preset.* The square cloth swatch is re-simulated using the estimated parameters using our XPBD solver. The set of targets used during optimization (training set) includes configurations from columns 2, 3 and 7. The vertex displacement error is color-coded in red with a maximum set to 2 mm. In the second column, errors are clustered on the large stretch areas with a maximum of 6 mm while it is below 2.4 mm in the remaining columns.

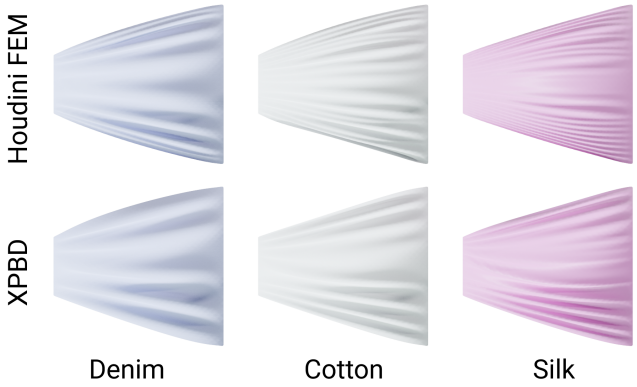


Figure 5.12: *Aesthetic evaluation.* By twisting a piece of cloth, unseen by optimization, we demonstrate that distinct wrinkle patterns generated by Houdini's FEM simulator are reproduced using our XPBD simulator with optimized material parameters.

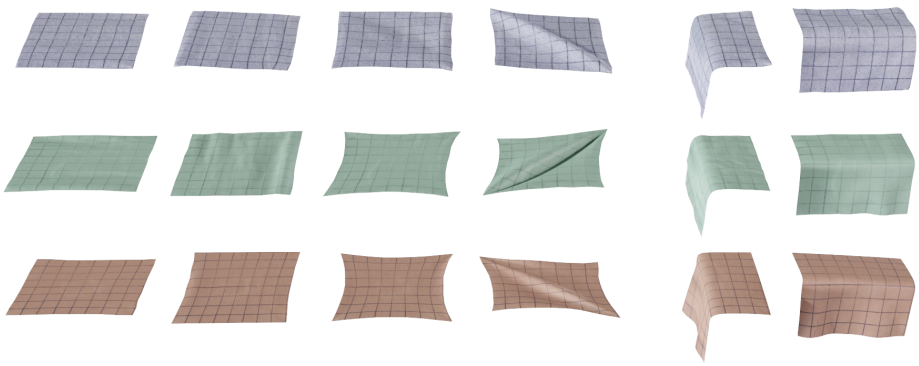


Figure 5.13: *Real-world targets.* A piece of denim (top), cotton (middle), and polyester (bottom) is deformed in various configurations. The target meshes are generated through template registration as described in Section 5.3.2.

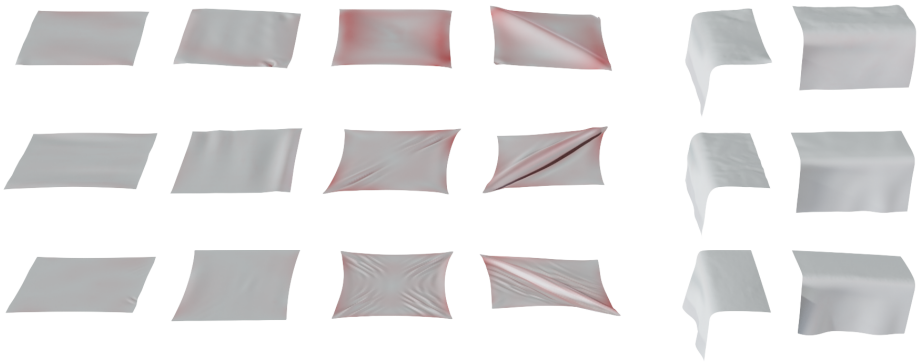


Figure 5.14: *Real-world results.* Denim (top), cotton (middle), and polyester (bottom) cloth is simulated with our estimated parameters. The vertex position error to the targets in Figure 5.13 is colored in red with a maximum of 3.7 mm.

and lower for the softer polyester material, as well as a larger bending stiffness in denim compared to cotton and polyester.

Note that the stiffness parameters in Table 5.3 are lower than in Table 5.2, since the forces applied on the boundary are smaller in the real-world case, e.g., 0.2 Mdyn vs. 3 Mdyn, which leads to a similar amount of strain. As our method aims at small strain deformations, we expect to

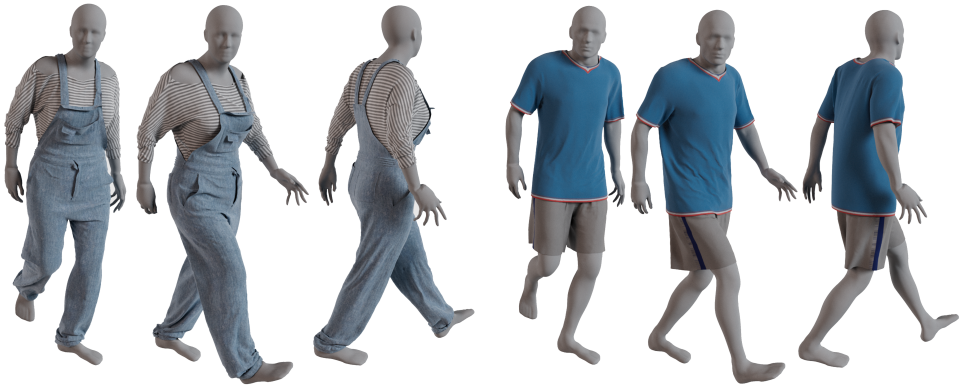


Figure 5.15: *Outfit simulation:* a denim overall and cotton top (left) and a polyester soccer outfit (right).

see lower stiffness estimates. However, we plan to extend our analysis to more sophisticated large strain models, and experiment with larger forces for captured data in the future.

To further validate that our results produce the desired look, we simulate full outfits using the estimated parameters in Figure 5.15.

5.6 Limitations and future work

We presented a novel approach to cloth captures and optimization that simplifies the inverse design problem of parameter estimation. Despite improving robustness and simplicity over prior work, several limitations remain. For instance, the accuracy of our template registration technique is limited to the resolution of the stamped pattern. Additionally, as is common with non-linear optimization, our system is sensitive to objective scaling even with our FFT metric, and heuristics are required to establish a reasonable scaling between the shape descriptor metric and the force metric.

As part of future work, we aim to implement the method on GPUs to optimally leverage available compute resources and to greatly speed up optimization time. Additionally, we intend to extend our research to different material models such as piece-wise constant stiffnesses for a piece-wise linear elastic model [143] or the stable neo-Hookean model for volumetric simula-

tions [94].

Our method uses a constant simulation time to reach equilibrium. While time to equilibrium is generally short close to the optimum, we can expect it to increase when initial parameters are chosen poorly. We plan to use adaptive simulation times in the future, which will further improve the robustness of our estimates.

5.7 Conclusion

We have presented a novel, simple pipeline to capture, register, and optimize for cloth material parameters using a simulation engine. Our proposed pipeline consists of three separate stages that can be improved upon individually. We proposed a novel metric that enables us to capture coupled cloth shearing, stretching, and bending effects, which in turn allows for cheaper and easier capture setups. Our material parameters are not limited to the XPBD framework and can be reused with different simulation techniques that employ the same material model.

This work provided the invaluable insight of the importance of efficiency for forward simulation. Even without back-propagation [42, 85], we found that typical optimization workloads can take weeks on CPU. However, having a differentiable frictional contact model can maintain the second-order convergence guarantees during parameter optimization.

	E_u	E_v	μ	ν	b
γ_{cotton}	$1.0e5$	$2.8e5$	$4.0e4$	0.4	50
γ_{denim}	$5.0e5$	$7.0e5$	$2.0e4$	0.45	200
γ_{silk}	$2.0e5$	$3.0e5$	$1.5e4$	0.35	10
$\gamma_{\text{initial,c}}$	$2.0e5$	$1.8e5$	$5.0e4$	0.3	10
$\gamma_{\text{FFT,c}}$	$9.98e4$	$2.79e5$	$3.98e4$	0.40	51.96

Table 5.1: *Material properties for validation with XPBD.* Hand-picked properties for three synthetic materials (top), perturbed initial conditions for cotton (middle), and the recovered parameter set $\gamma_{\text{FFT,c}}$ (bottom) with inputs as in Figure 5.9.

Material (ρ in g/cm^2)	E_u	E_v	μ	ν	b
Denim (0.0324)	34182	33735	5592	0.2609	11.0
Cotton (0.0224)	30823	30266	4971	0.3058	1.19
Silk (0.0187)	7492	7436	1193	0.03543	0.100

Table 5.2: *Estimation of material parameters for Houdini's FEM cloth solver.*

Material (ρ in g/cm^2)	E_u	E_v	μ	ν	b
Denim (0.0324)	3793	20590	6968	0.4286	28.5
Cotton (0.0224)	1840	2019	6538	0.4308	8.99
Polyester (0.0187)	1028	3271	9171	0.2731	2.35

Table 5.3: *Estimation of material parameters for real-world targets.*

Chapter 6

Conclusion

This dissertation investigates various limitations of modern methods for simulating real-world phenomena involving soft interacting objects. Frictional contact phenomena are particularly difficult to represent in the virtual world. Contacts between objects are not easily represented digitally. They are often modelled via constraints that enforce penetration-free states between discretizations of various elastic media. This makes friction modelling even more difficult since it further conditions the motion of objects along the boundary of constraint violation. It is thus unsurprising that performance bottlenecks and visual artifacts in simulation often occur due to limitations in friction and contact handling. To this day there are no standard methods for reliable simulation of friction and contact without compromises on performance, accuracy or robustness. In this work, we address specific deficiencies in the accuracy of sliding contacts and friction simulation in the context of elastodynamics.

In Chapter 3, we identify and address a limitation of common contact handling methods, which use piecewise linear representations to model the physical surface of touching objects. When modelling contact between organic objects like plants, animals, or clothing, piecewise linear representations form a poor approximation to the true surface. In sliding contacts, piecewise linear surface models can mutually interlock, creating artificial spikes in friction that otherwise may not exist if the surfaces were refined.

This can cause unexpected behaviour in simulations as meshes are refined for higher fidelity or coarsened for better performance. We address this issue by modelling contact using smooth deformable implicit surfaces, which are created from oriented point clouds coarsely sampling simulated meshes. Our method resolves artificial friction effects and unexpected change in behaviour due to mesh refinement or coarsening. Contact resolution using implicit surfaces offers many advantages. For one, it can be applied to meshless methods or even LiDAR point clouds [98]. Other applications can include contact in fluid simulation where frequent topology changes are automatically handled by implicit surface representations.

In Chapter 4, we address a fundamental issue affecting friction accuracy in current state-of-the-art methods for modelling frictional contacts. It is particularly difficult to accurately model the transition between sliding and sticking contacts. Traditional methods represent this behaviour via explicit bimodal models, where sticking contacts follow one set of equations, and sliding contacts are resolved via another. The transition is then inherently non-differentiable. This non-smoothness can be a strain on the robustness and performance of numerical methods, and it can make the simulation impractical to use for inverse problems. Modern methods resolve these issues via smooth approximations for these transitions at the cost of accuracy, where contacting objects at rest will slide, even when they should be sticking. In this work we show that smooth approximations can be made substantially more accurate when friction is handled implicitly — meaning that friction forces are resolved simultaneously with all other internal forces involved in the simulation. Furthermore, we take advantage of the smoothness of contact surfaces from the methods developed in Chapter 3 to develop a simple and fully differentiable formulation. We further showed how our method can reproduce volume preservation phenomena using soft constraints, maintaining differentiability of the pipeline.

Gaining popularity in computer graphics, differentiable simulation is a powerful tool for automating simulation pipelines. It liberates artists from arduously picking simulator parameters by hand, and closing the sim-to-real gap between captured real-world phenomena and simulation results. Differ-

entiable simulation unlocks so many applications, that simulators are often viewed merely as tools or a means to acquire plausible estimates of real-world behaviour. Due to a myriad of numerical difficulties like artificial stiffening, damping and other subtle modelling limitations, simulations are never true to real-world behaviour unless said behaviour is carefully controlled. However, simulations are controlled via material models and parameters, which can be tweaked to offset inconsistencies with real-world behaviour. If automated, tweaking simulation parameters has the potential to be powerful enough to significantly improve simulation performance by choosing simpler material models and simultaneously be more accurate since results can be effectively adjusted to match real-world captures.

In Chapter 5, we address a specific issue when matching fabric simulations to real-world captures in a differentiable simulation pipeline. Our aim is to automatically determine stretching and bending stiffnesses involved in driving cloth simulations, from real-world captured cloth. When matching real-world scans to simulated cloth it is essential to define a measure of how closely the two match. Most commonly, researchers use vertex based or locator based measures where distances of corresponding locations on the simulated mesh and the scan are measured, then an aggregate is computed. This often works well, especially for volumetric objects, however, matching cloth poses an interesting limitation. Assuming this similarity measure, we found that cloth simulation can exhibit vastly different equilibrium states for the same or slightly perturbed material properties or initial conditions. This is especially true in the presence of dissipative forces like friction, air drag or damping. To fix this issue, we developed an alternative measure to compare the shape of cloth patches in frequency space. This allowed for greater flexibility in the types of deformations we could perform to determine material parameters of real pieces of cloth. This allows us to build a more robust method for capturing real-world cloth behaviour using a simple and highly efficient simulation method.

6.1 Limitations and future work

This section summarizes some important limitations in the methods presented and proposes potential solutions and avenues for future work.

Self-contact. Our contact model using implicit surfaces does not support self-contact. While this model may work for many applications, it lacks generality to handle common scenarios like animation driven inputs where body parts may temporarily self-intersect. Currently, the most promising solution here is to evaluate the implicit surface in rest configuration.

Global convergence of nonlinear equations. Solving for fully implicit frictional contact forces in Chapter 4 comes with a fundamental trade-off. The lagged friction method benefits from good convergence guarantees of local optimization. On the other hand, our method relies on root-finding, which is more general but more difficult to solve. Although continuation or homotopy methods can address this shortcoming, convergence guarantees are not regained. However, it is not inconceivable that a cleverly designed merit function could avoid local minima.

Performance of friction solves. Large coefficients of friction μ or small friction error ϵ can produce particularly stiff ODEs, which can substantially affect performance. One potential approach here is to adaptively decrease ϵ to a desired accuracy. This could help guide Newton iterations towards a solution more quickly.

Smooth frictional contact method for XPBD. One advantage of our friction formulation in Chapter 4 is that it is simple and does not require a special algorithm for resolving frictional contacts. Since frictional contact is part of the equations of motion, it can be directly solved for in XPBD. However, it is unclear if such a system would produce practical results. This is in part because large friction forces can produce very stiff ODEs, which are difficult to solve even with Newton solvers as we have demonstrated in Chapter 4.

Differentiable simulation. Our method uses the forward simulation to compute derivatives with respect to all material parameters. This is efficient, however it can become impractical with large numbers of parameters, or when optimized parameters are unknown prior to generating the forward simulation. Modern differentiable simulation pipelines are built around the adjoint method [85], which back-propagate positional derivatives to allow for a more flexible pipeline. This can also be applied in XPBD simulation.

In this dissertation, we have demonstrated and addressed a number of issues in current methods for simulating elastodynamic systems and their ability to match real-world behaviour. Along the way, we have uncovered fruitful new areas of research and hinted at promising future directions. The problems and methods presented here aim to capture a small but modern snapshot of research bringing future virtual worlds closer to reality.

Bibliography

[1] V. Acary and B. Brogliato. *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics*. Springer Science & Business Media, 2008. 19, 20

[2] M. J. Adams, B. J. Briscoe, and S. A. Johnson. Friction and lubrication of human skin. *Tribology Letters*, 26(3):239–253, Apr. 2007. ISSN 1023-8883, 1573-2711. doi: 10.1007/s11249-007-9206-0. URL http://link.springer.com/10.1007/s11249-007-9206-0. 22

[3] A. Adamson and M. Alexa. Approximating and intersecting surfaces from points. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, page 230–239, Goslar, DEU, 2003. Eurographics Association. ISBN 1581136870. 17

[4] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org, 2022. 106

[5] Agisoft. Agisoft Metashape. https://www.agisoft.com/, 2022. 101

[6] J. P. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large data visualization. *Visualization Handbook*, 01 2005. 43, 85

[7] P. P. 'Alart and A. R. Curnier. A mixed formulation for frictional contact problems prone to newton like solution methods. *Computer Methods in Applied Mechanics and Engineering*, 92(3):353–375, 1991. ISSN 0045-7825. doi: https://doi.org/10.1016/0045-7825(91)90022-X. URL http://www.sciencedirect.com/science/article/pii/004578259190022X. 19

[8] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *Proceedings Visualization, 2001. VIS '01.*, pages 21–29, 537, 2001. 17

[9] R. Alexander. Diagonally implicit runge–kutta methods for stiff o.d.e.'s. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977. doi: 10.1137/0714068. URL https://doi.org/10.1137/0714068. 11

[10] S. Altmayer. rstar, 2020. URL https://github.com/Stoeoef/rstar. 43

[11] M. Anderson. Tennis ball hitting the wall at 100 mph, Feb. 2018. URL https://www.youtube.com/watch?v=FC8Tpi3U0H0. 93

[12] U. M. Ascher and L. R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998. 9, 80

[13] U. M. Ascher, E. Larionov, S. H. Sheen, and D. K. Pai. Simulating deformable objects for computer animation: A numerical perspective. *Journal of Computational Dynamics*, 9(2):47–68, 2022. 11, 12, 64

[14] J. Awrejcewicz, D. Grzelczyk, and Y. Pyryev. 404. a novel dry friction modeling and its impact on differential equations computation and lyapunov exponents estimation. *Journal of Vibroengineering*, 10(4), 2008. 20, 67

[15] R. Balasubramanian and V. J. Santos. *The human hand as an inspiration for robot hand development*, volume 95. Springer, 2014. 48

[16] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '89, page 223–232, New York, NY, USA, 1989. Association for Computing Machinery. ISBN 0897913124. doi: 10.1145/74333.74356. URL https://doi.org/10.1145/74333.74356. 18

[17] D. Baraff. 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, pages 19–28, Dallas, TX, USA, Sept. 1990. Association for Computing Machinery. ISBN 978-0-89791-344-7. doi: 10.1145/97879.97881. URL https://doi.org/10.1145/97879.97881. 16

[18] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1998. 18, 100

[19] M. Becker, M. Ihmsen, and M. Teschner. Corotated sph for deformable solids. In *NPH*, pages 27–34. Citeseer, 2009. 15

[20] J. Bender, M. Müller, and M. Macklin. A survey on position based dynamics, 2017. In *EUROGRAPHICS 2017 Tutorials*. Eurographics Association, 2017. 64, 103, 104

[21] F. Bertails-Descoubes, F. Cadoux, G. Daviet, and V. Acary. A nonsmooth newton solver for capturing exact coulomb friction in fiber assemblies. *ACM Transactions on Graphics (TOG)*, 30(1):6, 2011. 14, 18, 19, 24, 26, 38, 39, 67, 73

[22] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*, 2012. 77

[23] K. S. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popović, and S. M. Seitz. Estimating cloth simulation parameters from video. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, page 37–51, Goslar, DEU, 2003. Eurographics Association. ISBN 1581136595. 99

[24] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph.*, 33(4), July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601116. URL https://doi.org/10.1145/2601097.2601116. 15, 100

[25] F. P. Bowden and D. Tabor. *The Friction and Lubrication of Solids*. Clarendon Press Oxford, 1950. 22

[26] G. E. Brown, M. Overby, Z. Forootaninia, and R. Narain. Accurate dissipative forces in optimization integrators. *ACM Trans. Graph.*, 37(6), dec 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275011. URL https://doi.org/10.1145/3272127.3275011. 64

[27] Browzwear. Browzwear Fabric Analyzer. https://browzwear.com/products/fabric-analyzer/, 2022. 99

[28] M.-P. Cani. An implicit formulation for precise contact modeling between flexible solids. In *20th annual conference on Computer graphics and interactive techniques (SIGGRAPH'93))*, pages 313–320. ACM SIGGRAPH, 1993. 16

[29] M.-P. Cani and M. Desbrun. Animation of Deformable Models Using Implicit Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):39 – 50, 1997. doi: 10.1109/2945.582343. URL https://hal.inria.fr/inria-00537529. Published under the name Marie-Paule Cani-Gascuel. 16

[30] C. Canudas de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995. 21

[31] H.-y. Chen, E. Tretschk, T. Stuyck, P. Kadlecek, L. Kavan, E. Vouga, and C. Lassner. Virtual elastic objects. *arXiv preprint arXiv:2201.04623*, 2022. 100

[32] Z.-Q. Cheng, Y.-Z. Wang, B. Li, K. Xu, G. Dang, and S.-Y. Jin. A survey of methods for moving least squares surfaces. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'08, page 9–23, Goslar, DEU, 2008. Eurographics Association. ISBN 9783905674125. 17

[33] P. G. Ciarlet. *Lectures on three-dimensional elasticity*, volume 71.;71;. published for the Tata Institute of Fundamental Research [by] Springer, Berlin, 1983. ISBN 9780387123318;0387123318;. 7

[34] CLO 3D. CLO Fabric Kit 2.0. https://www.clo3d.com/, 2018. 99

[35] CLO 3D. CLO. https://support.clo3d.com/hc/en-us/articles/360041074334-Fabric-Kit-Manual, 2022. 99

[36] D. Clyde, J. Teran, and R. Tamstorf. Modeling and data-driven parameter estimation for woven fabrics. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350914. doi: 10.1145/3099564.3099577. URL https://doi.org/10.1145/3099564.3099577. 99, 100, 106, 107

[37] C. Coulomb. Théorie des machines simples en ayant égard au frottement de leurs parties et à la roideur de leurs cordages. *Bachelier, Librairie, quai des Augustins, Paris*, 1785. 20

[38] P. R. Dahl. A solid friction model. Technical report, Aerospace Corp El Segundo Ca, 1968. 20

[39] G. Daviet, F. Bertails-Descoubes, and L. Boissieux. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. volume 30, page 139. ACM, 2011. 19, 24, 26, 38, 57, 58, 73

[40] G. De Saxcé and Z. Q. Feng. The bipotential method: A constructive approach to design the complete contact law with friction and improved numerical algorithms. *Mathematical and Computer Modelling*, 28(4):225–245, Aug. 1998. ISSN 0895-7177. doi: 10.1016/S0895-7177(98)00119-8. URL http://www.sciencedirect.com/science/article/pii/S0895717798001198. 13, 25, 36, 73

[41] M. Desbrun and M.-P. Cani. Animating Soft Substances with Implicit Surfaces. In R. Cook, editor, *The 22nd Annual Conference on Computer Graphics (SIGGRAPH'95)*, volume 29 of *Annual Conference Series*, pages 287–290, Los Angeles, United States, Aug. 1995. ACM SIGGRAPH, Addison Wesley. doi: 10.1145/218380.218456. URL https://hal.inria.fr/inria-00537542. published under the name Marie-Paule Gascuel. 16

[42] T. Du, K. Wu, P. Ma, S. Wah, A. Spielberg, D. Rus, and W. Matusik. Diffpd: Differentiable projective dynamics with contact. *arXiv preprint arXiv:2101.05917*, 2021. 100, 118

[43] M. A. D. Ehrhardt, D. G. Ferreira, and S. A. Santos. A pattern search and implicit filtering algorithm for solving linearly constrained minimization problems with noisy objective functions. *Optimization Methods and Software*, 34:827 – 852, 2019. 99

[44] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact newton method. *SIAM Journal on Scientific Computing*, 17 (1):16–32, 1996. doi: 10.1137/0917003. URL https://doi.org/10.1137/0917003. 75

[45] E. English and R. Bridson. Animating developable surfaces using nonconforming elements. *ACM Trans. Graph.*, 27, 08 2008. doi: 10.1145/1360612.1360665. 15

[46] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A Hybrid Particle Level Set Method for Improved Interface Capturing. *Journal*

of Computational Physics, 183(1):83 – 116, 2002. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.2002.7166. URL http://www.sciencedirect.com/science/article/pii/S0021999102971664. 16

[47] M. Erdmann. On a representation of friction in configuration space. *The International Journal of Robotics Research*, 13(3):240–271, 1994. doi: 10.1177/027836499401300306. URL https://doi.org/10.1177/027836499401300306. 14

[48] K. Erleben. Rigid body contact problems using proximal operators. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '17, pages 13:1–13:12, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5091-4. doi: 10.1145/3099564.3099575. URL http://doi.acm.org/10.1145/3099564.3099575. 19, 24, 67, 73

[49] G. Fichera. *Problemi elastostatici con vincoli unilaterali: il problema di Signorini con ambigue condizioni al contorno*. Accademia nazionale dei Lincei, 1964. 34

[50] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–269, 2002. 36

[51] M. Fratarcangeli, V. Tibaldo, and F. Pellacini. Vivace: A practical gauss-seidel method for stable soft body dynamics. *ACM Transactions on Graphics (TOG)*, 35(6):1–9, 2016. 102

[52] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros. ADD: analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics*, 39(6):1–15, Nov. 2020. ISSN 0730-0301, 1557-7368. doi: 10.1145/3414685.3417766. URL https://dl.acm.org/doi/10.1145/3414685.3417766. 19, 33, 38, 63, 64, 65, 67, 73

[53] M. Géradin, D. Rixen, M. Géradin, and D. Rixen. *Théorie des vibrations: application à la dynamique des structures*, volume 2. Masson Paris, 1993. 11

[54] J. A. Greenwood and D. Tabor. The friction of hard sliders on lubricated rubber: The importance of deformation losses. *Proceedings of the Physical Society*, 71(6):989–1001, jun 1958. doi:

10.1088/0370-1328/71/6/312. URL https://doi.org/10.1088/0370-1328/71/6/312. 22

[55] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. D. Lillo, and Y. Lanthony. Alicevision Meshroom: An open-source 3D reconstruction pipeline. In *Proceedings of the 12th ACM Multimedia Systems Conference - MMSys '21*. ACM Press, 2021. doi: 10.1145/3458305.3478443. 101

[56] Q. Guo, X. Han, C. Fu, T. Gast, R. Tamstorf, and J. Teran. A material point method for thin shells with frictional contact. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201346. URL https://doi.org/10.1145/3197517.3201346. 15

[57] B. GUPTA. 3 - Friction behavior of fibrous materials used in textiles. In B. S. Gupta, editor, *Friction in Textile Materials*, Woodhead Publishing Series in Textiles, pages 67 – 94. Woodhead Publishing, 2008. ISBN 978-1-85573-920-8. doi: 10.1533/9781845694722.1.67. URL http://www.sciencedirect.com/science/article/pii/B9781855739208500035. 22

[58] X. Han, T. F. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran. A hybrid material point method for frictional contact with diverse materials. *Proc. ACM Comput. Graph. Interact. Tech.*, 2(2), July 2019. doi: 10.1145/3340258. URL https://doi.org/10.1145/3340258. 15

[59] J. Hansen. *Measurement and estimation of material parameters of real garments*. PhD thesis, University of British Columbia, 2020. URL https://open.library.ubc.ca/collections/ubctheses/24/items/1.0394092. 99

[60] M. D. Hersey. The laws of lubrication of horizontal journal bearings. *Journal of the Washington Academy of Sciences*, 4(19):542–552, 1914. ISSN 00430439. URL http://www.jstor.org/stable/24520857. 20

[61] S. E. Hieber and P. Koumoutsakos. A lagrangian particle level set method. *Journal of Computational Physics*, 210(1):342–367, 2005. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2005.04.013.

URL
http://www.sciencedirect.com/science/article/pii/S0021999105002238. 16

[62] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik. Chainqueen: A real-time differentiable physical simulator for soft robotics. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 100

[63] T. J. R. Hughes and G. E. . P. A. Knovel, Academic. *The finite element method: linear static and dynamic finite element analysis*. Dover Publications, Mineola, NY, 2000. ISBN 9780486411811;0486411818;. 11

[64] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, J. Parent-Levesque, K. Xie, K. Erleben, L. Paull, F. Shkurti, D. Nowrouzezahrai, and S. Fidler. gradsim: Differentiable simulation for system identification and visuomotor control. *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=c_E8kFWfhp0. 100

[65] E. Ju and M. G. Choi. Estimating cloth simulation parameters from a static drape using neural networks. *IEEE Access*, 8:195113–195121, 2020. 99

[66] C. Kane, J. E. Marsden, M. Ortiz, and M. West. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *International Journal for Numerical Methods in Engineering*, 49(10):1295–1325, 2000. doi: https://doi.org/10.1002/1097-0207(20001210)49:10⟨1295::AID-NME993⟩3.0.CO;2-W. 6, 73

[67] D. M. Kaufman and D. K. Pai. Geometric numerical integration of inequality constrained, nonsmooth hamiltonian systems. *SIAM Journal on Scientific Computing*, 34(5):A2670–A2703, 2012. 6

[68] D. M. Kaufman, T. Edmunds, and D. K. Pai. Fast frictional dynamics for rigid bodies. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 946–956. ACM, 2005. 19

[69] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai. Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*, 27(5): 164:1–164:11, 2008. 14, 19, 24, 38, 39, 40, 41, 67, 73

[70] D. M. Kaufman, R. Tamstorf, B. Smith, J.-M. Aubry, and E. Grinspun. Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Transactions on Graphics (TOG)*, 33(4):123, 2014. 19, 58, 73

[71] P. Kaufmann, S. Martin, M. Botsch, and M. Gross. Flexible simulation of deformable models using discontinuous galerkin fem. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 105–115, Goslar, DEU, 2008. Eurographics Association. ISBN 9783905674101. 15

[72] C. T. Kelley. Numerical methods for nonlinear equations. *Acta Numerica*, 27:207–287, 2018. doi: 10.1017/S0962492917000113. 76

[73] P. Kenins. Influence of Fiber Type and Moisture on Measured Fabric-to-Skin Friction. *Textile Research Journal*, 64(12):722–728, Dec. 1994. ISSN 0040-5175, 1746-7748. doi: 10.1177/004051759406401204. URL http://journals.sagepub.com/doi/10.1177/004051759406401204. 22

[74] N. Kikuchi and J. T. Oden. *Contact problems in elasticity: a study of variational inequalities and finite element methods*, volume 8. siam, 1988. 17

[75] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto. Fixed-step friction simulation: from classical Coulomb model to modern continuous models. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1009–1016, Edmonton, Alta., Canada, 2005. IEEE. ISBN 978-0-7803-8912-0. doi: 10.1109/IROS.2005.1545579. URL http://ieeexplore.ieee.org/document/1545579/. 20, 67

[76] P. G. Kry and D. K. Pai. Continuous contact simulation for smooth surfaces. *ACM Transactions on Graphics*, 22(1):106–129, Jan. 2003. ISSN 0730-0301. doi: 10.1145/588272.588280. URL https://doi.org/10.1145/588272.588280. 16

[77] P. G. Kry and D. K. Pai. Interaction capture and synthesis. *ACM Transactions on Graphics*, 25(3):872–880, July 2006. ISSN 0730-0301. doi: 10.1145/1141911.1141969. URL https://doi.org/10.1145/1141911.1141969. 48

[78] E. Larionov, Y. Fan, and D. K. Pai. Frictional contact on smooth elastic solids. *ACM Trans. Graph.*, 40(2), Apr. 2021. ISSN 0730-0301. doi: 10.1145/3446663. URL https://doi.org/10.1145/3446663. vi, 62, 65, 79

[79] E. Larionov, M.-L. Eckert, K. Wolff, and T. Stuyck. Estimating cloth elasticity parameters using position-based simulation of compliant constrained dynamics, 2022. URL https://arxiv.org/abs/2212.08790. vi

[80] E. Larionov, A. Longva, U. M. Ascher, J. Bender, and D. K. Pai. Fully implicit frictional dynamics with soft constraints, 2022. URL https://arxiv.org/abs/2211.10618. vi

[81] R. Leine, D. Van Campen, A. De Kraker, and L. Van Den Steen. Stick-slip vibrations induced by alternate friction models. *Nonlinear dynamics*, 16(1):41–54, 1998. 20

[82] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008. 102

[83] J. Li, G. Daviet, R. Narain, F. Bertails-Descoubes, M. Overby, G. E. Brown, and L. Boissieux. An implicit frictional contact solver for adaptive cloth simulation. *ACM Transactions on Graphics (TOG)*, 37(4):52, 2018. 19, 26, 38, 41, 43, 58

[84] M. Li, Z. Ferguson, T. Schneider, T. Langlois, D. Zorin, D. Panozzo, C. Jiang, and D. M. Kaufman. Incremental potential contact: Intersection- and inversion-free large deformation dynamics. *ACM Transactions on Graphics*, 39(4), 2020. 11, 17, 19, 26, 28, 33, 38, 61, 62, 63, 64, 65, 66, 67, 73, 76, 79, 85, 87, 142

[85] Y. Li, T. Du, K. Wu, J. Xu, and W. Matusik. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Trans. Graph.*, mar 2022. ISSN 0730-0301. doi: 10.1145/3527660. URL https://doi.org/10.1145/3527660. Just Accepted. 118, 124

[86] J. Liang, M. Lin, and V. Koltun. Differentiable cloth simulation for inverse problems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/28f0b864598a1291557bed248a998d4e-Paper.pdf. 100

[87] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan. Fast simulation of mass-spring systems. *ACM Transactions on Graphics (TOG)*, 32 (6):1–7, 2013. 100

[88] T. Liu, S. Bouaziz, and L. Kavan. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Trans. Graph.*, 36(4), May 2017. ISSN 0730-0301. doi: 10.1145/3072959.2990496. URL https://doi.org/10.1145/3072959.2990496. 15

[89] Y. F. Liu, J. Li, Z. M. Zhang, X. H. Hu, and W. J. Zhang. Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system. *Mechanical Sciences*, 6(1): 15–28, 2015. doi: 10.5194/ms-6-15-2015. URL https://www.mech-sci.net/6/15/2015/. 33

[90] P. Lötstedt. Time-dependent contact problems in rigid body mechanics. In *Nondifferential and Variational Techniques in Optimization*, pages 103–110. Springer, 1982. 18

[91] C. Luible. *Study of mechanical properties in the simulation of 3D garments*. PhD thesis, University of Geneva, 2008. 99

[92] F. Löschner, A. Longva, S. Jeske, T. Kugelstadt, and J. Bender. Higher-Order Time Integration for Deformable Solids. *Computer Graphics Forum*, 39(8):157–169, Dec. 2020. ISSN 0167-7055, 1467-8659. doi: 10.1111/cgf.14110. URL https://onlinelibrary.wiley.com/doi/10.1111/cgf.14110. 11, 64

[93] R. M. Sánchez-Banderas and M. A. Otaduy. Strain rate dissipation for elastic deformations. *Computer Graphics Forum*, 37(8):161–170, 2018. doi: https://doi.org/10.1111/cgf.13521. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13521. 7

[94] M. Macklin and M. Muller. A constraint-based formulation of stable neo-hookean materials. In *Motion, Interaction and Games*, pages 1–7. 2021. 118

[95] M. Macklin, M. Müller, and N. Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016. 98, 100

[96] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and V. Makoviychuk. Non-smooth newton methods for deformable multi-body dynamics. *ACM Transactions on Graphics*, 38(5):1–20, Oct 2019. ISSN 0730-0301. doi: 10.1145/3338695. URL http://dx.doi.org/10.1145/3338695. 19, 58

[97] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and Z. Corse. Local optimization for robust signed distance field collision. *Proc. ACM Comput. Graph. Interact. Tech.*, 3(1), Apr. 2020. doi: 10.1145/3384538. URL https://doi.org/10.1145/3384538. 16, 58

[98] A. Madan and D. I. W. Levin. Fast evaluation of smooth distance constraints on co-dimensional geometry. *ACM Trans. Graph.*, 41(4), jul 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530093. URL https://doi.org/10.1145/3528223.3530093. 121

[99] F. Mandl. *Statistical physics*. Wiley, London;New York;, 1971. ISBN 0471566586;9780471566588;. 69

[100] N. D. Matsakis and F. S. Klock, II. The rust language. *Ada Lett.*, 34 (3):103–104, Oct. 2014. ISSN 1094-3641. doi: 10.1145/2692956.2663188. URL http://doi.acm.org/10.1145/2692956.2663188. 42

[101] H. Mazhar, T. Heyn, D. Negrut, and A. Tasora. Using nesterov's method to accelerate multibody dynamics with friction and contact. *ACM Trans. Graph.*, 34(3), May 2015. ISSN 0730-0301. doi: 10.1145/2735627. URL https://doi.org/10.1145/2735627. 19, 22

[102] A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. Efficient elasticity for character skinning with contact and collisions. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 1–12, Vancouver, British Columbia, Canada, July 2011. Association for Computing Machinery. ISBN 978-1-4503-0943-1. doi: 10.1145/1964921.1964932. URL https://doi.org/10.1145/1964921.1964932. 16, 28

[103] E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. Data-driven estimation of cloth simulation models. *Computer Graphics Forum*, 31(2pt2):519–528, 2012. doi: https://doi.org/10.1111/j.1467-8659.2012.03031.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2012.03031.x. 99, 106, 107

[104] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *ACM Siggraph Computer Graphics*, volume 22, pages 289–298. ACM, 1988. 17

[105] J. J. Moreau. *On Unilateral Constraints, Friction and Plasticity*, pages 171–322. Springer Berlin Heidelberg, Berlin, Heidelberg, 1973. ISBN 978-3-642-10960-7. doi: 10.1007/978-3-642-10960-7_7. URL https://doi.org/10.1007/978-3-642-10960-7_7. 13, 37

[106] T. Most and C. Bucher. 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. doi: 10.12989/sem.2005.21.3.315. 30

[107] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007. 15, 18, 100

[108] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85(3):67–94, 1959. doi: 10.1061/JMCEA3.0000098. URL https://ascelibrary.org/doi/abs/10.1061/JMCEA3.0000098. 11

[109] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer New York, New York, NY, second edition, 2006. ISBN 1431-8598. 24, 74

[110] A. C. Oeztireli, G. Guennebaud, and M. Gross. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. *Computer Graphics Forum*, 2009. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2009.01388.x. 17

[111] B. Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2019. URL http://www.blender.org. 43

[112] B. Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2021. URL http://www.blender.org. 85

[113] Optitex. Optitex Fabric Analyzer. https://optitex.com/products/fabric-management/, 2022. 99

[114] S. Osher, R. Fedkiw, and K. Piechor. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.*, 57(3):B15–B15, 2004. 16

[115] M. A. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. Implicit contact handling for deformable objects. *Computer Graphics Forum*, 28(2):559–568, 2009. doi: 10.1111/j.1467-8659.2009.01396.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01396.x. 41

[116] M. Overby, G. E. Brown, J. Li, and R. Narain. *IEEE Transactions on Visualization and Computer Graphics*, 23(10):2222–2234, 2017. 100

[117] P. Painlevé. Sur les loi du frottement de glissement. In *Comptes Rendu des Séances de l'Academie des Sciences*, pages 112–115, Paris, 1895. 14, 19

[118] T. Piatkowski. Dahl and lugre dynamic friction models—the analysis of selected properties. *Mechanism and Machine Theory*, 73:91–100, 2014. 20

[119] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin. Scalable differentiable physics for learning and control. In *ICML*, 2020. 100

[120] J. W. S. B. Rayleigh. *The theory of sound*, volume 2. Macmillan & Company, 1896. 7

[121] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009. 13

[122] C. Schillings and A. M. Stuart. Analysis of the ensemble kalman filter for inverse problems. *SIAM Journal on Numerical Analysis*, 55 (3):1264–1290, 2017. doi: 10.1137/16M105959X. URL https://doi.org/10.1137/16M105959X. 99

[123] S. H. Sheen, E. Larionov, and D. K. Pai. Volume preserving simulation of soft tissue with skin. *Proceedings of the ACM on computer graphics and interactive techniques*, 4(3):1–23, 2021. 65, 69

[124] C. Shen, J. F. O'Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 896–904, New York, NY, USA, 2004. ACM. doi: 10.1145/1186562.1015816. URL http://doi.acm.org/10.1145/1186562.1015816. 29

[125] SideFX. Houdini, 2019. 18, 42, 112

[126] E. Sifakis and J. Barbic. Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 20:1–20:50, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1678-1. doi: 10.1145/2343483.2343501. URL http://doi.acm.org/10.1145/2343483.2343501. 15, 23

[127] A. Signorini. Sopra alcune questioni di elastostatica. *Atti della Societa Italiana per il Progresso delle Scienze*, 21(II):143–148, 1933. 34

[128] B. Smith, F. D. Goes, and T. Kim. Stable neo-hookean flesh simulation. *ACM Trans. Graph.*, 37(2), mar 2018. ISSN 0730-0301. doi: 10.1145/3180491. URL https://doi.org/10.1145/3180491. 23, 65, 79

[129] J. Spillmann and M. Teschner. An adaptive contact model for the robust simulation of knots. In *Computer Graphics Forum*, volume 27, pages 497–506. Wiley Online Library, 2008. 18

[130] M. Strecke and J. Stückler. DiffSDFSim: Differentiable rigid-body dynamics with implicit shapes. In *International Conference on 3D Vision (3DV)*, Dec. 2021. 100

[131] R. Stribeck. *Kugellager für beliebige Belastungen (Ball bearings for any load)*. Buchdruckerei AW Schade, Berlin N., 1901. 20

[132] R. Stribeck. Die wesentlichen eigenschaften der gleit-und rollenlager (characteristics of plain and roller bearings). *Zeitschrift des Vereines Deutscher Ingenieure*, 46:1341–1348, 1902. 20

[133] T. Stuyck. Cloth simulation for computer graphics. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, 10(3):1–121, 2018. 100

[134] J. Swevers, F. Al-Bender, C. G. Ganseman, and T. Projogo. An integrated friction model structure with improved presliding behavior for accurate friction compensation. *IEEE Transactions on Automatic Control*, 45(4):675–686, 2000. 21

[135] G. Turk and J. F. O'Brien. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.*, 21(4):855—873, Oct. 2002. ISSN 0730-0301. doi: 10.1145/571647.571650. URL https://doi.org/10.1145/571647.571650. 16

[136] R. Vaillant, L. Barthe, G. Guennebaud, M.-P. Cani, D. Rohmer, B. Wyvill, O. Gourmel, and M. Paulin. Implicit skinning: Real-time skin deformation with contact modeling. *ACM Trans. Graph.*, 32(4), July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461960. URL https://doi.org/10.1145/2461912.2461960. 17

[137] R. Vaillant, G. Guennebaud, L. Barthe, B. Wyvill, and M.-P. Cani. Robust iso-surface tracking for interactive character skinning. *ACM Trans. Graph.*, 33(6):189:1–189:11, Nov. 2014. ISSN 0730-0301. doi: 10.1145/2661229.2661264. URL http://doi.acm.org/10.1145/2661229.2661264. 16, 17

[138] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, Mar. 1992. ISSN 0196-5204. doi: 10.1137/0913035. URL https://doi.org/10.1137/0913035. 75

[139] M. Verschoor and A. C. Jalba. Efficient and accurate collision response for elastically deformable models. *ACM Trans. Graph.*, 38 (2), Mar. 2019. ISSN 0730-0301. doi: 10.1145/3209887. URL https://doi.org/10.1145/3209887. 19

[140] B. Vrande, van de, D. Campen, van, and A. Kraker, de. Some aspects of the analysis of stick-slip vibrations with an application to drill strings. In *Proceedings of the ASME design engineering technical conference : DETC '97*, United States, 1997. American Society of Mechanical Engineers. ISBN 079181243X. 20

[141] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006. ISSN 1436-4646. doi: 10.1007/s10107-004-0559-y. URL https://doi.org/10.1007/s10107-004-0559-y. 36, 43

[142] H. Wang, J. F. O'Brien, and R. Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. In *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11*, page 1, Vancouver, British Columbia, Canada, 2011. ACM Press. ISBN 978-1-4503-0943-1. doi: 10.1145/1964921.1964966. URL http://portal.acm.org/citation.cfm?doid=1964921.1964966. 7

[143] H. Wang, J. F. O'Brien, and R. Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011. 99, 100, 106, 107, 117

[144] S. Weiss, R. Maier, D. Cremers, R. Westermann, and N. Thuerey. Correspondence-free material reconstruction using sparse surface constraints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020. 100

[145] J. Wojewoda, A. Stefański, M. Wiercigroch, and T. Kapitaniak. Hysteretic effects of dry friction: modelling and experimental studies. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1866):747–765, Mar. 2008. ISSN 1364-503X, 1471-2962. doi: 10.1098/rsta.2007.2125. URL https://royalsocietypublishing.org/doi/10.1098/rsta.2007.2125. 21, 62, 67

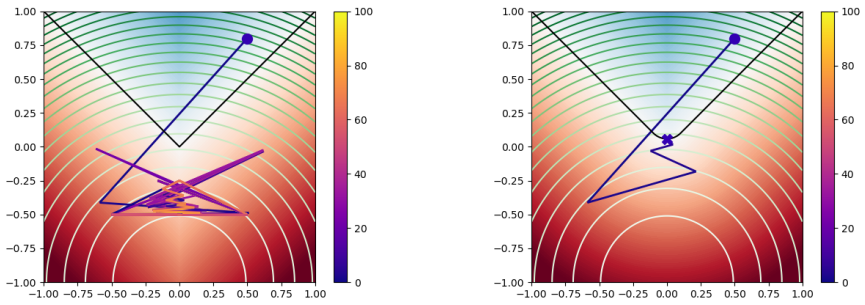
[146] S. Yang, J. Liang, and M. C. Lin. Learning-based cloth material recovery from video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4383–4393, 2017. 99

Appendix 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 3.4.

Consider a crude simplification of the contact problem in the context of polygonal contact, where a single vertex on a larger mesh at position $\mathbf{x}_0$ 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 A.1a we illustrate a scenario where a non-smooth implicit surface causes gradient-based optimizers to diverge. In Figure A.1b, 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 [84]. In our work, we use a smoothing of the entire surface using MLS, rather than employing local smoothing at polygon cusps. Note that this issue affects any line-search method, which uses the constraint Jacobian for computing the descent direction. In general, non-smoothness cannot be simply removed by splitting the contact polygonal constraints into multiple linear constraints,



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

Figure A.1: 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.

because the feasible region is non-convex.

Appendix 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 B.1. When the tension caused by the weight of the cloth at every contact point is included, we get

$$\frac{dT}{d\theta} = \mu T + \mu Rb\cos(\theta) - Rb\sin(\theta) \tag{B.1}$$

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 Eq. (B.1) 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 of the cloth at one end with $T(-\frac{\pi}{2})$, we will find the length of the hanging part to be

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

We can now plug in all known lengths and solve for μ. For the example in Figure 3.5d and assuming that $\ell_{-\pi/2}$ represents the shorter end, we have

$$\ell_{-\pi/2} = 0.8, \quad \ell = 4, \quad R = \frac{1.6}{\pi} \tag{B.3}$$

where all values are given in meters. Finally, the root of Eq. (B.2) given Eqs. (B.3) is

$$\mu \approx 0.14015161654962588...$$

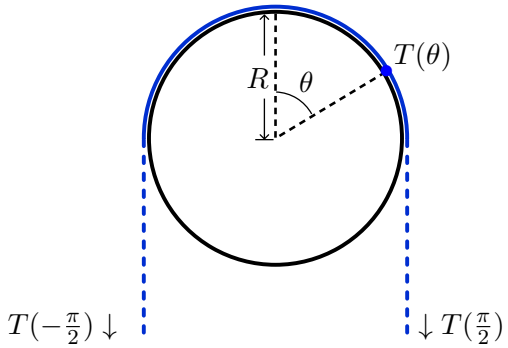


Figure B.1: *Cloth on cylinder.* This diagram shows a cross-section of our cloth on cylinder configuration for the capstan equation derived in Appendix B. The cloth (in blue) rests on the cylinder (in black), which has radius R. $T(\theta)$ is the tension on the cloth at each point on the top semi-circle.