UNIVERSITY OF CALIFORNIA
Los Angeles

The Material Point Method for Simulating Elastoplastic Materials

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

by

Chuyuan Fu

2018

ABSTRACT OF THE DISSERTATION

The Material Point Method
for Simulating Elastoplastic Materials

by

Chuyuan Fu
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2018
Professor Joseph M. Teran, Chair

Simulation in computer graphics must accommodate a wide range of material behaviors under large deformation, topological changes, contact or collision. This dissertation focuses on hybrid Lagrangian/Eulerian methods, especially the Material Point Method (MPM) in computer animation, and we list the major contributions here:

First, we simulate sand dynamics using an elastoplastic, continuum assumption. We demonstrate that the Drucker-Prager plastic flow model combined with a Hencky-strain-based hyperelasticity accurately recreates a wide range of visual sand phenomena with moderate computational expense. The Drucker-Prager model naturally represents the frictional relation between shear and normal stresses through a yield stress criterion. We develop a stress projection algorithm used for enforcing this condition with a non-associative flow rule.

We further extend the idea of simulating sand dynamics using an elastoplastic continuum assumption to codimensional objects. Our second contribution is to introduce a novel method for simulation of thin shells with frictional contact using a combination of the 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 decouple resolving contact and shearing from the bending resistance components of stress. Our approach is capable of simulating challenging

shell contact scenarios with hundreds of thousands to millions of degrees of freedom with a moderate cost of only a few minutes per frame.

Our third contribution is to introduce a novel transfer scheme that is used in hybrid Lagrangian/Eulerian simulations. Recently the Affine Particle-In-Cell (APIC) Method was introduced to improve the accuracy of the transfers in Particle-In-Cell (PIC) without suffering from the noise present in the historic alternative, Fluid-Implicit-Particle (FLIP). We generalize APIC by augmenting each particle with a more general local function. Our transfers are designed to select particle-wise polynomial approximations to the grid velocity that are optimal in the local mass-weighted L^2 norm. With only marginal additional cost, our generalization improves kinetic energy conservation during transfers which leads to better vorticity resolution in fluid simulations and less numerical damping in elastoplasticity simulations.

The dissertation of Chuyuan Fu is approved.

Luminita Aura Vese

Demetri Terzopoulos

Christopher R. Anderson

Joseph M. Teran, Committee Chair

University of California, Los Angeles

2018

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 continuous support for my PhD study and research. He shared insights and wisdom through near-daily meetings and emails and attention far beyond what was required. His passionate and diligent pursuit of academic excellence taught me how to be a devoted scientist.

I thank Professor Demetri Terzopoulos, Professor Chris Andersen, and Professor Luminita Vese for serving on my thesis committee and providing suggestions on improving and extending my work.

My appreciation goes to my friends and colleagues at UCLA. I thank Chenfanfu Jiang, Andre Pradhana Tampubolon, Craig Schroeder, Ted Gast, Greg Klar, Qi Guo, Xuchen Han, David Clyde, Mengyuan Ding, Stephanie Wang, and Masaki Nakada for many insightful discussions and being the most pleasant companions.

I would like to thank our collaborators Ken Museth at DreamWorks Animation and Rasmus Tamstorf at Walt Disney Animation Studios, for sharing their knowledge and their guidance.

I thank the two cats Kaku and Ginsan for being the most beautiful creatures and possessing the sweetest souls. Their continuous exploration of self-world relation is always inspirational.

Most of all, I thank my family for their understanding and unconditional love from across the Pacific Ocean.

VITA

2010–2013 BSc (Mathematics), University of Hong Kong, Hong Kong, China.

2013–2018 Research Assistant, Mathematics Department, UCLA, USA.

2014–2018 Teaching Assistant, Mathematics Department, UCLA, USA.

2017 Research Intern, Oculus Research, Redmond, USA.

PUBLICATIONS

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 Asia 2018).

C. Fu, Q. Guo, T. Gast, C. Jiang, J. Teran, A Polynomial Particle-In-Cell Method, ACM Transactions on Graphics (SIGGRAPH Asia 2017).

A. Pradhana, T. Gast, G. Klar, C. Fu, J. Teran, C. Jiang, K. Museth, Multi-species Simulation of Porous Sand and Water Mixtures, ACM Transactions on Graphics (SIGGRAPH 2017).

G. Klar, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, J. Teran, Drucker-Prager Elastoplasticity for Sand Animation, ACM Transactions on Graphics (SIGGRAPH 2016).

T. Gast, C. Fu, C. Jiang, J. Teran, Implicit-shifted Symmetric QR Singular Value Decomposition of 3x3 Matrices, UCLA Mathematics Department Technical Report (CAM16-19).

CHAPTER 1

Introduction

1.1 Contributions

We make the following key contributions:

- We introduce a novel information transfer scheme, a Polynomial Particle-In-Cell method. It generalizes APIC from locally affine to locally polynomial representations and improves kinetic energy conservation in particle/grid transfers (Figure 6.5 and Figure 6.6).
- We develop an elastoplastic formulation for frictional contact and resistance to bending and denting of thin shells. Furthermore, we develop a strain splitting technique to separate thin shell motion into Kirchhoff-Love and continuum shell components (Figure 5.8 and Figure 5.13).
- We derive a plane strain/stress formulation for Kirchhoff-Love thin shells that simplifies the return mapping algorithm for denting resistance (Figure 5.14).

1.2 Dissertation Overview

The dissertation is structured as follows:

In Chapter 2 we briefly review the mathematical background of continuum mechanics. We review the notation and the concepts of deformation gradient and elastic stress, which are key to our simulation methods.

Chapter 3 covers the method outline of MPM in the case of elastoplasticity. The simulation methods used in this dissertation follow this outline. We assume that readers have

basic knowledge about continuum mechanics and the MPM.

Chapter 4 presents modeling sand by considering granular media as an elastoplastic continuum. Plastic deformation is modeled by defining the Drucker-Prager yield criterion. In the discrete setting, the yield criterion is satisfied by defining return mapping through the flow rule. This section contains the essential treatise for elastoplastic simulation with the MPM which we will extend to dealing with codimensional objects in Chapter 5.

In Chapter 5, we present a novel method for simulation of thin shells with frictional contact using a combination of the MPM and subdivision finite elements. We decompose the shell kinematics into a Kirchhoff-Love motion and compression/extension of the material. 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.

In Chapter 6, we introduce a novel transfer scheme for hybrid Euler/Lagrangian approaches. We present a generalization of existing methods by augmenting each particle with a polynomial function to represent its local velocity field. By viewing the grid-to-particle transfer as a linear and angular momentum conserving projection of the particle-wise local grid velocities onto a reduced basis, we greatly improve the energy and vorticity conservation over the original methods with only marginal additional cost.

CHAPTER 2

Continuum Mechanics

2.1 Governing equations

We represent objects, either sand or thin shells, as volumetric elastoplastic continua. While shells are thin surfaces in 3D, it is useful to conceive of their dynamics as if they have appreciable thickness in a continuum. The state can be described at each location by its density $\rho(\mathbf{x}, t)$ and velocity $\mathbf{v}(\mathbf{x}, t)$. The governing equations come from conservation of mass and momentum (see [16, 5] for derivations)

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{v} = 0,\ \rho\frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho\mathbf{g}. \tag{2.1}$$

Here $\boldsymbol{\sigma}$ is the stress, $\mathbf{g}$ is gravity and $\frac{D}{Dt}$ is the material derivative.

2.2 Deformation gradient

The material deformation is characterized in terms of the flow map, $\boldsymbol{\phi}$ which maps points in the original configuration of the material $\mathbf{X}$ to points in the time t configuration $\mathbf{x}$ as $\boldsymbol{\phi}(\mathbf{X}, t) = \mathbf{x}$. The Jacobian of this mapping $\mathbf{F} = \frac{\partial\boldsymbol{\phi}}{\partial\mathbf{X}}(\mathbf{X}, t)$ is often referred to as deformation gradient, and it represents the local deformation of the material. That is, the deformation gradient yields the best local linear approximation to the mapping: $\boldsymbol{\phi}(\tilde{\mathbf{X}}, t) \approx \mathbf{F}(\tilde{\mathbf{X}} - \mathbf{X}) + \mathbf{x}$ for $\tilde{\mathbf{X}}$ near $\mathbf{X}$. For example, if the material is undeformed local to $\mathbf{X}$ then $\mathbf{F}$ will be a rotation. If $\det(\mathbf{F}) < 1$, the material loses volume locally, and if $\det(\mathbf{F}) > 1$, it gains volume locally. The elastic and frictional contact responses of our model are characterized in terms

of deformation gradient. The material derivative of deformation gradient is:

$$\frac{D\mathbf{F}}{Dt} = (\nabla \mathbf{v})\mathbf{F}.$$

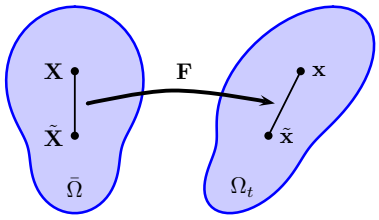


Figure 2.1: Deformation gradient.

We represent plasticity by factoring deformation gradient into elastic and plastic parts as $\mathbf{F} = \mathbf{F}^E\mathbf{F}^P$. $\mathbf{F}^P$ represents the portion of the material's history that has been forgotten. For example, If a metal can is twisted, then it *forgets* that it was straight and flat, and behaves as it was always twisted. The twisting and denting involved in this operation is stored in $\mathbf{F}^P$. If the can is twisted slightly, then it will feel strain (deformation). This is elastic deformation, which is stored in $\mathbf{F}^E$. The can remembers this deformation. In response, the material exerts stress to try to restore itself to its orignal shape. In this way, we see that only $\mathbf{F}^E$ should be used to compute stress.

2.3 Elastic stress

For potential energy density $\psi(\mathbf{F}^E)$, the Cauchy stress is

$$\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F})}\frac{\partial \psi(\mathbf{F}^E)}{\partial \mathbf{F}^E}\mathbf{F}^{E^T}. \tag{2.2}$$

It is often mathematically convenient when expressing the plasticity constraints as well as useful for our MPM implementation to work with the Kirchhoff stress $\boldsymbol{\tau}$. It is related to the

more commonly used Cauchy stress $\boldsymbol{\sigma}$ as $\boldsymbol{\tau} = \det(\mathbf{F})\boldsymbol{\sigma}$, i.e.,

$$\boldsymbol{\tau} = \frac{\partial \psi(\mathbf{F}^E)}{\partial \mathbf{F}^E} \mathbf{F}^{E^T}. \tag{2.3}$$

CHAPTER 3

The Material Point Method

3.1 Notation

It is helpful to establish the conventions for notation (see Table 3.1 for a list of notation used). Scalars are represented by non-bold Latin or Greek characters (e.g., m_p, $w_{\mathbf{i}p}^n$). Vectors are represented by bold lowercase Latin characters (e.g., $\mathbf{v}_p^n$, $\mathbf{x}_{\mathbf{i}}^{n+1}$). Matrices are represented by bold uppercase Latin characters or bold Greek characters (e.g., $\mathbf{I}$, $\mathbf{F}_p^{P,n+1}$, $\boldsymbol{\sigma}$). Derivatives alter this in the usual way, so that $(\nabla \mathbf{v})_p$ is a matrix.

Under discretized setting, many quantities are indexed with subscripts, which indicate where quantities are stored. Quantities that are stored at grid nodes are indexed with $\mathbf{i}$ and particle quantities have the index p. The quantity $\mathbf{F}_p^{n+1}$ represents the quantity corresponding to one index, and $\langle \mathbf{F}_p^{n+1} \rangle$ represents a vector of all such quantities.

Superscript n is used to indicate a quantity near the beginning of the time step, before forces are applied, (e.g., m_i^n, $\mathbf{v}_p^n$). Under the updated Lagrangian view (see Section (§3.4)), we use superscript $n+1$ to denote quantities after momentum update (e.g., $\mathbf{x}_{\mathbf{i}}^{n+1}$, $\mathbf{C}_p^{n+1}$).

Superscripts E and P are used to denote the elastic or plastic part of a deformation gradient (e.g., $\mathbf{F}_p^{E,n}$, $\mathbf{F}_p^{P,n}$). Superscript tr is used to denote the trial state (e.g., $\mathbf{F}_p^{E,\mathrm{tr}}$).

3.2 Method outline

We use MPM to discretize our elastoplastic model. 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,

Variable	Where	Type	Meaning
$\mathbf{I}$	-	matrix	identity matrix
Δx	-	scalar	grid resolution
$\frac{D}{Dt}$	-	-	material derivative
$\mathbf{g}$	-	vector	gravity
$\boldsymbol{\sigma}$	-	matrix	Cauchy stress
ρ	-	scalar	density
$\mathbf{v}$	-	vector	velocity
$\mathbf{F}$	-	matrix	deformation gradient
m_p	particles†	scalar	particle mass
V_p^0	particles†	scalar	initial particle volume
$\mathbf{F}_p^n$, $\hat{\mathbf{F}}_p$	particles	matrix	deformation gradient
$\mathbf{v}_p^n$, $\hat{\mathbf{v}}_p$	particles†	vector	particle velocity
$\mathbf{x}_p^n$, $\hat{\mathbf{x}}_p$	particles†	vector	particle position
$(\nabla \mathbf{v})_p$	particles	matrix	grid-based velocity gradient
N	-	vector $\rightarrow$ scalar	tensor product interpolation spline
$w_{\mathbf{i}p}$	mixed	scalar	interpolation weight
$\nabla w_{\mathbf{i}p}$	mixed	vector	interpolation weight gradient

Table 3.1: Table of notation used in this chapter. †These quantities are state on particles.

elastic deformation gradient $\mathbf{F}_p^{E,n}$, affine velocity $\mathbf{C}_p^n$ for all materials in the simulation. Note that the mass does not change with time in accordance with conservation of mass.

In MPM, the Eulerian grid can be viewed as an auxiliary structure for updating the Lagrangian state. In order to update the Lagrangian state to obtain $\mathbf{x}_p^{n+1}$, $\mathbf{v}_p^{n+1}$ etc., 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 $\mathbf{x}_{\mathbf{i}}$ 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 motion of the grid is then interpolated back to the particles to update the Lagrangian state without ever actually moving grid nodes.

The yield condition we define for plasticity 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 $\mathbf{F}^{E,\text{tr}}$, $\mathbf{F}^{P,\text{tr}}$ to denote this trial state of elastoplastic strains with associated trial stress. This stress may or may not satisfy the yield condition. The trial stress is then projected to the feasible region to satisfy the yield condition. The final elastic and plastic strains $\mathbf{F}^E, \mathbf{F}^P$ are then computed from the projected stress.

We summarize essential steps in the algorithm below.

1. **Transfer to grid:** Transfer mass and momentum from particles to grid. (§3.3)
2. **Update grid momentum:** Update grid momentum from potential-energy-based and body forces. (§3.4)
3. **Transfer to particles:** Transfer velocities from grid to particles. (§3.5)
4. **Update positions and trial elastic state:** Update particle position, deformation gradient and trial elastic state assuming no plasticity over the time step. (§3.6)
5. **Update plasticity:** Project trial elastic and plastic deformation gradients for plasticity return mapping. (§3.7)

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 $\mathbf{x_i}$ using APIC [24].

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

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

Here $w_{\mathbf{i}p}^n = N(\mathbf{x}_p^n - \mathbf{x_i})$ 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.

3.4 Grid Momentum Update

The grid momentum update uses the updated Lagrangian view of the governing physics. Although we never deform the grid, we conceive of the grid momentum update in terms of the motion it would cause on the grid. In particular this allows us to define the elastic grid

forces through the differentiation of a potential. This use of incremental grid node motion, is called the updated Lagrangian view, as we essentially view the motion of the grid as Lagrangian, albeit from the grid configuration at time t^n, instead of the rest configuration.

The grid momentum is updated from the force defined as the gradient of the potential energy with respect to grid node motion. 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 state after the time step, the grid discretization has the form

$$\mathbf{x_i^{n+1}} = \mathbf{x_i} + \frac{\Delta t}{m_\mathbf{i}^n}\mathbf{p_i^{n+1}} \tag{3.3}$$

$$\mathbf{p_i^{n+1}} = m_\mathbf{i}^n\mathbf{v_i^n} - \Delta t\frac{\partial \Psi}{\partial \mathbf{x_i}}(\langle \hat{\mathbf{x}}_\mathbf{i} \rangle) + \Delta t m_\mathbf{i}^n \mathbf{g} \tag{3.4}$$

where $\Psi(\mathbf{x})$ is the potential energy which depends on the positional state where we use $\langle \hat{\mathbf{x}}_\mathbf{i} \rangle = (\hat{\mathbf{x}}_{\mathbf{i}^1}, \hat{\mathbf{x}}_{\mathbf{i}^2}, \ldots)^T$ to denote the vector of all grid node positions. In the case of symplectic Euler integration, $\langle \hat{\mathbf{x}}_\mathbf{i} \rangle = \mathbf{x_i}$ and in the case of backward Euler, $\langle \hat{\mathbf{x}}_\mathbf{i} \rangle = \langle \mathbf{x_i^{n+1}} \rangle$. We note that the grid nodes are not actually moved from $\langle \mathbf{x_i} \rangle$ to $\langle \mathbf{x_i^{n+1}} \rangle$. Instead, the motion of the grid is interpolated to the particles (see (§3.5)).

Given elastic energy density ψ, the total potential is computed as $\Psi = \int_{\bar{\Omega}} \psi \, dV$ over material domain $\bar{\Omega}$. When discretely approximated on particles, we have

$$\Psi = \sum_p V_p^0 \psi(\mathbf{F}_p^{E,\mathrm{tr}}), \tag{3.5}$$

where V_p^0 is the undeformed volume of particle p, and the elastic deformation gradient is updated as

$$\mathbf{F}_p^{E,\mathrm{tr}}(\hat{\mathbf{x}}) = (\nabla \hat{\mathbf{x}})_p \mathbf{F}_p^{E,n}. \tag{3.6}$$

Here $(\nabla \hat{\mathbf{x}})_p = \sum_i \hat{\mathbf{x}}_i \otimes \nabla w_{ip}^n$ and $\mathbf{F}_p^0 = \mathbf{I}$ assuming no initial deformation. Putting it together using chain rule, the grid node forces $\mathbf{f_i}$ are computed as

$$\mathbf{f_i} = -\frac{\partial \Psi}{\partial \hat{\mathbf{x}}_\mathbf{i}} = -\sum_p V_p^0 \frac{\partial \psi}{\partial \mathbf{F}^E}(\mathbf{F}_p^{E,\mathrm{tr}}) : \frac{\partial \mathbf{F}_p^{E,\mathrm{tr}}}{\partial \mathbf{x}_i}. \tag{3.7}$$

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

$$\mathbf{v}_p^{n+1} = \sum_\mathbf{i} w_{\mathbf{i}p}^n \frac{\mathbf{p}_\mathbf{i}^{n+1}}{m_\mathbf{i}^n} \tag{3.8}$$

$$\tilde{\mathbf{C}}_p^{n+1} = \frac{12}{\Delta x^2 (d+1)} \sum_\mathbf{i} w_{\mathbf{i}p}^n \frac{\mathbf{p}_\mathbf{i}^{n+1}}{m_\mathbf{i}^n} \otimes (\mathbf{x}_\mathbf{i}^n - \mathbf{x}_p^n) \tag{3.9}$$

$$\mathbf{C}_p^{n+1} = (1-\nu)\,\tilde{\mathbf{C}}_p^{n+1} + \frac{\nu}{2}\left(\tilde{\mathbf{C}}_p^{n+1} - (\tilde{\mathbf{C}}_p^{n+1})^T\right) \tag{3.10}$$

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.[23] where $\nu = 0$ is completely undamped, corresponding to the original APIC transfer, while $\frac{1}{2}\left(\tilde{\mathbf{C}}_p^{n+1} - (\tilde{\mathbf{C}}_p^{n+1})^T\right)$is the skew symmetric part of $\tilde{\mathbf{C}}_p^{n+1}$ and corresponds to the RPIC transfer.

3.6 Update Positions and Trial Elastic State

Particle positions are moved with the interpolated grid node velocities.

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} = \sum_\mathbf{i} \mathbf{x}_\mathbf{i}^{n+1} w_{\mathbf{i}p}^n. \tag{3.11}$$

We first assume there was no additional plastic flow over the time step and consider a trial state of elastic deformation. The trial elastic deformation $\mathbf{F}_p^{E,\mathrm{tr}}$ is computed as in Equations (3.6) with $\langle \hat{\mathbf{x}}_\mathbf{i} \rangle = \langle \mathbf{x}_\mathbf{i}^{n+1} \rangle$.

3.7 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. $\mathbf{F}_p^{E,\text{tr}}$ is projected to $\mathbf{F}_p^{E,n+1}$ in accordance with whichever return mapping is being used. We will discuss the details of each plasticity for sand in Chapter 4, for shell in Chapter 5 . 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

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^{E,\text{tr}}\mathbf{F}_p^{P,\text{tr}} = \mathbf{F}_p^{E,n+1}\mathbf{F}_p^{P,n+1}. \tag{3.12}$$

CHAPTER 4

Drucker-Prager Elastoplasticity for Sand

We simulate sand dynamics using an elastoplastic, continuum assumption. We demonstrate that the Drucker-Prager plastic flow model combined with a Hencky-strain-based hyperelasticity accurately recreates a wide range of visual sand phenomena with moderate computational expense. We use the Material Point Method (MPM) to discretize the governing equations for its natural treatment of contact, topological change and history dependent constitutive relations. The Drucker-Prager model naturally represents the frictional relation between shear and normal stresses through a yield stress criterion.

This chapter explains plasticity in great detail. The idea of handling friction and contact with a continuum view and expressing the amount of friction allowed through plasticity constraints is going to be useful for the next Chapter when we talk about shells.

In this chapter, we will present the Drucker-Prager yield surface derivation as well as the return mapping it induces. The Drucker-Prager yield surface motivates from the Coulomb condition, defining the admissible states of stress. Plastic flow rule is therefore defined in order for the stress to satisfy this plasticity constraint. After we derive the proper plastic flow rule which satisfies the yield condition, second law of thermodynamics, and volume preservation, we use this flow rule to design its discretized version: the return mapping. We then show that under Hencky-strain, the return mapping will amount to a simple projection which allows for straightforward implementation.

4.1 Drucker-Prager yield surface derivation

In the continuum conception of sand, mechanical interactions are expressed through elasticity, modified with plasticity to model the effects of frictional contact. We use the Drucker-Prager plasticity model, which is built to enforce that shear stresses do not exceed a coefficient times normal stresses in magnitude. In this Section, we detail the connection between Coulomb friction, and the Drucker-Prager stress condition.

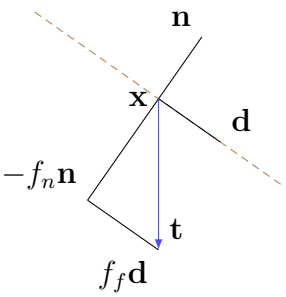
The stress condition defines a notion of admissibility for states of stress. In stress space, this is a region whose boundary is often referred to as the yield surface. This places a constraint on the constitutive model defining the mechanical response of the body. The multiplicative decomposition of the deformation gradient into elastic and plastic parts is a means for designing a constitutive model that meets these constraints. For states of stress in the interior of the feasible region, there is no plastic flow since the elastic constitutive model suffices. However, as a state on the boundary of the region (yield surface) is approached, plastic flow will be defined as means of modifying the constitutive model to satisfy the constraints. In Section 4.2 we derive the plastic flow as a means of satisfying the Drucker-Prager stress constraint.

4.1.1 Coulomb friction

Consider a Coulomb friction interaction between two grains in contact. If c_F is the positive coefficient of friction, then the frictional force f_f can only be as large as the coefficient of friction times the normal force f_n: $f_f \leq c_F f_n$. The Drucker-Prager model generalizes this to a continuum. At any point in the continuum body, the Cauchy stress $\boldsymbol{\sigma}$ expresses the local mechanical interactions in the material.

If we consider this interaction to be from friction, we can use the Coulomb model to relate the frictional force (per area) $f_f = \mathbf{d}^T\mathbf{t}$ to the normal force (per area) $f_n = -\mathbf{n}^T\mathbf{t}$ as $\mathbf{d}^T\mathbf{t} \leq -c_F\mathbf{n}^T\mathbf{t}$. Here, $\mathbf{d}$ is the normalized projection of the traction $\mathbf{t}$ into the plane orthogonal to $\mathbf{n}$. In terms of $\boldsymbol{\sigma}$, this is expressed as $\mathbf{d}^T\boldsymbol{\sigma}\mathbf{n} \leq -c_F\mathbf{n}^T\boldsymbol{\sigma}\mathbf{n}$.

The frictional force (per area) $f_f = \mathbf{d}^T\mathbf{t}$ is often referred to as the shear stress (at $\mathbf{x}$, in direction $\mathbf{n}$) and the normal force (per area) is often referred to as the normal stress (at $\mathbf{x}$, in direction $\mathbf{n}$). If we consider all shear stresses to arise from friction, then we get a notion of states of stress consistent with the Coulomb model of frictional interaction. That is, we consider the stress field $\boldsymbol{\sigma}$ as admissible (or consistent with the Coulomb model) if



$$\mathbf{d}^T\boldsymbol{\sigma}\mathbf{n} \leq -c_F\mathbf{n}^T\boldsymbol{\sigma}\mathbf{n} \tag{4.1}$$

for all $\mathbf{x}$ in the material and for arbitrary directions $\mathbf{d}$ and $\mathbf{n}$ with $\mathbf{d}^T\mathbf{n} = 0$. Specifically, at point $\mathbf{x}$, $\boldsymbol{\sigma}$ relates the force per area (or traction) $\mathbf{t}$ that material on one side of an imaginary plane with normal $\mathbf{n}$ exerts on material on the other side, as $\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$.

When the normal stress $\mathbf{n}^T\boldsymbol{\sigma}\mathbf{n}$ is positive, the material on one side of the imaginary plane is pulling on the material on the other side. This does not arise from a contact/frictional interaction and is a cohesive interaction. Note that Equation (4.1) implies that in the presence of a positive normal stress, the shear stress would have to be zero. In fact, it can be shown that it is not possible to be consistent with Equation (4.1) (for all $\mathbf{d}$ and $\mathbf{n}$) with a positive normal stress, and thus cohesion is not possible with this model.

4.1.2 Stress admissibility

Consider the two dimensional case and states of stress consistent with Inequality (4.1). In this case, given normal $\mathbf{n}$, there are only two directions $\mathbf{d}$ orthogonal to it, namely $\mathbf{d} = \pm\mathbf{R}\mathbf{n}$ where

$$\mathbf{R} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{4.2}$$

In this case, satisfaction of Inequality (4.1) is achieved when

$$\pm\,\mathbf{n}^T\mathbf{R}\boldsymbol{\sigma}\mathbf{n} + c_F\mathbf{n}^T\boldsymbol{\sigma}\mathbf{n} \leq 0 \tag{4.3}$$

for all directions $\mathbf{n}$. Since the Cauchy stress must be symmetric (by conservation of angular momentum), it has an eigen decomposition

$$\boldsymbol{\sigma} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T = \mathbf{Q} \begin{pmatrix} s_1 & \\ & s_2 \end{pmatrix} \mathbf{Q}^T \tag{4.4}$$

where $\mathbf{Q}$ is a rotation matrix. Rewriting Inequality (4.3) in terms of the eigenvalue decomposition gives

$$\pm\, \mathbf{n}^T\mathbf{R}\mathbf{Q}\mathbf{D}\mathbf{Q}^T\mathbf{n} + c_F\mathbf{n}^T\mathbf{Q}\mathbf{D}\mathbf{Q}^T\mathbf{n} \leq 0 \tag{4.5}$$

and since $\mathbf{R}$ and $\mathbf{Q}$ commute (2D rotations commute), satisfaction of Inequality (4.3) is the same as

$$\tilde{\mathbf{n}}^T \left(\pm\mathbf{R}\mathbf{D} + c_F\mathbf{D}\right) \tilde{\mathbf{n}} \leq 0 \tag{4.6}$$

where $\tilde{\mathbf{n}} = \mathbf{Q}\mathbf{n}$. Since Inequality (4.6) must be true for all $\tilde{\mathbf{n}}$ and choice of sign, it is equivalent to require that the maximum of

$$F(\tilde{\mathbf{n}}, h) = \tilde{\mathbf{n}}^T \left(h\mathbf{R}\mathbf{D} + c_F\mathbf{D}\right) \tilde{\mathbf{n}} \tag{4.7}$$

subject to $\|\tilde{\mathbf{n}}\|^2 = 1$ and $h^2 = 1$, is less than 0. Write $\tilde{\mathbf{n}} = (\sin\theta, \cos\theta)^T$, we calculate the maximum of $F(\tilde{\mathbf{n}}, h)$:

$$\begin{aligned}
F(\tilde{\mathbf{n}}, h) &= c_F s_1 \cos^2\theta + \frac{h}{2}(s_1 - s_2) 2\sin\theta\cos\theta + c_F s_2 \sin^2\theta \\
&= c_F s_1 (1 - \sin^2\theta) + \frac{h}{2}(s_1 - s_2) 2\sin\theta\cos\theta + c_F s_2 \sin^2\theta \\
&= c_F s_1 - c_F(s_1 - s_2)\sin^2\theta + \frac{h}{2}(s_1 - s_2) 2\sin\theta\cos\theta \\
&= c_F s_1 - c_F(s_1 - s_2)\frac{1 - \cos 2\theta}{2} + \frac{h}{2}(s_1 - s_2) 2\sin\theta\cos\theta \\
&= (s_1 + s_2)\frac{\tilde{\alpha}}{2} + \frac{1}{2}(s_1 - s_2)(c_F\cos 2\theta + h\sin 2\theta) \\
&\leq (s_1 + s_2)\frac{c_F}{2} + \frac{1}{2}|s_1 - s_2|\sqrt{1 + {c_F}^2}.
\end{aligned}$$

Dividing by $\frac{\sqrt{1+c_F^2}}{\sqrt{2}}$ we obtain that

$$(s_1 + s_2)\frac{c_F}{\sqrt{2}\sqrt{1+c_F^2}} + \frac{|s_1 - s_2|}{\sqrt{2}} \leq 0$$
$$\mathrm{tr}(\boldsymbol{\sigma})\alpha + \left\| \boldsymbol{\sigma} - \frac{\mathrm{tr}(\boldsymbol{\sigma})}{2}\mathbf{I} \right\|_F \leq 0 \tag{4.8}$$

Where $\|\cdot\|_F$ is the Frobenius norm and $\alpha = \frac{c_F}{\sqrt{2}\sqrt{1+c_F^2}}$.

If we solve the analogous maximization problem in three dimensions we obtain the Mohr-Coulomb yield surface [28]. However, there is a simple generalization of Inequality (4.8) that works for both two and three dimensions given by

$$\mathrm{tr}(\boldsymbol{\sigma})\alpha + \left\| \boldsymbol{\sigma} - \frac{\mathrm{tr}(\boldsymbol{\sigma})}{d}\mathbf{I} \right\|_F \leq 0. \tag{4.9}$$

where d is the number of space dimensions. The Drucker-Prager model uses Inequality (4.9) in both two and three dimensions, because it is easier to work with than the Mohr-Coulomb model in 3D and it is a decent approximation of Mohr-Coulomb in that case.

In summary, the Drucker-Prager model for the stress field $\boldsymbol{\sigma}$ requires that

$$y(\boldsymbol{\sigma}(\mathbf{x})) \leq 0 \tag{4.10}$$

for all points $\mathbf{x}$ in the domain occupied by the material, where $y(\boldsymbol{\sigma}) = \mathrm{tr}(\boldsymbol{\sigma})\alpha + \|\boldsymbol{\sigma} - \frac{\mathrm{tr}(\boldsymbol{\sigma})}{d}\mathbf{I}\|_F$ and d is the number of space dimensions.

4.1.3 Kirchhoff stress

As we mentioned in Section 2.3, the Kirchhoff stress $\boldsymbol{\tau}$ is related to the Cauchy stress $\boldsymbol{\sigma}$ as $\boldsymbol{\tau} = J\boldsymbol{\sigma}$ where $J = \det(\mathbf{F})$ is the determinant of the deformation gradient $\mathbf{F}$. We will find the Kirchhoff stress useful when deriving and analyzing properties of the plastic flow. Expressing the Drucker-Prager condition in terms of $\boldsymbol{\tau}$ is simply the requirement that $y(\boldsymbol{\tau}(\mathbf{x})) \leq 0$ for all $\mathbf{x}$ in the domain. Since the constraint can be evaluated as a function of the principal stresses, we can visualize it as the cone $(\tau_1 + \tau_2)\alpha + \frac{|\tau_1 - \tau_2|}{\sqrt{2}} \leq 0$ for 2D problems, or the cone

$(\tau_1+\tau_2+\tau_3)\alpha+\sqrt{\sum_{j=1}^{3}\left(\tau_j-\sum_{i=1}^{3}\frac{\tau_i}{3}\right)^2}\leq 0$ for 3D problems.

The plastic flow will be chosen as a means of satisfying this constraint. When the stress is in the feasible region, there is no plastic flow. However, when the stress reaches the boundary of this region, the plastic flow will be chosen in a manner that prevents the stress from leaving the feasible region. For this reason, the boundary of the feasible region is called the yield surface, since plastic "yield" occurs when the state of stress reaches it.

4.2 Plastic flow

The plastic flow is characterized by the multiplicative decomposition of the deformation gradient $\mathbf{F}=\mathbf{F}^E\mathbf{F}^P$, however it is convenient for analysis and constitutive modeling to consider evolution of the left elastic Cauchy-Green strain $\mathbf{B}^E=\mathbf{F}^E\mathbf{F}^{E^T}=\mathbf{F}\mathbf{C}^{P^{-1}}\mathbf{F}^T$ where $\mathbf{C}^P=\mathbf{F}^{P^T}\mathbf{F}^P$ is the right plastic Cauchy-Green strain. Recalling that the deformation then evolves as $\frac{D\mathbf{F}}{Dt}=(\nabla\mathbf{v})\mathbf{F}$,

$$\frac{D\mathbf{B}^E}{Dt}=(\nabla\mathbf{v})\mathbf{B}^E+\mathbf{B}^E(\nabla\mathbf{v})^T+\mathbf{F}\frac{D\mathbf{C}^{P^{-1}}}{Dt}\mathbf{F}^T.$$

The term $\mathbf{F}\frac{D\mathbf{C}^{P^{-1}}}{Dt}\mathbf{F}^T$ is the Lie derivative of the of $\mathbf{B}^E$ with respect to $\mathbf{v}$ so we denote it as $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E$. The Lie derivative of $\mathbf{B}^E$ is its rate of change independent of deformation in the flow, and it will be determined to define the plastic flow as a means of satisfying the stress feasibility condition in Inequality (4.10). When the stress is inside the feasible region, $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E=\mathbf{0}$. However, when the stress is on the yield surface, it will be chosen to guarantee that $\dot{y}(t)\leq 0$, thus preventing any future elastic stresses attaining values outside the feasible region. This can be done in infinitely many ways, however care must be taken to avoid artifacts associated with non-volume preserving plastic flows, as well as to guarantee that the plastic flow increases entropy (or decreases the total energy). In particular,

$$\begin{aligned}\dot{y}(t)&=\frac{\partial y}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}(\mathbf{B}^E(t))):\frac{\partial\boldsymbol{\tau}}{\partial\mathbf{B}^E}(\mathbf{B}^E(t)):\frac{D\mathbf{B}^E}{Dt}(t)\\&=\frac{\partial y}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}(\mathbf{B}^E(t))):\frac{\partial\boldsymbol{\tau}}{\partial\mathbf{B}^E}(\mathbf{B}^E(t)):\left((\nabla\mathbf{v})\mathbf{B}^E+\mathbf{B}^E(\nabla\mathbf{v})^T+\mathcal{L}_{\mathbf{v}}\mathbf{B}^E\right)\end{aligned}\tag{4.11}$$

Here, the : operator denotes a generalized dot product to express the chain rule when differentiating the composition of scalar and matrix valued functions of matrix argument. The material derivative $\frac{D}{Dt}$ appears in the chain rule because we are considering how y evolves with time for one particle of the continuum. Defining β as the rate of change of y in the absence of plasticity ($\mathcal{L}_{\mathbf{v}}\mathbf{B}^E = \mathbf{0}$) gives

$$\beta = \frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}(\mathbf{B}^E(t))) : \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{B}^E}(\mathbf{B}^E(t)) : \left((\nabla \mathbf{v})\mathbf{B}^E + \mathbf{B}^E(\nabla \mathbf{v})^T\right) \tag{4.12}$$

When the stress criteria is satisfied, we have $y < 0$ and there is no plastic flow, ($\mathcal{L}_{\mathbf{v}}\mathbf{B}^E = \mathbf{0}$). However, when we reach the boundary of the feasible region in stress space, we choose $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E$ so that $\dot{y}(t) = 0$. To conclude

$$\begin{cases} \mathcal{L}_{\mathbf{v}}\mathbf{B}^E = \mathbf{0}, & \text{if } y < 0 \text{ or if } y = 0 \text{ and } \beta \leq 0 \\ \text{choose } \mathcal{L}_{\mathbf{v}}\mathbf{B}^E \text{ so that } \dot{y}(t) = 0 & \text{if } y = 0 \text{ and } \beta > 0. \end{cases} \tag{4.13}$$

In the next section, we are going to talk about how to choose the direction of plastic flow $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E$.

4.2.1 Choosing the direction of the plastic flow

Energy dissipation In order to insure that stress never leaves the feasible region, the direction of plastic flow $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E$ only needs to have non-zero component on $\frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}(\mathbf{B}^E(t))) : \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{B}^E}(\mathbf{B}^E(t)) : \mathcal{L}_{\mathbf{v}}\mathbf{B}^E$. Thus, for a given value of $\mathbf{B}^E$, there are infinitely many choices of $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E$ that will suffice in preventing stresses outside the feasible region. However, care must be taken to insure that the plastic flow does not decrease the entropy of the system. Or more specifically, that it does not instantaneously increase the rate of change of the total energy and thus violate the second law of thermodynamics [16].

We refer to Section A.2 for detailed derivation of two important conclusions in the case of isotropic potential energy,:

- $\mathbf{B}^E$ and $\boldsymbol{\tau}$ have the same eigen decomposition.

- The total energy $E(t) = KE(t) + PE(t)$ satisfies

$$E(t+\Delta t) - E(t) = W^{\mathbf{t}}(t, \Delta t) - \int_t^{t+\Delta t} \int_{\Omega^0} \dot{w}^P(\mathbf{X}, s) d\mathbf{X} ds \tag{4.14}$$

where $W^{\mathbf{t}}(t, \Delta t)$ is the work done by external traction $\mathbf{t}$ boundary conditions and

$$\dot{w}^P = \boldsymbol{\tau} : -\frac{1}{2}\mathcal{L}_{\mathbf{v}}\mathbf{B}^E\mathbf{B}^{E^{-1}}.$$

In the absence of plasticity, the work done by the mechanical stresses is equal to the negative change in the potential, and this leads to exact conservation of energy (minus the effect of the boundary conditions and external forcing). In the case of plasticity, the total energy may go up or down from the work done by the mechanical stress, and the $\dot{w}^P$ term quantifies that. Specifically, the plastic flow must be designed in a way that ensures non-negative $\dot{w}^P$, otherwise total energy may increase due to plasticity, which would violate the second law of thermodynamics.

Principle of maximum plastic dissipation The principle of maximum plastic dissipation [5] seeks to design the plastic flow in a way that maximizes $\dot{w}^P$ to respect this concern. This leads to an associative plastic flow where

$$-\frac{1}{2}\mathcal{L}_{\mathbf{v}}\mathbf{B}^E\mathbf{B}^{E^{-1}} \in \frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}).$$

For Drucker-Prager yield surface from Inequality (4.10),

$$\frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}) = \alpha\mathbf{I} + \frac{\mathrm{dev}(\boldsymbol{\tau})}{\|\mathrm{dev}(\boldsymbol{\tau})\|_F} \tag{4.15}$$

where we introduce the deviatoric operator to act on matrices:

$$\mathrm{dev}(\mathbf{A}) := \mathbf{A} - \frac{1}{d}\mathrm{trace}(\mathbf{A})\mathbf{I}, \tag{4.16}$$

i.e. $\mathrm{dev}(\mathbf{A})$ gives the deviatoric part of any arbitrary square matrix $\mathbf{A}$ of size $d \times d$.

Volume preservation Unfortunately, there is a potential for excessive volume loss or gain in the model. Indeed, simply using $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E\mathbf{B}^{E^{-1}} \in \frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau})$ will tend to cause excessive volume gain during sheering [28]. However, we remedy the artifact by using the deviatoric part $\frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau})$ which induces the non-associative rule:

$$\mathcal{L}_{\mathbf{v}}\mathbf{B}^E\mathbf{B}^{E^{-1}} = -\gamma\mathbf{G}$$

where

$$\mathbf{G} = \operatorname{dev}\left(\frac{\frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau})}{\|\frac{\partial y}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau})\|_F}\right) = \frac{\operatorname{dev}(\boldsymbol{\tau})}{\|\operatorname{dev}(\boldsymbol{\tau})\|_F} \tag{4.17}$$

and γ is a scalar. The second equality here follows from Equation 4.15.

Furthermore, we show in Section A.4 that the modification still guarantees that $\dot{w}^P$ is non-negative and thus satisfies the second law of thermodynamics. In summary, the plasticity is expressed through $\mathcal{L}_{\mathbf{v}}\mathbf{B}^E$ as

$$\mathcal{L}_{\mathbf{v}}\mathbf{B}^E = \begin{cases} \mathbf{0}, & \text{if } y < 0 \text{ or if } y = 0 \text{ and } \beta \leq 0 \\ -\gamma\mathbf{G}, & \text{if } y = 0 \text{ and } \beta > 0. \end{cases} \tag{4.18}$$

4.3 Derivation of return mapping algorithm from plastic flow

The return mapping algorithm is the discrete equivalent to solving for a strain that satisfies the plastic flow rule in Equation (4.18) and that lies in the Drucker-Prager yield surface. In this section first we outline the method of Simo and Meschke [31] to derive the discrete equations from their continuous versions. This procedure starts by assuming there is no plastic flow and a return mapping algorithm is derived from the flow equations that shows how to project back to the yield surface if the assumption of no plastic flow is invalid.

Consider the evolution of $\mathbf{B}^E$ from time t^n to time $t^{n+1} = t^n + \Delta t$. We consider this evolution per particle, and thus it is useful to take a Lagrangian view. Recall from Section 2.2 that the deformation gradient $\mathbf{F}$ defines the deformation from the initial configuration $\bar{\Omega}$ to the time t configuration Ω^t, the Jacobian $\tilde{\mathbf{F}} = \frac{\partial \tilde{\phi}}{\partial \tilde{\mathbf{x}}}$ defines the deformation from the time t^n

configuration Ω^{t^n} to the time t configuration Ω^t, where $t \geq t^n$. These are related as $\mathbf{F} = \tilde{\mathbf{F}}\mathbf{F}^n$.

Define $\mathbf{B}^{E*} = \tilde{\mathbf{F}}^{-1}\mathbf{B}^E\tilde{\mathbf{F}}^{-T}$. Let us consider the difference between the evolution of $\mathbf{B}^{E*}$ and $\mathbf{B}^E$ in absence of plasticity at time $t^n < t < t^{n+1}$. By the definition of $\mathbf{B}^{E*}$,

$$\frac{D\mathbf{B}^{E*}}{Dt} = -2\gamma\tilde{\mathbf{F}}^{-1}\mathbf{G}\tilde{\mathbf{F}}\mathbf{B}^{E*},$$

therefore in absence of plasticity $\mathbf{B}^{E*}$ is constant since $\frac{D\mathbf{B}^{E*}}{Dt} = \mathbf{0}$. In contrast, $\mathbf{B}^E\big|_t = \tilde{\mathbf{F}}\Big|_t \mathbf{B}^E\big|_{t^n}\tilde{\mathbf{F}}^T\Big|_t$ in the same case. In other words, $\mathbf{B}^{E*}$ is constant along characteristics except for the effect of plasticity, but at the same time $\mathbf{B}^E$ would also be stretched by the flow. This isolation of the plastic part allows for a more intuitive discretization. Specifically, combined with the initial value $\mathbf{B}^{E*}\big|_{t^n} = \mathbf{B}^E\big|_{t^n}$, we can use the exponential approximation

$$\mathbf{B}^{E*}\big|_{t^{n+1}} \approx \exp(-2\delta\gamma\,\tilde{\mathbf{F}}^{-1}\mathbf{G}\tilde{\mathbf{F}})\Big|_{t^{n+1}}\mathbf{B}^E\big|_{t^n}$$

where $\delta\gamma \geq 0$ will be used to enforce the constraint $y(\boldsymbol{\tau}(\mathbf{B}^E\big|_{t^{n+1}})) \leq 0$. Multiplying the approximation by $\tilde{\mathbf{F}}\Big|_{t^{n+1}}$ on the left and $\tilde{\mathbf{F}}^T\Big|_{t^{n+1}}$ on the right, and recalling the definition of $\mathbf{B}^{E*}$, we obtain

$$\begin{aligned}
\mathbf{B}^E\big|_{t^{n+1}} &= \tilde{\mathbf{F}}\Big|_{t^{n+1}}\mathbf{B}^{E*}\big|_{t^{n+1}}\tilde{\mathbf{F}}^T\Big|_{t^{n+1}} && (4.19)\\
&\approx \tilde{\mathbf{F}}\Big|_{t^{n+1}}\exp(-2\delta\gamma\,\tilde{\mathbf{F}}^{-1}\mathbf{G}\tilde{\mathbf{F}})\Big|_{t^{n+1}}\mathbf{B}^E\big|_{t^n}\tilde{\mathbf{F}}^T\Big|_{t^{n+1}} && (4.20)\\
&= \tilde{\mathbf{F}}\Big|_{t^{n+1}}\tilde{\mathbf{F}}^{-1}\Big|_{t^{n+1}}\exp(-2\delta\gamma\,\mathbf{G})|_{t^{n+1}}\tilde{\mathbf{F}}\Big|_{t^{n+1}}\mathbf{B}^E\big|_{t^n}\tilde{\mathbf{F}}^T\Big|_{t^{n+1}} && (4.21)\\
&= \exp(-2\delta\gamma\,\mathbf{G})|_{t^{n+1}}\tilde{\mathbf{F}}\Big|_{t^{n+1}}\mathbf{B}^E\big|_{t^n}\tilde{\mathbf{F}}^T\Big|_{t^{n+1}}. && (4.22)
\end{aligned}$$

We use notation $\mathbf{B}^{E,\mathrm{tr}} = \tilde{\mathbf{F}}\Big|_{t^{n+1}}\mathbf{B}^E\big|_{t^n}\tilde{\mathbf{F}}^T\Big|_{t^{n+1}}$ for the elastic strain we would get without the effect of plasticity (trial elastic strain). We also use $\mathbf{B}^{E,n+1}$ for $\mathbf{B}^E\big|_{t^{n+1}}$ for slightly cleaner notation.

If the singular value decomposition of $\mathbf{F}^E$ is given by $\mathbf{F}^E = \mathbf{U}^E\boldsymbol{\Sigma}^E\mathbf{V}^{E^T}$, then $\mathbf{B}^{E,\mathrm{tr}} = \mathbf{F}^E\mathbf{F}^{E^T} = \mathbf{U}^E\boldsymbol{\Sigma}^{E^2}\mathbf{U}^{E^T}$. For isotropic yield surface function, we show in Section A.2 that $\frac{\partial y}{\partial \boldsymbol{\tau}}$ and $\mathbf{B}^{E,\mathrm{tr}}$ have the same eigenvectors, so $\mathbf{G}(\boldsymbol{\tau}(\mathbf{B}^{E,n+1})) = \mathbf{U}^E\hat{\mathbf{G}}(\boldsymbol{\Sigma}^{E,n+1})\mathbf{U}^{E^T}$, and $\mathbf{B}^{E,n+1} =$

$\mathbf{U}^E \left(\mathbf{\Sigma}^{E,n+1}\right)^2 \mathbf{U}^{E^T}$, then we may write Equation (4.19) as

$$\mathbf{U}^E \left(\mathbf{\Sigma}^{E,n+1}\right)^2 \mathbf{U}^{E^T} = \exp\left(-2\delta\gamma \mathbf{U}^E \hat{\mathbf{G}}(\mathbf{\Sigma}^{E,n+1}) \mathbf{U}^{E^T}\right) \mathbf{U}^E \mathbf{\Sigma}^{E^2} \mathbf{U}^{E^T} \tag{4.23}$$

$$= \mathbf{U}^E \exp\left(-2\delta\gamma \hat{\mathbf{G}}(\mathbf{\Sigma}^{E,n+1})\right) \mathbf{\Sigma}^{E^2} \mathbf{U}^{E^T}. \tag{4.24}$$

Multiplying both sides of Equation (4.23) by $\mathbf{U}^{E^T}$ on the left and by $\mathbf{U}^E$ on the right, and taking log results in

$$2\ln\left(\mathbf{\Sigma}^{E,n+1}\right) = -2\delta\gamma \hat{\mathbf{G}}(\mathbf{\Sigma}^E) + 2\ln\left(\mathbf{\Sigma}^E\right). \tag{4.25}$$

Writing the above equation in Hencky-strain by defining

$$\boldsymbol{\epsilon}^{\mathrm{tr}} := \ln \mathbf{\Sigma}^E \quad \text{and} \quad \boldsymbol{\epsilon}^{n+1} := \ln \mathbf{\Sigma}^{E,n+1}. \tag{4.26}$$

Here we use $\boldsymbol{\epsilon}^{\mathrm{tr}}$ and $\boldsymbol{\epsilon}^{n+1}$ instead of the more descriptive $\boldsymbol{\epsilon}^{E,\mathrm{tr}}$ and $\boldsymbol{\epsilon}^{E,n+1}$ for brevity of notation, but we keep in mind that the Hencky-strain is defined to describe the admissible stress and therefore corresponds to the elastic deformation $\mathbf{F}^E$. We may simplify and rearrange Equation (4.25)

$$\boldsymbol{\epsilon}^{\mathrm{tr}} - \boldsymbol{\epsilon}^{n+1} = \delta\gamma \hat{\mathbf{G}}. \tag{4.27}$$

This is our discrete flow rule. In the return mapping algorithm, we want to solve for $\boldsymbol{\epsilon}^{n+1}$ satisfies Equation (4.27) subject to the constraint

$$y(\boldsymbol{\tau}(\boldsymbol{\epsilon}^{n+1})) \leq 0. \tag{4.28}$$

Solving Equation (4.27) and (4.28) can be seen as a ray-cone intersection problem.

Equation (4.27) has no solution if $\mathrm{trace}(\boldsymbol{\epsilon}^{\mathrm{tr}}) \geq 0$. In this case the sand is in extension and we project to the tip $\boldsymbol{\epsilon}^{n+1} = \mathbf{0}$. Recall the expression for $\mathbf{G}$ from Equation (4.17), in principal space this becomes $\hat{\mathbf{G}} = \frac{\mathrm{dev}(\hat{\boldsymbol{\tau}})}{\|\mathrm{dev}(\hat{\boldsymbol{\tau}})\|_F}$, where $\hat{\boldsymbol{\tau}}$ and $\hat{\mathbf{G}}$ are diagonal. From (§A.3) we

have

$$\hat{\boldsymbol{\tau}} = \frac{\partial \psi}{\partial \boldsymbol{\epsilon}} = 2\mu\boldsymbol{\epsilon}^{n+1} + \lambda \mathrm{tr}(\boldsymbol{\epsilon}^{n+1})\mathbf{I}$$

because we use the energy density $\psi(\boldsymbol{\epsilon}) = \mu \mathrm{tr}((\boldsymbol{\epsilon})^2) + \frac{1}{2}\lambda \mathrm{tr}(\boldsymbol{\epsilon})^2$. Thus $\hat{\mathbf{G}} = \frac{\mathrm{dev}(\boldsymbol{\epsilon}^{n+1})}{\|\mathrm{dev}(\boldsymbol{\epsilon}^{n+1})\|_F}$. Using Equation (4.27), we can see that $\mathrm{tr}(\boldsymbol{\epsilon}) = \mathrm{tr}(\boldsymbol{\epsilon}^{n+1})$, since $\mathrm{tr}(\hat{\mathbf{G}}) = 0$. Thus

$$\mathrm{dev}(\boldsymbol{\epsilon}^{\mathrm{tr}}) - \mathrm{dev}(\boldsymbol{\epsilon}^{n+1}) = \delta\gamma \frac{\mathrm{dev}(\boldsymbol{\epsilon}^{n+1})}{\|\mathrm{dev}(\boldsymbol{\epsilon}^{n+1})\|_F},$$

and collecting like terms we have $\mathrm{dev}(\boldsymbol{\epsilon}^{\mathrm{tr}}) = \left(1 + \frac{\delta\gamma}{\|\mathrm{dev}(\boldsymbol{\epsilon}^{n+1})\|_F}\right)\mathrm{dev}(\boldsymbol{\epsilon}^{n+1})$. Thus $\hat{\mathbf{G}} = \frac{\mathrm{dev}(\boldsymbol{\epsilon}^{\mathrm{tr}})}{\|\mathrm{dev}(\boldsymbol{\epsilon}^{\mathrm{tr}})\|_F}$. Then plugging the equation for the ray $\boldsymbol{\epsilon}^{n+1} = \boldsymbol{\epsilon} - \delta\gamma \frac{\mathrm{dev}(\boldsymbol{\epsilon}^{\mathrm{tr}})}{\|\mathrm{dev}(\boldsymbol{\epsilon}^{\mathrm{tr}})\|_F}$, into the equation for the cone $y(\boldsymbol{\tau}(\boldsymbol{\epsilon}^{n+1})) = \mathbf{0}$, and solving for $\delta\gamma$, we obtain

$$\delta\gamma = \|\mathrm{dev}(\boldsymbol{\epsilon}^{\mathrm{tr}})\|_F + \left(\frac{d\lambda + 2\mu}{2\mu}\right)\mathrm{trace}(\boldsymbol{\epsilon}^{\mathrm{tr}})\alpha. \tag{4.29}$$

If $\delta\gamma \leq 0$ we intersect the cone from the inside and thus don't need to project and have $\boldsymbol{\epsilon}^{n+1} = \boldsymbol{\epsilon}^{\mathrm{tr}}$. Otherwise we project to the cone and we finally have our update rule for $\boldsymbol{\epsilon}$

$$\boldsymbol{\epsilon}^{n+1} = \boldsymbol{\epsilon}^{\mathrm{tr}} - \delta\gamma\boldsymbol{\epsilon}^{\mathrm{tr}}.$$

4.4 Results

4.4.1 Flowing and Piling

We demonstrate the accuracy of our model by showing the characteristic behaviors of sand flowing and piling. In Figure 4.1, we simulate sand flowing inside an hourglass. The sand forms a smooth granular flow and piles up at the bottom. Figure 4.2 shows a stream of sand inflow hitting a high frictional surface. We compare this simulation with real world footage. Our model successfully captures the interesting avalanche instability [42] of this experiment.

Figure 4.1: Sand falls through the narrow neck of an hourglass, accumulating at the bottom.

Figure 4.2: Sand is poured from a spout into a pile in a lab (left) and with our method (right)

4.4.2 Easy Tuning

In Figure 4.3, we simulate columns of dry sand with different friction angles collapsing on the ground. Different friction angles directly affect the interaction between sand grains, therefore the final piling angle. While the real Young's modulus of sand is 3.537×10^7, we found that sometimes choosing a moderately smaller value does not change the visual appearance. In Figure 4.4, we show 2D inflow simulations with different Young's modulus. A moderately smaller Young's modulus improves the efficiency of the implicit solve. However, the material may exhibit jiggling behavior if it is too small. We assert physically accurate

Young's modulus is always the best choice unless an artistic elastic effect is desirable.

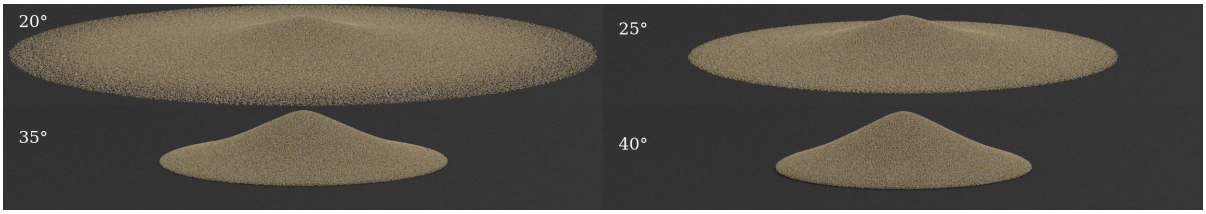


Figure 4.3: Varying the friction angle changes the shape of a pile of sand. A larger angle produces a taller sand pile with steeper sides.

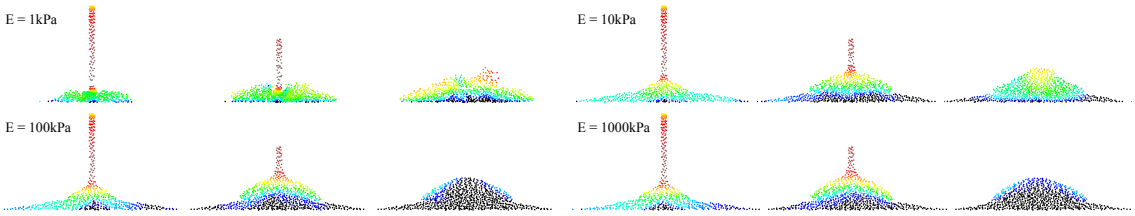


Figure 4.4: This simulation shows the effects of Young's modulus on the behavior of a simulation. Sand with a very low Young's modulus tends to be bouncy. The behavior is more like sand as the Young's modulus approaches its physical value.

4.4.3 Two-way Coupling

The benefits of using MPM include automatic self collision and coupling between different materials. In Figure 4.5, we show an elastic ball interacting with a dry sand castle. MPM naturally handles the two-way coupling without requiring any additional treatment other than assigning different constitutive models to different particles.

4.4.4 Drawing and Scooping

We further demonstrate the versatility of our method by performing various tasks in a sand box. Figure 4.6 shows drawing a butterfly with a wooden stick. Figure 4.7 shows raking sand in a Zen garden. Figure 4.8 shows scooping sand.

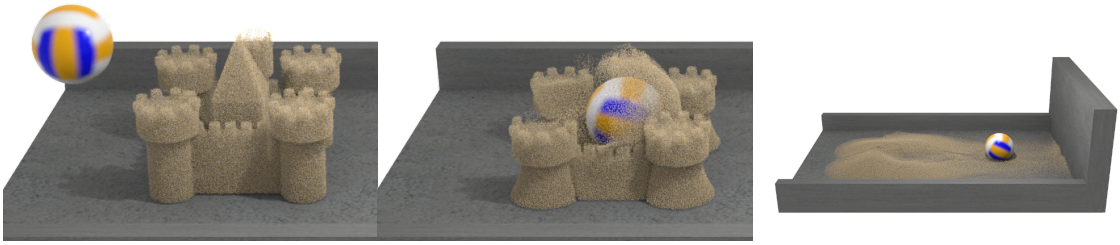


Figure 4.5: A sand castle is hit with a deformable ball while falling. The sand and ball are fully coupled in the imulation.

Figure 4.6: A stick is dragged through a bed of sand, tracing out a butterfly shape in the sand.

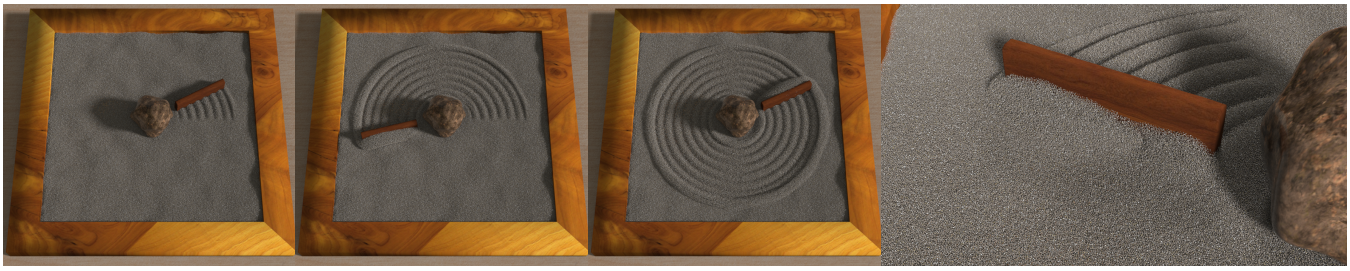


Figure 4.7: A rake is dragged around a rock, producing a circular pattern in the sand.

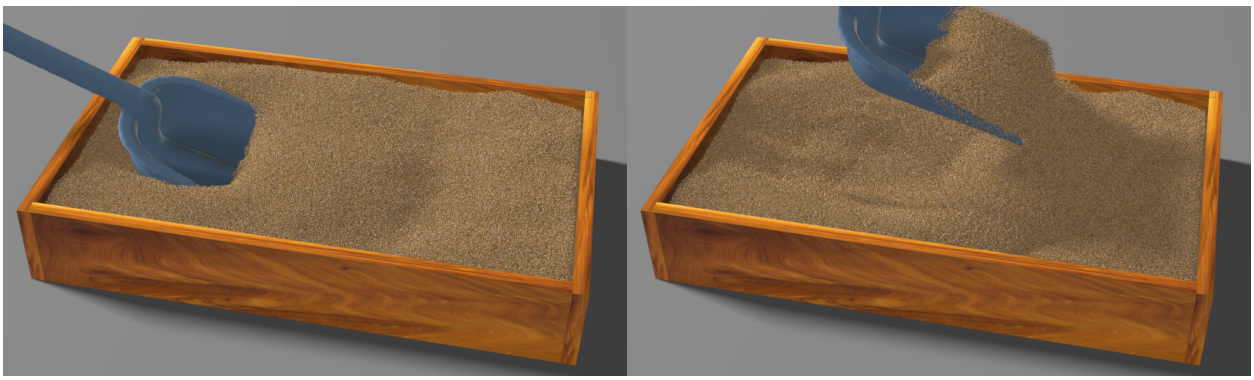


Figure 4.8: A shovel digs through sand and pushes it aside.

CHAPTER 5

Thin shell with frictional contact

Terzopoulos et al [38, 37] pioneered cloth simulation by presenting codimensional elastic objects by a Lagrangian mesh. Such approaches are still primarily used today. With the Lagrangian mesh model, individual particles are tracked and mesh polygons or segments are used to approximate the deformation gradient. The mechanics of elasticity are naturally discretized with this Lagrangian view. However additional modeling is required to include the effects of self and external contact since these phenomena may cause interactions between distant regions in the mesh. We present a new approach for codimensional elasticity that uses MPM discretization to model frictional contact with a continuum view. Unlike traditional approaches, our elastoplastic description completely characterizes all collision/contact response in the continuum and requires no separate post-processing.

In this chapter, we first discuss our notation as well as related mathematical details (Section (§5.1)). We then outline the kinematic details of continuum shells and our splitting into Kirchhoff-Love and shearing/compression modes (Section (§5.2)). Next, we describe our continuous elastoplasticity model in terms of the potential energy density in the shell and its relation to plasticity constraints and associated return mappings for frictional contact as well as wrinkling and denting (Section (§5.3)). We then discuss subd FEM discretization of the potential energy and the derivatives of the energy with respect to discrete degrees of freedom (Section (§5.4)). Lastly we show that, as with many models defined in terms of an elastic potential [34, 24, 23], it is straightforward to discretize our model with MPM (Section (§5.5)).

5.1 Mathematical Details and Notation

In addition to the general convention of notations we adopt in Section 3.1, we introduce a few more for the convenience of this chapter. 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 $[\mathbf{M}] \in \mathbb{R}^{3\times 3}$). Unless otherwise stated, we use the summation convention for repeated indices. 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 5.2. 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 5.1.

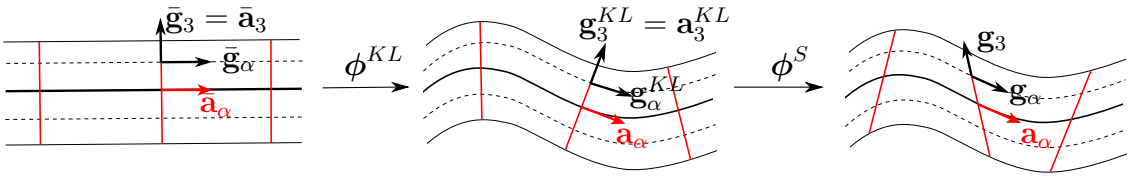


Figure 5.1: **Continuum shell/Kirchhoff-Love splitting**. Mid-surface tangents and fibers are shown in red. Laminae are shown as dashed curves, and the local frame at a point on a lamina is shown in black. On the left is the undeformed reference configuration, while the deformed configuration is on the right, and the middle shows the intermediate Kirchhoff-Love deformation.

5.2 Shell Kinematics

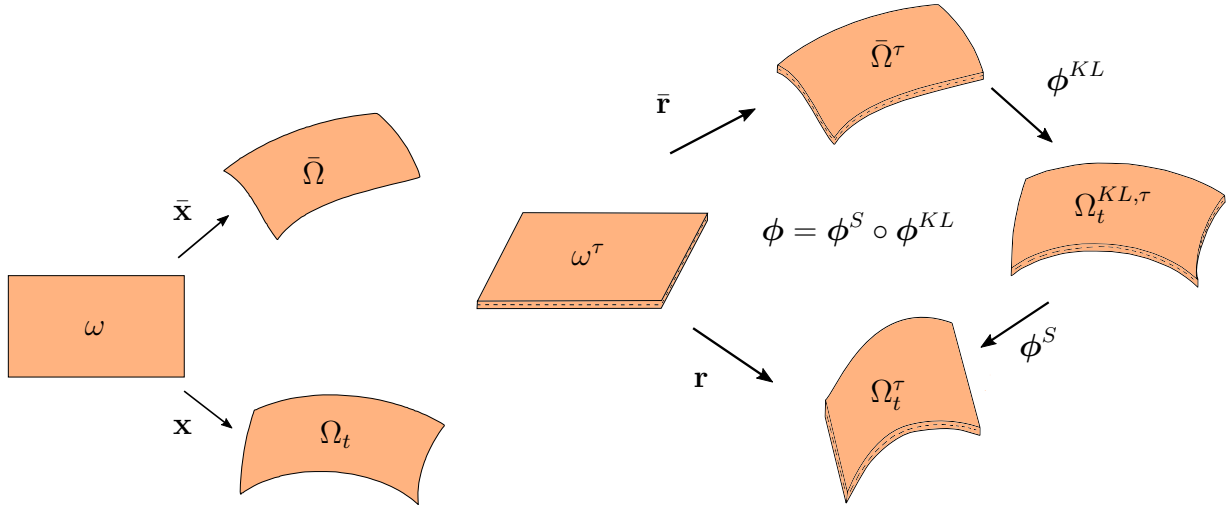


Figure 5.2: **Shell Kinematics**. On the left, the mid-surface mappings are illustrated, and on the right the corresponding volumetric shell mappings are shown.

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 5.1).

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 satifsies $\mathbf{a}_3^{KL} = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|}$ 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 5.1).

5.2.1 Deformation Gradient

In Section 2.2, we introduced how to represent and evolve deformation gradient $\mathbf{F}$ at each point $\mathbf{X}$ in space, and later on in Chapter 4 , each particle represents a small chunk of material that exhibits constitutive behaviors in response to the deformation gradient $\mathbf{F}_p$ (or $\mathbf{F}_p^E$ in presence of plasticity) associated to this particle. In this section, we will discuss how to represent the deformation gradient in continuum shell.

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 $\bar{\mathbf{g}}_\alpha \cdot \bar{\mathbf{g}}_3 = 0$.

5.2.2 Plasticity

As in Jiang et al.[23], 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.[23] 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. [23] 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 defomraiton. The exprssion 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

In this section, we discuss the elastic stress and plastic constraints used in the shell model. We first define our elastoplastic constitutive response to deformation and frictional contact in terms of potential energy. This consists of the response to Kirchhoff-Love motion, and compression and shearing motions. In the sections that follow, we discuss the two potential energy and their yield surface and return mapping accordingly.

5.3.1 Elastic stress

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.

5.3.2 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. According to Section A.3, the stress in the material is the derivative of this potential with respect to strain. 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. [10] in Figures 5.11a and 5.11b. 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 $\mathbf{F}^{KL,E} = F_{ij}^{KL,E}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j$. Here the directions $\bar{\mathbf{a}}_\alpha$ are the tangents to the midsurface 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 $F_{\alpha\beta}^{KL,E}$ 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 $C_{ij} = F_{ki}^{KL,E}F_{kj}^{KL,E}$. 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. As in Chapter 4, we use a model that is quadratic in the right Hencky-strain $\left[\boldsymbol{\epsilon}^R\right] = \frac{1}{2}\log([\hat{\mathbf{C}}])$

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

Here the $\epsilon_{\alpha\beta}^R$ 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 Section (§B.6)).

Similar to Chapter 2, the Kirchhoff stress $\boldsymbol{\tau}$ can derived as

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

Here we write the stress in terms of the basis defined by the directions $\mathbf{q}_i^{KL,E}$ obtained from the QR decomposition $\mathbf{F}^{KL,E} = r_{ij}^{KL,E}\mathbf{q}_i^{KL,E} \otimes \bar{\mathbf{a}}_j$ 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 $\mathbf{q}_\alpha^{KL,E}$ are tangent to the deformed lamina and the third direction $\mathbf{q}_3^{KL,E}$ is normal to the mid-surface. Therefore, since $\boldsymbol{\tau}^{KL}$ is expressed only in terms of $\mathbf{q}_\alpha^{KL,E}$, 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 $\epsilon_{\alpha\beta}^L$ 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 $r_{\alpha\beta}^{KL,E}$. These are the components of the deformation gradient $\mathbf{F}^{KL,E}$ related to the lamina strain.

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.5). 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.

As we described in Section 3.7, the yield condition is satisfied via projection (or return mapping) of the stress to the feasible region. Specifically, 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 $\boldsymbol{\tau}^{KL^{\mathrm{tr}}}$. 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 principle 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 (5.15). We provide more detail in this derivation in Section B.6.

5.3.3 Frictional Contact Potential

As in Jiang et al.[23], 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. 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 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

$$\chi(\mathbf{F}^{S,E}(\xi_1,\xi_2,0)) = \frac{\gamma}{2}\left(r_{13}^{S,E2} + r_{23}^{S,E2}\right) + f(r_{33}^{S,E}). \tag{5.25}$$

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 Section 5.3.4.

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. As in Section 4.1, 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.

Recall that 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 $\boldsymbol{\tau}^S$. Plugging in the expression for $\boldsymbol{\tau}^S$ from 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 principle stresses of $\boldsymbol{\tau}^{KL}$, only the components $s_i^E = r_{i3}^{S,E}$ 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 $\mathbf{a}_3^{E^{\text{tr}}} = s_i^{E^{\text{tr}}} \mathbf{q}_i^{S,E}$ to the projected $\mathbf{a}_3^E = s_i^E \mathbf{q}_i^{S,E}$ where the trial and projected coefficients are related through

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

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.[23] 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, 32]. 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.5). 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 Section B.1.

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 an 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 modeled in previous chapters (e.g. granular sand, snow and soil) simulated with MPM.

we classify particles as either: (*i*) traditional MPM particles, (*ii*) subd particles or (*iii*)

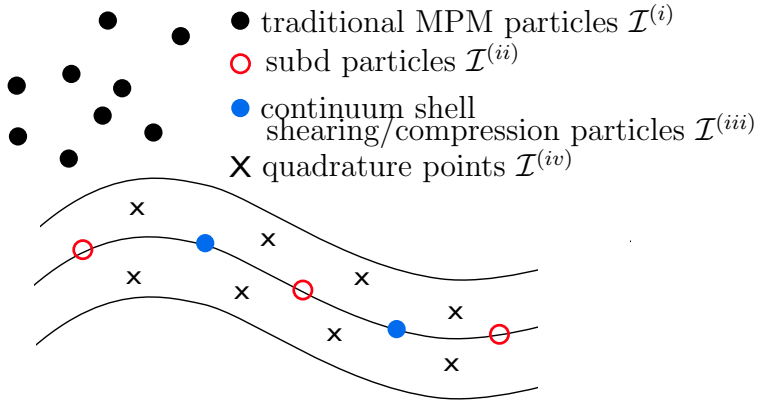


Figure 5.3: **Particle type classification**. A schematic illustration of the different types of MPM particles and quadrature points.

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

We follow the MPM algorithm presented in Chapter 3. In particular, the transfer to grid, and transfer to particles steps are identical to Section 3.3 and Section 3.5. However, we need to take care in the force computation and other plasticity related steps. We present

the details in this section.

5.5.1 Grid Momentum Update

The potential energy Ψ is a sum of the contributions from the shell Ψ^{CS} and from traditional MPM particles ψ^M as in Equation 3.5 used for coupling multiple materials.

$$\Psi(\hat{\mathbf{x}}) = \sum_{p\in\mathcal{I}^{(i)}} \psi^M\left(\mathbf{F}_p^{E,\mathrm{tr}}(\hat{\mathbf{x}})\right)V_p^0 + \Psi^{CS}(\hat{\mathbf{x}}) \tag{5.37}$$

$$\begin{aligned}\Psi^{CS}(\hat{\mathbf{x}}) = \sum_{p\in\mathcal{I}^{(iii)}} \chi\Big(&\mathbf{a}_{p\alpha}(\mathbf{x}^{KL}(\hat{\mathbf{x}}))\otimes\bar{\mathbf{a}}_{p\alpha} \\ &+ \mathbf{a}_{p3}^{E,\mathrm{tr}}(\mathbf{x}^{KL}(\hat{\mathbf{x}}))\otimes\bar{\mathbf{a}}_{p3}\Big)V_p^0 \\ + \sum_{q\in\mathcal{I}^{(iv)}} \psi\Big(&\mathbf{F}_q^{KL,E\mathrm{tr}}(\mathbf{x}^{KL}(\hat{\mathbf{x}}))\Big)V_q^0.\end{aligned} \tag{5.38}$$

Here ψ^M is the contribution from the standard MPM potential discretization 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 $\hat{\mathbf{x}}$.

For particles of type (i), this dependence follows from the updated Lagrangian formulation as in Equation Equations (3.6) . For particles of type (iii), the dependence follows from

the updated Lagrangian

$$\mathbf{a}_{p3}^{E,\mathrm{tr}}(\hat{\mathbf{x}}) = \left(\sum_{\mathbf{i}} \hat{\mathbf{x}}_{\mathbf{i}} \otimes \nabla w_{\mathbf{i}p}^{n} \right) \mathbf{a}_{p3}^{E,n} \tag{5.39}$$

and from interpolation the $\mathbf{x}^{KL}(\hat{\mathbf{x}})$ in Equation (5.40) in $\mathbf{a}_{p\alpha}(\mathbf{x}^{KL}(\hat{\mathbf{x}}))$. The mid-surface control points for the shell are interpolated from the grid degrees of freedom as

$$\hat{\mathbf{x}}_p = \sum_{\mathbf{i}} \hat{\mathbf{x}}_{\mathbf{i}} w_{\mathbf{i}p}^{n}, \quad p \in \mathcal{I}^{(ii)}. \tag{5.40}$$

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

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

$$\frac{\partial \Psi}{\partial \mathbf{x}_{\mathbf{i}}}(\hat{\mathbf{x}}) = \mathbf{f}_{\mathbf{i}}^{(i)}(\hat{\mathbf{x}}) + \mathbf{f}_{\mathbf{i}}^{(ii)}(\hat{\mathbf{x}}) + \mathbf{f}_{\mathbf{i}}^{(iii)}(\hat{\mathbf{x}}) \tag{5.41}$$

$$\mathbf{f}_{\mathbf{i}}^{(i)}(\hat{\mathbf{x}}) = \sum_{p \in \mathcal{I}^{(i)}} \frac{\partial \psi^{M}}{\partial \mathbf{F}^{E}}(\mathbf{F}_p^{E,\mathrm{tr}}(\hat{\mathbf{x}})) \mathbf{F}_p^{E,n^T} \nabla w_{\mathbf{i}p}^{n} V_p^0 \tag{5.42}$$

$$\mathbf{f}_{\mathbf{i}}^{(ii)}(\hat{\mathbf{x}}) = \sum_{p \in \mathcal{I}^{(ii)}} w_{\mathbf{i}p}^{n} \mathbf{f}_p^{KL}(\mathbf{x}^{KL}(\hat{\mathbf{x}})) \tag{5.43}$$

$$\mathbf{f}_{\mathbf{i}}^{(iii)}(\mathbf{x}^*) = \sum_{p \in \mathcal{I}^{(iii)}} \boldsymbol{\tau}_p^S \tilde{\mathbf{a}}_p^{\beta} : \frac{\partial \mathbf{a}_{p\beta}}{\partial \mathbf{x}_p} w_{\mathbf{i}p}^{n} + \boldsymbol{\tau}_p^S \tilde{\mathbf{a}}_p^{3} : \nabla w_{\mathbf{i}p}^{n} \mathbf{a}_{p3}^{E,n} \tag{5.44}$$

In Equation (5.43), $\mathbf{f}_p^{KL}$ is the generalized Kirchhoff-Love force from Equation (5.32). In Equation (5.44), 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 $\{\mathbf{a}_\alpha(\hat{\mathbf{x}}), \mathbf{a}_3^{E,\mathrm{tr}}(\hat{\mathbf{x}})\}$. We refer to Section B.2 for the calculation.

5.5.2 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).

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} = \sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}}^{n+1} w_{\mathbf{i}p}^n, \ p \in \mathcal{I}^{(i)} \cup \mathcal{I}^{(ii)} \tag{5.45}$$

$$\mathbf{x}_p^{n+1} = \sum_{p^{(ii)} \in \mathcal{I}^{(ii)}} \mathbf{x}_{p^{(ii)}}^{n+1} N_{p^{(ii)}}^{SD}(\xi_{p1}, \xi_{p2}), \ p \in \mathcal{I}^{(iii)}. \tag{5.46}$$

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 (3.6) and (5.39) respectively with $\hat{\mathbf{x}}_{\mathbf{i}} = \mathbf{x}_{\mathbf{i}}^{n+1}$. 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 $\mathbf{x}^{KL}(\hat{\mathbf{x}})$ is interpolated as in Equation (5.40) with $\hat{\mathbf{x}}_{\mathbf{i}} = \mathbf{x}_{\mathbf{i}}^{n+1}$.

5.5.3 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. 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 (5.29).

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 runtime performance for all of our examples in Table 6.3. All simulations were run on an Intel Xeon

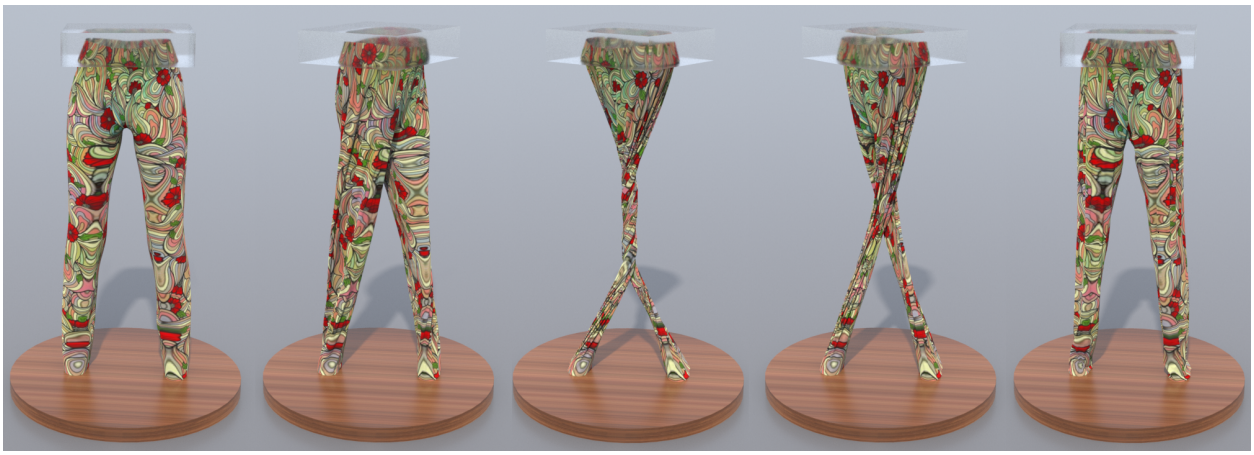


Figure 5.4: **Pants twister**. Our approach works for clothing simulation with many self-collisions as shown here in the legs of a twisted pair of pants. The subdivision mesh for the pants has 393K control points and the simulation runs at 78s per frame.

E5-2687W v4 system with 48 hyperthreads 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 5.7, six cylinders with increasing thickness from left to right free-fall and drop on the ground. In Figure 5.8, four cylinders of decreasing thickness from left to right buckle under lateral pressure and exhibit characteristic buckling patterns. In Figure 5.9, ribbons of varying 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.[10] with parameters fitted from experimental data. In Figure. 5.10a and 5.10b,

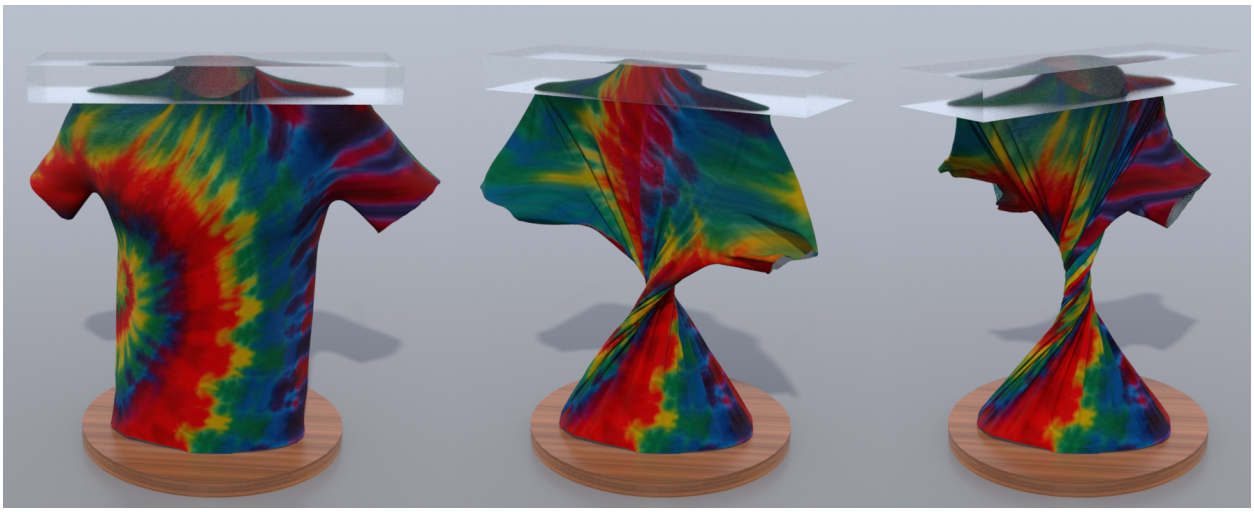


Figure 5.5: **Twisting shirt**. A shirt is undergoing intensive self-contact. Our method successful resolves the twisting and untwisting.

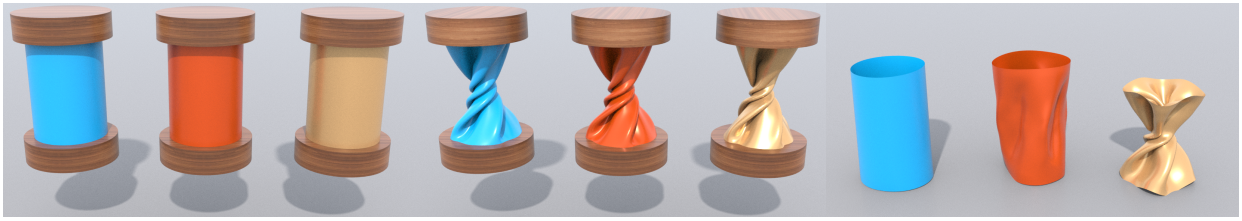


Figure 5.6: **Plastic shell deformation**. The effect of the yield condition in Equation (5.13) is shown here with decreasing values of the coefficient c_{vM} (from left to right). Larger values correspond to a larger stress needed for before denting plasticity is induced. The cylinders are twisted and then dropped to the ground to illustrate the plastic deformation.

we twist and compress sleeves made of denim and silk. In Figure. 5.11a and 5.11b, we suspend squares of silk and denim which then collide with moving spheres. Our model accurately captures the behaviors of these real-world materials.

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 5.12, the spheres and the diving boards, both modeled as shells, collide with each other. In Figure 5.5 and Figure 5.4, we demonstrate self-collisions resolution for clothing simulation stress tests. In Figure 5.13, four decks of cards collide and

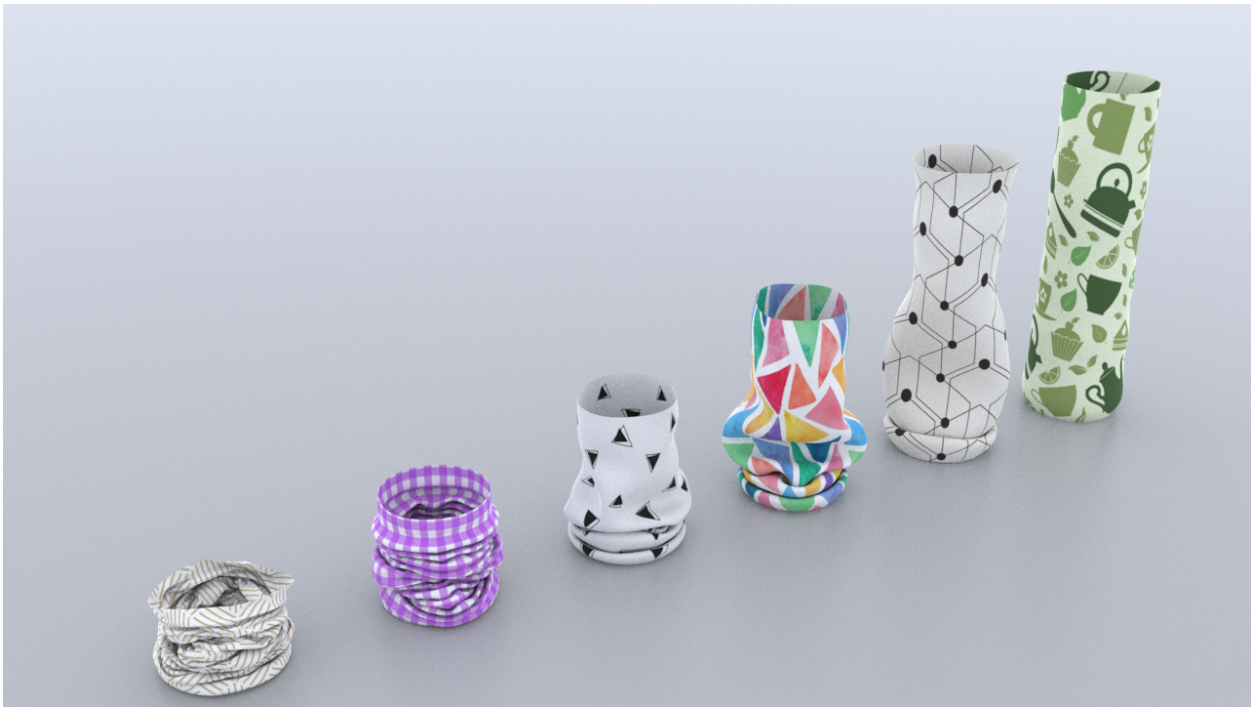


Figure 5.7: **Six cylinders**. Six cylinders with increasing thickness from left to right free-fall and drop on the ground.

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 5.6, 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 5.14, 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 5.15, a cup is filled with slush and then released and toppled. The cup is modeled as a shell and the slush is modeled as in Stomakhin et al.[34]. This example demonstrates that our method successfully resolves the interactions between two different materials of millions of particles

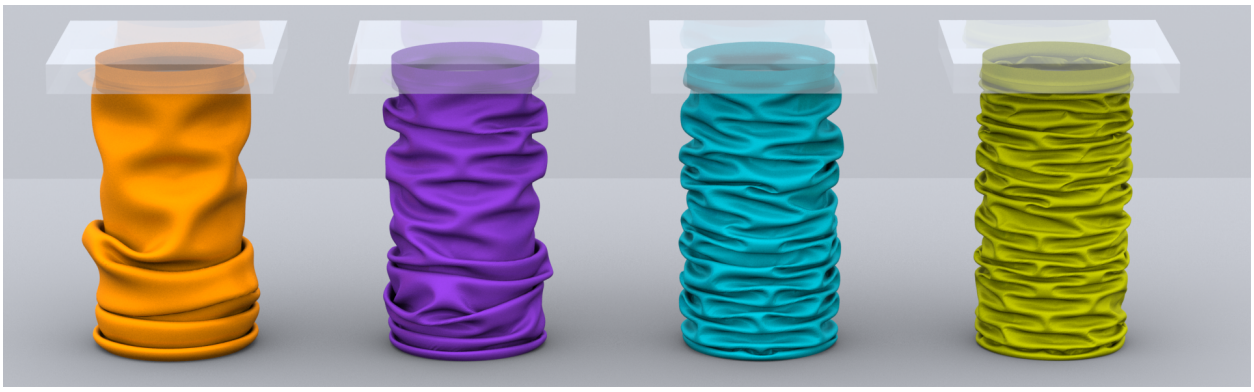


Figure 5.8: **Variation in shell thickness**. We demonstrate the effect of the shell thickness parameter in a compression comparison.

with moderate computation cost.

5.6.6 Resolution Refinement

In Figure 5.16 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.

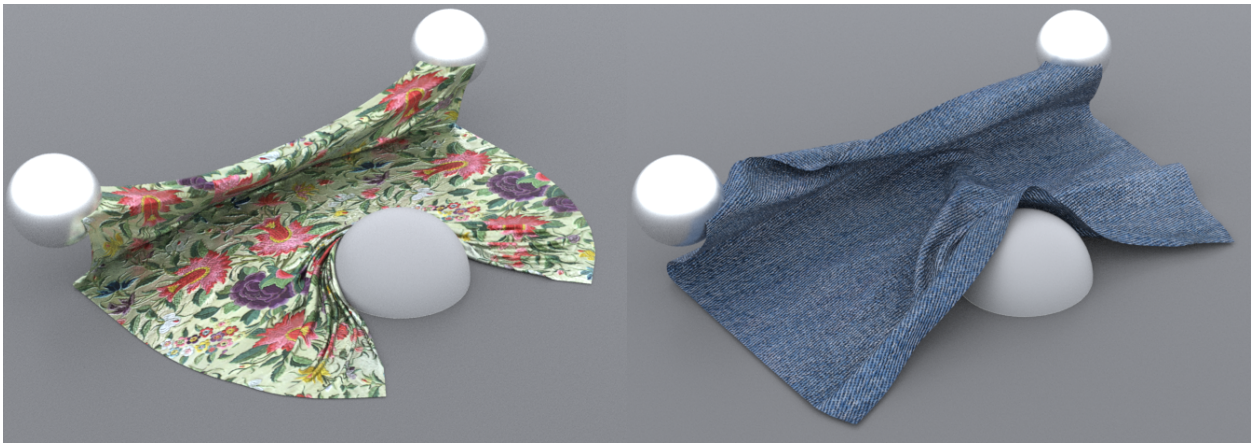
5.6.7 Bending with Jiang et al.

We demonstrate the failure of the Jiang et al. [23] model in achieving significant bending resistance. In Figure 5.17 we compare our model with the Jiang et al. generalized to bending with the addition of bending springs. The frictional contact model in Jiang et al. [23] 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.

Figure 5.9: **Ribbons**. We illustrate interesting dynamics achieved from colliding ribbons with increasing thickness (from left to right).

(a) (b)

Figure 5.10: **Twisting Orthotropic Model**. Using the data-driven model of Clyde et al. [10] 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 5.11: **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. [10] for woven silk (left) and denim (right) materials. The model naturally allows for characteristic buckling and wrinkling behaviors in this object collision test.

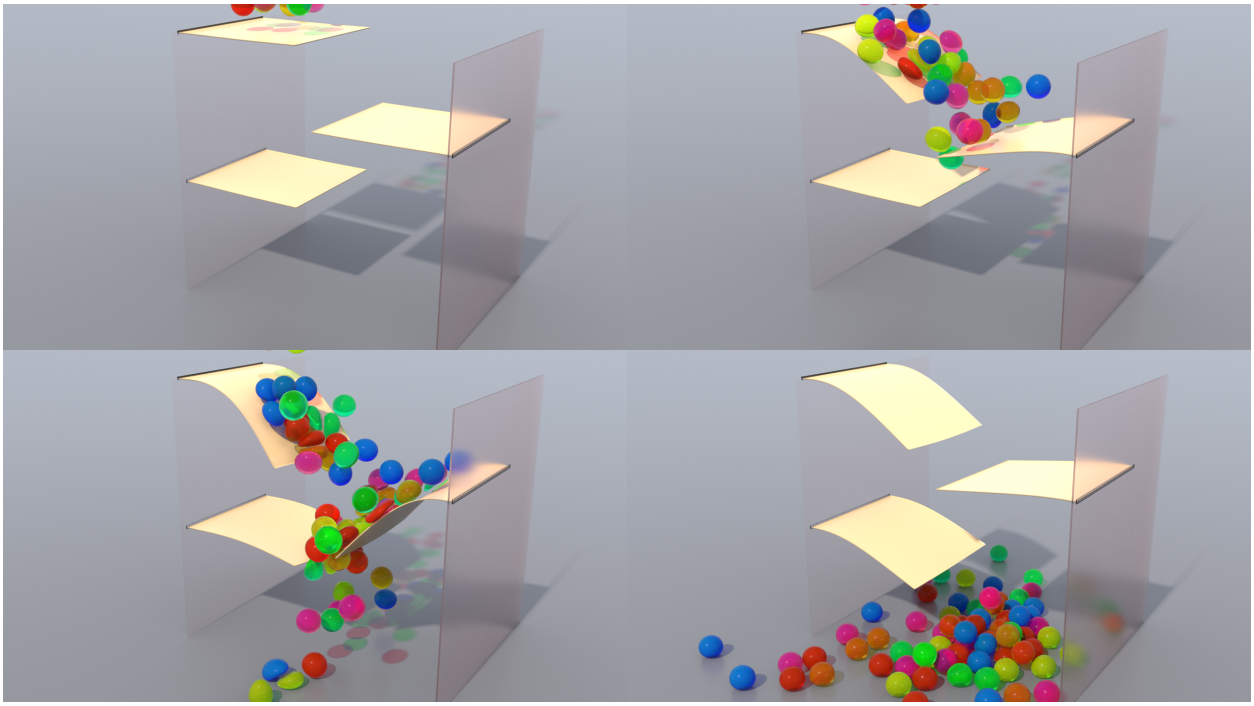


Figure 5.12: **Elastic spheres on diving boards**. We demonstrate appealing dynamics achieved with self-collision and appreciable bending for shells. Both the spheres and the diving boards are simulated as thin shells.

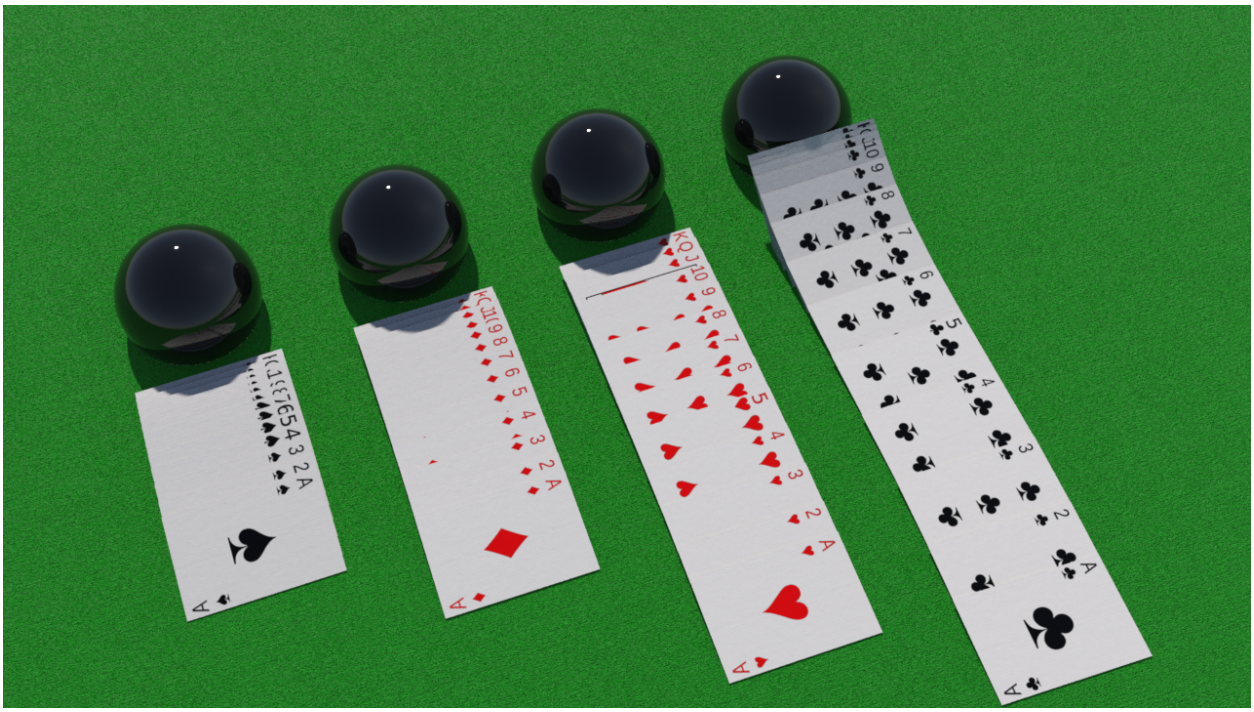


Figure 5.13: **Variation in Coulomb friction coefficient**. The effect of the friction parameter c_F can be seen in this card comparison. By decreasing c_F (from left to right) we demonstrate a range of surface frictions.

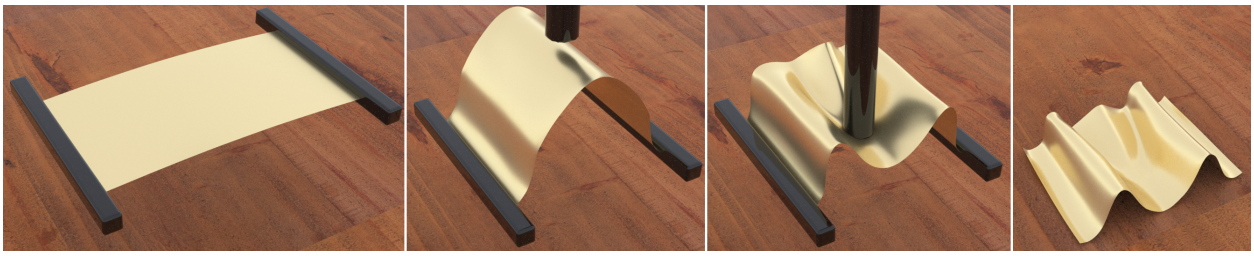


Figure 5.14: **Denting**. We demonstrate plastic deformation of foil induced by object collision.

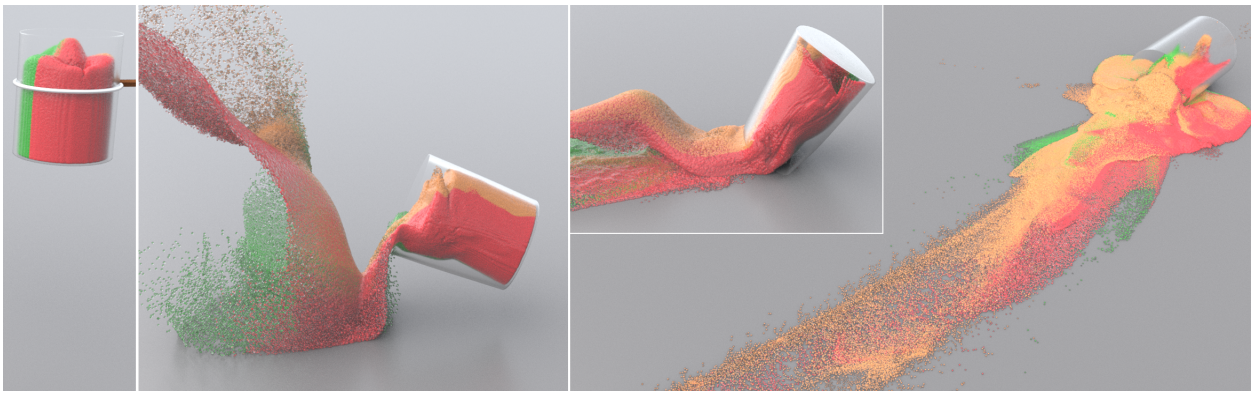


Figure 5.15: **Snow cup**. A cup is filled with slush and then released and toppled. Our method naturally couples with tranditional MPM particles and automatically resolve contact.

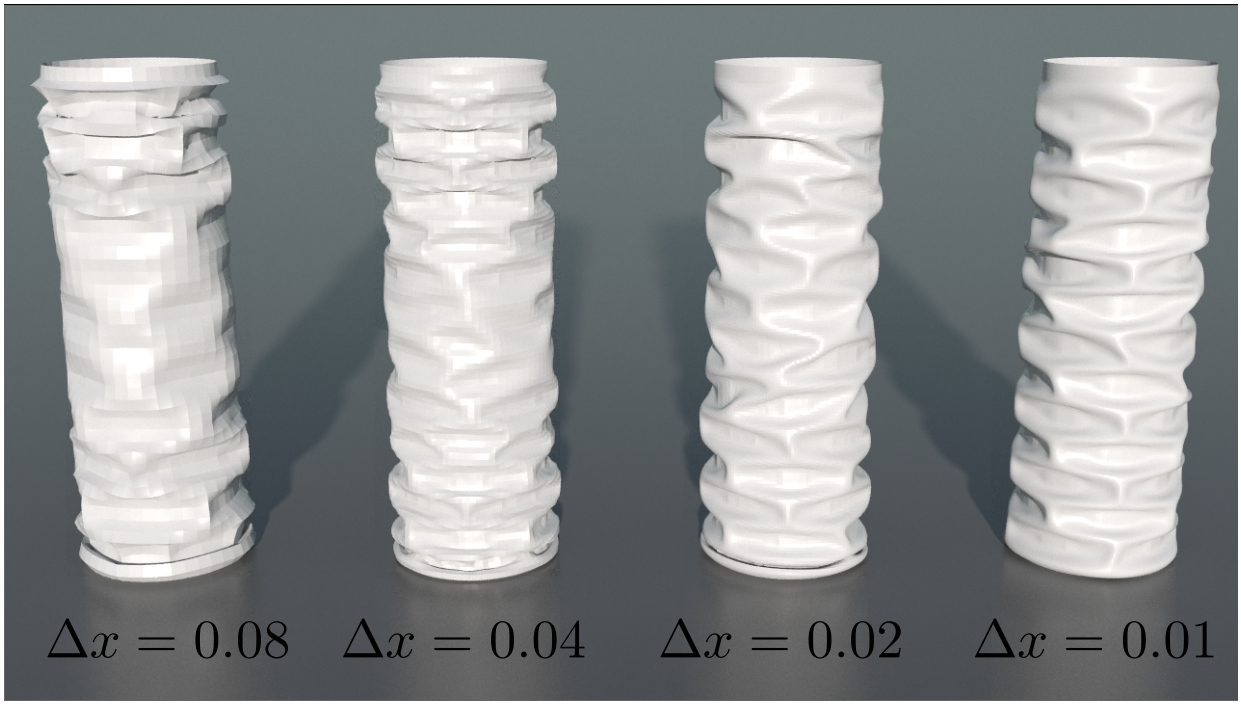


Figure 5.16: **Convergence under spatial refinement**. We demonstrate that our method converges under refinement of grid and subd mesh spatial resolution in this buckling example. The simulations have increasing spatial resolution from left to right.

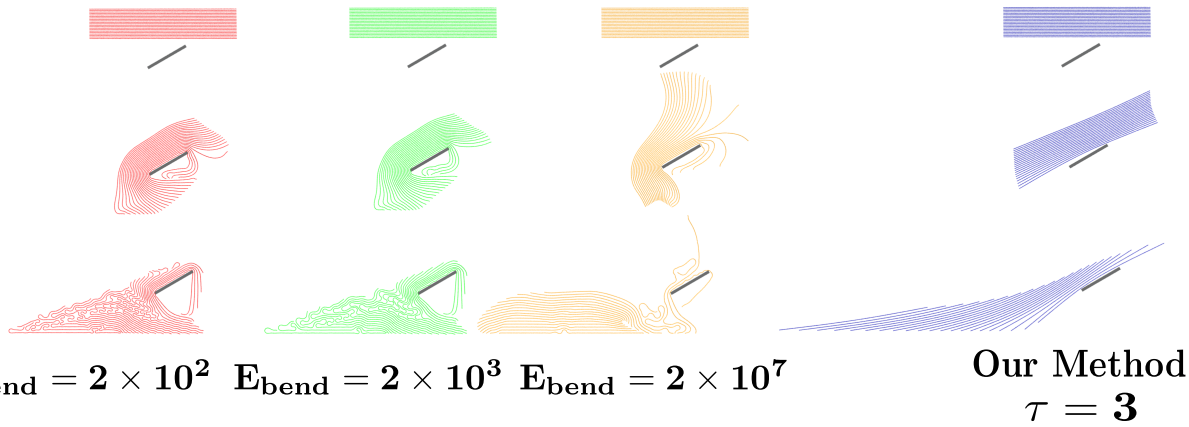


Figure 5.17: **Jiang et al. [23] comparison**. We demonstrate that only moderate bending is possible with the approach of Jiang et al. [23]. Our approach allows for a much wider range of bending resistance.

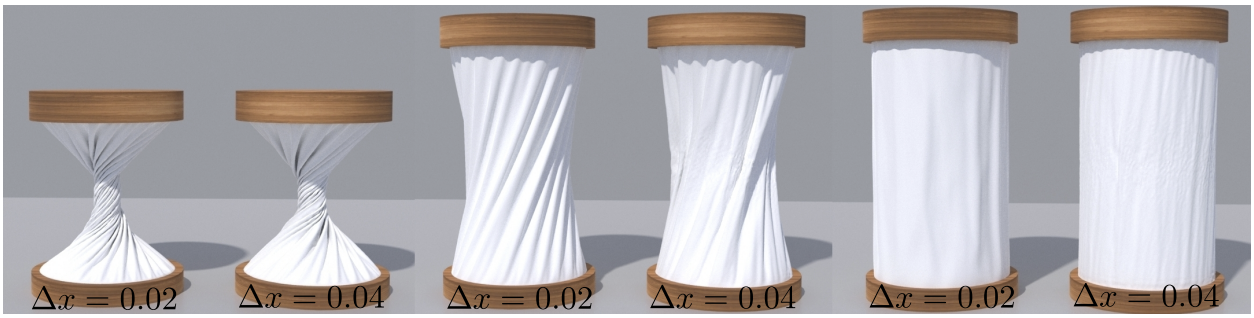


Figure 5.18: **Grid resolution dependent wrinkling.** Our method suffers from persistent wrinkling if the subd mesh resolution is too high relative to the grid resolution. We demonstrate this phenomenon here with a cloth twisting comparison example. In both examples, the subd mesh $\Delta x = 0.02$. The example on the left has grid $\Delta x = 0.02$ whereas the one on the right has grid $\Delta x = 0.04$.

CHAPTER 6

A Polynomial Particle-In-Cell Method

In this chapter, we present a Polynomial Particle-In-Cell (PolyPIC) Method to improve the accuracy of the transfers in Particle-In-Cell (PIC) [19] techniques. Recently the Affine Particle-In-Cell (APIC) Method was proposed by Jiang et al.[24] by augmenting each particle with a locally affine, rather than locally constant description of the velocity. This reduced the dissipation of the original PIC without suffering from the noise present in the historic alternative, Fluid-Implicit-Particle (FLIP) [7]. We present an improvement to APIC that allows for locally polynomial, rather than locally affine approximations to the grid velocity field: PolyPIC.

Our generalization improves kinetic energy conservation during transfers which leads to better vorticity resolution in fluid simulations and less numerical damping in elastoplasticity simulations. Our transfers are designed to select particle-wise polynomial approximations to the grid velocity that are optimal in the local mass-weighted L^2 norm. This is equivalent to the reduced basis APIC derivation in [25] for affine modes, but generalized to polynomial modes. Indeed our notion of transfers reproduces the original PIC if only constants are used and APIC if only affine polynomials are used. Furthermore, we derive a polynomial basis that is mass-orthogonal to facilitate rapid solution of the optimality condition. By design, this reduces the projection to the polynomial basis to the solution of a *diagonal* linear system of size equal to the number of local polynomial modes. This has the added benefit of simplifying applications with staggered grids. We summarize our contributions as:

- A generalization of APIC from locally affine to locally polynomial representations that improves kinetic energy conservation in particle/grid transfers.

- A mass weighted L^2 optimality condition that achieves linear and angular momentum conservation.
- A mass-orthogonal class of polynomials for rapid solution of projection to the polynomial basis.
- Natural treatment of staggered and collocated grids.

We demonstrate the benefits of our technique in a number of representative applications of incompressible flow and MPM simulation of elastoplastic materials.

6.1 Background

PIC [19] and PIC approaches like FLIP [7] for incompressible fluids and the MPM [35, 36] for history dependent materials have proven very effective in graphics applications in recent years. While very powerful, PIC techniques have a number of well known artifacts. Particles store the primary state like mass and momentum, but the effect of internal stress on momentum is added on an Eulerian grid. This is reconciled by transfers back and forth between the particles and grid. There is generally a mismatch in the number of particle and grid degrees of freedom which can lead to errors during the frequent transfers between representations [6]. The original PIC possesses a stabilizing filter property since particle velocities are interpolated from the grid after the stress response. However, this leads to excessive dissipation since particle modes are essentially overwritten by the generally lower resolution grid. FLIP removes this limitation by interpolating increments in velocity rather than velocity itself as in PIC; however this means that particle modes invisible to the grid persist despite not receiving a meaningful constitutive response. This can lead to particle artifacts like noise, instability, clumping and volume loss/gain.

6.1.1 Previous work

Various works have improved or modified aspects of the standard PIC techniques commonly used in graphics. Edwards and Bridson investigated higher-order accuracy [12]. Narrow band

and adaptive particle sampling techniques as well as adaptive/unstructured Eulerian grids dramatically increase efficiency [22, 21, 4, 2, 3, 13]. Ando et al. [1] use a stream function to enforce incompressibility, rather than the more commonly used MAC projection. Mercier et al. [29] increase apparent resolution of FLIP with secondary surface wave simulation. Adaptive shallow water height field and FLIP coupling achieve impressive simulation rates [9]. Smoothed-particle hydrodynamics (SPH) [11] has proven very powerful for graphics applications. Several works couple SPH with PIC techniques to improve aspects like performance, memory usage and discrete incompressibility [27, 21, 26, 15, 43, 30].

There are a number of recent PIC approaches designed to improve robustness to noise without sacrificing accurate energy and momentum conservation. Hammerquist and Nairn [18] developed a PIC extension designed to reduce the noise of the FLIP by adding a smoothing term to the FLIP velocity. This strikes a good balance between noise reduction and energy preservation. Edwards and Bridson also add a regularization term to diminish particle noise [12]. Gritton and Berzins [17] reduce noise by filtering spatial gradients based on a local SVD approximation of the null space of the particle-to-grid transfer operator. Wallstedt and Guilkey use a locally-affine assumption as in [24, 25], but they use FLIP grid-to-particle transfers that still suffer from noise [41]. Um et al. develop a particle repulsion force to improve particle bunching associated with the ringing instability [40].

Recently, Jiang et al. [24, 25] developed an Affine Particle-In-Cell (APIC) approach designed to prevent these artifacts, without incurring the excessive dissipation of PIC. The idea is to retain the filtering property, but to prevent dissipation by interpolating more information from the grid to the particles. By allowing particles to store both velocity and velocity derivative information, Jiang et al. design particle/grid transfers that conserve angular momentum and generally attain the benefits of both PIC and FLIP.

6.2 Notation and method outline

We follow the convention in Section (§3.1). In particular, the Lagrangian state associated with particle p at time t^n consists of mass m_p, position $\mathbf{x}_p^n$, generalized velocity coefficients $\mathbf{c}_p^n$

and auxiliary quantities $\mathbf{A}_p^n$. Both the particle velocity $\mathbf{v}_p^n$ and affine velocity $\mathbf{C}_p^n$ information at time t^n will be included in generalized velocity coefficients $\mathbf{c}_p^n$. The auxiliary quantities in $\mathbf{A}_p^n$ are not relevant to our particle/grid transfers but we include them for completeness. E.g. in an MPM calculation the deformation gradient $\mathbf{F}_p^n$ is auxiliary to transfers and would be included in $\mathbf{A}_p^n$. We will generally consider the update of the auxiliary quantities to be outside the scope of the chapter.

In order to update the Lagrangian state to obtain $\mathbf{x}_p^{n+1}$, $\mathbf{c}_p^{n+1}$ and $\mathbf{A}_p^{n+1}$, we first transfer mass and momentum from particle to grid (Section (§6.5.1)), then grid momentum is dynamically updated (Section (§6.5.2)) and finally, we transfer the generalized velocity information from grid to particle (Section (§6.5.3)). We use the notation $m_\mathbf{i}^n$ and $\mathbf{v}_\mathbf{i}^n$ to denote the mass and velocity transferred to the grid node $\mathbf{x_i}$ from the particles before the grid momentum update. The particle to grid transfer in Section (§6.5.1) and grid to particle transfer in Section (§6.5.3) are improvements to the tranfers described in Section (§3.3) and Section (§3.5) respectively. The grid momentum update in Section (§6.5.2) is generally outside the scope of this chapter, but it is a more general version of the update described in Section (§3.4). This process is illustrated in following commutative diagram.

$$\begin{array}{ccc} m_p,\ \mathbf{x}_p^n,\ \mathbf{c}_p^n,\ \mathbf{A}_p^n & \xrightarrow[\text{Lagrangian State}]{\text{Update}} & m_p,\ \mathbf{x}_p^{n+1},\ \mathbf{c}_p^{n+1},\ \mathbf{A}_p^{n+1} \\ \Big\downarrow \text{P2G} & & \Big\uparrow \text{G2P} \\ m_\mathbf{i}^n, \mathbf{v}_\mathbf{i}^n & \xrightarrow[\text{Grid Momentum}]{\text{Update}} & m_\mathbf{i}^n, \mathbf{v}_\mathbf{i}^{n+1} \end{array}$$

6.3 PIC and APIC Revisit

Grid-based interpolating functions $N(\mathbf{x} - \mathbf{x_i})$ provide the mechanism for the transfer of particle and grid quantities. As in many other recent approaches [33, 34, 24], the grid interpolating functions are constructed from dyadic products of one-dimensional B-splines. We use the notation $w_{\mathbf{i}p}^n = N(\mathbf{x_i} - \mathbf{x}_p^n)$ to denote the weight of interaction between node $\mathbf{x_i}$ and particle $\mathbf{x}_p^n$.

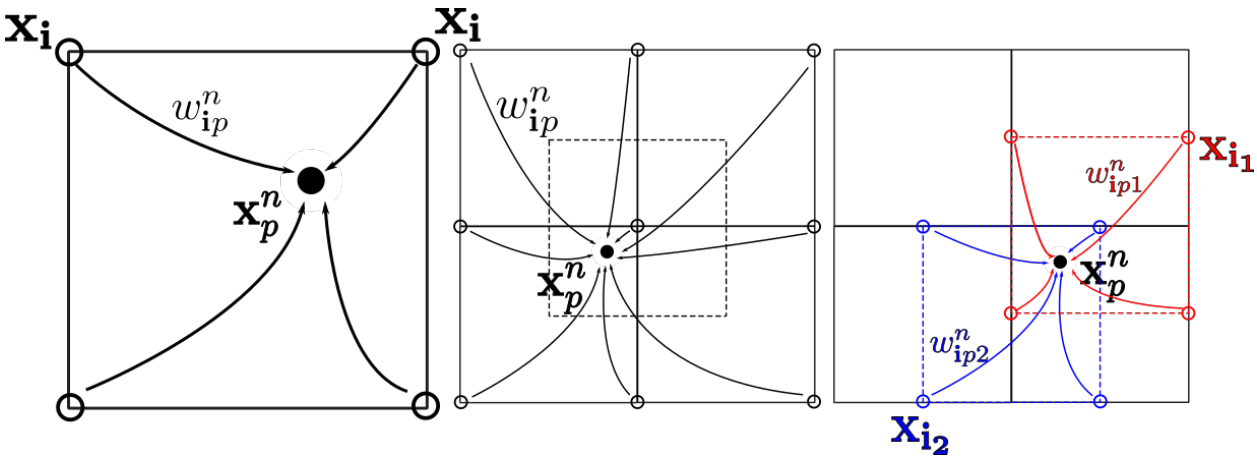


Figure 6.1: **Grid interpolation**. We visualize the weights $w^n_{\mathbf{i}p}$ for multilinear ($N_B = 1$), collocated (left), multiquadratic ($N_B = 2$), collocated (center) and weights $w^n_{\mathbf{i}_\alpha p}$ for linear ($N_B = 1$), MAC grids (right). We emphasize that the particle interpolates from $(N_B + 1)^d$ grid nodes.

Under this interpolation scheme, particle will affect and interpolate from $(N_B + 1)^d$ grid nodes where N_B is the B-spline interpolating order (1 for linear, 2 for quadratic, etc) and $d = 2, 3$ is the spatial dimension. In other words, the particle with position $\mathbf{x}^n_p$ will only have non-zero weights $w^n_{\mathbf{i}p}$ for the $(N_B + 1)^d$ grid nodes most local to it. We will use the notation $\mathcal{V}^n_{p\alpha} \in \mathbb{R}^{(N_B+1)^d}$ to denote the vector of grid-node velocities represented by the particle in the α spatial direction, and $\mathcal{V}^{n+1}_{p\alpha} \in \mathbb{R}^{(N_B+1)^d}$ that of updated grid-node velocities

$$\mathcal{V}^n_{p\alpha} = \begin{pmatrix} v^n_{\mathbf{i}^n_{1p}\alpha} \\ v^n_{\mathbf{i}^n_{2p}\alpha} \\ \vdots \\ v^n_{\mathbf{i}^n_{(N_B+1)^d p}\alpha} \end{pmatrix}, \qquad \mathcal{V}^{n+1}_{p\alpha} = \begin{pmatrix} v^{n+1}_{\mathbf{i}^n_{1p}\alpha} \\ v^{n+1}_{\mathbf{i}^n_{2p}\alpha} \\ \vdots \\ v^{n+1}_{\mathbf{i}^n_{(N_B+1)^d p}\alpha} \end{pmatrix}.$$

We use $\mathbf{i}^n_{kp}$ for $k = 1, 2, \ldots, (N_B + 1)^d$ as an index for nodes with non-zero weights $w^n_{\mathbf{i}^n_{kp}p}$. We illustrate this in Figure 6.1. When it is clear from context, we will use either $\mathbf{i}_k, w^n_{\mathbf{i}_{kp}}$ or even $\mathbf{i}_k, w^n_{\mathbf{i}p}$ in lieu of the more descriptive $\mathbf{i}^n_{kp}, w^n_{\mathbf{i}^n_{kp}p}$ since the sub and super indices can become excessive in some expressions.

In the original PIC particle-to-grid transfer, the contribution of a single particle's mass

and momentum to the grid node $\mathbf{x_i}$ is

$$m^n_{\mathbf{i}p} = w^n_{\mathbf{i}p} m_p$$
$$(m\mathbf{v})^n_{\mathbf{i}p} = w^n_{\mathbf{i}p}(m_p \mathbf{v}^n_p) = m^n_{\mathbf{i}p}\mathbf{v}^n_p,$$

and velocity on the grid node $\mathbf{x_i}$ is

$$\mathbf{v}^n_{\mathbf{i}p} = (m\mathbf{v})^n_{\mathbf{i}p}/m^n_{\mathbf{i}p} = \mathbf{v}^n_p = \sum_{\alpha=1}^{d} \mathbf{e}_\alpha v^n_{p\alpha},$$

where $\mathbf{e}_\alpha \in \mathbb{R}^d$ is the α^{th} standard basis vector. Written in matrix form:

$$\mathcal{V}^n_{p\alpha} = \begin{pmatrix} v^n_{\mathbf{i}_{1p}\alpha} \\ v^n_{\mathbf{i}_{2p}\alpha} \\ \vdots \\ v^n_{\mathbf{i}_{(N_B+1)^d p}\alpha} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} v^n_{p\alpha}.$$

In the APIC particle-to-grid transfer, the momentum transfer from a single particle is

$$(m\mathbf{v})^n_{\mathbf{i}p} = w^n_{\mathbf{i}p} m_p(\mathbf{v}^n_p + \mathbf{C}^n_p(\mathbf{x_i} - \mathbf{x}^n_p)),$$

so that velocity transferred to the grid node is

$$\mathbf{v}^n_{\mathbf{i}p} = (m\mathbf{v})^n_{\mathbf{i}p}/m^n_{\mathbf{i}p} = \mathbf{v}^n_p + \mathbf{C}^n_p(\mathbf{x_i} - \mathbf{x}_p),$$

written in matrix form

$$\mathcal{V}^n_{p\alpha} = \begin{pmatrix} v^n_{\mathbf{i}_{1p}\alpha} \\ v^n_{\mathbf{i}_{2p}\alpha} \\ \vdots \\ v^n_{\mathbf{i}_{(N_B+1)^d p}\alpha} \end{pmatrix} = \Bigg(\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} v^n_{p\alpha} + \sum_{\beta=1}^{d} \begin{pmatrix} x_{\mathbf{i}\beta} - x^n_{p\beta} \\ x_{\mathbf{i}\beta} - x^n_{p\beta} \\ \vdots \\ x_{\mathbf{i}\beta} - x^n_{p\beta} \end{pmatrix} C^n_{p\alpha\beta} \Bigg),$$

where $x_{\mathbf{i}\beta} - x_{p\beta}$ is the β-th spatial component of $\mathbf{x_i} - \mathbf{x}^n_p$.

6.4 Velocity Modes

Inspired by the orignal PIC and APIC transfers, we think to represent the velocity field local to the particle at position $\mathbf{x}_p^n$ at timestep n using

$$\mathbf{v}_p(\mathbf{x}) = \sum_{r=1}^{N_r} \sum_{\alpha=1}^{d} s_r(\mathbf{x} - \mathbf{x}_p^n)\mathbf{e}_\alpha c_{pr\alpha}^n \tag{6.1}$$

where the functions $s_r\mathbf{e}_\alpha : \mathbb{R}^d \to \mathbb{R}^d$ are generalized velocity modes, $\mathbf{e}_\alpha \in \mathbb{R}^d$ is the α^{th} standard basis vector and the $c_{pr\alpha}^n$ are the coefficients of the modes which are stored in the vector $\mathbf{c}_p^n \in \mathbb{R}^{dN_r}$ for each particle. We build our generalized velocity modes component-by-component in terms of the scalar functions $s_r : \mathbb{R}^d \to \mathbb{R}$. N_r indicates the total number of scalar modes that we use. We illustrate these modes in Figure 6.2. Notice that for each component, we use the same modes, only the coefficients are different, i.e., we can think of the α-th dimension of the velocity field local to the particle as a scalar field represented by

$$v_{p\alpha}(\mathbf{x}) = \sum_{r=1}^{N_r} s_r(\mathbf{x} - \mathbf{x}_p^n)c_{pr\alpha}^n. \tag{6.2}$$

This will allow not only efficient computation in the grid-to-particle transfer (Section (§6.5.3)), but also clean derivation for staggered MAC grids (Section (§6.6).

We primarily use polynomial modes of the form

$$s(\mathbf{z}) = \prod_{\beta=1}^{d} z_\beta^{i_\beta}. \tag{6.3}$$

Here z_β is the β^{th} component of $\mathbf{z} \in \mathbb{R}^d$, the $i_\beta \in \mathbb{N}$ are non-negative integer powers. We note that this reduces to the original PIC when $i_\beta = 0$ for $1 \leq \beta \leq d$. Furthermore, when we choose all s_r with exactly one of the $i_\beta = 1$ and the rest equal to zero, we obtain the affine modes and the method reduces to APIC. In general, we will modify the polynomial modes in Equation (6.3) slightly to ensure a mass-orthogonality condition that is essential for efficiency in the grid to particle transfer (see Section (§6.5.3)).

By approximating the velocity local to particle $\mathbf{x}_p^n$ in terms of more general functions, we allow for a wider range of local behavior than in the original APIC. Notably, we can write APIC in this way by choosing affine functions for s_r. Similarly, we can write PIC in this way by choosing constant functions for the s_r. In either case we note that the coefficients $\mathbf{c}_p^n \in \mathbb{R}^{dN_r}$ are equivalent to the $\mathbf{v}_p^n$ and $\mathbf{C}_p^n$ in the original APIC and PIC. Note that for APIC, $dN_r = d + d^2$ (d translations and d^2 linear functions) and similarly for PIC, $dN_r = d$.

The particle-wise local velocity in Equation (6.1) is used in the particle-to-grid and grid-to-particle transfers. As in [24, 25], it is used to define a particle's contribution to the grid node linear momentum in the particle-to-grid transfer (Section (§6.5.1)). In the grid-to-particle transfer (Section (§6.5.3)), the coefficients $\mathbf{c}_p^{n+1}$ are chosen so that the local velocity field represented by the particle at $\mathbf{x}_p^n$ approximates that directly interpolated from the grid, i.e.,

$$\mathbf{v}_p^{n+1}(\mathbf{y}) = \sum_{r=1}^{N_r} \sum_{\alpha=1}^{d} s_r(\mathbf{y} - \mathbf{x}_p^n)\mathbf{e}_\alpha c_{pr\alpha}^{n+1} \approx \sum_{\mathbf{i}} \mathbf{v}_{\mathbf{i}}^{n+1} N(\mathbf{y} - \mathbf{x}_{\mathbf{i}}) \tag{6.4}$$

for $\mathbf{y}$ near $\mathbf{x}_p^n$.

6.5 Method

6.5.1 Transfer from Particle to Grid

The velocity local to the particle $\mathbf{v}_p^n : \mathbb{R}^d \to \mathbb{R}^d$ from Equation (6.1) is used to design the momentum transfer to the grid. A single particle's contribution to the momentum local to the node $\mathbf{x}_\mathbf{i}$ is $(m\mathbf{v})_{\mathbf{i}p}^n = m_p w_{\mathbf{i}p}^n \mathbf{v}_p^n$ and $(m\mathbf{v})_\mathbf{i}^n = \sum_p (m\mathbf{v})_{\mathbf{i}p}^n$ is the total momentum of grid node from the contribution of all particles. Similarly, the contribution of the particle's mass to the grid node $\mathbf{x}_\mathbf{i}$ is $m_{\mathbf{i}p}^n = w_{\mathbf{i}p}^n m_p$ and the total grid node mass is the sum of the contributions from all particles $m_\mathbf{i}^n = \sum_p m_{\mathbf{i}p}^n$. Using this we can define the grid node velocity

$\mathbf{v}_{\mathbf{i}}^n$ by dividing momentum by mass. In summary, this transfer consists of

$$\begin{aligned} m_{\mathbf{i}p}^n &= w_{\mathbf{i}p}^n m_p, \;\; m_{\mathbf{i}}^n = \sum_p m_{\mathbf{i}p}^n \\ (m\mathbf{v})_{\mathbf{i}p}^n &= m_{\mathbf{i}p}^n \sum_{r=1}^{N_r} \sum_{\alpha=1}^{d} s_r(\mathbf{x}_{\mathbf{i}} - \mathbf{x}_p^n)\mathbf{e}_\alpha c_{pr\alpha}^n \\ (m\mathbf{v})_{\mathbf{i}}^n &= \sum_p (m\mathbf{v})_{\mathbf{i}p}^n, \\ \mathbf{v}_{\mathbf{i}}^n &= \frac{(m\mathbf{v})_{\mathbf{i}}^n}{m_{\mathbf{i}}^n}. \end{aligned} \tag{6.5}$$

We note that this is essentially the same transfer as in the original APIC approaches [24, 25], with the only modification being the more general notions of the local velocity.

It is useful to rewrite Equation (6.5) in matrix notation. Similar to the vector of grid-node velocities $\mathcal{V}_{p\alpha}^n \in \mathbb{R}^{(N_B+1)^d}$, we use the notation $\mathbf{Q}_p^n = \left[\mathcal{Q}_{p1}^n, \mathcal{Q}_{p2}^n, \ldots, \mathcal{Q}_{pN_r}^n\right] \in \mathbb{R}^{(N_B+1)^d \times N_r}$ where the columns $\mathcal{Q}_{pr}^n$ of $\mathbf{Q}_p^n$ are analogous to $\mathcal{V}_{p\alpha}^n$ and have entries equal to the particle-wise local modes $s_r(\mathbf{x}_{\mathbf{i}} - \mathbf{x}_p^n)\mathbf{e}_\alpha$ at the grid nodes with non-zero weights

$$\mathcal{Q}_{pr\alpha}^n = \begin{pmatrix} s_r(\mathbf{x}_{\mathbf{i}_1} - \mathbf{x}_p^n) \\ s_r(\mathbf{x}_{\mathbf{i}_2} - \mathbf{x}_p^n) \\ \vdots \\ s_r(\mathbf{x}_{\mathbf{i}_{(N_B+1)^d}} - \mathbf{x}_p^n) \end{pmatrix}.$$

In matrix notation, the momentum transfer in Equation (6.5) writes

$$\mathbf{M}_p^n \mathcal{V}_{p\alpha}^n = \mathbf{M}_p^n \mathbf{Q}_p^n \mathbf{c}_{p\alpha}^n, \tag{6.6}$$

where the matrix $\mathbf{M}_p^n \in \mathbb{R}^{(N_B+1)^d \times (N_B+1)^d}$ is diagonal with entries equal to $m_{\mathbf{i}p}^n$. We use the same matrices $\mathbf{M}_p^n$ and $\mathbf{Q}_p^n$ for all α directions.

6.5.2 Update Grid Momentum

The grid momentum update is outside the scope of this chapter. However, we include a generic description for representative cases that we used to generate our examples: incompressible Euler fluids and elastoplastic solids with MPM. In the case of the incompressible Euler, we used a MAC grid discretization of the pressure projection to update the fluid velocity. In the case of elastoplastic solids and MPM the update is from the elastic force (see [14] for more details).

$$\mathbf{v}_{\mathbf{i}}^{n+1} = \mathbf{v}_{\mathbf{i}}^{n} + \frac{\Delta t}{\rho}\nabla p, \qquad \text{(Euler/MAC)}$$

$$\mathbf{v}_{\mathbf{i}}^{n+1} = \mathbf{v}_{\mathbf{i}}^{n} + \frac{\Delta t}{m_{\mathbf{i}}^{n}}(\mathbf{f} + \mathbf{g}), \qquad \text{(elastoplastic/MPM)}$$

where $\mathbf{f}$ is the elastic force and $\mathbf{g}$ is the gravitational acceleration.

6.5.3 Transfer from Grid to Particle

The transfer from grid to particle is achieved by choosing the generalized velocity coefficients $\mathbf{c}_p^{n+1} \in \mathbb{R}^{dN_r}$ so that the approximation in Equation (6.4) is optimal in the appropriate sense. Here we show that we can solve a linear system for the coefficients $\mathbf{c}_p^{n+1} \in \mathbb{R}^{dN_r}$, and that by design our approach

- is equivalent to PIC and APIC if only constant or affine modes are used,
- conserves linear and angular momentum (see [14]) and
- has a *diagonal* system matrix in the equation for the $\mathbf{c}_p^{n+1} \in \mathbb{R}^{dN_r}$.

We choose the coefficients $\mathbf{c}_p^{n+1}$ to minimize the mass-weighted distance $d_p^n(\mathbf{c}_p^{n+1})$ between local velocities at the grid nodes and the updated grid-node velocities

$$
\begin{aligned}
d_p^n(\mathbf{c}_p^{n+1}) &= \sum_{\mathbf{i}} m_{\mathbf{i}p}^n \left|\mathbf{v}_{\mathbf{i}}^{n+1} - \mathbf{v}_p^{n+1}(\mathbf{x}_{\mathbf{i}})\right|^2 \\
&= \sum_{\mathbf{i}} m_{\mathbf{i}p}^n \left|\mathbf{v}_{\mathbf{i}}^{n+1} - \sum_{r=1}^{N_r}\sum_{\alpha=1}^{d} s_r(\mathbf{x}_{\mathbf{i}} - \mathbf{x}_p^n)\mathbf{e}_\alpha c_{pr\alpha}^{n+1}\right|^2 \\
&= \sum_{\alpha=1}^{d} \left(\mathcal{V}_{p\alpha}^{n+1} - \mathbf{Q}_p^n \mathbf{c}_{p\alpha}^{n+1}\right)^T \mathbf{M}_p^n \left(\mathcal{V}_{p\alpha}^{n+1} - \mathbf{Q}_p^n \mathbf{c}_{p\alpha}^{n+1}\right)
\end{aligned}
$$

where $m_{\mathbf{i}p}^n = m_p w_{\mathbf{i}p}^n$ as defined in Equation (6.5), the second equation comes from Equation (6.1), and the last equation is just a rewrite in terms of matrix notation. The minimizer of this mass weighted distance can be expressed more concisely in terms of the grid node locations that received non-zero mass from the particle $\mathbf{x}_p^n$. The optimal coefficients $\mathbf{c}_p^{n+1}$ can be expressed in terms of these vectors as

$$
\begin{aligned}
\mathbf{c}_p^{n+1} &= \underset{\mathbf{c} \in \mathbb{R}^{dN_r}}{\operatorname{argmin}} \; d_p^n(\mathbf{c}) \\
\mathbf{c}_{p\alpha}^{n+1} &= \left(\mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathbf{Q}_p^n\right)^{-1} \mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathcal{V}_{p\alpha}^{n+1}
\end{aligned}
\tag{6.7}
$$

for each α dimension.

6.5.3.1 Mass-orthogonal polynomial modes

Our approach is only efficient if the linear system for $\mathbf{c}_p^{n+1}$ in Equation (6.7) can be solved quickly. Fortunately, the individual blocks $\mathbf{Q}_p^{nT}\mathbf{M}_p^n\mathbf{Q}_p^n \in \mathbb{R}^{N_r \times N_r}$ have remarkable sparsity structure. If we assume that we number the modes with increasing degree (e.g. in 2D, constant modes first: $s_1 = 1$, followed by linear $s_2 = x, s_3 = y$, then multilinear: $s_4 = xy$, etc) and if we use modes s_r with $r \leq N_r \leq 2^d$, the matrix $\mathbf{Q}_p^{nT}\mathbf{M}_p^n\mathbf{Q}_p^n$ is diagonal. This can be verified directly using Mathematica and we provide Mathematica code in Section C.5. Notably, this means that constant modes ($r \leq 1$), linear modes ($1 < r \leq d$) and multilinear modes ($d < r \leq 2^d$) are mass-orthogonal and therefore the coefficients in Equation (6.7) can be obtained through the solution of a diagonal system.

In general for $2^d < r \leq N_r \leq (N_B+1)^d$, the matrix $\mathbf{Q}_p^{nT}\mathbf{M}_p^n\mathbf{Q}_p^n$ is not diagonal. We illustrate this in Table 6.1 with $d=2$ for brevity. However, we can obtain a diagonal system with a modified Gram-Schmidt approach that takes into account the inner product defined by $\mathbf{M}_p^n$. This amounts to simple modifications of the quadratic scalar modes $2^d < r \leq N_r \leq (N_B+1)^d$ in Equation (6.3). Remarkably, the Gram-Schmidt mass-orthogonalization does not modify any of the constant, linear or multilinear modes. Only the quadratic modes are modified and the change is very simple: each quadratic term z_β^2 in Equation (6.3) is replaced with $g_\beta(z_\beta)$ given by

$$g_\beta(w) = w^2 - \frac{x_{p\beta}^n\left(\Delta x^2 - 4(x_{p\beta}^n)^2\right)}{\Delta x^2}w - \frac{\Delta x^2}{4}. \tag{6.8}$$

E.g. the mode $s_5 = g_1(x)$ replaces x^2, $s_6 = g_2(y)$ replaces y^2, $s_7 = g_1(x)y$ replaces x^2y, etc. This trivial modification yields a diagonal $\mathbf{Q}_p^{nT}\mathbf{M}_p^n\mathbf{Q}_p^n$ whose entries we enumerate in Table 6.2. We give expressions for the individual entries in the solution $\mathbf{c}_p^{n+1}$ to Equation (6.7) with diagonal basis in Section C.1.

We note that $(N_B+1)^d$ is a natural upper bound on the number of reduces modes N_r since the minimization in Equation (6.7) is over determined for $N_r > (N_B+1)^d$.

s_r \ s_t	1	x	y	xy	x^2	y^2	x^2y	xy^2	x^2y^2
1	X				X	X			X
x		X			X			X	X
y			X			X	X		X
xy				X			X	X	X
x^2	X	X			X	X		X	X
y^2	X		X		X	X	X		X
x^2y			X	X		X	X	X	X
xy^2		X		X	X		X	X	X
x^2y^2	X	X	X	X	X	X	X	X	X

Table 6.1: **Sparsity pattern: unmodified.** We illustrate the sparsity pattern of the matrix $(\mathbf{S}_p^n)^T\mathbf{m}_p^n\mathbf{S}_p^n$ for dimension $d=2$ with scalar modes $s = x^{i_1}y^{i_2}$. X indicates a non-zero entry in the matrix. Note that the multilinear modes (indicated in red) are mass-orthogonal to one another, but that the multiquadratic modes couple extensively.

s_r \ s_t	1	x	y	xy	$g_1(x)$	$g_2(y)$	$g_1(x)y$	$xg_2(y)$	$g_1(x)g_2(y)$
1	a								
x		b							
y			b						
xy				c					
$g_1(x)$					$d(x)$				
$g_2(y)$						$d(y)$			
$g_1(x)y$							$e(x)$		
$xg_2(y)$								$e(y)$	
$g_1(x)g_2(y)$									$f(x,y)$

Table 6.2: **Sparsity pattern: modified.** We illustrate the sparsity pattern of the matrix $(\mathbf{S}_p^n)^T \mathbf{m}_p^n \mathbf{S}_p^n$ for dimension $d = 2$ with the modified quadratic modes given by Equation (6.8). $a = 1$, $b = \frac{\Delta x^2}{4}$, $c = \frac{\Delta x^2}{16}$, $d(z) = \frac{(\Delta x^2 - 4z^2)^2(3\Delta x^2 - 4z^2)}{16\Delta x^2}$, $e(z) = \frac{(\Delta x^2 - 4z^2)^2(3\Delta x^2 - 4z^2)}{64}$, $f(x,y) = \frac{(\Delta x^2 - 4x^2)^2(3\Delta x^2 - 4x^2)(\Delta x^2 - 4y^2)^2(3\Delta x^2 - 4y^2)}{256\Delta x^4}$.

6.6 MAC grids

For clarity of exposition, we only consider the case of collocated grids in Sections (§6.5.1) and (§6.5.3). For incompressible Euler we transfer to and from staggered velocity MAC grids [20]. Using $\mathbf{i}_\alpha$, $1 \leq \alpha \leq d$ to denote the face index for each of the staggered grids, MAC transfers are done component-wise (see Figure 6.1). Particle $\mathbf{x}_p^n$ transfers mass $m_{\mathbf{i}_\alpha p}^n$ to each α face grid from $m_{\mathbf{i}_\alpha p}^n = m_p^n w_{\mathbf{i}_\alpha p}^n$. The total mass on each grid face $m_{\mathbf{i}_\alpha}^n$ is equal to the sum of the contribution from each particle $m_{\mathbf{i}_\alpha}^n = \sum_p m_{\mathbf{i}_\alpha p}^n$. The weight of interaction $w_{\mathbf{i}_\alpha p}^n = N(\mathbf{x}_{\mathbf{i}_\alpha} - \mathbf{x}_p^n)$ is between the particle $\mathbf{x}_p^n$ and the MAC face $\mathbf{x}_{\mathbf{i}_\alpha}$. The component-wise particle-to-grid momentum transfer is

$$\begin{aligned} (mv)_{\mathbf{i}_\alpha p}^n &= m_{\mathbf{i}_\alpha p}^n \sum_r s_r(\mathbf{x}_{\mathbf{i}_\alpha} - \mathbf{x}_p^n) c_{pr\alpha}^n \\ (mv)_{\mathbf{i}_\alpha}^n &= \sum_p (mv)_{\mathbf{i}_\alpha p}^n, \;\; v_{\mathbf{i}_\alpha}^n = \frac{(mv)_{\mathbf{i}_\alpha}^n}{m_{\mathbf{i}_\alpha}^n}. \end{aligned} \tag{6.9}$$

These transfers are very similar to those in Equation (6.5). However, unlike in the collocated case, the mass matrix and scalar mode vectors will be different on each of the velocity face grids. We use the notation $\mathbf{M}_{p\alpha}^n$ and $\mathbf{Q}_{pr\alpha}^n$ to denote this, where the appearance of α emphasizes that they vary with each face grid. With this convention we can write the system for the reduced mode components $\mathbf{c}_p^{n+1} \in \mathbb{R}^{dN_r}$ as

$$\mathbf{c}_{p\alpha}^{n+1} = \left(\mathbf{Q}_{p\alpha}^{n\;T} \mathbf{M}_{p\alpha}^n \mathbf{Q}_{p\alpha}^n \right)^{-1} \mathbf{Q}_{p\alpha}^{n\;T} \mathbf{M}_{p\alpha}^n \mathcal{V}_{p\alpha}^{n+1} \tag{6.10}$$

for $1 \leq r \leq N_r$.

6.7 Simulation Results

We demonstrate our method on a number of examples with incompressible flow and MPM elastoplasticity. We compare PolyPIC with APIC and FLIP in a number of representative scenarios. All incompressible flow simulations were done using Manta Flow [39]. In a few of our incompressible examples, we use passive advected particles as a post-process to aid in visualization. We note that these are simply advected in the flow for post-process visualization and do not use PolyPIC transfers. Also, all grid interpolation is multilinear for the incompressible flow examples. All grid interpolation is multiquadratic for the MPM elastoplasticity examples.

6.7.1 Incompressible Flow

In Figure 6.3 we simulate a vortex sheet by setting the velocity inside a circle to be initially rotating relative to a stationary surrounding fluid. The discontinuity in the velocity induces vorticity at the interface which produces intricate flow patterns. We compare PolyPIC to FLIP and APIC and see that it better resolves the vorticial flow.

In Figure 6.4 we simulate an ink droplet in an ambient incompressible fluid by dropping liquid onto a free surface. We only render the particles in the jet. Note that the ink and water are both simulated as the same incompressible fluid. We compare PolyPIC to FLIP and APIC and see that it again better captures the transition to turbulence. We note that PolyPIC works well even when the grid resolution is rather low. Figure 6.4 was run with a relatively low grid resolution $64 \times 256 \times 64$. We used 8 simulated particles per cell, and 8000 passively advected tracer particles per cell in a post-process for visualization.

Figure 6.5 demonstrates a 3D version of the vortex sheet. The cylinder is initially rotating about its axis relative to a stationary ambient fluid. It was also run on a low resolution grid ($88 \times 132 \times 88$) with 8 simulated particles per cell for simulation and 216 passively ad-

vected tracer particles per cell in a post-process for visualization. Despite the low resolution simulation, intricate flow patterns are observed.

For all incompressible flow examples we use constant, linear and multilinear modes (i.e. $N_r = 2^d$) with PolyPIC. This is the maximum number of modes we can use because the grid interpolation in the incompressible flow solver is multilinear ($N_B = 1$) and, as discussed in Section (§6.5.3.1), the number of reduced modes is bounded by $N_r \leq (N_B + 1)^d$.

6.7.2 MPM elastoplasticity

In Figure 6.6 we demonstrated the increasingly energetic nature of PolyPIC elastoplasticity simulations as we add more polynomial modes. Note that with $N_r = 6$ modes the sand flows more freely and splashes off the jello more dramatically, while the Jell-O bounces more readily.

In Figure 6.8 we demonstrate the improved energy conservation of our method over APIC. In this scenario, a 2D hyperelastic square is initially compressed. The total energy of the system should be conserved with these initial and boundary conditions (zero traction). As we add more polynomial modes, the energy preservation improves. In Figure 6.9, we demonstrate how the increased energy retention affects the dynamics of a Jell-O cube dropped on the ground.

6.7.3 Accuracy and the number of modes

We verify that adding additional modes increases the accuracy of the simulation. In 6.6 and 6.7 we examine the case of granular sand flowing from a container onto Jell-0. In Figure 6.6 we see that PolyPIC with $N_r = 4$ and APIC are less energetic than PolyPIC with $N_r = 6$. The flow of the sand in the container suffers from more numerical friction with PolyPIC $N_r = 4$ and APIC, therefore sand flows out of the container much slower. We can see this because the containers are still quite full in the final frame with PolyPIC $N_r = 4$ and APIC compared to PolyPIC with $N_r = 6$. In Figure 6.7 we rerun the same simulations but with higher grid and particle resolution. At this resolution, the PolyPIC $N_r = 4$ and

APIC containers are all nearly empty in the final frame and as a result all flows are similarly energetic, indicating that PolyPIC with more modes gives a more accurate result since it is more predictive of the refined behavior.

6.7.4 Momentum conservation

We verify the angular momentum conservation properties of the PolyPIC transfers. In Figure 6.10 we plot the linear and angular momentum over the course of the time step for the falling Jell-O example shown in Figure 6.9. Even though the PolyPIC transfers conserve the momenta, the grid momentum update and the application of boundary conditions (Section (§6.5.2)) are not momentum conserving. To illustrate the conservation of the momentum in the transfers, we can monitor

$$\hat{\mathbf{l}}^n = \mathbf{l}^n_{P2G} + \sum_{m=1}^{n-1} \mathbf{l}^m_{\text{grid}} - \mathbf{l}^m_{P2G}, \;\; \hat{\mathbf{p}}^n = \mathbf{p}^n_{P2G} + \sum_{m=1}^{n-1} \mathbf{p}^m_{\text{grid}} - \mathbf{p}^m_{P2G} \tag{6.11}$$

where $\mathbf{l}^n = \sum_{\mathbf{i}} \mathbf{x}_{\mathbf{i}} \times m_{\mathbf{i}} \mathbf{v}^n_{\mathbf{i}}$ and $\mathbf{p}^n = \sum_{\mathbf{i}} m_{\mathbf{i}} \mathbf{v}^n_{\mathbf{i}}$ are the angular and linear momenta on the grid. $\mathbf{l}^m_{P2G}$ $\mathbf{l}^m_{P2G}$ are computed after the transfer from particle to grid (Section (§6.5.1)) and $\mathbf{l}^m_{\text{grid}}$ and $\mathbf{p}^m_{\text{grid}}$ are computed after the grid momentum update (Section (§6.5.2)). The quantities $\mathbf{l}^m_{\text{grid}} - \mathbf{l}^m_{P2G}$ and $\mathbf{p}^m_{\text{grid}} - \mathbf{p}^m_{P2G}$ are the momenta lost during the grid momentum update at time step t^m. This is the only source of angular momentum loss for APIC and PolyPIC and thus the quantities in Equation (6.11) should be constant for those methods. We visualize the angular momentum loss from transfers in Figure 6.10. The straight lines indicate conservation.

	Seconds/Frame	$\Delta t_{\max}$	Particles	Cores
Ink Drop(FLIP99)	20.569	5×10^{-2}	3.64M	16
Ink Drop(APIC)	23.188	5×10^{-2}	3.64M	16
Ink Drop(PolyPIC)	31.466	5×10^{-2}	3.64M	16
Cylinder(PolyPIC)	146.744	2×10^{-1}	7.86M	12
Vortex Sheet(FLIP99)	2.367	1×10^{-1}	0.97M	20
Vortex Sheet(APIC)	2.739	1×10^{-1}	0.97M	12
Vortex Sheet(PolyPIC)	2.760	1×10^{-1}	0.97M	20
Sand & Jello(APIC)	11.582	4×10^{-5}	59.7K	12
Sand & Jello(PolyPIC4)	12.616	4×10^{-5}	59.7K	12
Sand & Jello(PolyPIC6)	17.682	4×10^{-5}	59.7K	12
Jello(APIC)	4.882	2×10^{-4}	17.5K	48
Jello(PolyPIC8)	5.713	2×10^{-4}	17.5K	48
Jello(PolyPIC11)	5.562	2×10^{-4}	17.5K	48
Jello(PolyPIC14)	5.512	2×10^{-4}	17.5K	48
Jello(PolyPIC18)	5.852	2×10^{-4}	17.5K	48

Table 6.3: We list the time step sizes, run times, particle counts and number of cores used for our simulations. We note that the Jell-O examples demonstrate that increasing the number of reduced modes N_r in PolyPIC only moderately increases the computational cost over APIC.

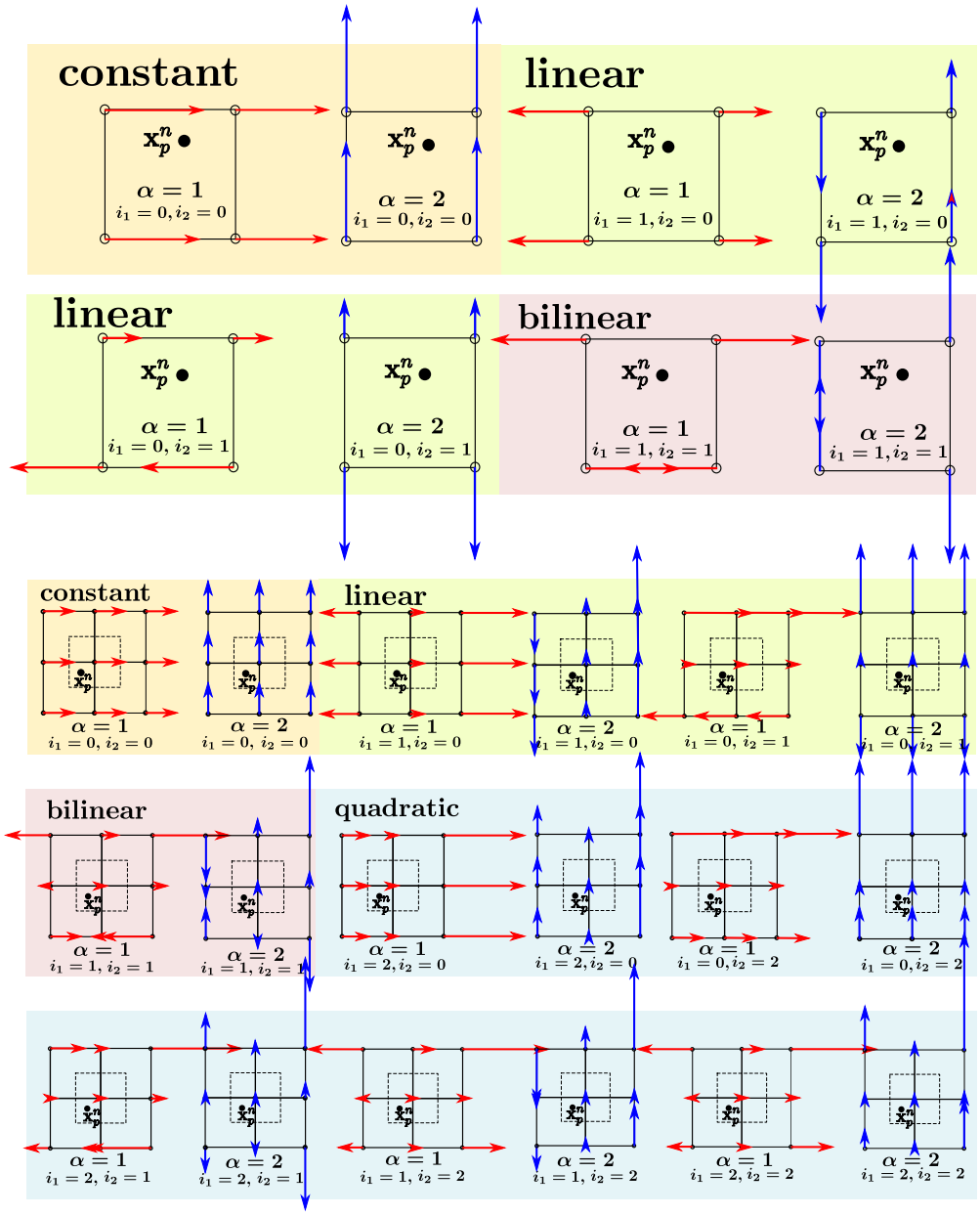


Figure 6.2: **Velocity modes.** We visualize the component-wise velocity modes from Equation (6.3) in 2D. The top shows bilinear interpolation and the bottom shows biquadratic interpolation. Constant (peach), linear (green), bilinear (pink) and biquadratic (light blue) modes are depicted for x (red) and y (blue) components.

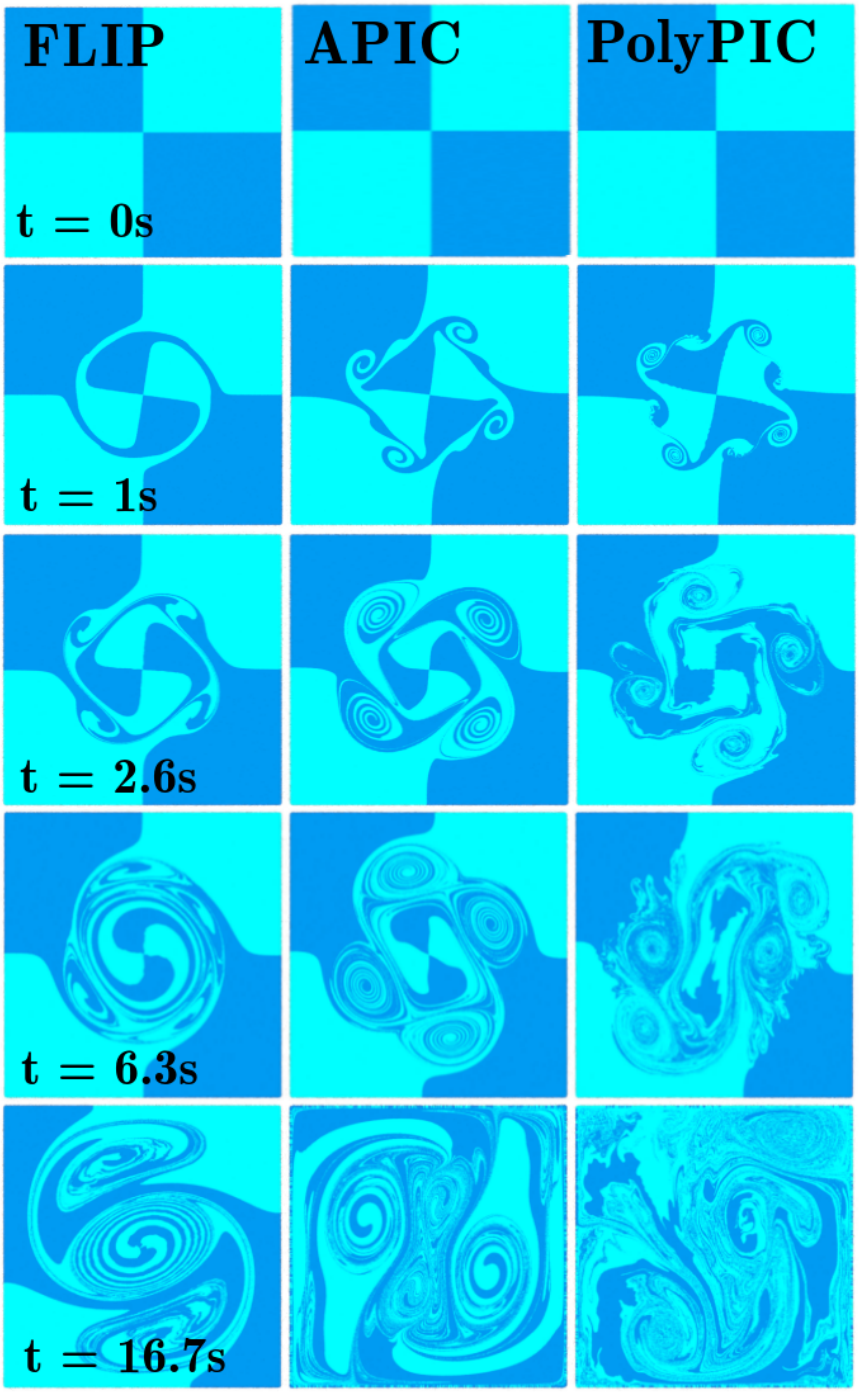


Figure 6.3: **Vortex sheet.** We compare from left to right FLIP, APIC, and PolyPIC with 2D incompressible flow. The initial conditions are of a rotating circle surrounded by stationary fluid. This creates a vortex sheet which our method effectively resolves. The bottom row shows that despite the energetic nature of our method, our simulations are stable at long runtimes.

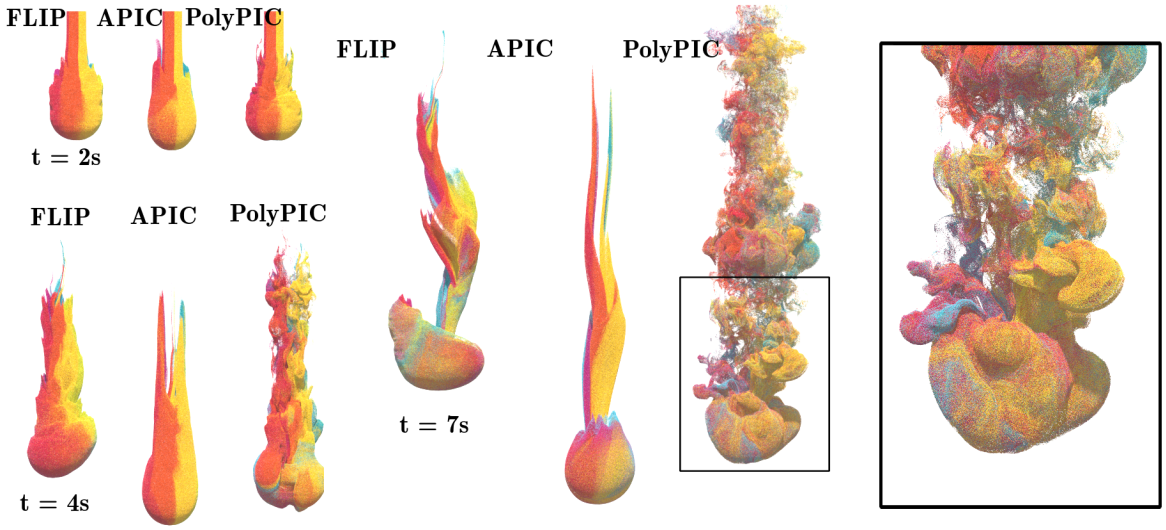


Figure 6.4: **Ink drop.** We compare from left to right FLIP, APIC, and PolyPIC for an inkjet in an ambient incompressible fluid. PolyPIC more effectively resolves the vorticial details.

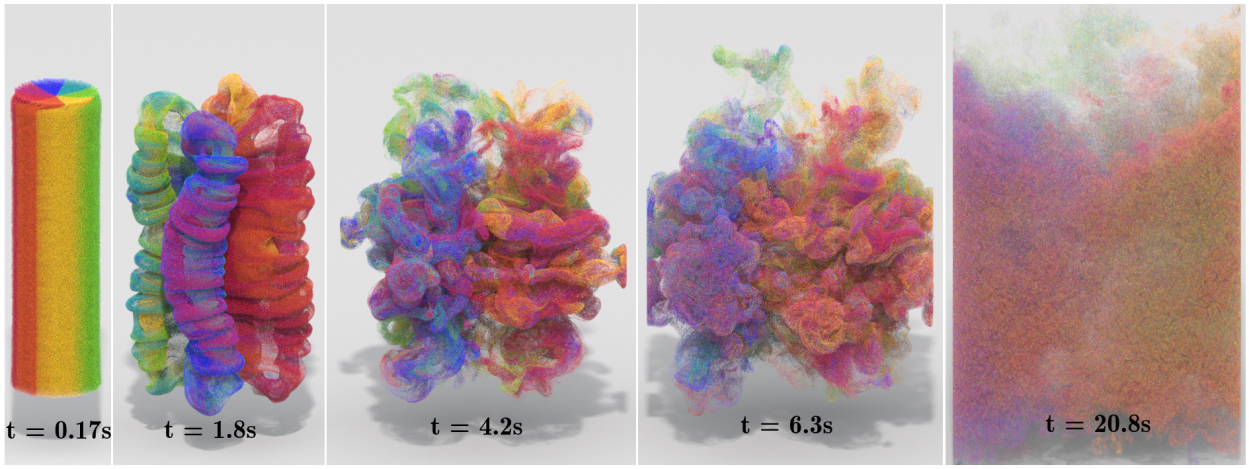


Figure 6.5: **Rotating column of colored dust**. We demonstrate intricate vorticial patterns that arise from simple initial conditions with incompressible flow. PolyPIC achieves great detail with modest spatial grid resolution ($88 \times 132 \times 88$). The rightmost image shows that despite the energetic nature of our method, our simulations are stable at long runtimes.

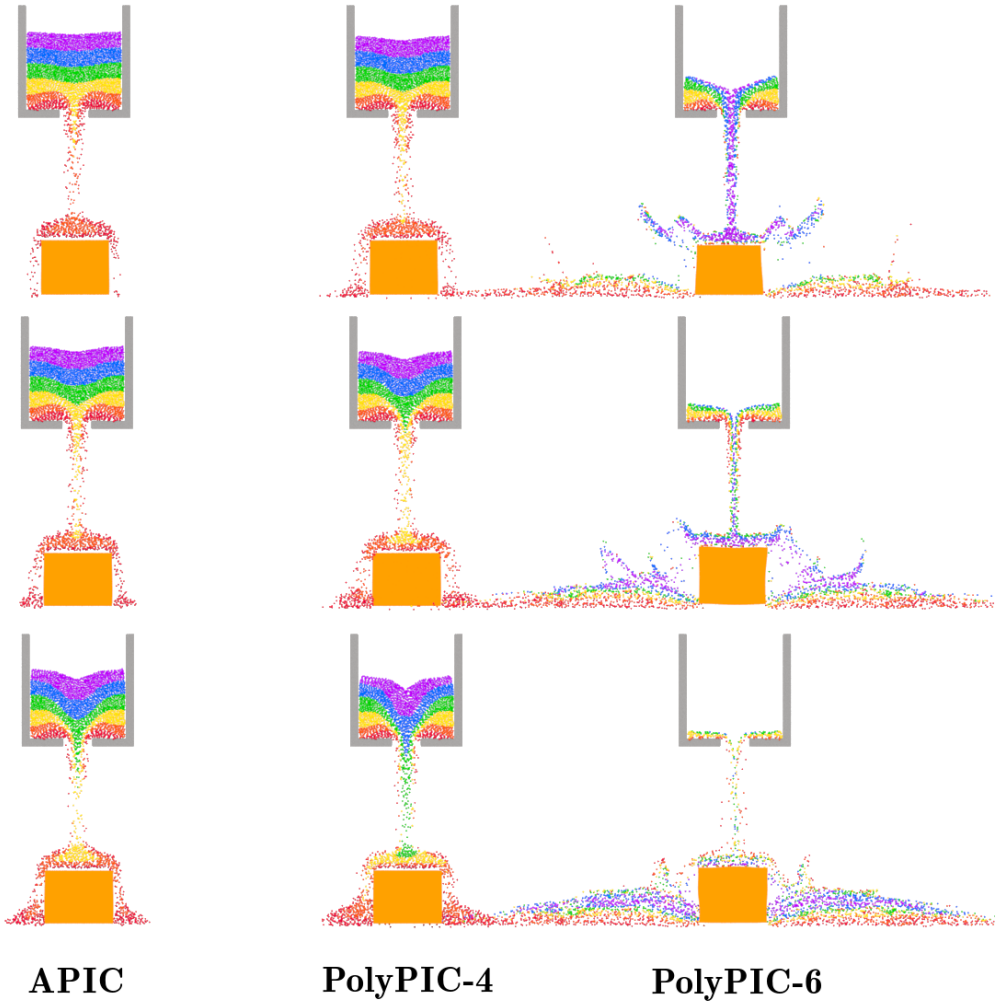


Figure 6.6: **MPM elastoplasticity**. Rainbow colored sand is poured onto an elastic Jell-O square. We compare APIC (left) vs. PolyPIC with (from left to right) $N_r = 4$ and $N_r = 6$. Notice that increasing degrees of PolyPIC allow for more energetic sand flowing and Jell-O bouncing.

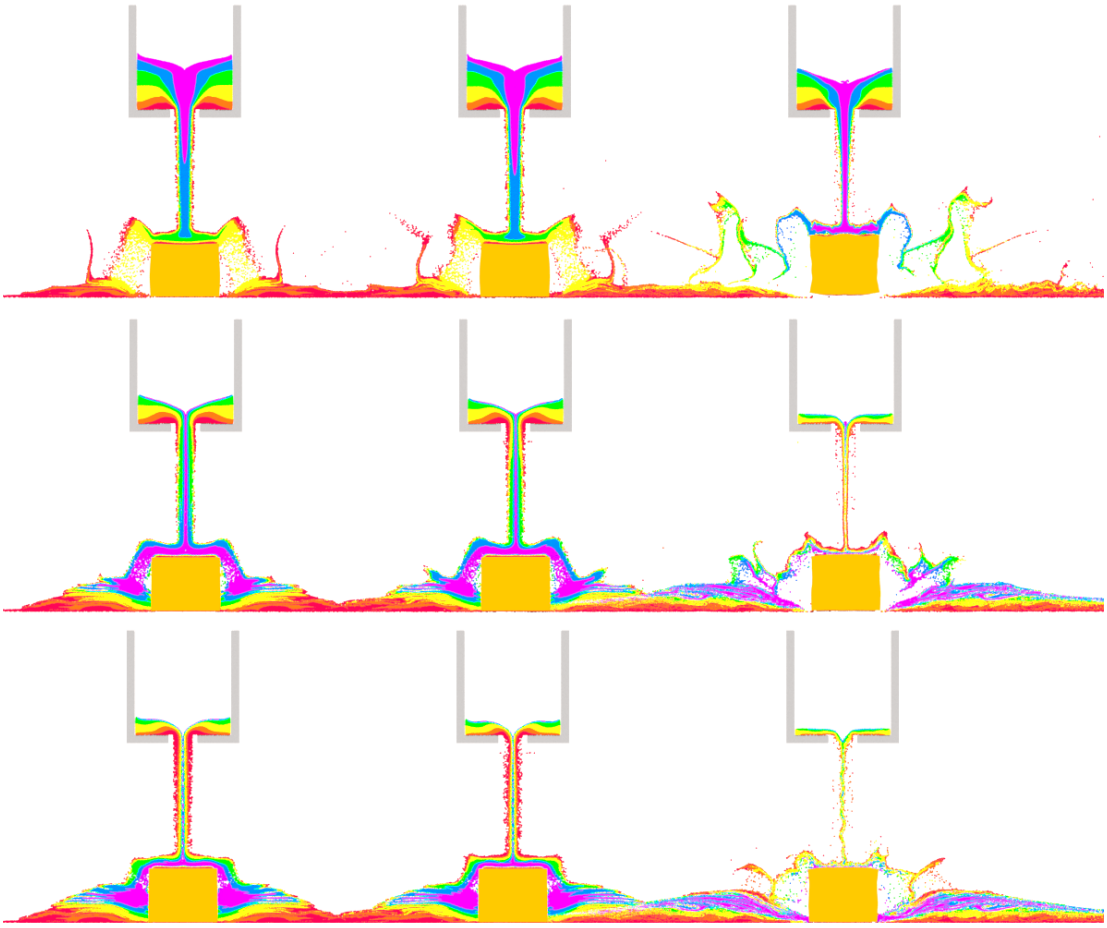


Figure 6.7: **MPM elastoplasticity refinement**. We verify that the behavior exhibited by PolyPIC with $N_r = 6$ at lower resolution in Figure 6.6 is exhibited by PolyPIC with $N_r = 4$ and APIC under refinement.

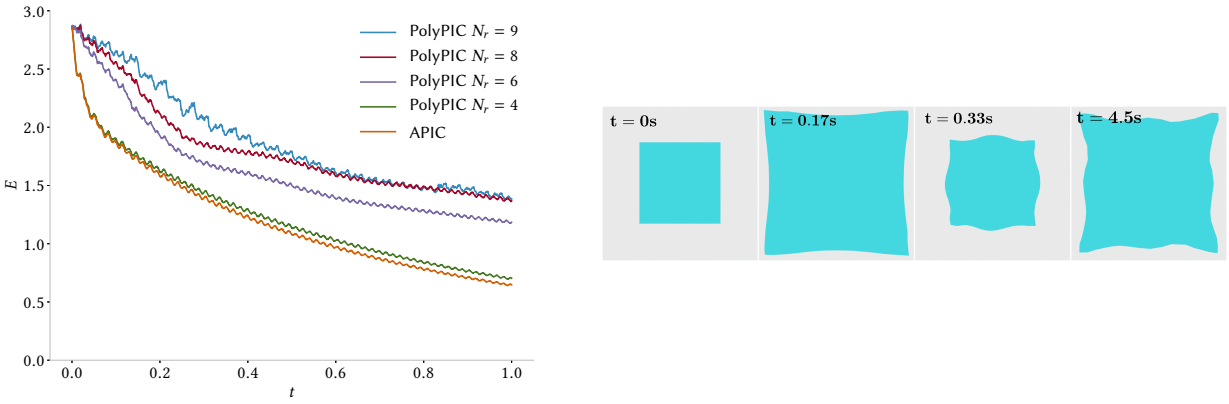


Figure 6.8: **Energy conservation**. We plot of the total energy as a function of time for an elastic square with initial compressive dilation. The energy is calculated as the sum of the elastic potential energy on the particles and the kinetic energy on the grid.

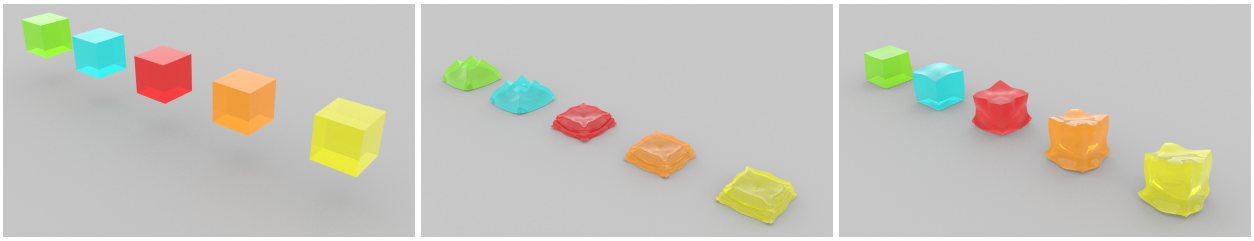


Figure 6.9: **MPM hyperelasticity**. We compare from left to right APIC (green) and PolyPIC with $N_r = 8$ (blue), $N_r = 11$ (red), $N_r = 14$ (orange), $N_r = 18$ (yellow). PolyPIC better conserves total energy which results in less numerical damping of the deformable motion.

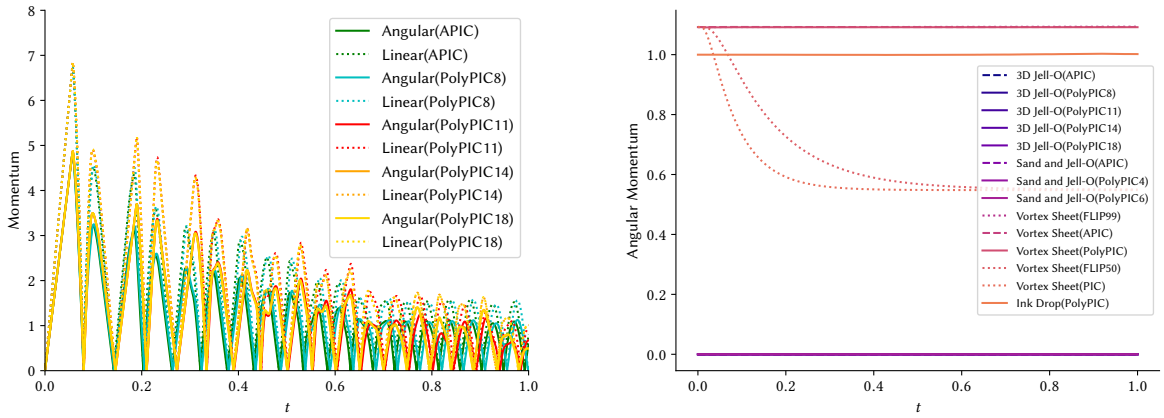


Figure 6.10: **Momentum conservation**. The top figure plots the linear and angular momenta for the falling Jell-O's in Figure 6.9. The bottom illustrates the angular momentum loss resulting form transfers. We plot the momenta $\hat{\mathbf{l}}^n$ and $\hat{\mathbf{p}}^n$ from Equation (6.11) to monitor the transfers effects on conservation. APIC and PolyPIC preserve angular momentum during transfers, however the FLIP/PIC blends are commonly used in incompressible flow simulations do not. We illustrate this by comparing with increasing amounts of PIC.

APPENDIX A

Drucker-Prager elastoplasticity derivations

A.1 Energy dissipation

The total energy is sum of potential and kinetic energy:

$$E(t;\bar{\Omega}) = \int_{\bar{\Omega}} \frac{R(\mathbf{X},0)}{2} \left|\mathbf{V}(\mathbf{X},t)\right|_2^2 d\mathbf{X} + \int_{\bar{\Omega}} \psi(\mathbf{F}^E(\mathbf{X},t), \mathbf{F}^P(\mathbf{X},t)) d\mathbf{X}. \tag{A.1}$$

The rate of change of the energy is

$$E'(t) = \int_{\bar{\Omega}} R(\mathbf{X},0)\mathbf{V}(\mathbf{X},t)\mathbf{A}(\mathbf{X},t) d\mathbf{X} + \int_{\bar{\Omega}} \frac{\partial \psi}{\partial \mathbf{F}^E}(\mathbf{F}^E(\mathbf{X},t), \mathbf{F}^P(\mathbf{X},t)) : \dot{\mathbf{F}^E}(\mathbf{X},t) d\mathbf{X}. \tag{A.2}$$

With $\mathbf{P} = \frac{\partial \psi}{\partial \mathbf{F}^E}(\mathbf{F}^E, \mathbf{F}^P)\mathbf{F}^{P-T}$,

$$\begin{aligned}
\int_{\bar{\Omega}} \frac{\partial \psi}{\partial \mathbf{F}^E} : \dot{\mathbf{F}^E} d\mathbf{X} &= \int_{\bar{\Omega}} \frac{\partial \psi}{\partial \mathbf{F}^E} : \left(\dot{\mathbf{F}}\mathbf{F}^{P-1} - \mathbf{F}^E \dot{\mathbf{F}}^P \mathbf{F}^{P-1}\right) d\mathbf{X} \\
&= \int_{\bar{\Omega}} \mathbf{P} : \left(\dot{\mathbf{F}} - \mathbf{F}^E \dot{\mathbf{F}}^P\right) d\mathbf{X} \\
&= -\int_{\bar{\Omega}} \mathbf{V} \cdot \left(\nabla^{\mathbf{X}} \cdot \mathbf{P}\right) d\mathbf{X} + \int_{\partial\bar{\Omega}} \mathbf{V} \cdot (\mathbf{P}\mathbf{N})\, ds(\mathbf{X}) - \int_{\bar{\Omega}} \left(\mathbf{F}^{E^T}\mathbf{P}\right) : \dot{\mathbf{F}}^P d\mathbf{X}.
\end{aligned}$$

Using $R(\mathbf{X},0)\mathbf{A}(\mathbf{X},t) = (\nabla^{\mathbf{X}} \cdot \mathbf{P})(\mathbf{X},t)$ with Equation (A.2) gives

$$E'(t) = \int_{\partial\bar{\Omega}} \mathbf{V} \cdot (\mathbf{P}\mathbf{N})\, ds(\mathbf{X}) - \int_{\bar{\Omega}} \left(\mathbf{F}^{E^T}\mathbf{P}\right) : \dot{\mathbf{F}}^P d\mathbf{X}.$$

The rate of energy dissipation due to plasticity is therefore

$$\dot{w}_p = \mathbf{F}^{E^T}\mathbf{P} : \dot{\mathbf{F}}^P \tag{A.3}$$

$$= \mathbf{P}\mathbf{F}^T : \mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1} = \boldsymbol{\tau} : \mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1} \tag{A.4}$$

A.2 Isotropy

In the case of isotropic potential energy, we have two conclusions:

- $\boldsymbol{\tau}$ and $\mathbf{B}^E$ commute: $\mathbf{B}^E\boldsymbol{\tau} = \mathbf{B}^E\boldsymbol{\tau}$.
- $\dot{w}_p = \boldsymbol{\tau} : (\mathcal{L}_\mathbf{v}\mathbf{B}^E\mathbf{B}^{E^{-1}})$.

We will prove the two statements in this section.

$\boldsymbol{\tau}$ and $\mathbf{B}^E$ commute

We make the assumption that the energy density ψ is isotropic, so we can write ψ as a function of the three invariants of $\mathbf{F}^E$

$$\psi(\mathbf{F}^E, \mathbf{F}^P) = \hat{\psi}(I(\mathbf{F}^E), II(\mathbf{F}^E), III(\mathbf{F}^E)).$$

Therefore

$$\frac{\partial\psi}{\partial\mathbf{F}^E}(\mathbf{F}^E, \mathbf{F}^P) = a\mathbf{F}^E + b\mathbf{B}^E\mathbf{F}^E + c\mathbf{F}^{E^{-T}}$$

for some scalar a, b, c and

$$\boldsymbol{\tau} = \mathbf{P}\mathbf{F}^T = \frac{\partial\psi}{\partial\mathbf{F}^E}(\mathbf{F}^E, \mathbf{F}^P)\mathbf{F}^{P^{-T}}\mathbf{F}^T = a\mathbf{B}^E + b\mathbf{B}^{E^2} + c\mathbf{I}.$$

This means that $\boldsymbol{\tau}$ and $\mathbf{B}^E$ have the principal space. In particular,$\boldsymbol{\tau}$ and $\mathbf{B}^E$ commute:

$$\mathbf{B}^E\boldsymbol{\tau} = \mathbf{B}^E\boldsymbol{\tau}.$$

Plastic rate of dissipation

We showed in Section A.4 that

$$\dot{w}_p = (\mathbf{F}^{E^T}\mathbf{P}) : \dot{\mathbf{F}^P} = \boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}^P}\mathbf{F}^{-1}\right).$$

We will show that this is also equal to $\boldsymbol{\tau} : (\mathcal{L}_\mathbf{v}\mathbf{B}^E\mathbf{B}^{E^{-1}})$. By definition of $\mathcal{L}_\mathbf{v}\mathbf{B}^E$, we have

$$\mathcal{L}_\mathbf{v}\mathbf{B}^E\mathbf{B}^{E^{-1}} = -\mathbf{F}^E\mathbf{F}^{P^{-T}}\dot{\mathbf{F}^P}^T\mathbf{F}^{E^{-1}} - \mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}.$$

Plug this in $\boldsymbol{\tau} : \left(\mathcal{L}_\mathbf{v}\mathbf{B}^E\mathbf{B}^{E^{-1}}\right)$,

$$\begin{aligned}
\boldsymbol{\tau} : \left(\mathcal{L}_\mathbf{v}\mathbf{B}^E\mathbf{B}^{E^{-1}}\right) &= -\boldsymbol{\tau} : \left(\mathbf{F}^E\mathbf{F}^{P^{-T}}\dot{\mathbf{F}^P}^T\mathbf{F}^{E^{-1}}\right) - \boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}\right) \\
&= -\text{trace}\left(\boldsymbol{\tau}\mathbf{F}^E\mathbf{F}^{P^{-T}}\dot{\mathbf{F}^P}^T\mathbf{F}^{E^{-1}}\right) - \boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}\right) \\
&= -\text{trace}\left(\boldsymbol{\tau}\mathbf{F}^E\mathbf{F}^E T\mathbf{F}^{E^{-T}}\mathbf{F}^{P^{-T}}\dot{\mathbf{F}^P}^T\mathbf{F}^{E^{-1}}\right) - \boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}\right) \\
&= -\text{trace}\left(\boldsymbol{\tau}\mathbf{B}^E\mathbf{F}^{-T}\dot{\mathbf{F}^P}^T\mathbf{F}^{E^{-1}}\right) - \boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}\right) \\
&= -\text{trace}\left(\mathbf{B}^E\boldsymbol{\tau}\mathbf{F}^{-T}\dot{\mathbf{F}^P}^T\mathbf{F}^{E^{-1}}\right) - \boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}\right) \\
&= -\text{trace}\left(\boldsymbol{\tau}\mathbf{F}^{-T}\dot{\mathbf{F}^P}^T\mathbf{F}^{E^{-T}}\right) - \boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}\right) \\
&= -2\boldsymbol{\tau} : \left(\mathbf{F}^E\dot{\mathbf{F}}^P\mathbf{F}^{-1}\right)
\end{aligned}$$

A.3 Kirchhoff stress and hencky strain

In this section we derive the expression for

$$\boldsymbol{\tau} = \tau_{\alpha\beta}\mathbf{q}_\alpha \otimes \mathbf{q}_\beta, \quad \tau_{\alpha\beta} = 2\mu\epsilon^L_{\alpha\beta} + \lambda\epsilon^L_{\gamma\gamma}\delta_{\alpha\beta}. \tag{A.5}$$

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.5) with index free notation assuming all variables are in 2D.

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

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,

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

A.4 Plastic Dissipation is Nonnegative

Recall that we have previously defined $\mathbf{s} = \boldsymbol{\tau} - \frac{1}{d}\mathrm{tr}(\boldsymbol{\tau})$, and that $\mathbf{G} = \frac{\partial y}{\partial \boldsymbol{\tau}} - \frac{1}{d}\mathrm{tr}(\frac{\partial y}{\partial \boldsymbol{\tau}})\mathbf{I}$, i.e. it satisfies $\mathbf{G} = -\gamma \mathcal{L}_{\mathbf{v}} \mathbf{B}^E \mathbf{B}^{E^{-1}}$ (e.g. see Section (§4.2.1)). Therefore

$$\begin{aligned}
\dot{w}^P &= \boldsymbol{\tau} : \mathbf{l}^P \\
&= -\boldsymbol{\tau} : \frac{1}{2}\mathcal{L}_{\mathbf{v}} \mathbf{B}^E \mathbf{B}^{E^{-1}} \\
&= \gamma \boldsymbol{\tau} : \mathbf{G} \\
&= \frac{\gamma}{\|\mathbf{s}\|_F} \boldsymbol{\tau} : \mathbf{s} \\
&= \frac{\gamma}{\|\mathbf{s}\|_F} \left(\mathbf{s} + \frac{1}{d}\mathrm{tr}(\boldsymbol{\tau})\mathbf{I} \right) : \mathbf{s} \\
&= \gamma \|\mathbf{s}\|_F .
\end{aligned}$$

Thus all that remains to prove is that $\gamma \geq 0$. To do this we use the constraint that $\frac{\partial y}{\partial t} \leq 0$ when $y = 0$.

$$\begin{aligned}
\frac{\partial y}{\partial t} &= \frac{\partial y}{\partial \boldsymbol{\tau}} : \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{B}^E} : \dot{\mathbf{B}}^E \\
&= \frac{\partial y}{\partial \boldsymbol{\tau}} : \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{B}^E} : \left(\mathbf{l}\mathbf{B}^E + \mathbf{B}^E \mathbf{l}^T - 2\gamma \frac{\partial y}{\partial \boldsymbol{\tau}} \mathbf{B}^E \right) \\
&= \underbrace{\frac{\partial y}{\partial \boldsymbol{\tau}} : \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{B}^E} : \left(\mathbf{l}\mathbf{B}^E + \mathbf{B}^E \mathbf{l}^T \right)}_{\eta} - 2\gamma \underbrace{\frac{\partial y}{\partial \boldsymbol{\tau}} : \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{B}^E} : \left(\frac{\partial y}{\partial \boldsymbol{\tau}} \mathbf{B}^E \right)}_{\nu} .
\end{aligned}$$

So

$$0 = \eta - 2\gamma\nu \Longrightarrow \gamma = \frac{\eta}{2\nu}$$

Note that η is what $\frac{\partial y}{\partial t}$ would be in the absence of plastic flow. Thus if $\eta \leq 0$ the material is deforming in such a way that the yield function is going down, and therefore is undergoing

elastic deformation which means $\gamma = 0$. Otherwise $\eta > 0$ and

$$\begin{aligned}\nu &= \frac{\partial y}{\partial \mathbf{B}^E} : \left(2 \frac{\mathbf{s}}{\|\mathbf{s}\|_F} \mathbf{B}^E \right) \\ &= 2 \frac{\partial y}{\partial \mathbf{B}^E} \mathbf{B}^E : \frac{\mathbf{s}}{\|\mathbf{s}\|_F}.\end{aligned}$$

Applying the Hencky strain derivative lemma to y we have

$$\begin{aligned}\nu &= \frac{\partial y}{\partial \boldsymbol{\epsilon}^E} : \frac{\mathbf{s}}{\|\mathbf{s}\|_F} \\ &= \frac{\partial y}{\partial \boldsymbol{\tau}} : \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{\epsilon}^E} : \frac{\mathbf{s}}{\|\mathbf{s}\|_F} \\ &= \left(\frac{\mathbf{s}}{\|\mathbf{s}\|} + \tilde{\eta} \mathbf{I} \right) : \mathbb{C} : \frac{\mathbf{s}}{\|\mathbf{s}\|_F} \\ &= 2\mu \|\mathbf{s}\|_F.\end{aligned}$$

APPENDIX B

Thin shell derivations

B.1 FEM Force computation

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 the paper 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

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

where

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

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

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

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

B.2 Grid force computation

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

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

Note that

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

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

Ignoring further plastic flow, we have

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

and thus,

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

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

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

B.3 QR and Elastic Potential

We can use QR orthogonalization of deformed material directions to define

$$\mathbf{q}_i r_{ij} = \mathbf{F}\bar{\mathbf{a}}_j,\ \mathbf{F} = r_{ij}\mathbf{q}_i \otimes \bar{\mathbf{a}}_j,\ r_{ij} = 0 \text{ for } i > j. \tag{B.2}$$

B.3.1 Change of basis tensor

Define the change of basis tensor

$$\mathbf{Q} = Q_{ij}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j \tag{B.3}$$

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

B.3.2 Differentials

The QR differential satisfies

$$\mathbf{q}_k \cdot \delta\mathbf{q}_i r_{ij} + \delta r_{kj} = \mathbf{q}_k \cdot (\delta\mathbf{F}\bar{\mathbf{a}}_j),\ \delta\mathbf{F} = \delta r_{ij}\mathbf{q}_i \otimes \bar{\mathbf{a}}_j + r_{ij}\delta\mathbf{q}_i \otimes \bar{\mathbf{a}}_j \tag{B.4}$$

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

$$\delta\mathbf{F} = \delta\mathbf{Q}\mathbf{R} + \mathbf{Q}\delta\mathbf{R} \tag{B.5}$$

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

$$\delta\mathbf{Q} = \delta Q_{ij}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j,\ \delta Q_{ij} = \delta\mathbf{q}_j \cdot \bar{\mathbf{a}}_i,\ \delta\mathbf{q}_i = \delta\mathbf{Q}\bar{\mathbf{a}}_i \tag{B.6}$$

$$\delta\mathbf{R} = \delta r_{ij}\bar{\mathbf{a}}_i \otimes \bar{\mathbf{a}}_j \tag{B.7}$$

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

B.4 Elastic potential and stresses

Define the hyperelastic potential as

$$\psi(\mathbf{F}) = \hat{\psi}([\mathbf{R}]) \tag{B.8}$$

where

$$[\mathbf{R}] = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \end{pmatrix}. \tag{B.9}$$

The differential satisfies

$$\delta\psi(\mathbf{F}) = \frac{\partial\psi}{\partial\mathbf{F}}(\mathbf{F}) : \delta\mathbf{F} = \mathbf{P} : \delta\mathbf{F} = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij} \tag{B.10}$$

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

$$\delta r_{ij}\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j) + r_{ij}\delta\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j) = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij}. \tag{B.11}$$

Similarly,

$$\mathbf{P} : \delta\mathbf{F} = \mathbf{P} : (\delta\mathbf{Q}\mathbf{R}) + \mathbf{P} : (\mathbf{Q}\delta\mathbf{R}) = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij} \tag{B.12}$$

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

$$\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j)\, \delta r_{ij} = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij} \tag{B.13}$$

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,

$$\mathbf{P} : (\mathbf{Q}\delta\mathbf{R}) = (\mathbf{Q}^T\mathbf{P}) : \delta\mathbf{R} = \delta r_{ij}\bar{\mathbf{a}}_i \cdot \left(\mathbf{Q}^T\mathbf{P}\bar{\mathbf{a}}_j\right) = \delta r_{ij}\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j) = \frac{\partial\hat{\psi}}{\partial r_{ij}}([\mathbf{R}])\delta r_{ij}. \tag{B.14}$$

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

$$0 = r_{ij}\delta\mathbf{q}_i \cdot (\mathbf{P}\bar{\mathbf{a}}_j)\,. \tag{B.15}$$

Similarly,

$$\begin{aligned} 0 = \mathbf{P} : (\delta\mathbf{Q}\mathbf{R}) &= \left(\mathbf{P}\mathbf{R}^T\right) : \delta\mathbf{Q} = \left(\mathbf{P}\mathbf{R}^T\right) : \left(\delta\mathbf{Q}\mathbf{Q}^T\mathbf{Q}\right) \\ &= \left(\mathbf{P}\mathbf{R}^T\mathbf{Q}^T\right) : \left(\delta\mathbf{Q}\mathbf{Q}^T\right) = \left(\mathbf{P}\mathbf{F}^T\right) : \left(\delta\mathbf{Q}\mathbf{Q}^T\right) \end{aligned} \tag{B.16}$$

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,

$$\mathbf{P} = P_{ij}\mathbf{q}_i \otimes \bar{\mathbf{a}}_j,\ \boldsymbol{\tau} = P_{ij}r_{kj}\mathbf{q}_i \otimes \mathbf{q}_k = \tau_{ik}\mathbf{q}_i \otimes \mathbf{q}_k \tag{B.17}$$

and we know $P_{ij} = \frac{\partial \hat{\psi}}{\partial r_{ij}}$ for $i \leq j$ from Equation B.13. Thus

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \begin{pmatrix} r_{11} & & \\ r_{12} & r_{22} & \\ r_{13} & r_{23} & r_{33} \end{pmatrix} \tag{B.18}$$

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

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

$$\begin{pmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{pmatrix} = \begin{pmatrix} \frac{\partial \hat{\psi}}{\partial r_{11}} r_{11} + \frac{\partial \hat{\psi}}{\partial r_{12}} r_{12} + \frac{\partial \hat{\psi}}{\partial r_{13}} r_{13} & \frac{\partial \hat{\psi}}{\partial r_{12}} r_{22} + \frac{\partial \hat{\psi}}{\partial r_{13}} r_{32} & \frac{\partial \hat{\psi}}{\partial r_{13}} r_{33} \\ \frac{\partial \hat{\psi}}{\partial r_{12}} r_{22} + \frac{\partial \hat{\psi}}{\partial r_{13}} r_{32} & \frac{\partial \hat{\psi}}{\partial r_{22}} r_{22} + \frac{\partial \hat{\psi}}{\partial r_{23}} r_{32} & \frac{\partial \hat{\psi}}{\partial r_{23}} r_{33} \\ \frac{\partial \hat{\psi}}{\partial r_{13}} r_{33} & \frac{\partial \hat{\psi}}{\partial r_{23}} r_{33} & \frac{\partial \hat{\psi}}{\partial r_{33}} r_{33} \end{pmatrix} \tag{B.20}$$

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

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

$$= \begin{pmatrix} \gamma s_1^2 & \gamma s_1 s_2 & \gamma s_1 s_3 \\ \gamma s_1 s_2 & \gamma s_2^2 & \gamma s_2 s_3 \\ \gamma s_1 s_3 & \gamma s_2 s_3 & f'(s_3) \end{pmatrix} \tag{B.22}$$

B.5 Frictional Contact Yield Condition

Coulomb friction places a constraint on the stress as

$$|\mathbf{t}_S| \leq -c_F \sigma_n \tag{B.23}$$

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

$$(c\mathbf{q}_1 + s\mathbf{q}_2) \cdot \boldsymbol{\sigma} \mathbf{q}_3 + c_F \mathbf{q}_3 \cdot \boldsymbol{\sigma} \mathbf{q}_3 \leq 0. \tag{B.24}$$

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

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

Substituting in the expression for $\boldsymbol{\tau}$ from equation (B.22), 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 the paper 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.

B.6 Denting Yield Condition and Return Mapping

We apply the von Mises yield condition to the Kirchhoff-Stress in Equation (A.5)

This condition states that the deviatoric component of the stress is less than a threshold

value c_{vM}

$$f_{vM}(\boldsymbol{\tau}) = |\boldsymbol{\tau} - \frac{\mathrm{tr}(\boldsymbol{\tau})}{3}\mathbf{I}|_F \leq c_{vM}. \tag{B.26}$$

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

$$f_{vM}(\boldsymbol{\tau}) = \sqrt{\frac{2}{3}\left({\tau_1}^2 + {\tau_2}^2 + {\tau_3}^2 - (\tau_1\tau_2 + \tau_2\tau_3 + \tau_1\tau_3)\right)} \tag{B.27}$$

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.

$$\mathbf{F}^{KL,E^{\mathrm{tr}}}, \mathbf{F}^{KL,P^{\mathrm{tr}}}, \boldsymbol{\tau}^{KL^{\mathrm{tr}}} \rightarrow \mathbf{F}^{KL,E}, \mathbf{F}^{KL,P}, \boldsymbol{\tau}^{KL}. \tag{B.28}$$

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

$$\mathbf{F}^{KL} = \mathbf{F}^{KL,E^{\mathrm{tr}}}\mathbf{F}^{KL,P^{\mathrm{tr}}} = \mathbf{F}^{KL,E}\mathbf{F}^{KL,P}. \tag{B.29}$$

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.5). This is done by computing the QR decomposition of the trial elastic deformation gradient $\mathbf{F}^{KL,E^{\mathrm{tr}}} = r_{\alpha\beta}^{KL,E^{\mathrm{tr}}} \mathbf{q}_\alpha^{KL,E} \otimes \bar{\mathbf{a}}_\beta + \mathbf{q}_3^{KL,E} \otimes \bar{\mathbf{a}}_3$. Then we compute the SVD

of matrix $[\mathbf{r}^{KL,E^{\text{tr}}}] \in \mathbb{R}^{2\times 2}$ and the trial strain $[\boldsymbol{\epsilon}^{L^{\text{tr}}}]$

$$[\mathbf{r}^{KL,E^{\text{tr}}}] = [\mathbf{U}^E] \begin{pmatrix} \sigma_1^{E^{\text{tr}}} & \\ & \sigma_2^{E^{\text{tr}}} \end{pmatrix} [\mathbf{V}^E]^T \tag{B.30}$$

$$[\boldsymbol{\epsilon}^{L^{\text{tr}}}] = [\mathbf{U}^E] \begin{pmatrix} \log(\sigma_1^{E^{\text{tr}}}) & \\ & \log(\sigma_2^{E^{\text{tr}}}) \end{pmatrix} [\mathbf{U}^E]^T \tag{B.31}$$

From Equation (A.5) 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 $[\boldsymbol{\tau}^{KL^{\text{tr}}}]$

$$[\boldsymbol{\tau}^{KL^{\text{tr}}}] = 2\mu[\boldsymbol{\epsilon}^{L^{\text{tr}}}] + \lambda \text{tr}([\boldsymbol{\epsilon}^{L^{\text{tr}}}])\mathbf{I} = [\mathbf{U}^E] \begin{pmatrix} \tau_1^{KL^{\text{tr}}} & \\ & \tau_2^{KL^{\text{tr}}} \end{pmatrix} [\mathbf{U}^E]^T. \tag{B.32}$$

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 [28] which greatly simplifies the return mapping. We illustrate this property in Figure B.1. After projection, we rebuild the matrix without changing the eigenvectors and rebuild $\boldsymbol{\tau}^{KL}$ from the matrix

$$[\boldsymbol{\tau}^{KL}] = [\mathbf{U}^E] \begin{pmatrix} {\tau^{KL}}_1 & \\ & {\tau^{KL}}_2 \end{pmatrix} [\mathbf{U}^E]^T, \ \boldsymbol{\tau}^{KL} = \tau_{\alpha\beta}^{KL} \mathbf{q}_{\alpha}^{KL,E} \otimes \mathbf{q}_{\beta}^{KL,E} \tag{B.33}$$

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

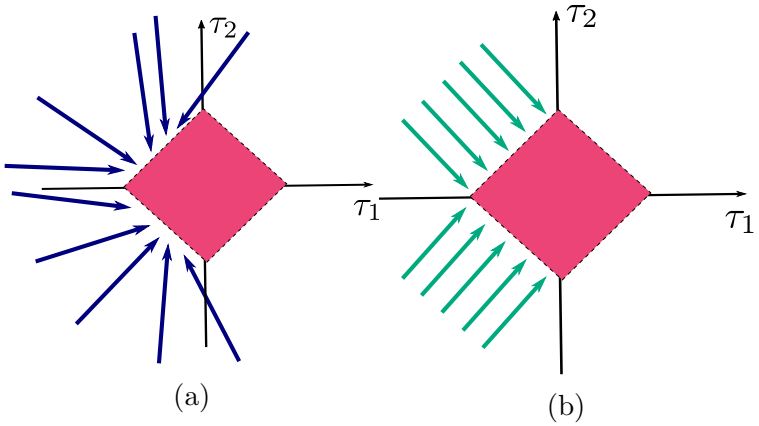


Figure B.1: **Simplified return Mapping**. In general in the return mapping direction is non trivial (left). Quadratic Hencky strain energy density simplifies the return mapping (right).

is computed from the projected principal stresses from

$$[\boldsymbol{\epsilon}^L] = [\mathbf{U}^E] \begin{pmatrix} \log(\sigma_1^E) & \\ & \log(\sigma_2^E) \end{pmatrix} [\mathbf{U}^E]^T \tag{B.34}$$

$$\begin{pmatrix} \log(\sigma_1^E) \\ \log(\sigma_2^E) \end{pmatrix} = \begin{pmatrix} 2\mu+\lambda & \lambda \\ \lambda & 2\mu+\lambda \end{pmatrix}^{-1} \begin{pmatrix} \tau^{KL}{}_1 \\ \tau^{KL}{}_2 \end{pmatrix} \tag{B.35}$$

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

$$[\hat{\mathbf{F}}^{KL,E}] = [\mathbf{U}^E] \begin{pmatrix} \sigma_1^E & \\ & \sigma_2^E \end{pmatrix} [\mathbf{V}^E]^T. \tag{B.36}$$

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 (5.15).

APPENDIX C

PolyPIC

C.1 List of Bases

C.1.1 Linear interpolation

Polynomials of the form

$$s(\mathbf{z}) = \prod_{\beta=1}^{d} z_\beta^{i_\beta}, \quad i_\beta = 0, 1$$

are all we need for linear interpolation. We have $\mathcal{S}_{pr}^n \cdot \left(\mathbf{m}_p^n \mathcal{S}_{pt}^n\right) = 0$ for all $r \neq t$.

C.1.2 Quadratic interpolation

For quadratic interpolation, by replacing $z_\beta^{i_\beta}$ with $g_\beta(w) = w^2 - \frac{x_{p\beta}^n\left(\Delta x^2 - 4(x_{p\beta}^n)^2\right)}{\Delta x^2} w - \frac{\Delta x^2}{4}$ in

$$s(\mathbf{z}) = \prod_{\beta=1}^{d} z_\beta^{i_\beta}, \quad i_\beta = 0, 1, 2$$

whenever $i_\beta = 2$, we get the full set of basis vectors. For completeness we list all the bases below.

In 2D,

$$s_1(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = 1$$
$$s_2(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = x_{\mathbf{i}_{kp1}} - x_{p1}^n \qquad s_3(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = x_{\mathbf{i}_{kp2}} - x_{p2}^n$$
$$s_4(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = (x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_5(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n) \qquad s_6(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_7(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n) \qquad s_8(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp1}} - x_{p1}^n)$$
$$s_9(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$

In 3D,

$$s_1(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = 1$$
$$s_2(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = x_{\mathbf{i}_{kp1}} - x_{p1}^n$$
$$s_3(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = x_{\mathbf{i}_{kp2}} - x_{p2}^n$$
$$s_4(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = x_{\mathbf{i}_{kp3}} - x_{p3}^n$$
$$s_5(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = (x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_6(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = (x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_7(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = (x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_8(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = (x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_9(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)$$
$$s_{10}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_{11}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_{12}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_{13}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_{14}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_{15}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$

$$s_{16}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_{17}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_{18}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$

$$s_{19}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp1}} - x_{p1}^n)$$
$$s_{20}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_{21}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$

$$s_{22}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)(x_{\mathbf{i}_{kp1}} - x_{p1}^n)$$
$$s_{23}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_{24}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)(x_{\mathbf{i}_{kp1}} - x_{p1}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$

$$s_{25}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)(x_{\mathbf{i}_{kp3}} - x_{p3}^n)$$
$$s_{26}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_1(x_{\mathbf{i}_{kp1}} - x_{p1}^n)g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)(x_{\mathbf{i}_{kp2}} - x_{p2}^n)$$
$$s_{27}(\mathbf{x}_{\mathbf{i}_{kp}^n}^n - \mathbf{x}_p^n) = g_2(x_{\mathbf{i}_{kp2}} - x_{p2}^n)g_3(x_{\mathbf{i}_{kp3}} - x_{p3}^n)(x_{\mathbf{i}_{kp1}} - x_{p1}^n)$$

The entries for $\mathcal{S}_{pr}^{n} \cdot \left(\mathbf{m}_{p}^{n}\mathcal{S}_{pr}^{n}\right)$, $r = 1, 2, \cdots, 27$ are:

$$
\begin{array}{l}
1, \\
\frac{\Delta x^2}{4}, \frac{\Delta x^2}{4}, \frac{\Delta x^2}{4}, \\
\frac{\Delta x^4}{16}, \frac{\Delta x^4}{16}, \frac{\Delta x^4}{16}, \frac{\Delta x^6}{64}, \\
\frac{\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right)}{16\Delta x^2}, \frac{\left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right)}{16\Delta x^2}, \frac{\left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right)}{16\Delta x^2}, \\
\frac{\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right) \left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right)}{256\Delta x^4}, \\
\frac{\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right) \left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right)}{256\Delta x^4}, \\
\frac{\left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right) \left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right)}{256\Delta x^4}, \\
\frac{\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right) \left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right) \left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right)}{4096\Delta x^6}, \\
\frac{1}{64}\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right), \frac{1}{64}\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right), \\
\frac{1}{256}\left(3\Delta x^2 - 4x^2\right)\left(\Delta x^3 - 4\Delta x x^2\right)^2, \\
\frac{1}{64}\left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right), \frac{1}{64}\left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right), \\
\frac{1}{256}\left(3\Delta x^2 - 4y^2\right)\left(\Delta x^3 - 4\Delta x y^2\right)^2, \\
\frac{1}{64}\left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right), \frac{1}{64}\left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right), \\
\frac{1}{256}\left(3\Delta x^2 - 4z^2\right)\left(\Delta x^3 - 4\Delta x z^2\right)^2, \\
\frac{\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right) \left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right)}{1024\Delta x^2}, \\
\frac{\left(\Delta x^2 - 4x^2\right)^2 \left(3\Delta x^2 - 4x^2\right) \left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right)}{1024\Delta x^2}, \\
\frac{\left(\Delta x^2 - 4y^2\right)^2 \left(3\Delta x^2 - 4y^2\right) \left(\Delta x^2 - 4z^2\right)^2 \left(3\Delta x^2 - 4z^2\right)}{1024\Delta x^2}
\end{array}
$$

C.2 Grid to Particle

From grid to particle, we wish to find $\mathbf{c}$ such that

$$\sum_{t=1}^{N_r} \mathcal{S}_{pr}^n \cdot \left(\mathbf{m}_p^n \mathcal{S}_{pt}^n\right) c_{pt\alpha}^{n+1} = \mathcal{Q}_{pr\alpha}^n \cdot \left(\mathbf{M}_p^n \hat{\mathcal{V}}_p^{n+1}\right)$$
$$= \sum_{k=1}^{(N_B+1)^d} m_{\mathbf{i}_{kp}^n p}^n s_r(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}.$$

The bases we choose satisfy the property that $\mathcal{S}_{pr}^n \cdot \left(\mathbf{m}_p^n \mathcal{S}_{pt}^n\right) = 0$ for $r \neq t$. So we have

$$\mathcal{S}_{pr}^n \cdot \left(\mathbf{m}_p^n \mathcal{S}_{pr}^n\right) c_{pr\alpha}^{n+1} = \mathcal{Q}_{pr\alpha}^n \cdot \left(\mathbf{M}_p^n \hat{\mathcal{V}}_p^{n+1}\right)$$
$$= \sum_{k=1}^{(N_B+1)^d} m_{\mathbf{i}_{kp}^n p}^n s_r(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}.$$

For linear interpolation, the grid to particle transfer is similar to APIC. For quadratic interpolation, $\mathcal{S}_{pr}^n \cdot \left(\mathbf{m}_p^n \mathcal{S}_{pr}^n\right)$ can be zero for some r when $x_{p\alpha}^n = \pm\frac{h}{2}, \pm\frac{\sqrt{3}}{2}h$. However, we can still find a meaningful expression for $\mathbf{c}$.

In 2D, $m_{\mathbf{i}_{kp}^n p}^n = m_p N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) = m_p N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)$, $\mathbf{c}$ can be computed from

the formula below:

$$
\begin{aligned}
c_{p1\alpha} &= \sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1} \\
c_{p2\alpha} &= \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^2}{4}} \\
c_{p3\alpha} &= \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp2}^n} - x_{p2})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^2}{4}} \\
c_{p4\alpha} &= \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(x_{\mathbf{i}_{kp2}^n} - x_{p2})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^4}{16}} \\
c_{p5\alpha} &= \frac{\sum_{k=1}^{(N_B+1)^d} N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp1}-1) \bmod 2}\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{2\Delta x^2} \\
c_{p6\alpha} &= \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp2}-1) \bmod 2}\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{2\Delta x^2} \\
c_{p7\alpha} &= \frac{2\sum_{k=1}^{(N_B+1)^d}(x_{\mathbf{i}_{kp1}} - x_{p1})N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp2}-1) \bmod 2}\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\Delta x^4} \\
c_{p8\alpha} &= \frac{2\sum_{k=1}^{(N_B+1)^d}(x_{\mathbf{i}_{kp2}} - x_{p2})N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp1}-1) \bmod 2}\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\Delta x^4} \\
c_{p9\alpha} &= \sum_{k=1}^{(N_B+1)^d} \frac{1}{4\Delta x^4}(-2)^{(i_{kp1}-1) \bmod 2}(-2)^{(i_{kp2}-1) \bmod 2}\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}
\end{aligned}
$$

In 3D, $m_{\mathbf{i}_{kp}^n p}^n = m_p N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) = m_p N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)$, $\mathbf{c}$ can be

computed from the formula below:

$$c_{p1\alpha} = \sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}$$

$$c_{p2\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^2}{4}}$$

$$c_{p3\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp2}^n} - x_{p2})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^2}{4}}$$

$$c_{p4\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp3}^n} - x_{p3})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^2}{4}}$$

$$c_{p5\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(x_{\mathbf{i}_{kp2}^n} - x_{p2})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^4}{16}}$$

$$c_{p6\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp2}^n} - x_{p2})(x_{\mathbf{i}_{kp3}^n} - x_{p3})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^4}{16}}$$

$$c_{p7\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(x_{\mathbf{i}_{kp3}^n} - x_{p3})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^4}{16}}$$

$$c_{p8\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(x_{\mathbf{i}_{kp2}^n} - x_{p2})(x_{\mathbf{i}_{kp3}^n} - x_{p3})\hat{v}_{\mathbf{i}_{kp}^n\alpha}^{n+1}}{\frac{\Delta x^6}{64}}$$

$$c_{p9\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp1}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{2\Delta x^2}$$

$$c_{p10\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp2}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{2\Delta x^2}$$

$$c_{p11\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{2\Delta x^2}$$

$$c_{p12\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp1}-1) \bmod 2}(-2)^{(i_{kp2}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{4\Delta x^4}$$

$$c_{p13\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp1}-1) \bmod 2}(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{4\Delta x^4}$$

$$c_{p14\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(-2)^{(i_{kp2}-1) \bmod 2}(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{4\Delta x^4}$$

$$c_{p15\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} (-2)^{(i_{kp1}-1) \bmod 2}(-2)^{(i_{kp2}-1) \bmod 2}(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{8\Delta x^6}$$

$$c_{p16\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp2}^n} - x_{p2})(-2)^{(i_{kp1}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^4}{2}}$$

$$c_{p17\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp3}^n} - x_{p3})(-2)^{(i_{kp1}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^4}{2}}$$

$$c_{p18\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp2}^n} - x_{p2})(x_{\mathbf{i}_{kp3}^n} - x_{p3})(-2)^{(i_{kp1}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^6}{8}}$$

$$c_{p19\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(-2)^{(i_{kp2}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^4}{2}}$$

$$c_{p20\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp3}^n} - x_{p3})(-2)^{(i_{kp2}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^4}{2}}$$

$$c_{p21\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(x_{\mathbf{i}_{kp3}^n} - x_{p3})(-2)^{(i_{kp2}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^6}{8}}$$

$$c_{p22\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^4}{2}}$$

$$c_{p23\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp2}^n} - x_{p2})(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^4}{2}}$$

$$c_{p24\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n) N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(x_{\mathbf{i}_{kp2}^n} - x_{p2})(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\frac{\Delta x^6}{8}}$$

$$c_{p25\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_3(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp3}^n} - x_{p3})(-2)^{(i_{kp1}-1) \bmod 2}(-2)^{(i_{kp2}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\Delta x^6}$$

$$c_{p26\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_2(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp2}^n} - x_{p2})(-2)^{(i_{kp1}-1) \bmod 2}(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\Delta x^6}$$

$$c_{p27\alpha} = \frac{\sum_{k=1}^{(N_B+1)^d} N_1(\mathbf{x}_{\mathbf{i}_{kp}^n} - \mathbf{x}_p^n)(x_{\mathbf{i}_{kp1}^n} - x_{p1})(-2)^{(i_{kp2}-1) \bmod 2}(-2)^{(i_{kp3}-1) \bmod 2} \hat{v}_{\mathbf{i}_{kp}^n \alpha}^{n+1}}{\Delta x^6}$$

C.3 PolyPIC is lossless

In this section, we prove that if we use PolyPIC to transfer grid momentum to particle and then directly transfer back without advecting, we get the exact same grid momentum back. For simplicity, we prove it for the one particle case.

Given grid mass $\mathbf{M}_p^n$ and grid velocity $\mathcal{V}_p^n$, the $\mathbf{c}_p^{n+1}$ we find by using full-interpolation PolyPIC is given by

$$\mathbf{c}_p^{n+1} = (\mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathbf{Q}_p^n)^{-1} \mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathcal{V}_p^n.$$

We want to show that the momentum $\hat{\mathbf{p}}$ we get from $\hat{\mathbf{p}} = \mathbf{M}_p^n \mathbf{Q}_p^n \mathbf{c}_p^{n+1}$ is equal to the original momentum on the grid $\mathbf{M}_p^n \mathbf{v}_i$. The key obversation is that $\mathbf{Q}_p^n$ is invertible: $(\mathbf{Q}_p^n)^T \mathbf{M}_p^n \mathbf{Q}_p^n$ is

full rank diagonal, which means that $\mathbf{Q}_p^n$ is also full rank and therefore invertible.

$$\begin{aligned}
\hat{\mathbf{p}} &= \mathbf{M}_p^n \mathbf{Q}_p^n \mathbf{c}_p^{n+1} \\
&= (\mathbf{Q}_p^{n-T} \mathbf{Q}_p^{nT}) \mathbf{M}_p^n \mathbf{Q}_p^n \mathbf{c}_p^{n+1} \\
&= (\mathbf{Q}_p^{n-T} \mathbf{Q}_p^{nT}) \mathbf{M}_p^n \mathbf{Q}_p^n (\mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathbf{Q}_p^n)^{-1} \mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathcal{V}_p^n \\
&= (\mathbf{Q}_p^n)^{-T} (\mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathbf{Q}_p^n)(\mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathbf{Q}_p^n)^{-1} \mathbf{Q}_p^{nT} \mathbf{M}_p^n \mathcal{V}_p^n \\
&= \mathbf{M}_p^n \mathcal{V}_p^n
\end{aligned}$$

C.4 PolyPIC is linear and angular momentum conserving

The ith component of the local linear momentum associated with velocity $\mathcal{U}$ is $\mathbf{C}_i^T (\mathbf{Q}_p^n)^T \mathbf{M}_p^n \mathcal{U}$ and the j^{th} component of the angular velocity is $\mathbf{C}_{j+d}^T (\mathbf{Q}_p^n)^T \mathbf{M}_p^n \mathcal{U}$ where $C_{ki} = \delta_{ki}$ for $1 \leq k \leq d$ and

$$C_{k(j+d)} = \begin{cases} 1, & k = 4 \\ -1, & k = 5 \end{cases}$$

when $d = 2$ and

$$C_{k4} = \begin{cases} 1, & k = 5 \\ -1, & k = 7 \end{cases}$$

$$C_{k5} = \begin{cases} 1, & k = 6 \\ -1, & k = 10 \end{cases}$$

$$C_{k6} = \begin{cases} 1, & k = 9 \\ -1, & k = 11 \end{cases}$$

Thus local linear and angular momentum conservation (which implies global) follows from

$$\mathbf{C}_i^T (\mathbf{Q}_p^n)^T \mathbf{M}_p^n \mathcal{V}_p^{n+1} = \mathbf{C}_i^T (\mathbf{Q}_p^n)^T \mathbf{M}_p^n \mathbf{Q}_p^n \mathbf{c}_p^{n+1}$$

C.5 Mathematica code

In this section we present the mathematica code we use to generate the bases and corresponding formula.

C.5.1 Linear interpolation in 2d

```
(* 2D linear *)
ClearAll[ "Global*"]
(* Assume x is in [0, h],
N[x, 1] is the weight of node at xi = 0,
N[x, 2] is the weight of node at h.
*)
N1[ x_, i_] = Piecewise[{{1 - x, i == 1}, {x, i == 2}}];
NN[ x_, y_, ii_, jj_] = N1[x/h, ii] * N1[y/h, jj];
(* xi - xp *)
r = ConstantArray[0, {4, 2}];
Do[{id = 2 * ( i1 - 1) + j1;
nodex = ( i1 - 1) * h;
nodey = ( j1 - 1) * h;
r[[id]][[1]] = (nodex - particlex)/h;
r[[id]][[2]] = (nodey - particley)/h; }, { i1, 1, 2}, { j1, 1, 2}];
(* mass *)
M = ConstantArray[0, {4, 4}];
Do[{id = 2 * ( i1 - 1) + j1;
weight = NN[ particlex, particley, i1, j1];
M[[id]][[id]] = mass * weight; }, { i1, 1, 2}, { j1, 1, 2}];
(* basis *)
(* row index is node index
 col index is basis index
```

```
 *)
 Do[{id = 2 * ( i1 - 1) +  j1;
 weight =  NN[ particlex,  particley,  i1,  j1];
M[[id]][[id]] =  mass *  weight; }, { i1, 1, 2}, { j1, 1, 2}];
 B1 = ConstantArray[0, {4, 4}];
 Do[{id = ( i1 - 1) * 2 +  j1;
 B1[[idr]][[id]]  = r[[idr]][[1]]^( i1 - 1) * r[[idr]][[2]]^( j1 - 1)}, { i1, 1, 2}, { j1, 1, 2}, {idr, 1, 4}];
 Diagonal[M] // MatrixForm;
 (* Verify the diagonal structure *)
 MatrixForm[ Transpose[ B1].M. B1] // Simplify
```

C.5.2 Linear interpolation in 3d

```
 (* 3D linear *)
 ClearAll[ "Global*"]
 (* Assume x is in[0, h],
 N[x, 1] is the weight of node at xi = 0,
 N[x, 2] is the weight of node ath.
 *)
 N1[ x_,  i_] = Piecewise[{{1 - x, i == 1}, {x, i == 2}}];
 NN[ x_,  y_,  z_,  ii_,  jj_,  kk_] =  N1[x/h,  ii] *  N1[y/h,  jj] *  N1[z/h,  kk];
 (* xi - xp *)
r = ConstantArray[0, {8, 3}];
 Do[{id = 4 * ( i1 - 1) + 2 * ( j1 - 1) +  k1;
 nodex = ( i1 - 1) * h;
 nodey = ( j1 - 1) * h;
 nodez = ( k1 - 1) * h;
r[[id]][[1]] = ( nodex -  particlex)/h;
r[[id]][[2]] = ( nodey -  particley)/h;
r[[id]][[3]] = ( nodez -  particlez)/h; }, { i1, 1, 2}, { j1, 1, 2}, { k1, 1, 2}];
```

```
 (* mass *)
M = ConstantArray[0, {8, 8}];
 Do[{id = 4 * ( i1 - 1) + 2 * ( j1 - 1) +  k1;
 weight =  NN[ particlex,  particley,  particlez,  i1,  j1,  k1];
M[[id]][[id]] =  mass *  weight; }, { i1, 1, 2}, { j1, 1, 2}, { k1, 1, 2}];
 (* basis *)
 (* row index is node index
  col index is basis index
 *)
 B1 =  ConstantArray[0, {4 * 2, 8}];
 Do[{id = ( i1 - 1) * 4 + ( j1 - 1) * 2 +  k1;
 B1[[idr]][[id]]  = r[[idr]][[1]]^( i1 - 1) * r[[idr]][[2]]^( j1 - 1) * r[[idr]][[3]]^( k1 - 1); },
     {i1, 1, 2}, { j1, 1, 2}, { k1, 1, 2}, {idr, 1, 8}];
 Diagonal[M] // MatrixForm;
 (* Verify the diagonal structure *)
 MatrixForm[ Transpose[ B1].M. B1] // Simplify
```

C.5.3 Quadratic interpolation in 2d

```
(* 2D quadratic *)
ClearAll["Global*"]
(* Assume x is in[-0.5h, 0.5h], we use quadratic interpolation.
N2[x, 1]is the weight of node at xi = -h,
 N2[x, 2]is the weight of node at xi = 0,
 N2[x, 3]is the weight of node at xi = h,
*)
N2[x_, i_] = Piecewise[{{1/2 * (1/2 - x)^2, i == 1}, {3/4 - x^2, i == 2},
{1/2 * (x + 1/2)^2, i == 3}}];
NN[x_, y_, ii_, jj_] = N2[x/h, ii] * N2[y/h, jj];
```

```
(* xi - xp *)
r = ConstantArray[0, {9, 2}];
Do[{id = 3 * (i1 - 1) + j1;
nodex = (i1 - 2) * h;
nodey = (j1 - 2) * h;
r[[id]][[1]] = (nodex - x);
r[[id]][[2]] = (nodey - y); }, {i1, 1, 3}, {j1, 1, 3}];
(* mass *)
M = ConstantArray[0, {9, 9}];
Do[{id = 3 * (i1 - 1) + j1;
weight = NN[x, y, i1, j1];
M[[id]][[id]] = mass * weight; }, {i1, 1, 3}, {j1, 1, 3}];
(* basis *)
(* row index is node index
 col index is basis index
*)
B = ConstantArray[0, {9, 9}];
Do[{id = (j1 - 1) * 3 + i1;
B[[idr]][[id]]  = r[[idr]][[1]]^(i1 - 1) * r[[idr]][[2]]^(j1 - 1);
}, {i1, 1, 3}, {j1, 1, 3}, {idr, 1, 9}];
(* Rearrange the basis so that the corresponding polynomials are in the order of
 1, x, y, xy, x^2, y^2, x^2 * y, x * y^2, x^2 * y^2.
*)
BS = B[[All, {1, 2, 4, 5, 3, 7, 6, 8, 9}]];
(* The first four basis vectors are already othogonal.
Use Gram-Schmidt and we get the basis vectors corresponding to x^2 and y^2.
*)
BS[[All, 5]] = BS[[All, 5]] - h^2/4 - x(h^2 - 4x^2)/h^2 * BS[[All, 2]];
BS[[All, 6]] = BS[[All, 6]] - h^2/4 - y(h^2 - 4y^2)/h^2 * BS[[All, 3]];
```

```
(* The rest are simply products of the previous ones *)
BS[[All, 7]] = BS[[All, 5]]BS[[All, 3]];
BS[[All, 8]] = BS[[All, 6]]BS[[All, 2]];
BS[[All, 9]] = BS[[All, 5]] BS[[All, 6]];
(* Verify the diagonal structure *)
BTMB = Transpose[BS].M.BS//Simplify;
(* Get the awesome formula to put in your code!*)
MB = M.BS//Simplify;
MB//MatrixForm;
Inverse[BTMB].Transpose[BS].M//MatrixForm//Simplify;
```

C.5.4 Quadratic interpolation in 3d

```
 (* 3D quadratic *)
 ClearAll[ "Global*"]
 (* Assume x is in[-0.5h, 0.5h], we use quadratic interpolation.
 N2[x, 1] is the weight of node at xi = -h,
  N2[x, 2] is the weight of node at xi = 0,
  N2[x, 3] is the weight of node at xi = h,
 *)
 N2[ x_, i_] = Piecewise[{{1/2 * (1/2 - x)^2, i == 1}, {3/4 - x^2, i == 2},
{1/2 * (x + 1/2)^2, i == 3}}];
 NN[ x_, y_, z_, ii_, jj_, kk_] = N2[x/h, ii] * N2[y/h, jj] * N2[z/h, kk];
 (* xi - xp *)
r = ConstantArray[0, {27, 3}];
 Do[{id = 9 * ( i1 - 1) + 3 * ( j1 - 1) + k1;
 nodex = ( i1 - 2) * h;
 nodey = ( j1 - 2) * h;
 nodez = ( k1 - 2) * h;
r[[id]][[1]] = ( nodex - x);
```

```
r[[id]][[2]] = ( nodey - y);
r[[id]][[3]] = ( nodez - z); }, { i1, 1, 3}, { j1, 1, 3}, { k1, 1, 3}];
 (* mass *)
M =  ConstantArray[0, {27, 27}];
 Do[{id = 9 * ( i1 - 1) + 3 * ( j1 - 1) +  k1;
 weight =  NN[x, y, z,  i1,  j1,  k1];
M[[id]][[id]] =  mass *  weight; }, { i1, 1, 3}, { j1, 1, 3}, { k1, 1, 3}];
 (* basis *)
 (* row index is node index
  col index is basis index
 *)
B =  ConstantArray[0, {27, 27}];
 Do[{id = ( i1 - 1) * 4 + ( j1 - 1) * 2 +  k1;
B[[idr]][[id]]  = r[[idr]][[1]]^( i1 - 1) * r[[idr]][[2]]^( j1 - 1) * r[[idr]][[3]]^( k1 - 1);
}, { i1, 1, 2}, { j1, 1, 2}, { k1, 1, 2}, {idr, 1, 27}];
 (* The basisvectors  corresponding to1, x, y, z are already orthogonal.
 Use gram -  schmidt and we get the basis corresponding to x^2, y^2,  andz^2.
 *)
B[[All, 9]] = B[[All, 5]]B[[All, 5]] - h^2/4 - x(h^2 - 4x^2)/h^2 * B[[All, 5]];
B[[All, 10]] = B[[All, 3]]B[[All, 3]] - h^2/4 - y(h^2 - 4y^2)/h^2 * B[[All, 3]];
B[[All, 11]] = B[[All, 2]]B[[All, 2]] - h^2/4 - z(h^2 - 4z^2)/h^2 * B[[All, 2]];

 (* The rest are simply products of the previous ones *)
B[[All, 12]] = B[[All, 2]]B[[All, 9]];
B[[All, 13]] = B[[All, 3]]B[[All, 9]];
B[[All, 14]] = B[[All, 2]]B[[All, 3]]B[[All, 9]];

B[[All, 15]] = B[[All, 2]]B[[All, 10]];
B[[All, 16]] = B[[All, 5]]B[[All, 10]];
```

```
B[[All, 17]] = B[[All, 2]]B[[All, 5]]B[[All, 10]];

B[[All, 18]] = B[[All, 3]]B[[All, 11]];
B[[All, 19]] = B[[All, 5]]B[[All, 11]];
B[[All, 20]] = B[[All, 3]]B[[All, 5]]B[[All, 11]];

B[[All, 21]] = B[[All, 9]]B[[All, 10]];
B[[All, 22]] = B[[All, 10]]B[[All, 11]];
B[[All, 23]] = B[[All, 9]]B[[All, 11]];

B[[All, 24]] = B[[All, 21]]B[[All, 2]];
B[[All, 25]] = B[[All, 22]]B[[All, 5]];
B[[All, 26]] = B[[All, 23]]B[[All, 3]];

B[[All, 27]] = B[[All, 9]]B[[All, 10]]B[[All, 11]];

(* Rearrange the basis so that the corresponding polynomials are in the order of
1, x, y, z, xy, xz, yz, xyz, x^2, y^2, z^2, x^2 * y^2, x^2 * z^2, y^2 * z^2, x^2 * y^2 * z^2,
x^2 * y, x^2 * z, x^2 * yz, y^2 * x, y^2 * z, y^2 * xz, z^2 * x, z^2 * y, z^2 * xy,
x^2 * y^2 * z, x^2 * z^2 * y, y^2 * z^2 * x
*)
BS = B[[All, {1, 5, 3, 2, 7, 6, 4, 8, 9, 10, 11, 21, 23, 22, 27, 13, 12, 14, 16,
15, 17, 19, 18, 20, 24, 26, 25}]];

(* Verify the diagonal structure *)
BTMB = Transpose[ BS].M. BS // Simplify;
BTMB // MatrixForm
(* Get the awesome formula to put in your code! *)
MB = M. BS // Simplify;
```

```
MB // MatrixForm;
BTMBinvBTM = Inverse[ BTMB]. Transpose[ BS].M;
Transpose[ BTMBinvBTM] // MatrixForm // Simplify;
```

REFERENCES

[1] R. Ando, N. Thuerey, and C. Wojtan. A stream function solver for liquid simulations. *ACM Trans Graph*, 34 (2):8, August 2015.

[2] R. Ando, N. Thurey, and R. Tsuruno. Preserving fluid sheets with adaptively sampled anisotropic particles. *IEEE Trans Vis Comp Graph*, 18(8):1202–1214, August 2012.

[3] R. Ando, N. Thurey, and C. Wojtan. Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans Graph*, 32(4):103:1–103:10, 2013.

[4] R. Ando and R. Tsuruno. A particle-based method for preserving fluid sheets. In *Proc ACM SIGGRAPH/Eurographics Symp Comp Anim*, SCA '11, pages 7–16, 2011.

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

[6] J. Brackbill. The ringing instability in particle-in-cell calculations of low-speed flow. *J Comp Phys*, 75(2):469–492, 1988.

[7] J. Brackbill and H. Ruppel. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J Comp Phys*, 65:314–343, 1986.

[8] Edwin Catmull and James Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978.

[9] N. Chentanez and M. Muller. Coupling 3d eulerian, height field and particle methods for the simulation of large scale liquid phenomena. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim*, SCA '14, 2014.

[10] 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, pages 17:1–17:11, New York, NY, USA, 2017. ACM.

[11] M. Desbrun and M. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In R. Boulic and G. Hegron, editors, *Eurographics Workshop on Computer Animation and Simulation (EGCAS)*, pages 61–76. Springer-Verlag, 1996.

[12] E. Edwards and R. Bridson. A high-order accurate particle-in-cell method. *Int J Numer Meth Eng*, 90:1073–1088, 2012.

[13] F. Ferstl, R. Ando, C. Wojtan, R. Westermann, and N. Thuerey. Narrow band flip for liquid simulations. *Comp Graph For*, 35(2):8, May 2016.

[14] C. Fu, Q. Guo, T. Gast, C. Jiang, and J. Teran. *Supplementary Technical Document*, 2017.

[15] Y. Gao, C. Li, S. Hu, and B. Barsky. Simulating gaseous fluids with low and high speeds. *Comp Graph Forum*, 28(28):1845–1852, 2009.

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

[17] C. Gritton and M. Berzins. Improving accuracy in the mpm method using a null space filter. *Comp Part Mech*, 4(1):131–142, 2017.

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

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

[20] F. Harlow and E. Welch. Numerical calculation of time dependent viscous flow of fluid with a free surface. *Phys Fluid*, 8(12):2182–2189, 1965.

[21] W. Hong, D. House, and J. Keyser. Adaptive particles for incompressible fluid simulation. *Vis Comp*, 24(7):535–543, 2008.

[22] W. Hong, D. House, and J. Keyser. An adaptive sampling approach to incompressible particle-based fluid. *Theory Pract Comp Graph*, pages 69–76, 2009.

[23] C. Jiang, T. Gast, and J. Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans Graph*, 36(4), 2017.

[24] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. The affine particle-in-cell method. *ACM Trans Graph*, 34(4):51:1–51:10, 2015.

[25] C. Jiang, C. Schroeder, and J. Teran. An angular momentum conserving affine-particle-in-cell method. *J Comp Phys*, 338:137 – 164, 2017.

[26] H. Lee, J. Hong, and C. Kim. Interchangeable SPH and level set method in multiphase fluids. *Vis Comp*, 25(5):713–718, 2009.

[27] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw. Two-way coupled SPH and particle level set fluid simulation. *IEEE Trans Vis Comp Graph*, 14:797–804, 2008.

[28] C. Mast. *Modeling landslide-induced flow interactions with structures using the Material Point Method*. PhD thesis, 2013.

[29] O. Mercier, C. Beauchemin, N. Thuerey, T. Kim, and D. Nowrouzezahrai. Surface Turbulence for Particle-Based Liquid Simulations. *ACM Trans Graph*, 34(6):10, Nov 2015.

[30] K. Raveendran, C. Wojtan, and G. Turk. Hybrid SPH. In *Proc 2011 ACM SIGGRAPH/Eurograp Symp Comp Anim*, SCA '11, pages 33–42, 2011.

[31] J. Simo and G. Meschke. A new class of algorithms for classical plasticity extended to finite strains. application to geomaterials. *Comput Mech*, 11(4):253–278, 1993.

[32] 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*, pages 395–404. ACM, 1998.

[33] M. Steffen, R. Kirby, and M. Berzins. Analysis and reduction of quadrature errors in the material point method (MPM). *Int J Numer Meth Eng*, 76(6):922–948, 2008.

[34] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. A material point method for snow simulation. *ACM Trans Graph*, 32(4):102:1–102:10, 2013.

[35] D. Sulsky, Z. Chen, and H. Schreyer. A particle method for history-dependent materials. *Comp Meth App Mech Eng*, 118(1):179–196, 1994.

[36] D. Sulsky, S. Zhou, and H. Schreyer. Application of a particle-in-cell method to solid mechanics. *Comp Phys Comm*, 87(1):236–252, 1995.

[37] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: viscolelasticity, plasticity, fracture. *SIGGRAPH Comp Graph*, 22(4):269–278, 1988.

[38] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Comput Graph*, 21(4):205–214, 1987.

[39] N. Thuerey and T. Pfaff. MantaFlow, 2016. *http://mantaflow.com*.

[40] K. Um, S. Baek, and J. Han. Advanced hybrid particle-grid method with sub-grid particle correction. *Comp Graph Forum*, 33:209–218, 2014.

[41] P. Wallstedt and J. Guilkey. Improved velocity projection for the material point method. *Comp Mod in Eng and Sci*, 19(3):223, 2007.

[42] Naoto Yoshioka. A sandpile experiment and its implications for self-organized criticality and characteristic earthquake. *Earth, planets and space*, 55(6):283–289, 2003.

[43] B. Zhu, X. Yang, and Y. Fan. Creating and preserving vortical details in sph fluid. *Comp Graph Forum*, 29(7):2207–2214, 2010.