Peer Reviewed

Title:
A Material Point Method for Complex Fluids

Author:
Ram, Daniel

Acceptance Date:
2015

Series:
UCLA Electronic Theses and Dissertations

Degree:
Ph.D., Mathematics 0540UCLA

Advisor(s):
Teran, Joseph M

Committee:
Anderson, Chris, Eldredge, Jeff D, Laub, Alan J

Permalink:
http://escholarship.org/uc/item/87d1h4dn

Abstract:

Copyright Information:

UNIVERSITY OF CALIFORNIA

Los Angeles

A Material Point Method for Complex Fluids

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

Daniel Ram

2015

ABSTRACT OF THE DISSERTATION

A Material Point Method for Complex Fluids

by

Daniel Ram

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2015

Professor Joseph M. Teran, Chair

We present a novel Material Point Method for simulating complex materials. The method achieves plasticity effects via the temporal evolution of the left elastic Cauchy-Green strain. We recast the upper-convected derivative of the strain in the Oldroyd-B constitutive model as a plastic flow and are able to simulate elastic and viscoelastic effects. Our model provides a volume-preserving rate-based description of plasticity that does not require singular value decompositions. Our semi-implicit discretization allows for high-resolution simulations. We also present novel discretizations of the temporal update of the left elastic Cauchy-Green strain for several constitutive models that preserve symmetry and positive-definiteness of the strain for use in the Material Point Method. A novel modification to a constitutive model is also presented that models material softening under plastic compression.

The dissertation of Daniel Ram is approved.

Christopher Anderson

Jeffrey D. Eldredge

Alan J. Laub

Joseph M. Teran, Committee Chair

University of California, Los Angeles

2015

To my wife

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I thank Professor Joseph Teran for the years of wonderful guidance he has provided. Of course, this would not have been possible without him. His dedication to his students is truly exceptional.

I also deeply thank my co-authors, Theodore Gast, Chenfanfu Jiang, Craig Schroeder, Alexey Stomakhin, and Pirouz Kavehpour.

I particularly thank Chenfanfu Jiang for always looking out for me and for teaching me so much about doing good research.

I also thank Andre Pradhana for his constant support.

My dear wife, Xiao Zhang, is beyond lovely and I am so happy to begin a new chapter in our life.

My family, Tamir, Bilha, Dror, Daphna, Adi, Yuval, and Nadav, I thank you for the support and love you have given me all of my life.

VITA

2008   B.A. (Mathematics), University of California, Berkeley.

2009–2014   Teaching Assistant, University of California, Los Angeles.

2011   M.A. (Mathematics), University of California, Los Angeles.

2014–2015   Research Assistant, University of California, Los Angeles.

PUBLICATIONS

D. Ram, T. Gast, C. Jiang, C. Schroeder, A. Stomakhin, J. Teran, P. Kavehpour, A Material Point Method for Viscoelastic Fluids, Foams and Sponges, *ACM SIGGRAPH/ Eurographics Symposium on Computer Animation (SCA)*, 2015.

K. Chong, C. Jiang, D. Ram, A. Santhanam, D. Terzopoulos, P. Benharash, J. Teran, J. Eldredge, Visualization of Vascular Injuries in Extremity Trauma, *In preparation.*

CHAPTER 1

Introduction

Simulating complex materials is challenging because of the wide range of behaviors exhibited and the nonlinear governing equations. These materials may display elastic effects, such as resistance to deformation, but can also undergo permanent deformation. Lagrangian methods typically resolve elastic effects well by storing deformation information on a mesh. On the other hand, Eulerian methods perform stress-based computations on a background grid, and can resolve large topology change and permanent deformation well.

The Material Point Method is a hybrid method that stores deformation information on particles and performs stress-based computations on the grid. Because of this, it is able to resolve elastic effects, large topology change, and self-collision. Wide-ranging plasticity effects are achieved by tracking strain on particles.

1.1 Contributions

This dissertation details a novel Material Point Method for simulating non-Newtonian materials, which was originally developed in [Ram et al., 2015]. The key contributions of this method are:

- A volume-preserving rate-based description of plasticity

- Recasting parts of the traditional Oldroyd-B viscoelasticity model as a plastic flow rate

- Semi-implicit discretization that allows for high resolution simulations

- A plasticity model that does not require the use of singular value decomposition

Furthermore, this dissertation proposes novel discretizations of commonly used plastic flows that preserve important physical properties:

- Simple first-order discretizations of Oldroyd-B and Herschel-Bulkley-like plastic flows that preserve positive-definiteness and symmetry of the left elastic Cauchy Green strain

The dissertation also provides a novel modification to an established constitutive model to model material softening under plastic compression. This may be used in conjunction with a model for materials that soften under plastic extension to convincingly simulate a variety of materials.

1.2   Dissertation Overview

We provide the necessary background in continuum mechanics for the rest of the dissertation in Chapter 2. The governing equations of motion are derived and the relevant stress tensors for our treatment of elasticity and plasticity are defined. Net working and hyperelasticity are also discussed in this chapter. Chapter 3 details the Material Point Method for simulating fully elastic materials. Chapter 4 covers the necessary plasticity theory for understanding the novel Material Point Method in Chapter 5 that handles plasticity. Chapter 4 also details novel simple discretizations of plastic flow that preserve important physical properties of strain, as well a novel modification to an established constitutive model that can model material softening. Chapter 6 discusses po-

tential extensions of the established results. In that chapter, a high-order discretization of the left-elastic Cauchy-Green strain that preserves symmetry and positive-definitess of the strain is presented may potentially be used in a high-order solver for the incompressible Oldroyd-B equations.

The appendix provides a thorough derivation of the energy derivatives necessary for computing the stress-based grid forces for the method detailed in Chapter 5. In order to aid implementation, the appendix also provides pseudocode that calculates these energy derivatives.

CHAPTER 2

Continuum Mechanics Background

This chapter introduces deformation and the deformation gradient, and details the conservation of mass and momentum. These principles will be essential to our treatment of modeling non-Newtonian fluids. As it is crucial in our treatment of plasticity, net working and hyperelasticity is explained in detail. Several examples of hyperelastic constitutive models are provided. Furthermore, a description of the right Cauchy-Green strain is presented.

2.1 Deformation Gradient and the Material Derivative

Let $\Omega, \Omega_t \subset \mathbb{R}^3$. We assume that there is an invertible and sufficiently smooth deformation map $\phi : (\Omega, \mathbb{R}) \to \Omega_t := \phi(\Omega, t) = \{\mathbf{y} | \exists \mathbf{X} \in \Omega \ \mathbf{y} = \phi(\mathbf{X}, t)\}$ that describes the deformation of the material. We define the deformation gradient as

$$\boldsymbol{F} := \frac{\partial \phi}{\partial \mathbf{X}}.$$

Note that given a point $\mathbf{X} \in \Omega$, $\phi(\mathbf{Y}, t) = \phi(\mathbf{X}, t) + \boldsymbol{F}(\mathbf{X}, t)(\mathbf{Y} - \mathbf{X})$ is a linearization of ϕ for all $\mathbf{Y}$ in a small neighborhood of $\mathbf{X}$.

We call $\Omega_0 = \Omega$ the reference configuration and Ω_t for $t > 0$ the material configuration. For points in the material configuration Ω_t, we use the shorthand $\boldsymbol{x} := \phi(\mathbf{X}, t)$. Any function $\mathbf{Q}$ defined on the reference configuration has a corresponding push-forward in

4
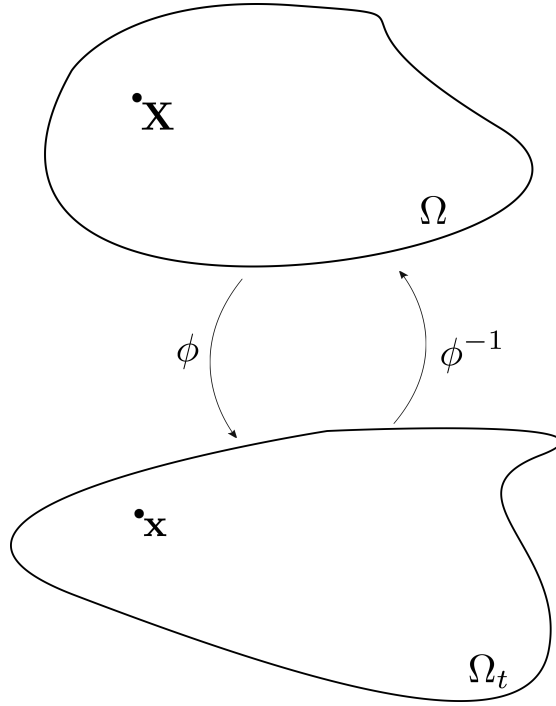
Figure 2.1: *The deformation map ϕ. Here, Ω is the reference configuration and $\Omega_t = \phi(\Omega, t)$. $\mathbf{X}$ is a point in the reference configuration that gets mapped to $\boldsymbol{x} = \phi(\mathbf{X}, t)$ in the material configuration. ϕ^{-1} is the deformation defined by $\phi^{-1}(\boldsymbol{x}, t) = \mathbf{X}$ such that $\phi(\mathbf{X}, t) = \boldsymbol{x}$.*

the material configuration, defined as

$$\mathbf{q}(\boldsymbol{x}, t) \coloneqq \mathbf{Q}(\phi^{-1}(\boldsymbol{x}, t), t)$$

for any $\boldsymbol{x} \in \Omega_t$. This implies that $\mathbf{q}(\phi(\mathbf{X}, t), t) = \mathbf{Q}(\mathbf{X}, t)$ for all $\mathbf{X} \in \Omega_0, t \in \mathbb{R}$. We use the convention that functions in the reference configuration are denoted with capital letters while functions in the material configuration are denoted with lowercase letters.

Given a deformation ϕ, a material $\mathbf{Q}$ with push-forward $\mathbf{q}$, and defining

$$\mathbf{V}(\mathbf{X}, t) \coloneqq \frac{\partial \phi}{\partial t}(\mathbf{X}, t)$$

5
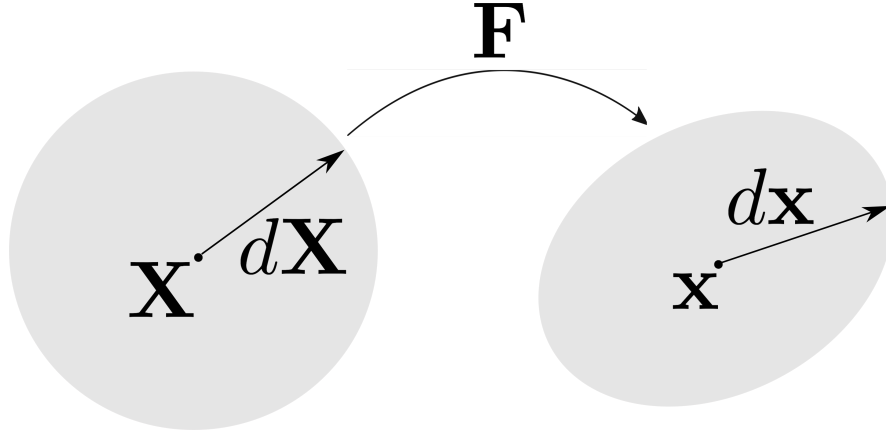
Figure 2.2: *The deformation gradient $\boldsymbol{F}$. At any point $\mathbf{X} \in \Omega, \boldsymbol{F}(\mathbf{X}, t)$ is a first-order approximation to the deformation map ϕ for all $\mathbf{Y}$ close to $\mathbf{X}$. We may write $\boldsymbol{F} = \frac{\partial \phi}{\partial \mathbf{X}} = \frac{\partial \boldsymbol{x}}{\partial \mathbf{X}}$. Here, a small neighborhood of $\mathbf{X}$ with radius $d\mathbf{X}$ is transformed.*

with corresponding push-forward $\boldsymbol{v}(\phi(\mathbf{X}, t), t) := \mathbf{V}(\mathbf{X}, t)$, note that $\partial_t(\mathbf{Q}) \neq \partial_t(\mathbf{q})$:

$$\partial_t(\mathbf{q}) = \frac{\partial \mathbf{q}}{\partial \boldsymbol{x}}(\phi(\mathbf{X}, t), t)\frac{\partial \phi}{\partial t}(\mathbf{X}, t) + \frac{\partial \mathbf{q}}{\partial t}(\phi(\mathbf{X}, t), t)$$
$$= \frac{\partial \mathbf{q}}{\partial \boldsymbol{x}}(\boldsymbol{x}, t)\mathbf{V}(\mathbf{X}, t) + \frac{\partial \mathbf{q}}{\partial t}(\phi(\mathbf{X}, t), t)$$
$$= \frac{\partial \mathbf{q}}{\partial \boldsymbol{x}}(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t) + \frac{\partial \mathbf{q}}{\partial t}(\boldsymbol{x}, t).$$

If we define the acceleration as

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

with corresponding push-forward $\mathbf{a}(\boldsymbol{x}, t) := \mathbf{A}(\mathbf{X}, t)$, we immediately observe that $\mathbf{a}(\boldsymbol{x}, t) \neq \frac{\partial \boldsymbol{v}}{\partial t}(\boldsymbol{x}, t)$.

We define the material derivative of a function defined in the material configuration as

$$\frac{D\mathbf{q}}{Dt} := \frac{\partial \mathbf{q}}{\partial \boldsymbol{x}}(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t) + \frac{\partial \mathbf{q}}{\partial t}(\boldsymbol{x}, t).$$

Note that the above argument yields that if $\mathbf{q}$ is the push-forward of $\mathbf{Q}$, then $\frac{D\mathbf{q}}{Dt}$ is the

push-forward of $\frac{\partial \mathbf{q}}{\partial t}$. In particular, $\frac{D\boldsymbol{v}}{Dt}(\boldsymbol{x}, t) = \frac{\partial \mathbf{V}}{\partial t}(\mathbf{X}, t)$.

2.2 Conservation of Mass

2.2.1 Lagrangian Form

The governing equations for motion are derived from the conservation of mass and momentum. We assume that a material is equipped with a mass density function m such that, assuming $A \subset \Omega$,

$$mass(\phi(A, t)) = \int_{\phi(A,t)} m(\boldsymbol{x}, t) d\boldsymbol{x}.$$

Letting M denote the pull-back of m, $\mathbf{f}$ denote the push-forward of the deformation gradient $\boldsymbol{F}$, and $J = \det(\boldsymbol{F})$ we get that

$$\begin{aligned}
mass(\phi(A, t)) &= \int_{\phi(A,t)} m(\boldsymbol{x}, t) d\boldsymbol{x} \\
&= \int_{\phi^{-1}(\phi(A,t))} m(\phi(\mathbf{X}, t), t) \det\left(\frac{\partial \phi}{\partial \mathbf{X}}(\mathbf{X}, t)\right) d\mathbf{X} \\
&= \int_A M(\mathbf{X}, t) \det\left(\frac{\partial \phi}{\partial \mathbf{X}}(\mathbf{X}, t)\right) d\mathbf{X} \\
&= \int_A M(\mathbf{X}, t) J(\mathbf{X}, t) d\mathbf{X}
\end{aligned}$$

The conservation of mass requires that for all $A \subset \Omega$ and $t \in \mathbb{R}$, $mass(\phi(A, t)) = mass(A)$. Rewriting $mass(A) = mass(\phi(A, 0))$, we get

$$mass(\phi(A, t)) = mass(\phi(A, 0))$$
$$\int_A M(\mathbf{X}, t) J(\mathbf{X}, t) d\mathbf{X} = \int_A M(\mathbf{X}, 0) J(\mathbf{X}, 0) d\mathbf{X}$$

As $J(\mathbf{X}, 0) = 1$ (since $\phi(\mathbf{X}, 0) = \mathbf{X}$), A is an arbitrary subset of Ω, and the above must

hold for all A, we get that the conservation of mass holds if and only if

$$M(\mathbf{X}, t) J(\mathbf{X}, t) = M(\mathbf{X}, 0)$$

for all $t, \mathbf{X}$.

$$\frac{d(M(\mathbf{X}, t) J(\mathbf{X}, t))}{dt} = 0 \quad \textbf{Conservation of Mass, Lagrangian form}$$

2.2.2 Eulerian Form

Using the material derivative identities above and letting j denote the push-forward of J, we can also deduce that conservation of mass holds if and only if $\frac{D(m(\boldsymbol{x}, t) j(\boldsymbol{x}, t))}{Dt} = 0$. Observe that

$$
\begin{aligned}
\frac{\partial \boldsymbol{F}}{\partial t} &= \frac{\partial \frac{\partial \phi}{\partial \mathbf{X}}}{\partial t} \\
&= \frac{\partial \frac{\partial \phi}{\partial t}}{\partial \mathbf{X}} \\
&= \frac{\partial \mathbf{V}}{\partial \mathbf{X}} \\
&= \partial_{\mathbf{X}}(\boldsymbol{v}(\phi(\mathbf{X}, t), t)) \\
&= \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \frac{\partial \phi}{\partial \mathbf{X}} \\
&= \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \boldsymbol{F}.
\end{aligned}
$$

Unraveling the Eulerian form of the conservation of mass and using the additional identity that

$$\frac{\partial J(\mathbf{X}, t)}{\partial \boldsymbol{F}} = J(\mathbf{X}, t) \boldsymbol{F}(\mathbf{X}, t)^{-T},$$

we get

$$0 = \frac{D(m(\boldsymbol{x}, t) j(\boldsymbol{x}, t))}{Dt}$$

$$
\begin{aligned}
&= \frac{Dm(\boldsymbol{x},t)}{Dt} j(\boldsymbol{x},t) + m(\boldsymbol{x},t) \frac{Dj(\boldsymbol{x},t)}{Dt} \\
&= \frac{Dm(\boldsymbol{x},t)}{Dt} J(\mathbf{X},t) + m(\boldsymbol{x},t) \frac{\partial J(\mathbf{X},t)}{\partial t} \\
&= \frac{Dm(\boldsymbol{x},t)}{Dt} J(\mathbf{X},t) + m(\boldsymbol{x},t) \frac{\partial J(\mathbf{X},t)}{\partial \boldsymbol{F}} : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t} \\
&= \frac{Dm(\boldsymbol{x},t)}{Dt} J(\mathbf{X},t) + m(\boldsymbol{x},t) J(\mathbf{X},t) \boldsymbol{F}(\mathbf{X},t)^{-T} : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t} \\
&= \frac{Dm(\boldsymbol{x},t)}{Dt} J(\mathbf{X},t) + m(\boldsymbol{x},t) J(\mathbf{X},t) \operatorname{tr}\left(\boldsymbol{F}(\mathbf{X},t)^{-T} \boldsymbol{F}(\mathbf{X},t)^{T} \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}(\boldsymbol{x},t)^{T} \right) \\
&= \frac{Dm(\boldsymbol{x},t)}{Dt} J(\mathbf{X},t) + m(\boldsymbol{x},t) J(\mathbf{X},t) \nabla \cdot \boldsymbol{v}(\boldsymbol{x},t)
\end{aligned}
$$

In our treatment, we assume that $J(\mathbf{X},t) > 0$ for all $\mathbf{X}$ and t. This simplifies the above equation to

$$
\frac{Dm(\boldsymbol{x},t)}{Dt} + m(\boldsymbol{x},t) \nabla \cdot \boldsymbol{v}(\boldsymbol{x},t) = 0 \qquad \textbf{Conservation of Mass, Eulerian form}
$$

2.3 Conservation of Linear Momentum

2.3.1 Eulerian Form

The conservation of momentum requires that the rate of change in momentum in the material is equal to the net forces on the material. We assume the existence of a Cauchy stress $\sigma : (\Omega, \mathbb{R}) \to \mathbb{R}^{n \times n}$[Gonzalez and Stuart, 2008], a spatially-varying tensor that computes the force per unit area that the positive side of the material exerts upon the negative side of the material (as determined by a unit vector $\mathbf{n}$) at $\boldsymbol{x} \in \Omega$ and time t. Given a subset of the material $A_t \subset \Omega_t$, this force can be written as $\int_{\partial A_t} \sigma(\boldsymbol{x},t) \mathbf{n}(\boldsymbol{x},t) ds(\boldsymbol{x})$. We also assume the existence of body forces on the material, which we denote by $\mathbf{g}(\boldsymbol{x},t)$. Hence, the conservation of momentum in Eulerian

form is

$$\frac{d}{dt} \int_{A_t} m(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)d\boldsymbol{x} = \int_{\partial A_t} \sigma(\boldsymbol{x},t)\mathbf{n}(\boldsymbol{x},t)ds(\boldsymbol{x}) + \int_{A_t} \mathbf{g}(\boldsymbol{x},t)d\boldsymbol{x}$$

We can simplify the left-hand side:

$$
\frac{d}{dt} \int_{A_t} m(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)d\boldsymbol{x}
$$
$$
= \frac{d}{dt} \int_{\phi^{-1}(\phi(A,t))} m(\phi(\mathbf{X},t),t)v(\phi(\mathbf{X},t),t)\det(\boldsymbol{F}(\mathbf{X},t))d\mathbf{X}
$$
$$
= \frac{d}{dt} \int_{A} M(\mathbf{X},t)J(\mathbf{X},t)\mathbf{V}(\mathbf{X},t)d\mathbf{X}
$$
$$
= \frac{d}{dt} \int_{A} M(\mathbf{X},0)J(\mathbf{X},0)\mathbf{V}(\mathbf{X},t)d\mathbf{X}
$$
$$
= \int_{A} M(\mathbf{X},0)\mathbf{A}(\mathbf{X},t)d\mathbf{X}
$$
$$
= \int_{A} M(\mathbf{X},t)J(\mathbf{X},t)\mathbf{A}(\mathbf{X},t)d\mathbf{X}
$$
$$
= \int_{A_t} m(\boldsymbol{x},t)\mathbf{a}(\boldsymbol{x},t)d\boldsymbol{x}
$$

Using the divergence theorem on $\int_{\partial A_t} \sigma(\boldsymbol{x},t)\mathbf{n}(\boldsymbol{x},t)ds(\boldsymbol{x})$, we get that

$$\int_{A_t} m(\boldsymbol{x},t)\mathbf{a}(\boldsymbol{x},t)d\boldsymbol{x} = \int_{A_t} \nabla \cdot \sigma(\boldsymbol{x},t)d\boldsymbol{x} + \int_{A_t} \mathbf{g}(\boldsymbol{x},t)d\boldsymbol{x}$$

Since this must hold for all $A_t \subset \Omega_t$, we get

$$m(\boldsymbol{x},t)\mathbf{a}(\boldsymbol{x},t) = \nabla \cdot \sigma(\boldsymbol{x},t) + \mathbf{g}(\boldsymbol{x},t) \quad \textbf{Conservation of Momentum, Eulerian form}$$

2.3.2 Lagrangian Form

Using a change of surface variables, we get [Gonzalez and Stuart, 2008]

$$\int_{\partial A_t} \sigma(\boldsymbol{x},t)\mathbf{n}(\boldsymbol{x},t)ds(\boldsymbol{x}) = \int_{\partial A} J(\mathbf{X},t)\sigma(\phi(\mathbf{X},t),t)\boldsymbol{F}(\mathbf{X},t)^{-T}\mathbf{N}(\mathbf{X})ds(\mathbf{X}).$$

10

We define the first Piola-Kirchhoff stress as

$$P(\mathbf{X},t) := J(\mathbf{X},t)\sigma(\phi(\mathbf{X},t),t)\boldsymbol{F}(\mathbf{X},t)^{-T},$$

so that $\int_{\partial A_t} \sigma(\boldsymbol{x},t)\mathbf{n}(\boldsymbol{x},t)ds(\boldsymbol{x}) = \int_{\partial A} \boldsymbol{P}(\mathbf{X},t)\mathbf{N}(\mathbf{X})ds(\mathbf{X})$. Using the third to last equality in the simplification above and the divergence theorem, we have

$$
\begin{aligned}
\int_A M(\mathbf{X},0)J(\mathbf{X},0)\mathbf{A}(\mathbf{X},t)d\mathbf{X} &= \int_{\partial A_t} \sigma(\boldsymbol{x},t)\mathbf{n}(\boldsymbol{x},t)ds(\boldsymbol{x}) + \int_{A_t} \mathbf{g}(\boldsymbol{x},t)d\boldsymbol{x} \\
&= \int_{\partial A} \boldsymbol{P}(\mathbf{X},t)\mathbf{N}(\mathbf{X})ds(\mathbf{X}) + \int_A \mathbf{G}(\mathbf{X},t)J(\mathbf{X},t)d\boldsymbol{x} \\
&= \int_A \nabla \cdot \boldsymbol{P}(\mathbf{X},t) + \int_A \mathbf{G}(\mathbf{X},t)J(\mathbf{X},t)d\mathbf{X}.
\end{aligned}
$$

Since this holds for all $A \subset \Omega$, we get

<div align="center">

Conservation of Momentum, Lagrangian form

</div>

$$M(\mathbf{X},0)\mathbf{A}(\mathbf{X},t) = \nabla \cdot \boldsymbol{P}(\mathbf{X},t) + J(\mathbf{X},t)\mathbf{G}(\mathbf{X},t)$$

2.4   Net Working

2.4.1   Lagrangian Form

This section closely follows the treatment in [Gonzalez and Stuart, 2008]. Define the kinetic energy as

$$KE(A_t) = \frac{1}{2}\int_{A_t} m(\boldsymbol{x},t)\|\boldsymbol{v}(\boldsymbol{x},t)\|^2 d\boldsymbol{x}$$

and the power as

$$P(A_t) = \int_{A_t} m(\boldsymbol{x},t)\mathbf{g}(\boldsymbol{x},t)\cdot\boldsymbol{v}(\boldsymbol{x},t)d\boldsymbol{x} + \int_{\partial A_t} \sigma(\boldsymbol{x},t)\mathbf{n}(\boldsymbol{x},t)\cdot\boldsymbol{v}(\boldsymbol{x},t)ds(\boldsymbol{x}).$$

To simplify expressions in this treatment, we assume **g** (as in [Gonzalez and Stuart, 2008]) is a force per unit mass so that the conservation of momentum becomes

$$M(\mathbf{X}, 0)\mathbf{A}(\mathbf{X}, t) = \nabla \cdot \boldsymbol{P}(\mathbf{X}, t) + M(\mathbf{X}, 0)\mathbf{G}(\mathbf{X}, t)$$

after applying the conservation of mass, $M(\mathbf{X}, t)J(\mathbf{X}, t) = M(\mathbf{X}, 0)$.

Using another application of the conservation of mass in Lagrangian form and moving to the reference configuration, we get

$$
\begin{aligned}
KE(A_t) &= \frac{1}{2}\int_{A_t} m(\boldsymbol{x}, t)\|\boldsymbol{v}(\boldsymbol{x}, t)\|^2 d\boldsymbol{x} \\
&= \frac{1}{2}\int_{\phi(A,t)} M(\phi^{-1}(\boldsymbol{x}, t), t)\|\mathbf{V}(\phi^{-1}(\boldsymbol{x}, t))\|^2 d\boldsymbol{x} \\
&= \frac{1}{2}\int_{\phi^{-1}(\phi(A,t),t)} M(\mathbf{X}, t)\|\mathbf{V}(\mathbf{X}, t)\|^2 J(\mathbf{X}, t)d\mathbf{X} \\
&= \frac{1}{2}\int_A M(\mathbf{X}, 0)\|\mathbf{V}(\mathbf{X}, t)\|^2 d\mathbf{X}.
\end{aligned}
$$

Differentiating with respect to time and using the conservation of momentum gives

$$
\begin{aligned}
\partial_t KE(A_t) &= \frac{1}{2}\int_A M(\mathbf{X}, 0)\partial_t\|\mathbf{V}(\mathbf{X}, t)\|^2 d\mathbf{X} \\
&= \frac{1}{2}\int_A M(\mathbf{X}, 0)2\mathbf{V}(\mathbf{X}, t)\cdot\mathbf{A}(\mathbf{X}, t)d\mathbf{X} \\
&= \int_A (\nabla\cdot\boldsymbol{P}(\mathbf{X}, t) + M(\mathbf{X}, 0)\mathbf{G}(\mathbf{X}, t))\cdot\mathbf{V}(\mathbf{X}, t)d\mathbf{X}.
\end{aligned}
$$

Define σ_m as the pull-back of σ. Evaluating $P(A_t)$ with quantities in the reference configuration is done by

$$
\begin{aligned}
P(A_t) &= \int_{A_t} m(\boldsymbol{x}, t)\mathbf{g}(\boldsymbol{x}, t)\cdot\boldsymbol{v}(\boldsymbol{x}, t)d\boldsymbol{x} + \int_{\partial A_t} \sigma(\boldsymbol{x}, t)\mathbf{n}(\boldsymbol{x}, t)\cdot\boldsymbol{v}(\boldsymbol{x}, t)ds(\boldsymbol{x}) \\
&= \int_{\phi(A,t)} M(\phi^{-1}(\boldsymbol{x}, t), t)\mathbf{G}(\phi^{-1}(\boldsymbol{x}, t), t)\cdot\mathbf{V}(\phi^{-1}(\boldsymbol{x}, t), t)d\boldsymbol{x} \\
&\quad + \int_{\partial\phi(A,t)} \sigma_m^T(\phi^{-1}(\boldsymbol{x}, t), t)\boldsymbol{v}(\phi^{-1}(\boldsymbol{x}, t), t)\cdot\mathbf{n}(\phi^{-1}(\boldsymbol{x}, t), t)ds(\boldsymbol{x})
\end{aligned}
$$

$$= \int_{\phi^{-1}(\phi(A,t),t)} M(\mathbf{X},t)\mathbf{G}(\mathbf{X},t)\cdot\mathbf{V}(\mathbf{X},t)J(\mathbf{X},t)d\mathbf{X}$$

$$+ \int_{\partial\phi^{-1}(\phi(A,t),t)} J(\mathbf{X},t)\sigma_m^T(\mathbf{X},t)\mathbf{V}(\mathbf{X},t)\cdot\mathbf{F}(\mathbf{X},t)^{-T}\mathbf{N}(\mathbf{X},t)ds(\mathbf{X})$$

$$= \int_A M(\mathbf{X},0)\mathbf{G}(\mathbf{X},t)\cdot\mathbf{V}(\mathbf{X},t)d\mathbf{X}+$$

$$+ \int_{\partial A} J(\mathbf{X},t)\mathbf{V}(\mathbf{X},t)\cdot\sigma_m(\mathbf{X},t)\mathbf{F}(\mathbf{X},t)^{-T}\mathbf{N}(\mathbf{X},t)ds(\mathbf{X})$$

$$= \int_A M(\mathbf{X},0)\mathbf{G}(\mathbf{X},t)\cdot\mathbf{V}(\mathbf{X},t)d\mathbf{X} + \int_{\partial A}\mathbf{V}(\mathbf{X},t)\cdot\mathbf{P}(\mathbf{X},t)\mathbf{N}(\mathbf{X},t)ds(\mathbf{X})$$

$$= \int_A M(\mathbf{X},0)\mathbf{G}(\mathbf{X},t)\cdot\mathbf{V}(\mathbf{X},t)d\mathbf{X} + \int_{\partial A}\mathbf{P}(\mathbf{X},t)^T\mathbf{V}(\mathbf{X},t)\cdot\mathbf{N}(\mathbf{X},t)ds(\mathbf{X})$$

$$= \int_A M(\mathbf{X},0)\mathbf{G}(\mathbf{X},t)\cdot\mathbf{V}(\mathbf{X},t) + \nabla\cdot(\mathbf{P}(\mathbf{X},t)^T\mathbf{V}(\mathbf{X},t))d\mathbf{X}$$

$$= \int_A (M(\mathbf{X},0)\mathbf{G}(\mathbf{X},t) + \nabla\cdot\mathbf{P}(\mathbf{X},t))\cdot\mathbf{V}(\mathbf{X},t) + \mathbf{P}(\mathbf{X},t):\frac{\partial\mathbf{V}(\mathbf{X},t)}{\partial\mathbf{X}}d\mathbf{X}.$$

If we plug in our expression for $\partial_t KE(A_t)$ above and use the identity that

$$\frac{\partial\mathbf{V}(\mathbf{X},t)}{\partial\mathbf{X}} = \frac{\partial\mathbf{F}(\mathbf{X},t)}{\partial t}$$

we get

$$P(A_t) = \partial_t KE(A_t) + \int_A \mathbf{P}:\frac{\partial\mathbf{V}(\mathbf{X},t)}{\partial\mathbf{X}}d\mathbf{X}$$

$$= \partial_t KE(A_t) + \int_A \mathbf{P}:\frac{\partial\mathbf{F}(\mathbf{X},t)}{\partial t}d\mathbf{X}.$$

This relates the kinetic energy in A_t with the power of external forces on A_t. Net working is defined as

$$W(A_t) := P(A_t) - \partial_t KE(A_t),$$

from which immediately deduce

$$W(A_t) = \int_A \mathbf{P}:\frac{\partial\mathbf{F}(\mathbf{X},t)}{\partial t}d\mathbf{X} \quad \textbf{Net Working, Lagrangian Form}$$

2.4.2 Other identities

The quantity $\boldsymbol{P} : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t}$ will be important in our development of plasticity. It is called the stress power and measures the power done by internal stresses.

Defining the Kirchhoff stress tensor

$$\tau(\mathbf{X},t) = J(\mathbf{X},t)\sigma_m(\mathbf{X},t),$$

we get [Bonet and Wood, 2008]

$$
\begin{aligned}
W(A_t) &= \int_A \boldsymbol{P} : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t} d\mathbf{X} \\
&= \int_A J(\mathbf{X},t)\sigma_m(\mathbf{X},t)\boldsymbol{F}(\mathbf{X},t)^{-T} : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t} d\mathbf{X} \\
&= \int_A \operatorname{tr}\left(J(\mathbf{X},t)\boldsymbol{F}(\mathbf{X},t)^{-1}\sigma_m^T(\mathbf{X},t)\frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t} \right) d\mathbf{X} \\
&= \int_A J(\mathbf{X},t)\sigma_m(\mathbf{X},t) : \left(\frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t}\boldsymbol{F}(\mathbf{X},t)^{-1} \right) d\mathbf{X} \\
&= \int_A J(\mathbf{X},t)\sigma_m(\mathbf{X},t) : \frac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{\partial \boldsymbol{x}} d\mathbf{X} \\
&= \int_A \tau(\mathbf{X},t) : \frac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{\partial \boldsymbol{x}} d\mathbf{X}.
\end{aligned}
$$

This shows that $\tau : \frac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{d\boldsymbol{x}}$ also expresses stress power with respect to the reference configuration. As in [Bonet and Wood, 2008], we will use this particular expression in our plasticity development.

2.4.3 Eulerian Form

From above, we have

$$W(A_t) = \int_A \tau(\mathbf{X},t) : \frac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{d\boldsymbol{x}} d\mathbf{X}$$

14

$$= \int_{\phi^{-1}(\phi(A,t),t)} J(\mathbf{X},t)\sigma(\phi(\mathbf{X},t)) : \frac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{\partial \boldsymbol{x}} d\mathbf{X}$$

$$= \int_{A_t} \sigma(\boldsymbol{x},t) : \frac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{\partial \boldsymbol{x}} d\boldsymbol{x}.$$

$$W(A_t) = \int_{A_t} \sigma(\boldsymbol{x},t) : \frac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{d\boldsymbol{x}} d\boldsymbol{x} \quad \textbf{Net Working, Eulerian Form}$$

2.5 Hyperelasticity

2.5.1 Definition

This section closely follows the treatment in [Gonzalez and Stuart, 2008]. From [Gonzalez and Stuart, 2008], a material that is hyperelastic must satisfy:

1. $\mathbf{T}(\mathbf{X},t) = \mathbf{T}^*(\boldsymbol{F}(\mathbf{X},t))$ for all stresses $\mathbf{T} \in \{\sigma_m, \boldsymbol{P}, \boldsymbol{F}^{-1}\boldsymbol{P}\}$. This means that the stress tensors must be functions of $\boldsymbol{F}$ only.

2. There is an energy density Ψ such that $\boldsymbol{P}^*(\boldsymbol{F}(\mathbf{X},t)) = \frac{\partial \Psi(\boldsymbol{F}(\mathbf{X},t),t)}{\partial \boldsymbol{F}}$.

3. $\frac{\partial \Psi}{\partial \boldsymbol{F}} \boldsymbol{F}^T = \boldsymbol{F} \left(\frac{\partial \Psi}{\partial \boldsymbol{F}} \right)^T$ for all $\boldsymbol{F}$ with $J > 0$.

Note that

$$\boldsymbol{P}(\mathbf{X},t) : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t} = \boldsymbol{P}^*(\boldsymbol{F}(\mathbf{X},t)) : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t}$$

$$= \frac{\partial \Psi(\boldsymbol{F}(\mathbf{X},t),t)}{\partial \boldsymbol{F}} : \frac{\partial \boldsymbol{F}(\mathbf{X},t)}{\partial t}$$

$$= \partial_t \Psi(\boldsymbol{F}(\mathbf{X},t)).$$

2.5.2 Net Working and Strain Energy

As in [Gonzalez and Stuart, 2008], define the strain energy as

$$E(A_t) := \int_A \Psi(\boldsymbol{F}(\mathbf{X}, t)) d\mathbf{X}.$$

From the Lagrangian form of net working, we have that

$$
\begin{aligned}
W(A_t) &= \int_A \boldsymbol{P} : \frac{\partial \boldsymbol{F}(\mathbf{X}, t)}{\partial t} d\mathbf{X} \\
&= \int_A \partial_t \Psi(\boldsymbol{F}(\mathbf{X}, t)) d\mathbf{X} \\
&= \frac{\partial}{\partial t} \int_A \Psi(\boldsymbol{F}(\mathbf{X}, t)) d\mathbf{X} \\
&= \frac{\partial}{\partial t} E(A_t).
\end{aligned}
$$

Recalling that $W(A_t) = P(A_t) - \partial_t KE(A_t)$, we immediately see that

$$W(A_t) = \frac{\partial}{\partial t} E(A_t) \quad \textbf{Net Working for Hyperelastic Materials}$$

and

$$\frac{\partial}{\partial t}(E(A_t) + KE(A_t)) = P(A_t) \quad \textbf{Energy Balance for Hyperelastic Materials}$$

2.5.3 Hyperelastic Constitutive Models

2.5.3.1 Linear Elasticity

The linear elasticity constitutive model defines the elastic response via

$$\boldsymbol{P}(\mathbf{X}, t) = 2\mu\epsilon + \lambda \operatorname{tr}(\epsilon)\mathbf{I},$$

[Zhu et al., 2010] where

$$\epsilon(\mathbf{X}, t) = \frac{1}{2}(\boldsymbol{F}(\mathbf{X}, t) + \boldsymbol{F}(\mathbf{X}, t)^T) - \mathbf{I}.$$

Here, μ and λ are Lamé parameters.

2.5.4 Neo-Hookean

The Neo-Hooken model is defined via the energy density [Stomakhin et al., 2012]

$$\psi(\boldsymbol{F}) = \frac{\mu}{2}\left(\mathrm{tr}(\boldsymbol{F}\boldsymbol{F}^T) - 3\right) - \mu\ln(\det(\boldsymbol{F})) + \frac{\lambda}{2}(\det(\boldsymbol{F}))^2.$$

2.6 Cauchy-Green strain

2.6.1 Definitions

Define the right Cauchy-Green strain as

$$\mathbf{C}(\mathbf{X}, t) = \boldsymbol{F}(\mathbf{X}, t)^T \boldsymbol{F}(\mathbf{X}, t)$$

and the left Cauchy green strain as

$$\mathbf{B}(\mathbf{X}, t) = \boldsymbol{F}(\mathbf{X}, t)\boldsymbol{F}(\mathbf{X}, t)^T.$$

Note that since $\mathbf{C}$ is symmetric there exists an orthogonal matrix $\mathbf{Q}$ and a diagonal matrix Λ such that $\mathbf{C} = \mathbf{Q}^T\Lambda\mathbf{Q}$. Furthermore, we claim that Λ_{ii} is positive. To see this, let $\mathbf{w}_i = \mathbf{Q}^T\mathbf{e}_i$ and as $\mathbf{w}_i \neq \mathbf{0}$ and $\mathbf{C}$ is positive-definite, we get

$$\mathbf{w}_i^T\mathbf{C}\mathbf{w}_i > 0 \Rightarrow$$

$$\mathbf{w}_i^T \mathbf{Q}^T \Lambda \mathbf{Q} \mathbf{w}_i > 0 \Rightarrow$$

$$\mathbf{e}_i^T \mathbf{Q} \mathbf{Q}^T \Lambda \mathbf{Q} \mathbf{Q}^T \mathbf{e}_i > 0 \Rightarrow$$

$$\mathbf{e}_i^T \Lambda \mathbf{e}_i > 0 \Rightarrow$$

$$\Lambda_{ii} > 0.$$

Similarly, all of the eigenvalues of $\mathbf{B}$ are positive.

2.6.2 Significance of the right Cauchy-Green strain

This section closely follows the treatment in [Gonzalez and Stuart, 2008]. Define the function $\mathbf{g} : (\mathbb{R}^3, \mathbb{R}^3) \to \mathbb{R}$ by $\mathbf{g}(\mathbf{a}, \mathbf{b}) = \boldsymbol{F}\mathbf{a} \cdot \boldsymbol{F}\mathbf{b}$, and the $\mathbf{C}$ norm on vectors by $\|\boldsymbol{v}\|_{\mathbf{C}} = \sqrt{\mathbf{g}(\boldsymbol{v}, \boldsymbol{v})}$. Note that

$$\begin{aligned}
\|\boldsymbol{v}\|_{\mathbf{C}} &= \sqrt{\mathbf{g}(\boldsymbol{v}, \boldsymbol{v})} \\
&= \sqrt{\boldsymbol{F}\boldsymbol{v} \cdot \boldsymbol{F}\boldsymbol{v}} \\
&= \sqrt{\boldsymbol{v} \cdot \mathbf{C}\boldsymbol{v}}
\end{aligned}$$

Let $\mathbf{X} \in \Omega$. Then

$$\phi(\mathbf{X} + d\mathbf{X}) = \phi(\mathbf{X}) + \boldsymbol{F}(\mathbf{X})d\mathbf{X} + \mathcal{O}(\|d\mathbf{X}\|^2) \Rightarrow$$

$$\phi(\mathbf{X} + d\mathbf{X}) - \phi(\mathbf{X}) = \boldsymbol{F}(\mathbf{X})d\mathbf{X} + \mathcal{O}(\|d\mathbf{X}\|^2)$$

Define $d\boldsymbol{x} = \phi(\mathbf{X} + d\mathbf{X}) - \phi(\mathbf{X})$. Then

$$\begin{aligned}
\|d\boldsymbol{x}\|^2 &= d\boldsymbol{x} \cdot d\boldsymbol{x} \\
&= \boldsymbol{F}(\mathbf{X})d\mathbf{X} \cdot \boldsymbol{F}(\mathbf{X})d\mathbf{X} + \mathcal{O}(\|d\mathbf{X}\|^3)
\end{aligned}$$

so that $\|d\boldsymbol{x}\| = \sqrt{\boldsymbol{F}(\mathbf{X})d\mathbf{X} \cdot \boldsymbol{F}(\mathbf{X})d\mathbf{X}} + \mathcal{O}(\|d\mathbf{X}\|^3)$. Fix a unit vector e and set $d\mathbf{X} = a\mathbf{e}$. Then $\|d\boldsymbol{x}\| = \sqrt{\boldsymbol{F}(\mathbf{X})a\mathbf{e} \cdot \boldsymbol{F}(\mathbf{X})a\mathbf{e}} + \mathcal{O}(a^3)$, and

$$\frac{\|d\boldsymbol{x}\|}{\|d\mathbf{X}\|} = \frac{\sqrt{\boldsymbol{F}(\mathbf{X})a\mathbf{e} \cdot \boldsymbol{F}(\mathbf{X})a\mathbf{e} + \mathcal{O}(a^3)}}{\sqrt{a\mathbf{e} \cdot a\mathbf{e}}} = \frac{a\sqrt{\boldsymbol{F}(\mathbf{X})\mathbf{e} \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e} + \mathcal{O}(a)}}{a}$$

$$= \sqrt{\boldsymbol{F}(\mathbf{X})\mathbf{e} \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}} \text{ as } a \to 0$$

$$= \|\mathbf{e}\|_{\mathbf{C}(\mathbf{X})}$$

This suggests that for a vector e, $\|\mathbf{e}\|_{\mathbf{C}(\mathbf{X})}$ quantifies the stretch in the direction of e under the deformation ϕ.

Consider two vectors with the same magnitude $\mathbf{X} \in \Omega$, $d\mathbf{X}_1, d\mathbf{X}_2$, in the directions of unit vectors $\mathbf{e}_1, \mathbf{e}_2$ so that we may write $d\mathbf{X}_1 = a\mathbf{e}_1, d\mathbf{X}_2 = a = \mathbf{e}_2$. Set $d\boldsymbol{x}_1 = \phi(\mathbf{X} + d\mathbf{X}_1) - \phi(\mathbf{X}), d\boldsymbol{x}_2 = \phi(\mathbf{X} + d\mathbf{X}_2) - \phi(\mathbf{X})$. Then the angle α between them is $\alpha = \cos^{-1}\left(\frac{d\boldsymbol{x}_1 \cdot d\boldsymbol{x}_2}{\|d\boldsymbol{x}_1\|\|d\boldsymbol{x}_2\|}\right)$ From above, we know that $d\boldsymbol{x}_1 = \boldsymbol{F}(\mathbf{X})d\mathbf{X}_1 + \mathcal{O}(\|d\mathbf{X}_1\|^2)$ and $d\boldsymbol{x}_2 = \boldsymbol{F}(\mathbf{X})d\mathbf{X}_2 + \mathcal{O}(\|d\mathbf{X}_2\|^2)$ so that

$$d\boldsymbol{x}_1 \cdot d\boldsymbol{x}_2 = \left(\boldsymbol{F}(\mathbf{X})a\mathbf{e}_1 + \mathcal{O}(a^2)\right) \cdot \left(\boldsymbol{F}(\mathbf{X})a\mathbf{e}_2 + \mathcal{O}(a^2)\right)$$

$$= a^2 \boldsymbol{F}(\mathbf{X})\mathbf{e}_1 \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}_2 + \mathcal{O}(a^3).$$

Also note that from above,

$$\|d\boldsymbol{x}_1\| = \sqrt{\boldsymbol{F}(\mathbf{X})a\mathbf{e}_1 \cdot \boldsymbol{F}(\mathbf{X})a\mathbf{e}_1} + \mathcal{O}(a^3)$$

$$= a\sqrt{\boldsymbol{F}(\mathbf{X})\mathbf{e}_1 \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}_1} + \mathcal{O}(a)$$

$$\|d\boldsymbol{x}_2\| = \sqrt{\boldsymbol{F}(\mathbf{X})a\mathbf{e}_2 \cdot \boldsymbol{F}(\mathbf{X})a\mathbf{e}_2} + \mathcal{O}(a^3)$$

$$= a\sqrt{\boldsymbol{F}(\mathbf{X})\mathbf{e}_2 \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}_2} + \mathcal{O}(a)$$

Then,

$$\alpha = \cos^{-1}\left(\frac{d\boldsymbol{x}_1 \cdot d\boldsymbol{x}_2}{\|d\boldsymbol{x}_1\|\|d\boldsymbol{x}_2\|}\right)$$

$$= \cos^{-1}\left(\frac{a^2\boldsymbol{F}(\mathbf{X})\mathbf{e}_1 \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}_2 + \mathcal{O}(a^3)}{a^2\sqrt{\boldsymbol{F}(\mathbf{X})\mathbf{e}_1 \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}_1 + \mathcal{O}(a)}\sqrt{\boldsymbol{F}(\mathbf{X})\mathbf{e}_2 \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}_2 + \mathcal{O}(a)}}\right)$$

$$= \cos^{-1}\left(\frac{\boldsymbol{F}(\mathbf{X})\mathbf{e}_1 \cdot \boldsymbol{F}(\mathbf{X})\mathbf{e}_2}{\|\mathbf{e}_1\|_{\mathbf{C}(\mathbf{X})}\|\mathbf{e}_2\|_{\mathbf{C}(\mathbf{X})}}\right) \text{ as } a \to 0$$

$$= \cos^{-1}\left(\frac{\mathbf{g}(\mathbf{e}_1, \mathbf{e}_2)}{\|\mathbf{e}_1\|_{\mathbf{C}(\mathbf{X})}\|\mathbf{e}_2\|_{\mathbf{C}(\mathbf{X})}}\right)$$

This suggests that for two vectors $\mathbf{e}_1, \mathbf{e}_2$, $\cos^{-1}\left(\frac{\mathbf{g}(\mathbf{e}_1,\mathbf{e}_2)}{\|\mathbf{e}_1\|_{\mathbf{C}(\mathbf{X})}\|\mathbf{e}_2\|_{\mathbf{C}(\mathbf{X})}}\right)$ is the angle between their images under the deformation ϕ.

2.6.3   Entries of the right Cauchy-Green strain

This section closely follows the treatment in [Gonzalez and Stuart, 2008].

Note that

$$\mathbf{C}_{ii} = \mathbf{e}_i \cdot \mathbf{C}\mathbf{e}_i = \|\mathbf{e}_i\|_{\mathbf{C}}^2,$$

so the diagonal entries of $\mathbf{C}$ are the squares of stretches of the standard basis under ϕ. Define β as the difference between the angle between two vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ and α. We call β the shear between these vectors under the deformation map ϕ, which quantifies how much the original angle changes under ϕ. Note that $\beta_{\mathbf{e}_i, \mathbf{e}_j} = \frac{\pi}{2}$ for $i \neq j$. Then, for $i \neq j$,

$$\mathbf{C}_{ij} = \mathbf{e}_i \cdot \mathbf{C}\mathbf{e}_j$$

$$= \mathbf{g}(\mathbf{e}_i, \mathbf{e}_j)$$

$$= \cos(\alpha_{\mathbf{u}_i, \mathbf{u}_j})\|\mathbf{u}_i\|\|\mathbf{u}_j\|$$

$$= \cos(\frac{\pi}{2} - \beta_{\mathbf{u}_i, \mathbf{u}_j})\|\mathbf{u}_i\|\|\mathbf{u}_j\|$$

$$= \sin(\beta_{\mathbf{u}_1, \mathbf{u}_j}) \|\mathbf{u}_i\| \|\mathbf{u}_j\|$$

This means that the off-diagonal entries of $\mathbf{C}$ quantify the shear angles between the basis vectors.

2.6.4 Left Cauchy-Green strain

The left Cauchy-Green strain $\mathbf{B} := \boldsymbol{F}\boldsymbol{F}^T$ plays a fundamental role in our treatment of plasticity. Relevant properties are detailed in Chapter 4.

CHAPTER 3

The Material Point Method

The Material Point Method is a hybrid Lagrangian-Eulerian method for simulating fluids. The method is Lagrangian in the sense that it that tracks positions, velocities, masses, and deformation information on Lagrangian particles. Relevant quantities are transferred to an Eulerian grid where stress-based force computations are performed. Stress-based forces are derived from the weak form in a manner similar to the finite element method. These quantities are then transferred back to the Lagrangian particles to update the system state. This chapter provides a derivation of the forces from the weak form and the subsequent discretization this yields. An overview which details a full timestep of the algorithm is presented at the end of this chapter.

3.1 Transfers

Quantities are transferred using standard grid-based interpolation functions. As described in [Stomakhin et al., 2013], if we define N as

$$N(x) = \begin{cases} \frac{1}{2}|x|^3 - x^2 + \frac{2}{3}, & |x| < 1 \\ -\frac{1}{6}|x|^3 + x^2 - 2|x| + \frac{4}{3}, & 1 \le |x| < 2 \\ 0, & \text{otherwise} \end{cases},$$

then our grid interpolation functions are defined as

$$N_g^{\Delta x}(\boldsymbol{x}) = N(\frac{1}{\Delta x}(x - i\Delta x))N(\frac{1}{\Delta x}(y - j\Delta x))N(\frac{1}{\Delta x}(z - k\Delta x)),$$

where (i, j, k) is the grid index for g and $\boldsymbol{x} = (x, y, z)$.

We assume that each particle p at $\boldsymbol{x}_p$ occupies a ball B_p^0 with volume $V_p^0 = \int_{B_p^0} d\mathbf{X}$. Furthermore, this allows us get the mass of a particle, namely,

$$m_p^n = \int_{B_p^n} m(\boldsymbol{x}, t)d\boldsymbol{x}$$

Next, we describe how to we get quantities from particles to the grid and from the grid to the particles.

Mass transfer from particles to grid

Given m_p on particles, grid nodes get mass via $m_g = \sum_p m_p N_g(\boldsymbol{x}_p)$

We note that this transfer is conservative:

$$\sum_g m_g = \sum_g \sum_p m_p N_g(\boldsymbol{x}_p)$$
$$= \sum_p m_p \sum_g N_g(\boldsymbol{x}_p)$$
$$= \sum_p m_p,$$

as $\sum_g N_g(\boldsymbol{x}) = 1$ for all $\boldsymbol{x}$.

Momentum transfer from particles to grid

Given $m_p, \boldsymbol{v}_p$ on particles, grid nodes get momentum via $(m\boldsymbol{v})_g = \sum_p m_p \boldsymbol{v}_p N_g(\boldsymbol{x}_p)$

This transfer is also conservative:

$$\sum_g (m\boldsymbol{v})_g = \sum_g \sum_p m_p \boldsymbol{v}_p N_g(\boldsymbol{x}_p)$$

$$= \sum_p m_p \boldsymbol{v}_p \sum_g N_g(\boldsymbol{x}_p)$$

$$= \sum_p m_p \boldsymbol{v}_p,$$

as $\sum_g N_g(\boldsymbol{x}) = 1$ for all $\boldsymbol{x}$.

Velocity transfer from particles to grid

Grid velocities are calculated as $\boldsymbol{v}_g = \frac{(m\boldsymbol{v})_g}{m_g}$

Once velocities are calculated on the Eulerian grid, quantities are transferred back to the particles to update particle velocities and the system state. Note that the mass on particles does not need to get updated.

Velocity transfer from grid to particles

Particle velocities are calculated as $\boldsymbol{v}_p = \sum_g \boldsymbol{v}_g N_g(\boldsymbol{x}_p)$

3.2 Weak Form

We use the Eulerian form of the conservation of momentum for our weak form to derive our forces on the grid. Let $\mathbf{w}(\boldsymbol{x}, t) : (\Omega_t, \mathbb{R}) \to \Omega_t$ be an arbitrary function. Then

$$\int_{\Omega_t} \mathbf{w}(\boldsymbol{x}, t) \cdot m(\boldsymbol{x}, t) \mathbf{a}(\boldsymbol{x}, t) d\boldsymbol{x} = \int_{\Omega_t} \mathbf{w}(\boldsymbol{x}, t) \cdot (\nabla \cdot \sigma(\boldsymbol{x}, t) + \mathbf{g}(\boldsymbol{x}, t)) d\boldsymbol{x}$$

$$= \int_{\Omega_t} \Big(\nabla \cdot (\sigma(\boldsymbol{x}, t)^T \mathbf{w}(\boldsymbol{x}, t)) - \nabla \mathbf{w}(\boldsymbol{x}, t) : \sigma(\boldsymbol{x}, t)$$

$$+ \mathbf{w}(\boldsymbol{x}, t) \cdot \mathbf{g}(\boldsymbol{x}, t) \Big) d\boldsymbol{x}$$

$$= \int_{\partial\Omega_t} \sigma(\boldsymbol{x},t)^T \mathbf{w}(\boldsymbol{x},t) \cdot \mathbf{n}(\boldsymbol{x}) ds(\boldsymbol{x})$$

$$- \int_{\Omega_t} \nabla\mathbf{w}(\boldsymbol{x},t) : \sigma(\boldsymbol{x},t) d\boldsymbol{x}$$

$$+ \int_{\Omega_t} \mathbf{w}(\boldsymbol{x},t) \cdot \mathbf{g}(\boldsymbol{x},t) d\boldsymbol{x}.$$

3.3   Discretization

Define $\boldsymbol{v}^n(\boldsymbol{x}) := \boldsymbol{v}(\boldsymbol{x},t^n)$ and $\boldsymbol{v}^{n+1}(\boldsymbol{x}) := \boldsymbol{v}(\boldsymbol{x},t^{n+1})$ for $\boldsymbol{x} \in \Omega_{t^n}$. Note that, fixing $\boldsymbol{x} \in \Omega_{t^n}$

$$\frac{1}{\Delta t}(\boldsymbol{v}^{n+1}(\boldsymbol{x}) - \boldsymbol{v}^n(\boldsymbol{x})) = \frac{1}{\Delta t} \left(\mathbf{V}(\phi^{-1}(\boldsymbol{x},t^n),t^{n+1}) - \mathbf{V}(\phi^{-1}(\boldsymbol{x},t^n),t^n) \right)$$

$$= \frac{1}{\Delta t} \left(\mathbf{V}(\mathbf{X},t^{n+1}) - \mathbf{V}(\mathbf{X},t^n) \right)$$

$$= \mathbf{A}(\mathbf{X},t^n) + \mathcal{O}(\Delta t)$$

$$= \mathbf{a}(\boldsymbol{x},t^n) + \mathcal{O}(\Delta t).$$

Then the weak form is discretized in time by

$$\frac{1}{\Delta t} \int_{\Omega_{t^n}} \mathbf{w}(\boldsymbol{x},t^n) \cdot m(\boldsymbol{x},t^n)(\boldsymbol{v}^{n+1}(\boldsymbol{x}) - \boldsymbol{v}^n(\boldsymbol{x})) d\boldsymbol{x} =$$

$$\int_{\partial\Omega_{t^n}} \sigma(\boldsymbol{x},t^n)^T \mathbf{w}(\boldsymbol{x},t^n) \cdot \mathbf{n}(\boldsymbol{x}) ds(\boldsymbol{x})$$

$$- \int_{\Omega_{t^n}} \nabla\mathbf{w}(\boldsymbol{x},t^n) : \sigma(\boldsymbol{x},t^n) d\boldsymbol{x}$$

$$+ \int_{\Omega_{t^n}} \mathbf{w}(\boldsymbol{x},t^n) \cdot \mathbf{g}(\boldsymbol{x},t^n) d\boldsymbol{x}.$$

We discretize in space by introducing basis functions on the grid $N_g(\boldsymbol{x})$ so that

$$\boldsymbol{v} = \sum_g \boldsymbol{v}_g N_g, \quad \mathbf{w} = \sum_g \mathbf{w}_g N_g.$$

Then

$$\frac{1}{\Delta t}\int_{\Omega_{t^n}}\sum_g \mathbf{w}_g N_g \cdot m(\boldsymbol{x},t^n)\sum_h (\boldsymbol{v}_h^{n+1}-\boldsymbol{v}_h^n)N_h d\boldsymbol{x} =$$

$$\int_{\partial\Omega_{t^n}}\sigma(\boldsymbol{x},t^n)^T \sum_g \mathbf{w}_g N_g \cdot \mathbf{n}(\boldsymbol{x})ds(\boldsymbol{x})$$

$$-\int_{\Omega_{t^n}}\sum_g \mathbf{w}_g \nabla N_g : \sigma(\boldsymbol{x},t^n)d\boldsymbol{x}$$

$$+\int_{\Omega_{t^n}}\sum_g \mathbf{w}_g N_g \cdot \sum_g \mathbf{g}_g^n N_g d\boldsymbol{x},$$

$$\frac{1}{\Delta t}\sum_{gh} \mathbf{w}_g \cdot (\boldsymbol{v}_h^{n+1}-\boldsymbol{v}_h^n)\int_{\Omega_{t^n}} N_g m(\boldsymbol{x},t^n)N_h d\boldsymbol{x} =$$

$$\int_{\partial\Omega_{t^n}}\sigma(\boldsymbol{x},t^n)^T \sum_g \mathbf{w}_g N_g \cdot \mathbf{n}(\boldsymbol{x})ds(\boldsymbol{x})$$

$$-\int_{\Omega_{t^n}}\sum_g \mathbf{w}_g \nabla N_g : \sigma(\boldsymbol{x},t^n)d\boldsymbol{x}$$

$$+\int_{\Omega_{t^n}}\sum_g \mathbf{w}_g N_g \cdot \sum_h \mathbf{g}_h^n N_h d\boldsymbol{x}.$$

This must hold for all choices of $\mathbf{w}$. In particular, if we pick $\mathbf{w}^*$ so that $\mathbf{w}^* = 1$ at grid node g and index i and $\mathbf{w}^* = 0$ otherwise, we get that

$$\frac{1}{\Delta t}\sum_h (\boldsymbol{v}_{hi}^{n+1}-\boldsymbol{v}_{hi}^n)\int_{\Omega_{t^n}} N_g m(\boldsymbol{x},t^n)N_h d\boldsymbol{x} = \int_{\partial\Omega_{t^n}} N_g \sigma(\boldsymbol{x},t^n)_{ki}\mathbf{n}(\boldsymbol{x})_k ds(\boldsymbol{x})$$

$$-\int_{\Omega_{t^n}} N_{g,k}\sigma(\boldsymbol{x},t^n)_{ik}d\boldsymbol{x}$$

$$+\int_{\Omega_{t^n}} N_g \sum_h \mathbf{g}_{hi}^n N_h d\boldsymbol{x}.$$

To simplify our computations, we approximate

$$\int_{\Omega_{t^n}} N_g m(\boldsymbol{x},t^n)N_h d\boldsymbol{x} \approx \delta_{gh}\int_{\Omega_{t^n}} N_g m(\boldsymbol{x},t^n)d\boldsymbol{x} \approx \delta_{gh}\sum_p N_g(\boldsymbol{x}_p)m_p = \delta_{gh}m_g.$$

From the transfers, $m_g \boldsymbol{v}_g = (m\boldsymbol{v}_g)$, so our equations become

$$\frac{1}{\Delta t}((m\boldsymbol{v})_{gi}^{n+1}-(m\boldsymbol{v})_{gi}^n) = \int_{\partial\Omega_{t^n}} N_g \sigma(\boldsymbol{x},t^n)_{ki}\mathbf{n}(\boldsymbol{x})_k ds(\boldsymbol{x})$$

$$-\int_{\Omega_{t^n}} N_{g,k}\sigma(\boldsymbol{x},t^n)_{ik}d\boldsymbol{x} + \int_{\Omega_{t^n}} N_g \sum_h \mathbf{g}_{hi}^n N_h d\boldsymbol{x}.$$

Furthermore, as we store deformation information on particles, we have an approximation to the Cauchy stress on the particles, σ_p^n. Using the approximation $V_p^n \approx V_p^0 J(\mathbf{X}_p, t^n)$ and approximating

$$J_p^n \approx J(\mathbf{X}_p, t^n) = \det(\boldsymbol{F}(\mathbf{X}_p, t^n)),$$

this allows us to approximate

$$\int_{\Omega_{t^n}} N_{g,k}\sigma(\boldsymbol{x},t^n)_{ik}d\boldsymbol{x} \approx \sum_p N_{g,k}(\boldsymbol{x}_p^n)\sigma_{pik}^n V_p^0 J_p^n$$

$$= \sum_p N_{g,k}(\boldsymbol{x}_p^n)\frac{1}{J_p^n}\boldsymbol{P}_{pij}^n \boldsymbol{F}_{pkj}^n V_p^0 J_p^n,$$

which simplifies to

$$\sum_p N_{g,k}(\boldsymbol{x}_p^n)\boldsymbol{P}_{pij}^n \boldsymbol{F}_{pkj}^n V_p^0. \tag{3.1}$$

This expression arises naturally as the gradient of an energy on the grid. To this end, recalling that for $\mathbf{X} \in \Omega$ $\boldsymbol{v}^{n+1}(\phi(\mathbf{X}, t^n)) = \mathbf{V}(\mathbf{X}, t^{n+1})$, note that

$$\begin{aligned}
\frac{\partial}{\partial t}\boldsymbol{F}(\mathbf{X}, t^{n+1}) &= \frac{\partial}{\partial t}\frac{\partial}{\partial \mathbf{X}}\phi(\mathbf{X}, t^{n+1}) \\
&= \frac{\partial}{\partial \mathbf{X}}\frac{\partial}{dt}\phi(\mathbf{X}, t^{n+1}) \\
&= \frac{\partial}{\partial \mathbf{X}}\mathbf{V}(\mathbf{X}, t^{n+1}) \\
&= \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{v}^{n+1}(\boldsymbol{x})\frac{\partial}{\partial \mathbf{X}}(\phi(\mathbf{X}, t^n)) \\
&= \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{v}^{n+1}(\boldsymbol{x})\boldsymbol{F}(\mathbf{X}, t^n).
\end{aligned}$$

Using a first-order approximation on particles, this becomes

$$\boldsymbol{F}_p^{n+1} = \left(\mathbf{I} + \Delta t\frac{\partial \boldsymbol{v}^{n+1}}{\partial \boldsymbol{x}}(\boldsymbol{x}_p^n)\right)\boldsymbol{F}_p^n.$$

Using the interpolation functions on the grid yields a discretization of $\frac{\partial \boldsymbol{v}^{n+1}}{\partial \boldsymbol{x}}(\boldsymbol{x}_p^n)$ so that

$$\boldsymbol{F}_p^{n+1} = \left(\mathbf{I} + \Delta t \sum_g \boldsymbol{v}_g^{n+1} \frac{\partial N_g}{\partial \boldsymbol{x}}(\boldsymbol{x}_p^n)\right) \boldsymbol{F}_p^n.$$

We can think of $\boldsymbol{v}^{n+1}$ as calculated from $\hat{\boldsymbol{x}}_g$ via $\boldsymbol{v}_g^{n+1} = \frac{1}{\Delta t}(\hat{\boldsymbol{x}}_g - \boldsymbol{x}_g)$, where $\boldsymbol{x}_g$ is the location of the grid node indexed by g [Stomakhin et al., 2013]. This allows us to write, for $\hat{\boldsymbol{x}}$ a vector on the grid,

$$\boldsymbol{F}_p^{n+1}(\hat{\boldsymbol{x}}) = \left(\mathbf{I} + \Delta t \sum_g \frac{1}{\Delta t}(\hat{\boldsymbol{x}}_g - \boldsymbol{x}_g)\frac{\partial N_g}{\partial \boldsymbol{x}}(\boldsymbol{x}_p^n)\right) \boldsymbol{F}_p^n.$$

As in [Stomakhin et al., 2013], we assume we have an energy $\Phi(\hat{\boldsymbol{x}})$ with an energy density such that $\Phi(\hat{\boldsymbol{x}}) = \sum_p \psi(\boldsymbol{F}_p(\hat{\boldsymbol{x}}))V_p^0$ with $\frac{\partial \psi}{\partial \boldsymbol{F}} = \boldsymbol{P}$. Let $_{,h\eta}$ denote differentiation with respect to the η coordinate of the grid node index by h. Observe that

$$\boldsymbol{F}_p(\hat{\boldsymbol{x}}) = (\mathbf{I} + \sum_g (\hat{\boldsymbol{x}}_g^n - \boldsymbol{x}_g^n)(\nabla N_g)^T(\boldsymbol{x}_p))\boldsymbol{F}^n$$

$$\boldsymbol{F}_{p\alpha\beta}(\hat{\boldsymbol{x}}) = \boldsymbol{F}_{p\alpha\beta}^n + \sum_g (\hat{x}_{g\alpha} - x_{g\alpha}^n)N_{g,\gamma}^n(\boldsymbol{x}_p)\boldsymbol{F}_{p\gamma\beta}^n$$

$$\boldsymbol{F}_{p\alpha\beta,h\eta}(\hat{\boldsymbol{x}}) = \delta_{\alpha\eta}N_{h,\gamma}(\boldsymbol{x}_p^n)\boldsymbol{F}_{p\gamma\beta}^n.$$

Then

$$\Phi = \sum_p V_p^0 \Psi_p$$

$$\Phi_{,h\eta} = \sum_p V_p^0 \left(\frac{\partial \Psi}{\partial \boldsymbol{F}}\right)_{\alpha\beta} \boldsymbol{F}_{\alpha\beta,h\eta}$$

$$\Phi_{,h\eta} = \sum_p V_p^0 \boldsymbol{P}_{p\eta\beta}N_{h,\gamma}(\boldsymbol{x}_p^n)\boldsymbol{F}_{p\gamma\beta}^n.$$

This is precisely expression 3.1 at $\hat{\boldsymbol{x}} = \boldsymbol{x}$, so that we may finally write our momentum

update as

$$\frac{1}{\Delta t}((m\boldsymbol{v})_{gi}^{n+1} - (m\boldsymbol{v})_{gi}^{n}) = \int_{\partial\Omega_{t^n}} N_g \sigma(\boldsymbol{x},t^n)_{ki}\mathbf{n}(\boldsymbol{x})_k ds(\boldsymbol{x})$$

$$- \int_{\Omega_{t^n}} N_{g,k}\sigma(\boldsymbol{x},t^n)_{ik}d\boldsymbol{x} + \int_{\Omega_{t^n}} N_g \sum_h \mathbf{g}_{hi}^n N_h d\boldsymbol{x}$$

$$\frac{1}{\Delta t}((m\boldsymbol{v})_{gi}^{n+1} - (m\boldsymbol{v})_{gi}^{n}) = \int_{\partial\Omega_{t^n}} N_g \sigma(\boldsymbol{x},t^n)_{ki}\mathbf{n}(\boldsymbol{x})_k ds(\boldsymbol{x})$$

$$- \frac{\partial\Phi}{\partial\hat{\boldsymbol{x}}_{gi}}(\boldsymbol{x}) + \int_{\Omega_{t^n}} N_g \sum_h \mathbf{g}_{hi}^n N_h d\boldsymbol{x}.$$

3.4   Overview

This follows the procedure outlined in [Stomakhin et al., 2013]. Define

$$w_{\mathbf{i}p} := N_{\mathbf{i}}^{\Delta x}(\boldsymbol{x}_p),$$

where $\mathbf{i}$ is a grid index, $\boldsymbol{x}_p$ is particle position, and $N_{\mathbf{i}}^{\Delta x}$ is the interpolating function centered at grid node indexed by $\mathbf{i}$, as defined above.

<p align="center"><strong style="color:red">Particles → Grid → Particles</p>

- Transfer mass using $m_{\mathbf{i}}^n = \sum_p m_p \mathbf{w}_{\mathbf{i}p}^n$

- Transfer velocity using $m_{\mathbf{i}}^n \boldsymbol{v}_{\mathbf{i}}^n = \sum_p \boldsymbol{v}_p m_p w_{\mathbf{i}p}^n$

<p align="center">Particles → <strong style="color:red">Grid → Particles</p>

- Compute the MPM approximation to the total potential energy,

  $\Phi(\boldsymbol{x}) = \sum_p \Psi(\boldsymbol{x}) V_p^0$

- Forces are given by $\mathbf{f}_{\mathbf{i}}(\boldsymbol{x}) = -\frac{\partial\Phi}{\partial\boldsymbol{x}_i}(\boldsymbol{x})$

- Implicit update is given by $\boldsymbol{v}_{\mathbf{i}}^{n+1} = \boldsymbol{v}_i^n + \Delta t m_{\mathbf{i}}^{-1}\mathbf{f}_{\mathbf{i}}(\boldsymbol{x}_i + \Delta t\boldsymbol{v}_{\mathbf{i}}^{n+1})$

- Perform grid-based collisions

- Solve the system given by the implicit update after grid-based collisions are processed

<div align="center">

Particles → **Grid** → **Particles**

</div>

- Calculate $\nabla \boldsymbol{v}_p^{n+1} = \sum_{\mathbf{i}} \boldsymbol{v}_{\mathbf{i}}^{n+1} (\nabla w_{\mathbf{i}p}^n)^T$

- Update deformation gradient $\boldsymbol{F}_p^{n+1} = (\mathbf{I} + \Delta t \nabla \boldsymbol{v}_p^{n+1}) \boldsymbol{F}_p^n$

- Update particle velocities

$$\boldsymbol{v}_p^{n+1} = (1 - \alpha) \sum_{\mathbf{i}} \boldsymbol{v}_{\mathbf{i}}^{n+1} w_{\mathbf{i}p}^n + \alpha \big(\boldsymbol{v}_p^n + \sum_{\mathbf{i}} (\boldsymbol{v}_{\mathbf{i}}^{n+1} - \boldsymbol{v}_{\mathbf{i}}^n) w_{\mathbf{i}p}^n \big)$$

- Perform particle-based body collisions on $\boldsymbol{v}_p^{n+1}$

- Update particle positions $\boldsymbol{x}_p^{n+1} = \boldsymbol{x}_p^n + \Delta t \boldsymbol{v}_p^{n+1}$

CHAPTER 4

Plasticity

Our plasticity treatment assumes a multiplicative decomposition of the deformation gradient. Detailed descriptions of the plastic rate of deformation and flow rule are provided. Von Mises, Bingham, Herschel-Bulkley and Oldroyd-B models are discussed, as well as isochoric plasticity, shear effects, and material softening and hardening. Novel discretizations are proposed that preserve symmetry and positive-definiteness of strain tensors.

4.1 Multiplicative Decomposition

4.1.1 Permanent Deformation

This section closely follows the treatment in [Bonet and Wood, 2008].

Let ϕ be a deformation map on Ω, and $\mathbf{Q} \in \Omega$. Consider the linearization of ϕ via the deformation gradient $\boldsymbol{F}$,

$$\phi(\mathbf{Q} + d\mathbf{X}, t) = \phi(\mathbf{Q}, t) + \boldsymbol{F}(\mathbf{Q}, t)d\mathbf{X}.$$

This approximation is valid for all $d\mathbf{X}$ sufficiently small. Define

$$B_\epsilon(\mathbf{Q}) := \{\mathbf{Q} + d\mathbf{Y} \mid \|d\mathbf{Y}\| \leq \epsilon\}$$

31

$$\phi(B_\epsilon(\mathbf{Q}), t) := \{\phi(\mathbf{X}, t) \mid \mathbf{X} \in B_\epsilon(\mathbf{Q})\}.$$

In the following, fix $\mathbf{Q}$ and ϵ.

Consider excising $\phi(B_\epsilon(\mathbf{Q}), t)$ from $\phi(\Omega, t)$ and letting it come to rest completely independently of $\phi(\Omega, t)$. Define

$$R(\phi(B_\epsilon(\mathbf{Q}), t))$$

as this rest configuration for $\phi(B_\epsilon(\mathbf{Q}), t)$. We assume that there is a tensor $\boldsymbol{F}_p$ such that

$$R(\phi(B_\epsilon(\mathbf{Q}), t)) = \{\phi(\mathbf{Q}, t) + \boldsymbol{F}_p(\mathbf{Q}, t)d\mathbf{Y} \mid \|d\mathbf{Y}\| \le \epsilon\},$$

and a tensor $\boldsymbol{F}_e$ such that

$$\boldsymbol{F} = \boldsymbol{F}_e \boldsymbol{F}_p.$$

Note that if $\boldsymbol{F}_p = \mathbf{I}$, then

$$R(\phi(B_\epsilon(\mathbf{Q}), t)) = \{\phi(\mathbf{Q}, t) + d\mathbf{Y} \mid \|d\mathbf{Y}\| \le \epsilon\},$$

so that $R(\phi(B_\epsilon(\mathbf{Q}), t))$ is simply a translation of the reference configuration. This means that ϕ has not caused any permanent deformation in $B_\epsilon(\mathbf{Q})$. In another case, consider if $\boldsymbol{F}_p = \boldsymbol{F}$, so that

$$R(\phi(B_\epsilon(\mathbf{Q}), t)) = \{\phi(\mathbf{Q}, t) + \boldsymbol{F}(\mathbf{Q}, t)d\mathbf{Y} \mid \|d\mathbf{Y}\| \le \epsilon\}$$
$$\approx \phi(B_\epsilon(\mathbf{Q}), t).$$

This would indicate that the deformation ϕ has permanently deformed $B_\epsilon(\mathbf{Q})$ to $\phi(B_\epsilon(\mathbf{Q}), t)$. In general, if $\boldsymbol{F}_p$ is not a rotation, the deformation ϕ has permanently deformed $B_\epsilon(\mathbf{Q})$. We summarize the principle below.

Multiplicative Decomposition of the Deformation Gradient

Given a deformation $\boldsymbol{F}$, we assume that it can be multiplicatively decomposed into the product of a recoverable elastic part $\boldsymbol{F}_e$ and a plastic part $\boldsymbol{F}_p$ as $\boldsymbol{F} = \boldsymbol{F}_e \boldsymbol{F}_p$. We call $\boldsymbol{F}_e$ and $\boldsymbol{F}_p$ the elastic and plastic deformation gradient, respectively.

4.1.2 Strains

This section closely follows the treatment in [Bonet and Wood, 2008]. We note that for any rotation matrix $\boldsymbol{Z}$,

$$\{\phi(\mathbf{Q},t) + \boldsymbol{Z}\boldsymbol{F}_p(\mathbf{Q},t)d\mathbf{Y} \mid \|d\mathbf{Y}\| \leq \epsilon\}$$

is a valid rest configuration for $\phi(B_\epsilon(\mathbf{Q}),t)$, so we expect the treatment of plasticity to be equivalent for all $\boldsymbol{F}_p$ with respect to rotation on the left.

Now, we can write

$$\mathbf{G}_p = \boldsymbol{Z}\boldsymbol{F}_p$$
$$\mathbf{G}_e = \boldsymbol{F}_e \boldsymbol{Z}^T$$

so that $\boldsymbol{F} = \mathbf{G}_e \mathbf{G}_p$. Our strain tensors become

$$\hat{\mathbf{C}}_p := \mathbf{G}_p^T \mathbf{G}_p = \boldsymbol{F}_p^T \boldsymbol{Z}^T \boldsymbol{Z} \boldsymbol{F}_p = \boldsymbol{F}_p^T \boldsymbol{F}_p = \mathbf{C}_p$$
$$\hat{\mathbf{C}}_e := \mathbf{G}_e^T \mathbf{G}_e = \boldsymbol{Z}\boldsymbol{F}_e^T \boldsymbol{F}_e \boldsymbol{Z}^T = \boldsymbol{Z}\mathbf{C}_e \boldsymbol{Z}^T$$
$$\hat{\mathbf{b}}_e := \mathbf{G}_e \mathbf{G}_e^T = \boldsymbol{F}_e \boldsymbol{Z}^T \boldsymbol{Z} \boldsymbol{F}_e^T = \boldsymbol{F}_e \boldsymbol{F}_e^T = \mathbf{b}_e$$
$$\hat{\mathbf{b}}_p := \mathbf{G}_p \mathbf{G}_p^T = \boldsymbol{Z}\boldsymbol{F}_p \boldsymbol{F}_p^T \boldsymbol{Z}^T = \boldsymbol{Z}\mathbf{b}_p \boldsymbol{Z}^T$$

Hence the only meaningful strains in this treatment are the right plastic Cauchy-Green

strain $\mathbf{C}_p$ and the left elastic Cauchy-Green strain $\mathbf{b}_e$.

4.2 Plastic flow

This section closely follows [Bonet and Wood, 2008].

We assume that the stress power has an additive decomposition into elastic and plastic parts,

$$\Lambda := \tau : \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} = \Lambda_e + \Lambda_p$$

where we define

$$\Lambda_e := \tau : \frac{\partial \boldsymbol{F}_e(\mathbf{X}, t)}{\partial t} \boldsymbol{F}_e(\mathbf{X}, t)^{-1}.$$

4.2.1 Plastic rate of deformation

It can be shown that [Bonet and Wood, 2008]

$$\begin{aligned}
\Lambda &= \tau : \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \\
&= \tau : \left(\frac{1}{2} \left(\frac{D\boldsymbol{F}}{Dt} (\mathbf{C}_p)^{-1} \boldsymbol{F}^T + \boldsymbol{F} (\mathbf{C}_p)^{-1} \frac{D\boldsymbol{F}^T}{Dt} \right) \mathbf{b}_e^{-1} \right) \\
\Lambda_e &= \tau : \frac{\partial \boldsymbol{F}_e(\mathbf{X}, t)}{\partial t} \boldsymbol{F}_e(\mathbf{X}, t)^{-1} \\
&= \tau : \left(\left(\frac{1}{2} \frac{D}{Dt} \mathbf{b}_e \right) \mathbf{b}_e^{-1} \right)
\end{aligned}$$

so that

$$\begin{aligned}
\Lambda_p &= \Lambda - \Lambda_e \\
&= \tau : \left(-\frac{1}{2} \left(\boldsymbol{F} \frac{D}{Dt} [(\mathbf{C}_p)^{-1}] \boldsymbol{F}^T \right) \mathbf{b}_e^{-1} \right).
\end{aligned}$$

We define [Bonet and Wood, 2008]

$$\Theta = -\frac{1}{2}\left(\boldsymbol{F}\frac{D}{Dt}[(\mathbf{C}_p)^{-1}]\boldsymbol{F}^T \right)\mathbf{b}_e^{-1}$$

as the plastic rate of deformation. The plastic flow is then determined by how Θ relates to the stress in the material. Note that we can rearrange the above equation to get

$$\boldsymbol{F}\frac{D}{Dt}[(\mathbf{C}_p)^{-1}]\boldsymbol{F}^T = -2\Theta\mathbf{b}_e.$$

4.2.2 Plastic flow rule

Per [Bonet and Wood, 2008], plastic behavior is determined by considering a plastic flow rule for the left elastic Cauchy-Green strain, $\mathbf{b}_e := \boldsymbol{F}_e\boldsymbol{F}_e^T$. Recall that $\mathbf{C}_p = \boldsymbol{F}_p\boldsymbol{F}_p^T$, so that

$$\mathbf{b}_e = \boldsymbol{F}\mathbf{C}_p^{-1}\boldsymbol{F}^T.$$

Then

$$\frac{D\mathbf{b}_e}{Dt} = \frac{D\boldsymbol{F}}{Dt}(\mathbf{C}_p)^{-1}\boldsymbol{F}^T + \boldsymbol{F}(\mathbf{C}_p)^{-1}\frac{D\boldsymbol{F}^T}{Dt} + \boldsymbol{F}\frac{D}{Dt}[(\mathbf{C}_p)^{-1}]\boldsymbol{F}^T.$$

Recalling that $\frac{D\boldsymbol{F}}{Dt} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\boldsymbol{F}$, we get

$$\frac{D\mathbf{b}_e}{Dt} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{b}_e + \mathbf{b}_e\frac{\partial \mathbf{v}}{\partial \mathbf{x}}^T + \mathbf{g}(\mathbf{b}_e)$$

where

$$\mathbf{g}(\mathbf{b}_e) = \boldsymbol{F}\frac{D}{Dt}[(\mathbf{C}^P)^{-1}]\boldsymbol{F}^T$$

is the plastic flow rate. The expression for the material derivative of $\mathbf{b}_e$ above will allow us to write $\mathbf{b}_e$ as a function of grid velocities in Chapter 5 (similar to the treatment of the deformation gradient $\boldsymbol{F}$) and control its behavior via the plastic flow rate. g typically damps $\mathbf{b}_e$ to a quantity that will result in the loss of elastic resistance to deformation (e.g., in the Oldroyd-B case, this quantity is $\mathbf{I}$). As the elastic stress is a function of $\mathbf{b}_e$,

the derivatives of $\mathbf{b}_e$ with respect to velocities will manifest in the resulting forces on grid nodes. A detailed example is shown in Chapter 5.

4.2.3 Oldroyd-B

In the Oldroyd-B case,

$$\mathbf{g}(\mathbf{b}_e) = \frac{1}{W}(\mathbf{I} - \mathbf{b}_e),$$

which we can see corresponds to

$$\Theta = -\frac{1}{2W}(\mathbf{b}_e^{-1} - \mathbf{I}).$$

Note that in the method discussed in Chapter 5, Θ is never computed, as the method directly uses $\mathbf{g}(\mathbf{b}_e)$ to model plastic effects.

4.2.4 Bingham Plasticity

In the Bingham Plasticity case,

$$\mathbf{g}(\mathbf{b}_e) = -\frac{2}{3}\operatorname{tr}(\mathbf{b}_e)\max\left(0, \frac{\|\hat{\mathbf{b}}_e - \frac{1}{d}\operatorname{tr}(\hat{\mathbf{b}}_e)\mathbf{I}\| - \left(\frac{2}{3}\right)^{\frac{1}{2}}\sigma_Y}{\eta}\right)\frac{\hat{\mathbf{b}}_e - \frac{1}{d}\operatorname{tr}(\hat{\mathbf{b}}_e)\mathbf{I}}{\|\hat{\mathbf{b}}_e - \frac{1}{d}\operatorname{tr}(\hat{\mathbf{b}}_e)\mathbf{I}\|},$$

[Yue et al., 2015], where η and σ_Y are material parameters and $\hat{\mathbf{b}}_e = J^{-\frac{2}{3}}\mathbf{b}_e$.

4.2.4.1 Yield stress and viscosity

From [Yue et al., 2015], σ_Y is called the yield stress. We can see from the definition that $\mathbf{g}(\mathbf{b}_e) \neq 0$ only when the norm of the deviatoric part of $\hat{\mathbf{b}}_e$ is larger than $\left(\frac{2}{3}\right)^{\frac{1}{2}}\sigma_Y$. Thus, σ_Y directly controls how easily the material enters the plastic regime. The larger

the magnitude of σ_Y, the less likely the flow will be plastic. Conversely, the smaller the value of σ_Y, the more likely the flow will enter the plastic regime. η is the viscosity parameter and is directly related to the viscosity in the fluid.

4.2.5 Herschel-Bulkley

In the Herschel-Bulkley case,

$$\mathbf{g}(\mathbf{b}_e) = -\frac{2}{3}\operatorname{tr}(\mathbf{b}_e)\max\left(0, \frac{\|\hat{\mathbf{b}}_e - \frac{1}{d}\operatorname{tr}(\hat{\mathbf{b}}_e)\mathbf{I}\| - \left(\frac{2}{3}\right)^{\frac{1}{2}}\sigma_Y}{\eta}\right)^{\frac{1}{h}} \frac{\hat{\mathbf{b}}_e - \frac{1}{d}\operatorname{tr}(\hat{\mathbf{b}}_e)\mathbf{I}}{\|\hat{\mathbf{b}}_e - \frac{1}{d}\operatorname{tr}(\hat{\mathbf{b}}_e)\mathbf{I}\|},$$

[Yue et al., 2015] where η, σ_Y, and h are material parameters.

4.2.5.1 Effective viscosity

For the discussion of the parameter h, suppose that the flow is in the plastic regime (i.e., $\mathbf{g}(\mathbf{b}_e) \neq 0$) and isolate the term

$$a := \|\hat{\mathbf{b}}_e - \frac{1}{d}\operatorname{tr}(\hat{\mathbf{b}}_e)\mathbf{I}\| - \sigma_Y.$$

Then h varies the effect of $\frac{a}{\eta}$ on $\mathbf{g}(\mathbf{b}_e)$. As in [Yue et al., 2015], rewrite $\left(\frac{a}{\eta}\right)^{\frac{1}{h}}$ as

$$b := \frac{a}{\eta^{\frac{1}{h}}a^{1-\frac{1}{h}}}.$$

We can see that the rewrite of $\left(\frac{a}{\eta}\right)^{\frac{1}{h}}$ as

$$b := \frac{a}{\eta^{\frac{1}{h}}a^{1-\frac{1}{h}}}$$

demonstrates that Herschel-Bulkley flow corresponds to Bingham plasticity with viscosity determined by $\eta^{\frac{1}{h}} a^{1-\frac{1}{h}}$. This quantity

$$\eta^{\frac{1}{h}} a^{1-\frac{1}{h}}$$

is called the effective viscosity [Yue et al., 2015].

4.2.5.2 Shear Thinning Phenomena

Note that if $h < 1$, then $\frac{1}{h} > 1$ so that $1 - \frac{1}{h} < 0$ and we can write $a^{1-\frac{1}{h}} = \frac{1}{a^q}$ for some $q > 0$, so that the denominator of b becomes

$$\frac{\eta^{\frac{1}{h}}}{a^q}.$$

Hence, the the denominator of b decreases with an increase in the norm of the deviatoric part of $\hat{\mathbf{b}}_e$. That is, the greater the stress, the smaller the viscosity, and the fluid flows more easily. This models shear thinning phenomena [Yue et al., 2015].

4.2.5.3 Shear Thickening Phenomena

In the other case, $h > 1$, so we may write the denominator of b as

$$\eta^{\frac{1}{h}} a^r$$

for some $r > 0$. This means the denominator of b increases with an increase in the norm of the deviatoric part of $\hat{\mathbf{b}}_e$. That is, the greater the stress, the larger the viscosity, and the fluid flows less easily. This models shear thickening phenomena [Yue et al., 2015].

4.2.6 Von Mises condition

In the Von Mises model, Θ (hence $\mathbf{g}(\mathbf{b}_e)$) relates to the stress via the principle of maximum plastic dissipation. The principle of maximum plastic dissipation [Bonet and Wood, 2008] requires that the flow maximize the rate of plastic dissipation $\Lambda_p = \tau : \Theta$ subject to $f(\tau, \epsilon) \leq 0$.

Let $\tau' = \tau - \mathrm{tr}(\sigma_m) J\mathbf{I}$ be the deviatoric component of the Kirchhof stress. Von Mises plasticity defines a yield function

$$f(\tau, \epsilon) = \left(\frac{3}{2} \tau' : \tau' \right)^{\frac{1}{2}} - Y(\epsilon)$$

where $Y > 0$ is called the yield stress and ϵ is called the hardening parameter. f determines the elastic and plastic regimes for the material. The material behaves elastically as long as $f < 0$. Otherwise, the material flows plastically as $f = 0$ is satisfied.

As in [Simo, 1988], we treat this as constrained minimization problem. Define

$$H(\tau) = -\tau : \Theta + \gamma f(\tau, \epsilon).$$

Using the Karush-Kuhn-Tucker conditions, we see that the solution to the constrained minimization problem satisfies

$$\frac{\partial H}{\partial \tau} = 0, \gamma \geq 0 \text{ and } \gamma f(\tau, \epsilon) = 0.$$

As $\frac{\partial (\tau : \Theta)}{\partial \tau} = \Theta$, this becomes

$$\Theta = \gamma \frac{\partial f}{\partial \tau}, \gamma \geq 0 \text{ and } \gamma f(\tau, \epsilon) = 0.$$

$\Theta = \gamma \frac{\partial f}{\partial \tau}$ explicitly relates the plastic flow to the stress. This allows us to calculate the

plastic flow as we are able to compute the stress. For details on computing γ, see [Simo, 1988], [Bonet and Wood, 2008].

4.3 Volume Preserving Plasticity for Von Mises Model

With the plastic flow rule defined above, we get [Bonet and Wood, 2008]

$$\Theta = \gamma \left(\frac{3}{2}\right)^{\frac{1}{2}} \frac{\tau'}{(\tau' : \tau')^{\frac{1}{2}}}.$$

Let

$$\mathbf{g}(\mathbf{b}_e) := -2\Theta\mathbf{b}_e,$$

which is our plastic flow from above. Note that $\mathrm{tr}(\Theta) = 0$ since τ' is the deviatoric part of τ.

Now,

$$
\begin{aligned}
\frac{d}{dt}(\det(\mathbf{b}_e)) &= \det(\mathbf{b}_e)\mathbf{b}_e^{-1} : \frac{d}{dt}\mathbf{b}_e \\
&= \det(\mathbf{b}_e)\,\mathrm{tr}\left(\mathbf{b}_e^{-1}\left(\frac{\partial \mathbf{V}}{\partial \mathbf{X}}\boldsymbol{F}^{-1}\mathbf{b}_e + \mathbf{b}_e\left(\frac{\partial \mathbf{V}}{\partial \mathbf{X}}\right)^T + \mathbf{g}(\mathbf{b}_e)\right)^T\right) \\
&= \det(\mathbf{b}_e)\left(2\,\mathrm{tr}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\right) + \mathrm{tr}\left(\mathbf{b}_e^{-1}\mathbf{g}(\mathbf{b}_e)^T\right)\right) \\
&= \det(\mathbf{b}_e)\left(2\,\mathrm{tr}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\right) + \mathrm{tr}\left(-\mathbf{b}_e^{-1}\mathbf{b}_e 2\Theta^T\right)\right) \\
&= 2\det(\mathbf{b}_e)\left(\mathrm{tr}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\right)\right) \\
&= 2\det(\mathbf{b}_e)\nabla \cdot \boldsymbol{v}
\end{aligned}
$$

Define

$$J_p := \det(\boldsymbol{F}_p).$$

Also note that

$$
\begin{aligned}
\frac{d}{dt}(J^2) &= 2J\frac{dJ}{dt} \\
&= 2J\frac{\partial J}{\partial \boldsymbol{F}} : \frac{\partial \boldsymbol{F}}{\partial t} \\
&= 2J^2 \boldsymbol{F}^{-T} : \frac{\partial \boldsymbol{F}}{\partial t} \\
&= 2J^2 \,\mathrm{tr}\left(\boldsymbol{F}^{-T}\left(\frac{\partial \boldsymbol{F}}{\partial t}\right)^T\right) \\
&= 2J^2 \,\mathrm{tr}\left(\boldsymbol{F}^{-T}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\boldsymbol{F}\right)^T\right) \\
&= 2J^2 \,\mathrm{tr}\left(\boldsymbol{F}^{-T}\boldsymbol{F}^T\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\right)^T\right) \\
&= 2J^2 \,\mathrm{tr}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\right) \\
&= 2J^2 \nabla \cdot \boldsymbol{v}
\end{aligned}
$$

Hence, $J^2 = \det(\mathbf{b}_e)$, and since $\det(\boldsymbol{F}) = J$, $\mathbf{b}_e = \boldsymbol{F}_e\boldsymbol{F}_e^T$ and $\boldsymbol{F} = \boldsymbol{F}_e\boldsymbol{F}_p$, we get that $J_p = 1$, so the flow is isochoric.

4.4 Plasticity via the SVD

A common technique [Stomakhin et al., 2013] for enforcing plastic behaviors of a material is to clamp the singular values of $\boldsymbol{F}_e$. This does not use a plastic flow rate but does take advantage of the multiplicative decomposition of the deformation gradient. First, perform a singular value decomposition of $\boldsymbol{F}_e$ to get

$$
\boldsymbol{F}_e = \mathbf{U}\Sigma\mathbf{V}^T.
$$

Let $c, s > 0$. The parameters c and s will determine how much compression and stretching the material is allowed to undergo before causing permanent deformation in the

object. Assume that the diagonal entries of Σ are $\sigma_1, \sigma_2, \sigma_3$. Then, define

$$\hat{\Sigma} = \text{diag}(\text{clamp}_{(1-c,1+s)}(\sigma_1), \text{clamp}_{(1-c,1+s)}(\sigma_2), \text{clamp}_{(1-c,1+s)}(\sigma_3)),$$

and set

$$\boldsymbol{F}_e \leftarrow \mathbf{U}\hat{\Sigma}\mathbf{V}^T$$

$$\boldsymbol{F}_p \leftarrow \mathbf{V}\hat{\Sigma}^{-1}\mathbf{U}^T\boldsymbol{F}$$

which will preserve $\boldsymbol{F} = \boldsymbol{F}_e\boldsymbol{F}_p$. Assuming $\hat{\Sigma} \neq \Sigma$, $\boldsymbol{F}_p$ will map to a new rest state. As a result, this modification transfers meaningful deformation information from $\boldsymbol{F}_e$ to $\boldsymbol{F}_p$, resulting in the loss of deformation resistance in the material.

4.5   Material softening and hardening

4.5.1   Hardening

The constitutive model in [Stomakhin et al., 2013] uses plastically-varying Lamé parameters to simulate material hardening under compression. In particular, they use

$$\mu(\boldsymbol{F}_p) = \mu_0 e^{\zeta(1-\det(\boldsymbol{F}_p))}$$

$$\lambda(\boldsymbol{F}_p) = \lambda_0 e^{\zeta(1-\det(\boldsymbol{F}_p))}$$

where $\zeta > 0$ and controls the severity of hardening. If a material is undergoing plastic compression, say $0 < \det(\boldsymbol{F}_p^{n+1}) < \det(\boldsymbol{F}_p^n) < 1$, then

$$\zeta(1 - \det(\boldsymbol{F}_p^n)) < \zeta(1 - \det(\boldsymbol{F}_p^{n+1})),$$

so μ, λ are increasing, hence the material is hardening.

4.5.2 Softening

We present a novel modification to the material hardening treatment. Note that we can also simulate material softening under compression by considering $\zeta < 0$. If a material is undergoing plastic compression, say $0 < \det(\boldsymbol{F}_p^{n+1}) < \det(\boldsymbol{F}_p^n) < 1$, then (as $\zeta < 0$)

$$\zeta(1 - \det(\boldsymbol{F}_p^{n+1})) < \zeta(1 - \det(\boldsymbol{F}_p^n))$$

so μ, λ are decreasing, hence the material is softening. An example with this modification from [Chong et al.] is shown in Figure 4.1.

4.6 b_e update that preserves symmetry and positive-definiteness of strain

As $\mathrm{b}_e := \boldsymbol{F}_e \boldsymbol{F}_e^T$, it is easy to see that b_e is a symmetric positive-definite matrix. The temporal update of b_e requires special care in order to ensure that b_e remain symmetric positive-definite for all time.

The update used in Chapter 5 can be improved with an update that preserves the symmetry and positive-definiteness of the strain for all time. As noted in that chapter, modifications to the algorithm were required in order to avoid inverted configurations of b_e. These modifications would not be necessary if a better update were used. This section will describe novel modifications that preserve symmetry and positive-definiteness of the strain.

First, we note that in the case of no plasticity, $\boldsymbol{F} = \boldsymbol{F}_e$, so that the update for $\boldsymbol{F}_e$ would be

$$\boldsymbol{F}_e^{n+1} = \left(\mathbf{I} + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1}\right) \boldsymbol{F}_e^n$$
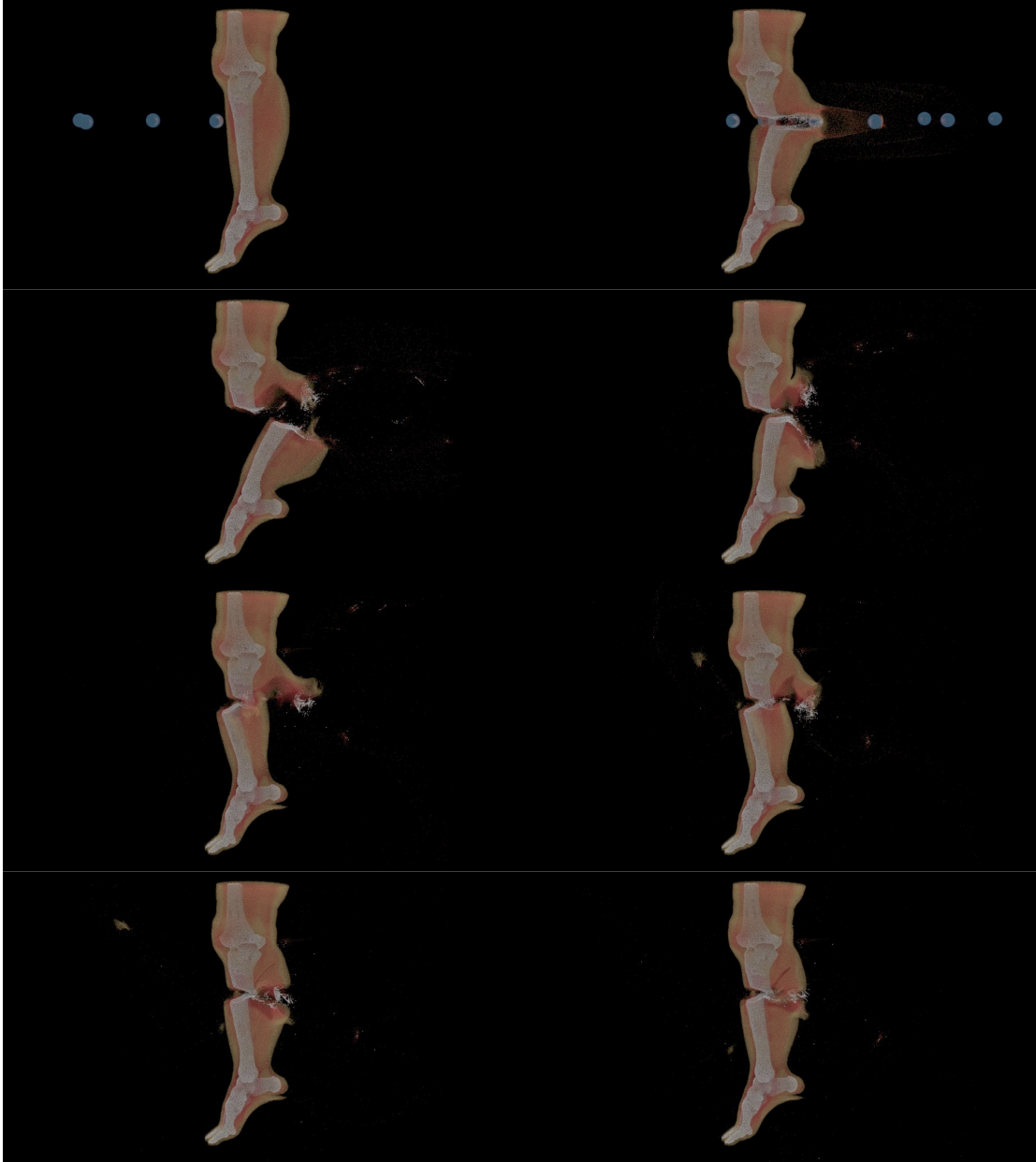
Figure 4.1: *Bullets shoot at a leg. The bone is modeled using the material softening modification from Chapter 4.*

and the corresponding update for $\mathbf{b}_e$ is

$$
\begin{aligned}
\mathbf{b}_e^{n+1} &= \boldsymbol{F}_e^{n+1}\left(\boldsymbol{F}_e^{n+1}\right)^T \\
&= \left(\mathbf{I} + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1}\right) \mathbf{b}_e^n \left(\mathbf{I} + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1}\right)^T \\
&= \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q}
\end{aligned}
$$

by setting

$$
\mathbf{Q} := \left(\mathbf{I} + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1}\right)^T.
$$

We immediately note that $\mathbf{b}_e^{n+1}$ is symmetric positive-definite assuming that $\mathbf{b}_e^n$ is symmetric positive-definite. Furthermore, in the case of no plasticity, this differs from the update used in Chapter 5 by $\Delta t^2 \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \mathbf{b}_e^n \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^T$, so adding this additional term to the right-hand side of the original update remains consistent. In what follows, we use this to define symmetric positive-definite updates for $\mathbf{b}_e$ with different plastic flow rates.

4.6.1 Oldroyd-B

In the Oldroyd-B case, $\mathbf{g}(\mathbf{b}_e) = \frac{1}{Wi}(\mathbf{I} - \mathbf{b}_e)$, where Wi is the Weissenberg number. If we use the consistent discretization

$$
\begin{aligned}
\frac{1}{\Delta t}(\mathbf{b}_e^{n+1} - \mathbf{b}_e^n) &= \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1} \mathbf{b}_e^n + \left(\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1} \mathbf{b}_e^n\right)^T + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1} \mathbf{b}_e^n \left(\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1}\right)^T \\
&\quad + \frac{1}{Wi}(\mathbf{I} - \mathbf{b}_e^{n+1}),
\end{aligned}
$$

then we can see that

$$
\left(1 + \frac{\Delta t}{Wi}\right) \mathbf{b}_e^{n+1} = \frac{\Delta t}{Wi}\mathbf{I} + \left(\mathbf{I} + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1}\right) \mathbf{b}_e^n \left(\mathbf{I} + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1}\right)^T.
$$

$Wi > 0$ so that $1 + \frac{\Delta t}{Wi} > 0$. For convenience, set $a = \frac{1}{1 + \frac{\Delta t}{Wi}}$, $\mathbf{Q} = \left(\mathbf{I} + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}} \right)^{n+1} \right)^T$. For symmetry, note that

$$\mathbf{b}_e^{n+1} = a \left(\frac{\Delta t}{Wi} \mathbf{I} + \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q} \right)$$

is the sum of symmetric matrices assuming $\mathbf{b}_e^n$ is symmetric, so is symmetric.

For positive-definiteness,

$$\mathbf{b}_e^{n+1} = a \left(\frac{\Delta t}{Wi} \mathbf{I} + \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q} \right) \Rightarrow$$

$$\boldsymbol{v}^T (\mathbf{b}_e^{n+1}) \boldsymbol{v} = a \left(\boldsymbol{v}^T \frac{\Delta t}{Wi} \mathbf{I} \boldsymbol{v} + \boldsymbol{v}^T \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q} \boldsymbol{v} \right) \Rightarrow$$

$$\boldsymbol{v}^T (\mathbf{b}_e^{n+1}) \boldsymbol{v} = a \left(\frac{\Delta t}{Wi} \boldsymbol{v}^T \boldsymbol{v} + (\mathbf{Q}\boldsymbol{v})^T \mathbf{b}_e^n \mathbf{Q} \boldsymbol{v} \right) \Rightarrow$$

$$\boldsymbol{v}^T (\mathbf{b}_e^{n+1}) \boldsymbol{v} > 0, \text{assuming } \mathbf{b}_e^n \text{ is positive} - \text{definite and } \boldsymbol{v} \neq 0.$$

As $\mathbf{b}_e^0 = \boldsymbol{F}_e^0 (\boldsymbol{F}_e^0)^T$ is symmetric positive-definite, we get that $\mathbf{b}_e^n$ is symmetric positive-definite for all n.

4.6.2 Herschel-Bulkley-like fluid

In a Herschel-Bulkley-like case,

$$\mathbf{g}(\mathbf{b}_e) = -\frac{2}{3} \text{tr}(\mathbf{b}_e) \max \left(0, \frac{\| \mathbf{b}_e - \frac{1}{d} \text{tr}(\mathbf{b}_e) \mathbf{I} \| - \sigma_Y}{\eta} \right)^{\frac{1}{h}} \frac{\mathbf{b}_e - \frac{1}{d} \text{tr}(\mathbf{b}_e) \mathbf{I}}{\| \mathbf{b}_e - \frac{1}{d} \text{tr}(\mathbf{b}_e) \mathbf{I} \|}.$$

Note that we have replaced $\hat{\mathbf{b}}_e$ with $\mathbf{b}_e$, as in a later example, we will use the analogous strain density (with $\mathbf{b}_e := \left(\frac{J}{J_{HB}} \right)^{\frac{2}{3}} \mathbf{b}_e^{HB}$, where $\mathbf{b}_e^{HB}$ satisfies the flow rule above and

46

$J_{HB} = \sqrt{\det(\mathbf{b}_e^{HB})})$ from Chapter 5. We discretize this as

$$\mathbf{b}_e^{n+1} = \mathbf{b}_e^n + \Delta t \left(\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \right)^{n+1} \mathbf{b}_e^n + \mathbf{b}_e^n \left(\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \right)^{n+1} \right)^T \right) + \Delta t^2 \left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \right)^{n+1} \mathbf{b}_e^n \left(\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \right)^{n+1} \right)^T$$

$$- \frac{2\Delta t}{3} \operatorname{tr}(\mathbf{b}_e^n) \max \left(0, \frac{\|\mathbf{b}_e^n - \frac{1}{d}\operatorname{tr}(\mathbf{b}_e^n)\mathbf{I}\| - \sigma_Y}{\eta} \right)^{\frac{1}{h}} \frac{\mathbf{b}_e^{n+1} - \frac{1}{d}\operatorname{tr}(\mathbf{b}_e^n)\mathbf{I}}{\|\mathbf{b}_e^n - \frac{1}{d}\operatorname{tr}(\mathbf{b}_e^n)\mathbf{I}\|}.$$

With $\mathbf{Q}$ as defined before, set

$$a := \frac{2\Delta t}{3} \operatorname{tr}(\mathbf{b}_e^n) \max \left(0, \frac{\|\mathbf{b}_e^n - \frac{1}{d}\operatorname{tr}(\mathbf{b}_e^n)\mathbf{I}\| - \sigma_Y}{\eta} \right)^{\frac{1}{h}} \|\mathbf{b}_e^n - \frac{1}{d}\operatorname{tr}(\mathbf{b}_e^n)\mathbf{I}\|^{-1}$$

$$b = \frac{1}{d} \operatorname{tr}(\mathbf{b}_e^n)$$

$$c = (1+a)^{-1}.$$

For symmetry, note that

$$(1+a)\mathbf{b}_e^{n+1} = \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q} + ab\mathbf{I}$$

is the sum of symmetric matrices assuming $\mathbf{b}_e^n$ is symmetric, so is symmetric.

Note that if $\mathbf{b}_e^n$ is positive-definite, then $a \geq 0, b > 0, c > 0$ and

$$\mathbf{b}_e^{n+1} = \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q} - a\mathbf{b}_e^{n+1} + ab\mathbf{I}$$

$$(1+a)\mathbf{b}_e^{n+1} = \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q} + ab\mathbf{I}$$

$$\mathbf{b}_e^{n+1} = c(ab\mathbf{I} + \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q}) \Rightarrow$$

$$\boldsymbol{v}^T \mathbf{b}_e^{n+1} \boldsymbol{v} = c(ab\boldsymbol{v}^T \boldsymbol{v} + \boldsymbol{v}^T \mathbf{Q}^T \mathbf{b}_e^n \mathbf{Q}\boldsymbol{v}) \Rightarrow$$

$$\boldsymbol{v}^T \mathbf{b}_e^{n+1} \boldsymbol{v} = c(ab\boldsymbol{v}^T \boldsymbol{v} + (\mathbf{Q}\boldsymbol{v})^T \mathbf{b}_e^n \mathbf{Q}\boldsymbol{v}) \Rightarrow$$

$$\boldsymbol{v}^T \mathbf{b}_e^{n+1} \boldsymbol{v} > 0, \text{assuming } \mathbf{b}_e^n \text{ is positive} - \text{definite and } \boldsymbol{v} \neq 0.$$

As $\mathbf{b}_e^0 = \boldsymbol{F}_e^0(\boldsymbol{F}_e^0)^T$, it is symmetric positive-definite, so that $\mathbf{b}_e^n$ is symmetric positive-definite for all n.

An example that uses this symmetric positive-definite update with the analogous strain density (with $\mathbf{b}_e := \left(\frac{J}{J_{HB}}\right)^{\frac{2}{3}} \mathbf{b}_e^{HB}$, where $\mathbf{b}_e^{HB}$ satisfies the flow rule above and $J_{HB} = \sqrt{\det(\mathbf{b}_e^{HB})}$) from Chapter 5 can be seen in Figure 4.2. As the method detailed in Chapter 5 required special care when $\mathbf{b}_e$ inverts, modifying the $\mathbf{b}_e$ update to guarantee that $\mathbf{b}_e$ is symmetric positive-definite benefits the method.

4.7 Recast $\mathbf{b}_e$ update in terms of $\mathbf{f}_e$

Define $\mathbf{f}_e$ as satisfying $\dot{\mathbf{f}}_e = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{f}_e + \frac{1}{2Wi}(\mathbf{f}_e^{-T} - \mathbf{f}_e)$. Then

$$\dot{\mathbf{f}}_e\mathbf{f}_e^T = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{f}_e\mathbf{f}_e^T + \frac{1}{2Wi}(\mathbf{I} - \mathbf{f}_e\mathbf{f}_e^T)$$

$$\mathbf{f}_e\dot{\mathbf{f}}_e^T = \mathbf{f}_e\mathbf{f}_e^T\frac{\partial \mathbf{v}}{\partial \mathbf{x}}^T + \frac{1}{2Wi}(\mathbf{I} - \mathbf{f}_e\mathbf{f}_e^T).$$

so that

$$\frac{D(\mathbf{f}_e\mathbf{f}_e^T)}{Dt} = \dot{\mathbf{f}}_e\mathbf{f}_e^T + \mathbf{f}_e\dot{\mathbf{f}}_e^T$$

$$= \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{f}_e\mathbf{f}_e^T + \mathbf{f}_e\mathbf{f}_e^T\frac{\partial \mathbf{v}}{\partial \mathbf{x}}^T + \frac{1}{Wi}(\mathbf{I} - \mathbf{f}_e\mathbf{f}_e^T).$$

Recall that

$$\dot{\mathbf{b}}_e = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{b}_e + \mathbf{b}_e\frac{\partial \mathbf{v}}{\partial \mathbf{x}}^T + \frac{1}{Wi}(\mathbf{I} - \mathbf{b}_e).$$

As $\mathbf{f}_e\mathbf{f}_e^T$ and $\mathbf{b}_e$ satisfy the same differential equation, we must have that $\mathbf{b}_e = \mathbf{f}_e\mathbf{f}_e^T$ with this definition of $\mathbf{f}_e$.
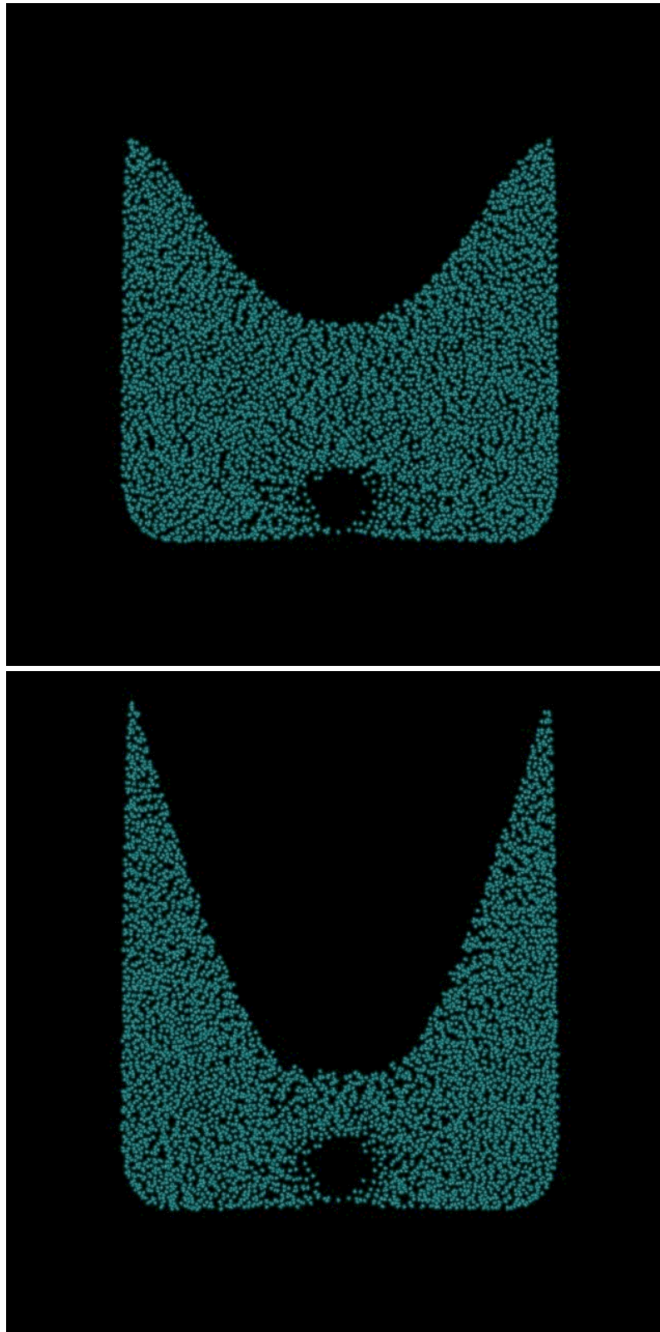
Figure 4.2: *Cross sections of Herschel-Bulkley-like fluids in containers. A cylinder spins at high speed at the bottom of the container. These simulations use the $\mathbf{b}_e$ update that preserves symmetry and positive-definiteness of $\mathbf{b}_e$ and the analogous strain density from Chapter 5. Fluid on the top uses $\sigma_Y = .1, h = 1.0101, \eta = 1$, fluid on the bottom uses $\sigma_Y = 1e - 6, h = 10, \eta = 1$.*

4.7.1   Discretization of f_e in the Material Point Method case

This suggests a separate approach to maintaining symmetry and positive-definiteness of $\mathbf{b}_e$. Note that

$$\mathbf{g} = \mathbf{h} + \mathcal{O}(\Delta t^m) \Rightarrow \mathbf{g}\mathbf{g}^T = \mathbf{h}\mathbf{h}^T + \mathcal{O}(\Delta t^m).$$

So, to update $\mathbf{b}_e$, we simply require an update for $\mathbf{f}_e$.

As an example for a Material Point Method update, we can use

$$\frac{1}{\Delta t}\left(\boldsymbol{F}_e^{n+1} - \boldsymbol{F}_e^n\right) = \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1} \boldsymbol{F}_e^n + \frac{1}{2W}\left((\boldsymbol{F}_e^n)^{-T} - \boldsymbol{F}_e^n\right)$$

from which we get

$$\boldsymbol{F}_e^{n+1} = \boldsymbol{F}_e^n + \Delta t \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)^{n+1} \boldsymbol{F}_e^n + \frac{\Delta t}{2W}\left((\boldsymbol{F}_e^n)^{-T} - \boldsymbol{F}_e^n\right).$$

Finally, for updating $\mathbf{b}_e$, we simply use $\mathbf{b}_e^{n+1} = \boldsymbol{F}_e^{n+1}(\boldsymbol{F}_e^{n+1})^T$, which will guarantee that $\mathbf{b}_e$ is symmetric positive-definite.

CHAPTER 5

The Material Point Method for Viscoelastic Fluids, Foams and Sponges

5.1 Abstract

We present a new Material Point Method (MPM) for simulating viscoelastic fluids, foams and sponges. We design our discretization from the upper convected derivative terms in the evolution of the left Cauchy-Green elastic strain tensor. We combine this with an Oldroyd-B model for plastic flow in a complex viscoelastic fluid. While the Oldroyd-B model is traditionally used for viscoelastic fluids, we show that its interpretation as a plastic flow naturally allows us to simulate a wide range of complex material behaviors. In order to do this, we provide a modification to the traditional Oldroyd-B model that guarantees volume preserving plastic flows. Our plasticity model is remarkably simple (foregoing the need for the singular value decomposition (SVD) of stresses or strains). Lastly, we show that implicit time stepping can be achieved in a manner similar to [Stomakhin et al., 2013] and that this allows for high resolution simulations at practical simulation times.

5.2 Introduction

Non-Newtonian fluid behavior is exhibited by a wide range of everyday materials including paint, gels, sponges, foams and various food components like ketchup and custard [Larson, 1999]. These materials are often special kinds of colloidal systems (a type of mixture in which one substance is dispersed evenly throughout another), where dimensions exceed those usually associated with colloids (up to 1μm for the dispersed phase) [Hiemenz and Rajagopalan, 1997; Larson, 1999]. For example, when a gas and a liquid are shaken together, the gas phase becomes a collection of bubbles dispersed in the liquid: this is the most common observation of foams. While a standard Newtonian viscous stress is a component of the mechanical response of these materials, they are non-Newtonian in the sense that there are other, often elastoplastic, aspects of the stress response to flow rate and deformation. Comprehensive reviews are given in [Morrison and Ross, 2002; Prudhomme and Kahn, 1996; Schramm, 1994; Larson, 1999].

Discretization of these materials is challenging because of the wide range of behaviors exhibited and by the nonlinear governing equations. These materials can behave with elastic resistance to deformation but can also undergo very large strains and complex topological changes characteristic of fluids. While Lagrangian approaches are best for resolving the solid-like behavior and Eulerian approaches most easily resolve the fluid-like behavior, these materials are in the middle ground and this makes discretization difficult. The Material Point Method is naturally suited for this class of materials because it uses a Cartesian grid to resolve topology changes and self-collisions combined with Lagrangian tracking of mass, momentum and deformation on particles. In practice, the particle-wise deformation information can be used to represent elastoplastic stresses arising from changes in shape, while an Eulerian background grid is used for implicit solves.

We show that the MPM approach in [Stomakhin et al., 2013] can be generalized to

achieve a wide range of viscoelastic, complex fluid effects. As in [Stomakhin et al., 2013], we show that implicit time stepping can easily be used to improve efficiency and allow for simulation at high spatial resolution. With our Oldroyd-inspired approach, we avoid the need for the SVD of either elastic or plastic responses. While SVD computation is not a bottleneck for MPM when done efficiently (see e.g [McAdams et al., 2011]), it is not a straightforward implementation. More standard SVD implementations can have a dramatic impact on performance (see e.g. [Chao et al., 2010]). Thus although it is not essential for performance to avoid the SVD, it is preferable to avoid the need to implement them when, as with our model, they are not necessary for achieving desired behaviors.

We summarize our specific contributions as

- A new volume-preserving Oldroyd-B rate-based description of plasticity

- Semi-implicit MPM discretization of viscoelasticity and viscoplasticity, allowing for high spatial resolution simulations

- Rate-based plasticity that does not require an SVD

5.3   Related work

Terzopoulos and Fleischer were the first in computer graphics to show the effects possible with simulated elastoplastic materials [Terzopoulos and Fleischer, 1988a,b]. Since those seminal works, many researchers have developed novel methods capable of replicating a wide range of material behaviors. Generally, these fall into one of three categories: Eulerian grid, Lagrangian mesh or particle based techniques. In addition to the following discussion, we summarize some aspects of our approach relative to a few representative approaches in Table 5.3.

Eulerian grid based approaches: Goktekin et al. [Goktekin et al., 2004] showed that the addition of an Eulerian elastic stress with Von Mises criteria plasticity to the standard level set based simulation of free surface Navier Stokes flows can capture a wide range of viscoelastic behaviors. Losasso et al. also use an Eulerian approach [Losasso et al., 2006]. Rasmussen et al. experiment with a range of viscous effects for level set based free surface melting flows in [Rasmussen et al., 2004]. Batty et al. use Eulerian approaches to efficiently simulate spatially varying viscous coiling and buckling [Batty and Bridson, 2008; Batty and Houston, 2011]. Carlson et al. also achieve a range of viscous effects in [Carlson et al., 2002].

Lagrangian mesh based approaches: Lagrangian methods naturally resolve deformation needed for elastoplasticity; however, large strains can lead to mesh tangling for practical flow scenarios and remeshing is required. Bargteil et al. show that this can achieve impressive results in [Bargteil et al., 2007]. This was later extended to embedded meshes in [Wojtan and Turk, 2008] and further treatment of splitting and merging was achieved in [Wojtan et al., 2009]. Batty et al. used a reduced dimension approach to simulate thin viscous sheets with adaptively remeshed triangle meshes in [Batty et al., 2012].

Particle Methods: Ever since Desbrun and Gascuel [Desbrun and Gascuel, 1996] showed that SPH can be used for a range of viscous behavior, particle methods have been popular for achieving complex fluid effects. Like Goktekin et al., Chang et al. [Chang et al., 2009] also use an Eulerian update of the strain for elastoplasic SPH simulations. Solenthaler et al. show that SPH can be used to compute strain and use this to get a range of elastoplastic effects [Solenthaler et al., 2007]. Becker et al. show that this can be generalized to large rotational motion in [Becker et al., 2009]. Gerszewski et al. also update deformation directly on particles [Gerszewski et al., 2009]. [Keiser et al., 2005] and [Müller et al., 2004] also add elastic effects into SPH formulations. Paiva et al. use a non-Newtonian model for fluid viscosity in [Paiva et al., 2006] and

[Paiva et al., 2009].

Although MPM is a hybrid grid/particle method, particles are arguably the primary material representation. MPM has recently been used to simulate elastoplastic flows to capture snow in [Stomakhin et al., 2013] and varied, melting materials in [Stomakhin et al., 2014]. Yue et al. use MPM to simulate Herschel-Bulkley plastic flows for foam in [Yue et al., 2015]. Their approach is very similar to ours, however their treatment of plasticity is much more accurate and can handle a wider range of phenomena (notably, shear thickening). They also provide a novel particle splitting technique useful for resolving shearing flows that are problematic for a wide range of MPM simulations. However, their plastic flow update is more complicated and this is likely why they resort to explicit time stepping. With our comparatively simple plastic flow model, we show that semi-implicit time stepping as in [Stomakhin et al., 2013] can be achieved.

5.4 Governing equations

The governing equations arise from basic conservation of mass and momentum as

$$
\begin{aligned}
\frac{D}{Dt}\rho + \rho\nabla \cdot \boldsymbol{v} &= 0 \\
\rho\frac{D}{Dt}\boldsymbol{v} &= \nabla \cdot \boldsymbol{\sigma} + \rho\mathbf{g}
\end{aligned}
\tag{5.1}
$$

where ρ is the mass density, $\boldsymbol{v}$ is the velocity, $\boldsymbol{\sigma}$ is the Cauchy stress and $\mathbf{g}$ is gravitational acceleration. As is commonly done with viscoelastic complex fluids, we write the Cauchy stress as $\boldsymbol{\sigma} = \boldsymbol{\sigma}^N + \boldsymbol{\sigma}^E$ where

$$
\boldsymbol{\sigma}^N = \frac{\mu^N}{2}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}^T\right)
$$

is the viscous Newtonian component and $\boldsymbol{\sigma}^E$ is the elastic component. We express the

constitutive behavior through the elastic component of the left Cauchy-Green strain. Specifically, the deformation gradient of the flow $\boldsymbol{F}$ can be decomposed as a product of elastic and plastic deformation as

$$\boldsymbol{F} = \boldsymbol{F}^E \boldsymbol{F}^P$$

and the elastic left Cauchy-Green strain is

$$\mathbf{b}^E = \boldsymbol{F}^E (\boldsymbol{F}^E)^T$$

[Bonet and Wood, 2008]. With this convention, we can define the elastic portion of the Cauchy stress via the stored elastic potential $\psi(\mathbf{b}^E)$ as

$$\boldsymbol{\sigma}^E = \frac{2}{J} \frac{\partial \psi}{\partial \mathbf{b}^E} \mathbf{b}^E.$$

5.4.1 Left Cauchy-Green strain plasticity and the upper convected derivative

We can define the plastic flow using the temporal evolution of the elastic right Cauchy-Green strain as in [Bonet and Wood, 2008]. Rewriting

$$\boldsymbol{F}^E = \boldsymbol{F}(\boldsymbol{F}^P)^{-1}, \mathbf{b}^E = \boldsymbol{F}(\mathbf{C}^P)^{-1} \mathbf{F}^T$$

where $\mathbf{C}^P = (\boldsymbol{F}^P)^T \boldsymbol{F}^P$ is the right plastic Cauchy-Green strain. The Eulerian form of the temporal evolution is then obtained by taking the material derivative of $\mathbf{b}^E$ to get

$$\frac{D\mathbf{b}^E}{Dt} = \frac{D\boldsymbol{F}}{Dt}(\mathbf{C}^P)^{-1}\boldsymbol{F}^T + \boldsymbol{F}(\mathbf{C}^P)^{-1}\frac{D\boldsymbol{F}^T}{Dt} + \boldsymbol{F}\frac{D}{Dt}[(\mathbf{C}^P)^{-1}]\boldsymbol{F}^T. \qquad (5.2)$$

With this view, the plastic flow is defined via $\frac{D}{Dt}[(\mathbf{C}^P)^{-1}]$. Combining this with

$$\frac{D}{Dt}\boldsymbol{F} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\boldsymbol{F}$$

(see e.g. [Bonet and Wood, 2008], Chapter 2), the previous equation can be rewritten as

$$\frac{D\mathbf{b}^E}{Dt} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{b}^E + \mathbf{b}^E\frac{\partial \mathbf{v}}{\partial \mathbf{x}}^T + \mathbf{g}(\mathbf{b}^E) \tag{5.3}$$

where

$$\mathbf{g}(\mathbf{b}^E) = \boldsymbol{F}\frac{D}{Dt}[(\mathbf{C}^P)^{-1}]\boldsymbol{F}^T$$

is used to describe the plastic flow rate. This equation is often abbreviated as

$$\overset{\triangledown}{\mathbf{b}}{}^E = \mathbf{g}(\mathbf{b}^E). \tag{5.4}$$

Here, the operator $\overset{\triangledown}{\mathbf{b}}{}^E$ (often referred to as the upper convected derivative) is defined to be

$$\overset{\triangledown}{\mathbf{b}}{}^E \equiv \frac{D}{Dt}\mathbf{b}^E - \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{b}^E - \mathbf{b}^E\frac{\partial \mathbf{v}}{\partial \mathbf{x}}^T$$

(see e.g. [Larson, 1999]).

5.4.2   Von Mises plasticity

The Von Mises model [Bonet and Wood, 2008] achieves plasticity through the rate

$$\mathbf{g}(\mathbf{b}^E) = -2\dot{\gamma}\delta\frac{\partial f(\tau)}{\partial \tau}\mathbf{b}^E,$$

where τ is the Kirchhoff stress, $\dot{\gamma}$ is the plastic multiplier, $f(\tau)$ is the Von Mises yield condition, and $\delta = 1$ if $f(\tau) \geq 0$, $\delta = 0$ otherwise. However, this is relatively difficult to discretize given the conditional nature of the function. It is often more straightforward to just work directly with $\boldsymbol{F}^E$ and $\boldsymbol{F}^P$ in that case (see e.g [Stomakhin et al., 2013]), however Yue et al [Yue et al., 2015] do discretize this directly.

5.4.3 Oldroyd-B plasticity

The Oldroyd-B model [Larson, 1999; Teran et al., 2008] can be see as an alternative definition of

$$\mathbf{g}(\mathbf{b}^E) = \frac{1}{Wi}(\mathbf{I} - \mathbf{b}^E).$$

Combining this with $\mathbf{g}(\mathbf{b}^E) = \boldsymbol{F}\frac{D}{Dt}[(\mathbf{C}^P)^{-1}]\boldsymbol{F}^T$ shows that the plastic flow of this model is

$$\frac{D}{Dt}[(\mathbf{C}^P)^{-1}] = \frac{1}{Wi}(\mathbf{C}^{-1} - (\mathbf{C}^P)^{-1})$$

where $\mathbf{C} = \mathbf{F}^T\mathbf{F}$ is the right Cauchy-Green strain. This expression for $\mathbf{g}(\mathbf{b}^E)$ is very simple in comparison with that of Von Mises. This simplicity allows for a much easier treatment of temporal discretization needed for implicit time stepping. Specifically, we show in Section 5.5 that this simple definition of $\mathbf{g}(\mathbf{b}^E)$ facilitates the implicit description of the plastic flow in terms of discrete grid node velocities. We can see, both from the $\frac{1}{Wi}(\mathbf{I} - \mathbf{b}^E)$ and $\frac{D}{Dt}[(\mathbf{C}^P)^{-1}] = \frac{1}{Wi}(\mathbf{C}^{-1} - (\mathbf{C}^P)^{-1})$ terms that the plasticity achieves a strong damping of the elastic component of the stress. The severity of this damping is inversely proportionate to the Weissenberg number Wi. That is, the smaller the Weissenberg number, the faster the elastic strain is damped to the identity, thus releasing elastic potential and associated resistance to deformation. Thus, the Weissenberg number directly controls the amount of the plasticity.

5.4.4 Volume preserving plasticity

The plastic flow in the Oldroyd model will not generally be volume preserving. Since many plastic flows, including those of foams, exhibit this behavior we provide a modification to the standard Oldroyd model that will satisfy this. If we define $\mathbf{b}_{OB}^E$ to obey

$\overset{\triangledown}{\mathbf{b}}{}^E_{OB} = \frac{1}{Wi}(\mathbf{I} - \mathbf{b}^E_{OB})$, then we define a new elastic left Cauchy-Green strain as

$$\mathbf{b}^E \equiv \left(\frac{J}{J_{OB}}\right)^{\frac{2}{3}}\mathbf{b}^E_{OB}, \tag{5.5}$$

where $J = \det(\boldsymbol{F})$ and $J_{OB} = \sqrt{\det(\mathbf{b}^E_{OB})}$. Using this definition, $\det(\mathbf{b}^E) = J^2$ and since by definition $\det(\mathbf{b}^E) = \det(\boldsymbol{F}^E)^2$ and $J = \det(\boldsymbol{F}^E)\det(\boldsymbol{F}^P)$ we see that it must be true that $\det(\boldsymbol{F}^P) = 1$, and thus the plastic flow is volume preserving. In detail,

$$\det(\mathbf{b}^E) = \left(\frac{J}{J_{OB}}\right)^2\det(\mathbf{b}^E_{OB}) = J^2$$

$$\mathbf{b}^E = \boldsymbol{F}^E(\boldsymbol{F}^E)^T \Rightarrow \det(\mathbf{b}^E) = \det(\boldsymbol{F}^E)^2$$

$$J = \det(\boldsymbol{F}) = \det(\boldsymbol{F}^E)\det(\boldsymbol{F}^P) \Rightarrow J^2 = \det(\boldsymbol{F}^E)^2\det(\boldsymbol{F}^P)^2$$

Plugging the first equation into the left side of the third equation and the second equation into the right-hand side of the third equation, we get that $\det(\boldsymbol{F}^P) = 1$, as desired.

5.4.5 Modified plastic flow

This modification to the Oldroyd plasticity obeys

$$\overset{\triangledown}{\mathbf{b}}{}^E = \frac{D}{Dt}\left(\left(\frac{J}{J_{OB}}\right)^{\frac{2}{3}}\right)\mathbf{b}^E_{OB} + \frac{1}{Wi}\left(\frac{J}{J_{OB}}\right)^{\frac{2}{3}}(\mathbf{I} - \mathbf{b}^E_{OB}) \tag{5.6}$$

which has the plastic flow

$$\frac{D}{Dt}[(\mathbf{C}^P)^{-1}] = \frac{D}{Dt}\left(\left(\frac{J}{J_{OB}}\right)^{\frac{2}{3}}\right)(\mathbf{C}^P_{OB})^{-1} +$$
$$\frac{1}{Wi}\left(\frac{J}{J_{OB}}\right)^{\frac{2}{3}}(\mathbf{C}^{-1} - (\mathbf{C}^P_{OB})^{-1}). \tag{5.7}$$

We do not need to solve for $\mathbf{b}^E$ using the definition of its plastic flow. In practice, we solve for the comparatively simple $\mathbf{b}_{OB}^E$ and then obtain the elastic stress as

$$\mathbf{b}^E \leftarrow \left(\frac{J}{J_{OB}}\right)^{\frac{2}{3}} \mathbf{b}_{OB}^E.$$

We only provide this derivation here to show that there is a plastic flow associated with this definition of the elastic strain.

5.4.6 Elasticity

We define constitutive behavior through the compressible Neo-Hookean elastic potential energy density as

$$\psi(\mathbf{b}^E) = \frac{\mu}{2}(\mathrm{tr}(\mathbf{b}^E) - 3) - \mu \ln(J) + \frac{\lambda}{2}(J - 1)^2 \tag{5.8}$$

with associated Cauchy stress

$$\boldsymbol{\sigma}^E = \frac{\mu}{J}(\mathbf{b}^E - \mathbf{I}) + \lambda(J - 1)\mathbf{I}. \tag{5.9}$$

5.5 Material Point Method

We closely follow the algorithm from [Stomakhin et al., 2013]. The only difference is in the discrete Eulerian grid node forces and force derivatives. All steps in the algorithm not related to the update of grid node velocities are the same; we simply change the nature of stress-based forces. In this section, we describe how to modify the potential-based definition of these forces to discretize our new governing equations. We refer the reader to [Stomakhin et al., 2013] for all other steps in the MPM time stepping algorithm.

Using the notation from [Stomakhin et al., 2013], we denote position, velocity and deformation gradient of particle p at time t^n as $\boldsymbol{x}_p^n$, $\boldsymbol{v}_p^n$ and $\boldsymbol{F}_p^n$ respectively. Eulerian grid node locations are denoted as $\boldsymbol{x}_{\mathbf{i}}$ where $\mathbf{i} = (i, j, k)$ is the grid node index. The weights at time t^n are

$$w_{\mathbf{i}p}^n = N_{\mathbf{i}}(\boldsymbol{x}_p^n),$$

where $N_{\mathbf{i}}(\boldsymbol{x})$ is the interpolation function associated with grid node $\mathbf{i}$ and the weight gradients are

$$\nabla w_{\mathbf{i}p}^n = \nabla N_{\mathbf{i}}(\boldsymbol{x}_p^n).$$

As in [Stomakhin et al., 2013], we define the forces on the Eulerian grid nodes as the derivative of an energy with respect to grid node locations. We do not actually move grid nodes, but we consider their movement to define grid node velocities $\boldsymbol{v}_{\mathbf{i}}$ as $\hat{\boldsymbol{x}}_{\mathbf{i}} = \boldsymbol{x}_{\mathbf{i}} + \Delta t \boldsymbol{v}_{\mathbf{i}}$. Using $\hat{\boldsymbol{x}}$ to denote the vector of all grid nodes, we define the potential

$$\Phi(\hat{\boldsymbol{x}}) = \sum_p \left(\Phi^E(\hat{\boldsymbol{x}}) V_p^0 + \Phi^N(\hat{\boldsymbol{x}}) V_p^n \right) \tag{5.10}$$

where $\Phi^E(\hat{\boldsymbol{x}})$ is the elastoplastic component of the potential energy density $\Phi^E(\hat{\boldsymbol{x}}) = \psi(\hat{\mathbf{b}}^E(\hat{\boldsymbol{x}}))$ and $\Phi^N(\hat{\boldsymbol{x}})$ is the Newtonian viscous potential energy density

$$\Phi^N(\hat{\boldsymbol{x}}) = \mu^N \hat{\boldsymbol{\epsilon}}_p(\hat{\boldsymbol{x}}) : \hat{\boldsymbol{\epsilon}}_p(\hat{\boldsymbol{x}}) = \sum_{i,j} \mu^N \hat{\epsilon}_{p_{ij}}(\hat{\boldsymbol{x}}) \hat{\epsilon}_{p_{ij}}(\hat{\boldsymbol{x}}). \tag{5.11}$$

Here

$$\hat{\boldsymbol{\epsilon}}_p(\hat{\boldsymbol{x}}) = \frac{1}{2} \left(\nabla \hat{\boldsymbol{v}}(\hat{\boldsymbol{x}}) + (\nabla \hat{\boldsymbol{v}}(\hat{\boldsymbol{x}}))^T \right)$$

is the strain rate at $\boldsymbol{x}_p^n$ induced by the grid node motion defined by $\hat{\boldsymbol{x}}$ over the time step and

$$\nabla \hat{\boldsymbol{v}}(\hat{\boldsymbol{x}}) = \sum_{\mathbf{i}} \frac{\hat{\boldsymbol{x}}_{\mathbf{i}} - \boldsymbol{x}_{\mathbf{i}}}{\Delta t} (\nabla w_{\mathbf{i}p}^n)^T.$$

As in [Stomakhin et al., 2013], V_p^0 is the volume of the material originally occupied by the particle p. However, for the viscous Newtonian potential, we are approximating an

integral over the time t^n configuration of the material so we have $V_p^n = \det(\boldsymbol{F}_p^n)V_p^0$.

As in [Stomakhin et al., 2013], we store a deformation gradient $\boldsymbol{F}_p^n$ on each particle and update it using

$$\hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}) = (\mathbf{I} + \Delta t \nabla \hat{\boldsymbol{v}}(\hat{\boldsymbol{x}}))\boldsymbol{F}_p^n. \tag{5.12}$$

We use this to define $\hat{J}_p(\hat{\boldsymbol{x}}) = \det(\hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}))$ in the definition of

$$\hat{\mathbf{b}}^E(\hat{\boldsymbol{x}}) = \left(\frac{\hat{J}_p(\hat{\boldsymbol{x}})^2}{\det(\hat{\mathbf{b}}_{OB_p}^E(\hat{\boldsymbol{x}}))} \right)^{\frac{1}{3}} \hat{\mathbf{b}}_{OB_p}^E(\hat{\boldsymbol{x}}). \tag{5.13}$$

Similar to the treatment in Equation 5.12, we store $\mathbf{b}_{OB_p}^{E^n}$ on each particle and discretize the upper convected derivative terms in the evolution equation for $\mathbf{b}_{OB}^E$ to get

$$\begin{aligned}
\hat{\mathbf{b}}_{OB_p}^E(\hat{\boldsymbol{x}}) =& \Delta t \nabla \hat{\boldsymbol{v}}(\hat{\boldsymbol{x}})\mathbf{b}_{OB_p}^{E^n} + \Delta t \mathbf{b}_{OB_p}^{E^n}(\nabla \hat{\boldsymbol{v}}(\hat{\boldsymbol{x}}))^T \\
& + \frac{\Delta t}{Wi}\mathbf{I} + \left(1 - \frac{\Delta t}{Wi}\right)\mathbf{b}_{OB_p}^{E^n}.
\end{aligned} \tag{5.14}$$

The force on the grid nodes is defined as $\mathbf{f}(\hat{\boldsymbol{x}}) = -\frac{\partial \Phi}{\partial \hat{\boldsymbol{x}}}(\hat{\boldsymbol{x}})$ and it is used in the implicit update of grid velocities $\mathbf{v}_{\mathbf{i}}^{n+1}$ exactly as in [Stomakhin et al., 2013]. We work out these derivatives as well as the $\frac{\partial \mathbf{f}}{\partial \hat{\boldsymbol{x}}}(\hat{\boldsymbol{x}})$ in the appendix.

5.5.1 MPM overview

We show how this implementation differs from the one provided in the Material Point Method section. Most importantly, we track the left eleastic Cauchy-Green strain on the particles. Furthermore, our model uses a different potential to account for viscosity effects. Define

$$w_{\mathbf{i}p} := N_{\mathbf{i}}^{\Delta x}(\boldsymbol{x}_p),$$

where $\mathbf{i}$ is a grid index, $\boldsymbol{x}_p$ is particle position, and $N_{\mathbf{i}}^{\Delta x}$ is the interpolating function centered at grid node indexed by $\mathbf{i}$ with spacing Δx, as defined above.

<div align="center">

Particles → Grid → Particles

</div>

- Transfer mass using $m_{\mathbf{i}}^n = \sum_p m_p \mathbf{w}_{\mathbf{i}p}^n$

- Transfer velocity using $m_{\mathbf{i}}^n \boldsymbol{v}_{\mathbf{i}}^n = \sum_p \boldsymbol{v}_p m_p w_{\mathbf{i}p}^n$

<div align="center">

Particles → **Grid** → Particles

</div>

- Compute the MPM approximation to the total potential energy,

$$\Phi(\boldsymbol{x}) = \sum_p (\Psi^E(\boldsymbol{x}) V_p^0 + \Psi(\boldsymbol{x})^N(\boldsymbol{x}) V_p^n)$$

- Forces are given by $\mathbf{f}_{\mathbf{i}}(\boldsymbol{x}) = -\frac{\partial \Phi}{\partial \boldsymbol{x}_i}(\boldsymbol{x})$

- Implicit update is given by $\boldsymbol{v}_{\mathbf{i}}^{n+1} = \boldsymbol{v}_i^n + \Delta t m_{\mathbf{i}}^{-1} \mathbf{f}_{\mathbf{i}}(\boldsymbol{x}_i + \Delta t \boldsymbol{v}_{\mathbf{i}}^{n+1})$

- Perform grid-based collisions

- Solve the system given by the implicit update after grid-based collisions are processed

<div align="center">

Particles → **Grid → Particles**

</div>

- Calculate $\nabla \boldsymbol{v}_p^{n+1} = \sum_{\mathbf{i}} \boldsymbol{v}_{\mathbf{i}}^{n+1} (\nabla w_{\mathbf{i}p}^n)^T$

- Update deformation gradient $\boldsymbol{F}_p^{n+1} = (\mathbf{I} + \Delta t \nabla \boldsymbol{v}_p^{n+1}) \boldsymbol{F}_p^n$

- Update left elastic Cauchy-Green strain

$$\mathbf{b}_{OB_p}^{E^{n+1}} = \Delta t \nabla \boldsymbol{v}_p^{n+1} \mathbf{b}_{OB_p}^{E^n} + \Delta t \mathbf{b}_{OB_p}^{E^n} (\nabla \boldsymbol{v}_p^{n+1})^T + \frac{\Delta t}{Wi} \mathbf{I} + (1 - \frac{dt}{Wi}) \mathbf{b}_{OB_p}^{E^n}$$

- Update particle velocities

$$\boldsymbol{v}_p^{n+1} = (1 - \alpha) \sum_{\mathbf{i}} \boldsymbol{v}_{\mathbf{i}}^{n+1} w_{\mathbf{i}p}^n + \alpha (\boldsymbol{v}_p^n + \sum_{\mathbf{i}} (\boldsymbol{v}_{\mathbf{i}}^{n+1} - \boldsymbol{v}_{\mathbf{i}}^n) w_{\mathbf{i}p}^n)$$

- Perform particle-based body collisions on $\boldsymbol{v}_p^{n+1}$

	Min/Frame	Particle #	Threads	CPU	Δx	Grid Resolution
Twisting sponge	5.3	9.1×10^5	20	3.00GHz	0.0366	245^3
Shooting sponge	2.0	7.2×10^5	16	2.90GHz	0.0402	175^3
Shaving foam	0.93	1.1×10^6	12	3.47GHz	0.0019	257^3
Toothpaste	0.28	2.8×10^5	16	2.90GHz	0.0082	$244 \times 487 \times 244$
Viennetta ice cream	1.11	1.2×10^6	12	2.67GHz	0.0026	$385 \times 96 \times 64$
Pie	23.6	1.3×10^6	12	3.07GHz	0.0024	333^3

Table 5.1: *Simulation performance.*

	ρ	μ	λ	μ^N	Wi
Twisting sponge	2	3.6×10^2	1.4×10^3	0	50
Shooting sponge	1	3.6×10^2	1.4×10^3	0	50
Shaving foam	0.2	5	50	1×10^{-4}	0.5
Toothpaste	1	0.839	8.39	1×10^{-1}	0.4
Viennetta ice cream	1	1	10	5×10^{-5}	0.1
Pie cream	0.2	5	50	1×10^{-7}	1×10^{-4}
Pie crust	0.5	5×10^5	4×10^6	1×10^{-8}	1×10^{30}
Pie crust scored	0.5	5	10	1×10^{-5}	1

Table 5.2: *Material parameters.*

- Update particle positions $x_p^{n+1} = x_p^n + \Delta t v_p^{n+1}$

5.6 Results

In Figure 5.1, a sponge is twisted with top and bottom fixed by Dirichlet boundary conditions. Dynamic fracture and self collision are naturally handled. In Figure 5.2, the top and bottom of a sponge are held in place as we shoot it with a kinematic bullet. The animation is in slow motion to show the detailed material response after the impact. In Figure 5.3, we simulate a stream of shaving foam hitting the ground, and

Method	Elastoplastic	Viscosity	No SVD	Implicit	No Remeshing
[Batty and Bridson, 2008]	✗	✓	✓	✓	✓
[Wojtan et al., 2009]	✓	✓	✗	○[†]	✗
[Batty and Houston, 2011]	✗	✓	✓	✓	○[‡]
[Batty et al., 2012]	✗	✓	✓	✓	✗
[Stomakhin et al., 2013]	✓	✗	✗	✓	✓
[Stomakhin et al., 2014]	✓	✗	✗	✓	✓
[Yue et al., 2015]	✓	✓	✓	✗	✓
Our method	✓	✓	✓	✓	✓

Table 5.3: *Feature comparison with some existing methods. [†]This method is not implicit in elasticity. [‡]This method requires adaptive refinement of a BCC lattice.*
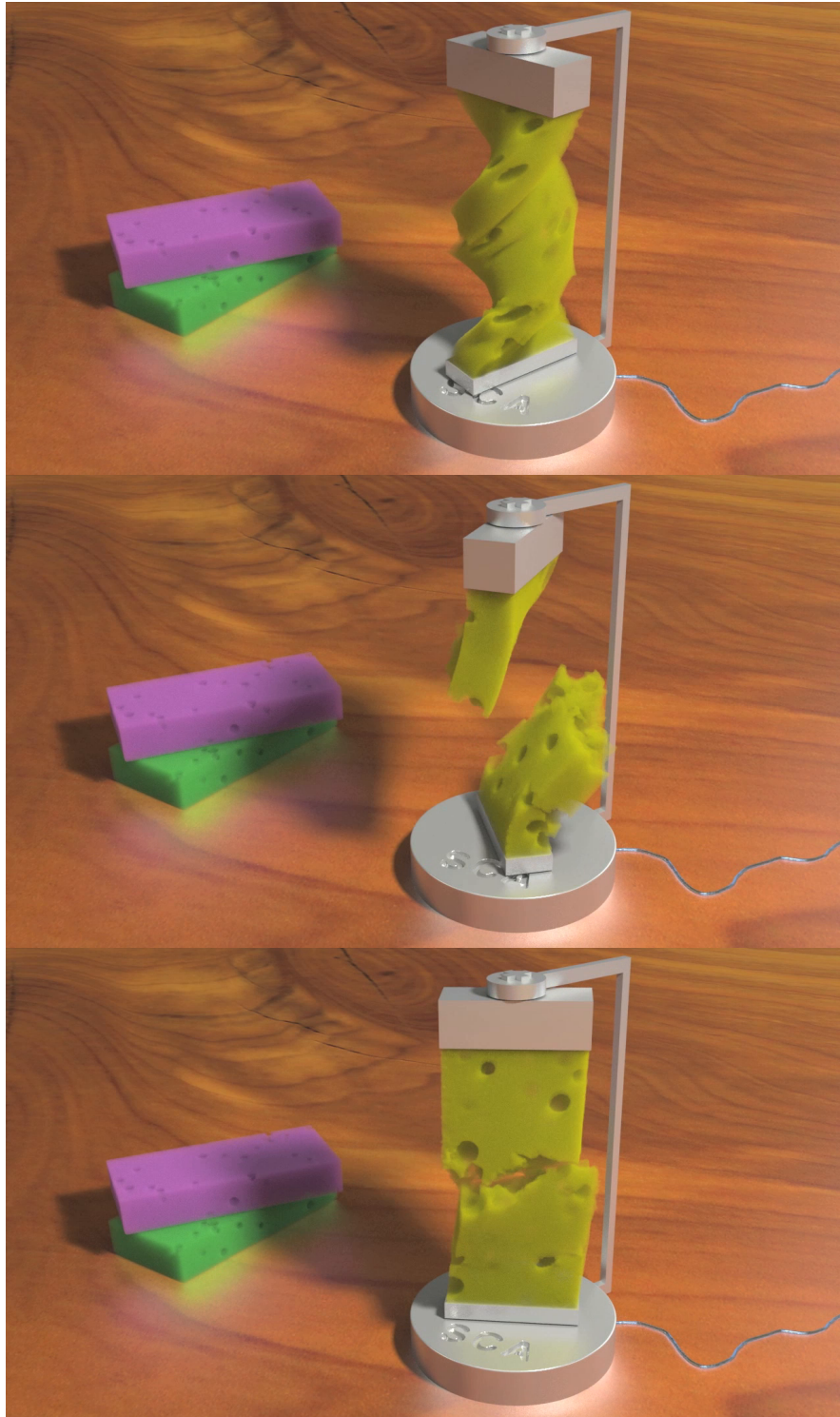
Figure 5.1: *A soft sponge is twisted. It fractures and collides with itself. The failure and contact phenomena are resolved automatically by the MPM approach.*
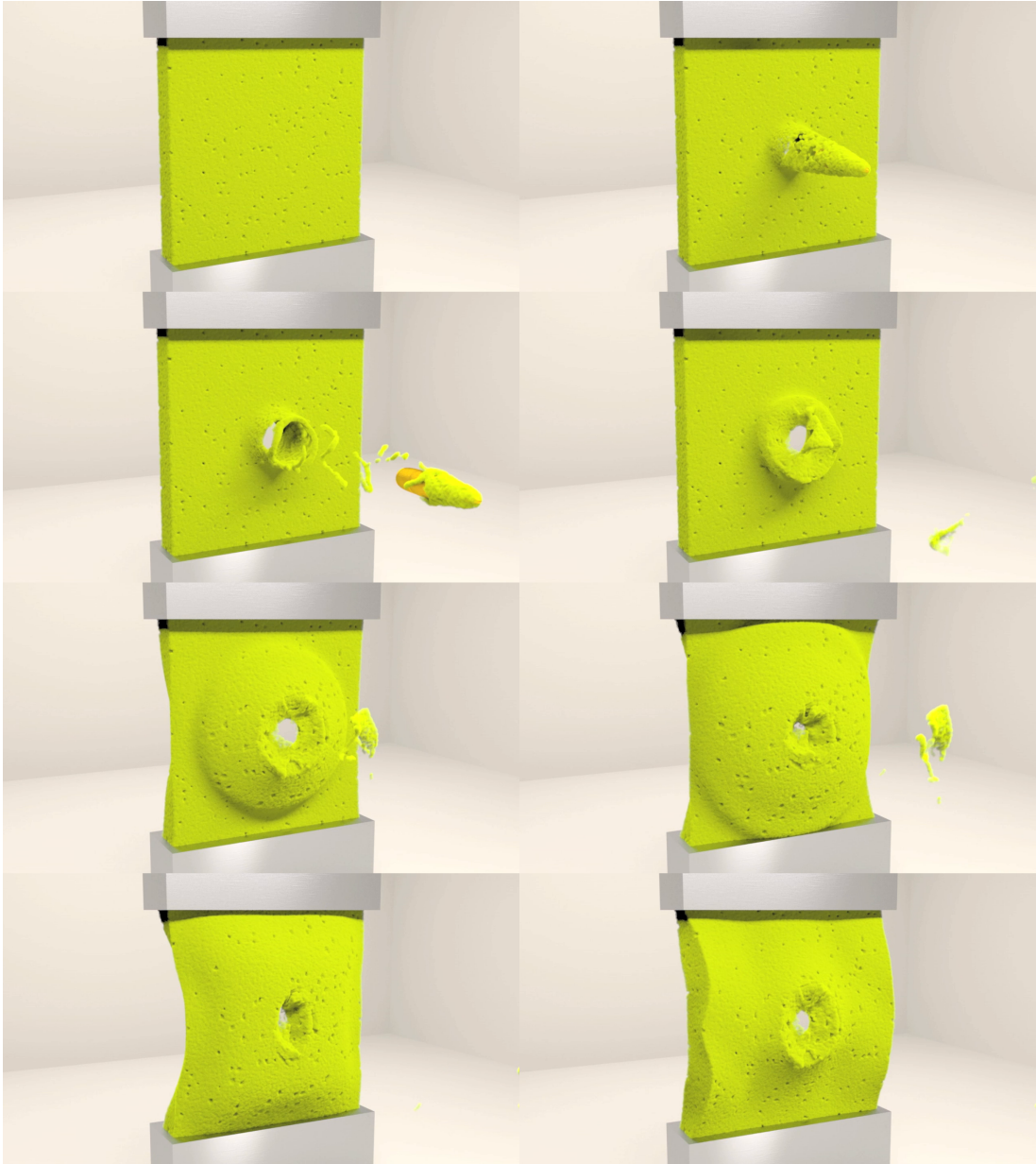
Figure 5.2: *A kinematic bullet is fired at a sponge, resulting in significant deformation and fracture.*
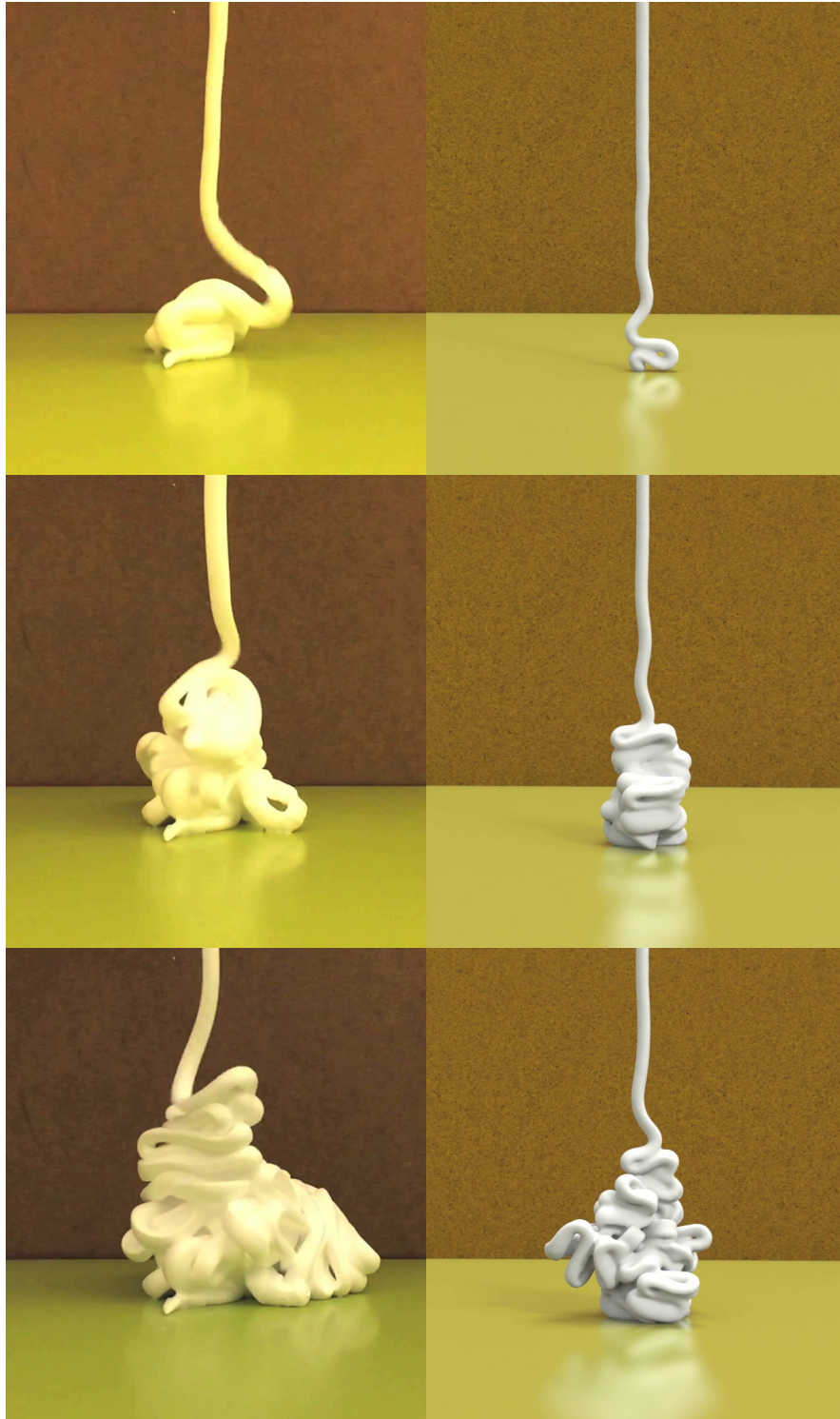
Figure 5.3: *Simulated shaving foam (right) is compared with real world footage (left). The simulation captures the characteristic S-shaped buckling and elastic behavior.*
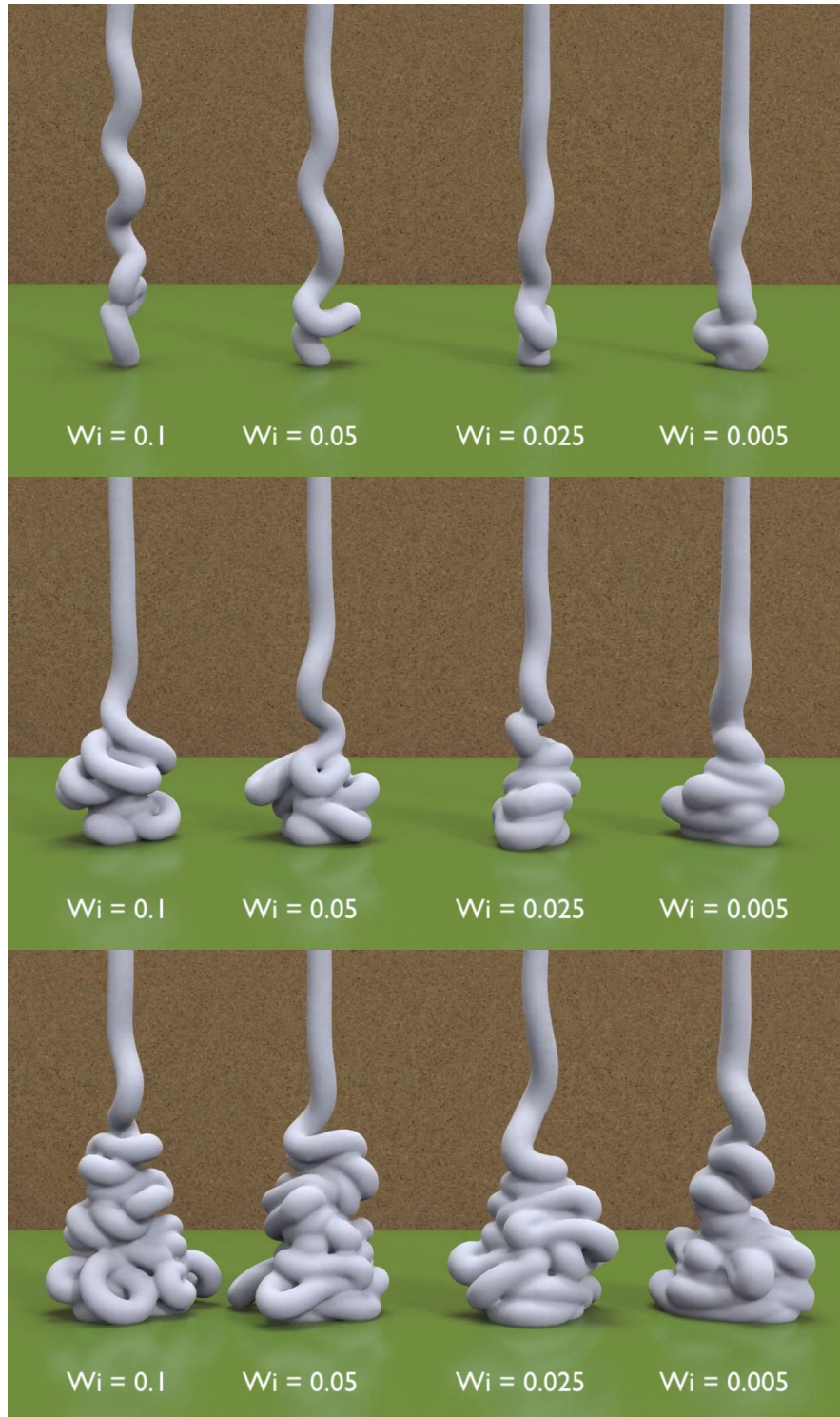
Figure 5.4: *The Weissenberg controls the damping of the left elastic Cauchy-Green strain to the identity. The smaller the Weissenberg, the more plastic the flow.*
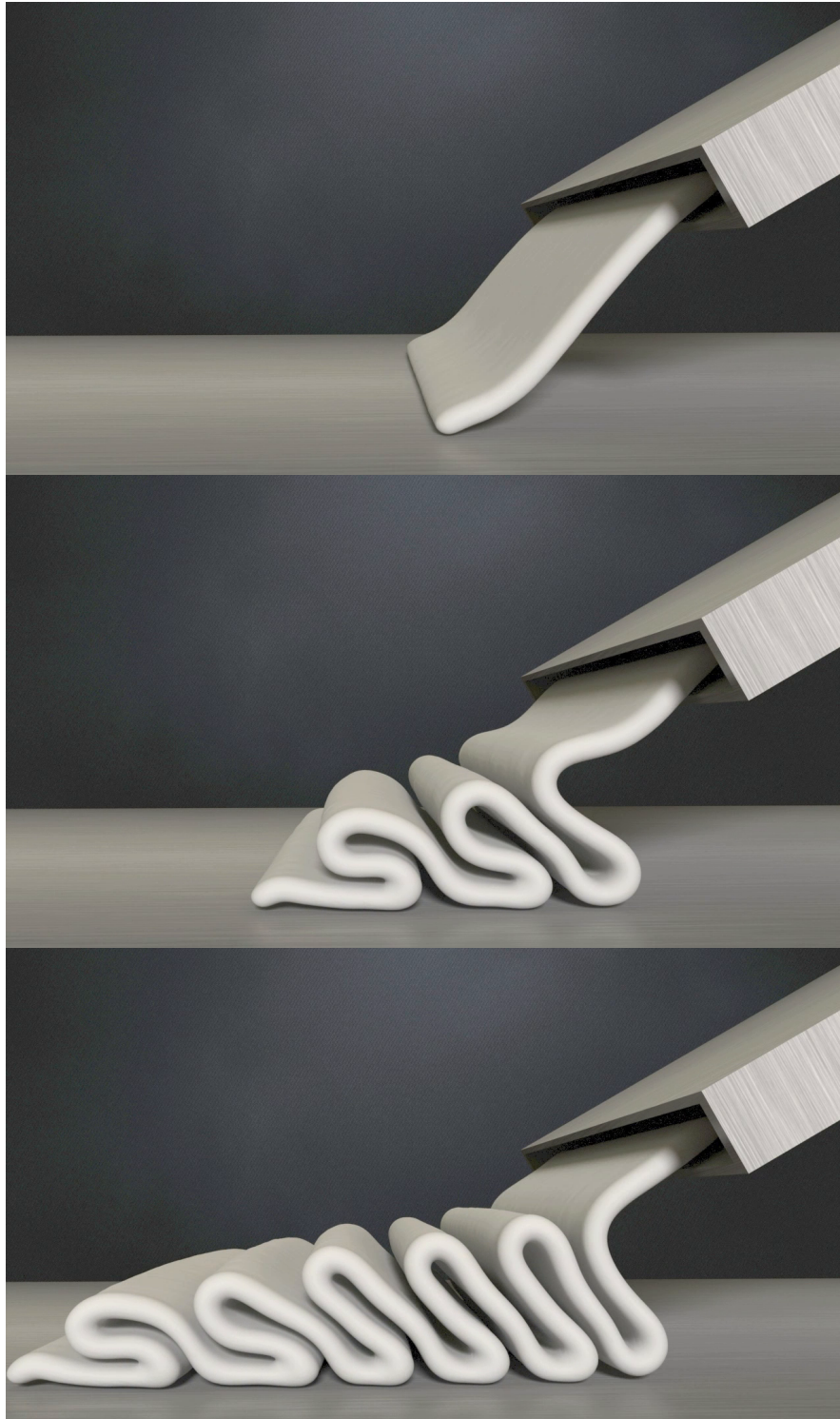
Figure 5.5: *Viennetta ice cream is poured onto a conveyor belt and forms characteristic folds.*

Figure 5.6: *Particle rendering of a pie thrown at a mannequin.*

Figure 5.7: *Particle rendering of Viennetta ice cream. Colored particles on the right correspond to the determinant of the left elastic Cauchy-Green strain on a particle: red particles have determinant greater than 1, blue particles have determinant less than 1.*
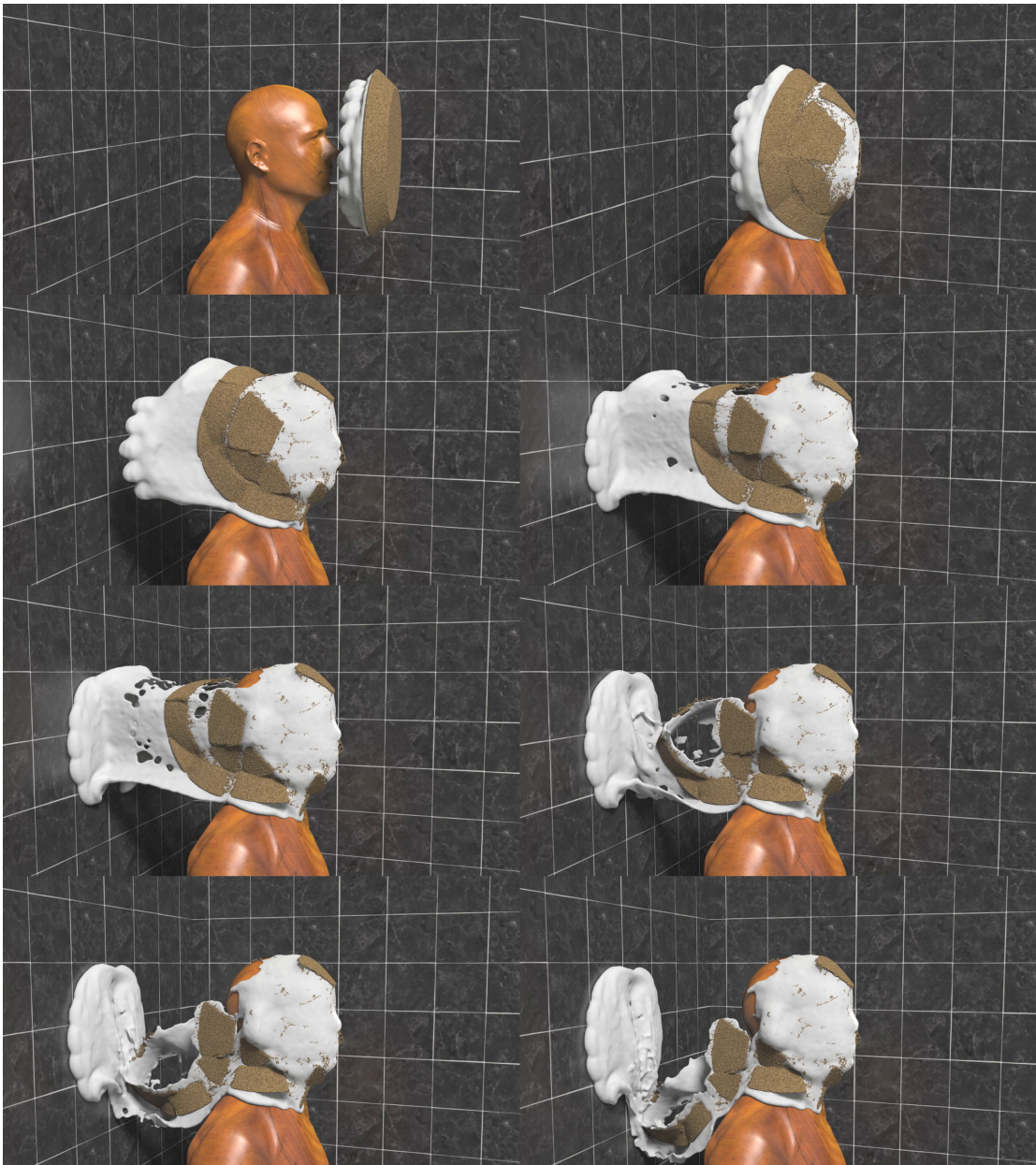
Figure 5.8: *A pie with a stiff crust and soft whipped cream is thrown at a mannequin.*
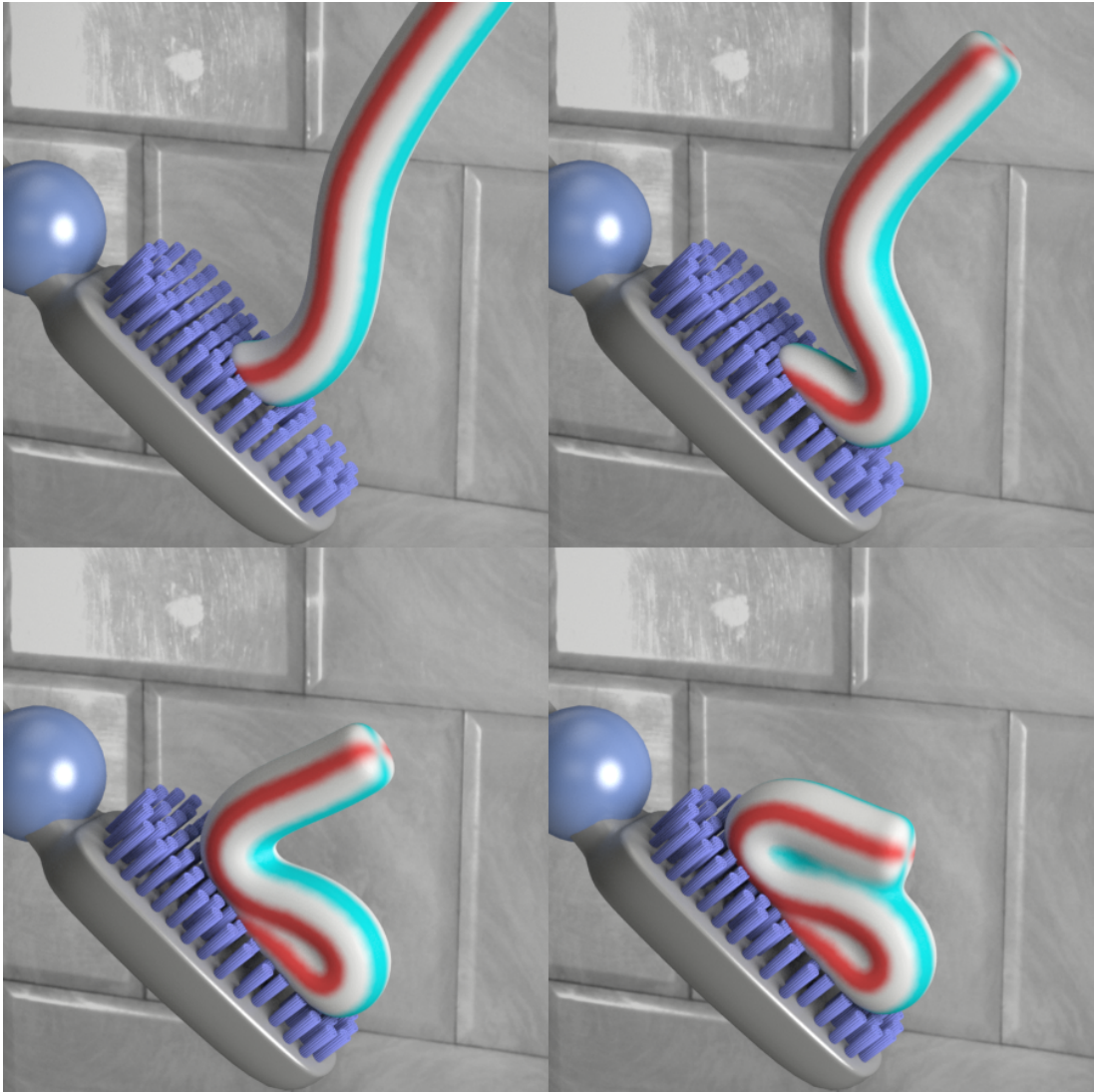
Figure 5.9: *A simulation of toothpaste. Unlike the shaving foam, Newtonian viscosity dominates material behavior.*

compare it with real world footage. Our method captures the S-shaped buckling and merging behaviors. It also exhibits similar elasto-plastic responses. In Figure 5.9, we simulate toothpaste falling onto a toothbrush. Unlike the shaving foam, Newtonian viscosity dominates material behavior. Figure 5.5 shows a simulation of manufacturing Viennetta ice cream. It captures the characteristic folding behavior. In Figure 5.8, we model a pie and throw it at a mannequin. The fracture pattern of the crust is prescored with weak MPM particles. The cream exhibits detailed splitting and merging behavior. For the particle-grid transfers, we used the affine Particle-In-Cell (APIC) method from [Jiang et al., 2015] [Jiang, 2015]. We found that using APIC greatly reduced positional artifacts of the pie particles. We do not perform any explicit particle resampling because self-collision and topology change are naturally handled by MPM.

The material parameters used in our examples are given in Table 5.2. The simulation times are shown in Table 5.1. All simulations were performed on Intel Xeon machines. All renderings were done with Mantra in Houdini. For foam, toothpaste, and Viennetta ice cream, surfaces were reconstructed with OpenVDB [Museth, 2014] and rendered with subsurface scattering. The sponges were rendered as a density field.

5.7 Discussions

We found that using a Jacobi preconditioner greatly reduced simulation run times. For example, in the shooting sponge test (Figure 5.2), the Jacobi preconditioner reduces the number of CG iterations by a factor of 6.

While we have used our method successfully in simulating a variety of materials, it has some limitations. Many of these are related to the Oldroyd-B model. For example, unlike the approach in [Yue et al., 2015], our approach cannot handle shear thickening. Therefore, the model cannot be applied to materials such as oobleck. Our method also does not handle material softening or hardening.

Our update rule of $\mathbf{b}_{OB}^E$ allows for inversion which the constitutive model cannot handle. While $\mathbf{b}_{OB}^E$ should remain positive definite, we have found this to be only partially required. In particular, (5.8) involves the quantity $\text{tr}(\mathbf{b}^E)$, which we must ensure is bounded from below. If $\mathbf{b}^E$ is positive definite, then $\text{tr}(\mathbf{b}^E) > 0$. We also compute $\det(\mathbf{b}_{OB}^E)^{-\frac{1}{3}}$, which is problematic if $\mathbf{b}_{OB}^E$ may become singular. We avoid these problems in practice by taking advantage of the optimization-based integrator from [Gast et al., 2015]. We add a large penalty to our objective when the determinant or trace of $\mathbf{b}_{OB}^E$ becomes infeasible; the line search in our optimizer then discards these configurations. While bounding the trace and determinant does not enforce definiteness in 3D, this strategy worked well in practice. Not enforcing these produces popping artifacts.

5.8 Acknowledgements

CHAPTER 6

Future Work

We discuss potential extensions to the established results. A high-order discretization of the left elastic Cauchy-Green strain is proposed that preserves symmetry and positive-definiteness of the strain. We discuss applying this high-order update to solve the incompressible Oldroyd-B equations.

6.1 High-order discretization of the left elastic Cauchy-Green strain update

6.1.1 Advection

Let G be a Lagrangian function with corresponding Eulerian g (i.e., g is push-forward of G).

Define the downwind quantities

$$g_d^n := g(\boldsymbol{x}^{n+1} - \Delta t \mathbf{u}(\boldsymbol{x}^{n+1} - \frac{1}{2}\Delta t \mathbf{u}(\boldsymbol{x}^{n+1}, t^n), t^{n+\frac{1}{2}}), t^n)$$

$$g_d^{n-1} := g(\boldsymbol{x}^{n+1} - 2\Delta t \mathbf{u}(\boldsymbol{x}^{n+1} - \Delta t \mathbf{u}(\boldsymbol{x}^{n+1}, t^n), t^n), t^{n-1}).$$

Now,

$$\frac{2}{\Delta t}(\boldsymbol{x}^{n+1} - \boldsymbol{x}^{n+\frac{1}{2}}) = \mathbf{u}(\boldsymbol{x}^{n+1}, t^{n+1}) + \mathcal{O}(\Delta t)$$

$$= \mathbf{U}(\phi^{-1}(\boldsymbol{x}^{n+1}, t^{n+1}), t^{n+1}) + \mathcal{O}(\Delta t)$$

$$= \mathbf{U}(\phi^{-1}(\boldsymbol{x}^{n+1}, t^n), t^n) + \mathcal{O}(\Delta t)$$

$$= \mathbf{u}(\boldsymbol{x}^{n+1}, t^n) + \mathcal{O}(\Delta t).$$

Plugging this into the definition of g_d^n, we get

$$g_d^n := g(\boldsymbol{x}^{n+1} - \Delta t\mathbf{u}(\boldsymbol{x}^{n+1} - \frac{1}{2}\Delta t\mathbf{u}(\boldsymbol{x}^{n+1}, t^n), t^{n+\frac{1}{2}}), t^n)$$

$$= g(\boldsymbol{x}^{n+1} - \Delta t\mathbf{u}(\boldsymbol{x}^{n+1} - (\boldsymbol{x}^{n+1} - \boldsymbol{x}^{n+\frac{1}{2}}) + \mathcal{O}(\Delta t^2), t^{n+\frac{1}{2}}), t^n)$$

$$= g(\boldsymbol{x}^{n+1} - \Delta t\mathbf{u}(\boldsymbol{x}^{n+\frac{1}{2}} + \mathcal{O}(\Delta t^2), t^{n+\frac{1}{2}}), t^n)$$

$$= g(\boldsymbol{x}^{n+1} - \Delta t\mathbf{u}(\boldsymbol{x}^{n+\frac{1}{2}}, t^{n+\frac{1}{2}}) + \mathcal{O}(\Delta t^3), t^n).$$

Also,

$$\frac{1}{\Delta t}(\phi(\mathbf{X}, t^{n+1}) - \phi(\mathbf{X}, t^n)) = \mathbf{U}(\mathbf{X}, t^{n+\frac{1}{2}}) + \mathcal{O}(\Delta t^2)$$

$$\frac{1}{\Delta t}(\boldsymbol{x}^{n+1} - \boldsymbol{x}^n) = \mathbf{u}(\boldsymbol{x}^{n+\frac{1}{2}}, t^{n+\frac{1}{2}}) + \mathcal{O}(\Delta t^2)$$

$$\boldsymbol{x}^{n+1} - \Delta t\mathbf{u}(\boldsymbol{x}^{n+\frac{1}{2}}, t^{n+\frac{1}{2}}) = \boldsymbol{x}^n + \mathcal{O}(\Delta t^3).$$

Plugging this in above, we get

$$g_d^n = g(\boldsymbol{x}^n + \mathcal{O}(\Delta t^3), t^n)$$

$$= g(\boldsymbol{x}^n, t^n) + \mathcal{O}(\Delta t^3).$$

Similarly, $g_d^{n-1} = g(\boldsymbol{x}^{n-1}, t^{n-1}) + \mathcal{O}(\Delta t^3)$. To summarize,

$$g_d^n = g(\boldsymbol{x}^n, t^n) + +\mathcal{O}(\Delta t^3)$$

$$g_d^{n-1} = g(\boldsymbol{x}^{n-1}, t^{n-1}) + \mathcal{O}(\Delta t^3).$$

6.1.2 Approximating matrix inverse

Suppose $\mathbf{A} = \mathbf{B} + \mathcal{O}(\Delta t^m)$. This allows us to write $\mathbf{A} = \mathbf{B} - \Delta t^m \mathbf{C}$. Then for small enough Δt,

$$\begin{aligned}
\mathbf{A}^{-1} &= (\mathbf{B} - \Delta t^m \mathbf{C})^{-1} \\
&= \left((\mathbf{I} - \Delta t^m \mathbf{C}\mathbf{B}^{-1})\mathbf{B} \right)^{-1} \\
&= \mathbf{B}^{-1}(\mathbf{I} - \Delta t^m \mathbf{C}\mathbf{B}^{-1})^{-1} \\
&= \mathbf{B}^{-1} \sum_{n=0}^{\infty} \left(\Delta t^m \mathbf{C}\mathbf{B}^{-1} \right)^n \\
&= \mathbf{B}^{-1} + \mathcal{O}(\Delta t^m).
\end{aligned}$$

To summarize,

$$\mathbf{A} = \mathbf{B} + \mathcal{O}(\Delta t^m) \Rightarrow \mathbf{A}^{-1} = \mathbf{B}^{-1} + \mathcal{O}(\Delta t^m),$$

so that (where $\mathbf{f}_e$ is the push-forward of the elastic deformation gradient $\boldsymbol{F}_e$)

$$\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}) = \mathbf{f}_e(\boldsymbol{x}^n, t^n) + \mathcal{O}(\Delta t^3) \Rightarrow$$

$$(\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}))^{-1} = \mathbf{f}_e^{-1}(\boldsymbol{x}^n, t^n) + \mathcal{O}(\Delta t^3)$$

$$\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1}) = \mathbf{f}_e(\boldsymbol{x}^{n-1}, t^{n-1}) + \mathcal{O}(\Delta t^3) \Rightarrow$$

$$(\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1}))^{-1} = \mathbf{f}_e^{-1}(\boldsymbol{x}^{n-1}, t^{n-1}) + \mathcal{O}(\Delta t^3).$$

Finally, using extrapolation, we get

$$\begin{aligned}
\mathbf{f}_e^{-1}(\boldsymbol{x}^{n+1}, t^{n+1}) &= 2\mathbf{f}_e^{-1}(\boldsymbol{x}^n, t^n) - \mathbf{f}_e^{-1}(\boldsymbol{x}^{n-1}, t^{n-1}) + \mathcal{O}(\Delta t^2) \\
&= 2(\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}))^{-1} - (\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1}))^{-1} + \mathcal{O}(\Delta t^2).
\end{aligned}$$

6.1.3 Backward Differentiation Formula

Let G be a Lagrangian function with corresponding Eulerian g (i.e., g is push-forward of G).

Note

$$G(\mathbf{X}, t^n) = G(\mathbf{X}, t^{n+1}) - \Delta t G_t(\mathbf{X}, t^n) + \frac{1}{2}\Delta t^2 G_{tt}(\mathbf{X}, t^n) + \mathcal{O}(\Delta t^3)$$

$$G(\mathbf{X}, t^{n-1}) = G(\mathbf{X}, t^{n+1}) - 2\Delta t G_t(\mathbf{X}, t^n) + 2\Delta t^2 G_{tt}(\mathbf{X}, t^n) + \mathcal{O}(\Delta t^3).$$

This means

$$G(\mathbf{X}, t^{n+1}) = 2G(\mathbf{X}, t^n) - G(\mathbf{X}, t^{n-1}) + \mathcal{O}(\Delta t^2).$$

Using previous identities and the fact that g is the push-forward of G, we get

$$g(\boldsymbol{x}^{n+1}, t^{n+1}) = 2g(\boldsymbol{x}^n, t^n) - g(\boldsymbol{x}^{n-1}, t^{n-1}) + \mathcal{O}(\Delta t^2)$$

$$= 2g_d^n(\boldsymbol{x}^{n+1}) - g_d^{n-1}(\boldsymbol{x}^{n+1}) + \mathcal{O}(\Delta t^2).$$

We also see that

$$\frac{1}{2\Delta t}(3G(\mathbf{X}, t^{n+1}) - 4G(\mathbf{X}, t^n) + G(\mathbf{X}, t^{n-1})) = G_t(\mathbf{X}, t^{n+1}) + \mathcal{O}(\Delta t^2).$$

Using previous identities, the fact that g is the push-forward of G, and that $\frac{Dg}{Dt}$ is the push-forward of G_t, we get

$$\frac{Dg}{Dt}(\boldsymbol{x}^{n+1}, t^{n+1}) = \frac{1}{2\Delta t}\left(3g(\boldsymbol{x}^{n+1}, t^{n+1}) - 4g(\boldsymbol{x}^n, t^n) + g(\boldsymbol{x}^{n-1}, t^{n-1})\right) + \mathcal{O}(\Delta t^2)$$

$$= \frac{1}{2\Delta t}\left(3g(\boldsymbol{x}^{n+1}, t^{n+1}) - 4g_d^n(\boldsymbol{x}^{n+1}) + g_d^{n-1}(\boldsymbol{x}^{n+1})\right) + \mathcal{O}(\Delta t^2).$$

6.1.4 Second order update

Using extrapolation, Taylor series expansions, and downwind approximations, we get

$$\frac{1}{2\Delta t}\left(3\mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1}) - 4\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}) + \mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})\right) =$$

$$\frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\boldsymbol{x}^{n+1},t^{n+1})\mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1})$$

$$+ \frac{1}{2Wi}\left(\mathbf{f}_e^{-T}(\boldsymbol{x}^{n+1},t^{n+1}) - \mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1})\right)$$

$$+ \mathcal{O}(\Delta t^2)$$

$$= \frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\boldsymbol{x}^{n+1},t^{n+1})\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}) - \mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})\right)$$

$$+ \frac{1}{2Wi}\left(\mathbf{f}_e^{-T}(\boldsymbol{x}^{n+1},t^{n+1}) - \mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1})\right)$$

$$+ \mathcal{O}(\Delta t^2)$$

$$= \frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\boldsymbol{x}^{n+1},t^{n+1})\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}) - \mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})\right)$$

$$+ \frac{1}{2Wi}\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})^{-T} - \mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})^{-T}\right.$$

$$\left. - \mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1})\right)$$

$$+ \mathcal{O}(\Delta t^2).$$

Solving for $\mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1})$, we get

$$(3 + \frac{\Delta t}{Wi})\mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1}) = 4\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}) - \mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})$$

$$+ 2\Delta t\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\boldsymbol{x}^{n+1},t^{n+1})\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1}) - \mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})\right)\right)$$

$$+ \frac{\Delta t}{Wi}\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})^{-T} - \mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})^{-T}\right) + \mathcal{O}(\Delta t^3).$$

Recall that if $\mathbf{g} = \mathbf{h} + \mathcal{O}(\Delta t^m)$ then $\mathbf{g}\mathbf{g}^T = \mathbf{h}\mathbf{h}^T + \mathcal{O}(\Delta t^m)$, and that $\mathbf{b}_e = \mathbf{f}_e\mathbf{f}_e^T$. So for updating $\mathbf{b}_e$, we use $\mathbf{b}_e^{n+1} = \mathbf{f}_e^{n+1}(\mathbf{f}_e^{n+1})^T$, which will guarantee that $\mathbf{b}_e$ is symmetric positive-definite and preserve second order accuracy. As the update is second order accurate, it is a candidate for use in a high-order method for solving the incompressible

Oldroyd-B equations.

Note that the update can also be used as an alternative to the $\mathbf{b}_e$ update in the Material Point Method by replacing any downwind quantities at time n with the Lagrangian particle quantity at time n.

6.2   Incompressible Oldroyd-B

A second order virtual node method for elliptic problems is discussed in [Bedrossian et al., 2010] with an extension to three dimensions in [Hellrung et al., 2012]. An extension of these methods to incompressible linear elasticity is developed in [Zhu et al., 2012], and Stokes flow in two-dimensions is treated in [Assêncio and Teran, 2013]. [Schroeder et al., 2014] extends these methods to a high order numerical method for solving multiphase Navier-Stokes flow in an irregular domain. It solves Navier-Stokes flow in three dimensions and handles interfacial discontinuities using a cut-cell finite-element method implementation.

The method is second order accurate for velocities in L^∞, first order accurate for pressures in L^∞, and second order in time. It is a virtual node method that duplicates grid cells to aid in discretizing jump conditions across an interface. The computational domain is embedded in a Cartesian Marker-and-Cell (MAC) grid that allows for efficient computations and prevents costly remeshing of the domain. The interface and boundaries of the domain are represented by levelsets, which are updated in time using Runge-Kutta and updated in space using a weighted essentially non-oscillatory (WENO) scheme. Inertial terms are discretized using an efficient variant of semi-Lagrangian and the backward differentiation formula and fluid variables are discretized using a variational formulation.

We propose using the update outlined in the previous section for the incompressible

Oldroyd-B equations,

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right)=\nabla\cdot(-p\mathbf{I}+\mu\epsilon+\mathbf{b}_e)$$

$$\nabla\cdot\mathbf{u}=0$$

$$\frac{D\mathbf{b}_e}{Dt}=(\nabla\mathbf{u})\mathbf{b}_e+\mathbf{b}_e(\nabla\mathbf{u})^T+\frac{1}{Wi}(\mathbf{I}-\mathbf{b}_e),$$

where ϵ is the rate of strain tensor.

As [Schroeder et al., 2014] solves the Navier-Stokes equations and incompressible Oldroyd-B adds an additional stress tensor to the force balance equation, we would use the framework in [Schroeder et al., 2014] for the terms not involving $\mathbf{b}_e$. In the update proposed above, we have

$$(3+\frac{\Delta t}{Wi})\mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1})=4\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})-\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})$$
$$+2\Delta t\left(\frac{\partial\mathbf{v}}{\partial\mathbf{x}}(\boldsymbol{x}^{n+1},t^{n+1})\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})-\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})\right)\right)$$
$$+\frac{\Delta t}{Wi}\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})^{-T}-\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})^{-T}\right),$$

which will affect the system present in [Schroeder et al., 2014], as velocities are at time $n+1$.

Alternatively, one could use extrapolated quantities for the velocities, so that these terms end up on the right-hand side of the linear solve in [Schroeder et al., 2014], which would not affect the resulting system. This amounts to changing the update to

$$(3+\frac{\Delta t}{Wi})\mathbf{f}_e(\boldsymbol{x}^{n+1},t^{n+1})=$$
$$4\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})-\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})$$
$$+2\Delta t\left(\left(2\left(\frac{\partial\mathbf{v}}{\partial\mathbf{x}}\right)_d^n-\left(\frac{\partial\mathbf{v}}{\partial\mathbf{x}}\right)_d^{n-1}\right)\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})-\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})\right)\right)$$
$$+\frac{\Delta t}{Wi}\left(2\mathbf{f}_{ed}^n(\boldsymbol{x}^{n+1})^{-T}-\mathbf{f}_{ed}^{n-1}(\boldsymbol{x}^{n+1})^{-T}\right).$$

Preliminary results suggest that the updates above are unstable. Further work is required to ensure stability, maintain second order accuracy, and preserve symmetry and positive-definiteness of the strain.

CHAPTER 7

Conclusion

We have provided a Material Point Method that presents a new volume-preserving (isochoric) Oldroyd-B rate-based description of plasticity. The method successfully models complex materials. Complex phenomena such as elastic deformation, large topology change, and plastic effects are handled by the method. The method also handles self-collision and fracture. The discretization is a straightforward extension of an existing Material Point Method and does not require the use of the singular value decomposition, which simplifies implementation. Furthermore, the discretization of viscoelasticity and viscoplasticity is semi-implicit which allows for high-resolution simulations. The method convincingly models materials such as foams, sponges, and toothpaste.

We have also discussed a modification to an existing constitutive model for modeling material weakening under plastic compression. This modification successfully models bone, and can be used in conjunction with a model for materials that weaken under extension, such as tissue and muscle, to model extreme deformation in human limbs.

Discretizations of the left elastic Cauchy-Green strain that preserve symmetry and positive-definiteness of the strain have also been provided. These intuitive discretizations preserve the accuracy of the updates and follow from considering a fully elastic update of the Cauchy-Green strain.

APPENDIX A

Material Point Method for Viscoelastic Fluids, Foams and Sponges

We provide the energy derivatives for the density presented in Chapter 5. Pseudocode for implementation is also detailed.

A.1   Energy Derivatives

To simplify expressions, we use $\mathbf{S} \coloneqq \mathbf{b}_{OB}^{E}$.

$$\Psi(\boldsymbol{F}, \varepsilon, \mathbf{S}) = \frac{\mu}{2}\det(\boldsymbol{F})^{\frac{2}{d}}\det(\mathbf{S})^{-\frac{1}{d}}\operatorname{tr}(\mathbf{S}) - \mu\log(\det(\boldsymbol{F})) + \frac{\lambda}{2}(\det(\boldsymbol{F}) - 1)^2$$

$$+ \mu^{N}\varepsilon : \varepsilon\det(\boldsymbol{F}^{n})$$

$$\frac{\partial\Psi}{\partial\boldsymbol{F}} = \left(\frac{\mu}{d}\det(\boldsymbol{F})^{\frac{2}{d}}\det(\mathbf{S})^{-\frac{1}{d}}\operatorname{tr}(\mathbf{S}) - \mu + \lambda(\det(\boldsymbol{F}) - 1)\det(\boldsymbol{F})\right)\boldsymbol{F}^{-T}$$

$$\frac{\partial\Psi}{\partial\mathbf{S}} = \frac{\mu}{2}\det(\boldsymbol{F})^{\frac{2}{d}}\det(\mathbf{S})^{-\frac{1}{d}}\left(-\frac{1}{d}\mathbf{S}^{-1}\operatorname{tr}(\mathbf{S}) + \mathbf{I}\right)$$

$$\frac{\partial\Psi}{\partial\varepsilon} = 2\mu^{N}\det(\boldsymbol{F}_{n})\varepsilon$$

$$\frac{\partial^2\Psi}{\partial\boldsymbol{F}_{mn}\partial\boldsymbol{F}_{ij}} = \frac{\mu}{d}\det(\boldsymbol{F})^{\frac{2}{d}}\det(\mathbf{S})^{-\frac{1}{d}}\operatorname{tr}(\mathbf{S})\left(\frac{2}{d}\boldsymbol{F}_{mn}^{-T}\boldsymbol{F}_{ij}^{-T} - \boldsymbol{F}_{jm}^{-1}\boldsymbol{F}_{ni}^{-1}\right)$$

$$+ \lambda\det(\boldsymbol{F})\left(\det(\boldsymbol{F})\boldsymbol{F}_{mn}^{-T}\boldsymbol{F}_{ij}^{-T} + (\det(\boldsymbol{F}) - 1)\left(\boldsymbol{F}_{mn}^{-T}\boldsymbol{F}_{ij}^{-T} - \boldsymbol{F}_{jm}^{-1}\boldsymbol{F}_{ni}^{-1}\right)\right)$$

$$- \mu\boldsymbol{F}_{jm}^{-1}\boldsymbol{F}_{ni}^{-1}$$

$$\frac{\partial^2\Psi}{\partial\mathbf{S}_{mn}\partial\mathbf{S}_{ij}} = -\frac{\mu}{2d}\det(\boldsymbol{F})^{\frac{2}{d}}\det(\mathbf{S})^{-\frac{1}{d}}\left(-\mathbf{S}_{im}^{-1}\mathbf{S}_{nj}^{-1}\operatorname{tr}(\mathbf{S}) + \mathbf{S}_{ij}^{-1}\delta_{mn}\right)$$

$$+ \mathbf{S}_{mn}^{-1}\left(-\frac{1}{d}\mathbf{S}_{ij}^{-1}\operatorname{tr}(\mathbf{S}) + \delta_{ij}\right)\Big)$$

$$\frac{\partial^2 \Psi}{\partial \mathbf{S}_{mn}\partial \boldsymbol{F}_{ij}} = \frac{\mu}{d}\det(\boldsymbol{F})^{\frac{2}{d}}\boldsymbol{F}_{ij}^{-T}\det(\mathbf{S})^{-\frac{1}{d}}\left(-\frac{1}{d}\mathbf{S}_{mn}^{-1}\operatorname{tr}(\mathbf{S}) + \delta_{mn}\right)$$

$$\frac{\partial^2 \Psi}{\partial \varepsilon_{mn}\partial \varepsilon_{ij}} = 2\mu^N \det(\boldsymbol{F}^n)\delta_{im}\delta_{jn}$$

$$\hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}) = \Big(\mathbf{I} + \sum_i (\hat{\boldsymbol{x}}_{\mathbf{i}}^n - \boldsymbol{x}_{\mathbf{i}}^n)(\nabla w_{\mathbf{i}}^n)^T\Big)\boldsymbol{F}^n$$

$$\hat{\boldsymbol{F}}_{\alpha\beta} = \boldsymbol{F}_{\alpha\beta}^n + (\hat{x}_{\mathbf{i}\alpha} - x_{\mathbf{i}\alpha}^n)w_{\mathbf{i},\gamma}^n \boldsymbol{F}_{\gamma\beta}^n$$

$$\hat{\boldsymbol{F}}_{\alpha\beta,\mathbf{j}\sigma} = \delta_{\alpha\sigma}w_{\mathbf{j},\gamma}^n \boldsymbol{F}_{\gamma\beta}^n$$

$$\hat{\varepsilon}(\hat{\boldsymbol{x}}) = \frac{1}{2\Delta t}\Big(\sum_i (\hat{\boldsymbol{x}}_{\mathbf{i}}^n - \boldsymbol{x}_{\mathbf{i}}^n)(\nabla w_{\mathbf{i}}^n)^T + \big(\sum_i (\hat{\boldsymbol{x}}_{\mathbf{i}}^n - \boldsymbol{x}_{\mathbf{i}}^n)(\nabla w_{\mathbf{i}}^n)^T\big)^T\Big)$$

$$\hat{\varepsilon}_{\alpha\beta} = \frac{1}{2\Delta t}\Big((\hat{x}_{\mathbf{i}\alpha} - x_{\mathbf{i}\alpha}^n)w_{\mathbf{i},\beta}^n + (\hat{x}_{\mathbf{i}\beta} - x_{\mathbf{i}\beta}^n)w_{\mathbf{i},\alpha}^n\Big)$$

$$\hat{\varepsilon}_{\alpha\beta,\mathbf{j}\sigma} = \frac{1}{2\Delta t}(\delta_{\alpha\sigma}w_{\mathbf{j},\gamma}^n\delta_{\gamma\beta} + \delta_{\beta\sigma}w_{\mathbf{j},\gamma}^n\delta_{\gamma\alpha})$$

$$\hat{\mathbf{S}}(\hat{\boldsymbol{x}}) = \mathbf{S}^n + \sum_i (\hat{\boldsymbol{x}}_{\mathbf{i}}^n - \boldsymbol{x}_{\mathbf{i}}^n)(\nabla w_{\mathbf{i}}^n)^T\mathbf{S}^n + \Big(\sum_i (\hat{\boldsymbol{x}}_{\mathbf{i}}^n - \boldsymbol{x}_{\mathbf{i}}^n)(\nabla w_{\mathbf{i}}^n)^T\mathbf{S}^n\Big)^T$$
$$+ \frac{\Delta t}{W}(\mathbf{I} - \mathbf{S}^n)$$

$$\hat{\mathbf{S}}_{\alpha\beta} = \mathbf{S}_{\alpha\beta}^n + (\hat{x}_{\mathbf{i}\alpha} - x_{\mathbf{i}\alpha}^n)w_{\mathbf{i},\gamma}^n\mathbf{S}_{\gamma\beta}^n + (\hat{x}_{\mathbf{i}\beta} - x_{\mathbf{i}\beta}^n)w_{\mathbf{i},\gamma}^n\mathbf{S}_{\gamma\alpha}^n + \frac{\Delta t}{W}(\delta_{\alpha\beta} - \mathbf{S}_{\alpha\beta}^n)$$

$$\hat{\mathbf{S}}_{\alpha\beta,\mathbf{j}\sigma} = \delta_{\alpha\sigma}w_{\mathbf{j},\gamma}^n\mathbf{S}_{\gamma\beta}^n + \delta_{\beta\sigma}w_{\mathbf{j},\gamma}^n\mathbf{S}_{\gamma\alpha}^n$$

$$\Phi = V_p\Psi_p$$

$$\Phi_{,\mathbf{j}\sigma} = \sum_p V_p^0\Big(\Psi_{p,\sigma\beta}^{\partial\boldsymbol{F}}w_{\mathbf{j}p,\gamma}^n\boldsymbol{F}_{p\gamma\beta}^n + \frac{1}{\Delta t}\Psi_{p,\sigma\beta}^{\partial\varepsilon}w_{\mathbf{j}p,\beta}^n + 2\Psi_{p,\sigma\beta}^{\partial\mathbf{S}}\mathbf{S}_{p\beta\gamma}^n w_{\mathbf{j}p,\gamma}^n\Big)$$

$$\Phi_{,(\mathbf{j}\sigma)(\boldsymbol{k}\tau)} = \sum_p V_p^0 w_{\mathbf{j}p,\gamma}^n\boldsymbol{F}_{p\gamma\beta}^n\Big(\Psi_{p,(\sigma\beta)(\tau\kappa)}^{\partial\boldsymbol{F}\partial\boldsymbol{F}}w_{\boldsymbol{k}p,\omega}^n\boldsymbol{F}_{p\omega\kappa}^n + (\Psi_{p,(\sigma\beta)(\tau\kappa)}^{\partial\mathbf{S}\partial\boldsymbol{F}}w_{\boldsymbol{k}p,\omega}^n\mathbf{S}_{p\omega\kappa}^n$$
$$+ \Psi_{p,(\sigma\beta)(\eta\tau)}^{\partial\mathbf{S}\partial\boldsymbol{F}}w_{\boldsymbol{k}p,\omega}^n\mathbf{S}_{p\omega\eta}^n)\Big)$$
$$+ \sum_p \frac{1}{2\Delta t^2}V_p^0 w_{\mathbf{j}p,\beta}^n\big(\Psi_{p,(\sigma\beta)(\tau\kappa)}^{\partial\varepsilon\partial\varepsilon}w_{\boldsymbol{k}p,\kappa}^n + \Psi_{p,(\sigma\beta)(\eta\tau)}^{\partial\varepsilon\partial\varepsilon}w_{\boldsymbol{k}p,\eta}^n\big) +$$
$$+ \sum_p 2V_p^0 w_{\mathbf{j}p,\gamma}^n\mathbf{S}_{p\gamma\beta}^n\big(\Psi_{p,(\sigma\beta)(\tau\kappa)}^{\partial\boldsymbol{F}\partial\mathbf{S}}w_{\boldsymbol{k}p,\omega}^n\boldsymbol{F}_{p\omega\kappa}^n + (\Psi_{p,(\sigma\beta)(\tau\kappa)}^{\partial\mathbf{S}\partial\mathbf{S}}w_{\boldsymbol{k}p,\omega}^n\mathbf{S}_{p\omega\kappa}^n$$

$$+ \Psi^{\partial\mathbf{S}\partial\mathbf{S}}_{p,(\sigma\beta)(\eta\tau)} w^n_{\boldsymbol{k}p,\omega} \mathbf{S}^n_{p\omega\eta}\big)\Big)$$

A.2 Computing Energy Derivatives

Breaking the potential energy into small pieces makes the implementation straightforward to implement and debug. We present pseudocode that may be used to compute the potential energy $\Phi = \sum_p \Phi_p$ along with its derivatives, $\frac{\partial\Phi}{\partial\boldsymbol{x_i}} = \sum_p \Phi_{p,\mathbf{i}}$ and $\frac{\partial^2\Phi}{\partial\boldsymbol{x_i}\partial\boldsymbol{x_j}} = \sum_p \Phi_{p,\mathbf{ij}}$. The following computational steps may be used to compute the potential energy contribution of a particle Φ_p. Note that all of the quantities computed below, except for the final result Φ_p, are intermediate quantities used to break the computation into many parts. Most of them have no particular physical significance, and most have no particular relationship to similarly named quantities elsewhere in this manuscript. The bold capitalized quantities are matrices, and the rest are scalars.

$$\mathbf{A}_p \leftarrow \sum_{\mathbf{i}} (\hat{\boldsymbol{x}}_{\mathbf{i}} - \boldsymbol{x}^n_{\mathbf{i}})(\nabla w^n_{\mathbf{i}p})^T \qquad\qquad \mathbf{B}_p \leftarrow \mathbf{A}_p \mathbf{b}^{E^n}_{OBp}$$

$$\mathbf{G}_p \leftarrow \mathbf{b}^{E^n}_{OBp} + \frac{\Delta t}{Wi}\big(\mathbf{I} - \mathbf{b}^{E^n}_{OBp}\big) \qquad\qquad \hat{\boldsymbol{F}}_p \leftarrow (\mathbf{I} + \mathbf{A}_p)\boldsymbol{F}^n_p$$

$$\mathbf{S}_p \leftarrow \mathbf{G}_p + \mathbf{B}_p + \mathbf{B}^T_p \qquad\qquad \boldsymbol{H}_p \leftarrow \hat{\boldsymbol{F}}^{-1}_p$$

$$J_p \leftarrow \det(\hat{\boldsymbol{F}}_p) \qquad\qquad a_p \leftarrow \frac{\lambda}{2}(J_p - 1)^2$$

$$q_p \leftarrow \frac{1}{2\Delta t^2}\big(\|\mathbf{A}_p\|^2_F + \mathbf{A}^T_p : \mathbf{A}_p\big) \qquad\qquad b_p \leftarrow \mu \ln(J_p)$$

$$c_p \leftarrow \mu^N_P q_p \det(\boldsymbol{F}^n_p) \qquad\qquad g_p \leftarrow \mathrm{tr}(\mathbf{S}_p)$$

$$\boldsymbol{K}_p \leftarrow \mathbf{S}^{-1}_p \qquad\qquad h_p \leftarrow \det(\mathbf{S}_p)$$

$$k_p \leftarrow h^{-\frac{1}{d}}_p \qquad\qquad m_p \leftarrow J^{\frac{2}{d}}_p$$

$$n_p \leftarrow k_p g_p \qquad\qquad p_p \leftarrow \frac{\mu}{2} m_p n_p$$

$$\Phi_p \leftarrow V_p(p_p - b_p + a_p + c_p)$$

The next set of routines are for the first derivatives of the quantities above, with the final result being the potential energy derivative for a particle, $\Phi_{p,\mathbf{i}}$. Note that these routines use the quantities computed above. Intermediate quantities of the form $c_{p,\mathbf{i}}$ are related to the intermediates above by $c_{p,\mathbf{i}} = \frac{\partial c_p}{\partial \hat{x}_{\mathbf{i}}}$, which allows for incremental testing. All quantities computed below are vectors.

$$\overline{b}_{p\mathbf{i}} \leftarrow \mathbf{b}_{OBp}^{E^n} \nabla w_{\mathbf{i}p}^n \qquad\qquad \overline{f}_{p\mathbf{i}} \leftarrow (\boldsymbol{F}_p^n)^T \nabla w_{\mathbf{i}p}^n$$

$$\overline{h}_{p\mathbf{i}} \leftarrow \boldsymbol{H}_p^T \overline{f}_{p\mathbf{i}} \qquad\qquad \overline{k}_{p\mathbf{i}} \leftarrow \boldsymbol{K}_p^T \overline{b}_{p\mathbf{i}}$$

$$J_{p,\mathbf{i}} \leftarrow J_p \overline{h}_{p\mathbf{i}} \qquad\qquad a_{p,\mathbf{i}} \leftarrow \lambda (J_p - 1) J_{p,\mathbf{i}}$$

$$q_{p,\mathbf{i}} \leftarrow \frac{1}{\Delta t^2} (\mathbf{A}_p \nabla w_{\mathbf{i}p}^n + \mathbf{A}_p^T \nabla w_{\mathbf{i}p}^n) \qquad\qquad b_{p,\mathbf{i}} \leftarrow \mu \overline{h}_{p\mathbf{i}}$$

$$c_{p,\mathbf{i}} \leftarrow \mu_P^N \det(\boldsymbol{F}_p^n) q_{p,\mathbf{i}} \qquad\qquad g_{p,\mathbf{i}} \leftarrow 2\overline{b}_{p\mathbf{i}}$$

$$k_{p,\mathbf{i}} \leftarrow -\frac{2k_p}{d} \overline{k}_{p\mathbf{i}} \qquad\qquad m_{p,\mathbf{i}} \leftarrow \frac{2m_p}{d} \overline{h}_{p\mathbf{i}}$$

$$p_{p,\mathbf{i}} \leftarrow \frac{\mu}{2} (m_{p,\mathbf{i}} n_p + m_p n_{p,\mathbf{i}}) \qquad\qquad n_{p,\mathbf{i}} \leftarrow k_{p,\mathbf{i}} g_p + k_p g_{p,\mathbf{i}}$$

$$\Phi_{p,\mathbf{i}} \leftarrow V_p (p_{p,\mathbf{i}} - b_{p,\mathbf{i}} + a_{p,\mathbf{i}} + c_{p,\mathbf{i}})$$

The final set of routines are for second derivatives, with the final result being the potential energy Hessian for a particle, $\Phi_{p,\mathbf{ij}}$. Intermediate quantities of the form $c_{p,\mathbf{ij}}$ are related to the intermediates above by $c_{p,\mathbf{ij}} = \frac{\partial c_{p,\mathbf{i}}}{\partial \hat{x}_{\mathbf{j}}}$. All quantities computed below are matrices.

$$J_{p,\mathbf{ij}} \leftarrow J_p \overline{h}_{p,\mathbf{i}} \overline{h}_{p,\mathbf{j}}^T - J_p \overline{h}_{p,\mathbf{j}} \overline{h}_{p,\mathbf{i}}^T$$

$$a_{p,\mathbf{ij}} \leftarrow \lambda J_{p,\mathbf{i}} J_{p,\mathbf{j}}^T + \lambda (J_p - 1) J_{p,\mathbf{ij}}$$

$$b_{p,\mathbf{ij}} \leftarrow -\mu \overline{h}_{p,\mathbf{j}} \overline{h}_{p,\mathbf{i}}^T$$

$$q_{p,\mathbf{ij}} \leftarrow \frac{1}{\Delta t^2} ((\nabla w_{\mathbf{i}p}^n)^T \nabla w_{\mathbf{j}} \mathbf{I} + \nabla w_{\mathbf{j}} (\nabla w_{\mathbf{i}p}^n)^T)$$

$$c_{p,\mathbf{ij}} \leftarrow \mu_P^N \det(\boldsymbol{F}_p^n) q_{p,\mathbf{ij}}$$

$$k_{p,\mathbf{ij}} \leftarrow \frac{4k_p}{d^2} \overline{k}_{p,\mathbf{i}} \overline{k}_{p,\mathbf{j}}^T + \frac{2k_p}{d} \overline{k}_{p,\mathbf{j}} \overline{k}_{p,\mathbf{i}}^T + \frac{2k_p}{d} \overline{b}_{p,\mathbf{i}}^T \overline{k}_{p,\mathbf{j}} \boldsymbol{K}_p$$

$$m_{p,\mathbf{ij}} \leftarrow \frac{4m_p}{d^2}\overline{h}_{p,\mathbf{i}}\overline{h}_{p,\mathbf{j}}^T - \frac{2m_p}{d}\overline{h}_{p,\mathbf{j}}\overline{h}_{p,\mathbf{i}}^T$$

$$n_{p,\mathbf{ij}} \leftarrow k_{p,\mathbf{ij}}g_p + k_{p,\mathbf{i}}g_{\mathbf{j}}^T + g_{p,\mathbf{i}}k_{\mathbf{j}}^T$$

$$p_{p,\mathbf{ij}} \leftarrow \frac{\mu}{2}\left(m_{p,\mathbf{ij}}n_p + m_{p,\mathbf{i}}n_{\mathbf{j}}^T + n_{p,\mathbf{i}}m_{\mathbf{j}}^T + n_{p,\mathbf{ij}}m_p\right)$$

$$\Phi_{p,\mathbf{ij}} \leftarrow V_p\left(p_{p,\mathbf{ij}} - b_{p,\mathbf{ij}} + a_{p,\mathbf{ij}} + c_{p,\mathbf{ij}}\right)$$

Bibliography

Assêncio, D. C. and Teran, J. M. (2013). A second order virtual node algorithm for stokes flow problems with interfacial forces, discontinuous material properties and irregular domains. *J. Comput. Phys.*, 250(C):77–105. 81

Bargteil, A., Wojtan, C., Hodgins, J., and Turk, G. (2007). A finite element method for animating large viscoplastic flow. *ACM Trans. Graph.*, 26(3). 54

Batty, C. and Bridson, R. (2008). Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Proc 2008 ACM/Eurographics Symp Comp Anim*, pages 219–228. 54, 64

Batty, C. and Houston, B. (2011). A simple finite volume method for adaptive viscous liquids. In *Proc 2011 ACM SIGGRAPH/Eurograp Symp Comp Anim*, pages 111–118. 54, 64

Batty, C., Uribe, A., Audoly, B., and Grinspun, E. (2012). Discrete viscous sheets. 31(4):113:1–113:7. 54, 64

Becker, M., Ihmsen, M., and Teschner, M. (2009). Corotated sph for deformable solids. In *Eurographics Conf. Nat. Phen.*, pages 27–34. 54

Bedrossian, J., von Brecht, J. H., Zhu, S., Sifakis, E., and Teran, J. M. (2010). A second order virtual node method for elliptic problems with interfaces and irregular domains. *J. Comput. Phys.*, 229(18):6405–6426. 81

Bonet, J. and Wood, R. (2008). *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press. 14, 31, 33, 34, 35, 39, 40, 56, 57

Carlson, M., Mucha, P., Horn, R. V., and Turk, G. (2002). Melting and flowing. In *ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, pages 167–174. 54

Chang, Y., Bao, K., Liu, Y., Zhu, J., and Wu, E. (2009). A particle-based method for viscoelastic fluids animation. In *ACM Symp. Virt. Real. Soft. Tech.*, pages 111–117. 54

Chao, I., Pinkall, U., Sanan, P., and Schröder, P. (2010). A simple geometric model for elastic deformations. In *ACM Transactions on Graphics (TOG)*, volume 29, page 38. ACM. 53

Chong, K., Jiang, C., Ram, D., Santhanam, A., Terzopoulos, D., Benharash, P., Teran, J., and Eldredge, J. Visualization of vascular injuries in extremity trauma. *In preparation*. 43

Desbrun, M. and Gascuel, M. (1996). Smoothed particles: A new paradigm for animating highly deformable bodies. In *Eurographics Workshop Comp. Anim. Sim.*, pages 61–76. 54

Gast, T., Schroeder, C., Stomakhin, A., Jiang, C., and Teran, J. (2015). Optimization integrator for large time steps. *IEEE Trans Vis Comp Graph*, pages 1–1. 75

Gerszewski, D., Bhattacharya, H., and Bargteil, A. (2009). A point-based method for animating elastoplastic solids. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim*, pages 133–138. 54

Goktekin, T., Bargteil, A., and O'Brien, J. (2004). A method for animating viscoelastic fluids. *ACM Trans Graph*, 23(3):463–468. 54

Gonzalez, O. and Stuart, A. (2008). *A First Course in Continuum Mechanics*. Cambridge texts in applied mathematics. Cambridge University Press. 9, 10, 11, 12, 15, 16, 18, 20

Hellrung, Jr., J. L., Wang, L., Sifakis, E., and Teran, J. M. (2012). A second order

virtual node method for elliptic problems with interfaces and irregular domains in three dimensions. *J. Comput. Phys.*, 231(4):2015–2048. 81

Hiemenz, P. and Rajagopalan, R. (1997). *Principles of Colloid and Surface Chemistry*. Marcel Dekker. 52

Jiang, C. (2015). The material point method for the physics-based simulation of solids and fluids. *UCLA Ph.D. Thesis*. 74

Jiang, C., Schroeder, C., Selle, A., Teran, J., and Stomakhin, A. (2015). The affine particle-in-cell method. *ACM Trans Graph*, 34(4). 74

Keiser, R., Adams, B., Gasser, D., Bazzi, P., Dutré, P., and Gross, M. (2005). A unified lagrangian approach to solid-fluid animation. In *Eurographics/IEEE VGTC Conf. Point-Based Graph.*, pages 125–133. 54

Larson, R. G. (1999). *The Structure and Rheology of Complex Fluids*. Oxford University Press: New York. 52, 57, 58

Losasso, F., Irving, G., Guendelman, E., and Fedkiw, R. (2006). Melting and burning solids into liquids and gases. *IEEE Trans. Vis. Comp. Graph.*, 12:343–352. 54

McAdams, A., Zhu, Y., Selle, A., Empey, M., Tamstorf, R., Teran, J., and Sifakis, E. (2011). Efficient elasticity for character skinning with contact and collisions. In *ACM Transactions on Graphics (TOG)*, volume 30, page 37. ACM. 53

Morrison, I. and Ross, S. (2002). *Colloidal Dispersions: Suspensions, Emulsions and Foams*. Wiley Interscience. 52

Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M., and Alexa, M. (2004). Point based animation of elastic, plastic and melting objects. In *ACM SIG-GRAPH/Eurographics Symp. Comp. Anim.*, pages 141–151. 54

Museth, K. (2014). A flexible image processing approach to the surfacing of particle-based fluid animation (invited talk). In *Mathematical Progress in Expressive Image Synthesis I*, volume 4 of *Mathematics for Industry*, pages 81–84. 74

Paiva, A., Petronetto, F., Lewiner, T., and Tavares, G. (2006). Particle-based non-newtonian fluid animation for melting objects. In *Conf. Graph. Patt. Images*, pages 78–85. 54

Paiva, A., Petronetto, F., Lewiner, T., and Tavares, G. (2009). Particle-based viscoplastic fluid/solid simulation. *Comp. Aided Des.*, 41(4):306–314. 55

Prudhomme, R. and Kahn, S. (1996). *Foams: Theory, Measurements, and Applications*. Marcel Dekker. 52

Ram, D., Gast, T., Jiang, C., Schroeder, C., Stomakhin, A., Teran, J., and Kavehpour, P. (2015). A material point method for viscoelastic fluids, foams and sponges. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '15, pages 157–163, New York, NY, USA. ACM. 1

Rasmussen, N., Enright, D., Nguyen, D., Marino, S., Sumner, N., Geiger, W., Hoon, S., and Fedkiw, R. (2004). Directable photorealistic liquids. In *ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, pages 193–202. 54

Schramm, L. (1994). *Foams: Fundamentals and Applications in the Petroleum Industry*. ACS. 52

Schroeder, C., Stomakhin, A., Howes, R., and Teran, J. M. (2014). A second order virtual node algorithm for navier-stokes flow problems with interfacial forces and discontinuous material properties. *J. Comput. Phys.*, 265:221–245. 81, 82

Simo, J. (1988). A framework for finite strain elastoplasticity based on maximum plas-

tic dissipation and the multiplicative decomposition: Part i. continuum formulation. *Comput. Methods Appl. Mech. Eng.*, 66(2):199–219. 39, 40

Solenthaler, B., Schläfli, J., and Pajarola, R. (2007). A unified particle model for fluid-solid interactions. *Comp. Anim. Virt. Worlds*, 18(1):69–82. 54

Stomakhin, A., Howes, R., Schroeder, C., and Teran, J. (2012). Energetically consistent invertible elasticity. In *ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, pages 25–32. 17

Stomakhin, A., Schroeder, C., Chai, L., Teran, J., and Selle, A. (2013). A material point method for snow simulation. *ACM Trans Graph*, 32(4):102:1–102:10. 22, 28, 29, 41, 42, 51, 52, 53, 55, 57, 60, 61, 62, 64

Stomakhin, A., Schroeder, C., Jiang, C., Chai, L., Teran, J., and Selle, A. (2014). Augmented mpm for phase-change and varied materials. *ACM Trans Graph*, 33(4):138:1–138:11. 55, 64

Teran, J., Fauci, L., and Shelley, M. (2008). Peristaltic pumping and irreversibility of a stokesian viscoelastic fluid. *Phys Fl*, 20(7). 58

Terzopoulos, D. and Fleischer, K. (1988a). Deformable models. *Vis Comp*, 4(6):306–331. 53

Terzopoulos, D. and Fleischer, K. (1988b). Modeling inelastic deformation: Viscolelasticity, plasticity, fracture. *SIGGRAPH Comp Graph*, 22(4):269–278. 53

Wojtan, C., Thürey, N., Gross, M., and Turk, G. (2009). Deforming meshes that split and merge. *ACM Trans. Graph.*, 28(3):76:1–76:10. 54, 64

Wojtan, C. and Turk, G. (2008). Fast viscoelastic behavior with thin features. *ACM Trans. Graph.*, 27(3):47:1–47:8. 54

Yue, Y., Smith, B., Batty, C., Zheng, C., and Grinspun, E. (2015). Continuum foam: A material point method for shear-dependent flows. *To appear, ACM Trans Graph.* 36, 37, 38, 55, 57, 64, 74

Zhu, Y., Sifakis, E., Teran, J., and Brandt, A. (2010). An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Trans. Graph.*, 29(2):16:1–16:18. 17

Zhu, Y., Wang, Y., Hellrung, J., Cantarero, A., Sifakis, E., and Teran, J. M. (2012). A second-order virtual node algorithm for nearly incompressible linear elasticity in irregular domains. *J. Comput. Phys.*, 231(21):7092–7117. 81