UNIVERSITY OF CALIFORNIA
Los Angeles

A Material Point Method for Simulating Frictional Contact with Diverse Materials

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Xuchen Han

2020

ABSTRACT OF THE DISSERTATION

A Material Point Method for Simulating Frictional Contact with Diverse Materials

by

Xuchen Han
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2020
Professor Joseph M. Teran, Chair

We present an extension to the Material Point Method (MPM) for simulating elastic objects with various co-dimensions like hair (1D), thin shells (2D), and volumetric objects (3D). We simulate thin shells with frictional contact using a combination of MPM and subdivision finite elements. The shell kinematics are assumed to follow a continuum shell model which is decomposed into a Kirchhoff-Love motion that rotates the mid-surface normals followed by shearing and compression/extension of the material along the mid-surface normal. We use this decomposition to design an elastoplastic constitutive model to resolve frictional contact by decoupling resistance to contact and shearing from the bending resistance components of stress. We show that by resolving frictional contact with a continuum approach, our hybrid Lagrangian/Eulerian approach is capable of simulating challenging shell contact scenarios with hundreds of thousands to millions of degrees of freedom. Furthermore our technique naturally couples with other traditional MPM methods for simulating granular materials. Without the need for collision detection or resolution, our method runs in a few minutes per frame in these high resolution examples. For the simulation of hair and volumetric elastic objects, we utilize a Lagrangian mesh for internal force computation and an Eulerian mesh for self collision as well as coupling with external materials. While the updated Lagrangian discretization where the Eulerian grid degrees of freedom are used to take variations of the potential energy is effective in simulating thin shells, its frictional contact response strategy does not generalize to volumetric objects. Therefore, we develop a hybrid approach that retains Lagrangian degrees of freedom while still allowing for natural coupling with other

materials simulated with traditional MPM. We demonstrate the efficacy of our technique with examples that involve elastic soft tissues coupled with kinematic skeletons, extreme deformation, and coupling with multiple elastoplastic materials. Our approach also naturally allows for two-way rigid body coupling.

The dissertation of Xuchen Han is approved.

Jeffrey D. Eldredge

Christopher R. Anderson

Luminita Aura Vese

Joseph M. Teran, Committee Chair

University of California, Los Angeles

2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis advisor Professor Joseph Teran for his continued and dedicated support of my study and research at UCLA. This dissertation would not have been possible without his ingenious insights and patient guidance. I also thank Professor Luminita Vese, Professor Chris Anderson, and Professor Jeff Eldredge for serving on my doctoral committee.

I thank my coauthors for their hard work and advice and my colleagues at UCLA for many insightful discussions. Specifically, I want to thank Qi Guo for the countless hours we spent hunting down bugs together, Mengyuan Ding for leading the charge in the baking project, Ted Gast for sharing his expertise in computer science in general, Chuyuan Fu for her help in implementing subdivision surfaces, and Stephanie Wang for mentoring me in my first PhD project.

Special thanks go to Chuyuan Fu, Qi Guo, Siting Liu, Yuejiao Sun, Anran Lu and Pumpki Su for being there for me in various times of hardship and pain. Their companionship and encouragement gave me hope when all seemed lost.

Most of all, I thank my parents for their unconditional love and support, and for doing everything they can to take care of me from the other side of planet so I can focus on my research.

This dissertation contains content from previously published work. Chapter 5 from [19], Chapter 4 and 6 from [21].

VITA

2011–2015 B.A. (Mathematics, Economics), Northwestern University, USA.

2015–2019 Research Assistant, Mathematics Department, UCLA, USA.

2016–2020 Teaching Assistant, Mathematics Department, UCLA, USA.

2019–2020 AI Resident, X, the Moonshot Factory, Mountain View, USA.

PUBLICATIONS

M. Ding, X. Han, S. Wang, T. Gast, J. Teran, A Thermomechanical Material Point Method for Baking and Cooking, ACM Transactions on Graphics (SIGGRAPH ASIA 2019).

X. Han, T. Gast, S. Wang, Q. Guo, J. Teran, A Hybrid Material Point Method for Frictional Contact with Diverse Materials, Proceedings of the ACM on Computer Graphics and Interactive Techniques (SCA 2019).

Q. Guo, X. Han, C. Fu, T. Gast, R. Tamstorf, J. Teran, A Material Point Method for Thin Shells with Frictional Contact, ACM Transactions on Graphics (SIGGRAPH 2018).

M. Gao, A. Pradhana, X. Han, Q. Guo, G. Kot, E. Sifakis, C. Jiang, Animating Fluid Sediment Mixture in Particle-Laden Flows , ACM Transactions on Graphics (SIGGRAPH 2018).

CHAPTER 1

Introduction

1.1 Overview

The Material Point Method (MPM) [42] was developed as a generalization of the Particle-In-Cell (PIC/FLIP) [22, 6] method to elastoplastic materials, and like PIC/FLIP, it has proven to be a very effective tool for many computer graphics problems. Phenomena like fracture/topological change, multiple material interactions, and challenging self contact scenarios with complex geometric domains are all commonplace in computer graphics applications. MPM naturally handles many of these. This was first demonstrated for snow dynamics by Stomakhin et al. [39]. Since then a wide variety of other phenomena, particularly those that can be described as elastoplastic, have been simulated with MPM in graphics applications. This includes the dynamics of non-Newtonian fluids and foams [45, 35], melting [40, 16], porous media [43, 14], and frictional contact between granular materials [13, 29, 46]. MPM has also been used to simulate contact and collision with volumetric elastic objects [27, 48] and frictional contact between thin hyperelastic materials [25].

Despite the many advantages, there are drawbacks associated with the MPM collision resolution, especially when simulating Lagrangian meshes such as hair (1D), thin shell (2D) and volumetric meshes (3D) in the updated Lagrangian views. Our method aims to improve the simulation quality under these circumstances.

1.1.1 Hair

Simulating hair on a virtual character is one of the most challenging aspects of computer graphics. As hair is an integral part of creating many virtual characters, the problem is

especially important. Unfortunately, the massive number of hairs interacting and colliding makes this task especially challenging. Many approximations for simulating hair exist, but they typically fail to provide the amount of detail that real hair exhibits. Our method is closely related to the hybrid methods proposed in by Jiang et al. [25] and McAdams et al. [33]. McAdams et al. [33] use a hybrid PIC/geometric impulse technique to resolve self collision of many thin straight hairs. They assume that hair is incompressible and interpret the PIC grid projection as a Lagrangian repulsion. They then apply the collision impulses of Bridson et al. [7] to catch cases not resolved on the grid. However, the incompressible solve is often expensive and is the bottleneck of the simulation. Furthermore, when the simulated hair is under intense collision (e.g. when forming a braid), the Lagrangian repulsion from the incompressible solve is often not enough, and would therefore put heavy burden on the geometric collision solve, which renders the simulation time intractable. Jiang et al. [25] treat the hair fibers in a continuum view which naturally resolves collisions and contact. Unfortunately, the continuum view does not hold in many situations, especially when the hair strands are in a disarray, causing severe numerical tangling. In addition, the constitutive model used in Jiang et al. does not support curly hair, which limits the versatility of the method. Our method builds on top of these ideas and decomposes the deformation of the hair strands into the deformation of the individual strands the deformation associated with frictional contact interactions among strands to achieve detailed simulations of straight or curved hair strands in large bundles with high efficiency.

1.1.2 Thin shell

Simulation of thin elastic surfaces with bending resistance and frictional contact is essential in many domains including visual effects, textile simulation, as well as engineering applications that involve thin metallic sheets. The most common computer graphics techniques for simulating these materials typically use mass/spring models or continuum elastic formulations over linear strain triangles. The popularity of these approaches is primarily due to their relative simplicity, computational efficiency and the favorable performance of contact/collision techniques. However, these techniques have well known limitations that include resolution

dependent behavior like mesh-based anisotropy and lack of convergence under refinement for mass/spring models. Furthermore, constitutive models for bending resistance can be difficult to discretize over triangle meshes since higher order derivatives needed for curvature calculations are not easily approximated with linear interpolation. In contrast, shell models commonly used in the engineering literature provide natural and accurate control of the constitutive behavior caused by bending and converge under refinement due to their continuum foundations. However, these models typically require H^2 regularity that is difficult to achieve with traditional interpolation. The H^2 regularity requirement means that interpolating functions and all their derivatives of order less than or equal to two are square integrable. In practice this means that the interpolating functions must also have continuous first derivatives (C^1 continuous). Non-uniform rational B-splines (NURBS) [31] and subdivision (subd) surface interpolation [10] yield the required regularity, but these more elaborate interpolation strategies can complicate collision detection and resolution. Thus with existing methods the choice is between on one hand using linear triangles and getting natural resolution of collisions and contact but having limited constitutive control and on the other hand using shell-based techniques with better constitutive control at the expense of more complicated collision and contact.

Jiang et al.[25] proposed a hybrid Lagrangian/Eulerian collision formulation technique that naturally resolves contact through an elastoplastic view of frictional contact with thin membranes. They showed that by letting the elastoplastic constitutive model resolve frictional contact through a continuum view, there is no need for collision detection or resolution via impulses [7, 37, 23, 44] or linear complimentary formulations of the constrained dynamics [34]. This greatly accelerates the performance since these routines are often a bottleneck. However, their approach was designed for thin membranes with no resistance to bending. It is possible to simply add bending springs to their approach, however that violates the key assumptions in their elastoplastic characterization of frictional contact. Moreover, when visibly significant resistance to bending is required (e.g. for a stiff leather or denim garment etc), artifacts are apparent. We show that their formulation can be generalized to shell model kinematics and that this in turn allows for the simulation of much larger ranges of bend-

ing stiffness without the artifacts inherent in ignoring bending related stress in elastoplastic continuum model.

1.1.3 Volumetric object

In a MPM simulation, information is typically lost when transferring from particles to grid, since there are generally many more particles than grid nodes [27, 15, 20]. Furthermore, volumetric elastic materials suffer from two additional drawbacks. First, while contact for materials such as grains [29, 13], membranes/shells and fibers [25] can be envisioned as a continuum process where elastoplasticity associated with frictional contact is defined by the directions orthogonal to the grain, curve or surface, volumetric objects have no non-elastic directions for which to apply the condition. Hence, all self-collision resolution will result from volumetric elasticity, which means that frictional sliding cannot be regulated in a Coulomb fashion via plasticity. The second drawback is that the Eulerian grid spacing must be approximately the same as the edge lengths in the volumetric Lagrangian mesh. If the Eulerian grid resolution is significantly lower, there is non-negligible information loss in the transfer from particles to grid, and there will be spurious interaction at a distance. If the grid resolution is significantly higher, collisions will not be resolved (see Figure 6.2). This is problematic because visual separation between elastic bodies is proportionate to the Eulerian grid spacing, which therefore mandates high spatial resolution of the volumetric Lagrangian mesh to reduce separation thickness. This problem is not present when simulating cloth and hair because they admit the use of elastoplasticity frictional contact particles [25, 19] and arbitrarily many can be added on each surface element or hair segment to accommodate high spatial grid resolution.

We design a novel hybrid Lagrangian Material Point Method to alleviate these drawbacks. Our approach utilizes more of the Lagrangian degrees of freedom to minimize artifacts while retaining aspects of MPM that allow for collision resolution without suffering from information loss when going from particles to grid. Our approach also resolves the Eulerian grid size (and artificial separation distance) limitations associated with volumetric elasticity, allowing

for Coulomb frictional contact with volumetric elastic meshes. We support coupling with materials simulated with standard MPM discretizations and we provide for simple two-way coupling with rigid bodies.

1.2 Contributions

This dissertation makes the following key contributions:

- An efficient method to simulate large bundles of hair strands with robust collision handling.
- A strain splitting technique to separate thin shell motion into Kirchhoff-Love and continuum shell components along with the resolution of frictional contact among thin shell objects.
- Novel collision impulses defined from the MPM particle to grid transfers that resolve frictional contact when simulating volumetric meshed objects.
- A unified solver for purely incompressible fluids and particle-laden flows using MPM background grids.
- A three species porous mixture model of batter and dough with water, CO2 and solid contents and a novel MPM discretization of the three species mixture.

1.3 Dissertation overview

The dissertation is structured as follows:

Chapter 2 briefly reviews the mathematical background of continuum mechanics and fluid mechanics. We review the notation and concepts such as deformation gradient, elastic stress and the incompressible Euler equation that are key to our method.

Chapter 3 reviews the standard Material Point Method (MPM) with elastoplasticity. The method presented in this dissertation improves upon the standard MPM. We assume that

readers have basic knowledge about continuum and fluid mechanics and MPM.

Chapter 4 presents an augmented Lagrangian framework for simulating large assembly of hair strands. We decompose the motion of the hair into individual strand deformation and the deformation induced by frictional contact to capture the intricate details of the individual hair strands while robustly handling the numerous collisions among the strands.

Chapter 5 presents an novel elastoplastic formulation for thin shells, where a strain splitting technique is used to separate the motion into Kirchhoff-Love and continuum shell components in order to decouple the stress induced from resistance to bending and denting of thin shells and that from frictional contact. The strain splitting also allows natural formulation of denting and wrinkling behaviors as plastic deformation in the Kirchhoff-Love component of the motion.

Chapter 6 presents a hybrid Lagrangian/MPM method that retains Lagrangian degrees of freedom while still allowing for natural coupling with other materials simulated with traditional MPM like sand and snow as well as rigid objects. Furthermore, we extend the resolution of frictional contact with 1D hair strand simulation in Chapter 4 and 2D thin shell simulation in Chapter 5 and generalize it to 3D volumetric materials.

CHAPTER 2

Mathematical Background

2.1 Notation

We use bold face (e.g. $\mathbf{v}$) to denote vector and tensor quantities and plain text (e.g. v) to denote scalar quantities. We use superscript T (e.g. $\mathbf{F}^T$) to denote the transpose of a vector or a tensor. Unless otherwise stated, we use the summation convention for repeated indices. $|\cdot|$ is used to denote the L_2 norm of a vector.

2.2 Kinematics and Deformation Gradient

At any given time t, we consider the motion of material to be determined by a mapping $\boldsymbol{\phi}(\cdot, t) : \Omega_0 \to \Omega_t$ for $\Omega_0, \Omega_t \subset \mathbb{R}^d$ where $d = 2$ or 3 is the dimension of the domain. The mapping $\boldsymbol{\phi}$ is called the flow map. Points in the set Ω_0 are referred to as material points and are denoted as $\mathbf{X}$. Points in Ω_t, referred to as $\mathbf{x}$, represent the location of material points at time t. Hence, we have

$$\mathbf{x} = \mathbf{x}(t) = \boldsymbol{\phi}(\mathbf{X}, t) \tag{2.1}$$

Taking the first and the second derivatives of the flow map gives the velocity and acceleration at each material point:

$$\mathbf{V}(t) = \frac{\partial \boldsymbol{\phi}(\mathbf{X}, t)}{\partial t} \tag{2.2}$$

$$\mathbf{A}(t) = \frac{\partial \mathbf{V}(\mathbf{X}, t)}{\partial t} \tag{2.3}$$

$$\tag{2.4}$$

There are occasions, especially in fluid simulations, where we take the "Eulerian view" and require these quantities in the space Ω_t. We can get them by using the push forward formula:

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{V}(\boldsymbol{\phi}^{-1}(\mathbf{x}, t), t), \tag{2.5}$$

$$\mathbf{a}(\mathbf{x}, t) = \mathbf{A}(\boldsymbol{\phi}^{-1}(\mathbf{x}, t), t). \tag{2.6}$$

Conversely, if we are given these quantities in the"Eulerian view" and require them in the space Ω_0, or the "Lagrangian view", we can apply the pull back formula:

$$\mathbf{V}(\mathbf{X}, t) = \mathbf{v}(\boldsymbol{\phi}(\mathbf{X}, t), t), \tag{2.7}$$

$$\mathbf{A}(\mathbf{X}, t) = \mathbf{a}(\boldsymbol{\phi}(\mathbf{X}, t), t). \tag{2.8}$$

For a general Eulerian function $f(\cdot, t) : \Omega_t \to \mathbb{R}$, we use the notation $\frac{D}{Dt}$ to denote the material derivative:

$$\frac{D}{Dt} f(\mathbf{x}, t) = \frac{\partial f}{\partial t}(\mathbf{x}, t) + \frac{\partial f}{\partial x_j}(\mathbf{x}, t) v_j(\mathbf{x}, t) \tag{2.9}$$

Note that $\frac{D}{Dt} f(\mathbf{x}, t)$ is the push forward of $\frac{\partial}{\partial t} F(\mathbf{X}, t)$ where F is a Lagrangian function that is the pull back of f.

The Jacobian of this flow map, $\mathbf{F} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}}(\mathbf{X}, t)$ is referred to as the deformation gradient, and it represents the local deformation of the material. That is, the deformation gradient yields the best local linear approximation to flow map near $\mathbf{X}$. The determinant of $\mathbf{F}$, $J = \det(\mathbf{F})$, measures the local volume change of the material, with $J > 1$ indicating volume gain and $J < 1$ indicating volume loss. When $J < 0$, the material locally inverts, which does not happen in real life but may happen in a numerical simulation.

2.3 Hyperelasticity

As we often work in material space for elastic solids, the strain stress relationship is most naturally expressed using deformation gradient and first Piola-Kirchoff stress. Many materials we simulate are hyperelastic materials, whose first Piola-Kirchoff stress $\mathbf{P}$ can be derived

from an strain energy density function $\Psi(\mathbf{F})$ via

$$\mathbf{P}(\mathbf{X}, t) = \frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{X}, t) \tag{2.10}$$

Intuitively, the energy density function produces stress to penalize nonrigid deformation. The first Piola-Kirchoff stress $\mathbf{P}$ can be related to the Cauchy stress $\boldsymbol{\sigma}$ more commonly used in the engineering literature via

$$\boldsymbol{\sigma} = \frac{1}{J}\mathbf{P}\mathbf{F}^T \tag{2.11}$$

We introduce concrete constitutive models in later chapters as we need them.

2.4 Plasticity

When simulating elastoplastic materials, we often multiplicatively decompose the deformation gradient into two parts $\mathbf{F} = \mathbf{F}^E\mathbf{F}^P$, where $\mathbf{F}^P$ is the local plastic deformation of the material that is forgotten, and $\mathbf{F}^E$ is the local elastic deformation of the material penalized by the elastic stress. Hence, the strain stress relationship in equation A.18 and 2.11 is modified to

$$\mathbf{P}(\mathbf{X}, t) = \frac{\partial \Psi}{\partial \mathbf{F}^E}(\mathbf{X}, t) \tag{2.12}$$

and

$$\boldsymbol{\sigma} = \frac{1}{J}\mathbf{P}\mathbf{F}^{E^T} \tag{2.13}$$

when the material at hand is elastoplastic. The evolution of the decomposition of the deformation gradient into $\mathbf{F}^P$ and $\mathbf{F}^E$ is called the plastic flow and will be described in later chapters as needed.

2.5 Governing Equations

The two essential governing equations used in our numerical simulation are the conservation of mass and the conservation of momentum. We follow the notation in [17] and list them

below. The Lagrangian view of the conservation of mass and momentum are:

$$R(\mathbf{X},t)J(\mathbf{X},t) = R(\mathbf{X},0), \tag{2.14}$$

$$R(\mathbf{X},0)\frac{\partial \mathbf{V}(\mathbf{X},t)}{\partial t} = \nabla \cdot \mathbf{P}(\mathbf{X},t) + R(\mathbf{X},0)\mathbf{g}, \tag{2.15}$$

where $R(\mathbf{X},0)$ is the mass density at point $\mathbf{X} \in \Omega_0$, $J(\mathbf{X},t) = \det(\mathbf{F}(\mathbf{X},t))$, and $\mathbf{g}$ is the gravitational force density.

In the Eulerian view, the governing equations are

$$\frac{D}{Dt}\rho(\mathbf{x},t) + \rho(\mathbf{x},t)\nabla \cdot \mathbf{v}(\mathbf{x},t) = 0 \tag{2.16}$$

$$\rho(\mathbf{x},t)\frac{D}{Dt}\mathbf{v}(\mathbf{x},t) = \nabla \cdot \boldsymbol{\sigma} + \rho(\mathbf{x},t)\mathbf{g} \tag{2.17}$$

where ρ is the push forward of R.

When simulating incompressible and inviscid fluids, we often switch the notation of the velocity from $\mathbf{v}$ to $\mathbf{u}$, and the conservation of momentum in Eulerian view becomes

$$\rho\frac{D}{Dt}\mathbf{u} = -\nabla p + \mathbf{g}, \tag{2.18}$$

that expands into

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla p + \mathbf{g}, \tag{2.19}$$

where p is the pressure in the fluid. Substituting in the conservation of mass, we get

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \mathbf{u}\nabla \cdot (\rho\mathbf{u}) + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mathbf{g}, \tag{2.20}$$

that simplifies to

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \mathbf{g}, \tag{2.21}$$

where we use $\otimes$ to denote outer product of two tensors. Equation 2.21 is often referred to as the incompressible Euler equation.

CHAPTER 3

The Material Point Method

In this chapter, we lay out the essential steps in a typical Material Point Method (MPM) simulation. We assume knowledge of these typical steps in later chapters and only highlight the aspects in our method that deviates from the standard MPM algorithm.

3.1 Notation

We follow the convention from the previous chapter that vector and tensor quantities are denoted with bold face (e.g. $\mathbf{v}$) and scalar quantities are denoted with plain text (e.g. m).

In a discrete setting, many quantities are indexed with subscripts, which indicate where quantities are stored. Quantities that are stored at grid nodes are indexed with $\mathbf{i}$ (notice the bold face, indicating the index is a vector, as the grid is often 2D or 3D) and particle quantities have the index p.

With discrete time steps, we use superscript n to indicate quantities at the beginning of a time step, before forces are applied, (e.g., $m_{\mathbf{i}}^n$, $\mathbf{v}_p^n$), and we use superscript $n+1$ to denote quantities after momentum update (e.g., $\mathbf{x}_{\mathbf{i}}^{n+1}$, $\mathbf{C}_p^{n+1}$). Intermediate quantities are usually denoted with a hat (e.g., $\hat{\mathbf{x}}_{\mathbf{i}}^{n+1}$).

3.2 Method Outline

MPM is a hybrid Lagrangian/Eulerian approach. The primary representation of material for MPM is the Lagrangian states. At time t_n, we store the particle position $\mathbf{x}_p^n$, velocity $\mathbf{v}_p^n$, mass m_p, initial volume V_p^0, deformation gradient $\mathbf{F}_p^n$, affine velocity field $\mathbf{C}_p^n$ as in [27] for

all materials in the simulation. The Eulerian grid can be viewed as an auxiliary structure for updating the Lagrangian states. We first transfer the particle mass m_p and momentum $(m\mathbf{v})_p$ to their grid counterparts $m_\mathbf{i}$, $(m\mathbf{v})_\mathbf{i}$ in a way that conserves mass, linear and angular momentum (Section 3.3). Then we view the positions of the Eulerian grid nodes as the actual degrees of freedom in the system to compute the forces on the nodes and update momentum on the grid (Section 3.4). When simulating incompressible fluids, we solve the Poisson equation on the velocity grid to enforce the discrete divergence free condition. The motion of the grid is then interpolated back to the particles to update the Lagrangian state without moving the grid nodes (Section 3.5). After obtaining the new position of the particles, we update the deformation gradient of the particle $\mathbf{F}_p$. If the material is elastoplastic, we further compute the trial elastic deformation gradient, $\mathbf{F}_p^{E,\text{tr}}$ assuming no plasticity over the time step and then project the trial deformation gradient outside of the yield surface back to the yield surface as dictated by the return mapping of the plastic model (Section 3.6).

3.3 Grid Transfers: Particle to Grid

To update the Lagrangian state, we transfer mass and momentum from particles $\mathbf{x}_p^n$ to the grid nodes $\mathbf{x}_\mathbf{i}$ using Affine Particle In Cell (APIC) method [27]:

$$m_\mathbf{i}^n = \sum_p w_{\mathbf{i}p}^n m_p \tag{3.1}$$

$$\mathbf{v}_\mathbf{i}^n = \frac{1}{m_\mathbf{i}^n} \sum_p w_{\mathbf{i}p}^n m_p (\mathbf{v}_p^n + \mathbf{C}_p^n(\mathbf{x}_\mathbf{i}^n - \mathbf{x}_p^n)). \tag{3.2}$$

Here, $w_{\mathbf{i}p}^n = N(\mathbf{x}_p^n - \mathbf{x}_\mathbf{i})$ is the weight of the interaction between particle $\mathbf{x}_p^n$ and grid node $\mathbf{x}_\mathbf{i}$. $N(\mathbf{x})$ is the dyadic products of one-dimensional interpolation functions:

$$N(\mathbf{x}_p) = N\left(\frac{x_p}{\Delta x}\right) N\left(\frac{y_p}{\Delta x}\right) N\left(\frac{z_p}{\Delta x}\right), \tag{3.3}$$

where Δx is the grid spacing and $\mathbf{x}_p = (x_p, y_p, z_p)$. We usually choose quadratic B-spline kernels

$$N(x) = \begin{cases} \frac{3}{4} - |x|^2 & 0 \leq |x| < \frac{1}{2} \\ \frac{1}{2}\left(\frac{3}{2} - |x|\right)^2 & \frac{1}{2} \leq |x| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |x| \end{cases} \tag{3.4}$$

as the one-dimensional interpolation function as it conserves mass and linear momentum [26].

Linear interpolation functions do not produce continuous derivatives and thus create cross-cell instability when the derivative of the interpolation function is required in the computation. We only use it when simulating fluid, as derivatives of interpolation function are not required.

$$N(x) = \begin{cases} 1 - |x| & 0 \leq |x| < 1 \\ 0 & 1 \leq |x| \end{cases} \tag{3.5}$$

Computation of the interpolation function gradient is done similarly by differentiating the one dimensional functions:

$$\nabla N(\mathbf{x}_p) = \begin{pmatrix} \frac{1}{\Delta x} N'\left(\frac{x_p}{\Delta x}\right) N\left(\frac{y_p}{\Delta x}\right) N\left(\frac{z_p}{h}\right) \\ \frac{1}{\Delta x} N\left(\frac{x_p}{\Delta x}\right) N'\left(\frac{y_p}{\Delta x}\right) N\left(\frac{z_p}{h}\right) \\ \frac{1}{\Delta x} N\left(\frac{x_p}{\Delta x}\right) N\left(\frac{y_p}{\Delta x}\right) N'\left(\frac{z_p}{h}\right) \end{pmatrix}$$

where $N'(x)$ is the derivative of $N(x)$.

In Equation 3.2, $\mathbf{v}_p^n$ and $\mathbf{C}_p^n$ can be thought of as defining an affine velocity field local the particle p at position $\mathbf{x}_p^n$ as in [27]:

$$\mathbf{v}^n(\hat{\mathbf{x}}) = \mathbf{C}_p^n(\hat{\mathbf{x}} - \mathbf{x}_p^n) + \mathbf{v}_p^n \tag{3.6}$$

3.4 Grid Momentum Update

The grid momentum update uses the updated Lagrangian view of the governing physics [1, 15]. The grid at time t^n, after transferring state from the Lagrangian particles, is an

alternative Lagrangian mesh with degrees of freedom $\mathbf{x_i}$, $\mathbf{v_i^n}$ and mass $m_\mathbf{i}^n$. Its update is derived from the Lagrangian FEM discretization of a problem with a notion of potential energy. The internal force is the negative gradient of the potential energy with respect to positional changes. Using $\mathbf{x_i^{n+1}}$ and $\mathbf{p_i^{n+1}}$ to denote the new position and linear momentum states after the time step and $\mathbf{f_i}$ to denote the force acting on the grid node $\mathbf{i}$, the grid discretization has the form

$$\mathbf{x_i^{n+1}} = \mathbf{x_i} + \frac{\Delta t}{m_\mathbf{i}^n}\mathbf{p_i^{n+1}} \tag{3.7}$$

$$\mathbf{p_i^{n+1}} = m_\mathbf{i}^n\mathbf{v_i^n} + \Delta t\mathbf{f_i} + \Delta t m_\mathbf{i}^n\mathbf{g} \tag{3.8}$$

$$= m_\mathbf{i}^n\mathbf{v_i^n} - \Delta t\frac{\partial \Psi}{\partial \mathbf{x_i}}(\mathbf{x}^*) + \Delta t m_\mathbf{i}^n\mathbf{g} \tag{3.9}$$

where $\Psi(\mathbf{x})$ is the potential energy which depends on the positional states where we use $\mathbf{x}^* = \left(\mathbf{x}^*_{\mathbf{i}^1}, \mathbf{x}^*_{\mathbf{i}^2}, \ldots\right)^T$ to denote the vector of all grid node positions. In the case of symplectic Euler integration, $\mathbf{x_i^*} = \mathbf{x_i}$ and in the case of backward Euler, $\mathbf{x_i^*} = \mathbf{x_i^{n+1}}$. We note that the grid nodes are not actually moved from $\mathbf{x_i}$ to $\mathbf{x_i^{n+1}}$. Instead, the motion of the grid is interpolated to the particles (see Section 3.5). For simulating solids, the temporary new grid node velocity $\hat{\mathbf{v}}_\mathbf{i}^{n+1}$ is set using

$$\hat{\mathbf{v}}_\mathbf{i}^{n+1} = \frac{\mathbf{p_i^{n+1}}}{m_\mathbf{i}^n}. \tag{3.10}$$

For simulating incompressible fluids, we define an intermediate velocity $\mathbf{v_i^*} = \frac{\mathbf{p_i^{n+1}}}{m_\mathbf{i}^n}$ and take an additional pressure projection step to ensure numerical incompressibility. The standard pressure solve and projection [9] can be described by

$$\frac{\Delta t}{\rho}\nabla \cdot (\nabla p) = -\nabla \cdot \mathbf{v_i^*} \tag{3.11}$$

$$\frac{\hat{\mathbf{v}}_\mathbf{i}^{n+1} - \mathbf{v_i^*}}{\Delta t} = -\frac{1}{\rho}\nabla p \tag{3.12}$$

Pressures p are stored at cell centers, while all velocities are stored at cell nodes and are transferred between MPM particles and the background grid via the standard multi-linear kernel described in Equation 3.5 with the angular momentum conserving APIC [27] for stability and low dissipation. We further adopt the RPIC damping as described in [26] as a

controllable artificial viscosity for changing liquid behavior. The left-hand side of Equation 3.11 can be discretized with a standard 7-point (5-point in 2D) Laplacian stencil. For the right-hand side, we compute the divergence of velocity in the semi-staggered discretization by first computing the partial derivatives on the edges and then average them to the cell center. It is well known that this semi-staggered discretization causes spurious hourglass velocity modes which cannot be eliminated by the pressure projection. We mitigate this problem by using an hourglass damping as proposed by [47]. Then in solving Equation 3.12, the pressure gradients are naturally calculated on the edges and are then averaged to the grid nodes.

After obtaining the temporary new grid velocity $\hat{\mathbf{v}}_{\mathbf{i}}^{n+1}$, we take the boundary conditions or collision objects into account as in [30] to obtain the final new grid velocity $\mathbf{v}_{\mathbf{i}}^{n+1}$. In the case of explicit integration, each nodal velocity can be independently set to the desired value due to Dirichlet boundary conditions or rigid object collisions.

3.5 Grid Transfers: Grid to Particle

The grid to particle transfer defines the time t^{n+1} affine velocity local to particle $\mathbf{x}_p^n$ in terms of $\mathbf{v}_p^{n+1}$ and $\mathbf{C}_p^{n+1}$ from

$$\mathbf{v}_p^{n+1} = \sum_{\mathbf{i}} w_{\mathbf{i}p}^n \mathbf{v}_{\mathbf{i}}^{n+1} \tag{3.13}$$

$$\tilde{\mathbf{C}}_p^{n+1} = \frac{12}{\Delta x^2 (d+1)} \sum_{\mathbf{i}} w_{\mathbf{i}p}^n \mathbf{v}_{\mathbf{i}}^{n+1} \otimes (\mathbf{x}_{\mathbf{i}}^n - \mathbf{x}_p^n) \tag{3.14}$$

$$\mathbf{C}_p^{n+1} = (1-\nu)\,\tilde{\mathbf{C}}_p^{n+1} + \frac{\nu}{2}\left(\tilde{\mathbf{C}}_p^{n+1} - \tilde{\mathbf{C}}_p^{n+1^T}\right) \tag{3.15}$$

Here d is the B-spline degree ($d = 3$ for cubic B-spline interpolation, $d = 2$ for quadratic B-spline interpolation) and Δx is the Eulerian grid spacing. ν is the explicit damping coefficient [25] where $\nu = 0$ is completely undamped and $\frac{1}{2}\left(\tilde{\mathbf{C}}_p^{n+1} - \tilde{\mathbf{C}}_p^{n+1^T}\right)$ is the RPIC transfer from [27]. Jiang and co-authors showed that this transfer scheme conserves mass, linear and angular momentum in [26].

3.6 Plasticity

After obtaining the new particle velocity $\mathbf{v}_p^{n+1}$, we update the particle position as

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} \tag{3.16}$$

At the same time, we also update the deformation gradient as

$$\mathbf{F}_p^{n+1} = \left(\mathbf{I} + \Delta t \sum_{\mathbf{i}} \mathbf{v}_{\mathbf{i}}^{n+1} (\nabla w_{\mathbf{i}p}^n)^T \right) \mathbf{F}_p^n, \tag{3.17}$$

For purely elastic material, $\mathbf{F}_p^{E,n+1} = \mathbf{F}_p^{n+1}$, and the entire deformation will be used in the stress calculation in the next time step. For elastoplastic materials, we assume that there is no plastic flow over the time step, and calculate the trial elastic deformation of the material in the next time step as

$$\mathbf{F}_p^{E,\mathrm{tr}} = \mathbf{F}_p^{n+1} \left(\mathbf{F}_p^{P,n} \right)^{-1}. \tag{3.18}$$

We then check whether the stress generated by $\mathbf{F}_p^{E,\mathrm{tr}}$ lies within the yield surface of the plasticity model. If not, we project the stress back to the yield surface according to the return mapping dictated by the plastic model, and calculate the new elastic deformation gradient $\mathbf{F}_p^{E,n+1}$ associated with the projected stress. The product of the projected elastic and plastic deformation gradients must be equal to the original deformation gradient:

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^{E,\mathrm{tr}} \mathbf{F}_p^{P,n} = \mathbf{F}_p^{E,n+1} \mathbf{F}_p^{P,n+1} \tag{3.19}$$

Therefore, we find the new plastic deformation gradient as

$$\mathbf{F}_p^{P,n+1} = \left(\mathbf{F}_p^{E,n+1} \right)^{-1} \mathbf{F}_p^{n+1} \tag{3.20}$$

We will introduce the various plastic models we use in the following chapters as needed.

CHAPTER 4

Hybrid MPM Hair Strands

As we point out in Chapter 1, although MPM proved to be an immense success in its computer graphics application with its effortless collision detection and handling, there are still drawbacks associated with it. Several researches [27, 15, 20] have shown that information is typically lost when transferring from particles to grid, since there are generally many more particles than grid nodes. Even when utilizing Lagrangian meshes in the updated Lagrangian view as in [27, 25, 19, 48], information is still lost, which can lead to persistent wrinkles and apparent interaction at a distance, as discussed in [25, 19]. Another common artifact caused by this information loss is excessive numerical friction and cohesion. In Figure 4.1 (top left), we show that a ball of hair simulated with the technique in [25] suffers from such artifacts. In this chapter, we present a hybrid elastoplastic model for hair and strand self collision that supports bending, torsion and stretching resistance and that does not suffer from information loss in particle to grid transfers. With our method, we are able to remove the excessive cohesion and regulate friction with Coulomb friction model. As a result, we are able to capture the rich dynamics of individual strands as in Figure 4.1 (top right).

4.1 Continuous formulation

We follow the codimensional approaches of [25, 19] and penalize frictional contact between hairs and thin strands using a continuum assumption. Following their formulation, we decompose the deformation of the material $\boldsymbol{\phi}$ into the deformation of the individual strands $\boldsymbol{\phi}^s$ and the deformation associated with frictional contact interactions among strands $\boldsymbol{\phi}^d$,

namely

$$\boldsymbol{\phi} = \boldsymbol{\phi}^d \circ \boldsymbol{\phi}^s. \tag{4.1}$$

Consequently, the deformation gradient is decomposed into $\mathbf{F} = \mathbf{F}^d\mathbf{F}^s$. We treat the deformation of the strand $\mathbf{F}^s$ as purely elastic using standard rod and curve models [2, 3, 4, 33], and decompose $\mathbf{F}^d$ into elastic and plastic components,

$$\mathbf{F}^d = \mathbf{F}^{d,E}\mathbf{F}^{d,P} \tag{4.2}$$

to handle frictional contact among hair strands.

We utilize the continuum Coulomb friction view from [29, 25, 19] to place a constraint on admissible stress. Shear stresses resisting sliding motions between strands cannot be larger than a frictional constant times the normal stress holding them together. When the shear stress exceeds that threshold, the strands will start to slide against each other, inducing plastic deformation. Mathematically, the Coulomb friction model states that $\mathbf{s}^T\boldsymbol{\sigma}\mathbf{n} + c_F\mathbf{n}^T\boldsymbol{\sigma}\mathbf{n} \leq 0$, where $\mathbf{n}$ is the normal to the contact surface, $\mathbf{s}$ is any unit vector along the contact surface, and c_F is the friction coefficient. While Jiang et al. [25] considers only directions $\mathbf{n}$ orthogonal to the tangent of the midline of the strand, we enforce this condition for all directions. The continuum assumption in Jiang et al. [25] is that of a tube of parallel strands, which holds well for simulating knits but is less effective in the more complicated contact scenarios that occur when simulating hair and thin strands. To accommodate this more general constraint, we use an isotropic potential to resist collision, rather than the transversely isotropic potential of Jiang et al. [25].

With this convention, we define the potential energy as a combination of the DER energy for strand elasticity and the St. Venant-Kirchhoff Hencky energy from [29] to penalize collision and shearing,

$$\Psi = \Psi^s(\mathbf{F}^{d,E}) + \Psi^{DER}(\mathbf{F}^s). \tag{4.3}$$

The St.Venant-Kirchhoff Hencky energy, chosen for the ease of plasticity return mapping,

takes the form

$$\Psi^s = \int_\Omega \psi^s dV \tag{4.4}$$

$$\psi^s = \mu \text{tr}\left((\ln \boldsymbol{\Sigma})^2\right) + \frac{1}{2}\lambda \left(\text{tr}\left(\ln \boldsymbol{\Sigma}\right)^2\right) \tag{4.5}$$

where $\mathbf{F}^{d,E} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ is the singular value decomposition of the elastic deformation, Ω is the original domain the material occupies, and μ and λ are Lamé parameters. The DER energy Ψ^{DER} consists of stretching, twisting, and bending potentials. We refer readers to [3] for details on this energy and the time parallel transport required to calculate the force. The derivatives of the potential with respect to deformation are needed for computation and satisfy

$$\frac{\partial \psi^S}{\partial \mathbf{F}^E}(\mathbf{F}^E) = \mathbf{U}\left(2\mu\boldsymbol{\Sigma}^{-1}\ln(\boldsymbol{\Sigma}) + \lambda\boldsymbol{\Sigma}^{-1}\ln(\boldsymbol{\Sigma})\right)\mathbf{V}^T. \tag{4.6}$$

4.2 Discretization

As discussed in §4.1, we decompose the motion of the hair into that representing individual strand deformation $\boldsymbol{\phi}^s$ and that of frictional sliding and compression $\boldsymbol{\phi}^d$. As in [25, 19], we discretize these two motions in different ways. Since $\boldsymbol{\phi}^s$ only considers single hair strands, it suffices to discretize the energy and forces with traditional FEM. We do this using the approach of [2, 3]. However, unlike the approaches in [25, 19], we do not make use of an updated Lagrangian discretization of $\boldsymbol{\phi}^s$. To do so severely limits the ability of the hair to resolve collisions without a prohibitively high-resolution Eulerian grid (see Figure 4.1). Rather, we split the updates of $\boldsymbol{\phi}^s$ and $\boldsymbol{\phi}^d$, where the velocities for $\boldsymbol{\phi}^s$ are first updated in a Lagrangian manner and $\boldsymbol{\phi}^d$ with a standard updated Lagrangian MPM discretization. We then adopt the approach of McAdams et al. [33] where the grid-based updates are interpreted as impulsive changes in velocities on the strand that prevent self collision. However, by foregoing the updated Lagrangian discretization of $\boldsymbol{\phi}^s$, we cannot guarantee that self collision is prevented and thus revert to geometric impulses after the correction from $\boldsymbol{\phi}^d$.

The discrete state for each strand at time t^n consists of centerline particle positions $\mathbf{x}_p^n$, with velocities $\mathbf{v}_p^n$, masses m_p, APIC matrix $\mathbf{C}_p^n$, and elastic and plastic deformation

gradients associated with $\boldsymbol{\phi}^d$, $\mathbf{F}_p^{E,n}$ and $\mathbf{F}_p^{P,n}$. Furthermore, each edge e connecting particles $\mathbf{x}_e^n$ and $\mathbf{x}_{e+1}^n$ stores orientation angle θ_e as in [3]. Note that we do not add additional particles on segment centers to resolve $\boldsymbol{\phi}^d$ as in [25] and simply resolve them at the strand midline vertices $\mathbf{x}_p^n$. We summarize essential steps in the algorithm for updating the discrete state to time t^{n+1} below.

1. **Lagrangian update:** Update particle velocities from strand model of [3]. §4.2.1
2. **Transfer to grid:** Transfer mass and momentum from particles to grid using APIC as in [27]. §4.2.2
3. **Update grid momentum:** Compute effect of collision potential and friction elasto-plasticity. §4.2.3
4. **Apply impulses:** Interpolate the change in grid velocity to particles and then apply geometric collision handling. §4.2.4
5. **Update positions:** Update particle positions as in Equation (4.13).

4.2.1 Lagrangian update

We adopt a time splitting scheme for the velocity update where the velocity is first updated according to the force induced by the energy Ψ^{DER}. Specifically, we have

$$\mathbf{v}_p^* = \mathbf{v}_p^n + \Delta t \frac{\mathbf{f}_p}{m_p} \tag{4.7}$$

where $\mathbf{f}_p$ is calculated as in [3]. This new velocity $\mathbf{v}_p^*$ is then transferred to the MPM background grid $\mathbf{v}_\mathbf{i}^*$ in the next step.

4.2.2 Transfer to grid

We transfer mass and momentum from hair particles $\mathbf{x}_p^n$ to grid nodes $\mathbf{x_i}$ using APIC transfers

$$m_{\mathbf{i}}^n = \sum_p w_{\mathbf{i}p}^n m_p \tag{4.8}$$

$$\mathbf{v}_{\mathbf{i}}^* = \frac{1}{m_{\mathbf{i}}^n} \sum_p w_{\mathbf{i}p}^n m_p \left(\mathbf{v}_p^* + \mathbf{C}_p^n (\mathbf{x}_i^n - \mathbf{x}_p^n) \right). \tag{4.9}$$

Note that this transfer scheme is the similar to the standard MPM transform described in §3.3 with the only difference being the velocity being transferred is $\mathbf{v}_p^*$ calculated in §4.2.1 instead of $\mathbf{v}_p$.

4.2.3 Grid momentum update

The grid momentum is then updated according to the elastoplasticity model for the $\boldsymbol{\phi}^s$ motion and associated potential energy Ψ^s:

$$\mathbf{v}_{\mathbf{i}}^\star = \mathbf{v}_{\mathbf{i}}^* - \frac{dt}{m_{\mathbf{i}}^n} \sum_p \frac{\partial \psi^S}{\partial \mathbf{F}^E} (\tilde{\mathbf{F}}_p^E(\tilde{\mathbf{x}}^{n+\alpha}))(\mathbf{F}_p^{E,n})^T \nabla w_{\mathbf{i}p}^n V_P^0 + \Delta t \mathbf{g}. \tag{4.10}$$

Here, $\tilde{\mathbf{F}}_p^E(\tilde{\mathbf{x}}^{n+\alpha})$ is the trial elastic strain and $\tilde{\mathbf{x}}^{n+\alpha}$ is the vector of all Eulerian grid node positions, moved according to

$$\mathbf{x}_{\mathbf{i}}^{n+\alpha} = \mathbf{x}_{\mathbf{i}} + \alpha \Delta t \mathbf{v}_{\mathbf{i}}^\star, \;\; \tilde{\mathbf{F}}_p^E = (\mathbf{I} + \alpha \Delta t \sum_{\mathbf{i}} \mathbf{v}_{\mathbf{i}}^\star \nabla w_{\mathbf{i}p}^n) \mathbf{F}_p^{E,n} \tag{4.11}$$

where $\alpha = 0$ corresponds to symplectic Euler and $\alpha = 1$ corresponds to backward Euler for the grid momentum update. We also update APIC matrix $\mathbf{C}_p$ using grid velocity $\mathbf{v}_{\mathbf{i}}^\star$ as in [27, 28].

4.2.4 Impulses

To interpret the motion in $\mathbf{v}_{\mathbf{i}}^\star$ as inducing impulsive change in momentum on the midline, we interpolate the change in the grid velocity to the particles. However, we blend in the updated Lagrangian response weighted with parameter ξ

$$\mathbf{v}_p^\star = (1 - \xi) \left(\mathbf{v}_p^* + \sum_{\mathbf{i}} (\mathbf{v}_{\mathbf{i}}^\star - \mathbf{v}_{\mathbf{i}}^*) w_{\mathbf{i}p}^n \right) + \xi \sum_{\mathbf{i}} \mathbf{v}_{\mathbf{i}}^\star w_{\mathbf{i}p}^n. \tag{4.12}$$

This is equivalent to the PIC/FLIP blend used in [33]. Typically, we introduce $\xi = 0.95$. However, abandoning the updated Lagrangian update can leave collisional modes unresolved for hair. We apply geometric collision handling similar to [7] to resolve remaining collisional modes.

Collision impulses are applied based on proximity between strand edges. We use acceleration structures for efficient proximity queries as in [7]. However, we use regular grid-based structures inherent in MPM implementations. We divide the domain into calculation pads in space with edge length l. Then we extend the pad in the positive axis direction by proximity threshold δ so that neighboring pads have an overlap of length at least δ and thus any proximity pair will appear in at least one pad. In parallel, each extended pad collects all segments that have at least one endpoint contained in the pad, and then registers any proximity pairs contained in its set of segments. We apply an impulse to any proximity pair on a colliding trajectory as determined by relative velocity component on the direction separating the pair. The inelastic impulses from [7] are then calculated and distributed to particles. Also as proposed in Bridson et al. [7], we divide the total impulse on a particle by the number of impulses it receives from all pads and perform Jacobi iteration. After a fixed number of iterations, we obtain the particle velocity $\mathbf{v}_p^{n+1}$, and then advect particles using

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}. \tag{4.13}$$

4.3 Results

We demonstrate that our method preserves the intricate dynamics of individual hair strands and robustly handles the numerous collisions among them. In Figure 4.1, 32 thousand strands of hair with 60 segments per strand are simulated subject to intense boundary motions. Our algorithm is able to run this challenging example at 122 seconds per frame. In Figure 4.2 and Figure 4.3, we show that our method effortlessly resolves the intense self collisions occurring in braiding examples. In Figure 4.4 and Figure 4.5, we show a mannequin with a full head of hair in motions common in everyday life, such as walking and dancing. Because of the use of MPM quadrature particles, two-way coupling with granular materials simulated with

the standard MPM is effortless. In Figure 4.6, we show a hairy ball that is first hit by a snowball and then shakes the snow off.

In Figure 4.7, we compare our method with McAdams et al. [33] in a numerical experiment where a bundle of hair strands falls and bounces off another bundle. The experiments are run with a total of 2700 hair strands with 175 segments per strand. Five iterations of impulse application are applied to resolve the collisions missed by advecting the segments with the velocity in Equation (4.12) in our method and the velocity satisfying incompressibility condition in [33]. Notice that our method preserves the volume of the hair bundle and does not suffer from numerical cohesion. We run the test for 100 frames until the hair bundles are apparently separated and track the missed collisions in the process by calculating the collision interactions between pairs of segments using the cubic solve proposed in [7]. The test using McAdams et al. [33] registers more than 543 thousand missed collisions whereas the test using our method registers 120 missed collisions. Our method runs three times faster (see Table 4.1). Note that our method not only avoids the expensive Poisson solve for incompressibility, but it also serves as a better approximate collision response and therefore reduces the run time and number of missed collisions in the collision impulse step.

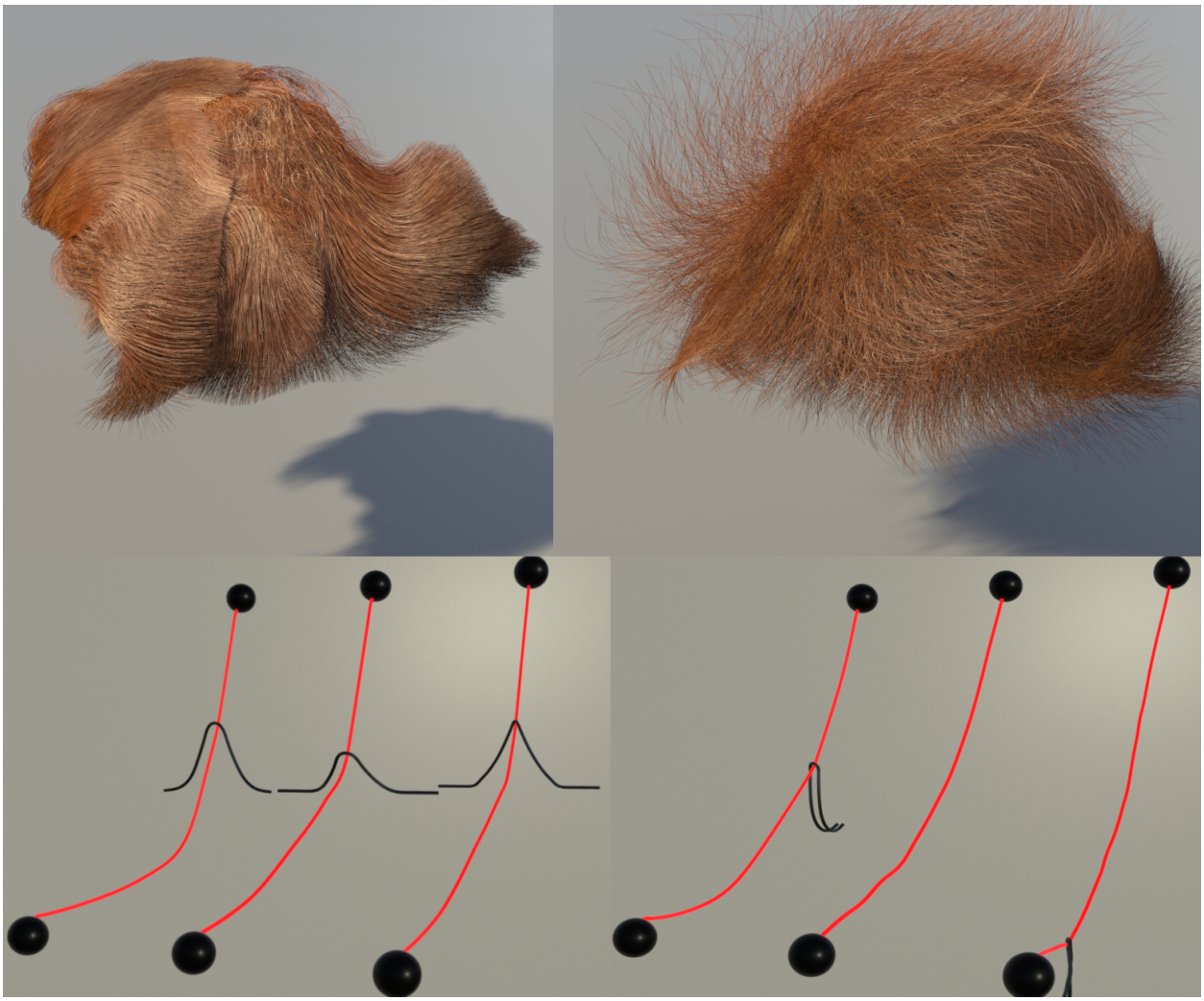


Figure 4.1: **Hair comparison with MPM**. **Top row:** MPM simulation of hair exhibits excessive friction and cohesion whereas our method captures the rich dynamics of individual strands. **Bottom row:** We demonstrate the dynamics of two hair strands, colored black and red, at two time steps. MPM (left) results in uncontrolled friction. Hybrid method without geometric collision (middle) misses the collision. Our method (right) captures the sliding behavior between two strands.

Figure 4.2: **Braids**. Our method captures the dynamics of a braid by robustly resolving many collisions.

Figure 4.3: **Braiding**. Two bundles of hair are intertwined into a braid and then separated.

Figure 4.4: **Hair**. A walking mannequin with a full head of hair.

Figure 4.5: **Dancer**. Hair of a dancer in motion.

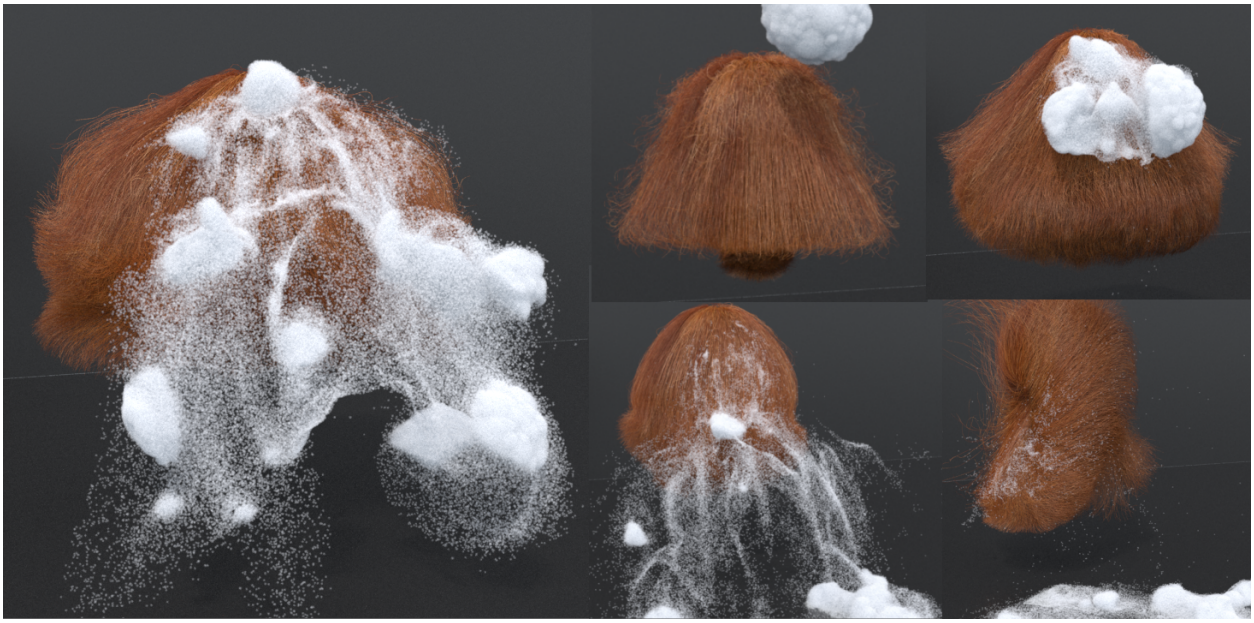


Figure 4.6: **Coupling hair with snow**. Our method captures the dynamics of a snowball falling on a head of hair.

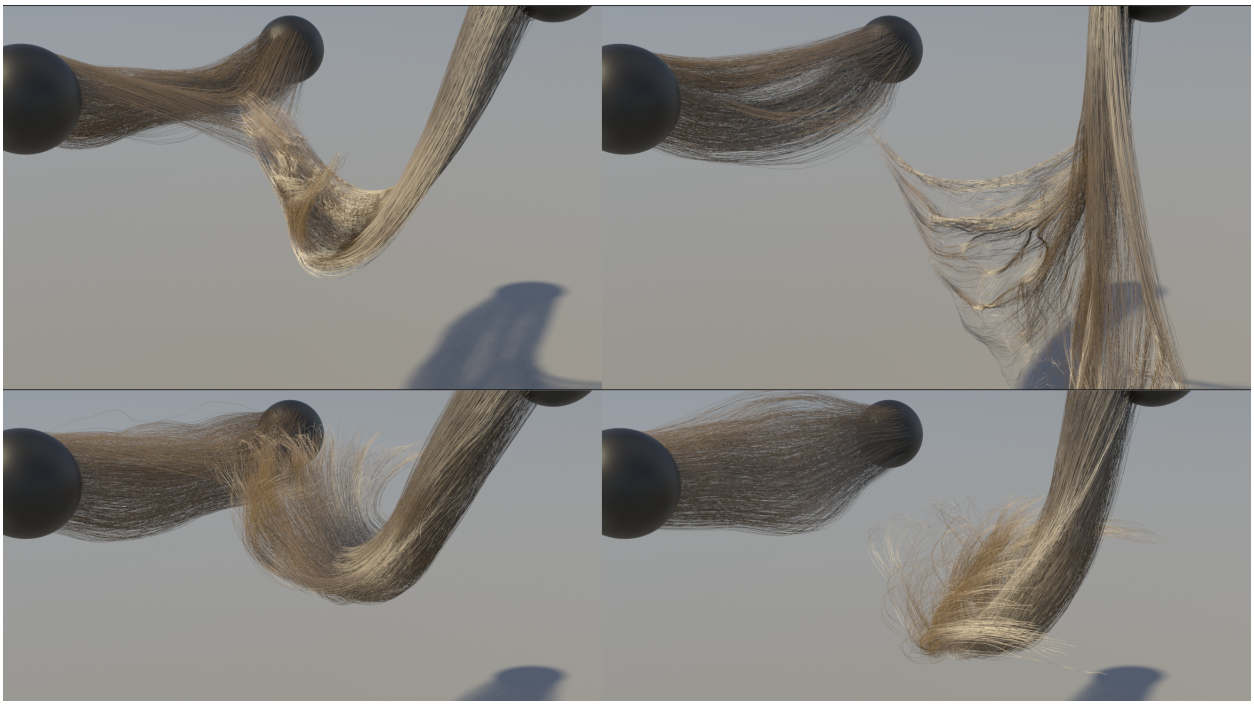


Figure 4.7: **Hair tubes comparison**. Comparison between McAdams et al. [33] (top row) and our method (bottom row) in resolving the collisions between two bundles of hair strands.

Table 4.1: All simulations were run on an Intel Xeon E5-2690 V2 system with 20 threads and 128GB of RAM. Simulation time is measure in seconds per frame. Time spent on geometric collision per frame is recorded in the second entry of the timing column where applicable. Element # denotes number of segments for hair simulations. Particle # denotes the total number of MPM particles.

	Time	Element #	Particle #	Δx	CFL
Hair ball (MPM) (Fig. 4.1 top left)	84	$1.92M$	N/A	0.05	0.1
Hair ball (Hybrid) (Fig. 4.1 top right)	122/83	$1.92M$	N/A	0.05	0.1
Hair tubes ([33]) (Fig. 4.7 top)	156/56	$47.5K$	N/A	0.08	0.6
Hair tubes (Hybrid) (Fig. 4.7 bottom)	55/11	$47.5K$	N/A	0.08	0.6
Braiding (Fig. 4.3)	87/73	$372K$	N/A	0.15	0.2
Braids (Fig. 4.2)	25/9	$323K$	N/A	0.03	0.2
Hair (Fig. 4.4)	127/46	$1.01M$	N/A	0.05	0.6
Snow on hair (Fig. 4.6)	153/38	$1.92M$	$2.16M$	0.05	0.2
Dancer (Fig. 4.5)	117/27	$490K$	N/A	0.04	0.2

CHAPTER 5

Thin Shell with Frictional Contact

5.1 Mathematical Details and Notation

We follow the notation conventions laid out in Chapter 2. In addition, we use brackets around bold face to denote matrices associated with a tensor in a given basis (e.g $[\mathbf{M}] \in \mathbb{R}^{3\times 3}$ is the matrix of entries $m_{ij} \in \mathbb{R}$ where tensor $\mathbf{m} = m_{ij}\mathbf{e}_i \otimes \mathbf{e}_j$). We use the convention that Greek indices (e.g. a_α) range from $1-2$ and Latin indices (e.g. b_i) range from $1-3$. We use hat notation to indicate the upper left 2×2 sub matrix of a given matrix (e.g. $\left[\hat{\mathbf{M}}\right] \in \mathbb{R}^{2\times 2}$ consists of entries $m_{\alpha\beta}$ from $[\mathbf{M}] \in \mathbb{R}^{3\times 3}$). For a set of (covariant) basis vectors $\mathbf{v}_i$, we use $\mathbf{v}^j$ to denote the corresponding contravariant basis vectors satisfying $\mathbf{v}_i \cdot \mathbf{v}^j = \delta_i^j$.

We assume shells have constant thickness τ and use $\omega^\tau = \omega \times [-\frac{\tau}{2}, \frac{\tau}{2}]$ to parameterize the domain of the shell where ω is two-dimensional parameter domain for the mid-surface of the shell. We use $\bar{\mathbf{x}} : \omega \to \bar{\Omega}$ and $\mathbf{x} : \omega \to \Omega_t$ to denote the mappings from the mid-surface parameter domain to the reference ($\bar{\Omega}$) and time t (Ω_t) configurations of the mid-surface. Similarly we use $\bar{\mathbf{r}} : \omega^\tau \to \bar{\Omega}^\tau$ and $\mathbf{r} : \omega^\tau \to \Omega_t^\tau$ to denote mappings from the shell parameter domain to the reference ($\bar{\Omega}^\tau$) and time t (Ω_t^τ) configurations of the shell. We illustrate this in Figure 5.1. We will use $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3) \in \omega^\tau$ to denote coordinates in parameter space. We refer to surfaces $\mathbf{s}(\xi_1, \xi_2) = \mathbf{r}(\xi_1, \xi_2, \hat{\xi_3})$ in the shell with fixed values of the thickness parameter $\hat{\xi_3}$ as laminae and we refer to lines in the $\mathbf{l}(\xi_3) = \mathbf{r}(\hat{\xi_1}, \hat{\xi_2}, \xi_3)$ with fixed values of the surface parameters $\hat{\xi_1}, \hat{\xi_2}$ as fibers. We illustrate fibers and laminae in Figure 5.2.

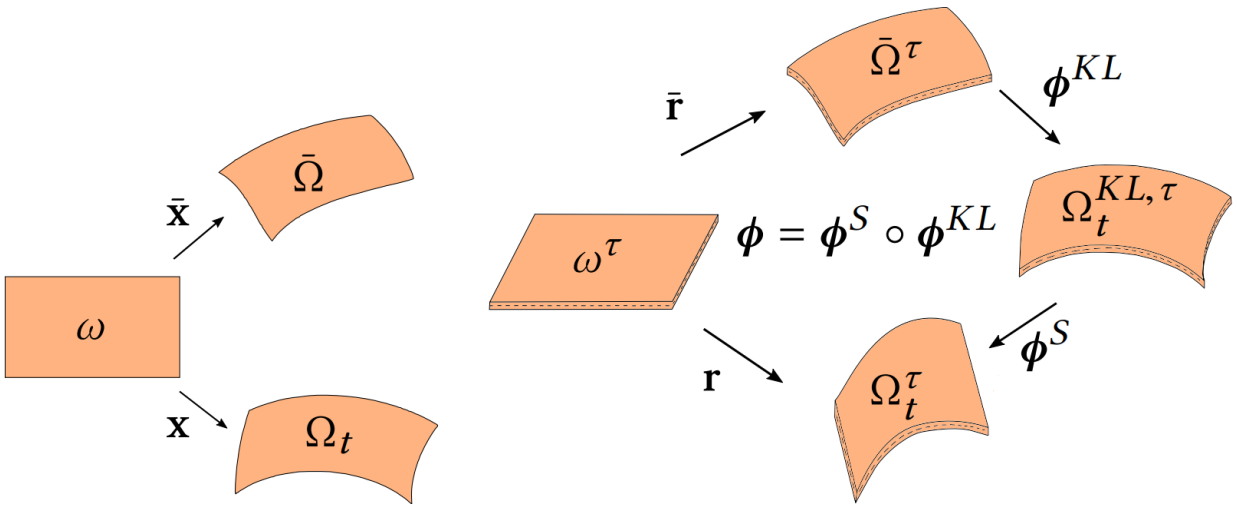


Figure 5.1: **Shell Kinematics**. On the left, the mid-surface mappings are illustrated, and on the right the corresponding volumetric shell mappings are shown.

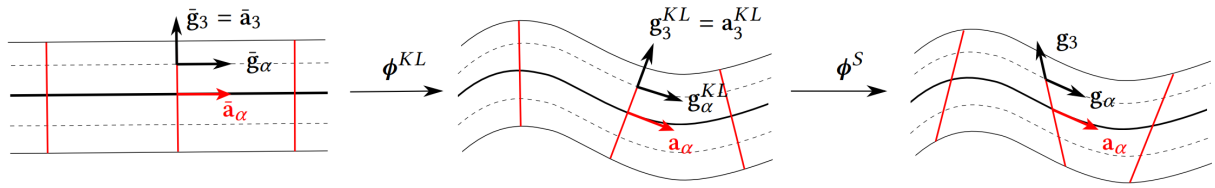


Figure 5.2: **Continuum shell/Kirchhoff-Love splitting**. Mid-surface tangents and fibers are shown in red. Laminae are shown as dashed curves, and the local frame at a point on a lamina is shown in black. On the left is the undeformed reference configuration, while the deformed configuration is on the right, and the middle shows the intermediate Kirchhoff-Love deformation.

5.2 Shell Kinematics

We assume the kinematics of a continuum shell

$$\bar{\mathbf{r}}(\boldsymbol{\xi}) = \bar{\mathbf{x}}(\xi_1, \xi_2) + \xi_3 \bar{\mathbf{a}}_3(\xi_1, \xi_2), \;\; \mathbf{r}(\boldsymbol{\xi}) = \mathbf{x}(\xi_1, \xi_2) + \xi_3 \mathbf{a}_3(\xi_1, \xi_2) \tag{5.1}$$

where $\bar{\mathbf{a}}_3$ is the unit normal to the mid-surface and $\mathbf{a}_3$ is the stretched and sheared image of $\bar{\mathbf{a}}_3$ under the motion of the shell. We use $\bar{\mathbf{a}}_\alpha = \frac{\partial \bar{\mathbf{x}}}{\partial \xi_\alpha}$ to denote the tangents to the mid-surface

of the reference shell. When combined with $\bar{\mathbf{a}}_3 = \frac{\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2}{|\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2|}$, they form a complete basis for $\mathbb{R}^3$ (see Figure 5.2).

We decompose the motion of the shell into two steps

$$\mathbf{r}(\boldsymbol{\xi}) = \boldsymbol{\phi}^S(\mathbf{r}^{KL}(\boldsymbol{\xi})). \tag{5.2}$$

The first step $\mathbf{r}^{KL} : \omega^\tau \to \Omega_t^{KL,\tau}$ does not see shearing or compression normal to the mid-surface. That is, lines originally normal to the mid-surface rotate and translate with the mid-surface so that they remain constant length and normal to the mid-surface. This is consistent with a Kirchhoff-Love kinematic assumption

$$\mathbf{r}^{KL}(\boldsymbol{\xi}) = \mathbf{x}(\xi_1, \xi_2) + \xi_3 \mathbf{a}_3^{KL}(\xi_1, \xi_2). \tag{5.3}$$

Here $\mathbf{a}_3^{KL}$ is the unit normal to the mid-surface which satisfies $\mathbf{a}_3^{KL} = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|}$ where $\mathbf{a}_\alpha = \frac{\partial \mathbf{x}}{\partial \xi_\alpha}$. The second step $\boldsymbol{\phi}^S : \Omega_t^{KL,\tau} \to \Omega_t^\tau$ does not move the mid-surface but captures the shearing and compression/extension of material normal to the mid-surface. That is, lines that remained normal to the mid-surface and with constant length in the Kirchhoff-Love mapping $\mathbf{r}^{KL}$ are allowed to change length and shear under the mapping $\boldsymbol{\phi}^S$, thus becoming non-normal to the mid-surface in general (see Figure 5.2).

5.2.1 Deformation Gradient

The motion of the shell from the reference configuration to the time t configuration is then obtained from the composition $\boldsymbol{\phi} : \bar{\Omega}^\tau \to \Omega_t^\tau$, $\boldsymbol{\phi}(\mathbf{X}) = \mathbf{r}(\bar{\mathbf{r}}^{-1}(\mathbf{X}))$ for $\mathbf{X} \in \bar{\Omega}^\tau$. The elastic and frictional contact responses of our model are characterized in terms of the spatial derivative (our deformation gradient) of this mapping. The deformation gradient of the motion is $\mathbf{F} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}} \left(\frac{\partial \bar{\mathbf{r}}}{\partial \boldsymbol{\xi}} \right)^{-1}$, which can further be expressed in terms of derivatives from the parameter space $\mathbf{g}_i = \frac{\partial \mathbf{r}}{\partial \xi_i}$ and $\bar{\mathbf{g}}_i = \frac{\partial \bar{\mathbf{r}}}{\partial \xi_i}$ as $\mathbf{F} = \mathbf{g}_i \otimes \bar{\mathbf{g}}^i$. Here $\bar{\mathbf{g}}^i$ are the contravariant basis vectors associated with $\bar{\mathbf{g}}_i$. Furthermore, the composition of motion in Equation (5.2) leads to the multiplicative decomposition

$$\mathbf{F} = \mathbf{F}^S \mathbf{F}^{KL}, \quad \mathbf{F}^S = \mathbf{g}_i \otimes \mathbf{g}^{KL,i}, \quad \mathbf{F}^{KL} = \mathbf{g}_i^{KL} \otimes \bar{\mathbf{g}}^i \tag{5.4}$$

where $\mathbf{g}_i^{KL} = \frac{\partial \mathbf{r}^{KL}}{\partial \xi_i}$ and $\mathbf{g}^{KL,j}$ form the corresponding contravariant basis. We note that the third contravariant counterparts to the Kirchhoff-Love and material configuration bases are the same as their covariant counterparts because of the preservation of mid-surface normals in these mappings. That is, $\mathbf{g}_3^{KL} = \mathbf{g}^{KL,3} = \mathbf{a}_3^{KL}$ and $\bar{\mathbf{g}}_3 = \bar{\mathbf{g}}^3 = \bar{\mathbf{a}}_3$ since $\mathbf{g}_\alpha^{KL} \cdot \mathbf{g}_3^{KL} = 0$ and $\bar{\mathbf{g}}_\alpha \cdot \bar{\mathbf{g}}_3 = 0$ (see [12] for details).

5.2.2 Plasticity

As in Jiang et al.[25], we use an elastoplastic decomposition of the motion to resolve frictional contact. Following that approach, we allow for plastic deformation in the fiber directions to enable material separation and frictional sliding. However, in order to decouple the frictional contact stress from the bending stress, we only apply the frictional contact elastoplastic decomposition to the shearing component of the motion. Furthermore, unlike in Jiang et al.[25] we also allow for plastic deformation in the laminae to account for yielding and denting of the shell. This plastic decomposition is applied to the motion in the Kirchhoff-Love component of the motion.

The frictional contact elastic stress model in Jiang et al. [25] penalizes compression and shearing of the surface normals. Since the Kirchhoff-Love component of the motion does not see any sliding or compression relative to the mid-surface, it is not capable of resolving frictional contact in this manner. We therefore apply this model to the shearing and compression/extension component of the shearing motion $\mathbf{F}^S = \mathbf{F}^{S,E}\mathbf{F}^{S,P}$ as

$$\mathbf{F}^{S,E} = \mathbf{g}_\alpha \otimes \mathbf{g}^{KL,\alpha} + \mathbf{a}_3^E \otimes \mathbf{g}_3^{KL}, \tag{5.5}$$

$$\mathbf{F}^{S,P} = \mathbf{g}_\alpha^{KL} \otimes \mathbf{g}^{KL,\alpha} + \mathbf{a}_3^P \otimes \mathbf{g}_3^{KL}. \tag{5.6}$$

Here $\mathbf{a}_3^E$ represents the shearing and compression/extension of normals in the shell that is penalized elastically. Coulomb friction constrains how much shearing and compression is penalized. $\mathbf{a}_3^P$ is the discarded shearing and extension in the fiber direction from plastic yielding associated with this constraint. They are related through $\mathbf{F}^{S,E}\mathbf{a}_3^P = \mathbf{a}_3$. We note $\mathbf{F}^{S,P}$ does not affect components in the laminae since we do not want the frictional contact response to couple with the elastoplasticity of the Kirchhoff-Love component of the shell

motion.

To allow for yielding and denting of the shell in response to loading, we decompose the Kirchhoff-Love component of the motion into lamina elastic and plastic parts $\mathbf{F}^{KL} = \mathbf{F}^{KL,E}\mathbf{F}^{KL,P}$

$$\mathbf{F}^{KL,E} = \mathbf{g}_\alpha^{KL} \otimes \mathbf{g}^{P,\alpha} + \mathbf{g}_3^{KL} \otimes \bar{\mathbf{g}}_3, \tag{5.7}$$

$$\mathbf{F}^{KL,P} = \mathbf{g}_\alpha^{P} \otimes \bar{\mathbf{g}}^{\alpha} + \bar{\mathbf{g}}_3 \otimes \bar{\mathbf{g}}_3 \tag{5.8}$$

Here the form of $\mathbf{F}^{KL,P}$ is designed to not affect the motion normal to the mid-surface since the elastoplasticity of denting and wrinkling is expressed only in terms of the lamina components of deformation. The expression for $\mathbf{F}^{KL,E}$ is then what remains to satisfy the constraint $\mathbf{F}^{KL} = \mathbf{F}^{KL,E}\mathbf{F}^{KL,P}$. We note that the $\mathbf{g}_\alpha^P$ (with $\mathbf{g}_\alpha^P \cdot \bar{\mathbf{g}}_3 = 0$) in Equation (5.8) for $\mathbf{F}^{KL,P}$ express the forgotten deformation of plastic yielding in the lamina that is associated with denting and wrinkling. The $\{\mathbf{g}^{P\alpha}, \bar{\mathbf{g}}_3\}$ are the contravariant counterparts to $\{\mathbf{g}_\alpha^P, \bar{\mathbf{g}}_3\}$. Lastly, $\bar{\mathbf{g}}_3$ is the same in the covariant and contravariant bases as in Equation (5.4).

5.3 Elastic Stress and Plastic Constraints

We define our elastoplastic constitutive response to deformation and frictional contact terms of potential energy in the shell. We decompose the total elastic potential as a sum of contributions from the Kirchhoff-Love (lamina elasticity, denting wrinkling etc.) and shearing (frictional contact) potentials. The contribution from the Kirchhoff-Love motion is

$$\Psi^{KL} = \int_\omega \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \psi(\mathbf{F}^{KL,E}) \left| \frac{\partial \bar{\mathbf{r}}}{\partial \xi} \right| d\boldsymbol{\xi}. \tag{5.9}$$

and the total elastic potential energy of the shell is

$$\Psi^{CS} = \Psi^{KL} + \int_\omega \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \chi(\mathbf{F}^{S,E}) \left| \frac{\partial \bar{\mathbf{r}}}{\partial \xi} \right| d\boldsymbol{\xi} \tag{5.10}$$

where $\psi(\mathbf{F}^{KL,E})$ is the elastic potential energy density of the Kirchhoff-Love motion and $\chi(\mathbf{F}^{S,E})$ is the energy density of the normal shearing and compression in the continuum shell motion.

These potentials are defined from energy densities $\psi(\mathbf{F}^{KL,E})$ and $\chi(\mathbf{F}^{S,E})$ respectively. In general, a potential energy density Ξ of this type is related to the material Kirchhoff stress $\boldsymbol{\tau}$ through $\boldsymbol{\tau} = \frac{\partial \Xi(\mathbf{F}^E)}{\partial \mathbf{F}^E}\mathbf{F}^E$. It is the stress defined through this relation that will directly affect our MPM implementation. In our elastoplastic model, the stress must satisfy certain constraints related to bending and denting as well as frictional contact. In the sections that follow we define these elastic stresses and their associated plastic constraints.

5.3.1 Bending and Lamina Potential

The energy density $\psi(\mathbf{F}^{KL,E})$ penalizes only the deformation in the laminae (zero transverse normal stress) since the Kirchhoff-Love kinematics preclude shearing and compression of the fibers. The stress in the material is the derivative of this potential with respect to strain (see Appendix A for derivation). Our approach supports any potential used in Kirchhoff-Love shell models. In particular we use the orthotropic model for woven fabrics from Clyde et al.[11] in Figures 5.8a and 5.8b. Here we provide the derivation of a simple energy density useful for applications with denting that is isotropic in the lamina directions while satisfying the zero transverse normal stress condition.

With Kirchhoff-Love kinematics, the lamina directions $\bar{\mathbf{g}}_\alpha = \bar{\mathbf{a}}_\alpha + \xi_3 \bar{\mathbf{a}}_{3,\alpha}$ and $\mathbf{g}^{KL}_\alpha = \mathbf{a}_\alpha + \xi_3 \mathbf{a}^{KL}_{3,\alpha}$ are always tangent to the mid-surface since $\bar{\mathbf{g}}_\alpha \cdot \bar{\mathbf{a}}_3 = \mathbf{g}^{KL}_\alpha \cdot \mathbf{a}^{KL}_3 = 0$. In order to satisfy the zero transverse normal stress conditions, we design a potential density with respect to the lamina directions by first writing the Kirchhoff-Love deformation in the reference mid-surface lamina/fiber basis $\mathbf{F}^{KL,E} = F^{KL,E}_{ij} \bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j$. Here the directions $\bar{\mathbf{a}}_\alpha$ are the tangents to the mid-surface in the reference configuration and $\bar{\mathbf{a}}_3$ is the normal. This choice of basis more clearly identifies deformations in the laminae and normal directions since $F^{KL,E}_{\alpha\beta}$ are then components of deformation in the laminae. The right Cauchy-Green strain is $\mathbf{C} = C_{ij} \bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j$ with $C_{ij} = F^{KL,E}_{ki} F^{KL,E}_{kj}$. We define the matrix $[\hat{\mathbf{C}}] \in \mathbb{R}^{2\times 2}$ with entries $C_{\alpha\beta}$. This is the upper left 2×2 block of the matrix of C_{ij} entries and it represents strain in the lamina. We use a model that is quadratic in the right Hencky strain $\left[\boldsymbol{\epsilon}^R\right] = \frac{1}{2}\log([\hat{\mathbf{C}}])$

$$\psi(\mathbf{F}^{KL,E}) = \mu \epsilon^R_{\alpha\beta} \epsilon^R_{\alpha\beta} + \frac{\lambda}{2}(\epsilon^R_{\delta\delta})^2. \tag{5.11}$$

Here the $\epsilon^R_{\alpha\beta}$ are the entries in $\left[\boldsymbol{\epsilon}^R\right] \in \mathbb{R}^{2\times 2}$ and μ, λ are Lame parameters that can be set intuitively from Young's modulus and Poisson ratio to control stiffness and incompressibility in the lamina. We choose the quadratic in Hencky strain model because it simplifies the return mapping during plastic yielding (see §5.3.2).

It is convenient for our MPM implementation as well as for the plasticity constraints to work with the Kirchhoff stress $\boldsymbol{\tau}$. It is related to the more commonly used Cauchy stress $\boldsymbol{\sigma}$ as $\boldsymbol{\tau} = \det(F)\boldsymbol{\sigma}$. The derivation of the Kirchhoff stress in terms of the potential is given in Appendix A. We summarize the expression as

$$\boldsymbol{\tau}^{KL} = \tau_{\alpha\beta}\mathbf{q}^{KL,E}_{\alpha} \otimes \mathbf{q}^{KL,E}_{\beta}, \quad \tau^{KL}_{\alpha\beta} = 2\mu\epsilon^{L}_{\alpha\beta} + \lambda\epsilon^{L}_{\gamma\gamma}\delta_{\alpha\beta}. \tag{5.12}$$

Here we write the stress in terms of the basis defined by the directions $\mathbf{q}^{KL,E}_i$ obtained from the QR decomposition $\mathbf{F}^{KL,E} = r^{KL,E}_{ij}\mathbf{q}^{KL,E}_i \otimes \bar{\mathbf{a}}_j$ with respect to the reference lamina/fiber basis $\bar{\mathbf{a}}_j$. Since the Kirchhoff-Love component of the motion preserves normals to the mid-surface, the first two directions $\mathbf{q}^{KL,E}_{\alpha}$ are tangent to the deformed lamina and the third direction $\mathbf{q}^{KL,E}_3$ is normal to the mid-surface. Therefore, since $\boldsymbol{\tau}^{KL}$ is expressed only in terms of $\mathbf{q}^{KL,E}_{\alpha}$, we see that it satisfies the zero transverse normal stress condition since it has no components in the directions normal to the laminae. We use $\epsilon^L_{\alpha\beta}$ to denote the entries in the left Hencky strain matrix $[\boldsymbol{\epsilon}^L] = \frac{1}{2}\log([\hat{\mathbf{r}}^{KL,E}][\hat{\mathbf{r}}^{KL,E}]^T) \in \mathbb{R}^{2\times 2}$. Here, $[\hat{\mathbf{r}}^{KL,E}] \in \mathbb{R}^{2\times 2}$ is the matrix with entries $r^{KL,E}_{\alpha\beta}$. These are the components of the deformation gradient $\mathbf{F}^{KL,E}$ related to the lamina strain. This formula follows directly from the definition of the energy in Equation (5.11) and we provide details of the derivation in Appendix A.

5.3.2 Denting Yield Condition and Return Mapping

In order to produce permanent denting and wrinkling phenomena resulting from excessive straining, we introduce a notion of yield stress. Intuitively, stresses satisfying the yield stress criteria are those associated with elastic, non-permanent deformation in the shell. Those that do not satisfy the condition are non-physical and permanent plastic deformation will occur to prevent them from happening. We apply the von Mises yield condition to the Kirchhoff-Stress in Equation (A.2). This condition states that the shear stress (or magnitude of the

deviatoric component of the stress) must be less than a threshold c_{vM} before permanent plastic deformation occurs

$$f_{vM}(\boldsymbol{\tau}) = |\boldsymbol{\tau} - \frac{\mathrm{tr}(\boldsymbol{\tau})}{3}\mathbf{I}|_F \leq c_{vM}. \tag{5.13}$$

This condition defines a cylindrical region of feasible states in the principal stress space since

$$f_{vM}(\boldsymbol{\tau}) = \sqrt{\frac{2}{3}\left({\tau_1}^2 + {\tau_2}^2 + {\tau_3}^2 - (\tau_1\tau_2 + \tau_2\tau_3 + \tau_1\tau_3)\right)} \tag{5.14}$$

where $\boldsymbol{\tau} = \sum_i \tau_i \mathbf{u}_i \otimes \mathbf{u}_i$ with principal stresses τ_i. Stresses with principal values in the cylinder do not produce any permanent deformation. Note that zero stress is inside the cylinder. As deformation becomes significant enough that the principal stresses reach the boundary of the cylinder, permanent plastic denting and wrinkling will occur. The zero transverse normal stress nature of $\boldsymbol{\tau}^{KL} = \sum_\alpha \tau_\alpha^{KL} \mathbf{u}_\alpha \otimes \mathbf{u}_\alpha$ means that its principal stresses are always in a plane and thus the feasible region is ellipsoidal intersection of the cylinder and the plane (see Appendix A for illustration).

In practice, the yield condition is satisfied via projection (or return mapping) of the stress to the feasible region. During simulation, we first take a time step to create a trial state of stress ignoring the yield condition. By ignoring the condition, we essentially assume the material undergoes no further plastic deformation. We use $\mathbf{F}^{KL,E^{\mathrm{tr}}}$, $\mathbf{F}^{KL,P^{\mathrm{tr}}}$ to denote this trial state of elastoplastic strains with associated trial stress $\boldsymbol{\tau}^{KL^{\mathrm{tr}}}$. This stress may or may not satisfy the yield condition. The trial stress $\boldsymbol{\tau}^{KL^{\mathrm{tr}}}$ is then projected to the feasible region to create $\boldsymbol{\tau}^{KL}$ which satisfies the yield condition. The elastic and plastic strains are then computed from the projected stress. We use $\mathbf{F}^{KL,E}, \mathbf{F}^{KL,P}$ to denote final elastic and plastic deformation associated with the projected stress $\boldsymbol{\tau}^{KL}$. The product of the projected elastic and plastic deformation gradients must be equal to the original deformation gradient, creating a constraint on the return mapping

$$\mathbf{F}^{KL} = \mathbf{F}^{KL,E^{\mathrm{tr}}}\mathbf{F}^{KL,P^{\mathrm{tr}}} = \mathbf{F}^{KL,E}\mathbf{F}^{KL,P}. \tag{5.15}$$

We describe the process as $\mathbf{F}^{KL,E^{\mathrm{tr}}}, \mathbf{F}^{KL,P^{\mathrm{tr}}} \rightarrow \mathbf{F}^{KL,E}, \mathbf{F}^{KL,P}$.

The projection is naturally done in terms of the QR decomposition of the trial elastic deformation gradient $\mathbf{F}^{KL,E^{\mathrm{tr}}} = r_{\alpha\beta}^{KL,E^{\mathrm{tr}}} \mathbf{q}_\alpha^{KL,E} \otimes \bar{\mathbf{a}}_\beta + \mathbf{q}_3^{KL,E} \otimes \bar{\mathbf{a}}_3$. The trial principal stresses

are

$$\tau_1^{KL,\mathrm{tr}} = (2\mu + \lambda)\log(\sigma_1^{E\mathrm{tr}}) + \lambda \log(\sigma_2^{E\mathrm{tr}}) \tag{5.16}$$

$$\tau_2^{KL,\mathrm{tr}} = (2\mu + \lambda)\log(\sigma_2^{E\mathrm{tr}}) + \lambda \log(\sigma_1^{E\mathrm{tr}}) \tag{5.17}$$

where $\sigma_\alpha^{E\mathrm{tr}}$ are the singular values of the matrix $[\hat{\mathbf{r}}^{KL,E\mathrm{tr}}] \in \mathbb{R}^{2\times 2}$ with entries $r_{\alpha\beta}^{KL,E\mathrm{tr}}$ from the QR decomposition

$$[\hat{\mathbf{r}}^{KL,E\mathrm{tr}}] = [\mathbf{U}^E] \begin{pmatrix} \sigma_1^{E^{\mathrm{tr}}} & \\ & \sigma_2^{E^{\mathrm{tr}}} \end{pmatrix} [\mathbf{V}^E]^T. \tag{5.18}$$

We project the trial $\tau_\alpha^{KL,\mathrm{tr}}$ to the intersection of the von Mises yield surface and the $(1,2)$ plane to obtain the projected τ_α^{KL} from which

$$\begin{pmatrix} \log(\sigma_1^E) \\ \log(\sigma_2^E) \end{pmatrix} = \begin{pmatrix} 2\mu + \lambda & \lambda \\ \lambda & 2\mu + \lambda \end{pmatrix}^{-1} \begin{pmatrix} \tau^{KL}{}_1 \\ \tau^{KL}{}_2 \end{pmatrix}. \tag{5.19}$$

We then express the deformation gradient associated with this stress projection as $\mathbf{F}^{KL,E} = F_{\alpha\beta}^{KL,E}\mathbf{q}_\alpha^{KL,E} \otimes \bar{\mathbf{a}}_\beta + \mathbf{q}_3^{KL,E} \otimes \bar{\mathbf{a}}_3$ where $F_{\alpha\beta}^{KL,E}$ are the components of the elastic deformation gradient

$$[\hat{\mathbf{F}}^{KL,E}] = [\mathbf{U}^E] \begin{pmatrix} \sigma_1^E & \\ & \sigma_2^E \end{pmatrix} [\mathbf{V}^E]^T. \tag{5.20}$$

The projected plastic deformation gradient is computed from $\mathbf{F}^{KL,P} = \mathbf{F}^{KL,E^{-1}}\mathbf{F}^{KL}$ in order to maintain the constraint in Equation (A.30). We provide more detail in this derivation in Appendix A.

5.3.2.1 Associativity and Hencky Strain

The projection of the trial stress to the feasible region is done using a generalized notion of closest point. This generalized projection is derived from the associative plastic flow assumption. Associativity requires that the closest points to the feasible stress region are not traced back along lines normal to the boundary, but rather along lines parallel to a matrix times the normal [5]. This matrix is associated with the linearization of the constitutive

model and in general it varies along the boundary. However, with the quadratic in Hencky strain model given in Equation (5.11), the matrix is constant along the boundary of the feasible region, which greatly simplifies the process of finding the generalized closest point. We illustrate this further in Appendix A.

5.3.3 Frictional Contact Potential

As in Jiang et al.[25], we resolve collision and contact through the continuum. We design the potential energy density $\chi(\mathbf{F}^{S,E})$ to penalize compression and shearing in the direction normal to the mid-surface as in Jiang et al.[25]. The deformation of the fiber from the Kirchhoff-Love configuration is given by $\mathbf{a}_3^E = \mathbf{F}^{S,E}\mathbf{a}_3^{KL}$. We decompose this into shear ($\mathbf{a}_{3S}^E$) and normal ($s_3^E\mathbf{a}_3^{KL}$) components $\mathbf{a}_3^E = \mathbf{a}_{3S}^E + s_3^E\mathbf{a}_3^{KL}$ where $s_3^E = \mathbf{a}_3^E \cdot \mathbf{a}_3^{KL}$. As material normal to the cloth is compressed, the normal component s_3^E will decrease and as the material separates, it will increase. Similarly, as material slides tangentially to the shell $|\mathbf{a}_{3S}^E|$ will increase. We therefore write our potential as

$$\chi(\mathbf{F}^{S,E}) = \frac{\gamma}{2}|\mathbf{a}_{3S}^E|^2 + f(s_3^E) \tag{5.21}$$

where γ represents the amount of shear resistance and

$$f(s_3^E) = \begin{cases} \frac{k^c}{3}(1 - s_3^E)^3 & 0 \leq s_3^E \leq 1 \\ 0 & s_3^E > 1 \end{cases} \tag{5.22}$$

represents the resistance to compression/contact which increases with the parameter $k^c > 0$. This potential is designed to increase, and thus penalize, increasing compressive contact and shear. Note that in the case of fiber elongation ($s_3^E > 1$) there is no elastic penalty as this would be associated with cohesive contact.

The potential in Equation (5.21) is constant in the fiber direction since $\mathbf{a}_3^{KL}$ is constant along the fiber from the continuum shell kinematics. Therefore it is convenient to express the contact potential χ at all points in the fibers in terms of their values at the mid-surface $\chi(\mathbf{F}^{S,E}(\xi_1, \xi_2, \xi_3)) = \chi(\mathbf{F}^{S,E}(\xi_1, \xi_2, 0))$ since

$$\int_\omega \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \chi(\mathbf{F}^{S,E}) \left|\frac{\partial \bar{\mathbf{r}}}{\partial \xi}\right| d\boldsymbol{\xi} = \int_\omega \chi(\mathbf{F}^{S,E}) \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \left|\frac{\partial \bar{\mathbf{r}}}{\partial \xi}\right| d\boldsymbol{\xi} \tag{5.23}$$

in Equation (5.10). On the mid-surface $\mathbf{F}^{S,E}(\xi_1, \xi_2, 0) = \mathbf{a}_\alpha \otimes \mathbf{a}^{KL,\alpha} + \mathbf{a}_3^E \otimes \mathbf{a}_3^{KL}$. Furthermore, since the potential varies with the normal and tangential components of $\mathbf{a}_3^E$, it is equivalent to write the energy as a function of the tensor $\mathbf{a}_\alpha \otimes \bar{\mathbf{a}}_\alpha + \mathbf{a}_3^E \otimes \bar{\mathbf{a}}_3$ since its QR decomposition with respect to the $\bar{\mathbf{a}}_i$ basis satisfies

$$\mathbf{a}_\alpha \otimes \bar{\mathbf{a}}_\alpha + \mathbf{a}_3^E \otimes \bar{\mathbf{a}}_3 = r_{ij}^{S,E} \mathbf{q}_i^{S,E} \otimes \bar{\mathbf{a}}_j \tag{5.24}$$

and the energy density can then be written in terms of the QR decomposition as was done in Jiang et al.[25]

$$\chi(\mathbf{F}^{S,E}(\xi_1, \xi_2, 0)) = \frac{\gamma}{2} \left(r_{13}^{S,E^2} + r_{23}^{S,E^2} \right) + f(r_{33}^{S,E}). \tag{5.25}$$

This follows because the normal and shear components of $\mathbf{a}_3^E$ can be written in terms of the basis vectors $\mathbf{q}_i^{S,E}$ from the QR decomposition $\mathbf{a}_3^E = r_{i3}^{S,E} \mathbf{q}_i^{S,E}$. With this convention, $s_3^E = r_{33}^{S,E}$ since $\text{span}\{\mathbf{a}_\alpha\} = \text{span}\{\mathbf{q}_\alpha^{S,E}\}$ and $\mathbf{q}_3^{S,E} = \mathbf{a}_3^{KL}$. Using $s_i^E = r_{i3}^{S,E}$ for conciseness

$$\boldsymbol{\tau}^S = \gamma s_i^E s_j^E \mathbf{q}_i^{S,E} \otimes \mathbf{q}_j^{S,E} + \left(f'(s_3^E) s_3^E - \gamma s_3^{E^2} \right) \mathbf{q}_3^{S,E} \otimes \mathbf{q}_3^{S,E}. \tag{5.26}$$

We provide a more detailed derivation of energies defined in terms of the QR decomposition and this specific case in Appendix A.

5.3.4 Frictional Contact Yield Condition and Return Mapping

With a continuum view of frictional contact, Coulomb friction defines a constraint on the types of stress that are admissible. This can be done concisely in terms of the Cauchy stress $\boldsymbol{\sigma}$. This stress measure is defined through contact interactions internal to a continuum body [17]. Specifically, the contact force per unit area across a surface with normal $\mathbf{n}$ is $\boldsymbol{\sigma}\mathbf{n}$. In the shell, the contact direction is $\mathbf{a}_3^{KL}$. Coulomb friction places a constraint on the stress as

$$|\mathbf{t}_S| \leq -c_F \sigma_n \tag{5.27}$$

where $\boldsymbol{\sigma}\mathbf{a}_3^{KL} = \sigma_n \mathbf{a}_3^{KL} + \mathbf{t}_S$. Here $\boldsymbol{\sigma}\mathbf{a}_3^{KL}$ is contact force per unit area, $\sigma_n \mathbf{a}_3^{KL}$ is its normal component and $\mathbf{t}_S$ is the shearing component orthogonal to $\mathbf{a}_3^{KL}$. The condition in Equation (5.27) states that the magnitude of the shearing component can be no larger than a

coefficient of friction times the normal component, with the convention that no shearing is allowed in the case of $\sigma_n > 0$ since this would be a separating rather than a compressive state. We note that each object can have its own coefficient of friction which provides a simple way of modeling interactions between many objects.

The Kirchhoff stress is related to the Cauchy stress as $\boldsymbol{\tau} = \det(\mathbf{F})\boldsymbol{\sigma}$. By design, the Kirchhoff-Love Kirchhoff stress has no component in the $\mathbf{a}_3^{KL}$ direction $\boldsymbol{\tau}^{KL}\mathbf{a}_3^{KL} = \mathbf{0}$. Therefore, the Coulomb friction constraint applies only to the shearing Kirchhoff stress $\boldsymbol{\tau}^S$. Using Equation (5.26) we can see that the continuum stress Coulomb friction condition is

$$\sqrt{s_1^{E\,2} + s_2^{E\,2}} \leq \begin{cases} \frac{c_F k^c}{\gamma}\left(1 - s_3^E\right)^2, & 0 < s_3^E \leq 1 \\ 0, & s_3^E > 1 \end{cases} \tag{5.28}$$

Whereas the plastic constraint associated with denting involved the principal stresses of $\boldsymbol{\tau}^{KL}$, only the components s_i^E of the elastic $\mathbf{a}_3^E$ in the $\mathbf{q}_i^{S,E}$ basis are constrained under the Coulomb condition. It is satisfied with a return mapping of trial elastic $\mathbf{a}_3^{E^{\mathrm{tr}}} = s_i^{E^{\mathrm{tr}}}\mathbf{q}_i^{S,E}$ to the projected $\mathbf{a}_3^E = s_i^E\mathbf{q}_i^{S,E}$ where the trial and projected coefficients are related through

$$s_\alpha^E = \begin{cases} h(\mathbf{a}_3^{E^{\mathrm{tr}}})s_\alpha^{E^{\mathrm{tr}}}, & 0 < s_3^{E^{\mathrm{tr}}} \leq 1 \\ 0, & s_3^{E^{\mathrm{tr}}} > 1 \end{cases}, \; s_3^E = \begin{cases} s_3^{E^{\mathrm{tr}}}, & 0 < s_3^{E^{\mathrm{tr}}} \leq 1 \\ 1, & s_3^{E^{\mathrm{tr}}} > 1 \end{cases} \tag{5.29}$$

with

$$h(\mathbf{a}_3^{E^{\mathrm{tr}}}) = \begin{cases} \frac{c_F k^c\left(1 - s_3^{E^{\mathrm{tr}}}\right)^2}{\gamma\sqrt{s_1^{E^{\mathrm{tr}}2} + s_2^{E^{\mathrm{tr}}2}}}, & \sqrt{s_1^{E\mathrm{tr}2} + s_2^{E\mathrm{tr}2}} > \frac{c_F k^c}{\gamma}\left(1 - s_3^{E\mathrm{tr}}\right)^2 \\ 1, & \sqrt{s_1^{E\mathrm{tr}2} + s_2^{E\mathrm{tr}2}} \leq \frac{c_F k^c}{\gamma}\left(1 - s_3^{E\mathrm{tr}}\right)^2. \end{cases} \tag{5.30}$$

This is the projection from Jiang et al.[25] where $0 < s_3^{E^{\mathrm{tr}}} \leq 1$ implies material is compressed from contact in the normal direction. In this case, the function h regulates the amount of shearing allowed relative to compression from the Coulomb constraint. In the case $s_3^{E^{\mathrm{tr}}} > 1$, material is separating in the normal direction and thus no resistance to shearing or compression is allowed.

5.4 Subdivision and B-spline FEM

The Kirchhoff-Love kinematics require higher regularity for mid-surface interpolating functions in FEM calculations. This arises from the use of the normal $\mathbf{a}_3^{KL}$ in the definition of the kinematics in Equation (5.3) since the deformation gradient in the shell then depends on second order derivatives of the kinematics of the mid-surface. Technically the requirement is H^2 regularity, meaning that the interpolating functions and all their derivatives of order less than or equal to two are square integrable over the mid-surface. In practice this means that the interpolating functions must also have continuous first derivatives (C^1 continuous) over the mid-surface. This is a challenging constraint on the interpolating functions. We represent the shell mid-surfaces as Catmull-Clark subdivision surfaces since they posses the required regularity.

The Catmull-Clark subdivision scheme takes as input an arbitrary polygonal mesh and returns a subdivided, refined mesh. The input polygonal mesh is referred to as the control mesh, and the limiting result of the subdivision process yields a H^2 surface [8, 38]. As the output mesh from Catmull-Clark subdivisions only consists of quadrilateral faces, we may assume that all input meshes have quadrilateral faces by replacing the control mesh with its first subdivision if necessary.

We denote the world space locations of the control points by $\mathbf{x}_p$, where $p = 1, ..., n_p$ and n_p is the number of control points. We use $\mathbf{x}^{KL} = \left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_p}\right)^T$ to denote the collection of all $\mathbf{x}_p$. The limiting surface from subdivision is represented as

$$\mathbf{x}(\mathbf{x}^{KL}, \xi_1, \xi_2) = \mathbf{x}_p N_p^{SD}(\xi_1, \xi_2),$$

where $N_p^{SD} \in H^2\Big(\omega \to [0,1]\Big)$ is the FEM basis weight function corresponding to the control point p. The N_p^{SD} have only local support and for each $(\xi_1, \xi_2) \in \omega$, only a sparse subset of $N_p^{SD}(\xi_1, \xi_2)$ are nonzero. We use the OpenSubdiv library to evaluate the basis functions $N_p^{SD}(\xi_1, \xi_2)$ and their first and second derivatives.

For each control mesh face, we sample rectangular quadrature points on either side of the face with $\xi_3 = -\frac{\tau}{4}$ and $\xi_3 = \frac{\tau}{4}$ for energy density evaluation. The generalized force on each

of the control points is calculated as the negative derivative of the Kirchhoff-Love energy in Equation (5.10) which we approximate using quadrature

$$\Psi^{KL} = \sum_q V_q^0 \psi(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL})) \tag{5.31}$$

The derivatives satisfy

$$\mathbf{f}_p^{KL} = -\frac{\partial \Psi^{KL}(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL}))}{\partial \mathbf{x}_p} \tag{5.32}$$

$$= -\sum_q V_q^0 \frac{\partial \psi}{\partial \mathbf{F}}(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL}))) : \frac{\partial \mathbf{F}_q^{KL,E\mathrm{tr}}}{\partial \mathbf{x}_p}(\mathbf{x}^{KL}). \tag{5.33}$$

Here ξ_{q1}, ξ_{q2} are the locations of the quadrature points in parameter space and V_q^0 are their associated volumes. For each quadrature point q, the Kirchhoff-Love deformation gradient at mid-surface configuration $\mathbf{x}^{KL}$ is computed from

$$\mathbf{F}_q^{KL}(\mathbf{x}^{KL}) = \sum_{i=1}^{3} \mathbf{g}_{qi}(\mathbf{x}^{KL}) \otimes \bar{\mathbf{g}}_q^i. \tag{5.34}$$

Furthermore, in Equation (5.33),

$$\frac{\partial \psi}{\partial \mathbf{F}}(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL})) = \boldsymbol{\tau}^{KL}(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL})) \left(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL})\right)^T$$

where $\boldsymbol{\tau}^{KL}$ is from Equation (A.2). This relation follows from the definition of the first Piola-Kirchhoff stress and its relation to the Kirchhoff stress [5].

The trial elastic deformation $\mathbf{F}^{KL,E\mathrm{tr}}$ and its derivative with respect to control points $\frac{\partial \mathbf{F}_q^{KL,E\mathrm{tr}}}{\partial \mathbf{x}_p}(\mathbf{x}^{KL})$ are computed assuming no further plastic flow over the time step

$$\mathbf{F}_q^{KL,Etr} = \mathbf{F}_q^{KL} {\mathbf{F}_q^{KL,P,n}}^{-1} \tag{5.35}$$

$$\frac{\partial \mathbf{F}_q^{KL,E\mathrm{tr}}}{\partial \mathbf{x}_p}(\mathbf{x}^{KL}) = \frac{\partial \mathbf{F}_q^{KL}}{\partial \mathbf{x}_p}(\mathbf{x}^{KL}) {\mathbf{F}_q^{KL,P,n}}^{-1} \tag{5.36}$$

Note that when calculating the generalized force in Equation (5.32)-(5.33), $\mathbf{F}^{KL,E\mathrm{tr}}$ is used even though the associated stress may not satisfy the yield criteria. This is a consequence of the variational FEM discretization of the analogous formula for the stress in terms of derivative of the strain energy density[5]. We provide the calculation of $\mathbf{F}_q^{KL}(\mathbf{x}^{KL})$ and $\frac{\partial \mathbf{F}_q^{KL}}{\partial \mathbf{x}_p}(\mathbf{x}^{KL})$ in Appendix A.

5.5 MPM Discretization

We use MPM to discretize our elastoplastic model for frictional contact. We represent the shell using particles connected with subd interpolation as in §5.4. That is, we consider the subd FEM control point as particles in a MPM method. This allows us to resolve contact and collision automatically through the elastoplastic constitutive behavior when we transfer to the background grid. There is no need for any collision detection or resolution other than that inherent in the MPM discretization of the continuum model. Furthermore, our approach naturally allows for coupling with materials (e.g. granular sand, snow and soil) simulated with MPM.

MPM is a hybrid Lagrangian/Eulerian approach. However, the primary representation of material for MPM is the Lagrangian state. At time t^n, we store particle position $\mathbf{x}_p^n$, velocity $\mathbf{v}_p^n$, initial mass m_p, initial volume V_p^0, affine velocity $\mathbf{C}_p^n$ for all materials in the simulation. Similar to Jiang et al.[25], we classify particles as either: (i) traditional MPM particles, (ii) subd particles or (iii) continuum shell shearing/compression particles. Particles of type (i) are used for coupling with traditional MPM materials like sand or snow. Types (ii) and (iii) are associated with elasticity and frictional contact respectively in the subd shell mesh. Furthermore, particles of type (ii) are control vertices in $\mathbf{x}^{KL}$ (see §5.4) for the subd shell and particles of type (iii) are quadrature points for the shearing component of the energy in Equation (5.10) and lie on the subd surface. For particles of type (i), we store the elastic deformation gradient $\mathbf{F}_p^{E,n}$. For particles of type (iii), we store the time t^n elastic shearing $\mathbf{a}_{p3}^E$ and the parameters in the mid-surface (ξ_{p1}, ξ_{p2}) associated with the particle. As in Jiang et al.[25], we use the notation $\mathcal{I}^{(i)}, \mathcal{I}^{(ii)}, \mathcal{I}^{(iii)}$ to represent the sets of particle indices of types (i), (ii) and (iii) respectively. At each of the quadrature points used in the Kirchhoff-Love energy, we store the deformation gradient and its elastic and plastic components $\mathbf{F}_q^{KL,n}$, $\mathbf{F}_q^{KL,E,n}$, $\mathbf{F}_q^{KL,P,n}$, the reference contravariant basis vectors $\bar{\mathbf{g}}_q^i$ needed for deformation gradient computation, and the mid-surface parameters (ξ_{p1}, ξ_{p2}) associated with the point. Although these quadrature points are not MPM particles and are not used in transfers to and from the grid etc., we additionally use $\mathcal{I}^{(iv)}$ to denote the collection of quadrature points used in the

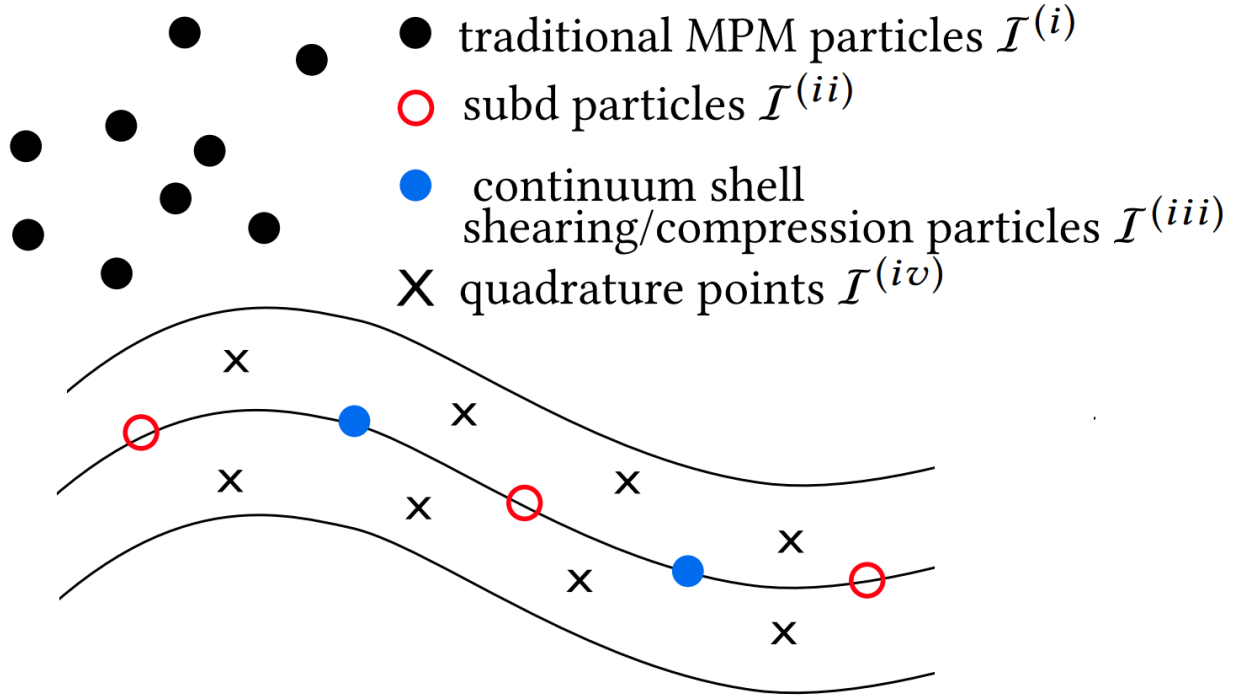


Figure 5.3: **Particle type classification**. A schematic illustration of the different types of MPM particles and quadrature points.

Kirchhoff-Love energy. We illustrate all particle and quadrature point types in Figure 5.3.

In MPM, the Eulerian grid can be viewed as an auxiliary structure for updating the Lagrangian state. We first transfer the particle mass and momentum state to an equivalent grid counterpart. We use $m_{\mathbf{i}}^n$ to denote the mass of Eulerian grid node $\mathbf{x_i}$ at time t^n, $\mathbf{v}_{\mathbf{i}}^n$ to denote its velocity and $\mathbf{p}_{\mathbf{i}}^{n+1}$ to denote its linear momentum after the grid update. The grid momentum is updated from the force defined as the gradient of the potential energy with respect to grid node motion. The motion of the grid is then interpolated to the particles to update the Lagrangian state without ever actually moving grid nodes. Our approach is ultimately very similar to other MPM methods that define forces from a notion of potential energy [45, 13, 39, 27, 29] and particularly Jiang et al.[25]. We briefly discuss aspects common to the approach of Jiang et al.[25] and discuss our novel modifications needed for subd shells in more detail. We summarize essential steps in the algorithm below.

1. **Transfer to grid:** Transfer mass and momentum from particles to grid. §5.5.1

2. **Update grid momentum:** Update grid momentum from potential-energy-based and body forces.

3. **Transfer to particles:** Transfer velocities from grid to particles. §5.5.3

4. **Update positions and trial elastic state:** Update particle position, deformation gradient and trial elastic state assuming no plasticity over the time step. §5.5.4

5. **Update plasticity:** Project trial elastic and plastic deformation gradients for plasticity return mapping. §5.5.5

5.5.1 Grid Transfers: Particle to Grid

To update the Lagrangian state, we transfer mass and momentum from particles $\mathbf{x}_p^n$ to the grid nodes $\mathbf{x_i}$ using APIC [27].

$$m_{\mathbf{i}}^n = \sum_p w_{\mathbf{i}p}^n m_p \tag{5.37}$$

$$\mathbf{v}_{\mathbf{i}}^n = \frac{1}{m_{\mathbf{i}}^n} \sum_p w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{C}_p^n (\mathbf{x}_{\mathbf{i}}^n - \mathbf{x}_p^n)) \tag{5.38}$$

Here $w_{\mathbf{i}p}^n = N(\mathbf{x}_p^n - \mathbf{x_i})$ is the weight of interaction between particle $\mathbf{x}_p^n$ and grid node $\mathbf{x_i}$. The $N(\mathbf{x})$ are linear, quadratic or cubic B-spline kernels used for interpolation over the grid. $\mathbf{v}_p^n$ and $\mathbf{C}_p^n$ define an affine notion of velocity local to the particle.

5.5.2 Grid Momentum Update

The grid momentum update uses the updated Lagrangian view of the governing physics [1, 15]. The grid at time t^n, after transferring state from the Lagrangian particles, is an alternative Lagrangian mesh with degrees of freedom $\mathbf{x_i}$, $\mathbf{v}_{\mathbf{i}}^n$ and mass $m_{\mathbf{i}}^n$. Its update is derived from the Lagrangian FEM discretization of a problem with a notion of potential energy. The internal force is the negative gradient of the potential energy with respect to positional changes. Using $\mathbf{x}_{\mathbf{i}}^{n+1}$ and $\mathbf{p}_{\mathbf{i}}^{n+1}$ to denote the new position and linear momentum

state after the time step, the grid discretization has the form

$$\mathbf{x}_\mathbf{i}^{n+1} = \mathbf{x}_\mathbf{i} + \frac{\Delta t}{m_\mathbf{i}^n}\mathbf{p}_\mathbf{i}^{n+1} \tag{5.39}$$

$$\mathbf{p}_\mathbf{i}^{n+1} = m_\mathbf{i}^n \mathbf{v}_\mathbf{i}^n - \Delta t \frac{\partial \Psi}{\partial \mathbf{x}_\mathbf{i}}(\mathbf{x}^*) + \Delta t m_\mathbf{i}^n \mathbf{g} \tag{5.40}$$

where $\Psi(\mathbf{x})$ is the potential energy which depends on the positional state where we use $\mathbf{x}^* = \left(\mathbf{x}_{\mathbf{i}^1}^*, \mathbf{x}_{\mathbf{i}^2}^*, \ldots\right)^T$ to denote the vector of all grid node positions. In the case of symplectic Euler integration, $\mathbf{x}_\mathbf{i}^* = \mathbf{x}_\mathbf{i}$ and in the case of backward Euler, $\mathbf{x}_\mathbf{i}^* = \mathbf{x}_\mathbf{i}^{n+1}$. We note that the grid nodes are not actually moved from $\mathbf{x}_\mathbf{i}$ to $\mathbf{x}_\mathbf{i}^{n+1}$. Instead, the motion of the grid is interpolated to the particles (see §5.5.3).

The potential energy Ψ is a sum of the contributions from the shell Ψ^{CS} and from traditional MPM particles ψ^M used for coupling multiple materials.

$$\Psi(\mathbf{x}^*) = \sum_{p \in \mathcal{I}^{(i)}} \psi^M\left(\mathbf{F}_p^{E,\mathrm{tr}}(\mathbf{x}^*)\right) V_p^0 + \Psi^{CS}(\mathbf{x}^*) \tag{5.41}$$

$$\begin{aligned}\Psi^{CS}(\mathbf{x}^*) = &\sum_{p \in \mathcal{I}^{(iii)}} \chi\Big(\mathbf{a}_{p\alpha}(\mathbf{x}^{KL}(\mathbf{x}^*)) \otimes \bar{\mathbf{a}}_{p\alpha} \\ &\qquad + \mathbf{a}_{p3}^{E,\mathrm{tr}}(\mathbf{x}^{KL}(\mathbf{x}^*)) \otimes \bar{\mathbf{a}}_{p3}\Big) V_p^0 \\ &+ \sum_{q \in \mathcal{I}^{(iv)}} \psi\left(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL}(\mathbf{x}^*))\right) V_q^0.\end{aligned} \tag{5.42}$$

Here ψ^M is the contribution from the standard MPM potential discretization (see e.g. Stomakhin et al.[39]) and Ψ^{CS} is the contribution from the continuum shell. An advantage of the MPM approach is that coupling is achieved between any materials whose constitutive behaviors can be defined from potential energies. With any such models, coupling is achieved by first representing the motion of the materials in a Lagrangian way (e.g. discrete particles or Lagrangian meshes) and defining their motion and the way it effects their potential energy in terms of interpolation from the grid. With this model, coupling is a simple as defining the total potential energy as the sum of the varied materials.

The energy Ψ^{CS} is the sum of the discretization of the Kirchhoff-Love component in Equation (5.10) given in Equation (5.31) and the frictional contact energy in Equation (5.23) obtained from the quadrature points $q \in \mathcal{I}^{(iv)}$ and $p \in \mathcal{I}^{(iii)}$ respectively. We highlight

the dependence of these potentials on the grid motion $\mathbf{x}^*$. For particles of type (i), this dependence follows from the updated Lagrangian formulation

$$\mathbf{F}_p^{E,\mathrm{tr}}(\mathbf{x}^*) = \left(\sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}^* \otimes \nabla w_{\mathbf{i}p}^n\right) \mathbf{F}_p^{E,n} \tag{5.43}$$

Here $\nabla w_{\mathbf{i}p}^n = \nabla N(\mathbf{x}_p^n - \mathbf{x}_{\mathbf{i}})$ is the gradient of the grid interpolating function (or weight gradient) and $\left(\sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}^* \otimes \nabla w_{\mathbf{i}p}^n\right)$ represents deformation induced by the grid motion $\mathbf{x}^*$. For particles of type (iii), the dependence follows from the updated Lagrangian

$$\mathbf{a}_{p3}^{E,\mathrm{tr}}(\mathbf{x}^*) = \left(\sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}^* \otimes \nabla w_{\mathbf{i}p}^n\right) \mathbf{a}_{p3}^{E,n} \tag{5.44}$$

and from interpolation the $\mathbf{x}^{KL}(\mathbf{x}^*)$ in Equation (5.45) in $\mathbf{a}_{p\alpha}(\mathbf{x}^{KL}(\mathbf{x}^*))$. Following the approaches in Jiang et al.[27, 25], the mid-surface control points for the shell are interpolated from the grid degrees of freedom as

$$\mathbf{x}_p^* = \sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}^* w_{\mathbf{i}p}^n, \quad p \in \mathcal{I}^{(ii)}. \tag{5.45}$$

This interpolation also affects the discrete Kirchhoff-Love term through quadrature points $q \in \mathcal{I}^{(iv)}$.

Taking the $\mathbf{x}^*$ dependence into account and using the chain rule, the potential energy based forces obtained from the gradient of Ψ with respect to $\mathbf{x}^*$ are

$$\frac{\partial \Psi}{\partial \mathbf{x}_{\mathbf{i}}}(\mathbf{x}^*) = \mathbf{f}_{\mathbf{i}}^{(i)}(\mathbf{x}^*) + \mathbf{f}_{\mathbf{i}}^{(ii)}(\mathbf{x}^*) + \mathbf{f}_{\mathbf{i}}^{(iii)}(\mathbf{x}^*) \tag{5.46}$$

$$\mathbf{f}_{\mathbf{i}}^{(i)}(\mathbf{x}^*) = \sum_{p \in \mathcal{I}^{(i)}} \frac{\partial \psi^M}{\partial \mathbf{F}^E}(\mathbf{F}_p^{E,\mathrm{tr}}(\mathbf{x}^*)) \mathbf{F}_p^{E,n\,T} \nabla w_{\mathbf{i}p}^n V_p^0 \tag{5.47}$$

$$\mathbf{f}_{\mathbf{i}}^{(ii)}(\mathbf{x}^*) = \sum_{p \in \mathcal{I}^{(ii)}} w_{\mathbf{i}p}^n \mathbf{f}_p^{KL}(\mathbf{x}^{KL}(\mathbf{x}^*)) \tag{5.48}$$

$$\mathbf{f}_{\mathbf{i}}^{(iii)}(\mathbf{x}^*) = \sum_{p \in \mathcal{I}^{(iii)}} \boldsymbol{\tau}_p^S \tilde{\mathbf{a}}_p^\beta : \frac{\partial \mathbf{a}_{p\beta}}{\partial \mathbf{x}_p} w_{\mathbf{i}p}^n + \boldsymbol{\tau}_p^S \tilde{\mathbf{a}}_p^3 : \nabla w_{\mathbf{i}p}^n \mathbf{a}_{p3}^{E,n} \tag{5.49}$$

In Equation (5.48), $\mathbf{f}_p^{KL}$ is the generalized Kirchhoff-Love force from Equation (5.32). In Equation (5.49), the stress $\boldsymbol{\tau}_p^S$ is from Equation (5.26) and the vector $\tilde{\mathbf{a}}_p^3$ is the third contravariant basis vector with respect to the covariant basis $\{\mathbf{a}_\alpha(\mathbf{x}^*), \mathbf{a}_3^{E,\mathrm{tr}}(\mathbf{x}^*)\}$. We refer to Appendix A for this derivation.

5.5.3 Grid Transfers: Grid to Particle

The grid to particle transfer defines the time t^{n+1} affine velocity local to particle $\mathbf{x}_p^n$ in terms of $\mathbf{v}_p^{n+1}$ and $\mathbf{C}_p^{n+1}$ from

$$\mathbf{v}_p^{n+1} = \sum_{\mathbf{i}} w_{\mathbf{i}p}^n \frac{\mathbf{p}_{\mathbf{i}}^{n+1}}{m_{\mathbf{i}}^n} \tag{5.50}$$

$$\tilde{\mathbf{C}}_p^{n+1} = \frac{12}{\Delta x^2 (d+1)} \sum_{\mathbf{i}} w_{\mathbf{i}p}^n \frac{\mathbf{p}_{\mathbf{i}}^{n+1}}{m_{\mathbf{i}}^n} \otimes (\mathbf{x}_{\mathbf{i}}^n - \mathbf{x}_p^n) \tag{5.51}$$

$$\mathbf{C}_p^{n+1} = (1-\nu)\,\tilde{\mathbf{C}}_p^{n+1} + \frac{\nu}{2}\left(\tilde{\mathbf{C}}_p^{n+1} - \tilde{\mathbf{C}}_p^{n+1T}\right) \tag{5.52}$$

Here d is the B-spline degree ($d = 3$ for cubic b-spline interpolation, $d = 2$ for quadratic B-spline interpolation) and Δx is the Eulerian grid spacing. ν is the explicit damping coefficient from Jiang et al.[25] where $\nu = 0$ is completely undamped and $\frac{1}{2}\left(\tilde{\mathbf{C}}_p^{n+1} - \tilde{\mathbf{C}}_p^{n+1T}\right)$ is the RPIC transfer from Jiang et al.[27].

5.5.4 Update Positions and Trial Elastic State

For particles of type (i) and (ii), positions are moved with the interpolated grid node velocities. For particles of type (iii), positions are updated based on interpolation from updated particles of type (ii).

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} = \sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}^{n+1} w_{\mathbf{i}p}^n,\ p \in \mathcal{I}^{(i)} \cup \mathcal{I}^{(ii)} \tag{5.53}$$

$$\mathbf{x}_p^{n+1} = \sum_{p^{(ii)} \in \mathcal{I}^{(ii)}} \mathbf{x}_{p^{(ii)}}^{n+1} N_{p^{(ii)}}^{SD}(\xi_{p1}, \xi_{p2}),\ p \in \mathcal{I}^{(iii)}. \tag{5.54}$$

We first assume there was no additional plastic flow over the time step and consider a trial state of elastic deformation. For particles of type (i) and (iii), the trial elastic deformation $\mathbf{F}_p^{E,\text{tr}}$ and $\mathbf{a}_{p3}^{E,\text{tr}}$ are computed as in Equations (5.43) and (5.44) respectively with $\mathbf{x}_{\mathbf{i}}^* = \mathbf{x}_{\mathbf{i}}^{n+1}$. For Kirchhoff-Love quadrature points $q \in \mathcal{I}^{(iv)}$, the trial elastic deformation gradient $\mathbf{F}_q^{KL,E\text{tr}}$ is computed from Equation (5.35) where $\mathbf{x}^{KL}(\mathbf{x}^*)$ is interpolated as in Equation (5.45) with $\mathbf{x}_{\mathbf{i}}^* = \mathbf{x}_{\mathbf{i}}^{n+1}$.

5.5.5 Update Plasticity

The assumption of no plastic flow over the time step is often safe. However, if the trial state of elastic stresses are not inside the yield surfaces associated with denting, frictional contact, etc. then they must be projected to satisfy the constraint. For particles $p \in \mathcal{I}^{(i)}$, $\mathbf{F}_p^{E,\text{tr}}$ is projected to $\mathbf{F}_p^{E,n+1}$ in accordance with whichever yield surface is being used (e.g. the Drucker-Prager law in Klár et al.[29]). For quadrature points $q \in \mathcal{I}^{(iv)}$, $\mathbf{F}_q^{E,\text{tr}}$ and $\mathbf{F}_q^{P,\text{tr}}$ are projected to $\mathbf{F}_q^{E,n+1}$ and $\mathbf{F}_q^{P,n+1}$ in accordance with the denting return mapping in §5.3.2. Lastly, the $\mathbf{a}_{p3}^{E,\text{tr}}$ are projected to $\mathbf{a}_{p3}^{n+1}$ in accordance with the frictional contact return mapping in Equation (6.23).

5.6 Results

We demonstrate the efficacy of our method on a number of representative examples that exhibit appreciable bending and persistent self-collision and show that our method automatically allows for coupling with granular materials. Furthermore, we demonstrate the range of behaviors that are possible with the parameters in our model. We list the run time performance for all of our examples in Table 5.1. All simulations were run on an Intel Xeon E5-2687W v4 system with 48 hyper-threads and 128GB of RAM. We report the timing in terms of average seconds of computation per frame. We chose Δt in an adaptive manner that is restricted by a CFL condition when the particle velocities are high. In all of our simulations we use a CFL number equal to 0.3, i.e., we do not allow particles to move further than $0.3\Delta x$ in a time step.

5.6.1 Effect of Shell Thickness

We control the bending stiffness of the shell by varying the thickness τ. In Figure 5.4, six cylinders with increasing thickness from left to right free-fall and drop on the ground. In Figure 5.5, four cylinders of decreasing thickness from left to right buckle under lateral pressure and exhibit characteristic buckling patterns. In Figure 5.6, ribbons of varying

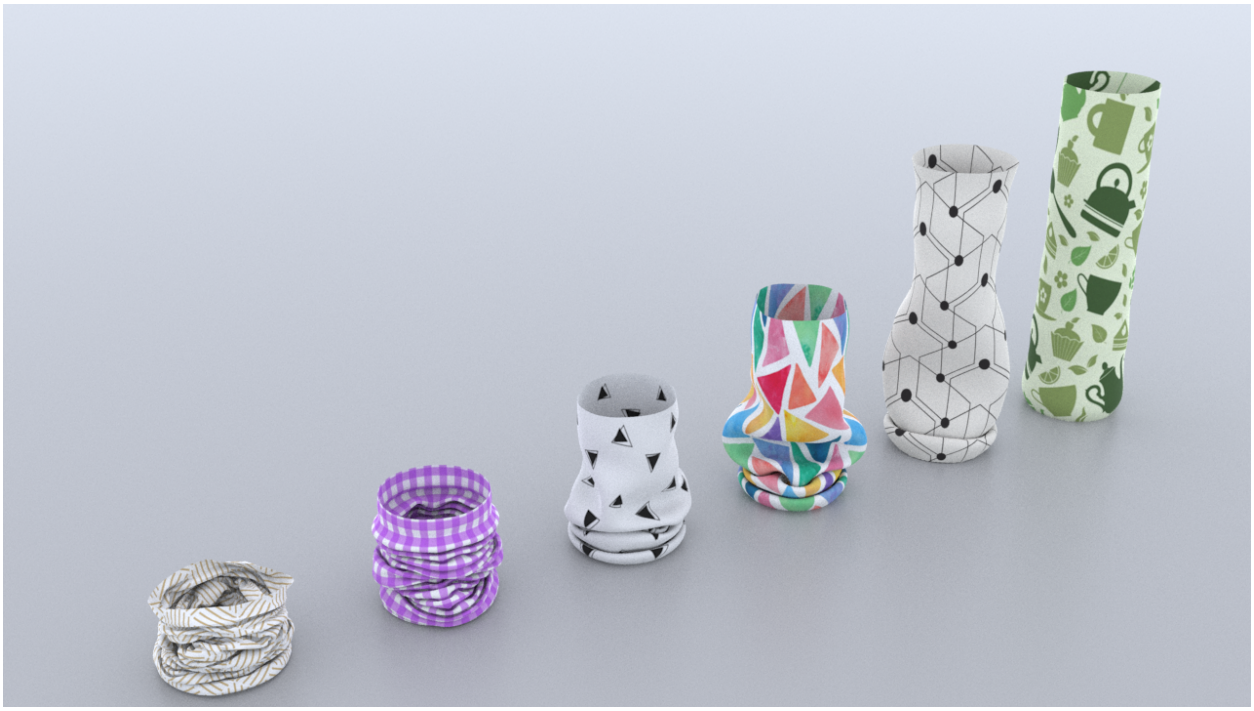


Figure 5.4: **Elastic cylinders**. We demonstrate the effect of increasing thickness (from left to right) for six collapsing elastic cylinders.

thickness are planted in plates and twisted to produce interesting buckling phenomena.

5.6.2 Woven Fabrics

We demonstrate that our method supports any potential function in the Kirchhoff-Love shell model. In particular, we implement the data-driven orthotropic model for woven fabrics from Clyde et al.[11] with parameters fitted from experimental data. In Figure. 5.7a and 5.7b, we twist and compress sleeves made of denim and silk. In Figure. 5.8a and 5.8b, we suspend squares of silk and denim which then collide with moving spheres. Our model accurately captures the behaviors of these real world materials. In Figure. 5.9, we demonstrate the behaviors of woven fabrics in a walk cycle.

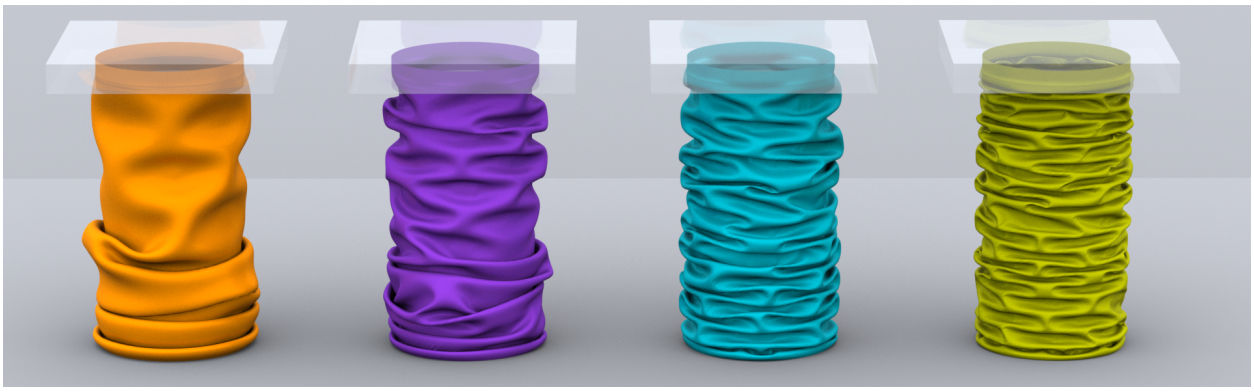


Figure 5.5: **Variation in shell thickness**. We demonstrate the effect of the shell thickness parameter in a compression comparison.

5.6.3 Self Collisions

Our model successfully resolves self-collision without any use of collision detection or constraint modeling outside the MPM discretization. We demonstrate this in a number of representative scenarios. In Figure 5.10, the spheres and the diving boards, both modeled as shells, collide with each other. In Figure 5.11 and Figure 5.12, we demonstrate self-collisions resolution for clothing simulation stress tests. In Figure 5.13, four decks of cards collide and then slide against each other to demonstrate the effect of varying friction coefficients.

5.6.4 Plasticity for Denting

Our method naturally incorporates the effect of plasticity in the shell. In Figure 5.14, three cylinders with different yield stress are twisted and then released. By changing the yield stress, we are able to control the amount of denting. In Figure 5.15, a square sheet of metal is compressed and then dented with a rod. The effect of plasticity creates permanent buckling and denting deformation.

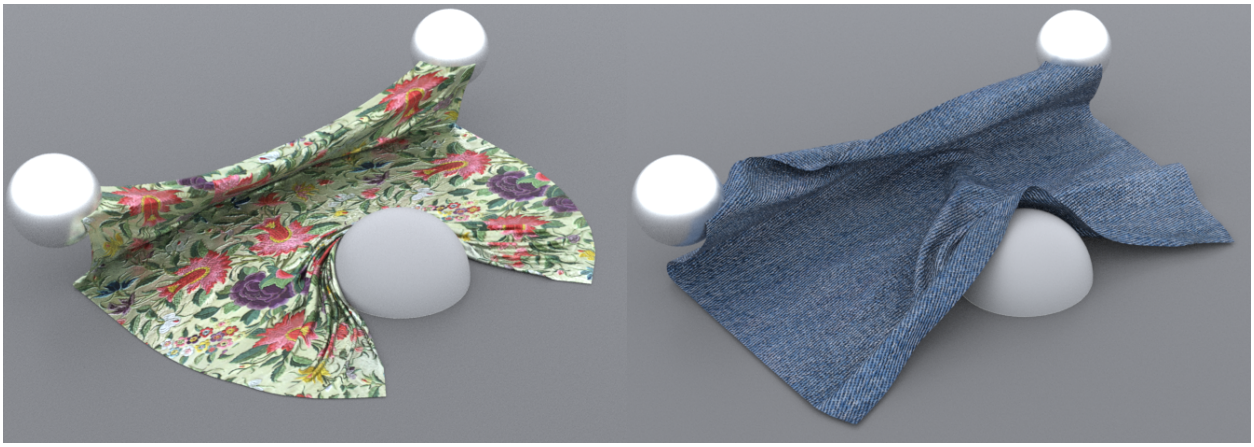
5.6.5 Two-way Coupling

Our MPM approach automatically resolves coupling of different materials. In Figure 5.16, a cup is filled with slush and then released and toppled. The cup is modeled as a shell and the

Figure 5.6: **Ribbons**. We illustrate interesting dynamics achieved from colliding ribbons with increasing thickness (from left to right).

(a) (b)

Figure 5.7: **Twisting Orthotropic Model**. Using the data-driven model of Clyde et al. [11] for woven materials, the characteristic wrinkling of silk (left) and denim (right) is obtained. Our method naturally resolves the many self-collisions induced by the twisting boundary conditions.



(a) (b)

Figure 5.8: **Orthotropic Model**. A range of materials can be simulated with our continuum shell formulation. Here we use the data-driven model of Clyde et al. [11] for woven silk (left) and denim (right) materials. The model naturally allows for characteristic buckling and wrinkling behaviors in this object collision test.

Figure 5.9: **Walk cycle**.

slush is modeled as in Stomakhin et al.[39]. This example demonstrates that our method successfully resolves the interactions between two different materials of millions of particles with moderate computation cost.

5.6.6 Resolution Refinement

In Figure 5.17 we examine the behavior of our method under refinement of grid and subd mesh spatial resolution. This refinement study is done on a sleeve-buckling simulation with boundary conditions compressing the material at top and bottom. As the spatial resolution is increased, the simulation converges to the characteristic buckling pattern that is expected.

5.6.7 Bending with Jiang et al.

We demonstrate the failure of the Jiang et al. [25] model in achieving significant bending resistance. In Figure 5.18 we compare our model with the Jiang et al. generalized to bending

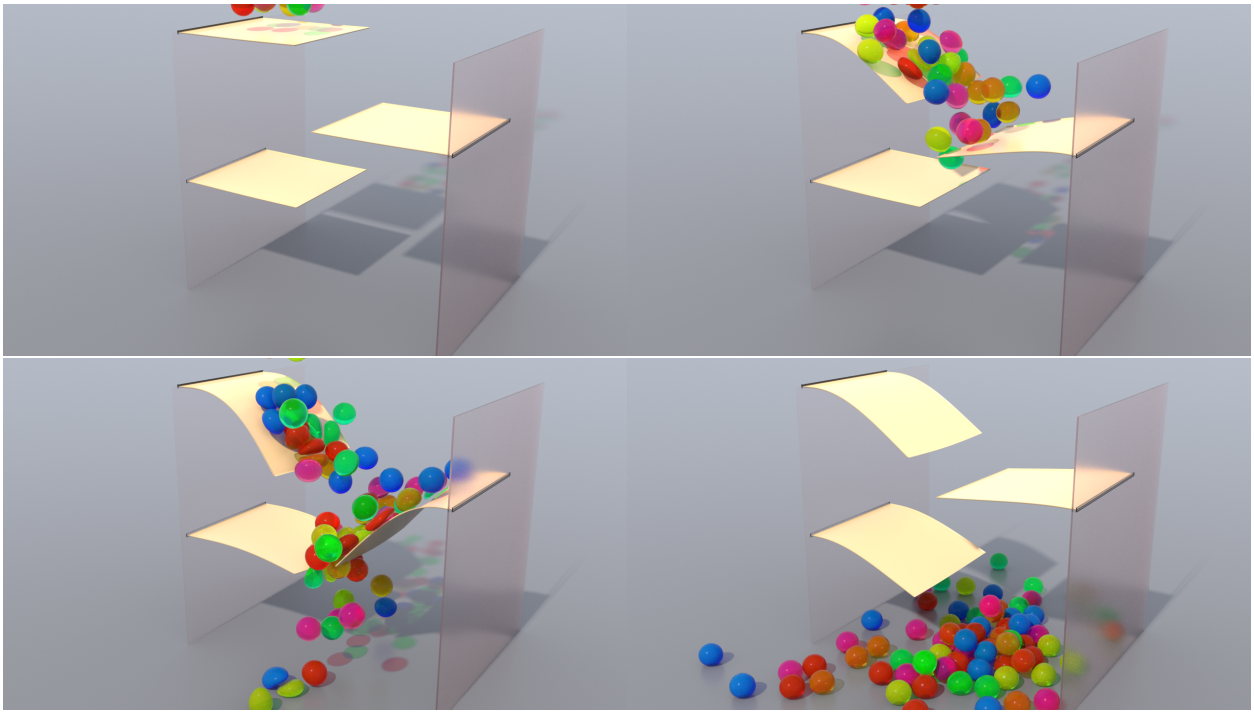


Figure 5.10: **Elastic spheres on diving boards**. We demonstrate appealing dynamics achieved with self-collision and appreciable bending for shells. Both the spheres and the diving boards are simulated as thin shells.

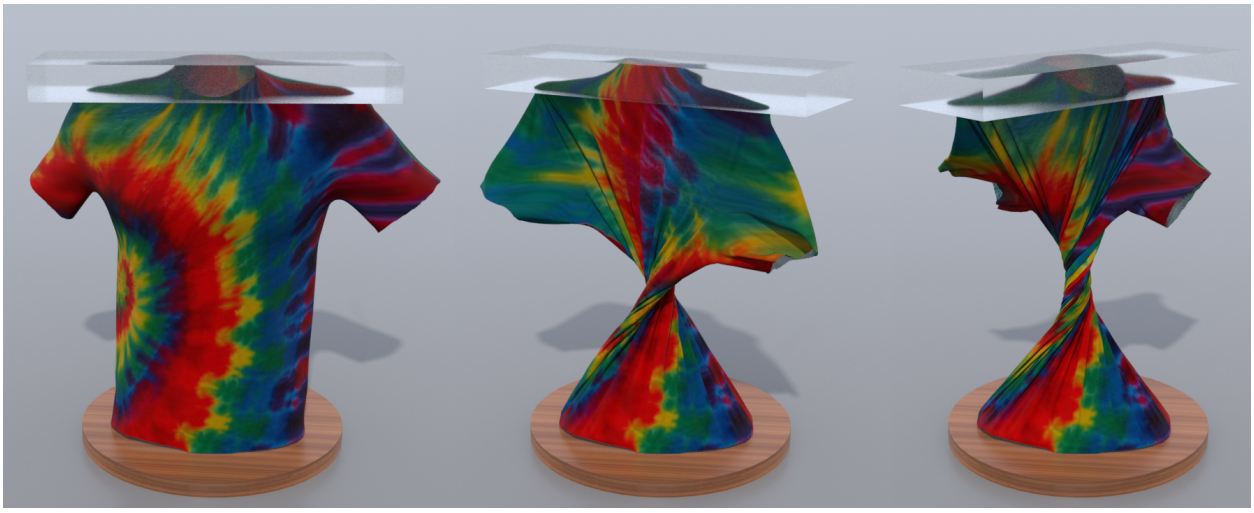


Figure 5.11: **Shirt twister**.

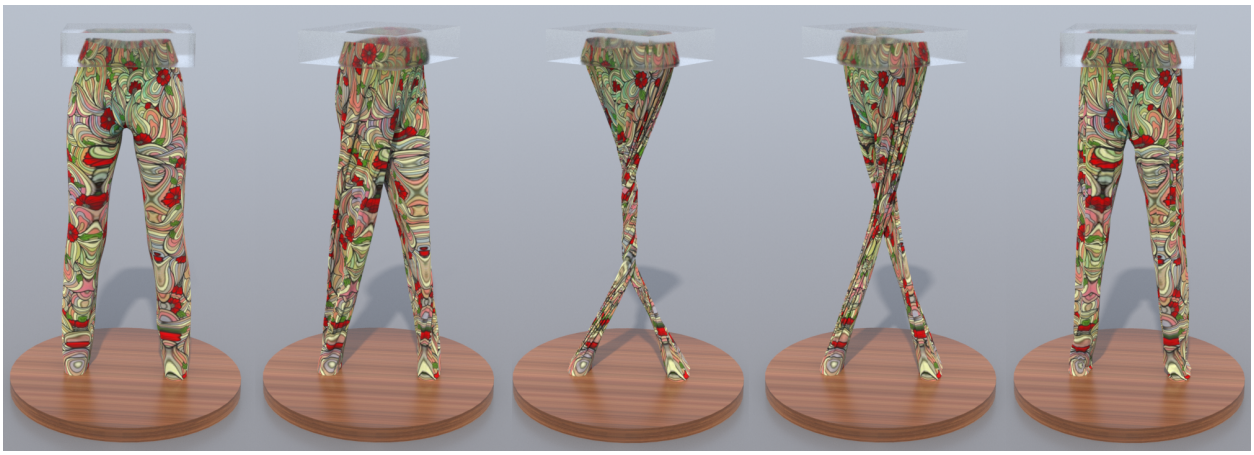


Figure 5.12: **Pants twister**. Our approach works for clothing simulation with many self collisions as shown here in the legs of a twisted pair of pants. The subdivision mesh for the pants has 393K control points and the simulation runs at 78s per frame.

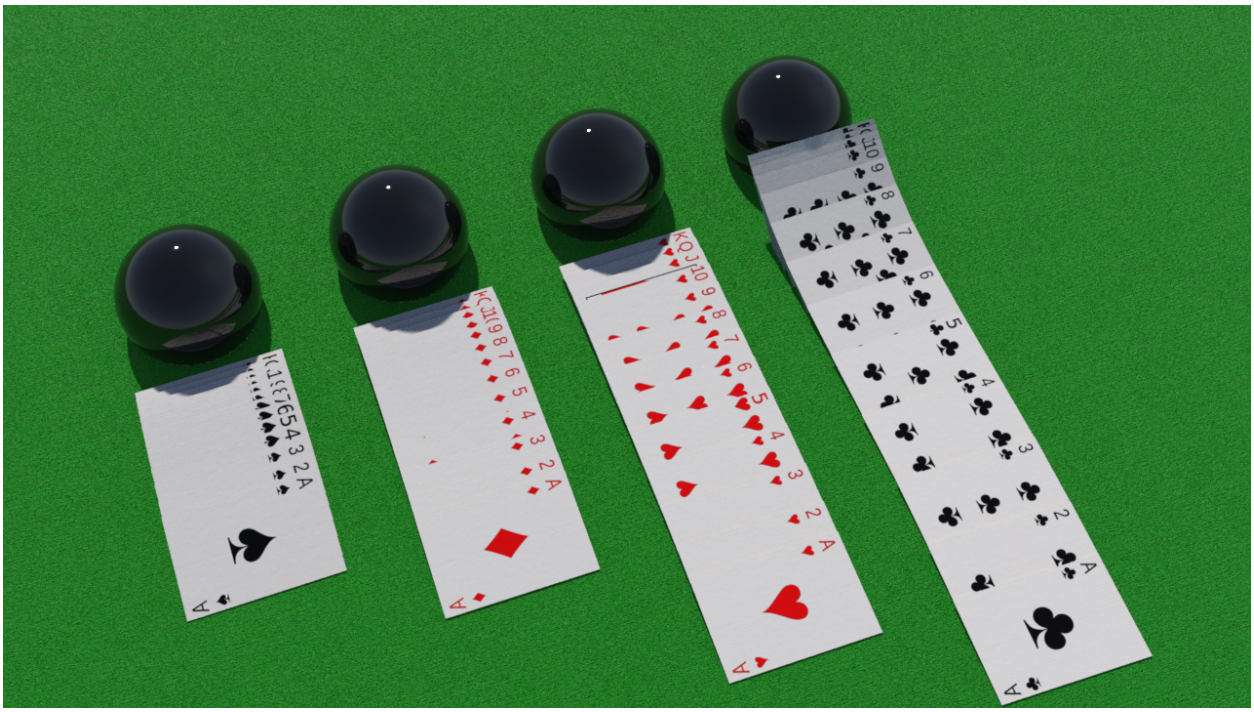


Figure 5.13: **Variation in Coulomb friction coefficient**. The effect of the friction parameter c_F can be seen in this card comparison. By decreasing c_F (from left to right) we demonstrate a range of surface frictions.

with the addition of bending springs. The frictional contact model in Jiang et al. [25] was not designed for bending resistance, however it is possible to simply add bending cross springs to their model even though it violates the stress assumptions. We show that this is not capable of generating significant resistance to bending whereas our approach is designed to support stiff shells and thin membranes.

5.7 Discussion and Limitations

While our method can efficiently simulate thin shells with extreme contact and collision, there are a number of notable limitations. First, we have the same artifacts as Jiang et al. [25], namely visible separation if Δx is too large, persistent wrinkles if subd mesh resolution is too high relative to the grid resolution and self-penetration if the resolution is too low relative to the grid. Also, the time step size is generally smaller than those used for membranes in Jiang et al. [25]. This is due to the added stiffness associated with shell thickness and bending. With MPM, the increase in time step size with implicit time stepping is bounded above since particles cannot move more than a grid cell in a time step without causing bunching, self collision or material inversion. Therefore the demand on the efficiency of nonlinear solver for the implicit systems is very high. Unfortunately this demand is difficult to meet since the nonlinear systems have non-symmetric linearizations that result from the plasticity [29]. "Lagging" the plasticity as in Stomakhin et al.[39] provides a symmetric linearization but can cause cohesion artifacts that are unacceptable for frictional contact applications. Development of a solver that is more efficient than Newton's method with GMRES for the linearized systems is an interesting area of future work.

	Figure	seconds/frame	Element #	Particle #	Δx
Cup of Slush	5.16	273	$19.5K$	$3.1M$	0.04
Shirt Twister	5.11	188	$168K$	$504K$	0.005
Six Cylinders	5.4	2/2/2/2/2/4	$20K$	$60K$	0.025
Walk Cycle	5.9	75	$33K$	$100K$	10
Silk Curtain	5.8a	167	$75K$	$227K$	0.004
Denim Curtain	5.8b	3	$8K$	$25K$	0.012
Pants Twister	5.12	78	$131K$	$393K$	0.005
Silk Twister	5.7a	47	$63K$	$315K$	0.02
Denim Twister	5.7b	3	$15.8K$	$47K$	0.04
Spheres On Diving Board	5.10	87	$150K$	$450K$	0.027
Playing Cards	5.13	55	$23K$	$115K$	0.02
Plastic Twister	5.14	< 1	$5K$	$14K$	0.06
Sleeves (Yellow)	5.5	97	$126K$	$378K$	0.01
Sleeves (Others)	5.5	8	$31K$	$93K$	0.02
Fixed Ribbons	5.6	3/8/30	$12K$	$93K$	0.02
Free Ribbons	5.6	4/4/7	$12K$	$85K$	0.02
Denting with Rod	5.15	< 1	$5K$	$15K$	0.01

Table 5.1: All simulations were run on Intel Xeon E5-2687W v4 system with 48 hyperthreads and 128GB of RAM. Element # denotes number of quadrilaterals. Particle # denotes the number of type (i), (ii) and (iii) particles.

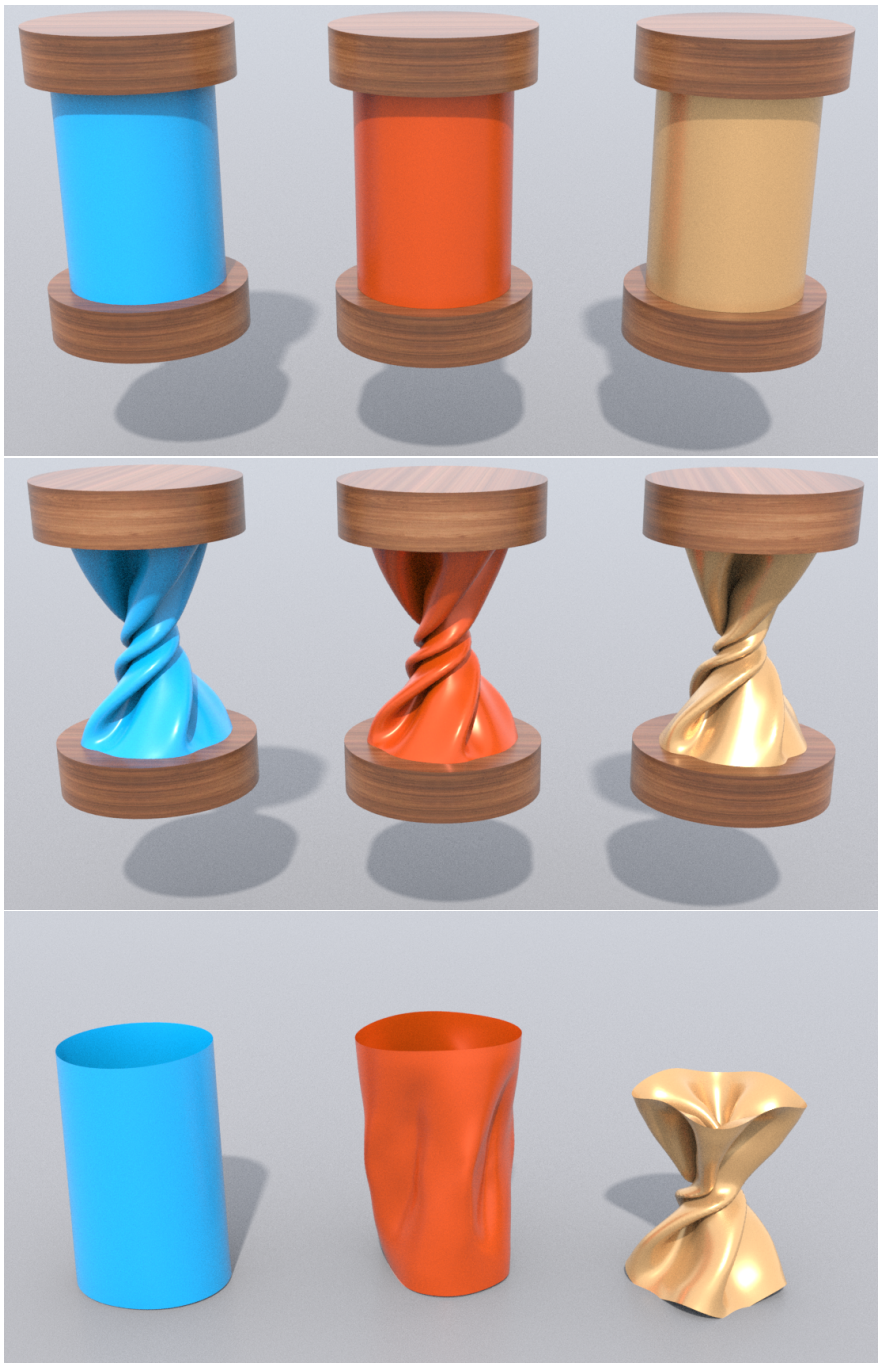


Figure 5.14: **Plastic shell deformation**. The effect of the yield condition in Equation (5.13) is shown here with decreasing values of the coefficient c_{vM} (from left to right). Larger values correspond to a larger stress needed for before denting plasticity is induced. The cylinders are twisted and then dropped to the ground to illustrate the plastic deformation.

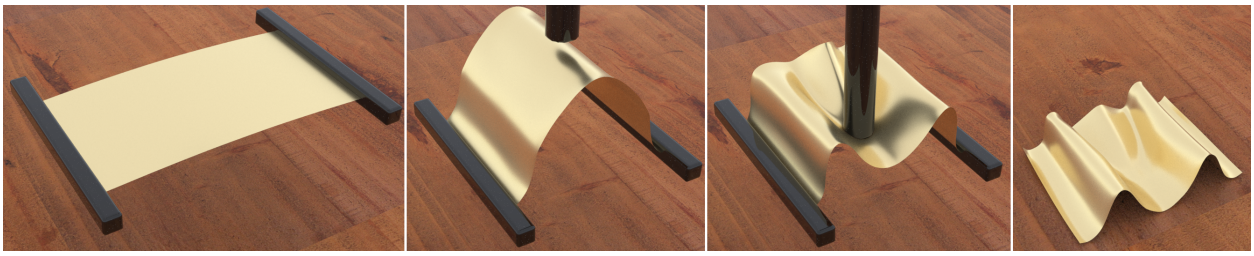


Figure 5.15: **Denting**. We demonstrate plastic deformation of foil induced by object collision.

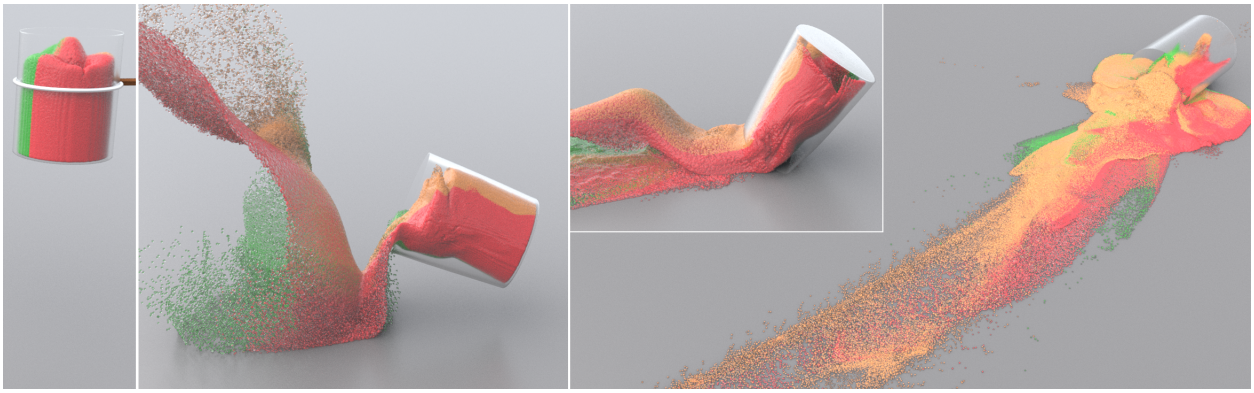


Figure 5.16: **Slushies**. Simulation of shells coupled with granular materials.

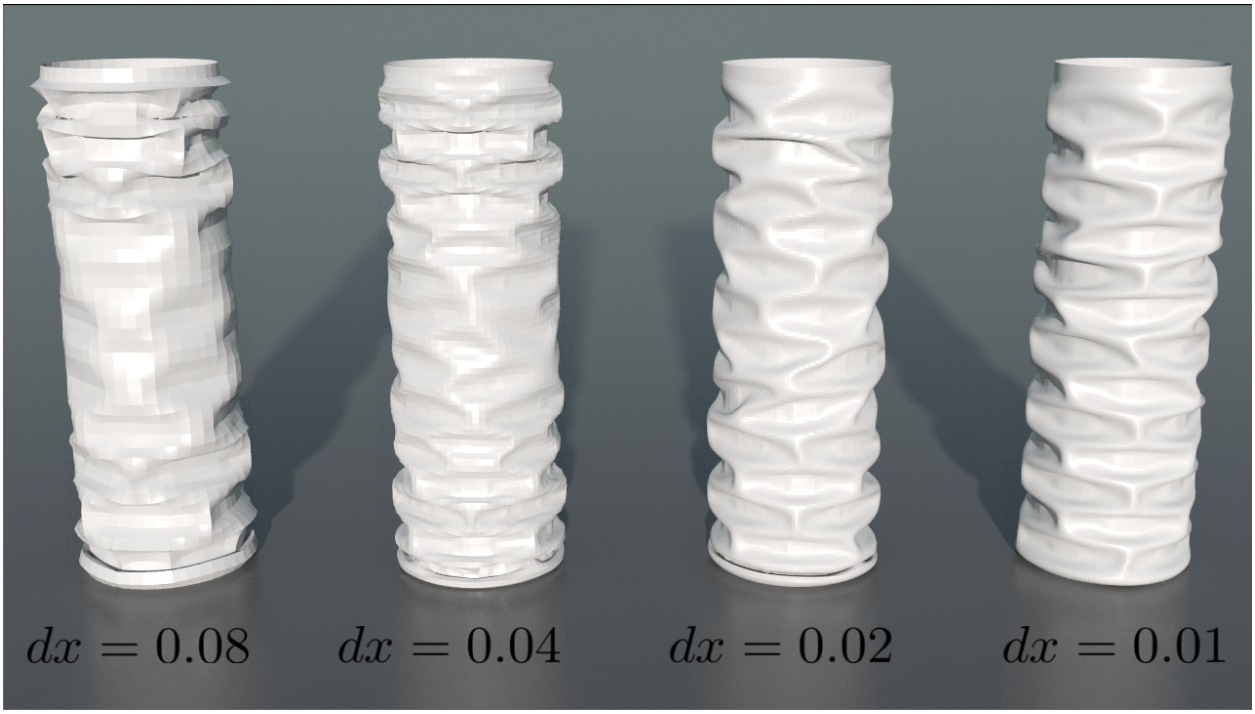


Figure 5.17: **Convergence under spatial refinement**. We demonstrate that our method converges under refinement of grid and subd mesh spatial resolution in this buckling example. The simulations have increasing spatial resolution from left to right.

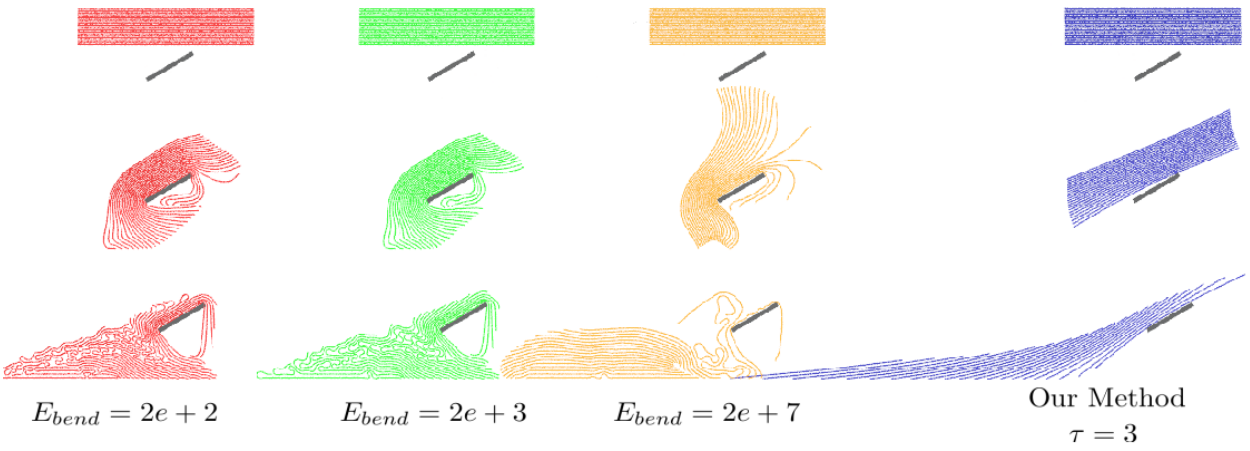


Figure 5.18: **Jiang et al. [25] comparison**. We demonstrate that only moderate bending is possible with the approach of Jiang et al. [25]. Our approach allows for a much wider range of bending resistance.

CHAPTER 6

Hybrid MPM with Frictional Contact

In this chapter, we present a new hybrid Lagrangian Material Point Method for simulating volumetric elastic objects like rubber and soft tissues that utilizes a Lagrangian mesh for internal force computation and an Eulerian mesh for self collision as well as coupling with external materials. While recent Material Point Method (MPM) techniques allow for natural simulation of hyperelastic materials represented with Lagrangian meshes, they utilize an updated Lagrangian discretization where the Eulerian grid degrees of freedom are used to take variations of the potential energy. This often coarsens the degrees of freedom of the Lagrangian mesh and can lead to artifacts. We develop a hybrid approach that retains Lagrangian degrees of freedom while still allowing for natural coupling with other materials simulated with traditional MPM, e.g. sand, snow, etc. Furthermore, while the techniques developed in Chapter 5 allow for resolution of frictional contact with simulation of thin shells, they do not generalize to the case of volumetric materials. In this chapter, we show that our hybrid approach resolves these issues. We demonstrate the efficacy of our technique with examples that involve elastic soft tissues coupled with kinematic skeletons, extreme deformation, and coupling with multiple elastoplastic materials. Our approach also naturally allows for two-way rigid body coupling.

6.0.1 Continuous formulation

For volumetric elastic objects, we adopt the fixed corotational model from [41], though any hyperelastic potential may be used. With this choice, the stress satisfies

$$\begin{aligned} \psi(\mathbf{F}) &= \mu \sum_i (\sigma_i - 1)^2 + \frac{\lambda}{2}(J-1)^2, \\ \mathbf{P} &= \mu(\mathbf{F} - \mathbf{R}) + \lambda(J-1)J\mathbf{F}^{-T}. \end{aligned} \tag{6.1}$$

Here μ and λ are the Lamé coefficients that express the material resistance for deformation and volume change, and σ_i are the singular values of the deformation gradient $\mathbf{F}$ computed according to the polar SVD convention of [24] to allow for extreme deformation.

6.1 Discretization: Hyperelastic Solids

Our hybrid approach utilizes aspects of traditional Finite Element Methods (FEM) for hyperelasticity [36]. However, our approach is largely motivated by the the MPM treatment of volumetric objects from Jiang et al. [27] and Zhu et al. [48]. These methods were originally designed to prevent the numerical fracture that would occur with volumetric objects in traditional particle-based MPM. We first discuss this approach and how it resolves self collision, followed by its drawbacks.

In Jiang et al. [27] and Zhu et al. [48], the state at time t^n consists of particles with positions $\mathbf{x}_p^n$ connected with a tetrahedron mesh with elements indexed by e, as in Lagrangian FEM. Furthermore, particles store velocities $\mathbf{v}_p^n$ and masses m_p. The MPM time step from time t^n to t^{n+1} consists of three steps: (1) mass (m_p) and momentum ($m_p\mathbf{v}_p^n$) are transferred from particles to the grid using weights ($w_{\mathbf{i}p}^n = N(\mathbf{x}_p^n - \mathbf{x}_\mathbf{i})$) that describe the degree of interaction between particle p and grid node $\mathbf{i}$ and which are defined by Eulerian grid interpolation functions $N(\mathbf{x})$, (2) the grid momentum ($m_\mathbf{i}^n\mathbf{v}_\mathbf{i}^n$) is updated in a variational way from the potential energy in the system and finally, (3) the motion of the grid under the updated momentum is interpolated to the particles. The process of updating the grid momentum in step (2) uses the updated Lagrangian [1, 28, 18] convention where the time t^n configuration serves as the reference, rather than the $t = 0$ configuration in a Lagrangian

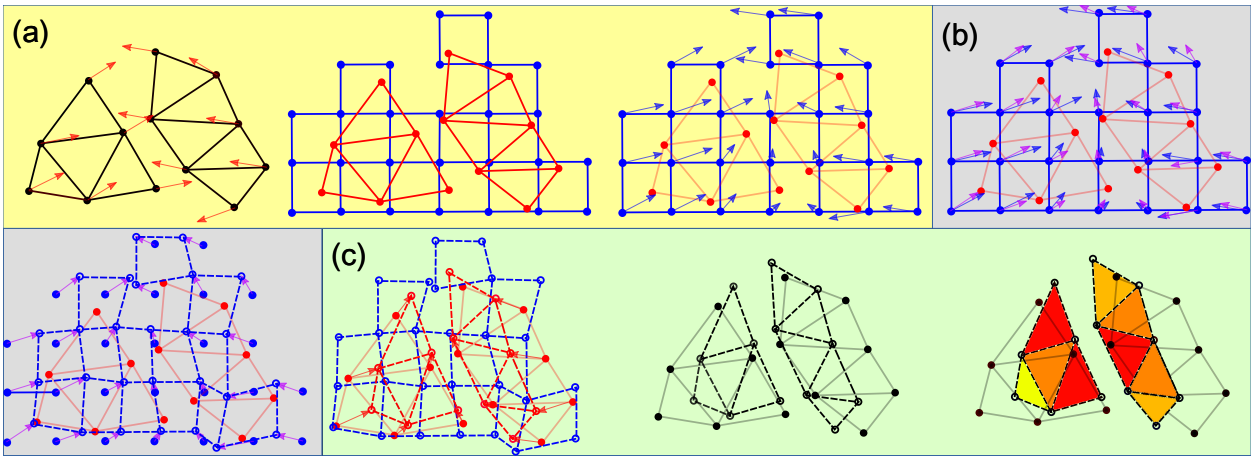


Figure 6.1: **MPM Overview**. The steps in the MPM update are: (a) The Lagrangian quantities (black and red) are transferred to an Eulerian grid (blue), which may be viewed as a new FEM mesh. (b) Grid nodes receive new velocities (purple) from updated Lagrangian elastic updates and are temporarily moved with those velocities. (c) The Lagrangian quantities are updated by interpolating from the new positions and velocities of the Eulerian grid nodes. The triangles are colored based on the amount of compression.

discretization. With this updated Lagrangian convention, the particles $\mathbf{x}_p^n$ are moved by the grid via interpolation $\mathbf{x}_p^{n+1} = \sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}^{n+1} w_{\mathbf{i}p}^n$, and they change the potential energy via the per-element deformation gradient computed as in standard FEM (see Equation (6.2)). The grid node vertices $\mathbf{x}_{\mathbf{i}}$, which are allowed to move temporarily as $\mathbf{x}_{\mathbf{i}}^{n+1} = \mathbf{x}_{\mathbf{i}} + \Delta t \mathbf{v}_{\mathbf{i}}^{n+1}$, serve as degrees of freedom. When the spatial discretization is done variationally from the potential energy, this step is almost identically what is done in a Lagrangian FEM discretization of elastoplasticity [36]. In this sense, the method can be interpreted as continually remeshing the domain of the material, where the transfer process in step (1) is all that is needed to define the mesh at a given time step (see Figure 6.1).

The MPM update only considers the variation of the potential energy with respect to grid degrees of freedom; nothing explicit is done to model self collision. Self collision is modeled as if it were an elastic phenomenon, and by virtue of switching between particle and grid representations. We describe these two aspects of collision resolution as **type (i)** and **type (ii)**.

Type (i) The grid transfers in step (1) ultimately remesh the domain (see Figure 6.1). By transferring to the grid, and using an updated Lagrangian formulation where the grid nodes are updated based on the variation of the potential energy in Equation (6.1), MPM essentially uses a new FEM mesh (blue in Figure 6.1) to calculate the elastic update. This process creates new connections in the updated Lagrangian mesh and once they are made, collision inducing modes are penalized via the potential energy in the system (see Figure 6.1). For example, collision trajectories of the particles will induce compression in elements of the Eulerian grid which would be penalized from the elastic potential in the system.

Type (ii) In particle systems, collisions occur because of discontinuities in the velocity, e.g. consider two particles next to each other with opposing velocities. Transferring to and from the grid smooths the particle velocities, which ultimately prevents collision. Since the motion of the Eulerian grid after the momentum update in step (2) is interpolated to the particles using continuous interpolating functions, particle collisions cannot occur as long as the Eulerian mesh is not tangled by the motion. This can be guaranteed with a CFL restriction since the tangling is a temporal discretization artifact. In fact, an updated Lagrangian MPM simulation with no constitutive model on the particles at all can still prevent material collision, simply by virtue of the **type (ii)** interactions (see Figure 6.2).

These modes of collision resolution are simplistic, but limited by several drawbacks. For volumetric objects, the **type (i)** interactions are unable to regulate the potential energy with a plasticity model derived from Coulomb friction as in [25, 19]. The mesh is volumetric and therefore does not have the flexibility of codimension that can be used to model contact through the continuum. There are no directions left for plastic flow of the type designed in [25] that could be used to satisfy the Coulomb friction stress constraints. This can lead to unregulated resistance to shearing and cohesion as the elastic potential will still increase with these modes, even though that is not consistent with Coulomb friction (see Figure 6.5). Furthermore, the updated Lagrangian treatment of the stress-based momentum leads to

visual interaction at a distance and persistent wrinkling when the grid resolution is too low [27, 15, 20]. Additionally, when the grid resolution is too high, **type (i)** and **type (ii)** interactions have no effect and the method does not prevent collision (see Figure 6.2). To prevent this, the Lagrangian mesh resolution must be about the same as the Eulerian grid resolution. This is sub-optimal when a coarse Lagrangian mesh suffices to resolve deformation.

6.1.1 Hybrid Lagrangian MPM for elastic solids

Our method is designed by abandoning the **type (i)** collision prevention for volumetric meshes and the updated Lagrangian integration of the elastic forces in general. Instead we use a splitting approach where elastic forces are applied in a Lagrangian way, and **type (ii)** interactions are integrated by MPM with no elastic force computation. We achieve this by introducing collision particles $\mathbf{x}_q^n$ which are sampled uniformly at random on the boundary of the volumetric elastic mesh. The mass of the collision particle m_q is found by dividing the mass of the boundary element by the number of collision particles on that element. These particles are not true degrees of freedom and are tied to the mesh during the Lagrangian update. They are then used to generate **type (ii)** collision prevention. We show that their response defines a type of impulse that can be regulated by Coulomb friction and applied to the mesh at the end of the time step. Furthermore, because the collision particles can be sampled at a density proportional to the grid spacing, we show that they remove the effect of grid resolution on collision resolution (see Figure 6.3).

Our approach uses the same discrete state as in the standard MPM described in Chapter 3: time t^n, particle positions $\mathbf{x}_p^n$ connected with a tetrahedron mesh, velocities $\mathbf{v}_p^n$, and masses m_p. In addition, we store the collision particles $\mathbf{x}_q^n$ sampled on the boundary of the tetrahedron mesh. We summarize essential steps in the algorithm for updating our discrete state to time t^{n+1} below.

1. **Lagrangian update:** Update particle velocities from potential-energy-based and body forces, and interpolate velocities to collision particles. §6.1.2

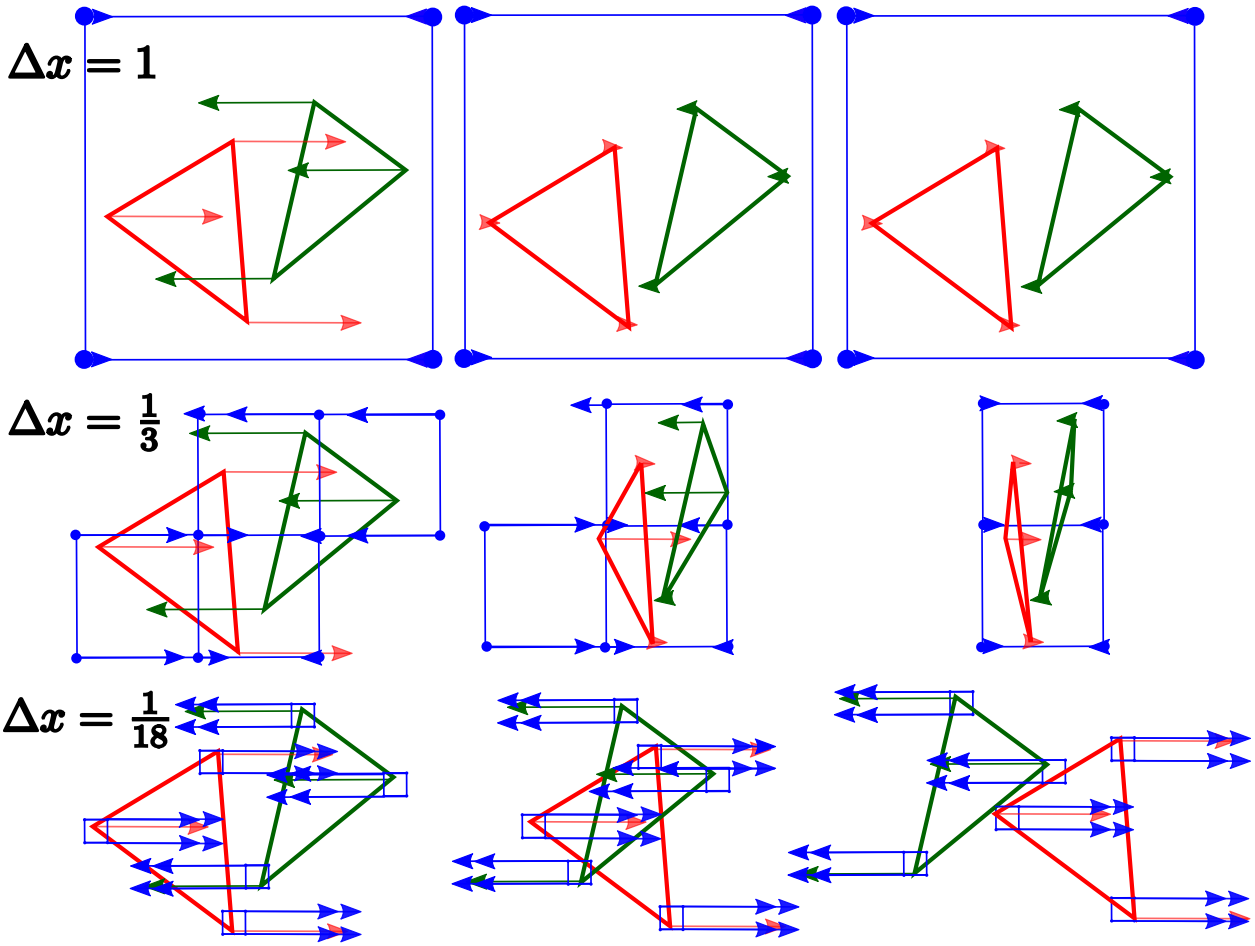


Figure 6.2: **Type (ii) interactions with different Δx, columns indicating consecutive time steps**. At appropriate grid resolution (middle row), MPM prevents material collision even without constitutive model. However, when the grid resolution is too low (top row), objects are separated at a distance, and when the grid resolution is too high (bottom row), the MPM grids may miss a collision.

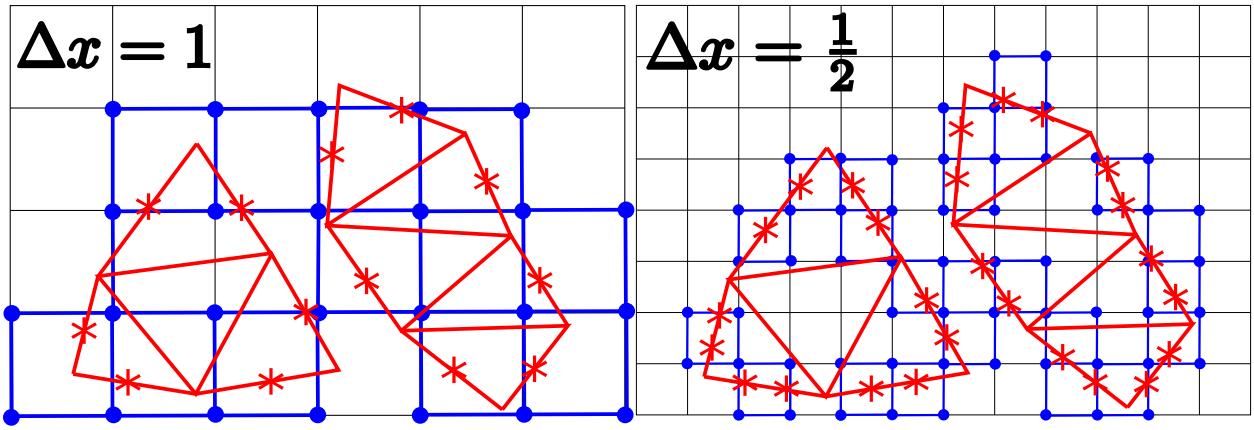


Figure 6.3: **Collision particles**. Sampling density based on Eulerian grid Δx.

2. **Transfer to grid:** Transfer mass and momentum from collision particles to grid. §6.1.3.1

3. **Transfer to collision particles**: Transfer velocities from grid back to collision particles. §6.1.3.2

4. **Apply impulses:** Calculate the impulse applied to each boundary mesh using the velocity change in collision particles and update velocities of particles on the boundary mesh.

5. **Update positions:** Update particle positions and elastic states. §6.1.5.

6.1.2 Lagrangian update

We consider the case of piece-wise linear interpolation over a tetrahedron mesh. The deformation gradient varies in a piece-wise constant manner with each element, which we denote as $\mathbf{F}_e$. With this convention, the FEM force per particle $\mathbf{f}_p$ can be seen as the negative

gradient of the the total potential energy Ψ with respect to particle positions:

$$\mathbf{F}_e(\mathbf{x}) = \sum_p \mathbf{x}_p \frac{\partial \tilde{N}_p}{\partial \mathbf{X}}(\mathbf{X}_e) \tag{6.2}$$

$$\Psi(\mathbf{x}) = \sum_e \psi(\mathbf{F}_e(\mathbf{x}))V_e^0 \tag{6.3}$$

$$\mathbf{f}_p(\mathbf{x}) = -\sum_e \frac{\partial \psi}{\partial \mathbf{F}}(\mathbf{F}_e(\mathbf{x})) : \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_p}(\mathbf{x})V_e^0 \tag{6.4}$$

$$= -\sum_e \mathbf{P}(\mathbf{F}_e(\mathbf{x}))\frac{\partial \tilde{N}_p}{\partial \mathbf{X}}V_e^0. \tag{6.5}$$

Here $\mathbf{x} \in \mathbb{R}^{3n_p}$ refers to the vector of all particles $\mathbf{x}_p$, where n_p is the total number of particles, Ψ is the total potential energy which is a sum of tetrahedron element contributions $\psi(\mathbf{F}_e)V_e^0$, where ψ is the potential energy density in Equation (6.1), V_e^0 is the volume of the element in the initial state, $\tilde{N}_p$ is the piece-wise linear interpolating function associated with particle $\mathbf{x}_p$, and $\mathbf{X}_e$ is the tetrahedron barycenter in the time $t = 0$ configuration. We refer the reader to Sifakis and Barbic [36] for a more detailed derivation.

The FEM update uses the usual Lagrangian view of the governing physics. The internal force is the negative gradient of the potential energy in Equation (6.5). Particle velocities are updated according to forces computed at particle positions $\mathbf{x}_p^{n+\alpha}$, where symplectic Euler integration corresponds to $\alpha = 0$ and backward Euler corresponds to $\alpha = 1$:

$$\mathbf{v}_p^* = \mathbf{v}_p^n + \Delta t \frac{\mathbf{f}_p(\mathbf{x}^{n+\alpha})}{m_p}. \tag{6.6}$$

When damping is required while using symplectic Euler integration, we construct a background Eulerian grid with Δx comparable to the mesh size and transfer the velocity to and then back from the grid using APIC with RPIC damping as described in [25]. We can even perform the transfers multiple times when more damping is desired. For interior particles, $\mathbf{v}_p^{n+1} = \mathbf{v}_p^*$. On the other hand, for particles on the boundary mesh, we interpolate their velocities and positions to collision particles using

$$\mathbf{v}_q^* = \sum_p b_{pq}\mathbf{v}_p^* \tag{6.7}$$

$$\mathbf{x}_q^n = \sum_p b_{pq}\mathbf{x}_p^n \tag{6.8}$$

where b_{pq} is the barycentric weight of the point q relative to p. We also assign to each point q an outward normal vector $\mathbf{n}_q$ inherited from the face of the mesh that q is tied to.

6.1.3 Grid transfers

6.1.3.1 Particle to Grid

To process collision and contact, we transfer mass and momentum from collision particles $\mathbf{x}_q^n$ to grid nodes $\mathbf{x_i}$ using standard MPM transfers

$$m_{\mathbf{i}}^n = \sum_q w_{\mathbf{i}q}^n m_q \tag{6.9}$$

$$\mathbf{v}_{\mathbf{i}}^* = \frac{1}{m_{\mathbf{i}}^n} \sum_q w_{\mathbf{i}q}^n m_q \mathbf{v}_q^*. \tag{6.10}$$

Here $w_{\mathbf{i}q}^n = N(\mathbf{x}_q^n - \mathbf{x_i})$ is the weight of interaction between particle $\mathbf{x}_q^n$ and grid node $\mathbf{x_i}$, as in standard MPM.

6.1.3.2 Grid to Particle

Without any constitutive model on the grid, we proceed directly to the grid to particle step. The grid to particle transfer defines the velocity local to collision particle $\mathbf{x}_q^n$ in terms of $\mathbf{v}_q^\star$ from

$$\mathbf{v}_q^\star = \sum_{\mathbf{i}} w_{\mathbf{i}q}^n \mathbf{v}_{\mathbf{i}}^*. \tag{6.11}$$

6.1.4 Apply impulse

Since the velocity $\mathbf{v}_q^\star$ is interpolated from an updated Lagrangian background grid, the boundary of the mesh is safe from self-intersection if it is moved with $\mathbf{v}_q^\star$. However, the change may not be consistent with a Coulomb friction interaction, and the response can even be cohesive. In the case of a cohesive response after collision, we reject the change.

That is, when

$$\mathbf{v}_r = \mathbf{v}_q^\star - \mathbf{v}_q^* \tag{6.12}$$

$$\mathbf{v}_r \cdot \mathbf{n}_q \geq 0 \tag{6.13}$$

the updated Lagrangian mesh detects a separation instead of collision, and the collision particle keeps the velocity from the FEM update $\mathbf{v}_q^*$. On the other hand, if

$$\mathbf{v}_r \cdot \mathbf{n}_q < 0 \tag{6.14}$$

we apply an elastic impulse $I_q \mathbf{n}_q$ to the mesh at position $\mathbf{x}_q^n$ where $I_q = 2m_q \mathbf{v}_r \cdot \mathbf{n}_q$. We also allow for friction using Coulomb's model with the friction parameter μ. When an elastic impulse of magnitude I_q would be applied based on condition (6.14), Coulomb friction admits a change in magnitude of tangential velocity of at most $-\mu \frac{I_q}{m_q}$. So the combined velocity change on collision particle q is then

$$\Delta \mathbf{v}_q = \frac{I_q \mathbf{n}_q}{m_q} + \min\left(\|\mathbf{v}_t\|, -\mu \frac{I_q}{m_q}\right) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|}, \tag{6.15}$$

where $\mathbf{v}_t = \mathbf{v}_r - \mathbf{v}_r \cdot \mathbf{n}_q \mathbf{n}_q$. We then transfer this change to the particles p as

$$\Delta \mathbf{v}_p = \mathbf{v}_p^{n+1} - \mathbf{v}_p^* = \sum_q \tilde{b}_{pq} \Delta \mathbf{v}_q \tag{6.16}$$

where

$$\tilde{b}_{pq} = \frac{b_{pq} m_q}{\sum_r b_{pr} m_r} \tag{6.17}$$

are the normalized weights defined from the barycentric weights used to transfer from particles to collision particles.

6.1.5 Update positions and elastic state

For boundary particles, we adopt symplectic Euler time integration

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta \mathbf{v}_p \tag{6.18}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} \tag{6.19}$$

For interior particles, the update is in accordance with either symplectic Euler or backward Euler, depending on the choice of α in Equation (6.6):

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^* \tag{6.20}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}. \tag{6.21}$$

6.2 Rigid bodies

Two-way rigid body coupling may be achieved with a treatment similar to volumetric elastic objects. We sample collision particles on the boundary in the same fashion as in §6.1.1 and then uniformly distribute the mass of the rigid body to the collision particles. However, we found that unlike for volumetric elastic objects, **type (ii)** interactions on the grid alone are not enough to resolve collisions. Instead we endow the collision particles with the potential described in [25, 19] to penalize contact. Specifically, we update the deformation gradient $\mathbf{F}_q$ from time t_n to t_{n+1} in the following way. Let $\mathbf{x}_\alpha$ and $\mathbf{X}_\alpha$, $\alpha \in \{0, 1, 2\}$ be the current and initial positions of the vertices of the triangle that collision particle q is tied to. Let $\mathbf{D}_{q,\beta} = \mathbf{X}_\beta - \mathbf{X}_0$ be the undeformed mesh element edge vectors (where $\beta = 1, 2$), and $\hat{\mathbf{d}}^E_{q,\beta} = \mathbf{x}^n_\beta - \mathbf{x}^n_0$ be the deformed edge vectors. We choose each $\mathbf{D}_3$ to be unit-length and normal to $\mathbf{D}_1$ and $\mathbf{D}_2$, and evolve each one as in traditional MPM via $\hat{\mathbf{d}}^E_{q,3} = \boldsymbol{\nabla}\mathbf{x}_q \mathbf{d}^E_{q,3}$. Then $\hat{\mathbf{F}}^E_q = \hat{\mathbf{d}}^E_q \mathbf{D}_q^{-1}$. Following [25, 19], we let $\hat{\mathbf{F}}^E_q = \mathbf{Q}\hat{\mathbf{R}}$ be the QR decomposition of $\hat{\mathbf{F}}^E_q$ and design a collision energy density $\psi(\hat{\mathbf{R}}) = f(\hat{\mathbf{R}}) + g(\hat{\mathbf{R}})$,

$$f(\hat{\mathbf{R}}) = \begin{cases} \frac{k^c}{3}(1 - \hat{r}_{33})^3 & 0 \le \hat{r}_{33} \le 1 \\ 0 & \hat{r}_{33} > 1 \end{cases}, \; g(\hat{\mathbf{R}}) = \frac{\gamma}{2}(\hat{r}^2_{13} + \hat{r}^2_{23}) \tag{6.22}$$

where $\hat{r}_{ij}$ is the ij-th entry of $\hat{\mathbf{R}}$. We resolve the force which is the negative derivative of this energy on the MPM background grid, and we refer the reader to [27, 28] for more details. Plasticity is then applied according to [25, 19] to give $\mathbf{R}$

$$r_{33} = \begin{cases} \hat{r}_{33} & 0 < \hat{r}_{33} \le 1 \\ 1 & \hat{r}_{33} > 1 \end{cases}, \; r_{\beta 3} = h(\hat{r}_{13}, \hat{r}_{23}, r_{33})\hat{r}_{\beta 3} \tag{6.23}$$

$$h(\hat{r}_{13}, \hat{r}_{23}, r_{33}) = \min\left(1, \frac{c_F k^c \left(1 - r_{33}\right)^2}{\gamma\sqrt{\hat{r}_{13}^2 + \hat{r}_{23}^2}}\right) \tag{6.24}$$

Finally, we update the deformation gradient with $\mathbf{F}_q^{n+1} = \mathbf{QR}$.

Let $\mathbf{v}_q^* = \sum_{\mathbf{i}} w_{\mathbf{i}q}^n \mathbf{v}_{\mathbf{i}}^*$, where $\mathbf{v}_{\mathbf{i}}^*$ is the grid velocity after the MPM force update, and let $\mathbf{v}_r = \mathbf{v}_q^* - \mathbf{v}_q$. If $\mathbf{v}_r \cdot \mathbf{n}_q < 0$, we apply an impulse $\mathbf{I}_q$ to the rigid bodies to update velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega}$ via

$$I_q = m_q \mathbf{v}_r \cdot \mathbf{n}_q \tag{6.25}$$

$$\mathbf{v}_t = \mathbf{v}_r - \mathbf{v}_r \cdot \mathbf{n}_q \mathbf{n}_q \tag{6.26}$$

$$\mathbf{I}_q = I_q \mathbf{n}_q + m_q \min\left(\|\mathbf{v}_t\|, -\mu\frac{I_q}{m_q}\right)\frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \tag{6.27}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \sum_q \frac{\mathbf{I}_q}{m_q} \tag{6.28}$$

$$\boldsymbol{\omega}^{n+1} = \boldsymbol{\omega}^n + \sum_q \mathbf{J}^{-1}(\mathbf{r} \times \mathbf{I}_q) \tag{6.29}$$

where $\mathbf{r}$ is the vector from the rigid body's center of mass to the application point of the impulse, and $\mathbf{J}$ is the inertia tensor.

6.3 Coupling with traditional MPM

Our method easily couples with traditional MPM particles such as snow, sand and clothing. To prevent numerical cohesion between phases common to MPM, we adopt two separate background MPM grids, one for volumetric elastic and rigid objects, and the other for general MPM materials. We denote quantities associated with the two grids with subscripts 1 and 2 respectively. We denote quantities associated with traditional MPM particles with

subscript p and quantities associated with quadrature points with subscript q. So we have,

$$m^n_{1,\mathbf{i}} = \sum_q w^n_{\mathbf{i}q} m_q, m^n_{2,\mathbf{i}} = \sum_p w^n_{\mathbf{i}p} m_p \tag{6.30}$$

$$\mathbf{v}^*_{1,\mathbf{i}} = \frac{1}{m^n_{1,\mathbf{i}}} \sum_q w^n_{\mathbf{i}q} m_q \mathbf{v}^*_q \tag{6.31}$$

$$\mathbf{v}^n_{2,\mathbf{i}} = \frac{1}{m^n_{2,\mathbf{i}}} \sum_p w^n_{\mathbf{i}p} m_p \left(\mathbf{v}_p + \mathbf{C}_p(\mathbf{x}_\mathbf{i} - \mathbf{x}_p)\right) \tag{6.32}$$

$$\mathbf{n}^n_\mathbf{i} = \frac{\sum_q w_{\mathbf{i}q} \mathbf{n}_q}{\| \sum_q w_{\mathbf{i}q} \mathbf{n}_q \|} \tag{6.33}$$

Grid velocity $\mathbf{v}^n_{2,\mathbf{i}}$ is updated as in [27, 28] to get $\mathbf{v}^*_{2,\mathbf{i}}$. Then the collision between phases is handled through an inelastic collision on collocated grid nodes.

$$\mathbf{v}_r = \mathbf{v}^*_{1,\mathbf{i}} - \mathbf{v}^*_{2,\mathbf{i}} \tag{6.34}$$

$$\mathbf{v}_t = \mathbf{v}_r - \mathbf{v}_r \cdot \mathbf{n}^n_\mathbf{i} \mathbf{n}^n_\mathbf{i} \tag{6.35}$$

$$I_\mathbf{i} = \max\left(\frac{m^n_{2,\mathbf{i}} m^n_{1,\mathbf{i}}}{m^n_{2,\mathbf{i}} + m^n_{1,\mathbf{i}}} \mathbf{v}_r \cdot \mathbf{n}^n_\mathbf{i}, 0\right) \tag{6.36}$$

$$\mathbf{v}^{**}_{1,\mathbf{i}} = \mathbf{v}^*_{1,\mathbf{i}} - \frac{I_\mathbf{i} \mathbf{n}_\mathbf{i}}{m^n_{1,\mathbf{i}}} - \min\left(\frac{\mu I_\mathbf{i}}{m^n_{1,\mathbf{i}}}, \|\mathbf{v}_t\|\right) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \tag{6.37}$$

$$\mathbf{v}^{n+1}_{2,\mathbf{i}} = \mathbf{v}^*_{2,\mathbf{i}} + \frac{I_\mathbf{i} \mathbf{n}_\mathbf{i}}{m^n_{2,\mathbf{i}}} + \min\left(\frac{\mu I_\mathbf{i}}{m^n_{2,\mathbf{i}}}, \|\mathbf{v}_t\|\right) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \tag{6.38}$$

Finally, we interpolate the the grid velocity $\mathbf{v}^{n+1}_{2,\mathbf{i}}$ to MPM particles with APIC as in [27, 28], and Equation (6.11) is replaced with

$$\mathbf{v}^\star_q = \sum_\mathbf{i} w^n_{\mathbf{i}q} \mathbf{v}^{**}_{1,\mathbf{i}}. \tag{6.39}$$

6.4 Results

We demonstrate the efficacy of our method with a number of representative examples that illustrate the dynamics of volumetric objects, and show that our method couples with granular materials, clothing and rigid bodies. We list the run-time performance for our examples in Table 6.1. All simulations were run on an Intel Xeon E5-2690 V2 system with 20 threads and 128GB of RAM. We report the timing in terms of average seconds of computation per

frame. We chose Δt in an adaptive manner that is restricted by a CFL condition when the particle velocities are high, i.e., we do not allow particles to move further than the CFL number times Δx in a time step.

6.4.1 Volumetric objects

We demonstrate the robustness of our method for resolving collisions between volumetric objects. Our method correctly resolves frictional sliding without artifacts. In Figure 6.4, we show a skin simulation with walking characters in various body shapes. In Figure 6.5 and Figure 6.6, we compare our approach with updated Lagrangian MPM, which exhibits excessive cohesion and numerical friction. We also show that our method removes the requirement of comparable grid and mesh resolution. We use a moderate resolution Lagrangian mesh to resolve the dynamics of the bunnies and a high resolution Eulerian grid to resolve more detailed behaviors of the sand. In contrast, updated Lagrangian MPM would require a high resolution Lagrangian mesh for bunnies in order to resolve collisions between phases. Furthermore, traditional MPM methods often have difficulties recovering from element inversions, as the particle modes needed to uninvert the material are lost in the transfers between particles and the grid due to the **type (ii)** interactions discussed in §6.1. On the other hand, our method handles extreme deformation and even element inversion as demonstrated in Figure 6.7. MPM fails to recover the original shape of the object when the grid resolution is low and **type (ii)** interactions are effective and exhibits high frequency noise when the grid resolution is too high for **type (ii)** interactions to be effective. On the other hand, the elastic object recovers its original shape with any grid resolutions using our method.

6.4.2 Coupling with MPM and rigid bodies

Our method also supports coupling with rigid bodies as well as traditional MPM particles such as snow, sand and clothing. In Figure 6.8, we demonstrate the coupling of soft tissues with clothing material simulated with MPM as in [25]. In Figure 6.9, elastic characters and a column of sand are poured on a series of pinwheels simulated as rigid bodies, setting them in

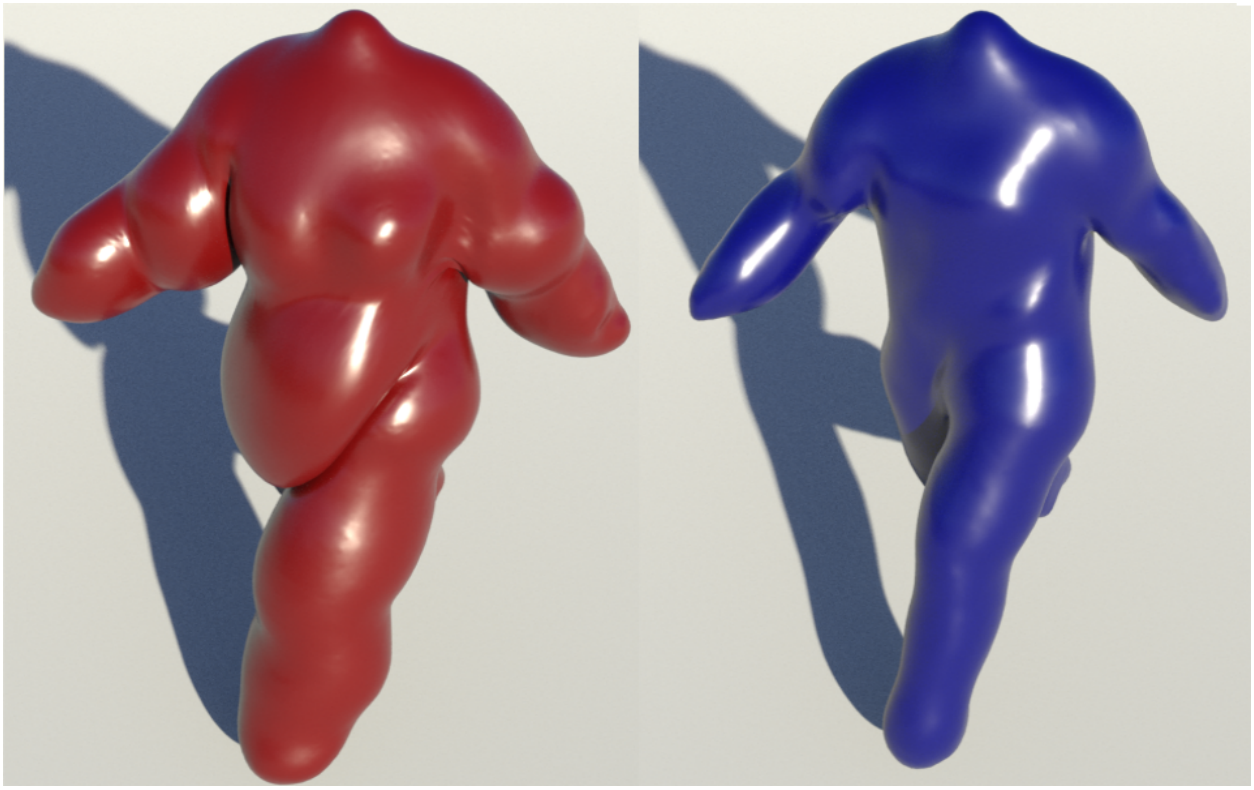


Figure 6.4: **Walking mannequins**. Our method handles the numerous collisions occurring in the scene with walking characters.

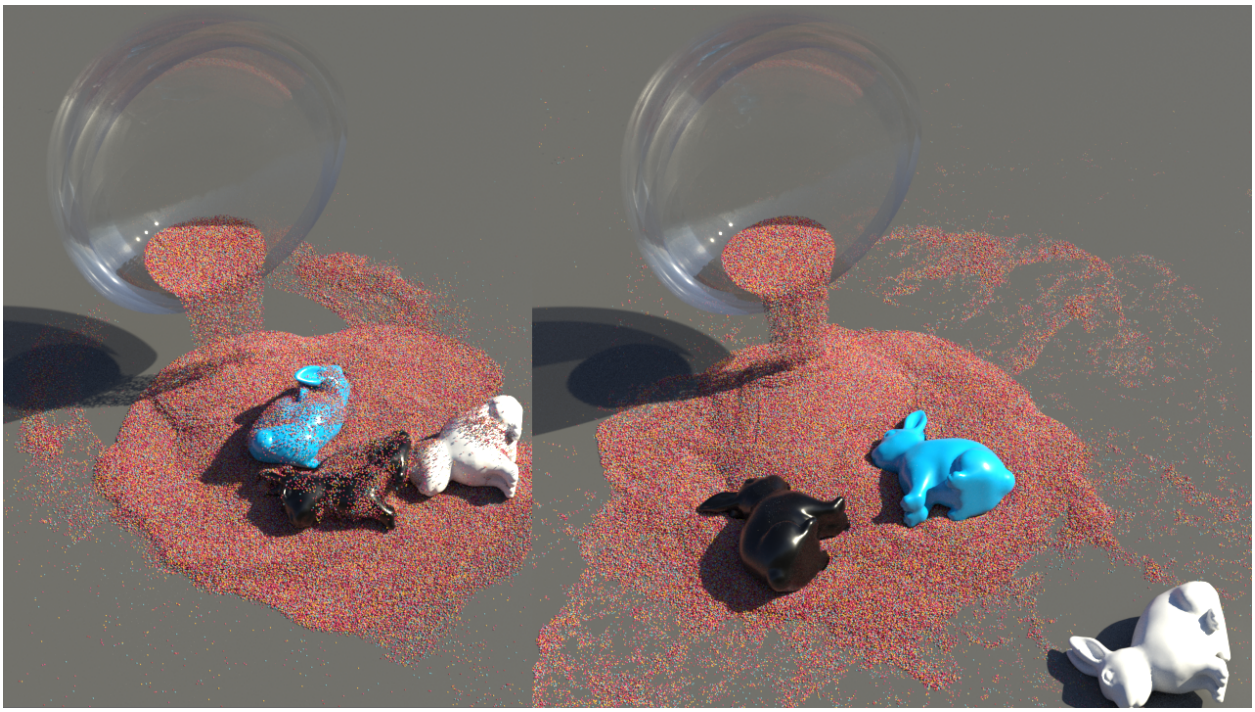


Figure 6.5: **Friction**. Our method (right) removes the excessive numerical friction common to traditional MPM (left), and regulates friction with the Coulomb friction model. With low friction coefficients, the colored sand freely slides off the bunnies.

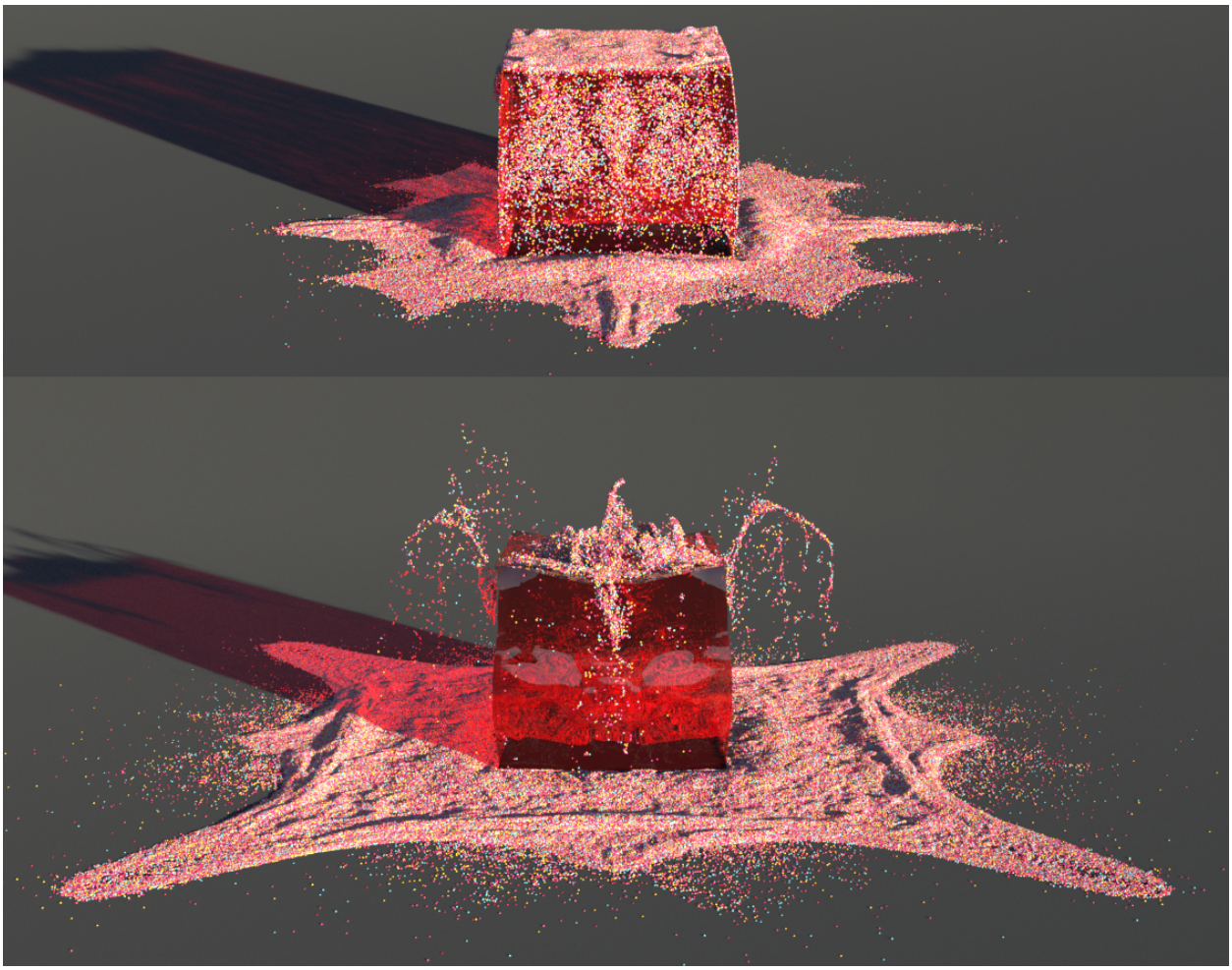


Figure 6.6: Comparison between MPM (top) and our method (bottom).

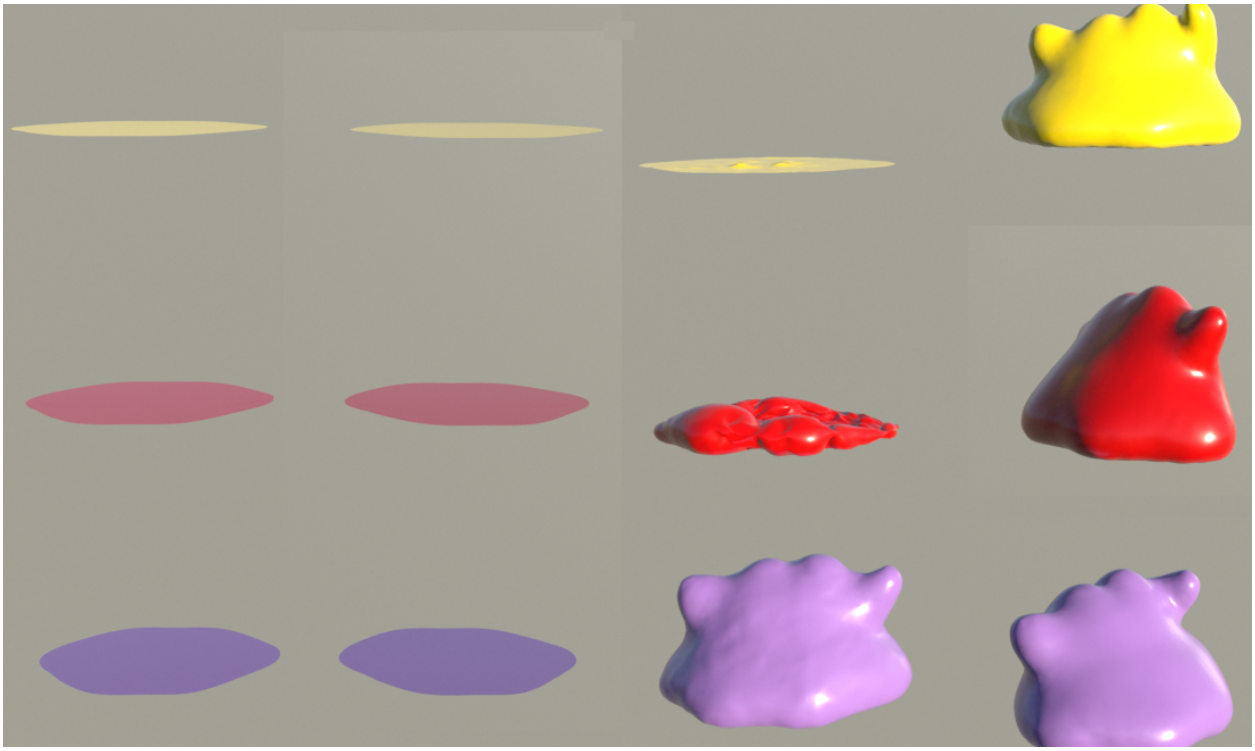


Figure 6.7: **Element inversion**. MPM (left) has difficulties when elements invert, especially with low grid resolution (yellow and red). Our method (right) handles element inversions with ease.

Figure 6.8: **Skin and shirt**. The skin of a mannequin is coupled with clothing simulated with MPM.

motion. In Figure 6.10, colored sand is poured on top of three Jell-O's with various stiffness, generating interesting patterns. In Figure 6.11, an elastic character walk through several walls made with elastic materials with different elastic stiffness simulated with MPM. The elastic walls numerically fracture under contact with the elastic character.

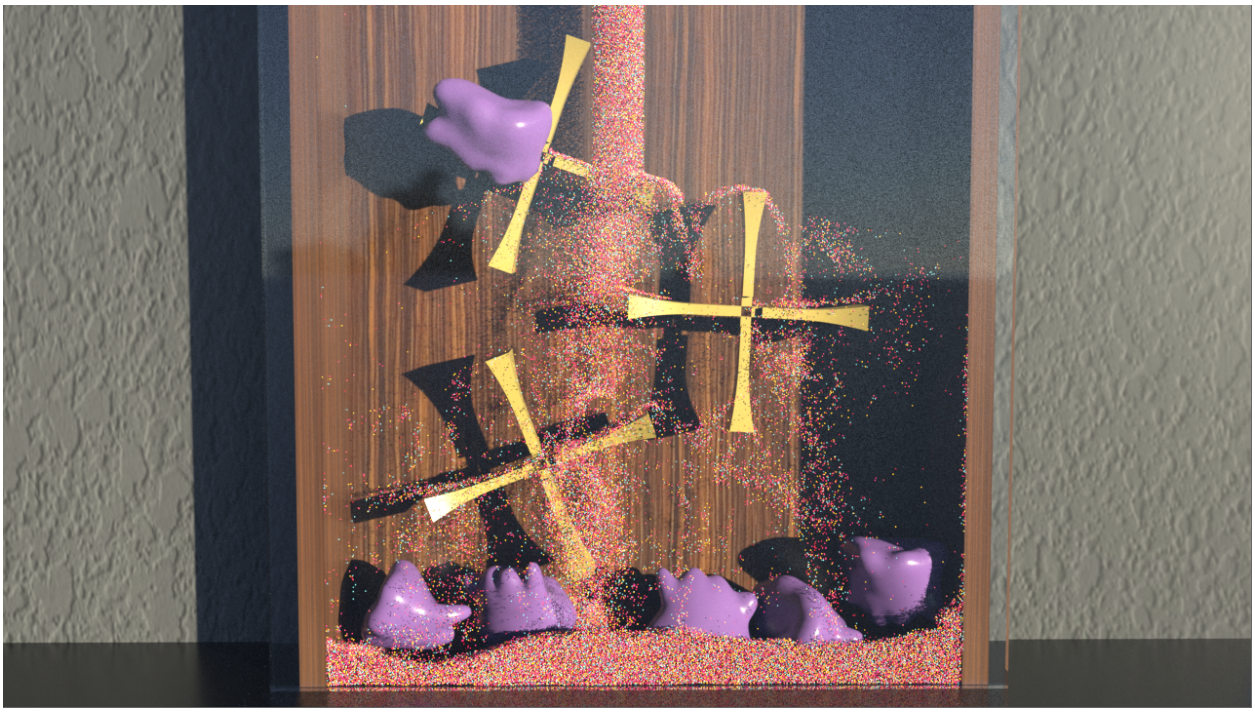


Figure 6.9: **Pinwheel**. Colored sand and elastic characters are poured into a cabinet, setting rigid pinwheels in motion.

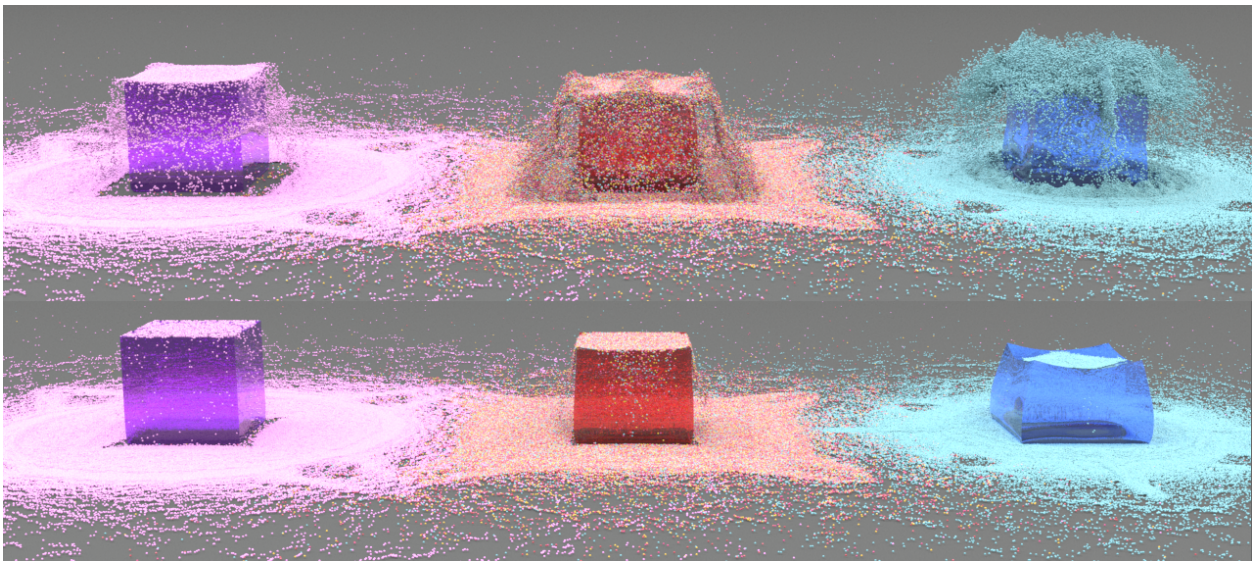


Figure 6.10: **MPM particle coupling**. Elastic Jell-O's with varying stiffness are two-way coupled with MPM particles.

Table 6.1: All simulations were run on an Intel Xeon E5-2690 V2 system with 20 threads and 128GB of RAM. Simulation time is measure in seconds per frame. Time spent on geometric collision per frame is recorded in the second entry of the timing column where applicable. Element # denotes number of segments for hair simulations and number of tetrahedra for volumetric simulations. Particle # denotes the total number of MPM particles, and the number of collision particles are recorded in the second entry where applicable.

	Time	Element #	Particle #	Δx	CFL
Mannequin (Fig. 6.4 left)	39	$933K$	$41K/41K$	0.05	0.6
Mannequin (Fig. 6.4 right)	27	$641K$	$31K/31K$	0.05	0.6
Pinwheel (Fig. 6.9)	89	$93K$	$930K/57K$	0.5	0.6
Jell-O (MPM) (Fig. 6.6 top)	73	$8.64M$	$5.41M$	0.005	0.6
Jell-O (Hybrid) (Fig. 6.6 bottom)	220	$1.08M$	$3.81M/172k$	0.005	0.6
Bunnies (MPM) (Fig. 6.5 left)	186	$3.97M$	$2.67M$	0.1	0.6
Bunnies (Hybrid) (Fig. 6.5 right)	66	$201K$	$1.99M/25K$	0.1	0.6
Skin and shirt (Fig. 6.8)	3	$207K$	$120K/40K$	0.006	0.6
"Kool-aid Man" (Fig. 6.11)	50	$933K$	$2.29M/41K$	0.05	0.6

Figure 6.11: **"Kool-aid Man"**. Simulation of a mannequin breaking through an elastic wall.

APPENDIX A

Supporting Calculations

A.1 FEM Force computation for thin shells

We compute forces on the control points $\mathbf{x}_p$ by

$$\begin{aligned}
\mathbf{f}_p^{KL} &= -\frac{\partial \Psi^S(\mathbf{F}^{KL,Etr}(\mathbf{x}^{KL}))}{\partial \mathbf{x}_p^{KL}} \\
&= -\sum_q V_q^0 \frac{\partial \psi(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}_q))}{\partial \mathbf{x}_p^{KL}} \\
&= -\sum_q V_q^0 \frac{\partial \psi}{\partial \mathbf{F}^{KL}}(\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}_q)) : \frac{\partial \mathbf{F}_q^{KL,E\mathrm{tr}}}{\partial \mathbf{x}_p^{KL}}(\mathbf{x}_q),
\end{aligned}$$

where $\mathbf{x}_q$'s are positions of the quadrature points. We give expressions for each $\frac{\partial \mathbf{F}_q^{KL}}{\partial x_{p_k}^{KL}}(\mathbf{x}_q)$ with fixed p, q and k, where k represents the x, y, or z direction. For simplicity of notation, we omit the subscripts p, q and superscript KL for now.

Recall from Chapter 5 that we have

$$\mathbf{F} = \sum_{i=1}^{3} \mathbf{g}_i \otimes \bar{\mathbf{g}}^i, \text{ with } \mathbf{g}_\alpha = \mathbf{a}_\alpha + \xi_3 \mathbf{a}_{3,\alpha}, \ \mathbf{g}_3 = \mathbf{a}_3,$$

where

$$\begin{aligned}
\mathbf{a}_\alpha &= \sum_j \mathbf{x}_j \frac{\partial N_j^{SD}}{\partial \xi_\alpha}(\xi_1, \xi_2), \quad \alpha = 1, 2 \\
\mathbf{a}_3 &= \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|} \\
\mathbf{a}_{3,\alpha} &= (\mathbf{I} - \mathbf{a}_3 \otimes \mathbf{a}_3) \frac{\mathbf{a}_{1,\alpha} \times \mathbf{a}_2 + \mathbf{a}_1 \times \mathbf{a}_{2,\alpha}}{|\mathbf{a}_1 \times \mathbf{a}_2|} \\
&= \tilde{\mathbf{a}} - \mathbf{a}_3(\mathbf{a}_3 \cdot \tilde{\mathbf{a}})
\end{aligned}$$

in which we define $\tilde{\mathbf{a}}$ to be

$$\tilde{\mathbf{a}} = \frac{\mathbf{a}_{1,\alpha} \times \mathbf{a}_2 + \mathbf{a}_1 \times \mathbf{a}_{2,\alpha}}{|\mathbf{a}_1 \times \mathbf{a}_2|}.$$

Now we compute $\frac{\partial \mathbf{F}}{\partial x_k}$.

$$\frac{\partial \mathbf{F}}{\partial x_k} = \sum_{i=1}^{3} \frac{\partial \mathbf{g}_i}{\partial x_k} \otimes \bar{\mathbf{g}}^i,$$

and

$$\frac{\partial \mathbf{g}_\alpha}{\partial x_k} = \frac{\partial \mathbf{a}_\alpha}{\partial x_k} + \xi_3 \frac{\partial \mathbf{a}_{3,\alpha}}{\partial x_k}$$
$$\frac{\partial \mathbf{g}_3}{\partial x_k} = \frac{\partial \mathbf{a}_3}{\partial x_k}$$

where

$$\frac{\partial \mathbf{a}_\alpha}{\partial x_k} = \frac{\partial N_k^{SD}(\xi_1, \xi_2)}{\partial \xi_\alpha} \mathbf{e}_k \text{ (summation convention does not apply here)} \tag{A.1}$$

$$\frac{\partial \mathbf{a}_3}{\partial x_k} = \frac{\frac{\partial \mathbf{a}_1}{\partial x_k} \times \mathbf{a}_2 + \mathbf{a}_1 \times \frac{\partial \mathbf{a}_2}{\partial x_k} - \frac{|\mathbf{a}_1 \times \mathbf{a}_2|}{\partial x_k} \mathbf{a}_3}{|\mathbf{a}_1 \times \mathbf{a}_2|},$$

and

$$\frac{|\mathbf{a}_1 \times \mathbf{a}_2|}{\partial x_k} = \mathbf{a}_3 \cdot (\frac{\partial \mathbf{a}_1}{\partial x_k} \times \mathbf{a}_2 + \mathbf{a}_1 \times \frac{\partial \mathbf{a}_2}{\partial x_k})$$

Finally,

$$\frac{\partial \mathbf{a}_{3,\alpha}}{\partial x_k} = \frac{\partial \tilde{\mathbf{a}}}{\partial x_k} - \mathbf{a}_3 (\frac{\partial \mathbf{a}_3}{\partial x_k} \cdot \tilde{\mathbf{a}} + \mathbf{a}_3 \cdot \frac{\partial \tilde{\mathbf{a}}}{\partial x_k}) - \frac{\partial \mathbf{a}_3}{\partial x_k} (\mathbf{a}_3 \cdot \tilde{\mathbf{a}}),$$

where

$$\frac{\partial \tilde{\mathbf{a}}}{\partial x_k} = \frac{\frac{\mathbf{a}_{1,\alpha}}{\partial x_k} \times \mathbf{a}_2 + \mathbf{a}_{1,\alpha} \times \frac{\partial \mathbf{a}_2}{\partial x_k} + \frac{\partial \mathbf{a}_1}{\partial x_k} \times \mathbf{a}_{2,\alpha} + \mathbf{a}_1 \times \frac{\partial \mathbf{a}_{2,\alpha}}{\partial x_k}}{|\mathbf{a}_1 \times \mathbf{a}_2|} - \frac{\mathbf{a}_{1,\alpha} \times \mathbf{a}_2 + \mathbf{a}_1 \times \mathbf{a}_{2,\alpha}}{|\mathbf{a}_1 \times \mathbf{a}_2|^2} \frac{\partial |\mathbf{a}_1 \times \mathbf{a}_2|}{\partial x_k},$$

in which

$$\frac{\partial \mathbf{a}_{\alpha,\beta}}{\partial x_k} = \frac{N_k^{SD}(\xi_1, \xi_2)}{\partial \xi_\beta \partial \xi_\alpha} \mathbf{e}_k \text{ (summation convention does not apply here).}$$

A.2 Grid force computation for thin shells

The force on the MPM grid $f_{\mathbf{i}}^{iii}(\mathbf{x}^*)$ computes as follows:

$$\begin{aligned}
\mathbf{f}_{\mathbf{i}}^{(iii)}(\mathbf{x}^*) &= \sum_{p\in\mathcal{I}^{(iii)}} \frac{\partial\chi(\mathbf{a}_{p\alpha}\otimes\bar{\mathbf{a}}_{p\alpha}+\mathbf{a}_{p3}^E\otimes\bar{\mathbf{a}}_{p3})}{\partial\mathbf{x}_{\mathbf{i}}}V_p^0 \\
&= \sum_{p\in\mathcal{I}^{(iii)}} \frac{\partial\chi(\mathbf{a}_{p\alpha}\otimes\bar{\mathbf{a}}_{p\alpha}+\mathbf{a}_{p3}^E\otimes\bar{\mathbf{a}}_{p3})}{\partial\mathbf{F}^E} : \frac{\partial\left(\mathbf{a}_{p\alpha}\otimes\bar{\mathbf{a}}_{p\alpha}+\mathbf{a}_{p3}^E\otimes\bar{\mathbf{a}}_{p3}\right)}{\partial\mathbf{a}_{p\beta}} : \frac{\partial\mathbf{a}_{p\beta}}{\mathbf{x}_{\mathbf{i}}}V_p^0 \\
&+ \sum_{p\in\mathcal{I}^{(iii)}} \frac{\partial\chi(\mathbf{a}_{p\alpha}\otimes\bar{\mathbf{a}}_{p\alpha}+\mathbf{a}_{p3}^E\otimes\bar{\mathbf{a}}_{p3})}{\partial\mathbf{F}^E} : \frac{\partial\left(\mathbf{a}_{p\alpha}\otimes\bar{\mathbf{a}}_{p\alpha}+\mathbf{a}_{p3}^E\otimes\bar{\mathbf{a}}_{p3}\right)}{\partial\mathbf{a}_{p3}^E} : \frac{\partial\mathbf{a}_{p3}^E}{\mathbf{x}_{\mathbf{i}}}V_p^0.
\end{aligned}$$

Then, omitting the subscript p, we compute each term in the contraction:

$$\begin{aligned}
\frac{\partial\chi(\mathbf{a}_\alpha\otimes\bar{\mathbf{a}}_\alpha+\mathbf{a}_3^E\otimes\bar{\mathbf{a}}_3)}{\partial\mathbf{F}^E} &= \boldsymbol{\tau}^S\left(\mathbf{a}_\alpha\otimes\bar{\mathbf{a}}_\alpha+\mathbf{a}_3^E\otimes\bar{\mathbf{a}}_3)\right)^{-T} \\
&= \boldsymbol{\tau}^S\left(\tilde{\mathbf{a}}^\alpha\otimes\bar{\mathbf{a}}^\alpha+\tilde{\mathbf{a}}^3\otimes\bar{\mathbf{a}}_3)\right)
\end{aligned}$$

where $\boldsymbol{\tau}^S$ is the Kirchhoff stress and $\tilde{\mathbf{a}}^\alpha$ and $\tilde{\mathbf{a}}^3$ are the contravariant counterparts of $\mathbf{a}_\alpha$ and $\mathbf{a}_3^E$ respectively.

And using index notation, we see that

$$\begin{aligned}
\frac{\partial\left(\mathbf{a}_\alpha\otimes\bar{\mathbf{a}}_\alpha+\mathbf{a}_3^E\otimes\bar{\mathbf{a}}_3\right)}{\partial\mathbf{a}_\beta} &= \frac{\partial a_{\alpha_i}\bar{a}_{\alpha_j}}{\partial a_{\beta_k}} \\
&= \delta_{\alpha\beta}\delta_{ik}\bar{a}_{\alpha_j} \\
&= \delta_{ik}\bar{a}_{\beta_j}
\end{aligned}$$

Similarly,

$$\frac{\partial\left(\mathbf{a}_\alpha\otimes\bar{\mathbf{a}}_\alpha+\mathbf{a}_3^E\otimes\bar{\mathbf{a}}_3\right)}{\partial\mathbf{a}_3^E} = \delta_{ik}\bar{a}_{3_j}$$

Hence, contracting the first two terms in the summation, each term in the summation becomes

$$\begin{aligned}
&\boldsymbol{\tau}^S\left(\tilde{\mathbf{a}}^\alpha\otimes\bar{\mathbf{a}}^\alpha+\tilde{\mathbf{a}}_3\otimes\bar{\mathbf{a}}_3\right)\bar{\mathbf{a}}_\beta : \frac{\partial\mathbf{a}_\beta}{\partial\mathbf{x}_{\mathbf{i}}} + \boldsymbol{\tau}^S\left(\tilde{\mathbf{a}}^\alpha\otimes\bar{\mathbf{a}}^\alpha+\tilde{\mathbf{a}}_3\otimes\bar{\mathbf{a}}_3\right)\bar{\mathbf{a}}_3 : \frac{\partial\mathbf{a}_3^E}{\partial\mathbf{x}_{\mathbf{i}}} \\
&= \boldsymbol{\tau}^S\tilde{\mathbf{a}}^\beta : \frac{\partial\mathbf{a}_\beta}{\partial\mathbf{x}_{\mathbf{i}}} + \boldsymbol{\tau}^S\tilde{\mathbf{a}}^3 : \frac{\partial\mathbf{a}_3^E}{\partial\mathbf{x}_{\mathbf{i}}}
\end{aligned}$$

Note that

$$\frac{\partial \mathbf{a}_\beta}{\partial \mathbf{x_i}} = \frac{\partial \mathbf{a}_\beta}{\partial \mathbf{x}_p}\frac{\partial \mathbf{x}_p}{\partial \mathbf{x_i}} = \frac{\partial \mathbf{a}_\beta}{\partial \mathbf{x}_p} w^n_{\mathbf{i}p},$$

and the expression for $\frac{\partial \mathbf{a}_\beta}{\partial \mathbf{x}_p}$ is given equation (A.1).

Ignoring further plastic flow, we have

$$\mathbf{a}_3^E(\mathbf{x}^*) = \left(\sum_{\mathbf{j}} \mathbf{x}^*_{\mathbf{j}} \otimes \nabla w^n_{\mathbf{j}p}\right) \mathbf{a}_3^{E,n},$$

and thus,

$$\frac{\partial \mathbf{a}_3^E}{\partial \mathbf{x_i}} = \nabla w^n_{\mathbf{i}p} \mathbf{a}_3^{E,n}$$

Therefore, we arrive at the final expression for the force of type (iii):

$$\mathbf{f}_{\mathbf{i}}^{(iii)}(\mathbf{x}^*) = \sum_{p \in \mathcal{I}^{(iii)}} \boldsymbol{\tau}^S_p \tilde{\mathbf{a}}^\beta_p : \frac{\partial \mathbf{a}_{p\beta}}{\partial \mathbf{x}_p} w^n_{\mathbf{i}p} + \boldsymbol{\tau}^S_p \tilde{\mathbf{a}}^3_p : \nabla w^n_{\mathbf{i}p} \mathbf{a}^{E,n}_{p3}$$

A.3 Laminate Stress

In this section we derive the expression for

$$\boldsymbol{\tau}^{KL} = \tau_{\alpha\beta} \mathbf{q}^{KL,E}_\alpha \otimes \mathbf{q}^{KL,E}_\beta, \quad \tau^{KL}_{\alpha\beta} = 2\mu\epsilon^L_{\alpha\beta} + \lambda\epsilon^L_{\gamma\gamma}\delta_{\alpha\beta}. \tag{A.2}$$

First notice that we may replace the right Hencky strain with left Hencky strain in the definition of energy because of the isotropic nature of the energy function. We now give the drivation of Equation (A.2) with index free notation assuming all variables are in 2D.

$$\begin{aligned}
\psi(\mathbf{F}) &= \psi(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) \\
\mathbf{P}(\mathbf{F}) &= \mathbf{P}(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) = \mathbf{U}\mathbf{P}(\boldsymbol{\Sigma})\mathbf{V}^T
\end{aligned}$$

because the energy is isotropic.

Hence,

$$\begin{aligned}
\mathbf{P}(\mathbf{F}) &= \mathbf{U}\mathbf{P}(\boldsymbol{\Sigma})\mathbf{V}^T \\
&= \mathbf{U}\frac{\partial \psi}{\partial \boldsymbol{\Sigma}}\mathbf{V}^T \\
&= \mathbf{U}\left(2\mu \log(\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1} + \lambda \mathrm{tr}(\log \boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}\right)\mathbf{V}^T.
\end{aligned}$$

Therefore,

$$\begin{aligned}\boldsymbol{\tau}^{KL} &= \left(\mathbf{U}\left(2\mu\log(\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1} + \lambda\mathrm{tr}(\log\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}\right)\mathbf{V}^T\right)\mathbf{F}^T \\ &= \mathbf{U}\left(2\mu\log(\boldsymbol{\Sigma}) + \lambda\mathrm{tr}(\log\boldsymbol{\Sigma})\right)\mathbf{U}^T \\ &= 2\mu\boldsymbol{\epsilon}^L + \lambda\mathrm{tr}(\boldsymbol{\epsilon}^L)\end{aligned}$$

A.4 QR and Elastic Potential

We can use QR orthogonalization of deformed material directions to define

$$\mathbf{q}_i r_{ij} = \mathbf{F}\bar{\mathbf{a}}_j,\ \mathbf{F} = r_{ij}\mathbf{q}_i \otimes \bar{\mathbf{a}}_j,\ r_{ij} = 0 \text{ for } i > j. \tag{A.3}$$

A.4.1 Change of basis tensor

Define the change of basis tensor

$$\mathbf{Q} = Q_{ij}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j \tag{A.4}$$

with $Q_{ij} = \mathbf{q}_j \cdot \bar{\mathbf{a}}_i$. With this convention we see that $\mathbf{Q}\bar{\mathbf{a}}_i = \mathbf{q}_i$ and $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Furthermore, defining

$$\mathbf{R} = r_{ij}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j$$

we have $\mathbf{F} = \mathbf{QR}$.

A.4.2 Differentials

The QR differential satisfies

$$\mathbf{q}_k \cdot \delta\mathbf{q}_i r_{ij} + \delta r_{kj} = \mathbf{q}_k \cdot (\delta\mathbf{F}\bar{\mathbf{a}}_j),\ \delta\mathbf{F} = \delta r_{ij}\mathbf{q}_i \otimes \bar{\mathbf{a}}_j + r_{ij}\delta\mathbf{q}_i \otimes \bar{\mathbf{a}}_j \tag{A.5}$$

where $\mathbf{q}_k \cdot \delta\mathbf{q}_i = -\mathbf{q}_i \cdot \delta\mathbf{q}_k$ from orthogonality of the $\mathbf{q}_i$. And

$$\delta\mathbf{F} = \delta\mathbf{QR} + \mathbf{Q}\delta\mathbf{R} \tag{A.6}$$

where $\delta\mathbf{Q}^T\mathbf{Q} = -\mathbf{Q}^T\delta\mathbf{Q}$ from $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Furthermore,

$$\delta\mathbf{Q} = \delta Q_{ij}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j,\ \delta Q_{ij} = \delta\mathbf{q}_j \cdot \bar{\mathbf{a}}_i,\ \delta\mathbf{q}_i = \delta\mathbf{Q}\bar{\mathbf{a}}_i \tag{A.7}$$

$$\delta\mathbf{R} = \delta r_{ij}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j \tag{A.8}$$

and the $\delta r_{ij} = 0$ for $i > j$.

A.5 Elastic potential and stresses

Define the hyperelastic potential as

$$\psi(\mathbf{F}) = \hat{\psi}([\mathbf{R}]) \tag{A.9}$$

where

$$[\mathbf{R}] = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \end{pmatrix}. \tag{A.10}$$

The differential satisfies

$$\delta\psi(\mathbf{F}) = \frac{\partial\psi}{\partial\mathbf{F}}(\mathbf{F}) : \delta\mathbf{F} = \mathbf{P} : \delta\mathbf{F} = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij} \tag{A.11}$$

where $\mathbf{P} = \frac{\partial\psi}{\partial\mathbf{F}}(\mathbf{F})$. Therefore

$$\delta r_{ij}\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j) + r_{ij}\delta\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j) = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij}. \tag{A.12}$$

Similarly,

$$\mathbf{P} : \delta\mathbf{F} = \mathbf{P} : (\delta\mathbf{Q}\mathbf{R}) + \mathbf{P} : (\mathbf{Q}\delta\mathbf{R}) = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij} \tag{A.13}$$

Choosing $\delta\mathbf{F} = \delta r_{ij}\mathbf{q}_i \otimes \bar{\mathbf{a}}_j$ (i.e. $\delta\mathbf{q}_i = \mathbf{0}$), we can conclude that

$$\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j)\, \delta r_{ij} = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij} \tag{A.14}$$

for arbitrary δr_{ij} with $i \leq j$. Therefore the $\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j) = \frac{\partial \hat{\psi}}{\partial r_{ij}}([\mathbf{R}])$ for $i \leq j$. Similarly,

$$\mathbf{P} : (\mathbf{Q}\delta\mathbf{R}) = (\mathbf{Q}^T\mathbf{P}) : \delta\mathbf{R} = \delta r_{ij}\bar{\mathbf{a}}_i \cdot \left(\mathbf{Q}^T\mathbf{P}\bar{\mathbf{a}}_j\right) = \delta r_{ij}\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j) = \frac{\partial \hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij}. \tag{A.15}$$

Choosing $\delta\mathbf{F} = r_{ij}\delta\mathbf{q}_i \otimes \bar{\mathbf{a}}_j$ (i.e. $\delta r_{ij} = 0$), we can conclude that

$$0 = r_{ij}\delta\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j)\,. \tag{A.16}$$

Similarly,

$$0 = \mathbf{P} : (\delta\mathbf{Q}\mathbf{R}) = \left(\mathbf{P}\mathbf{R}^T\right) : \delta\mathbf{Q} = \left(\mathbf{P}\mathbf{R}^T\right) : \left(\delta\mathbf{Q}\mathbf{Q}^T\mathbf{Q}\right) = \left(\mathbf{P}\mathbf{R}^T\mathbf{Q}^T\right) : \left(\delta\mathbf{Q}\mathbf{Q}^T\right) = \left(\mathbf{P}\mathbf{F}^T\right) : \left(\delta\mathbf{Q}\mathbf{Q}^T\right) \tag{A.17}$$

In other words, the Kirchhoff stress $\boldsymbol{\tau} = \mathbf{P}\mathbf{F}^T$ is symmetric since $\delta\mathbf{Q}\mathbf{Q}^T$ is arbitrary skew. Furthermore,

$$\mathbf{P} = P_{ij}\mathbf{q}_i \otimes \bar{\mathbf{a}}_j,\ \boldsymbol{\tau} = P_{ij}r_{kj}\mathbf{q}_i \otimes \mathbf{q}_k = \tau_{ik}\mathbf{q}_i \otimes \mathbf{q}_k \tag{A.18}$$

and we know $P_{ij} = \frac{\partial \hat{\psi}}{\partial r_{ij}}$ for $i \leq j$ from Equation A.14. Thus

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \begin{pmatrix} r_{11} & & \\ r_{12} & r_{22} & \\ r_{13} & r_{23} & r_{33} \end{pmatrix} \tag{A.19}$$

$$= \begin{pmatrix} P_{11}r_{11} + P_{12}r_{12} + P_{13}r_{13} & P_{12}r_{22} + P_{13}r_{32} & P_{13}r_{33} \\ P_{21}r_{11} + P_{22}r_{12} + P_{23}r_{13} & P_{22}r_{22} + P_{23}r_{32} & P_{23}r_{33} \\ P_{31}r_{11} + P_{32}r_{12} + P_{33}r_{13} & P_{32}r_{22} + P_{33}r_{32} & P_{33}r_{33} \end{pmatrix}, \tag{A.20}$$

and since $\boldsymbol{\tau} = \boldsymbol{\tau}^T$ and $P_{ij} = \frac{\partial \hat{\psi}}{\partial r_{ij}}$ for $i \leq j$,

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} \frac{\partial \hat{\psi}}{\partial r_{11}}r_{11} + \frac{\partial \hat{\psi}}{\partial r_{12}}r_{12} + \frac{\partial \hat{\psi}}{\partial r_{13}}r_{13} & \frac{\partial \hat{\psi}}{\partial r_{12}}r_{22} + \frac{\partial \hat{\psi}}{\partial r_{13}}r_{32} & \frac{\partial \hat{\psi}}{\partial r_{13}}r_{33} \\ \frac{\partial \hat{\psi}}{\partial r_{12}}r_{22} + \frac{\partial \hat{\psi}}{\partial r_{13}}r_{32} & \frac{\partial \hat{\psi}}{\partial r_{22}}r_{22} + \frac{\partial \hat{\psi}}{\partial r_{23}}r_{32} & \frac{\partial \hat{\psi}}{\partial r_{23}}r_{33} \\ \frac{\partial \hat{\psi}}{\partial r_{13}}r_{33} & \frac{\partial \hat{\psi}}{\partial r_{23}}r_{33} & \frac{\partial \hat{\psi}}{\partial r_{33}}r_{33} \end{pmatrix} \tag{A.21}$$

In particular, the matrix representation of $\boldsymbol{\tau}^S$ reads

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \gamma s_1 \\ 0 & 0 & \gamma s_2 \\ 0 & 0 & f'(s_3) \end{pmatrix} \begin{pmatrix} 0 & & \\ 0 & 0 & \\ s_1 & s_2 & s_3 \end{pmatrix} \tag{A.22}$$

$$= \begin{pmatrix} \gamma s_1^2 & \gamma s_1 s_2 & \gamma s_1 s_3 \\ \gamma s_1 s_2 & \gamma s_2^2 & \gamma s_2 s_3 \\ \gamma s_1 s_3 & \gamma s_2 s_3 & f'(s_3) \end{pmatrix} \tag{A.23}$$

A.6 Frictional Contact Yield Condition

Coulomb friction places a constraint on the stress as

$$|\mathbf{t}_S| \leq -c_F \sigma_n \tag{A.24}$$

where $\sigma_n = \mathbf{a}_3^{KL} \cdot \boldsymbol{\sigma} \mathbf{a}_3^{KL}$. Recall that $\mathbf{a}_3^{KL} = \mathbf{q}_3$ and thus $\sigma_n = \mathbf{q}_3 \cdot \boldsymbol{\sigma} \mathbf{q}_3$. On the other hand, $\mathbf{t}_S$ is the tangential component of the force density and has the form $\mathbf{t}_S = (c\mathbf{q}_1 + s\mathbf{q}_2) \cdot \boldsymbol{\sigma} \mathbf{q}_3$ for some c and s such that $c^2 + s^2 = 1$. Hence, we may rewrite the constraint on stress as

$$(c\mathbf{q}_1 + s\mathbf{q}_2) \cdot \boldsymbol{\sigma} \mathbf{q}_3 + c_F \mathbf{q}_3 \cdot \boldsymbol{\sigma} \mathbf{q}_3 \leq 0. \tag{A.25}$$

Using the fact that $\boldsymbol{\sigma} = \det(\mathbf{F})\boldsymbol{\tau}$, we rewrite the constraint as

$$(c\mathbf{q}_1 + s\mathbf{q}_2) \cdot \boldsymbol{\tau} \mathbf{q}_3 + c_F \mathbf{q}_3 \cdot \boldsymbol{\tau} \mathbf{q}_3 \leq 0. \tag{A.26}$$

Substituting in the expression for $\boldsymbol{\tau}$ from equation (A.23), we find that the maximum on the left-hand-side is

$$\pm \gamma s_3 \sqrt{s_1^2 + s_2^2} + c_F f' s_3$$

We apply the particular form of f in Chapter 5 where $f(x) = \frac{1}{3} k^c (1 - x)^3$ for $x \leq 1$ and 0 otherwise. When $s_3 > 1$, the maximum is $\gamma s_3 \sqrt{s_1^2 + s_2^2}$. In this case the return mapping set s_1 and s_2 to 0. If $0 < s_3 \leq 1$, the maximum is

$$\gamma s_3 \sqrt{s_1^2 + s_2^2} - c_F k^c (s_3 - 1)^2 s_3,$$

and thus we need

$$\sqrt{s_1^2 + s_2^2} \leq \frac{c_F k^c}{\gamma}(1 - s_3)^2.$$

In this case we uniformly scale back s_1 and s_2 to satisfy the constraint.

A.7 Denting Yield Condition and Return Mapping

We apply the von Mises yield condition to the Kirchhoff-Stress in Equation (A.2)

This condition states that the deviatoric component of the stress is less than a threshold value c_{vM}

$$f_{vM}(\boldsymbol{\tau}) = |\boldsymbol{\tau} - \frac{\mathrm{tr}(\boldsymbol{\tau})}{3}\mathbf{I}|_F \leq c_{vM}. \tag{A.27}$$

This condition defines a cylindrical region of feasible states in the principal stress space since

$$f_{vM}(\boldsymbol{\tau}) = \sqrt{\frac{2}{3}\left({\tau_1}^2 + {\tau_2}^2 + {\tau_3}^2 - (\tau_1\tau_2 + \tau_2\tau_3 + \tau_1\tau_3)\right)} \tag{A.28}$$

where $\boldsymbol{\tau} = \sum_i \tau_i \mathbf{u}_i \otimes \mathbf{u}_i$ with principal stresses τ_i. The plane stress nature of $\boldsymbol{\tau}^{KL} = \sum_\alpha \tau_\alpha^{KL} \mathbf{u}_\alpha \otimes \mathbf{u}_\alpha$ means that feasible stresses are those where the principal stresses are in the ellipsoidal intersection of the cylinder and the τ_α^{KL} plane.

The yield condition is satisfied via associative projection (or return mapping) of the stress to the feasible region. The elastic and plastic strains are then computed to be consistent with the projected stress. We use $\mathbf{F}^{KL,E^{\mathrm{tr}}}, \mathbf{F}^{KL,P^{\mathrm{tr}}}$ to denote the trial state of elastoplastic strains with associated trial stress $\boldsymbol{\tau}^{KL^{\mathrm{tr}}}$. We use $\mathbf{F}^{KL,E}, \mathbf{F}^{KL,P}, \boldsymbol{\tau}^{KL}$ to denote their projected counterparts.

$$\mathbf{F}^{KL,E^{\mathrm{tr}}}, \mathbf{F}^{KL,P^{\mathrm{tr}}}, \boldsymbol{\tau}^{KL^{\mathrm{tr}}} \rightarrow \mathbf{F}^{KL,E}, \mathbf{F}^{KL,P}, \boldsymbol{\tau}^{KL}. \tag{A.29}$$

The deformation gradient constraint must be equal to the product of trial and projected elastic and plastic deformation gradients, creating the constraint on the projection

$$\mathbf{F}^{KL} = \mathbf{F}^{KL,E^{\mathrm{tr}}}\mathbf{F}^{KL,P^{\mathrm{tr}}} = \mathbf{F}^{KL,E}\mathbf{F}^{KL,P}. \tag{A.30}$$

The projection is completed by first computing the trial state of stress $\boldsymbol{\tau}^{KL^{\mathrm{tr}}}$ from $\mathbf{F}^{KL,E^{\mathrm{tr}}}$ using Equation (A.2). This is done by computing the QR decomposition of the trial elastic

deformation gradient $\mathbf{F}^{KL,E^{\text{tr}}} = r_{\alpha\beta}^{KL,E^{\text{tr}}} \mathbf{q}_\alpha^{KL,E} \otimes \bar{\mathbf{a}}_\beta + \mathbf{q}_3^{KL,E} \otimes \bar{\mathbf{a}}_3$. Then we compute the SVD of matrix $[\mathbf{r}^{KL,E^{\text{tr}}}] \in \mathbb{R}^{2\times 2}$ and the trial strain $[\boldsymbol{\epsilon}^{L^{\text{tr}}}]$

$$[\mathbf{r}^{KL,E^{\text{tr}}}] = [\mathbf{U}^E] \begin{pmatrix} \sigma_1^{E^{\text{tr}}} & \\ & \sigma_2^{E^{\text{tr}}} \end{pmatrix} [\mathbf{V}^E]^T \tag{A.31}$$

$$[\boldsymbol{\epsilon}^{L^{\text{tr}}}] = [\mathbf{U}^E] \begin{pmatrix} \log(\sigma_1^{E^{\text{tr}}}) & \\ & \log(\sigma_2^{E^{\text{tr}}}) \end{pmatrix} [\mathbf{U}^E]^T \tag{A.32}$$

From Equation (A.2) we see that the two non-zero principal stresses ${\tau^{KL}}_\alpha^{\text{tr}}$ of $\boldsymbol{\tau}^{KL^{\text{tr}}}$ are equal to the eigenvalues of the matrix $[\boldsymbol{\tau}^{KL^{\text{tr}}}]$

$$[\boldsymbol{\tau}^{KL^{\text{tr}}}] = 2\mu[\boldsymbol{\epsilon}^{L^{\text{tr}}}] + \lambda \text{tr}([\boldsymbol{\epsilon}^{L^{\text{tr}}}])\mathbf{I} = [\mathbf{U}^E] \begin{pmatrix} \tau_1^{KL^{\text{tr}}} & \\ & \tau_2^{KL^{\text{tr}}} \end{pmatrix} [\mathbf{U}^E]^T. \tag{A.33}$$

We therefore project the eigenvalues (${\tau^{KL}}_\alpha^{\text{tr}} \rightarrow {\tau^{KL}}_\alpha$) into the ellipsoidal intersection the von Mises yield surface and the (τ_1, τ_2) plane in the direction that maximizes energy dissipation. We approximate this region by the diamond shaped region whose boundaries have slopes of ± 1 to simplify the return mapping. Note that the direction of the return that maximizes energy dissipation is a function of the Cauchy-Green strain derivative of the Kirchhoff stress and thus is non-trivial to find in general. Fortunately, the quadratic Hencky strain model has the favorable property that the return direction is perpendicular to the yield surface [32] which greatly simplifies the return mapping. After projection, we rebuild the matrix without changing the eigenvectors and rebuild $\boldsymbol{\tau}^{KL}$ from the matrix

$$[\boldsymbol{\tau}^{KL}] = [\mathbf{U}^E] \begin{pmatrix} {\tau^{KL}}_1 & \\ & {\tau^{KL}}_2 \end{pmatrix} [\mathbf{U}^E]^T, \ \boldsymbol{\tau}^{KL} = \tau_{\alpha\beta}^{KL} \mathbf{q}_\alpha^{KL,E} \otimes \mathbf{q}_\beta^{KL,E} \tag{A.34}$$

where $\tau_{\alpha\beta}^{KL}$ are the entries in the projected matrix $[\boldsymbol{\tau}^{KL}] \in \mathbb{R}^{2\times 2}$. The projected strain $[\boldsymbol{\epsilon}^L]$ is computed from the projected principal stresses from

$$[\boldsymbol{\epsilon}^L] = [\mathbf{U}^E] \begin{pmatrix} \log(\sigma_1^E) & \\ & \log(\sigma_2^E) \end{pmatrix} [\mathbf{U}^E]^T \tag{A.35}$$

$$\begin{pmatrix} \log(\sigma_1^E) \\ \log(\sigma_2^E) \end{pmatrix} = \begin{pmatrix} 2\mu + \lambda & \lambda \\ \lambda & 2\mu + \lambda \end{pmatrix}^{-1} \begin{pmatrix} {\tau^{KL}}_1 \\ {\tau^{KL}}_2 \end{pmatrix} \tag{A.36}$$

and the projected elastic deformation gradient is $\mathbf{F}^{KL,E} = F^{KL,E}_{\alpha\beta}\mathbf{q}^{KL,E}_{\alpha} \otimes \bar{\mathbf{a}}_\beta + \mathbf{q}^{KL,E}_3 \otimes \bar{\mathbf{a}}_3$ where

$$[\hat{\mathbf{F}}^{KL,E}] = [\mathbf{U}^E] \begin{pmatrix} \sigma^E_1 & \\ & \sigma^E_2 \end{pmatrix} [\mathbf{V}^E]^T. \tag{A.37}$$

The projected plastic deformation gradient is computed from $\mathbf{F}^{KL,P} = \mathbf{F}^{KL,E^{-1}}\mathbf{F}^{KL}$ in order to maintain the constraint in Equation (A.30).

Bibliography

[1] T. Belytschko et al. *Nonlinear finite elements for continua and structures*. John Wiley and sons, 2013.

[2] Miklós Bergou et al. "Discrete elastic rods". In: *ACM transactions on graphics (TOG)* 27.3 (2008), p. 63.

[3] Miklós Bergou et al. "Discrete viscous threads". In: *ACM Transactions on Graphics (TOG)*. Vol. 29. 4. ACM. 2010, p. 116.

[4] F. Bertails et al. "Super-helices for Predicting the Dynamics of Natural Hair". In: *ACM Trans Graph* 25.3 (2006), pp. 1180–1187.

[5] J. Bonet and R. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 2008.

[6] J. Brackbill and H. Ruppel. "FLIP: A method for adaptively zoned, Particle-In-Cell calculations of fluid flows in two dimensions". In: *J Comp Phys* 65 (1986), pp. 314–343.

[7] R. Bridson, R. Fedkiw, and J. Anderson. "Robust Treatment of Collisions, Contact and Friction for Cloth Animation". In: *ACM Trans Graph* 21.3 (2002), pp. 594–603.

[8] Edwin Catmull and James Clark. "Recursively generated B-spline surfaces on arbitrary topological meshes". In: *Computer-aided design* 10.6 (1978), pp. 350–355.

[9] Alexandre Joel Chorin. "The numerical solution of the Navier-Stokes equations for an incompressible fluid". In: *Bulletin of the American Mathematical Society* 73.6 (1967), pp. 928–931.

[10] F. Cirak, M. Ortiz, and P. Schröder. "Subdivision surfaces: a new paradigm for thin-shell finite-element analysis". In: *Int J Num Meth Eng* 47.12 (2000), pp. 2039–2072.

[11] D. Clyde, J. Teran, and R. Tamstorf. "Modeling and data-driven parameter estimation for woven fabrics". In: *Proc ACM SIGGRAPH / Eurograp Symp Comp Anim*. SCA '17. Los Angeles, California: ACM, 2017, 17:1–17:11. ISBN: 978-1-4503-5091-4.

[12] David Clyde. "Numerical Subdivision Surfaces for Simulation and Data Driven Modeling of Woven Cloth". PhD thesis. UCLA, 2017.

[13] G. Daviet and F. Bertails-Descoubes. "A Semi-implicit Material Point Method for the Continuum Simulation of Granular Materials". In: *ACM Trans Graph* 35.4 (2016), 102:1–102:13.

[14] Y. Fei et al. "A multi-scale model for simulating liquid-fabric interactions". In: *ACM Trans Graph* 37.4 (2018), 51:1–51:16. DOI: 10.1145/3197517.3201392.

[15] C. Fu et al. "A Polynomial Particle-in-cell Method". In: *ACM Trans Graph* 36.6 (Nov. 2017), 222:1–222:12.

[16] M. Gao et al. "An adaptive generalized interpolation material point method for simulating elastoplastic materials". In: *ACM Trans Graph* 36.6 (2017), 223:1–223:12. DOI: 10.1145/3130800.3130879.

[17] O. Gonzalez and A. Stuart. *A first course in continuum mechanics*. Cambridge University Press, 2008.

[18] J. E. Guilkey and J. A. Weiss. "Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method". In: *Int J Numer Meth Eng* 57.9 (2003), pp. 1323–1338.

[19] Q. Guo et al. "A material point method for thin shells with frictional contact". In: *ACM Trans Graph* 37.4 (2018), p. 147. DOI: 10.1145/3197517.3201346.

[20] C. Hammerquist and J. Nairn. "A new method for material point method particle updates that reduces noise and enhances stability". In: *Comp Meth App Mech Eng* 318 (2017), pp. 724 –738.

[21] Xuchen Han et al. "A Hybrid Material Point Method for Frictional Contact with Diverse Materials". In: *Proc. ACM Comput. Graph. Interact. Tech.* 2.2 (July 2019). DOI: 10.1145/3340258. URL: https://doi.org/10.1145/3340258.

[22] F. Harlow. "The particle-in-cell method for numerical solution of problems in fluid dynamics". In: *Meth Comp Phys* 3 (1964), pp. 319–343.

[23] D. Harmon et al. “Robust Treatment of Simultaneous Collisions”. In: *ACM Trans Graph* 27.3 (2008), 23:1–23:4.

[24] G. Irving, J. Teran, and R. Fedkiw. “Invertible Finite Elements for Robust Simulation of Large Deformation”. In: *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim.* 2004, pp. 131–140.

[25] C. Jiang, T. Gast, and J. Teran. “Anisotropic elastoplasticity for cloth, knit and hair frictional contact”. In: *ACM Trans Graph* 36.4 (2017), p. 152.

[26] C. Jiang, C. Schroeder, and J. Teran. “An angular momentum conserving affine-particle-in-cell method”. In: *J Comp Phys* 338 (2017), pp. 137 –164.

[27] C. Jiang et al. “The Affine Particle-In-Cell Method”. In: *ACM Trans Graph* 34.4 (2015), 51:1–51:10.

[28] Chenfanfu Jiang et al. “The Material Point Method for Simulating Continuum Materials”. In: *ACM SIGGRAPH 2016 Course*. 2016, 24:1–24:52.

[29] G. Klár et al. “Drucker-prager Elastoplasticity for Sand Animation”. In: *ACM Trans Graph* 35.4 (2016), 103:1–103:12.

[30] G. Klar et al. “Drucker-Prager Elastoplasticity for Sand Animation”. In: *ACM Trans Graph* 35.4 (July 2016).

[31] J. Lu and C. Zheng. “Dynamic cloth simulation by isogeometric analysis”. In: *Comp Meth App Mech Eng* 268.Supplement C (2014), pp. 475 –493.

[32] C. Mast. “Modeling landslide-induced flow interactions with structures using the Material Point Method”. PhD thesis. 2013.

[33] A. McAdams et al. “Detail Preserving Continuum Simulation of Straight Hair”. In: *ACM Trans Graph* 28.3 (2009), 62:1–62:6.

[34] M. Otaduy et al. “Implicit Contact Handling for Deformable Objects”. In: *Comp Graph Forum* 28.2 (2009).

[35] D. Ram et al. “A material point method for viscoelastic fluids, foams and sponges”. In: *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim.* 2015, pp. 157–163.

[36] 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. Los Angeles, California: ACM, 2012, 20:1–20:50. DOI: 10.1145/2343483.2343501.

[37] E. Sifakis, S. Marino, and J. Teran. "Globally Coupled Collision Handling Using Volume Preserving Impulses". In: *Proc 2008 ACM SIGGRAPH/Eurographics Symp Comp Anim*. 2008, pp. 147–153.

[38] Jos Stam. "Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values". In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 1998, pp. 395–404.

[39] A. Stomakhin et al. "A Material Point Method for snow simulation". In: *ACM Trans Graph* 32.4 (2013), 102:1–102:10.

[40] A. Stomakhin et al. "Augmented MPM for phase-change and varied materials". In: *ACM Trans Graph* 33.4 (2014), 138:1–138:11.

[41] A. Stomakhin et al. "Energetically consistent invertible elasticity". In: *Proc Symp Comp Anim*. 2012, pp. 25–32.

[42] D. Sulsky, Z. Chen, and H. Schreyer. "A particle method for history-dependent materials". In: *Comp Meth App Mech Eng* 118.1 (1994), pp. 179–196.

[43] A. P. Tampubolon et al. "Multi-species simulation of porous sand and water mixtures". In: *ACM Trans Graph* 36.4 (2017).

[44] M. Tang et al. "CAMA: Contact-Aware Matrix Assembly with Unified Collision Handling for GPU-based Cloth Simulation". In: *Comp Graph Forum* 35.2 (2016), pp. 511–521.

[45] Y. Yue et al. "Continuum foam: a material point method for shear-dependent flows". In: *ACM Trans Graph* 34.5 (2015), 160:1–160:20.

[46] Y. Yue et al. "Hybrid grains: adaptive coupling of discrete and continuum simulations of granular media". In: *ACM Trans Graph* 37.6 (2018), 283:1–283:19.

[47] Fan Zhang et al. “Incompressible material point method for free surface flow”. In: *Journal of Computational Physics* 330 (2017), pp. 92–110.

[48] F. Zhu et al. “Dynamically enriched MPM for invertible elasticity”. In: *Comp Graph Forum*. Vol. 36. 6. Wiley Online Library. 2017, pp. 381–392.