

Multi-Scale Models to Simulate Interactions between Liquid and Thin Structures

Yun (Raymond) Fei

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2019

©2019
Yun Fei
All rights reserved

ABSTRACT

Multi-Scale Models to Simulate Interactions between Liquid and Thin Structures

Yun (Raymond) Fei

In this dissertation, we introduce a framework for simulating the dynamics between liquid and thin structures, including the effects of buoyancy, drag, capillary cohesion, dripping, and diffusion. After introducing related works, **Part I** begins with a discussion on the interactions between Newtonian fluid and fabrics. In this discussion, we treat both the fluid and the fabrics as continuum media; thus, the physical model is built from *mixture theory*. In **Part II**, we discuss the interactions between Newtonian fluid and hairs. To have more detailed dynamics, we no longer treat the hairs as continuum media. Instead, we treat them as discrete Kirchhoff rods. To deal with the thin layer of liquid that clings to the hairs, we augment each hair strand with a height field representation, through which we introduce a new reduced-dimensional flow model to solve the motion of liquid along the longitudinal direction of each hair. In addition, we develop a faithful model for the hairs' cohesion induced by surface tension, where a penalty force is applied to simulate the collision and cohesion between hairs. To enable the discrete strands interact with continuum-based, shear-dependent liquid, in **Part III**, we develop models that account for the volume change of the liquid as it passes through strands and the momentum exchange between the strands and the liquid. Accordingly, we extend the reduced-dimensional flow model to simulate liquid with elastoviscoplastic behavior.

Furthermore, we adopt a constraint-based model to replace the penalty-force model to handle contact, which enables an accurate simulation of the frictional and adhesive effects between wet strands. We also present a principled method to preserve the total momentum of a strand and its surface flow, as well as an analytic plastic flow approach for Herschel-Bulkley fluid that enables stable semi-implicit integration at larger time steps.

We demonstrate a wide range of effects, including the challenging animation scenarios involving splashing, wringing, and colliding of wet clothes, as well as flipping of hair, animals shaking, spinning roller brushes from car washes being dunked in water, and intricate hair coalescence effects. For complex liquids, we explore a series of challenging scenarios, including strands interacting with oil paint, mud, cream, melted chocolate, and pasta sauce.

Contents

List of Figures	vi
List of Nomenclature	x
Acknowledgements	xv
Chapter 1 Introduction	1
1.1 Overview	1
1.2 Challenges	3
1.3 Necessity of a Multi-Scale Model	5
1.4 Liquid-Fabric Interactions	5
1.5 Water-Strand Interactions	6
1.6 Interactions between Strands and Shear-Dependent Liquids	8
1.7 Contributions to Computer Graphics	9
Chapter 2 Related Work	12
2.1 Cloth and Yarn Simulation	12
2.2 Strand Simulation	13
2.3 Cohesion between Wet Strands	14

2.4	Liquid Simulation	17
2.5	Depth-Averaged Flows	19
2.6	Theory of Mixtures	21
Part I	Multi-Scale Model for Simulating Liquid-Fabric Interactions	29
Chapter 3	Mixture Model for Wet Fabrics	30
3.1	Mixture Theory	31
3.2	Pressure Gradient	36
3.3	Drag between Two Continuous Phases	40
3.4	Dynamic and Quasi-Static Model	43
Chapter 4	Numerical Simulation of Wet Fabrics	47
4.1	Codimensional Quasi-Static Simulation	49
4.2	Grid Simulation	53
Chapter 5	Simulated Results of Wet Fabrics	67
5.1	Didactic Examples	67
5.2	General Examples	72
5.3	Performance Numbers	73
5.4	Surface Reconstruction and Rendering	74
5.5	Fabric Parameters	75
Chapter 6	Limitations of the Wet Fabrics Simulator	77

Part II Multi-Scale Model for Simulating Liquid-Hair Interactions	81
Chapter 7 From Wet Fabrics to Wet Hairs	82
Chapter 8 Wet Hair Physics	84
8.1 Force between Hairs	84
8.2 Liquid Flows over the Hair Surface	93
Chapter 9 Numerical Simulation of Wet Hairs	101
9.1 Hair Simulation	103
9.2 Surface Liquid Simulation on Strands	114
9.3 Bulk Fluid Simulation	117
9.4 Surface Liquid Capture and Dripping	123
Chapter 10 Simulated Results of Wet Hairs	127
10.1 Validation Examples	127
10.2 Large-Scale Examples	129
10.3 Performance Numbers	134
Chapter 11 Future Works on Wet Hairs	136
Part III Multi-Scale Model for Coupling Strands with Shear-Dependent Liquid	140
Chapter 12 From a Newtonian Liquid to a Shear-Dependent Liquid	141

Chapter 13 Physical Models for Coupling Strands with a Shear-Dependent Liquid	144
13.1 Shear-Dependent Liquid	144
13.2 Non-Newtonian Strand Surface Flow	150
13.3 Strand Dynamics	164
13.4 Bulk Liquid in a Mixture	168
13.5 Coupling Forces	173
13.6 Contact Between Wet Strands	185
Chapter 14 Discretization of the Strands and the Shear-Dependent Liquid	196
14.1 Discrete Constrained Dynamics	197
14.2 Algorithm	211
Chapter 15 Simulated Results of the Strands Coupled with Shear-Dependent Liquids	215
15.1 Didactic Examples	215
15.2 Large-Scale Examples	218
15.3 Performance Numbers	220
15.4 Surface Reconstruction	220
15.5 Parameters	222
Chapter 16 Discussions and Limitations	223
16.1 Limitations	224

Chapter 17 Conclusion	227
17.1 Summary of Contributions	228
17.2 Our Recommendation	229
17.3 Perspectives	232
Bibliography	234
Appendix: Gradient and Hessians of the Discrete Curvatures in Discrete Elastic Rods	256
Integrated Curvature Vector	257
Material Vectors	258
Discrete Curvatures	258
Hessian of the Discrete Curvatures	261

List of Figures

Figure 1.1 Simulated results of wet clothes.	6
Figure 1.2 Hair is submerged in water and then rapidly flipped, resulting in wet locks and dripping.	7
Figure 1.3 Simulated results of strands coupled with shear-dependent liquids.	8
Figure 3.1 Fabric as porous material.	32
Figure 3.2 Pore pressure example.	35
Figure 3.3 Comparison with and without the liquid pressure gradient applied to clothes.	36
Figure 3.4 Fiber pack, cloth and yarn orientation.	38
Figure 3.5 Comparison between nonlinear and linear drag models.	41
Figure 4.1 Overview of our numerical method for the mixture of two continuum phases.	47
Figure 4.2 Codimensional objects.	49
Figure 4.3 Liquid capturing.	52
Figure 5.1 Large examples of wet fabrics	68
Figure 5.2 Comparison for different sets of fabric and liquid parameters.	70

Figure 5.3	Volume conservation of the liquid with wet fabrics.	71
Figure 5.4	Performance breakdown for <i>Tighten the Towel</i> .	74
Figure 7.1	Recursive structures in wet hair coalescence.	83
Figure 8.1	Water flowing over and through a sloped mat of fur.	85
Figure 8.2	A liquid bridge.	86
Figure 8.3	Notation for a liquid bridge cross section.	87
Figure 8.4	Two wet hairs held close and then pulled apart.	90
Figure 8.5	Plot of the potential energy of the combined cohesion/repulsion force vs. distance.	91
Figure 8.6	Inertial force.	94
Figure 8.7	A droplet sliding down a single vertical hair as it moves under uniform horizontal velocity.	95
Figure 8.8	A droplet sliding down a single hair, causing it to swing.	96
Figure 8.9	Surface liquid flow and dripping on a single strand.	98
Figure 8.10	A comparison of liquid flow on a coiled hair strand.	99
Figure 9.1	Algorithm overview and data flow for simulating wet hairs.	102
Figure 9.2	An adaptive quadrature to evaluate the cohesion forces.	104
Figure 9.3	Discretized surface flow along hair.	115
Figure 9.4	A hair suspended by its tip moving left to right through liquid.	121
Figure 9.5	Capture of liquid from a flowing stream.	123
Figure 10.1	Wringing out water from drenched hair.	130

Figure 10.2 Wet hairs are whipped onto a wall and then pulled off.	130
Figure 10.3 Volume conservation of water with wet hairs.	131
Figure 10.4 Color-coded visualization of the age of fluid particles.	132
Figure 10.5 Water is poured over four fur mats of the same hair density but of differing hair lengths.	133
Figure 10.6 Comparison between different solvers for the hair dynamics. . . .	134
Figure 10.7 Large examples of wet hair simulation.	135
Figure 13.1 Surface flows of various materials.	150
Figure 13.2 Coordinate system defined along the strand centerline.	152
Figure 13.3 Surface flows under increasing centrifugal force.	154
Figure 13.4 Comparison between different methods to integrate the additional inertia.	167
Figure 13.5 Buoyancy of the strands.	174
Figure 13.6 Comparison between varying and fixed volume fraction.	176
Figure 13.7 Simulated cream with a wide range of bulk moduli.	178
Figure 13.8 Drag coefficient over its different parameters.	182
Figure 13.9 Comparison of different drag coefficients.	183
Figure 13.10 Comparison between constraint- and penalty-based collision. . . .	187
Figure 13.11 Regular and modified second-order Coulomb cone.	188
Figure 13.12 The profile of the contact-cohesion force in the normal direction. .	192
Figure 13.13 Contact between strands with surface flows of different materials.	195
Figure 14.1 Comparing implicit and explicit integration.	200

Figure 14.2 A strand with staggered discrete flow variables. 209

Figure 15.1 *Shaking a Hairball, Splashing the Paint, Chocolate “dog”, and Soba with Oyster Sauce.* 218

Figure 15.2 Timing statistics for the *Chocolate “Dog”*. 220

Figure A1 Discrete elastic rods, adapted from the book by Jawed et al. [117]. 257

List of Nomenclature

α	the angle between the liquid-air interface and the centerline of liquid bridge
\mathbf{c}_p^t	the affine strain rate of the p -th liquid particle (1/s)
$\mathbf{t}(x)$	strand's centerline unit tangent at x
ω	the angular velocity in the degree of freedom for hair twisting (rad/s)
ζ	gaps between strands in contact (cm)
\mathbf{a}	the external acceleration applied (cm/s ²)
\mathbf{A}_τ	surface flow cross-sectional area on strand vertices (cm ²)
\mathbf{e}_i	i -th grid face direction
\mathbf{h}_τ	the height of reduced liquid (cm)
\mathbf{p}	liquid pressure (dyn · cm ²)
\mathbf{q}_f	liquid particle position (cm)
\mathbf{q}_s	strand vertex position (cm)
\mathbf{r}	contact impulse (dyn · s)

\mathbf{u}_f	liquid particle velocity (cm/s)
\mathbf{u}_s	strand vertex velocity (cm/s)
\mathbf{u}_τ	surface flow velocity on strand elements (cm/s)
\mathbf{v}	strand velocity in contact space (cm/s)
\mathbf{x}_i	i -th grid face position (cm)
Δx	the length of a grid cell (cm)
ds	a small piece of strand segment
d	differential operator
η	the liquid's dynamic viscosity or its flow consistency index (cP)
Γ	the curve of a hair strand (cm)
$\hat{\mathbf{x}}_i$	predicted i -th face position (cm)
$\langle \cdot \rangle_j^i$	the operator that converts edge-based quantities into vertex-based quantities
\mathbf{E}	mapping contact space to world space
\mathbf{G}_{cg}	gradient on faces (from centers, cm^{-1})
\mathbf{G}_{cg}^T	(negative) divergence on centers (from faces, cm^{-1})
$\mathbf{G}_{w,cv}$	gradient on vertices (from centers, cm^{-1})
$\mathbf{G}_{w,cv}^T$	(negative) divergence on centers (from vertices, cm^{-1})

\mathbf{H}_f	Jacobian of the liquid's shear force (dyn/cm)
\mathbf{H}_s	Jacobian of the strand force (dyn/cm)
$\mathbf{H}_{s,G}$	the Jacobian of the inter-strand coupling force (dyne/cm)
\mathbf{I}_d	the identity matrix of dimension d
\mathbf{M}_f	mass matrix of the liquid (g)
\mathbf{M}_s	mass matrix of the strands (g)
$\mathbf{M}_{\tau,n}$	mass matrix of flow on vertices (g)
\mathbf{S}_i	a diagonal <i>selection matrix</i> whose j -th term is one if the j -th degree of freedom belongs to the i -th strand and zero otherwise
\mathbf{T}	mapping element tangent space to world space
\mathbf{V}_c	volume of liquid at centers (cm^3)
\mathbf{V}_f	volume of liquid at faces (cm^3)
\mathbf{V}_p	volume of particles (cm^3)
$\mathbf{V}_{s,v}$	volume of solid vertices (cm^3)
\mathbf{W}_{ge}	weight mapping faces to elements
\mathbf{W}_{ge}^T	weight mapping elements to faces
\mathbf{W}_{gv}	weight mapping faces to vertices
\mathbf{W}_{gv}^T	weight mapping vertices to faces

\mathbf{W}_{ve}	weight mapping vertices to elements
\mathbf{W}_{ne}^T	weight mapping elements to vertices
ϕ_f	the volume fraction of fluid
ϕ_s	the volume fraction of strands/clothes
ρ_f	the mass density of liquid (g/cm ³)
ρ_s	the mass density of hair (g/cm ³)
σ	surface tension coefficient of the liquid-air interface (dyne/cm)
θ	equilibrium contact angle
ς_v	twisting torque defined on a solid vertex (dyne/cm)
ϑ	the angle between the hair strand direction and the direction of relative velocity of the hair and the surrounding bulk liquid
A_L	the cross-sectional area of the liquid bridge (cm ²)
E_s	surface energy of liquid bridges (erg)
h	the time step size (s)
l^i	the vertex-based length for hair vertex (cm)
l_A	arc length of the liquid-air boundary (cm)
l_i	the length of edge i (cm)
l_S	arc length of the hair-air boundary (cm)

m_i^t	the mass of liquid for grid face i at time step t (g)
N_c	the number of grid centers
N_e	the number of strand/cloth elements
N_g	the number of grid faces
N_k	the number of contacts
N_p	the number of liquid particles
N_v	the number of strand/cloth vertices
r	the reciprocal of the mean curvature of the solid surface / the radius of hairs / the radius of capillarity (cm)
S_r	the saturation of partially-filled mixture
$u_{f,i}^t$	the velocity of liquid for grid face i direction at time step t (cm/s)
$V_{\tau,i}^t$	the volume of reduced liquid around strand/cloth vertex i at time step t (cm ³)
$V_{f,i}^t$	the volume of bulk liquid for particle i at time step t (cm ³)
$V_{s,i}^n$	the volume of strand/cloth vertex i at time step t (cm ³)
w_{FF}	the factor acts to smoothly disable the cohesion force if the quadrature pair lies beneath the bulk liquid surface
$w_{p,i}^t$	the interpolation weight that transfers the information on particle p to grid face i

Acknowledgements

Throughout my study and research as a Ph.D. student, I have received a great deal of support and assistance.

I would like first to thank my advisors, Dr. Eitan Grinspan and Dr. Changxi Zheng, as well as Dr. Christopher Batty, whose constant presence and support aided me in conducting my research. Your expertise in the methodology of research, the forming of a research topic, and the bar to identify an excellent publication are invaluable.

I also want to acknowledge people who have discussed and helped with my research, including but not limited to, Dr. Gilles Daviet, Dr. Florence Bertails-Descoubes, Dr. Chenfanfu Jiang, Dr. Bo Zhu, Dr. Fang Da, Ryan Goldade, and Peter Yichen Chen. Without you and your valuable discussions and comments, finding the correct path to develop my research would have been unimaginably more difficult.

I would also like to thank people who have helped with my publications or who have contributed or allowed me to use their code in my projects. These individuals include but are not limited to, Dr. Yonghao Yue, Dr. Danny Kaufman, Dr. Xinxin

Zhang, Dr. Gabriel Cirio, Dr. Jean-Marie Aubry, Dr. Miklos Bergou, Dr. Breannan Smith, Mark Leone, Susan Howard, Henrique Maia, Chang Xiao, and Oded Stein.

Furthermore, I would like to thank those who have advised me or helped me during my internships, including but not limited to, Dr. Jovan Popovic, Dr. Ken Museth, and Dr. Theodore Kim.

In addition, I would like to thank the team at the Computing Research Facilities of Columbia University, including Daisy Nyugen, Derrick Lim, Robert Lane, and Jorge Espinoza. Without your help and contribution to the CPU cores and hard disk space, my research would have been impossible to finish. I would also like to thank Cristin Barghiel and other people in SideFX for your continued generous contribution of the Houdini licenses.

Finally, I would like to thank my supportive family and friends, especially my wife Adela. You are always there for me and have accompanied me through the toughest moments during my research.

To family, friends, advisors, reviewers, and all the other people who once helped me.

Chapter 1

Introduction

1.1 Overview

A beach vacation offers many opportunities to discover the unique aspects of the interactions between liquid and thin structures. In the water, submerged hairs and board shorts drag along with the ocean waves and are lifted buoyantly by the surf. When a swimmer surfaces, the water flows along with the hair and gathers into streams that coalesce the strands into bundles. While the swimmer returns to dry land, the water drips distinctively from the hair strands and shorts. Somewhere on the beach, a glass of piña colada is tipped, and the liquid splashes onto the beachwear, diffusing its wetness to dampen a larger area.

Thin structures such as cloth and hair have long been a focus of computer graphics because of their critical role in the appearance of humans and animals. Likewise, the ubiquity of fluid phenomena in the physical world has led to extensive research into the simulation of gases and liquids. Recently, the pursuit of realism has given birth to many algorithms for hairs, clothes, and liquids. Physics theories invented hundreds years ago are rediscovered to model the dynamics of these thin structures.

These theories include the Kirchhoff Rod theory [133] for describing the dynamics of strands, the Kirchhoff-Love plate theory [155] for modeling the motions of clothes, and Cauchy's momentum equation [49] for modeling the movement of general fluids. Correspondingly, researchers propose some discretized models. For example, they develop *Discrete Elastic Rod* (DER) [27, 28, 126] to simulate hair strands, the *Discrete Shell* [92] for simulating clothes, and the *Material Point Method* (MPM) [228] for simulating complex liquids.

Due to their effectiveness, these methods have been extensively adopted in the visual effects (VFX) industry, and have dramatically pumped up the verisimilitude of visual effects in recent films. For example, in the movie *Alita: Battle Angel* (2019)¹, more than 200K individual hairs are simulated with detailed collisions and frictions. In Pixar's animation, *Coco* (2017)², multi-layered shirts and cloaks are simulated along with complex body movements. In Disney's *Moana* (2016)³, artists simulate liquid containing millions of cubic meters with intricated details. All of these fantastic effects attest to the growing computational power and the recent development of novel simulation algorithms.

Nevertheless, in real life, we are not dealing with just a single phenomenon or a single kind of material. Instead, multiple types of materials or phenomena are usually co-present in a scenario, whose dynamics couple together. Methods to capture the

¹Twentieth Century Fox, producers: James Cameron, Robert Rodriguez, and Jon Landau. Visual effects done by Weta Digital.

²Walt Disney Pictures, directors: Lee Unkrich, Adrian Molina.

³Walt Disney Pictures, directors: Ron Clements, John Musker.

array of complex interactions between these ubiquitous phenomena have received less attention in the literature, and for coupling with thin structures such as hairs or clothes, no model compatible with a standard grid-based fluid solver has been proposed.

1.2 Challenges

The interaction between thin structures and liquid is inherently a challenging topic, even for a single strand or a single piece of cloth coupled with liquid, because multiphysics phenomena span multiple spatial scales. When fully submerged in a body of liquid, thin underwater structures collectively behave as a kind of porous medium and introduce a drag effect on the surrounding flow. When removed from the liquid, each inevitably retains a certain amount of liquid on its surface or within. Meanwhile, each strand or cloth also interacts with another, not only through dry collisions but also through surface tension effects of *liquid bridges* that connect structures nearby — the latter denoting the physics that causes them to cohere and coalesce into clumps.

An examination of the consequences of these multi-scale physics is critical to the design of an appropriate model. One tiny length scale — the thickness of thin structures — contrasts with a much larger length scale, such as the length of hairs or the expansion of cloth. The very thin layer of liquid flow immediately surrounding these thin structures contrasts with the bulk volumes of liquid with which these structures may also interact. Finally, the drag forces that affect fully submerged thin structures differ dramatically from the out-of-liquid surface tension forces that

rearrange them significantly.

Standard approaches to fluid simulation make use of volumetric representations that divide up space into a grid [43]. When sufficient computational power is available, practitioners perform fine-scale simulations of the interaction between structures (such as hair) and fluid, often using the methods referred to as embedded or immersed boundary methods [165]; or they apply conceptually analogous solid-fluid coupling schemes to animation problems involving rigid bodies, deformable, or cloth (e.g., [22, 200]). For thin structures, it is necessary to simulate fluid on a grid that is exceedingly fine to capture interactions with a strand having a radius that spans at least several cells.

Consider the scenario where a person with long hairs drills out from the water and flip his/her hairs for example — the scenario is typically $1\text{m} \sim 3\text{m}$ wide or high, while the liquid flowing over a single hair can be as thin as 0.001cm . In this case, a standard grid-based simulator would require a grid with at least 100000^3 cells to cover the entire scene while capturing the detailed motion of liquid flowing on a strand. A usual simulation using a $100^3 \sim 300^3$ grid would typically take several hours or a day on a modern workstation for an animation with several seconds (e.g., [122]). The simulation on a grid with 100000^3 cells, even with an oracle machine that can linearly scale, would take millions of years, which in practice is impossible.

In summary, existing methods cannot capture the multiscale multiphysics of liquid-strand or liquid-fabric interactions with a reasonable computational cost, and prac-

tical high-fidelity simulation of liquid-strand or liquid-fabric interactions remains an open problem.

1.3 Necessity of a Multi-Scale Model

In this thesis, we develop a multi-scale method for strands or fabrics interacting with liquids, whether they are fully submerged or coated with just a thin layer of liquid. The crucial observation to sidestep the computational cost in the standard approach is to view strands/fabric as a porous medium when they are submerged in liquid, and as structures enhanced with height fields when they are exposed in the air. This strategy works because we only care about the macroscale interactions between submerged thin structures and liquid (e.g., drag and pressure), and only care about the longitudinal (for strands) or tangential (for fabrics) motion of liquid on or inside the exposed thin structures.

With this general strategy in mind, we propose a series of new methods to simulate a variety of interaction effects between the liquid and the clothes/hairs.

1.4 Liquid-Fabric Interactions

Our exploration begins in **Part I** from the liquid-fabric interactions (Figure 1.1), where we develop a macroscopic model built on a modern mixture theory [8] (§3): We model fabric as a continuous porous media through which fluid may flow. The model accounts for the material’s anisotropic structure, and the evolution of its saturation,

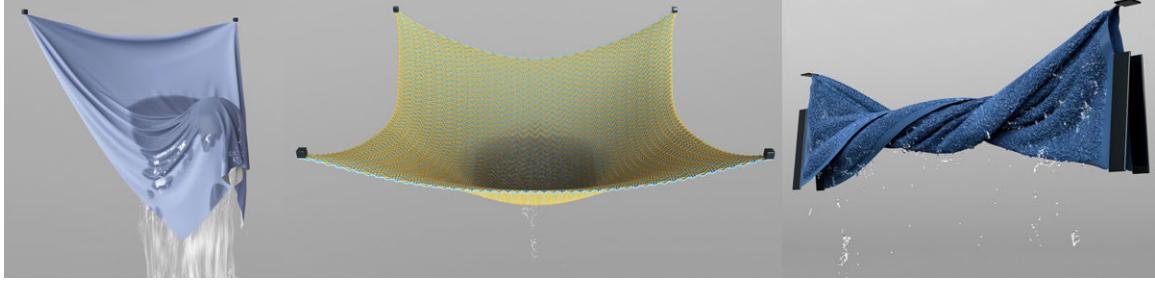


Figure 1.1: **Simulated results of wet clothes.** **Left:** A piece of mesh-based cloth draped over a solid obstacle is splashed with water. **Middle:** Water flows through a piece of yarn-based handwoven fabric. **Right:** After being vigorously wrung out, a thick-textured towel continues to drip.

to capture buoyancy, drag, small-scale capillary (surface tension) effects, and fluid convection. Our numerical treatment (§4) integrates a piecewise linear Lagrangian cloth or rod model [28, 92] with a hybrid Eulerian-Lagrangian (APIC) fluid simulator [43, 122].

We apply this model to several scenarios (§5) involving mesh-based cloth, yarn-based fabric, and thick-textured fabric in contact with water. We also examine the qualitative comparisons against simple real-world experiments, including a test on liquid spreading and suctioning (Figure 3.2).

1.5 Water-Strand Interactions

In **Part II**, we explore the liquid-hair interactions (Figure 1.2). Differing from wet fabrics, wet hairs may have dispersive or collectively-nonlinear motions, which cannot be captured with a continuum model. Hence we solve the hairs' motion individually as discrete rods. Besides, the liquid usually flows unsteadily over the hair surface instead of creeping inside the hair strands [20].

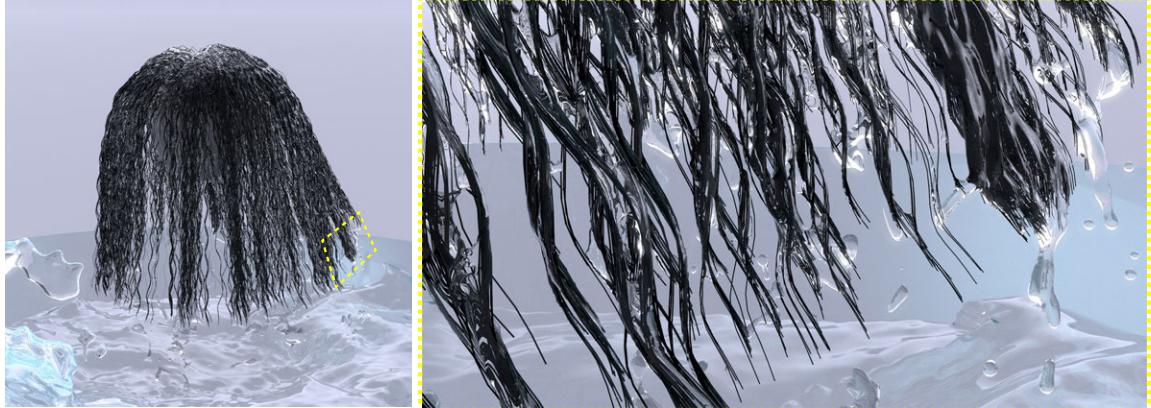


Figure 1.2: **Hair is submerged in water and then rapidly flipped, resulting in wet locks and dripping.**

This observation motivates our addition of a *surface* fluid model to represent and track liquid directly *on the surface of hair* (§8.2). Specifically, we introduce a height-field representation for the smaller volume of liquid along and around each individual hair strand. Given this set of physical representations for the hairs and the liquid, we can further derive a penalty-based cohesion force (§8.1) that is required to faithfully reproduce wet hairs’ coalescence effect.

We evaluate our models quantitatively against physical laboratory experiments (Figure 7.1), as well as perceptually in artistic contexts. For the latter, we show complex effects including hair flipping, animal shaking, and rapid rotation of a car wash roller in water (§10).

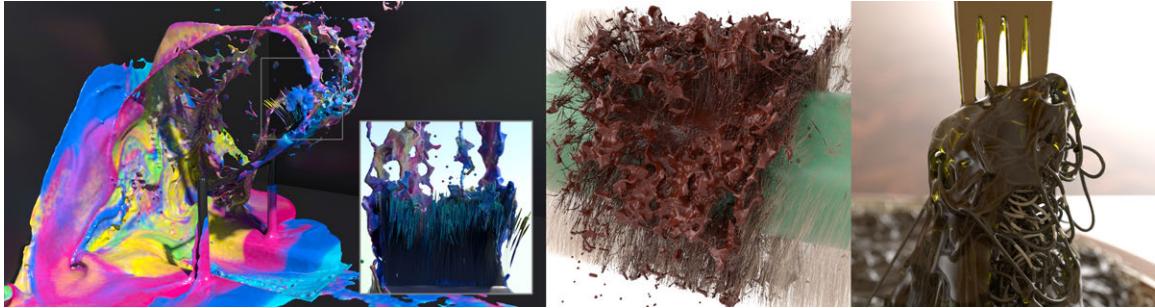


Figure 1.3: Simulated results of strands coupled with shear-dependent liquids. **Left:** A rotating brush splashing through thick oil paint. The inset shows a zoomed view of paint on the bristles. **Middle:** Melted chocolate poured onto a hair-covered cylinder that rotates to mimic the shaking behavior of mammals. **Right:** Soba noodles covered with oyster sauce pulled upwards by a fork.

1.6 Interactions between Strands and Shear-Dependent Liquids

In **Part III**, we generalize the models developed in prior sections and propose models and algorithms to capture the rich dynamic interactions between the shear-dependent liquids and strands (Figure 1.3), which is useful for simulating, for example, a brush stirring and spreading oil paint, shaving cream patted onto a beard, or spaghetti tossed in tomato sauce. These scenarios are challenging to simulate due to the complex rheology of the liquid and the intricate interactions between the liquid and the strands.

A key element of what makes these situations unique is that shear-dependent liquids have strain- and time-dependent viscosity. For example, the macroscopic behavior of shaving cream derives from the microscopic bubbles that it comprises. The stochastic rearrangement of the bubbles causes the liquid to exhibit shear-thinning ef-

fects, in which the continuum begins to flow more easily under high stress. In addition, shaving cream can undergo rate-dependent permanent deformations, a characteristic behavior of viscoplastic flows. These properties contrast starkly with Newtonian fluids in which viscous forces have a simple linear dependence on velocity.

Therefore, we need to extend the liquid-hair interaction model proposed in the previous two chapters. We first derive a set of 1D equations that simulates the shear-dependent liquid [102] on the hair surface §13.2). We then derive a principled coupling scheme (§13.5) between the compressible, non-Newtonian liquid, simulated as a continuum, and strands, simulated as DERs. Finally, we specialized a shifted cone model [2, 65, 87, 119, 125, 195, 230] implemented in ADONIS [129, 130] and So-Bogus [30, 66, 69] to deal with the adhesive-frictional contact between wet strands (§13.6), which provides better stability and a non-tunneling guarantee over the penalty method used in the prior chapters. More importantly, the non-smooth mechanics behind this model delivers a realistic frictional effect between wet strands.

To evaluate, we explore a series of challenging scenarios (§15), involving splashing, shaking, and agitating the liquid which causes the strands to stick together and become entangled.

1.7 Contributions to Computer Graphics

Here we summarize the contributions of this thesis. In **Part I**, for the first time, a piece of wet fabric is treated as a continuous mixture, with an on-manifold equation

derived from the mixture theory to describe the convection and diffusion of liquid within fabrics. In **Part II**, for the first time, a shallow water equation is solved on a strand, and the cohesion force between strands is computed analytically. In **Part III**, we develop the first 1D reduced model for elastoviscoplastic and shear-dependent liquids. In the same section, for the first time, the coupling between complex liquid and discrete elastic rods is rigorously derived and analyzed.

In addition to these novel contributions on physical models, in each part of this thesis, we also develop efficient numerical methods to solve the equations. In **Part I**, we introduce a splitting scheme that approximates an extensive, ill-conditioned, asymmetric system with three symmetric positive definite systems, making the equations from mixture theory effectively solved. In **Part II**, for hairs with penalty-based collision, we introduced a local-global solver that is $120\times$ faster than a diagonally-preconditioned conjugate gradient (PCG) solver and $14383\times$ faster than a direct LDLT solver. Also, in **Part III**, we derive an analytic plastic flow model that enables a semi-implicit integration of the liquid, which allows a time step $24\times$ of an explicit solver for shear stress, and $45\times$ of a full explicit solver.

We have published most of the results of this thesis in leading academic conferences in the field of computer graphics. More specifically, we have published three papers in SIGGRAPH 2018, SIGGRAPH 2017, and SIGGRAPH Asia 2019, respectively, for **Parts I**, **II**, and **III**. Major media have reported exclusively about our work, including *Deutschlandfunk*⁴, a nationwide radio station in Germany, and 80

⁴<https://www.deutschlandfunk.de/computergrafik-digitales-auswringen-von-nassen-101.html>

*Level*⁵, a top tier media in the gaming industry.

Our research also has wide applications in the film, architecture, and cosmetic industries. Several industry-leading companies have adopted or shown interest in our research work. Particularly, the author of this thesis received credit for the movie *Alita: Battle Angel (2019)* where his name appears under “IT, technology & research, and supporting departments” for the adoption of the wet hair simulation algorithm in Weta Digital. The approval and utilization of the research achievements can also be reflected by the popularity of the affiliated open-source code libraries released on GitHub, namely, the *libWetCloth*⁶, *libWetHair*⁷, and *CreamyStrand*⁸, which correspond to the techniques introduced in **Parts I, II** and **III**, respectively. In total, hundreds of developers worldwide have followed these research projects and kept track of their latest updates.

`handtuechern.684.de.html?dram:article_id=433457`

⁵<https://80.lv/articles/a-multi-scale-model-for-simulating-liquid-hair-interactions/>

⁶<https://github.com/nepluno/libwetcloth>

⁷<https://github.com/nepluno/libWetHair>

⁸<https://github.com/nepluno/creamystrand>

Chapter 2

Related Work

2.1 Cloth and Yarn Simulation

Cloth simulation has a long history in computer animation; we refer to the survey of Thomaszewki et al. [233] for a thorough review. Two of the key aspects of a cloth simulation system are the numerical model for the cloth dynamics and the approach used for contact- and collision-handling. In **Part I**, we adopt the discrete shell model [44, 92] to treat the bending and stretching of cloth, based on its simplicity and effectiveness. To handle the contacts and collisions between clothes, we make use of Jiang et al.’s [121] recently proposed method, which exploits a background volumetric grid to efficiently treat contact forces among complex colliding materials. Nevertheless, our approach presented in **Part I** is not intrinsically dependent on these choices, and should be compatible with other cloth simulation frameworks.

Because real cloth is composed of many individual threads, a more costly but potentially much more faithful strategy is to simulate every strand of yarn or thread. This was first suggested by Kaldor et al. [127], and further explored by Kaldor et al. [126] and Cirio et al. [57] with more efficient treatments of inter-yarn contact. As

noted above, in **Part I**, we instead make use of Jiang et al.’s [121] work that models yarns as Lagrangian rods, but handles complicated collisions on a grid.

2.2 Strand Simulation

Decades of effort have gone toward simulating hair strands. A survey of earlier hair simulation work can be found in the reviews of Ward et al. [247] and Hadap et al. [97]. Popular models for single-strand dynamics include mass-spring models [183, 210], various Cosserat rod models [91, 223], the Super-Helix model [31], and the discrete elastic rod (DER) model [26, 28, 126]. Given that some strands (e.g., human hairs) are stiff and often do not exhibit significant stretching, a number of efficient constraint-based models have also recently been developed [72, 98, 136, 169, 221]. Additionally, Iben et al. [107] proposed an artistic way to animate curly hairs. As for collision between hair strands, Kaufman et al. [129] combined a nonlinear integrator with a Gauss-Seidel collision solver, whereas Gornowicz and Borac [90] proposed a hybrid algorithm for higher performance and better stability. In **Parts II** and **III**, to simulate the underlying dynamics of hair, we adopt the DER framework, which has also been exploited in mechanics and robotics research [117], for its efficiency and accuracy.

A complex step of using DER is computing the gradient and Hessian of the discrete curvatures (which are then used to compute $\mathbf{f}_{\text{int}}^t$ and \mathbf{H}_s). Prior works have provided the formulas of some terms [26, 28, 126], however, without details on their derivation. Jawed et al. [117] have provided a derivation in their book, which, however, inherits

a problematic setting of the bending energy and discrete curvatures from Bergou’s work [28]. To eliminate any future confusion and difficulty on the reproduction, we begin with the correct settings of the bending energy and discrete curvatures proposed by Kaldor et al. [126], and present a rigorous derivation in Appendix I.

2.3 Cohesion between Wet Strands

At small scales, surface tension forces on liquid-air interfaces exhibit *elastocapillarity*, in which liquid “bridges” [163] arise that can deform elastic solids, as Bico et al. [34] surveyed recently. In other words, a continuous body of liquid combines the two solids together, often taking on a minimal surface-like shape. A liquid bridge induces attractive forces between the solids that it joins together because surface tension acts to minimize the total exposed liquid surface area.

This effect is responsible for the cohesion of wet hair and wet cloth, leading to the familiar effect of hair clumping (or fiber coalescence) [33, 76, 191, 218]. Brute-force volumetric simulation has been used to study the behavior of liquid bridges and the resulting forces [240]. The cohesive behavior of liquid bridges is an instance of a broader set of phenomena driven by *elastocapillarity*, or the interplay between surface tension in liquids and elasticity in solids [201].

In **Part I**, we adopt a simple cohesive model for wet cloth to approximate the perimeter of the wetted area and calculate the corresponding cohesion force. Although the cohesion between two planar objects has been extensively researched [244], this

effect was not methodically studied on textiles until recent work by Lou et al. [152–154]. The authors considered liquid bridges with a circular area, and showed that the coalescence force between a textile and water increases monotonically with the perimeter of the circular wetted area.

Princen has systematically discussed the capillary phenomena in assemblies of parallel cylinders [187–189]. More recently, Liu et al. [150] generalized Princen’s analytical formulas to two cylinders with an arbitrary radius. Following their work, together with the empirical criterion of liquid bridge-breaking [147], we develop in **Part II** an analytical formula for the cohesion force between two strands, which is computationally cheaper than a full simulation of the liquid bridge.

In computer animation, the cohesion is usually modeled as a penalty-like force [148, 149]. However, the penalty model is known to have multiple issues. First, it requires a finely-tuned stiffness parameter to balance collision and cohesion effects, and thereby avoid both instability and tunneling. Second, wet strands also tend to have a higher friction coefficient than dry strands [32]; handling friction accurately tends to be difficult for penalty models.

Prior work on rod-contact using constraints [30, 69] can avoid tunneling as long as the detection phase captures all the contacts and the solver converges. More recent approaches [90, 129] have improved stability using nonlinear Newton solvers. To solve the difficulties brought by the penalty model, Gascón et al. [87] adopted non-smooth constraints from prior work in computational mechanics [195] to solve

cohesion and collision altogether. In their method, the constraints modeled with second-order Coulomb cones were shifted in their normal directions to produce the adhesive effects [195], which can be applied to simulating Jell-O and wet cloth. Besides discrete elements, a similar idea has been also adopted for simulating continuous media. For example, Daviet [65]¹ and Tempubolon et al. [230] adopted a shifted cone to produce the cohesive effect in granular flows.

In computational mechanics, the use of non-smooth contact dynamics to handle adhesion has been widely adopted [195], usually for simulating cracks [119, 125]. Especially, the adhesive effect in Coulomb's law has been extensively discussed in the book by Acary and Brogliato [2]². In these works, the second-order Coulomb cone is also displaced in its normal direction to achieve the adhesion effects.

In **Part III**, we adopt and extend the shifted cone model [2, 65, 66, 87, 119, 125, 130, 195, 230] to simulate wet hairs covered with cohesive elastoplastic liquid. To prevent the re-cohesion of cracked parts, the work by Jean et al. [119] used different cohesion coefficients based on whether two colliders are separating. Inspired by their work, in particular, we introduce contact hysteresis on *both distance and relative velocity* between two colliders. In addition, we introduce a cohesive force designed for viscoplastic materials.

Also, a few authors have treated strands or thin shells as a fluid-like material and hence adopted continuum-based collision handling [95, 96, 121, 143, 162], where

¹Page 120

²Page 158, chapter 3.9.4.4.

contact and friction are solved globally on a uniform grid. These methods either need an extra pass of the traditional penalty-based collision handling or can be numerically difficult for simulating stiff strands. With a constraint-based solver, however, the contacts are solved iteratively, then the impulses are integrated on each strand individually, which can be much more efficient, especially when each strand has a limited number of vertices and collisions. In all examples presented in this dissertation, we have at most 80 vertices on a single strand, which is typical in a strand simulation [129].

2.4 Liquid Simulation

Recently, hybrid grid/particle-based methods [82, 104, 122, 267] for simulating liquids have become popular due to their visual plausibility and efficacy for pressure computation. Bridson’s [43] book includes a complete review of the basics of these methods for fluid simulation via computer graphics. A common alternative also frequently used for liquid-cloth and liquid-strand interactions is the family of the pure Lagrangian smoothed particle hydrodynamics (SPH) methods [108, 166, 168]. Although SPH can also be used to simulate liquids with complex rheology [257], throughout this dissertation we adopt a hybrid method in which our use of a grid simplifies the momentum exchange between the liquid and the thin structures, especially when both phases need to be integrated (semi-)implicitly for a better stability.

The particle-in-cell method for fluid dynamics was extended to handle general elastoviscoplastic materials, leading to the material point method (MPM) [123, 228].

Models based on MPM have been extensively used in computer animation to simulate various materials and phenomena [67, 88, 104, 135, 172, 194, 225, 226, 254]. More recently, some variants of the MPM have been proposed to increase computational efficiency [78, 84, 86, 104]. One recent approach coupled the MPM-simulated granular material with the granular material simulated with the discrete element method (DEM) to deliver more detailed dynamics [262] at modest cost. In **Part III**, we adopt the moving-least-squares MPM (MLS-MPM) [104] as our discretization scheme for its efficiency, and couple this MPM with DERs. To avoid nullspace issues when solving for pressure, we adapt MLS-MPM to a staggered grid, similar to the work of Stomakhin et al. [226].

To handle shear-dependent fluids, we need a constitutive model that is both accurate and efficient. In **Part III**, we adopt the constitutive model from Yue et al. [261], which adopts the Herschel-Bulkley model [102] and von Mises yield condition [164], as well as employing a $J2$ -flow model for plasticity [215]. The Herschel-Bulkley model covers a wide range of materials such as mud, cream, chocolate, and pasta sauce. In Yue et al.’s [261] work, the stress computed from this model was integrated explicitly. However, this leads to tiny time steps that are at odds with the larger time steps that are possible with DER. Therefore, we develop an analytical formulation of plastic flow for Herschel-Bulkley fluids, which enables semi-implicit integration of the stress and dramatically increases the size of a stable time step.

2.5 Depth-Averaged Flows

Depth-averaged models for fluid flows have been an active topic of research for centuries. For example, the original shallow water equations [205] describe flow on planar boundaries where the vertical velocity is negligible; the Hele-Shaw flow [100] describes non-inertial flow between two thin plates; and lubrication theory [176] is used to model the dynamics of thin liquid whose viscosity dominates over inertia. Ockendon's [175] book includes a thorough introduction to such reduced flows. In computer animation, Wang et al. [242] generalized the shallow water equations to mesh surfaces. Segall et al. [209] proposed an efficient model for the Hele-Shaw flow using generalized barycentric coordinates.

Another alternative assumption in the literature is that of *thin film* or *coating* flow, which is similar to the creeping regime, but with convection considered. In this case, the Navier-Stokes equations can be reduced to a fourth-order partial differential equation (PDE) in which viscosity dominates and inertia is negligible [59]. Azencot and Vantzos [239] proposed a numerical scheme to efficiently evolve thin film flow on arbitrary meshes. More recently, they proposed a *fractional step* scheme to accelerate the computation for real-time applications [238].

Research in textile engineering has studied how the pores in cloth and yarn affect the behavior of liquid propagation, or *wicking* [134]. Cloth and yarn are usually modeled as capillary tubes and the classic Lucas-Washburn equation [156, 249] has been widely applied to the prediction of the position of the hydraulic head in one-

dimensional scenarios. Masoodi and Pillai's [159] book includes a detailed discussion on this topic. Modern research focuses on the experimental estimation of the *capillary radius* [61, 160], which is the effective radius of pores, and on modeling the *suction tensor* that describes the stress due to surface tension [207]. Chwastiak [56], Amico and Lekakou [7], and Williams et al. [252] studied wicking along fibers, whereas Senoguz et al. [211], Ahn et al. [4], Lekakou and Bader [144], and Pillai and Advani [184] studied wicking across fibers. Further work has studied the suction tensor in woven and non-woven fabrics [4, 132].

Inspired by these previous works, in **Part I** we introduce a generalized variant of the Richards equation [198] on cloth, yarn, and junctions between them. In addition, we adopt a general model for textiles from Masoodi and Pillai [158] and propose to construct the anisotropic suction tensor by aligning to specific axes. Several of our examples show the effect of wicking in cloth or yarn.

In **Part II**, we adopt the shallow water flow for liquid on wet hairs. Indeed, according to Q. Wang et al. [243], who studied the behavior of paint brushes, liquid flows mostly on the hair surface or between hairs, rather than *inside* the hair itself. Barba et al.'s [20] experimental studies likewise indicate that flow inside hair cuticles is very slow (on the order of hours) and contributes only to secondary effects. Although interpreting hair as a porous material is a reasonable approximation, inertial effects must still be considered for porous media at moderate to large Reynolds numbers, as discussed by Hellström and Lundström [101]. This is certainly the case for the wet hair flipping or spinning car wash brushes we consider, for example. We therefore

choose to treat flow along individual hair strands with a shallow water model that includes inertia. Our shallow water model is adapted from H. Wang et al. [242], who considered generalized shallow water on triangle mesh surfaces rather than on rods.

Research on depth-averaged models for non-Newtonian liquid is relatively sparse [99, 206]. Recently, Ionescu et al. extended the shallow water model to handle general viscoplastic liquids [109–113]. Prior work, however, did not consider the elastic deformation of the liquid. In **Part III**, we draw inspiration from these approaches to derive a depth-averaged model for elastoviscoplastic liquid on strands, where the liquid deforms purely elastically before yielding and deforms plastically afterward.

2.6 Theory of Mixtures

The history of modeling porous media can be traced back to the late 18th century, when empirical models for fluid and porous solids were adopted to solve hydraulics problems for architectural designs [255]. A review of works in this field’s earlier era can be found in Bedford and Drumheller’s [25] survey and de Boer [70]’s book. Some physical models developed during this era are still widely used today in numerical simulations. For example, Fick’s second law [79] can be used to describe moisture transmission through homogeneous fabric material [64]. Darcy’s law [62], which can be used to calculate the velocity of liquid through porous media for a given pressure drop, viscosity coefficient, and permeability, is a popular choice for the calculation of viscous drag in a numerical simulation [18].

Early soil mechanics researchers studied the effect of water pressure on soil. Filtlunger [80] and Terzaghi [231] found that the total stress applied on a mixture is the combination of the *effective stress* (compression and shear resistance) and the pore *water pressure*, an effect which is now known as *Terzaghi's principle*. Later Biot [35] combined Terzaghi's principle with linear elasticity and fluid dynamics to develop the theory of dynamic poroelasticity (sometimes called Biot's theory), which became the foundation of mixture theory [8]. Mixture theory was initially developed for saturated porous media with incompressible solids, where the interaction forces between porous solids and liquid include two parts: drag and pore pressure. Pore pressure is usually formulated as the pressure gradient applied to the solid and fluid with their respective volume fractions [186]. Recently, Borja [39] generalized mixture theory for unsaturated porous media with compressible solids, and formalized it in a mathematical framework [222]. Borja's formulation has been used in several papers simulating two-way coupled porous media. For example, Abe et al. [1] used the MPM to solve the generalized Darcy equation to simulate creeping flow in porous soil. Bandara and Soga [18] later extended this method to include the inertial effects of liquid to address porous media undergoing large deformations. Daviet and Bertails-Descoubes [68] combined mixture theory with an implicit non-smooth treatment of the Drucker-Prager rheology to simulate immersed granular flows.

Traditional mixture theory treats both the liquid and solid as continuous phases [8]. A more recent method, called CFD-DEM, instead integrates over the solid simulated as discrete elements and coupled with the liquid simulated as a continuous phase [264,

265]. This method is evolved from the mixture theory. Due to its wide applicability to fluidization and pneumatic conveying, CFD–DEM has become increasingly popular for the simulation of particles immersed in flowing fluids.

In **Part I**, we introduce a method coupling wet cloth with a liquid built on mixture theory. We derive a two-scale mixture model targeting bulk fluid and diffusive porous flow, respectively, to simulate thin, unsaturated porous media undergoing large deformations. For the fluid inside cloth/yarn, instead of combining another model, we show that the diffusive flow is a specific case of the equations for bulk liquids and solids, and can be derived from mixture theory.

In **Part III**, we introduce a framework coupling wet strands with liquid built on CFD–DEM, where strands simulated as DERs are coupled with liquid through homogenization.

Fluid Structure Interactions in Animation

When sufficient computational power is available, fine-scale simulations of the interactions between structures (such as hair or fabrics) and grid-based fluids can be performed, often using embedded or immersed boundary methods [165]; conceptually analogous solid-fluid coupling schemes have been applied to animation problems involving rigid bodies, deformables, or cloth (e.g., [22, 200]). However, to properly handle fine structures, the fluid would have to be simulated on a grid that is sufficiently fine and in a context where the fine structure spans at least a few cells. While such brute-force methods have the potential to be extremely accurate, and potentially

useful for deriving coarse-scale models, this strategy is far too costly to be directly applied to scenarios with hundreds or thousands of hairs.

Another branch of research has focused on carefully handling of boundary conditions for water interacting with *impermeable* thin shells, for both Eulerian and Lagrangian fluids. In the context of Eulerian methods, Guendelman et al. [93] used a variable-density pressure solver to account for weakly coupled interaction forces, whereas Robinson et al. [200] proposed a strong coupling approach by temporarily lumping together the momentum of thin shells and fluid. Azevedo et al. [16] used conforming interpolation and exact cut cells to prevent fluid from crossing over impermeable thin boundaries. Among the SPH methods, Akinci et al. [6] carefully sampled thin deformable objects with SPH particles to improve the accuracy of pressure forces and ensure that the cloth remains impermeable to liquid, assuming appropriate time step sizes. In contrast, Huber et al. [105] used the cloth triangle mesh itself directly, combining repulsion forces and continuous collision detection to strictly enforce impermeability. Throughout this dissertation, we look at fabrics and strands whose thickness (or diameter) is much less than the cell size and therefore requires a weak coupling approach that uses drag and buoyancy forces to transfer momentum between liquid and thin structures.

Diffusion Flow in Animation

In computer animation, the earliest explorations of wetting effects addressed painting techniques or the simulation of flows on static planar objects. Curtis et al. [60]

simulated shallow water on paper textiles for watercolor painting effects and solved a diffusion equation to treat capillary effects that capture the spreading of fluid through pores in paper. Later, Chu and Tai [55] proposed a sophisticated system to simulate the ink percolation process. In their work, they designed permeability and boundary conditions based on artistic considerations. Instead of solving a simple diffusion flow, Huber et al. [106] solved Fick’s second law on cloth with an additional gravitational term, and also demonstrated liquid absorption.

Fluid-solid interaction is a many-faceted phenomenon, and some previous works have therefore sought to address one or two of those facets in isolation. With an approach relying on fractional derivatives, Ozgen et al. [177] simulated the deformation of a completely submerged cloth without simulating water at all. Chen et al. [52] proposed modified saturation, wrinkling, and friction models to better approximate the look of wet clothing. Um et al. [236] combined a shallow water model and the diffusion equation to address fluid flow on and within dynamic cloth.

Darcy-Type Porous Flow in Animation

Lenaerts et al. [145] proposed the simulation of more general deformable wet materials. They used an SPH method to solve porous (Darcy) flow inside a solid object. Similarly, Rungjiratananon et al. [203] considered fluid interactions with dynamic porous media in the context of wet sand, simulating sand, water, and their mutual interactions using SPH. Rungjiratananon et al. [202] also proposed two-way interactions between SPH-based water and a shape-matching hair model using a porous flow approach for the

propagation of liquid inside hair. They discretized the porous flow model on a regular Cartesian grid in a bounding box surrounding the hair, adapting earlier SPH-based porous flow techniques [145]. Subsequent research focused on various simplifications intended to achieve higher performance. Saket and Parag [179] presented an SPH method for the simulation of wet cloth, using a geometric diffusion method to simulate interior flow for increased efficiency. Lin et al. [148, 149] proposed a similar porous flow model with SPH but further incorporated two-way fluid-hair interactions.

Mixtures in Animation

Mixture theory was first introduced for animation by Nielsen and Østerby [173], who simulated fluid spray and air as continua. Later, Ren et al. [196] and Yang et al. [258] proposed an SPH-based framework to handle a wide range of multi-fluid flow phenomena including extraction and partial dissolution. Yan et al. [256] generalized the multi-fluid SPH framework to incorporate solids, adopting a diffusion model for the relative motion between a solid and liquid. More recently, Yang et al. [257] extended their previous SPH framework with a phase-field method to simulate phase-changing phenomena for multi-materials. Their method exchanged the momentum between different phases by incorporating a viscous term between particles, and inside each particle, different materials shared the same momentum. In **Part I**, we adopted a similar physical model, but solved the equations on both polygonal meshes and an Eulerian grid to capture the diffusion and pressure forces more accurately, and incorporate stiff elastoplastic materials with large drag forces more effectively.

The authors of recent works on simulating porous sand mixed with water [85, 230] adopted a formulation by Bandara and Soga [18] to compute buoyancy forces, but concluded that buoyancy is largely negligible in their problem. For the phenomena considered in this dissertation, the buoyancy force significantly affects the motion of both fabrics and hairs.

Drag Force for Two-Phase Continuum Media

Drag models between two continuous phases have a long history. Some were even developed before mixture theory. Forchheimer [81] extended Darcy's (linear) drag model with a quadratic model for high Reynolds number flows. Ergun [77] extended the empirical Kozeny-Carman equation [48] (another extension of Darcy's law for modeling linear permeability) and proposed a non-linear version that is a function of the Reynolds number. The Ergun equation can also be reformulated to discover the relationship between linear and non-linear drag forces, which can be applied to various materials [5, 174]. In **Part I**, we adopt a modern, unified drag formulation [259], and use the Ergun equation to relate the linear and non-linear terms. In addition, we calculate the permeability of fibers following Stylianopoulos et al.'s [227] empirically determined equations. This scheme is appropriate for liquid at both low and high Reynolds numbers, and confirms with dimensional analysis that our drag force is physically consistent.

Drag Force between Continuum and Discrete Elements

Drag forces used for CFD–DEM are most often formulated for a *single discrete element*. These drag forces, however, also differ widely in terms of the particular *drag models* and *drag coefficients*. Among various drag models, Di Felice’s is widespread and reasonably general [73]; it also suits our needs because it is compatible with different drag coefficients and fits experimental data for both spherical and non-spherical particles [103], as well as for either Newtonian or non-Newtonian fluids [146]. In **Part III**, we demonstrate how to use Di Felice’s formula to compute drag between strands and liquid.

The accuracy of the drag force, finally, depends on choosing the *drag coefficient*. Among the abundant literatures [264], we focus on the drag coefficients applicable to complex fluid and irregular particles. Chhabra’s [54] book presents an extensive survey of these studies. In **Part III**, we show our adoption of Mauret and Renaud’s [161, 197] proposed model, which has been extensively validated against physical experiments [15, 24, 29, 193, 229].

Part I

Multi-Scale Model for Simulating Liquid-Fabric Interactions

Chapter 3

Mixture Model for Wet Fabrics

In this section, we develop a multi-scale framework capturing the interactions between fabric and fluid. To develop a computational model of these varied liquid-fabric interactions, we must understand the composition of the fabric. Fabric is composed of individual strands (“threads” or “yarn”) packed into thin oriented fibers (e.g., Figure 3.1a). Tiny pockets within and between these fibers collect fluid and are largely responsible for the wetting behavior we observe at the coarse scale. Because these pockets are numerous and individually imperceptible to the naked eye, it can be wasteful or intractable to represent them as discrete elements for animation applications.

Therefore, *for the first time*, we model the fabric as a *continuum mixture* of water, air, and fabric material (Figure 3.1). The governing equations for such a continuum are provided by *mixture theory* [192], where we describe the fabric geometry through two scalar fields: the field of fabric volume fraction, and the field of saturation. With such a continuous representation, the calculation is much simpler than simulating liquid interacting with individual fibers.

In the following sections, we begin with a background introduction on mixture

theory and discuss how we adopt mixture theory for simulating cloth fabrics. To model the diffusion of liquid inside the fabrics, we simplify the equations from mixture theory, where we derive a set of reduced-dimensional (2D or 1D) convection-diffusion equations. Finally, we introduce some numeric techniques to solve these equations.

Our contributions include

- the adaptation of mixture theory and porous flow to partially saturated fabrics with buoyancy in a particle-in-cell framework,
- the development of an approximate anisotropic fabric microstructure model to support nonlinear drag and pore pressure forces,
- treatments for liquid capture, and dripping,
- a quasi-static model of fluid flow within the fabric based on convection-diffusion,
- an efficient numerical solver for the resulting complex systems.

3.1 Mixture Theory

Mixture theory models multiphase systems consisting of several interpenetrable continua. The theory assumes that all three phases are present, in some ratio, at every point of the material. The theory develops the momentum and mass balance equations for such a mixture.

As water penetrates, fabric *saturates* from dry to damp to soaked (Figure 3.1). Saturation is the measure that determines *volume fractions*, the relative occupancy of the water, air, and solid fabric.

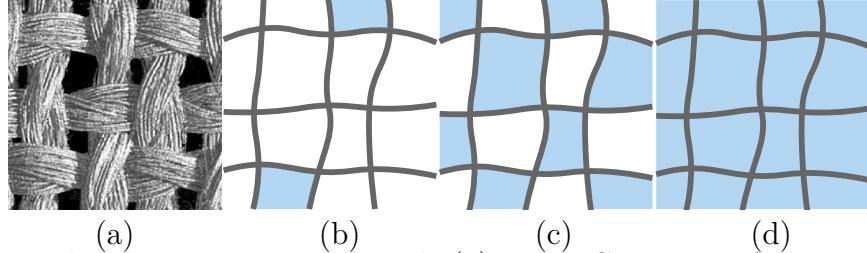


Figure 3.1: **Fabric as porous material.** (a) Micro-CT image of plain woven fabric, adapted from [213]. (b) Barely-saturated fabric ($S_r \approx 0.1$). (c) Half-saturated fabric ($S_r \approx 0.5$). (d) Fully-saturated fabric ($S_r = 1.0$).

Saturated continuity equations In the (maximally) saturated state, fabric pores are entirely filled with liquid [8, 68]. The motion of both the porous medium and the liquid are described by the equations for incompressible mixture, where

$$\rho_s \phi_s \frac{D_{\mathbf{u}_s} \mathbf{u}_s}{D_{\mathbf{u}_s} t} - \nabla \cdot \boldsymbol{\sigma}_s - \rho_s \phi_s \mathbf{g} - \mathbf{f}_{f \rightarrow s} = 0, \quad (3.1a)$$

$$\rho_f (1 - \phi_s) \frac{D_{\mathbf{u}_f} \mathbf{u}_f}{D_{\mathbf{u}_f} t} - \nabla \cdot \boldsymbol{\sigma}_f - \rho_f (1 - \phi_s) \mathbf{g} + \mathbf{f}_{f \rightarrow s} = 0, \quad (3.1b)$$

$$\frac{\partial \phi_s}{\partial t} + \nabla \cdot (\phi_s \mathbf{u}_s) = 0, \quad (3.1c)$$

$$\frac{\partial (1 - \phi_s)}{\partial t} + \nabla \cdot [(1 - \phi_s) \mathbf{u}_f] = 0. \quad (3.1d)$$

Here the fields \mathbf{u} (velocity), ρ (density), and $\boldsymbol{\sigma}$ (Cauchy stress tensor) have values for both the porous medium and the liquid, indicated by their respective subscripts: “s” for the (solid) porous media and “f” for the fluid. The *volume fraction* of the solid in the porous material is given by ϕ_s (so $1 - \phi_s$ gives the complementary non-solid fraction), and \mathbf{g} represents any external forces, such as gravity. The operator $\frac{D_{\mathbf{u}}}{D_{\mathbf{u}} t}$ is the Eulerian material derivative under the flow velocity \mathbf{u} , defined as $\frac{D_{\mathbf{u}}}{D_{\mathbf{u}} t} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$. Lastly, $\mathbf{f}_{f \rightarrow s}$ is the interaction force between the liquid and the solid porous medium. It is this force that we must derive to properly model wet cloth and yarn.

Equations (3.1a) and (3.1b) are the momentum equations of incompressible solid and fluid, respectively, while equations (3.1c) and (3.1d) are the corresponding laws of mass conservation (or continuity equations). We will elaborate below on the solid stress and interaction forces, including buoyancy and drag. But first, we must drop an assumption that we have made.

The continuity equations (3.1c) and (3.1d) assume that pores are fully filled with liquid, and thus the liquid volume in a unit material volume is given by $1 - \phi_s$. How do we model a porous medium *partially* filled with liquid? One way to approach this is to (fully) saturate our porous medium with a fluid that represents both liquid and air components [39].

Consider a fluid mixture of liquid and air. Since the air density is orders of magnitude smaller than the liquid density, we ignore the mass of the air. We assume that the fluid velocity field is shared by the liquid and air components moving in unison. We use the saturation variable S_r to indicate the volume fraction of liquid in the fluid (thus $1 - S_r$ indicates the volume fraction of air in the fluid). In such a mixture, the fluid density ρ_f (recall (3.1b)) becomes a fraction of the water density ρ_w (i.e., $\rho_f = S_r \rho_w$). We can substitute this liquid-air fluid mixture, in place of only liquid, to obtain continuity equations that do not assume liquid saturation.

Unsaturated continuity equation Consider a porous medium that is not necessarily (fully) saturated with *liquid*. Such a medium *is* (fully) saturated with our liquid-air *fluid* mixture. A unit volume of the porous fabric medium is the sum of

three parts,

$$\phi_s + (1 - \phi_s)S_r + (1 - \phi_s)(1 - S_r) = 1, \quad (3.2)$$

where the three terms correspond to the volume fraction of solid, liquid, and air, respectively. The continuity equation (3.1d) of liquid can be modified to account for partial saturation using a slightly different form,

$$\frac{\partial(1 - \phi_s)S_r}{\partial t} + \nabla \cdot [(1 - \phi_s)S_r \mathbf{u}_f] = 0. \quad (3.3)$$

Lastly, subtracting (3.1d) from (3.1c) yields the incompressibility condition for the solid-fluid mixture,

$$\nabla \cdot [\phi_s \mathbf{u}_s + (1 - \phi_s) \mathbf{u}_f] = 0. \quad (3.4)$$

In summary, equations (3.1a-3.1c) together with (3.3-3.4) form the mixture theory model for unsaturated porous media.

Solid Stress. The effect of porosity on solid stresses is that, under the same deformation, the effective stress σ_s of a porous solid material is smaller than the corresponding stress σ_c exhibited by a densely packed or non-porous material (i.e., with zero porosity). Given an applied deformation (or strain), σ_c can be evaluated using a particular constitutive model, the choice of which depends on whether we are simulating wet cloth or yarn (see §4). The relationship between σ_c and σ_s has been experimentally and numerically established by Makse et al. [157], namely, $\sigma_s = \phi_s^\lambda \sigma_c$, where the parameter λ is material-dependent, usually taking values from $1 \sim 3$. In

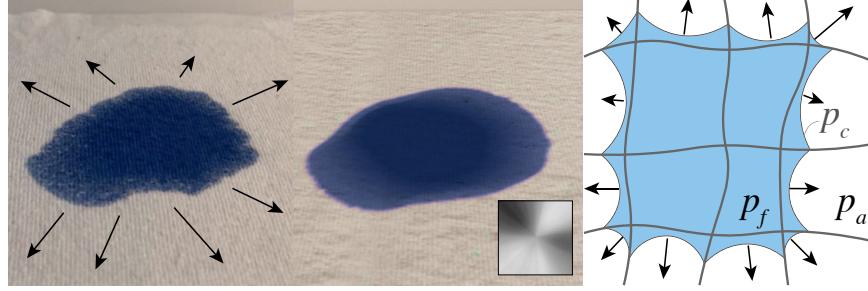


Figure 3.2: **Pore pressure example.** Consider a piece of fabric lying on a table. The fabric is wet initially in a circular region. Near the boundary of the circle, the saturation S_r changes from zero to one, along the directions indicated by the arrows. On the **left** is a real photograph, and in the **middle** is our simulated result. To better match the laboratory result, we experimented with using a manually specified volume fraction field on the textile (visualized as an inset figure). In the **right**, P_c is the pore pressure introduced by the water-air surface between the textile fibers. Because of the change of S_r along the radial directions, the gradient $\nabla(1 - S_r)P_c$ in (3.8) generates interaction forces between the textile fibers and the water between. Macroscopically, these forces point along the directions of the arrows. Since the forces are mostly uniform in all directions, the fabric remains static, but the water spreads outwards.

all our examples, we use the value $\lambda = 2$.

Interaction Forces. There are two relevant types of interaction forces between solid and liquid [8]: the pressure gradient force $\mathbf{f}_{f \rightarrow s}^p$ and the drag force $\mathbf{f}_{f \rightarrow s}^d$. The total interaction force is

$$\mathbf{f}_{f \rightarrow s} = \mathbf{f}_{f \rightarrow s}^p + \mathbf{f}_{f \rightarrow s}^d. \quad (3.5)$$

The pressure gradient acts when cloth and yarn are submerged (Figure 3.3). The drag force, on the other hand, is due to liquid-solid friction and wake turbulence. The next two subsections are dedicated to our derivation of the specific forms of these forces for wet cloth and yarn.



Figure 3.3: Comparison with and without the liquid pressure gradient applied to clothes. A simulation with the liquid pressure gradient applied to cloth yields correct buoyancy (left), where cloth lighter than water floats and cloth heavier than water sinks. Without the pressure gradient, all the cloth erroneously sinks (right).

3.2 Pressure Gradient

In a saturated solid-liquid mixture, the pressure gradient is

$$\mathbf{f}_{\text{f} \rightarrow \text{s}}^{\text{p}} = -\phi_{\text{s}} \nabla p. \quad (3.6)$$

Here we neglect the liquid stress induced by the porous solid, a standard assumption for open pores [186]. The liquid pressure p is smoothly varying except for a jump at the liquid free surface induced by surface tension.

An *unsaturated* porous medium has tiny air pockets, for which the surface tension force exactly balances the liquid-air pressure jump. The myriad air pockets make for a markedly more complex liquid surface, since air and liquid are present “everywhere;” the jumps due to surface tension are densely distributed and more appropriately captured in a homogenized force balance, $p_{\text{f}} - p_{\text{a}} = p_{\text{c}}$, referring to the liquid, air, and

pore pressure, respectively [39].

From mixture theory, the effective pressure p of an unsaturated porous medium is given by weighting the component pressures by the saturation S_r [39],

$$p = S_r p_f + (1 - S_r) p_a. \quad (3.7)$$

Substituting the force balance $p_f - p_c = p_a$ into (13.72), then into (3.6),

$$\mathbf{f}_{f \rightarrow s}^p = -\phi_s \nabla p = \underbrace{-\phi_s \nabla p_f}_{\text{buoyancy}} + \underbrace{\phi_s \nabla ((1 - S_r)p_c)}_{\text{pore pressure}}. \quad (3.8)$$

The first term of the pressure gradient governs buoyancy, the force that pushes lighter objects up toward the fluid surface. As we can see, the second term is present only for an unsaturated porous medium ($S_r < 1$). We now explore this *pore pressure* term.

Suction Tensor. It has been confirmed [207] that pore pressure depends on the porous solid microstructure. Here, we develop a pore pressure model suited to our application.

The void space between textile fibers, which is oriented along individual yarns, yields an *anisotropic* microstructure. Consequently, our pore pressure is also anisotropic, and must therefore be described by a second-order tensor rather than a scalar. This tensor is called the *suction tensor* in the mechanics literature.

Drawing on the literature on porous flow through fibers, we propose a model for

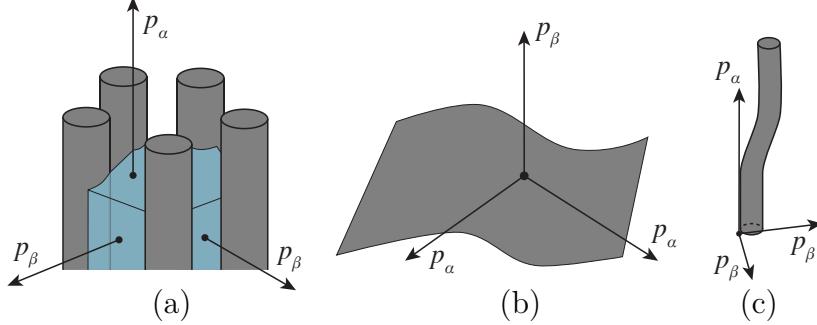


Figure 3.4: **Fiber pack, cloth and yarn orientation.** In our derivation of the pressure gradient and drag forces, we use a canonical frame of reference to orient the fiber pack, cloth, and yarn. Cloth and yarn in arbitrary orientations are first rotated into this frame of reference to compute the force tensors, and then rotated back to their original frame.

the suction tensor specialized to the case of cloth, yarn, and combinations of the two. Consider a pack of fibers along a yarn segment (Figure 3.4-a). When mixed with water the void spaces between individual fibers effectively form capillary tubes that act to transport water. The pore (or suction) pressures along the fiber direction and the perpendicular direction are, respectively,

$$p_\alpha = \frac{2\phi_s \gamma \cos \theta}{(1 - \phi_s)r_b} \text{ and } p_\beta = \frac{p_\alpha}{2}, \quad (3.9)$$

as Masoodi and Pillai [158] introduced and experimentally verified. Here ϕ_s is again the volume fraction of the capillary tubes (in our case the textile fibers), γ is the surface tension coefficient of liquid (i.e., $\gamma = 72.0 \text{ dyn/cm}$ for water), θ is the equilibrium contact angle between liquid and the fibers, and r_b is the radius of the capillary tubes.

Adapting this concept to our setting, we note that if a yarn segment is aligned along the Z -direction, we can write its suction tensor as a diagonal matrix whose

diagonal elements are $[p_\beta \ p_\beta \ p_\alpha]$. Similarly, in a small piece of cloth with its *normal* aligned along the Z-direction (Figure 3.4-b), the textile fibers are instead oriented along the X- and Y-directions. Then, the suction tensor is another diagonal matrix with diagonal elements $[p_\alpha \ p_\alpha \ p_\beta]$. When individual yarn strands extend perpendicularly from the cloth surface (Figure 3.4-c) — for example, when simulating a thick-textured fabric such as terrycloth — we express the suction tensor as a weighed combination of both diagonal matrices,

$$\hat{\mathbf{P}}_c = s_f \begin{bmatrix} p_\alpha & 0 & 0 \\ 0 & p_\alpha & 0 \\ 0 & 0 & p_\beta \end{bmatrix} + (1 - s_f) \begin{bmatrix} p_\beta & 0 & 0 \\ 0 & p_\beta & 0 \\ 0 & 0 & p_\alpha \end{bmatrix}, \quad (3.10)$$

where we call s_f the “shape fraction”: when we consider the suction tensor in an infinitesimal region of cloth or yarn, $s_f = 0$ if this region is occupied entirely by a yarn strand, $s_f = 1$ if it is entirely occupied by cloth, and s_f lies between 0 and 1 if the region is near the root of a yarn strand extending from a piece of cloth (Figure 3.4-c).

With the suction tensor $\hat{\mathbf{P}}_c$ defined for cloth and yarn in the canonical orientation above, the suction tensor \mathbf{P}_c in an arbitrary orientation will be a rotated version of $\hat{\mathbf{P}}_c$, namely

$$\mathbf{P}_c = \mathbf{R}^T \hat{\mathbf{P}}_c \mathbf{R}. \quad (3.11)$$

In cloth, \mathbf{R} is the rotation matrix that rotates the cloth normal to the Z-direction, and in yarn, \mathbf{R} rotates the yarn tangent to the Z-direction. (In the mixtures we described above, these directions are mutually aligned.)

Finally, having developed our new application-specific definition of the suction tensor, the pressure gradient force in (3.8) can be re-written as

$$\mathbf{f}_{\text{f} \rightarrow \text{s}}^{\text{p}} = -\phi_{\text{s}} \nabla p_f + \phi_{\text{s}} \nabla \cdot ((1 - S_{\text{r}}) \mathbf{P}_{\text{c}}), \quad (3.12)$$

where the divergence of our anisotropic suction (stress) *tensor* has taken the place of the gradient of the *scalar* pore pressure.

Remark. The *total stress*, a frequently used quantity when modeling porous materials such as sand and soil [222], is the sum of the solid stress $\boldsymbol{\sigma}_{\text{s}}$ and fluid stress $\boldsymbol{\sigma}_{\text{f}}$. The equation above effectively states that $\boldsymbol{\sigma}_{\text{f}} = -p_{\text{f}} \mathbf{I}_3 + (1 - S_{\text{r}}) \mathbf{P}_{\text{c}}$, where \mathbf{I}_3 is a 3×3 identity. In §3.1, we saw that $\boldsymbol{\sigma}_{\text{s}} = \phi_{\text{s}}^{\lambda} \boldsymbol{\sigma}_{\text{c}}$. Our total stress is therefore $\phi_{\text{s}}^{\lambda} \boldsymbol{\sigma}_{\text{c}} - p_{\text{f}} \mathbf{I}_3 + (1 - S_{\text{r}}) \mathbf{P}_{\text{c}}$. For saturated and densely packed porous material ($\lambda \rightarrow 0$ and $S_{\text{r}} \rightarrow 1$), our definition of total stress becomes $\boldsymbol{\sigma}_{\text{c}} - p_{\text{f}} \mathbf{I}_3$, which is precisely consistent with the classic Terzaghi's total stress [232].

3.3 Drag between Two Continuous Phases

Force. Liquid flow through a porous solid is resisted by a *drag* force proportional to the relative velocity, $\mathbf{f}_{\text{f} \rightarrow \text{s}}^{\text{d}} = \tilde{\mathbf{C}}(\mathbf{u}_{\text{f}} - \mathbf{u}_{\text{s}})$, where $\tilde{\mathbf{C}}$ is a diagonal matrix of drag coefficients, \hat{C}_i .

For a pack of fibers oriented along the Z axis (Figure 3.4), the drag coefficients along the transversal ($1 \leq i \leq 2$) and lateral ($i = 3$) directions can be expressed in

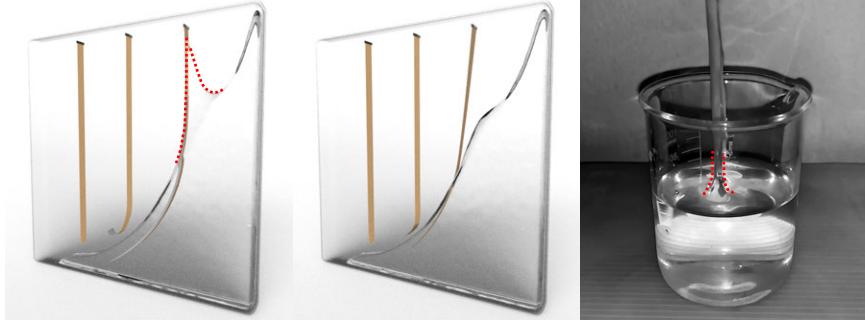


Figure 3.5: **Comparison between nonlinear and linear drag models.** Nonlinear drag (left) exhibits a sharp "kink" (red dashed curve) around the liquid-solid interface due to fast-moving cloth having pulled water with it. Since linear drag is not suitable for high Reynolds number flows, this effect is not seen for linear drag (middle). This effect can, however, be readily observed in physical experiments (right, with red dashed curve).

terms of anisotropic permeability.

Permeability is a measure of the ability of a porous material to allow liquids to pass through it [23]. The permeabilities of flows lateral and transverse to a pack of fibers are [227]

$$k_\alpha = \frac{-\ln\phi_s - 1.476 + 2\phi_s - 0.5\phi_s^2}{16\phi_s} d^2 \text{ and} \\ k_\beta = \frac{-\ln\phi_s - 1.476 + 2\phi_s - 1.774\phi_s^2 + 4.078\phi_s^3}{32\phi_s} d^2, \quad (3.13)$$

respectively, in terms of the volume fraction ϕ_s and fiber diameter d .

Drag Coefficient. Yazdchi and Luding [259] relate the drag coefficient to the permeability of a fibrous material. The drag coefficient \hat{C}_i is normalized by the liquid viscosity η and fiber diameter d to define a dimensionless drag, or modified friction factor, $f_i = -d^2\hat{C}_i/\eta$. Similarly, the permeability is normalized as $K_i = k_id^2$. The

dimensionless drag and permability are related via

$$-f_i = \frac{1}{K_i} + \chi_i \text{Re}_i^c, \quad (3.14)$$

where the exponent $c = 1.6$ is a constant; $\text{Re}_i = \rho_f(\mathbf{u}_{f,i} - \mathbf{u}_{s,i})d/\eta$ is the Reynolds number, where the subscript i of \mathbf{u}_f and \mathbf{u}_s indicates the X -, Y -, or Z -component of the velocity.

The coefficient χ_i weights the nonlinear second term relative to the linear first term. Many models exist for computing χ_i , and we choose to use the classic Ergun equation [77], as validated by Yazdchi and Luding [259]. Replacing the various quantities in (3.14), we obtain our final formula for the drag coefficient,

$$\hat{C}_i = \frac{\eta}{k_i} + \frac{1.75}{\sqrt{150}} \frac{\rho_f^c d^{c-1} \eta^{1-c}}{(1 - \phi_s)^{\frac{3}{2}} \sqrt{k_i}} \|\mathbf{u}_f - \mathbf{u}_s\|_2^c. \quad (3.15)$$

These coefficients allow us to compute the drag force when a pack of fibers are oriented along the Z-direction. Given cloth or yarn with an arbitrary orientation, we construct a rotated drag tensor \mathbf{C} in a similar manner to the suction tensor in (3.11): namely, $\mathbf{C} = \mathbf{R}^T \hat{\mathbf{C}} \mathbf{R}$, where $\hat{\mathbf{C}}$ is a diagonal matrix. For cloth, $\hat{\mathbf{C}}$ has the diagonal elements $[\hat{C}_\beta \hat{C}_\beta \hat{C}_\alpha]$, while in yarn $\hat{\mathbf{C}}$ has the diagonal elements $[\hat{C}_\alpha \hat{C}_\alpha \hat{C}_\beta]$. Here \hat{C}_α and \hat{C}_β are determined by substituting k_α and k_β of (3.13) into the k_i of (3.15), respectively. In general, $\hat{\mathbf{C}}$ is a combination of the two cases, defined in the same way as in (3.10). \mathbf{R} is a rotational matrix that aligns the cloth's normal direction or the

yarn's tangential direction with the Z-direction, as in (3.11).

Finally, the drag force between two continual phases is computed as

$$\mathbf{f}_{\text{f} \rightarrow \text{s}}^{\text{d}} = \mathbf{C}(\mathbf{u}_f - \mathbf{u}_s) = \mathbf{R}^T \hat{\mathbf{C}} \mathbf{R} (\mathbf{u}_f - \mathbf{u}_s). \quad (3.16)$$

Remark I. Dimensional analysis provides a useful sanity check on our derivation.

The value \hat{C}_i in (3.15) has units of $\text{g}\cdot\text{cm}^{-3}\cdot\text{s}^{-1}$ for any positive c value. Therefore,

$\mathbf{f}_{\text{f} \rightarrow \text{s}}^{\text{d}}/\rho_s$ always has units of $\text{cm}\cdot\text{s}^{-2}$, which are precisely the units of acceleration.

Remark II. Drag force models have been used in many computer graphics simulations, yet almost all such models have been linear with respect to the relative velocity. For example, recent work on simulating sand and water mixtures [230] adopts a linear model. Meanwhile, studies in porous mechanics have shown that the drag force is nonlinear, especially when the Reynolds number is not small [158]. In Figure 3.5, we compare our nonlinear drag model in (3.15) with the linear drag model that ignores the second term of (3.15), to demonstrate their very distinct visual difference.

3.4 Dynamic and Quasi-Static Model

Dynamic Model. Putting together all the forces derived above, our mixture model for cloth and yarn is comprised of five equations: namely, the momentum equations (3.1a) and (3.1b); the continuity equation (3.1c) for solid material; the continuity equation (3.3) for the liquid, which also advects the porous saturation S_r ; and

the incompressibility condition (3.4). To complete (3.1a) and (3.1b), the interaction force $\mathbf{f}_{\text{f} \rightarrow \text{s}}$ is the sum of the pressure gradient force (3.12) and the drag force (3.16). We refer to these equations as the *dynamic equations* of wet cloth and yarn. In the next section, we will numerically solve them by discretizing the entire domain of the liquid, wet cloth, and wet yarn using Eulerian grids.

Quasi-Static Model. To capture liquid diffusion and convecting within the thin volume of cloth and yarn, direct discretization of the dynamic equations in 3D necessitates the use of very fine grids, resulting in prohibitive simulation costs. We therefore treat this case specially. We observe that water travels along the cloth and yarn volume *slowly* even while the cloth and yarn might undergo large deformation. This suggests that we can model the liquid motion in the frame of reference attached to the cloth or yarn. In this frame of reference, the Reynolds number is relatively low, so we choose to model the liquid motion *quasi-statically* as diffusion on the codimensional objects (i.e., 2D surface for cloth and 1D curve for yarn). In particular, we ignore the inertia term in (3.1b). We further note that the cloth and yarn material is isotropic in the codimensional space, and hence so is the pressure tensor. Then, Equation (3.1b) after substituting (3.12) and (3.16) can be simplified into

$$\frac{1}{\rho_{\text{f}}} \nabla [(1 - S_{\text{r}}) p_{\alpha} - p_{\text{f}}] - \frac{\hat{C}_{\alpha}}{\rho_{\text{f}}(1 - \phi_{\text{s}})} (\mathbf{u}_{\text{f}} - \mathbf{u}_{\text{s}}) + \tilde{\mathbf{g}} = 0. \quad (3.17)$$

Because the frame of reference is non-inertial, the force $\tilde{\mathbf{g}}$ must now include not only the external force \mathbf{g} but also additional fictitious forces, such as the centrifugal and

Coriolis forces (to be discussed further in §4).

Equation (3.17) allows us to express \mathbf{u}_f with respect to \mathbf{u}_s , p_f , and p_α , by isolating \mathbf{u}_f on one side of the equation. Substituting it in (3.3) yields the equation to be solved in the codimensional space:

$$\frac{\partial \epsilon_L}{\partial t} + \nabla \cdot \left[\frac{(1 - \phi_s)\epsilon_L (\nabla(\tilde{p}_\alpha - p_f) + \rho_f \tilde{\mathbf{g}})}{\hat{C}_\alpha} + \epsilon_L \mathbf{u}_s \right] = 0. \quad (3.18)$$

where $\tilde{p}_\alpha = (1 - S_r)p_\alpha$, and we define $\epsilon_L = (1 - \phi_s)S_r$ as the volume fraction of liquid in the unsaturated mixture. This is a convection-diffusion equation describing how ϵ_L is transported quasi-statically along cloth surfaces and yarn strands.

Remark. If we ignore external forces and the pressure from the bulk fluid, and assume the porous solid is static, then p_f , $\tilde{\mathbf{g}}$, and \mathbf{u}_s in (3.18) all vanish, while ϕ_s remains constant. Then this equation reduces to the famous *Richards equation* in soil mechanics [198], that describes the movement of water in unsaturated soils:

$$\frac{\partial S_r}{\partial t} = \nabla \cdot \left[\tilde{D}(S_r) \nabla S_r \right], \quad (3.19)$$

where $\tilde{D}(S_r)$ is called the *diffusivity*, and is usually some function of S_r and the permeability. In our model, $\tilde{D}(S_r) = \frac{(1-\phi_s)p_\alpha S_r}{\hat{C}_\alpha}$, which has a linear dependence on S_r and corresponds to a linear water retention curve. Other popular models, such as Brooks-Corey [46], van Genutchen [237], or models from experimental data fitting [141], usually assume an infinite suction pressure when S_r approaches zero. We

adopt a linear model since it is effective and numerically stable, and can be derived from a standard modification of mixture theory for unsaturated porous media.

Chapter 4

Numerical Simulation of Wet Fabrics

Having laid down the governing equations, we turn our attention to the numerics.

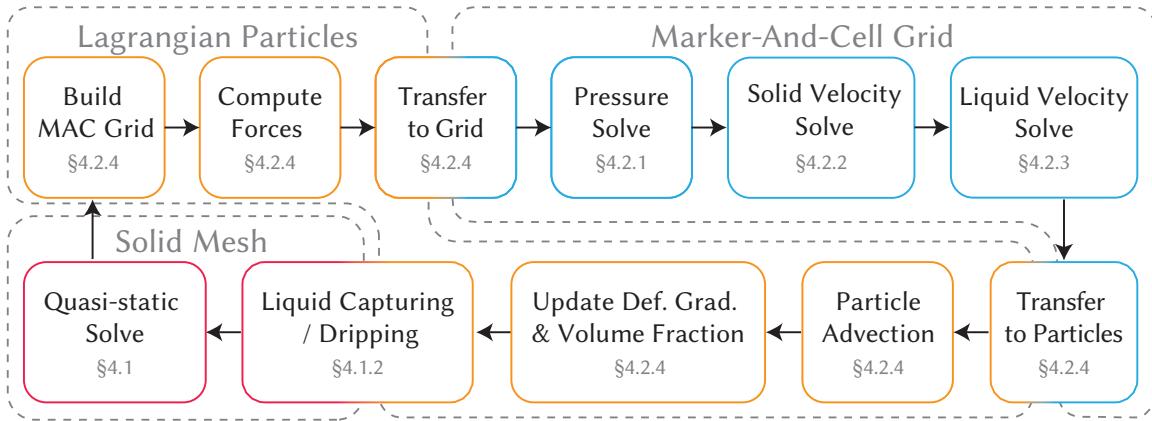


Figure 4.1: **Overview of our numerical method for the mixture of two continuum phases.**

Method Overview. We discretize the quasi-static equation (3.18) over Lagrangian fabric “solid” meshes, and the remaining dynamic equations over a “background” Eulerian marker-and-cell (MAC) grid augmented with Lagrangian particles for advection.

Bulk liquid is simulated using the affine particle-in-cell (APIC) method [122]. Each Lagrangian liquid particle carries a scalar volume and two set of velocities: the liquid velocity \mathbf{u}_f and the solid porous material’s velocity \mathbf{u}_s .

The porous fabric solid is simulated using a Lagrangian mesh, with each vertex

carrying the solid velocity \mathbf{u}_s , a porous volume fraction ϕ , and a liquid saturation fraction S_r . We use the APIC method to distribute data from Lagrangian points (liquid particles and solid mesh vertices) to the Eulerian grid faces, and vice versa.

The elastic forces of the fabric are computed using discrete shells [92] for woven cloth, and discrete elastic rods [28, 127] for knitted garments. These elastic forces are coupled with the background grid using the method of Jiang et al. [121] to resolve collision and frictional forces.

At each time step, our method performs the following steps (see Figure 4.1 for a visual overview of our algorithm):

1. Build the MAC grid (§4.2)
2. Compute solid internal forces and apply the flow-rule for solid plasticity (§4.2),
3. Map liquid and solid particles onto the Eulerian grid (§4.2),
4. Solve the pressure projection (§4.2),
5. Solve the solid velocity (§4.2),
6. Update the liquid velocity (§4.2),
7. Map the liquid and solid velocity back to particles, update solid deformation gradient, advect particles (§4.2),
8. Handle liquid capture and dripping for cloth and yarn (§4.1),
9. Solve the quasi-static equation on solid meshes (§4.1).

4.1 Codimensional Quasi-Static Simulation

Because fabric strand features are $\sim 4\text{-}8\times$ smaller than a grid cell, we solve (3.18) on the Lagrangian meshes directly without relying on the MAC grid. We must consider three types of mesh configurations: (woven) cloth triangles, (individual or knitted) yarn segments, and the junctions between them (see Figure 4.2). Junctions are useful not only for modeling cloth-knit assemblies, but also other non-manifold structures, such as the cloth-yarn connection in a piece of terrycloth (see Figure 5.1c).

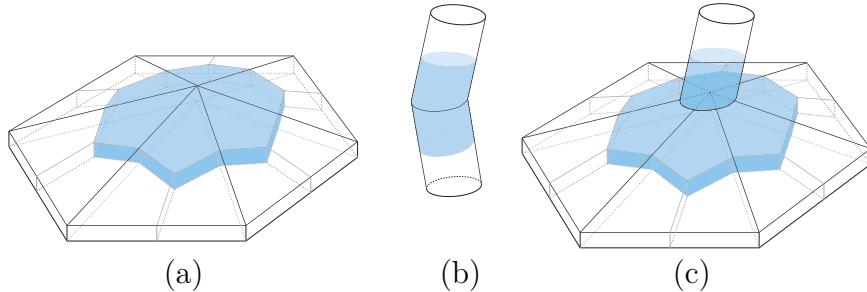


Figure 4.2: **Codimensional objects.** (a) Cloth modeled as a triangle mesh. (b) Yarn modeled as a sequence of cylinders. (c) A cloth-yarn joint. The volume of the shaded region is used to compute vertex weights. Each triangle is uniformly divided according to its barycenter and edge bisectors.

Notation. The subscript e indicates that a field is discretized over mesh elements, triangle faces and yarn segments, for example, $\mathbf{u}_{s,e}$ represents solid velocity defined on elements. By contrast, the subscript v indicates that a field is discretized over vertices. Individual time steps are indicated by a superscript t (e.g., p_v^t for pressure stored on vertices at time step t), and h always denotes the time step size.

Codimensional Solve

To solve (3.18) on irregular meshes, we first define the necessary mesh-based discrete differential operators [40].

Each mesh element (cloth triangle or rod segment) is associated with a time-invariant finite volume (for all the elements, their volumes are put together into a vector \mathbf{V}_e) in the physical space. For a cloth triangle, this is computed from its undeformed area and fabric thickness. For a rod segment, this is computed from its undeformed length and yarn thickness.

Each vertex is also associated with a time-invariant finite volume (put together into a vector \mathbf{V}_v for the volumes of all vertices) (Figure 4.2), computed in a typical barycentric style: incident cloth triangles contribute a third of their volume to each vertex, and incident rod segments contribute half of their volume to each vertex.

The liquid volume discretized on vertices is given by the vector $\mathbf{V}_{f,v} = (\mathbf{1}_v - \phi_v) * \mathbf{S}_{r,v} * \mathbf{V}_v$, where $*$ is the element-wise product, ϕ_v and $\mathbf{S}_{r,v}$ are the solid volume fraction and liquid saturation per vertex (recall §3.1), and $\mathbf{1} \in \mathbb{R}^{N_v}$ is the vector filled with one whose length is the total number of vertices. Similarly, $\mathbf{V}_{f,e}$ is a vector describing the liquid volume per element. With these expressions, we can discretize the convection-diffusion equation (3.18) for the liquid fraction using implicit integration as

$$\mathbf{V}_{f,v}^{t+1} = \mathbf{V}_{f,v}^t - h \mathbf{G}_{ev}^T \left[\hat{\mathbf{C}}_{\alpha,e}^{-1} \left([\mathbf{1} - \phi_e] [\mathbf{V}]_{f,e}^{t+1} (\mathbf{G}_{ev} [\tilde{\mathbf{p}}_\alpha - \mathbf{p}_f]_v^{t+1} + \rho_f \mathbf{g}_e) \right) + [\mathbf{V}]_{f,e}^{t+1} \mathbf{u}_{s,e}^{t+1} \right], \quad (4.1)$$

where the notation $[\cdot]$ denotes the operator that converts a vector into a diagonal matrix.

On triangle meshes, we use the standard gradient and divergence operators described in detail by Botsch et al. [40]. The gradient operator $\mathbf{G}_{\text{ev}} \in \mathbb{R}^{3N_e \times N_v}$ maps the vector form of a quantity defined on vertices to its gradient on elements, and its adjoint, the divergence operator $\mathbf{G}_{\text{ev}}^T \in \mathbb{R}^{N_v \times 3N_e}$, maps a vector quantity on elements to its divergence on vertices. N_v and N_e indicate the total number of vertices and elements, respectively. Construction of \mathbf{G}_{ev} relies on the same weight contributions used to compute \mathbf{V}_v .

This is a system of equations with respect to $\mathbf{V}_{f,v}^{t+1}$ that is nonlinear, and, in the general case, very difficult to solve [178]. In our case, the Reynolds number of liquids flowing through cloth and yarn is low, so $\mathbf{V}_{f,v}^{t+1}$ remains fairly close to $\mathbf{V}_{f,v}^t$ over a time step. Thus, we solve (4.1) using fixed-point iterations [47]: in each iteration, we update $\mathbf{V}_{f,e}^{t+1}$ by interpolating $\mathbf{V}_{f,v}^{t+1}$ from the previous iteration, and then update $\mathbf{V}_{f,v}^{t+1}$ using (4.1). In practice, this method converged within four iterations for the scenes we tested.

Liquid Capturing and Dripping

When cloth and yarn come in contact with water they begin to absorb it, and become wet. On the other hand, if cloth or yarn becomes locally oversaturated, water starts to drip off. Therefore, our codimensional simulation must also exchange liquid with the background fluid grid.

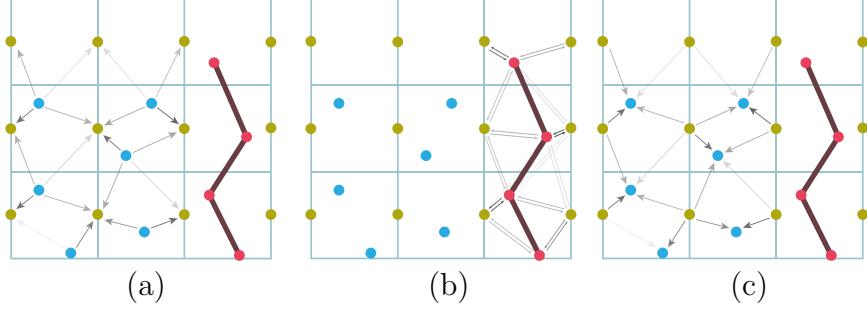


Figure 4.3: **Liquid capturing.** • liquid particles; • solid vertices; • grid faces. Opacity indicates kernel weight. For legibility, only the faces in the x -axis are shown; similar operations are done for the y - and z -axis. (a) Liquid volume from particles is distributed onto grid faces. (b) Solid vertices absorb volume from grid faces whose volume is reduced correspondingly. (c) The retained volume is stored back onto particles.

Water absorption is performed with the following steps (see figure 4.3):

1. For the i -th grid face, we update its liquid volume by summing contributions from liquid particles in the grid, $V_{f,g,i}^t = \sum_j V_{f,p,j}^t w_{i,j}$, where $V_{f,p,j}^t$ is the volume of the j -th liquid particle, $w_{i,j}$ is the kernel function between the particle j and grid face i , as defined by Jiang et al. [122].
2. The j -th solid vertex (cloth or yarn) captures liquid by taking liquid volume contributions from the background grid: $V_{f,v,j}^t = \sum_i V_{f,g,i}^t w_{i,j}$, where $w_{i,j}$ is the kernel function between grid face i and solid vertex j .
3. For the i -th grid face, we compute the amount of liquid removed by the cloth or yarn: $V_{f,g,i}^- = V_{f,g,i}^t \sum_j w_{i,j}$, which totals the i -th face's liquid contribution to all solid vertices.
4. Lastly, we update the j -th liquid particle's volume by computing the weighted sum of the updated liquid volume on grid faces: $V_{f,p,j}^{t+1} = (\sum_i (V_{f,g,i}^t - V_{f,g,i}^-) w_{i,j}) / (\sum_i \sum_k w_{i,k} w_{i,k})$.

The updated volume of liquid particles will be used in the next time step of the grid simulation (see §4.2).

Liquid drips off of cloth or yarn when the i -th solid vertex is oversaturated. This is indicated by the condition (for the i -th vertex) $V_{f,v,i} > V_{v,i}(1 - \phi_{v,i})$. If it is satisfied, we inject liquid particles back into the grid. Each generated liquid particle has a fixed volume V_r^1 . The number of liquid particles that the i -th (oversaturated) vertex can generate is $N_{dp,i} \equiv \left\lfloor \frac{V_{f,v,i} - V_{v,i}(1 - \phi_{v,i})}{V_r} \right\rfloor$. We then uniformly sample $N_{dp,i}$ positions on the elements (triangles and yarn segments) incident to the i -th vertex, placing a liquid particle at each. Since liquid flow on the cloth and yarn is assumed to be quasi-static, the velocity of a new particle is set to the solid velocity at its position. Afterward, the liquid volume at the i -th solid vertex (that is oversaturated) is updated to $V_{f,v,i} - N_{dp,i}V_r$.

4.2 Grid Simulation

We solve the dynamic equations of wet cloth and yarn on the MAC grid. First, we discretize the incompressibility condition (3.4) and obtain

$$h\mathbf{G}_{cg}^T[\phi_s]\mathbf{u}_s^{t+1} + h\mathbf{G}_{cg}^T[1 - \phi_s]\mathbf{u}_f^{t+1} = 0. \quad (4.2)$$

¹We follow a standard rule of thumb to set V_r . As discussed by Um et al. [235], a common practice is to have eight particles in a grid cell, each having sufficient volume to cover half of the cell size in each dimension. This means that $V_r \equiv \frac{\pi\sqrt{3}\Delta x^3}{16}$ where Δx is the size of a grid cell.

Here we use \mathbf{G}_{cg} and \mathbf{G}_{cg}^T to denote the (finite-volume) gradient and divergence operators, analogous to those in (4.1), but on the grid.

When discretizing the momentum equations (3.1a) and (3.1b), we ignore the advection terms (i.e., the $\mathbf{u} \cdot \nabla$ term in the material derivative), because we advect the liquid and solid materials in a separate substep via background particles with the APIC method (to be discussed in §4.2). Moreover, since cloth and yarn can often be highly stiff, they demand an implicit discretization of (3.1a). Otherwise, very small time step sizes are needed, which would dramatically slow down the simulation. Thus, discretizing the momentum equations (3.1a) and (3.1b) yields

$$[(\mathbf{M}_s + h[\mathbf{C}]\mathbf{V}_c) + h^2\mathbf{H}_s] \mathbf{u}_s^{t+1} - h[\mathbf{C}]\mathbf{V}_c \mathbf{u}_f^{t+1} + h\mathbf{V}_s \mathbf{G}_{\text{cg}} \mathbf{p}^{t+1} = h\mathbf{f}_s + \mathbf{M}_s \mathbf{u}_s^t, \quad (4.3)$$

and

$$(\mathbf{M}_f + h[\mathbf{C}]\mathbf{V}_c) \mathbf{u}_f^{t+1} - h[\mathbf{C}]\mathbf{V}_c \mathbf{u}_s^{t+1} + h\mathbf{V}_f \mathbf{G}_{\text{cg}} \mathbf{p}^{t+1} = h\mathbf{f}_f + \mathbf{M}_f \mathbf{u}_f^t, \quad (4.4)$$

where \mathbf{M}_s , \mathbf{M}_f , \mathbf{V}_c , \mathbf{V}_s , and \mathbf{V}_f are all diagonal matrices. We obtain \mathbf{M}_s by distributing the mass of cloth and yarn vertices to the face centers of grid cells (see §4.2). We obtain \mathbf{V}_c similarly by distributing vertex volumes V_v to the face centers of grid cells. Since a vertex volume is occupied by solid, liquid, and air, $\mathbf{V}_s = [\phi_s]\mathbf{V}_c$ is the solid portion of \mathbf{V}_c , while $\mathbf{V}_f = [1 - \phi_s][\mathbf{S}_r]\mathbf{V}_c$ is the liquid portion of \mathbf{V}_c . \mathbf{M}_f is the mass matrix of liquid. $[\mathbf{C}]$ is a tridiagonal matrix whose 3×3 diagonal subblocks are the drag tensors (as defined in (3.16)) evaluated at grid face centers. Lastly, \mathbf{f}_s includes forces on solid vertices and are distributed to the grid's face centers, \mathbf{f}_f are forces

applied on the liquid, and \mathbf{H}_s is the Jacobian matrix of solid force \mathbf{f}_s with respect to the solid vertex positions. Their specific forms will be given in §4.2.

Assembling the discrete equations (4.2-4.4), we obtain a system of linear equations with respect to the unknowns \mathbf{u}_s^{t+1} , \mathbf{u}_f^{t+1} , and \mathbf{p}^{t+1} . However, solving this linear system is rather challenging since it is large and unsymmetric. It couples \mathbf{u}_s^{t+1} , \mathbf{u}_f^{t+1} , and \mathbf{p}^{t+1} together, and its size is about seven times the number of grid faces, which makes direct solvers impractical. To make matters worse, the linear system can be stiff due to the large stretching stiffness of cloth and yarn or large pressure gradient applied, thus requiring many iterations for iterative solvers to converge to the solution. We initially attempted to use BiCGSTAB, but it successfully converged only under impractically small time step sizes (Courant number less than 10^{-5}). We therefore propose an efficient alternative solution strategy.

Solver overview. We begin by summarizing the three main steps of our solver. First, by discretizing (3.1a) *explicitly*, we reduce the linear system to a smaller one involving \mathbf{p}^{t+1} alone. After obtaining \mathbf{p}^{t+1} , we return to an implicit discretization of (3.1a), and solve another system of linear equations with respect to \mathbf{u}_s^{t+1} alone. Lastly, we construct a linear system to solve for \mathbf{u}_f^{t+1} . This final system will be diagonal and hence trivially inverted. In this process, \mathbf{u}_s^{t+1} for the solid porous materials is obtained with an implicit solve, ensuring that the time step size is not restricted by explicit integration. We now elaborate on each of these three steps.

Pressure Solve

We start with the explicit discretization of (3.1a), for which the $h^2\mathbf{H}$ term in front of \mathbf{u}_s^{t+1} in (4.3) vanishes. Then, the linear system consisting of (4.2), (4.4), and the explicit counterpart of (3.1a) can be written as

$$\begin{bmatrix} \mathbf{D}_s & 0 & h(\mathbf{V}_s + \mathbf{V}_f \mathbf{P}) \mathbf{G}_{cg} \\ 0 & \mathbf{D}_f & h(\mathbf{V}_f + \mathbf{V}_s \mathbf{Q}) \mathbf{G}_{cg} \\ h\mathbf{G}_{cg}^T[\phi] & h\mathbf{G}_{cg}^T[\mathbf{1} - \phi] & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_s^{t+1} \\ \mathbf{u}_f^{t+1} \\ \mathbf{p}^{t+1} \end{bmatrix} = \begin{bmatrix} h\mathbf{f}_s + \mathbf{M}_s \mathbf{u}_s^t + \mathbf{P}(\mathbf{M}_f \mathbf{u}_f^t + h\mathbf{f}_f) \\ h\mathbf{f}_f + \mathbf{M}_f \mathbf{u}_f^t + \mathbf{Q}(\mathbf{M}_s \mathbf{u}_s^t + h\mathbf{f}_s) \\ 0 \end{bmatrix}, \quad (4.5)$$

where the matrices \mathbf{P} , \mathbf{Q} , \mathbf{D}_s , and \mathbf{D}_f are

$$\begin{aligned} \mathbf{P} &\equiv (\mathbf{M}_f + h[\mathbf{C}] \mathbf{V}_c)^{-1} h[\mathbf{C}] \mathbf{V}_c, \\ \mathbf{Q} &\equiv (\mathbf{M}_s + h[\mathbf{C}] \mathbf{V}_c)^{-1} h[\mathbf{C}] \mathbf{V}_c, \\ \mathbf{D}_s &\equiv \mathbf{M}_s + \mathbf{P} \mathbf{M}_f, \text{ and } \mathbf{D}_f \equiv \mathbf{M}_f + \mathbf{Q} \mathbf{M}_s. \end{aligned} \quad (4.6)$$

Derivation of Equation 4.5. Explicit integration of the solid and liquid dynamics yields the following equations:

$$(\mathbf{M}_s + h[\mathbf{C}] \mathbf{V}_c) \mathbf{u}_s^{t+1} - h[\mathbf{C}] \mathbf{V}_c \mathbf{u}_f^{t+1} + h\mathbf{V}_s \mathbf{G}_{cg} \mathbf{p}^{t+1} = h\mathbf{f}_s + \mathbf{M}_s \mathbf{u}_s^t, \quad (4.7)$$

and

$$(\mathbf{M}_f + h[\mathbf{C}] \mathbf{V}_c) \mathbf{u}_f^{t+1} - h[\mathbf{C}] \mathbf{V}_c \mathbf{u}_s^{t+1} + h\mathbf{V}_f \mathbf{G}_{cg} \mathbf{p}^{t+1} = h\mathbf{f}_f + \mathbf{M}_f \mathbf{u}_f^t. \quad (4.8)$$

To simplify the derivation, we first add these two equations together, which produces the equation of momentum conservation of the mixture:

$$\mathbf{M}_s \mathbf{u}_s^{t+1} + \mathbf{M}_f \mathbf{u}_f^{t+1} + h(\mathbf{V}_s + \mathbf{V}_f) \mathbf{G}_{cg} \mathbf{p}^{t+1} = h \mathbf{f}_s + h \mathbf{f}_f + \mathbf{M}_s \mathbf{u}_s^t + \mathbf{M}_f \mathbf{u}_f^t. \quad (4.9)$$

Substituting \mathbf{u}_s^{t+1} in equation (4.8) with equation (4.9), and combining the terms yields

$$\begin{aligned} & ((\mathbf{I} + h[\mathbf{C}] \mathbf{V}_c \mathbf{M}_s^{-1}) \mathbf{M}_f + h[\mathbf{C}] \mathbf{V}_c) \mathbf{u}_f^{t+1} + h(h[\mathbf{C}] \mathbf{V}_c \mathbf{M}_s^{-1} (\mathbf{V}_s + \mathbf{V}_f) + \mathbf{V}_f) \mathbf{G}_{cg} \mathbf{p}^{t+1} \\ & = (\mathbf{I} + h[\mathbf{C}] \mathbf{V}_c \mathbf{M}_s^{-1})(\mathbf{M}_f \mathbf{u}_f^t + h \mathbf{f}_f) + h[\mathbf{C}] \mathbf{V}_c (\mathbf{u}_s^t + h \mathbf{M}_s^{-1} \mathbf{f}_s). \end{aligned} \quad (4.10)$$

From this point on \mathbf{u}_f^{t+1} has been decoupled from \mathbf{u}_s^{t+1} . Multiplying both sides of equation (4.10) with $(\mathbf{M}_s + h[\mathbf{C}] \mathbf{V}_c)^{-1} \mathbf{M}_s$ yields

$$(\mathbf{M}_f + \mathbf{Q} \mathbf{M}_s) \mathbf{u}_f^{t+1} + h(\mathbf{V}_f + \mathbf{V}_s \mathbf{Q}) \mathbf{G}_{cg} \mathbf{p}^{t+1} = \mathbf{M}_f \mathbf{u}_f^t + h \mathbf{f}_f + \mathbf{Q}(\mathbf{M}_s \mathbf{u}_s^t + h \mathbf{f}_s), \quad (4.11)$$

where

$$\mathbf{Q} \equiv (\mathbf{M}_s + h[\mathbf{C}] \mathbf{V}_c)^{-1} h[\mathbf{C}] \mathbf{V}_c. \quad (4.12)$$

Similarly, substituting \mathbf{u}_f^{t+1} in equation (4.7) with equation (4.9), and multiplying both sides with $(\mathbf{M}_f + h[\mathbf{C}] \mathbf{V}_c)^{-1} \mathbf{M}_f$ yields

$$(\mathbf{M}_s + \mathbf{P} \mathbf{M}_f) \mathbf{u}_s^{t+1} + h(\mathbf{V}_s + \mathbf{V}_f \mathbf{P}) \mathbf{G}_{cg} \mathbf{p}^{t+1} = \mathbf{M}_s \mathbf{u}_s^t + h \mathbf{f}_s + \mathbf{P}(\mathbf{M}_f \mathbf{u}_f^t + h \mathbf{f}_f), \quad (4.13)$$

where

$$\mathbf{P} \equiv (\mathbf{M}_f + h[\mathbf{C}]\mathbf{V}_c)^{-1}h[\mathbf{C}]\mathbf{V}_c. \quad (4.14)$$

Combining equations (4.11) and (4.13) with equation (4.2), and introducing notations

$\mathbf{D}_s \equiv \mathbf{M}_s + \mathbf{P}\mathbf{M}_f$ and $\mathbf{D}_f \equiv \mathbf{M}_f + \mathbf{Q}\mathbf{M}_s$, we have the form given by equation (4.5). ■

Recall that $[\mathbf{C}]$ is a tridiagonal matrix. Let $\tilde{\mathbf{C}}$ denote one of its 3×3 diagonal subblocks defined by (3.16). Its off-diagonal element $\tilde{\mathbf{C}}_{ij}$ depicts the drag force along the axis i induced by the liquid-solid velocity difference along a different axis j . The cross-axis terms in the drag tensor are responsible for rotational and shear effects, which can be assumed negligible under moderate Reynolds number [17]. We therefore lump the off-diagonal elements of $[\mathbf{C}]$ into its diagonal elements, turning $[\mathbf{C}]$ into a fully diagonal matrix. This approximation can also be justified from a numerical point of view. When the drag force is large (e.g., for fast liquid flows wherein the Reynolds number is high), $[\mathbf{C}]$, without lumping, dominates over \mathbf{M}_f and \mathbf{M}_s . Thus, \mathbf{P} and \mathbf{Q} in (4.6) are both nearly identity matrices, and \mathbf{D}_s and \mathbf{D}_f are nearly diagonal; lumping simply $[\mathbf{C}]$ approximates \mathbf{D}_s and \mathbf{D}_f as fully diagonal. On the other hand, when the drag force is very small, $[\mathbf{C}]$ approaches zero. Then, \mathbf{P} and \mathbf{Q} are close to zero, and \mathbf{D}_s and \mathbf{D}_f remain almost diagonal, so lumping $[\mathbf{C}]$ to be diagonal matrix is again a reasonable approximation.

With \mathbf{D}_s and \mathbf{D}_f being diagonal, the first two equations of (4.5) allow us to easily express \mathbf{u}_s^{t+1} and \mathbf{u}_f^{t+1} with respect to \mathbf{p}^{t+1} . After substituting this expression into

the third equation of (4.5), we obtain a system of equations with respect to \mathbf{p}^{t+1} ,

$$\begin{aligned} h\mathbf{G}_{cg}^T & \left[[\mathbf{1} - \phi_s]\mathbf{D}_f^{-1}(\mathbf{V}_f + \mathbf{V}_s\mathbf{Q}) + [\phi_s]\mathbf{D}_s^{-1}(\mathbf{V}_s + \mathbf{V}_f\mathbf{P}) \right] \mathbf{G}_{cg}\mathbf{p}^{t+1} \\ & = \mathbf{G}_{cg}^T \left[\Phi_{fs}(\mathbf{u}_f^t + h\mathbf{M}_f^{-1}\mathbf{f}_f) + \Phi_{sf}(\mathbf{u}_s^t + h\mathbf{M}_s^{-1}\mathbf{f}_s) \right], \quad (4.15) \end{aligned}$$

where the matrices Φ_{fs} and Φ_{sf} have the following forms,

$$\begin{aligned} \Phi_{fs} & \equiv [\mathbf{1} - \phi_s]\mathbf{D}_f^{-1}\mathbf{M}_f + [\phi_s]\mathbf{D}_s^{-1}\mathbf{M}_f\mathbf{P}, \\ \Phi_{sf} & \equiv [\phi_s]\mathbf{D}_s^{-1}\mathbf{M}_s + [\mathbf{1} - \phi_s]\mathbf{D}_f^{-1}\mathbf{M}_s\mathbf{Q}. \end{aligned} \quad (4.16)$$

Equation (4.15) is analogous to the pressure projection step in standard fluid simulation, but for solid-liquid mixtures.

Solid Velocity Solve

After obtaining \mathbf{p}^{t+1} , we are ready to solve for \mathbf{u}_s^{t+1} . Because of the high stretch stiffness of cloth and yarn, we adopt the implicit discretization in (4.3). Then, the $h^2\mathbf{H}_s$ term multiplying \mathbf{u}_s^{t+1} in (4.3) will appear in the first row of equations in (4.5): the first subblock \mathbf{D}_s becomes $\mathbf{D}_s + h^2\mathbf{H}_s$. We obtain a system of equations with respect to \mathbf{u}_s^{t+1} :

$$(\mathbf{D}_s + h^2\mathbf{H}_s)\mathbf{u}_s^{t+1} = -h(\mathbf{V}_s + \mathbf{V}_f\mathbf{P})\mathbf{G}_{cg}\mathbf{p}^{t+1} + h\mathbf{f}_s + \mathbf{M}_s\mathbf{u}_s^t + \mathbf{P}(\mathbf{M}_f\mathbf{u}_f^t + h\mathbf{f}_f), \quad (4.17)$$

where the pressure \mathbf{p}^{t+1} is already known at this point. On the left-hand side, $\mathbf{D}_s + h^2\mathbf{H}_s$ is a symmetric positive definite matrix. We then solve this system with a

matrix-free conjugate-residual solver preconditioned with \mathbf{D}_s^{-1} .

Fluid Velocity Solve

Lastly, we substitute \mathbf{p}^{t+1} and \mathbf{u}_s^{t+1} into the second line of (4.5) to solve for \mathbf{u}_f^{t+1} . As the matrix \mathbf{D}_s multiplying \mathbf{u}_f^{t+1} is diagonal, this equation is trivially solved, where

$$\mathbf{u}_f^{t+1} = \mathbf{D}_f^{-1} \left[-h(\mathbf{V}_f + \mathbf{V}_s \mathbf{Q}) \mathbf{G}_{cg} \mathbf{p}^{t+1} + h \mathbf{f}_f + \mathbf{M}_f \mathbf{u}_f^t + \mathbf{Q} (\mathbf{M}_s \mathbf{u}_s^t + h \mathbf{f}_s) \right]. \quad (4.18)$$

Remark: While we require implicit integration for stability of the fabric, we observed that within a single time step, the explicit and implicit methods produce similar fabric motion, especially when the time step is not too large. Therefore, we choose a semi-implicit approach in exchange for computational performance: by explicitly integrating velocity for the pressure solve, and implicitly integrating the fabric velocity after the pressure solve, we reduce the large, unsymmetric linear system to three smaller symmetric and positive definite systems which are much easier to solve.

When solving for the liquid velocity, we may either insert the pressure and solid velocity into (4.4) or only insert the pressure into the second line of (4.5). We tested both options. For the first, we observed an average of $\sim 3\%$ difference in the divergence in all of our examples using the time step given in Table 5.1, while for the latter, we observed an average difference in the divergence an order of magnitude smaller. The intuition is that in the first an additional Jacobian matrix $h^2 \mathbf{H}_s$ is added to the divisor when solving for the matrix \mathbf{Q} in (4.6), which further increases the mismatch

between the solved pressure and the divergence of liquid velocity. Hence we choose the latter solution.

We also found that the difference has approximately linear growth with respect to time step and the viscosity of the liquid. In practice, for liquid up to a moderate viscosity coefficient (e.g., olive oil), we did not observe any visual artifacts due to the difference. Nevertheless for high viscosity liquid (e.g., honey) there is indeed some instability due to the mismatch between the explicitly integrated solid velocity used by the drag force, and the actual implicitly integrated solid velocity. A method with a strict guarantee of incompressibility and which can handle high viscosity liquid requires future investigation.

Implementation Details

In the aforementioned three steps, we need to construct \mathbf{M}_s , \mathbf{V}_c , \mathbf{V}_s , \mathbf{V}_f , and \mathbf{M}_f for the face centers of the MAC grid. For \mathbf{M}_s , \mathbf{V}_c , \mathbf{V}_s , and \mathbf{V}_f , we first compute the corresponding quantities on cloth and yarn vertices. For example, the liquid volume $V_{f,i}$ at a vertex i is computed with $V_{v,i}(1 - \phi_{v,i})S_{r,i}$, where $\phi_{v,i}$ is the solid volume fraction at vertex i , and $S_{r,i}$ is its saturation. We then distribute the quantities from the vertices to face centers, using the kernel functions defined in the APIC method [122]. Similarly, for constructing \mathbf{M}_f , we compute the liquid mass $\rho_f V_{f,p,i}$ on the i -th liquid particle, and distribute it to its nearby face centers.

Force Computation. In the discretized equations (4.3) and (4.4), the forces are computed as

$$\mathbf{f}_s = \mathbf{V}_s \mathbf{G}_{cg} : ((\mathbf{1} - \mathbf{S}_r) * \mathbf{P}_c) + \mathbf{f}_F \text{ and } \mathbf{f}_f = \mathbf{V}_f \mathbf{G}_{cg} : ((\mathbf{1} - \mathbf{S}_r) * \mathbf{P}_c) + \mathbf{M}_f \mathbf{g}, \quad (4.19)$$

where \mathbf{P}_c , as defined in (3.11), is the suction tensor computed using the quantities stored on grid cell centers, and \mathbf{S}_r is the solid vertex saturation distributed on the grid cell centers. We use $\mathbf{A} : \mathbf{B}$ to denote the tensor product between two tensors where one has dimensions $N_g \times N_c \times 3$ (we reshape \mathbf{G}_{cg} into this form) and another has dimensions $N_c \times 3 \times 3$. Also we use $*$ to denote the element-wise product that modulates the suction tensor on each cell center with the saturation on each cell center. Then the first terms of both \mathbf{f}_s and \mathbf{f}_f are initially evaluated on grid cell centers and then computed on face centers through the tensor product with the gradient operator; $\mathbf{M}_f \mathbf{g}$ is the liquid's gravity force evaluated on the grid as well. On the other hand, \mathbf{f}_F are forces firstly evaluated on solid vertices, including the internal elastic forces, collision and frictional forces, and gravity forces, and then distributed to grid faces using the APIC method. We adopt existing models to compute these forces. In particular, the cloth internal forces are computed using the discrete shell model [92], the internal forces of yarn follow the discrete elastic rod model [28], and the collision and frictional forces are computed following Jiang et al. [121].

We highlight one detail related to the distribution of yarn torques using APIC. The discrete viscous thread model uses a scalar ς_j at the j -th yarn segment to indicate the

strength of torque with respect to the tangential direction \mathbf{t}_j of the j -th yarn segment.

In order to distribute the j -th torque to a background grid face i , we convert the scalar into a vector $\varsigma_j \mathbf{t}_j \times \nabla w_{i,j}$ (see derivation below) before adding it to the grid face i . Here $w_{i,j}$ is the kernel function between grid face i and yarn segment j .

Distribution of Torque from Vertex to Grid. For simplicity of

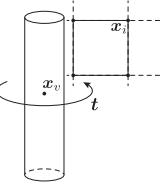
presentation, consider a yarn segment j along the Y-direction (see the adjacent figure). If there is a torque $\boldsymbol{\varsigma} \equiv \varsigma_v \mathbf{t}_v$ applied with respect to the centerline of the yarn to twist the yarn, then, when we distribute the torque $\boldsymbol{\varsigma}$ to a grid node i , this node receives a twisting force produced by the torque weighted by the (scalar) kernel function,

$$w_{i,j} \mathbf{D}_j^{-1} [\mathbf{t}_j \times (\mathbf{x}_i - \mathbf{x}_j)] = [\mathbf{t}_j \times w_{i,j} \mathbf{D}_j^{-1} (\mathbf{x}_i - \mathbf{x}_j)], \quad (4.20)$$

where \mathbf{x}_i is the position of grid face i , \mathbf{x}_j is the position of yarn segment j , $w_{i,j}$ is the kernel function for the contribution of yarn segment j at the grid node i , and \mathbf{D}_j is analogous to an inertia tensor defined as

$$\mathbf{D}_j = \sum_i w_{i,j} (\mathbf{x}_i - \mathbf{x}_j)^* (\mathbf{x}_i - \mathbf{x}_j)^{*T}$$

where \mathbf{x}^* is the cross-product matrix associated with vector \mathbf{x} . When $w_{i,j}$ is a trilinear function, the relationship $w_{i,j} \mathbf{D}_j^{-1} (\mathbf{x}_i - \mathbf{x}_j) = \nabla w_{i,j}$ holds, as noted in [122]. Then,



the right-hand side of (4.20) can be simplified into

$$\mathbf{t}_j \times \nabla w_{i,j}.$$

This is the formula that we use to distribute the torque of yarn vertices on the grid. ■

Jacobian matrix computation. The matrix \mathbf{H}_s in (4.3) and (4.4) is the Jacobian of \mathbf{f}_s distributed on the grid. This matrix emerges when we integrate the force terms implicitly. Because the stiffest force terms in \mathbf{f}_s are the internal elastic forces, we compute their contributions to the Jacobian matrix, and ignore the contributions from collision and frictional forces, instead integrating them explicitly. We compute the Jacobian matrix $\mathbf{H}_{s,j}$ of the elastic forces at the j -th solid vertex, and add its contribution to the grid face i using

$$\mathbf{H}_{i,j} = \mathbf{W}_{i \rightarrow j}^T \mathbf{H}_{s,j} \mathbf{W}_{i \rightarrow j}, \quad (4.21)$$

where $\mathbf{W}_{i \rightarrow j}$ is the weight that distributes a force vector from grid face i to solid vertex j in the augmented MPM method, as defined by Stomakhin et al. [226].

Cohesion force between pieces of cloth. When two pieces of wet cloth are in close proximity, the surface tension of the liquid between introduces cohesion forces. Accurately computing surface tension requires the reconstruction of detailed liquid surface shapes between the cloth pieces, which in turn demands an extremely high

grid resolution. Even for a moderate size piece of cloth, computing this effect through brute force is intractable. In this work, we use a simple model to approximate cohesion forces at cloth vertices, and use the APIC method again to distribute the forces to grid nodes. We describe our model below, while leaving a full investigation of this surface tension-induced effect to future research. We will discuss the cohesion between yarns in **Part II**.

We assume that the surface tension appears when the distance between two pieces of wet cloth is less than a small constant. Also the water-air surface is assumed perpendicular to the cloth surface. Under these simplifications, the surface tension energy becomes $E_T = \int_{\Omega} \gamma d\Omega$, where Ω is the boundary of the wet cloth regions that are connected by water, d is the distance between two cloth pieces, and γ is the surface tension coefficient.

Then, the surface tension force generated at each small segment of the water boundary Ω is

$$df_T = \gamma dl. \quad (4.22)$$

In the discrete setting, we need to compute the surface tension force at every vertex. We perform the following steps:

1. For each triangle element i , find the closest non-neighboring element j within a distance threshold in the cone of θ degrees around the normal direction.
2. For each pair (i, j) of elements identified in step (1), connect a line segment s

between the two centers of the elements. We iterate through all background cells that s passes by, and compute their average liquid volume fraction $\bar{\phi}_c$. The liquid volume fraction in each cell is computed using the method of Batty et al. [22].

3. If the averaged threshold $\bar{\phi}_c$ is within the range $[0.5 - \varphi, 0.5 + \varphi]$ where φ is a user-controlled threshold, then we add a surface tension force $\gamma\sqrt{\frac{S_i+S_j}{2}}$ to both elements along their respective normal directions. The square root term is to approximate dl in (4.22) using the effective length of the average triangle area. The force is then distributed evenly to the triangle element vertices.

Chapter 5

Simulated Results of Wet Fabrics

We divide our results into two classes: *i*) a group of didactic cases designed to validate individual components of our framework, and *ii*) a set of more general scenarios of liquid interaction with cloth and yarn that demonstrate the diversity of practical effects that can be achieved by our system. Details of our surface reconstruction, the rendering method, and a summary and discussion of the physics parameters used throughout this part can be found in the end of this section.

5.1 Didactic Examples

Ring Test. A classic experiment in the textiles industry is a *ring test* [180], where a controlled volume of liquid is released onto the center of a piece of cloth. We compare our simulation with a physical experiment in Figure 3.2. When the liquid touches the cloth, wicking can be observed in both the physical experiment and our simulation. Although our numerical experiment does not quite reproduce the noisy details of the real-world surface, the liquid in both the experiment and our simulation yielded visually and qualitatively consistent wicking behaviors.

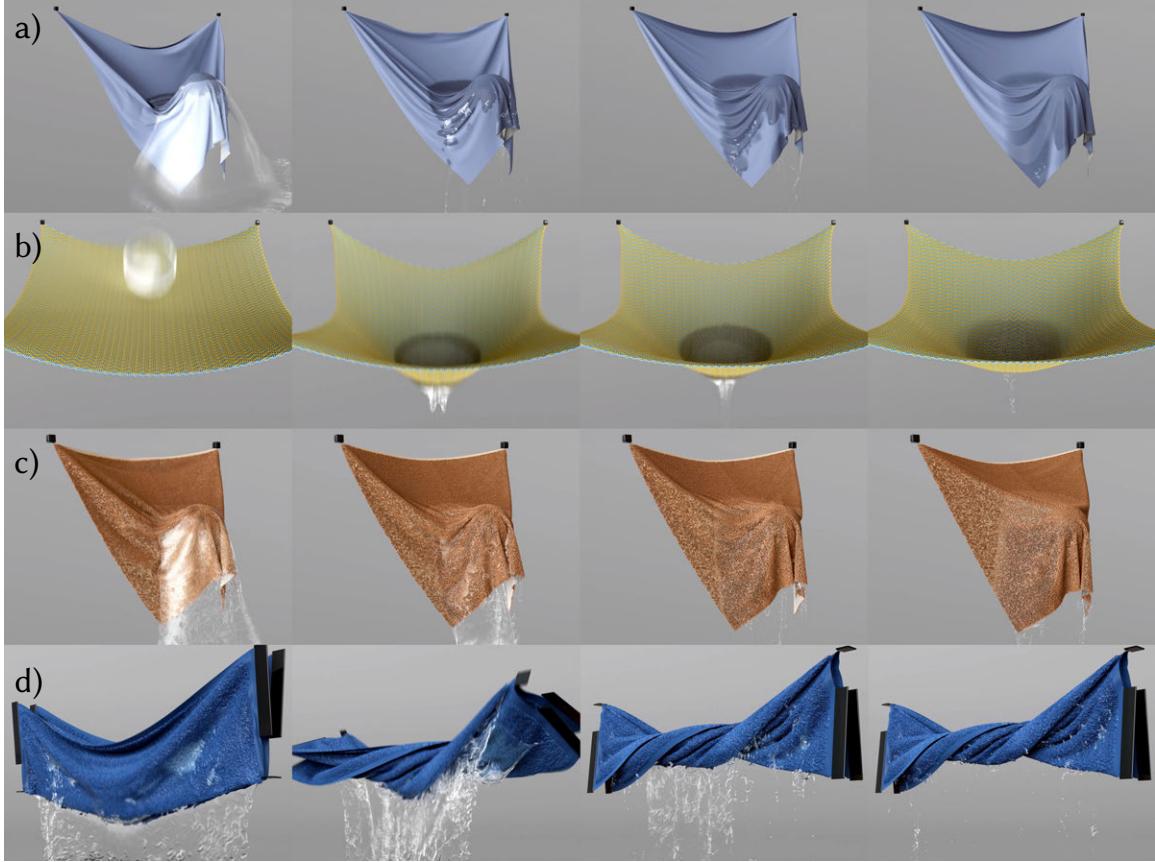


Figure 5.1: **Large examples of wet fabrics.** (a) A ball of water splashes on a mesh-based cloth. (b) A ball of water splashes on a yarn-based fabric. (c) A ball of water splashes on a piece of terrycloth. (d) A towel is pulled rapidly out of water and wrung out.

Drag Forces. The nonlinearity of drag forces has a significant impact on the look of real liquid-cloth interactions. Figure 3.5 presents a comparison between nonlinear and linear drag force models. The most obviously distinct visual phenomenon that can be seen in the nonlinear case is the formation of “kinks” around regions where the relative velocity between the cloth and liquid is large; the cloth has dragged the liquid along with it to create this characteristic shape. This phenomenon cannot be readily observed with the linear drag force. The same figure illustrates this “kink” effect in a real experiment in which large relative velocities are induced by pulling a

cloth rapidly out of liquid.

Buoyancy Forces. In Figure 3.3, we highlight the importance of the pressure gradient, using fabrics with differing mass densities. The leftmost has density $\rho_s = 0.25\text{g}/\text{cm}^3$, which is lower than water's; the middle fabric has the same density $\rho_s = 1.0\text{g}/\text{cm}^3$ as water (i.e., neutrally buoyant); and the rightmost has density $\rho_s = 4.0\text{g}/\text{cm}^3$, which is higher than water's. With the correct pressure gradient applied to the fabrics, as expected, the left one rises to the water surface; the middle one drifts along with the fluid water; and the right one sinks quickly to the bottom. By contrast, if the pressure gradient is neglected, the fabrics sink and come to rest at the bottom, regardless of their mass densities.

Various Parameters. Different fabric and liquid parameters can also drastically alter the look of cloth-liquid interaction [63]: permeability decreases quadratically with fiber diameter and nonlinearly with volume fraction through (3.13); while the pore pressure increases with volume fraction and decreases with contact angle through (3.9). In Figure 5.2 we compare simulation with different sets of parameters varied from reference (Figure 5.2a). All the simulations are done with the same initial geometries, and the screenshots are captured at 4.0 seconds. The expected effects are recovered in our numerical experiments. In Figure 5.2b and 5.2c, as we adjust the fiber diameter d , we simultaneously hold the rest solid fraction ϕ_0 constant by appropriately adjusting the thread count n_t and capillary radius r_b to compensate; the cloth with smaller d is less readily penetrated by the liquid, the liquid attaches

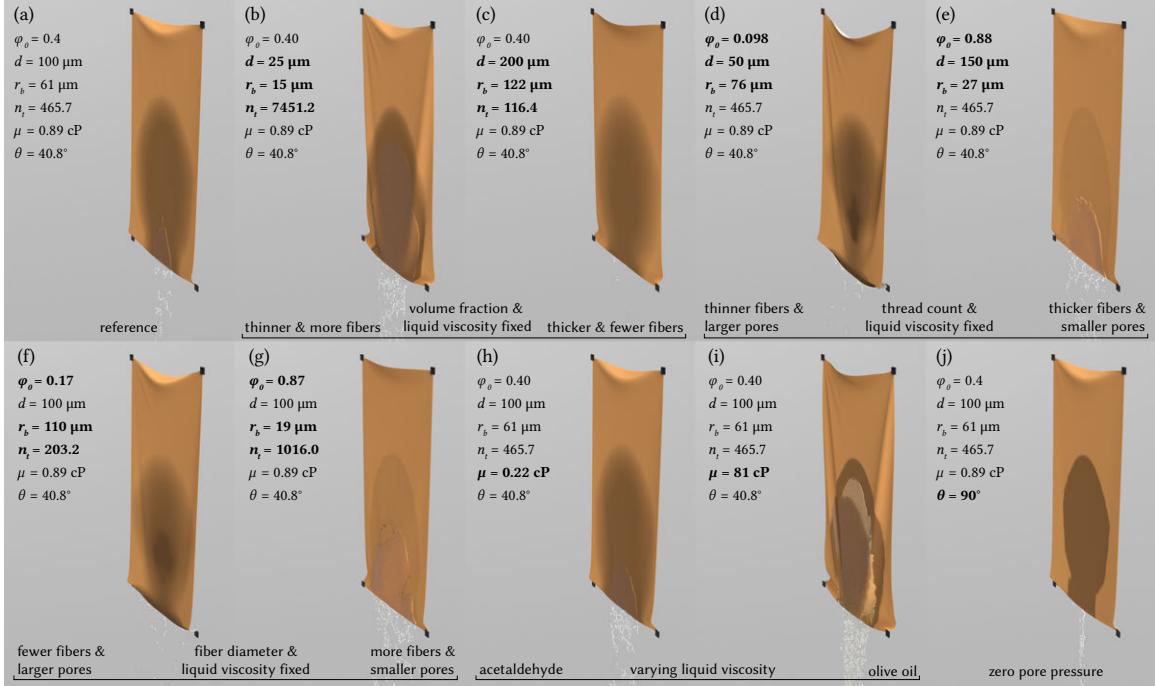


Figure 5.2: **Comparison for different sets of fabric and liquid parameters.** Parameters different from the reference's are highlighted with bold text. The fabric parameters include rest solid fraction ϕ_0 (unitless), fiber diameter d (micrometers), capillary radius r_b (micrometers), and thread count per inch n_t (unitless), where any two of them can be determined by the other two. The liquid parameters include viscosity η (centipoise) and contact angle ϕ (degrees).

more readily to the cloth surface, and a shorter wicking distance is observed. In Figure 5.2d and 5.2e, as we adjust the fiber diameter d , we simultaneously hold the thread count n_t constant; as d and the rest solid fraction ϕ_0 increased, the cloth is less easily penetrated by the liquid, also with less liquid retention inside the cloth. In Figure 5.2f and 5.2g, we change the thread count n_t , while holding the fiber diameter d constant; as n_t and the rest solid fraction ϕ_0 increased, the cloth shows a similar behavior. We also compare between different liquid parameters. In Figure 5.2h and 5.2i, we demonstrate the different behavior of acetaldehyde and olive oil, where the former is less viscous and the latter is much more viscous than water: the cloth

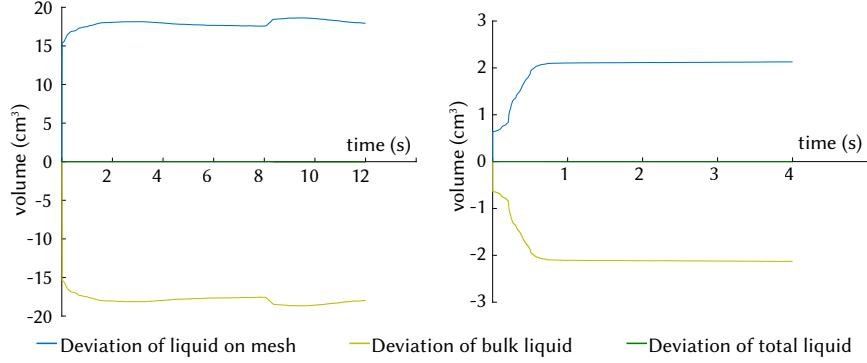


Figure 5.3: **Volume conservation** is demonstrated by plotting the deviation of fluid volume in bulk form (orange), on the cloth (blue), and their total (green). Left: *Tighten the Towel* example. Right: *Drag Forces* example.

is less easily penetrated by olive oil, which also has a much shorter wicking distance.

In Figure 5.2j, we demonstrate the effect where zero pore pressure is applied when the contact angle is exactly 90 degrees: there is then no wicking effect and the liquid is less attracted to the cloth surface.

Impermeable Cloth. We show that our method can also simulate liquid coupled with a pinned impermeable (non-porous) cloth, which corresponds to an infinitely small pore size and infinitely large drag force in our model. Since the drag force is implicitly integrated with (4.5), (4.17) and (4.18), our simulation is valid even when the drag tensor approaches infinity. Around the overlapped region, the liquid and solid share the same velocity, corresponding to a no-slip boundary condition.



5.2 General Examples

Splash on Cloth. Figure 5.1a demonstrates wetting, dragging, dripping, and wicking effects of liquid-cloth interaction. When the liquid has high velocity, it can penetrate through the cloth from one side to another, but as it is slowed down by viscous drag, it will attach to the cloth surface and start to slip. As more liquid attaches to the cloth, the cloth also starts drooping due to the added mass.

Splash on Yarns. Similarly, in Figure 5.1b we show that our model can handle yarn-based fabrics by dropping a ball of water on a piece of pinned handwoven fabric. Some of the liquid is captured by the fibers, while the majority of it flows through the pores and forms a liquid jet on the other side. The fabric is also noticeably tightened by the initial impact of the water ball.

Splash on Terrycloth. Beyond cloth and yarn, we show that our model can handle a scenario involving both kinds of structure: in Figure 5.1c we splash a ball of water onto a piece of terrycloth that has many short strands protruding from its surface. This cloth has a stiffer visual look than regular cloth, it absorbs more water, and the drag force is also stronger.

Tighten the Towel. Lastly, we show an example with more complicated dynamics in which the motion of a piece of terrycloth actively affects the flow of a liquid. Specifically, in Figure 5.1d we simulate the tightening of a towel. The towel is rapidly yanked out of water and tightened. As the towel twists, a sudden rush of liquid

Examples	s/step (Avg.)	# particle (Max)	# element	h (s)	Mem. (Peak GB)
Drag Forces	5.93	570K	8.54K	2×10^{-4}	5.41
Buoyancy Forces	6.02	742K	105K	2×10^{-4}	6.06
Various Parameters	1.88	199K	64.1K	2×10^{-4}	4.39
Impermeable Cloth	3.20	112K	71.8K	2×10^{-4}	2.44
Splash on Cloth	18.73	729K	336K	2×10^{-4}	18.35
Splash on Yarns	2.72	277K	79.6K	2×10^{-4}	6.19
Splash on Terrycloth	9.31	282K	123K	2×10^{-4}	5.27
Tighten the Towel	8.97	390K	71.2K	2×10^{-4}	4.90

Table 5.1: **Timings and storage statistics.**

flows out of the towel. As time goes on, the flow of liquid leaving the towel steadily decreases to a trickle.

For both *Tighten the Towel* and *Drag Forces* we measured the total volume of liquid on the towel and in bulk form over the course of the simulation. The volume of the bulk liquid is calculated as the sum of the spherical volumes of liquid associated with each APIC particle, according to each particle's radius. The volume of the liquid on the towel is calculated as the sum of the liquid stored on the vertices. For each solid vertex the liquid volume is simply the saturation multiplied by the empty pore space. Figure 5.3, left, shows that the net increase of water on the mesh (blue curve) was always offset by the net decrease in bulk liquid (orange curve), yielding remarkably good conservation of total liquid volume (green curve).

5.3 Performance Numbers

In Table 5.1 we collected timing data to evaluate the computational cost of our method and its various components on our examples, using a workstation with two Intel Xeon E5-2687W CPUs with eight cores each running at 3.10GHz. For the towel example we

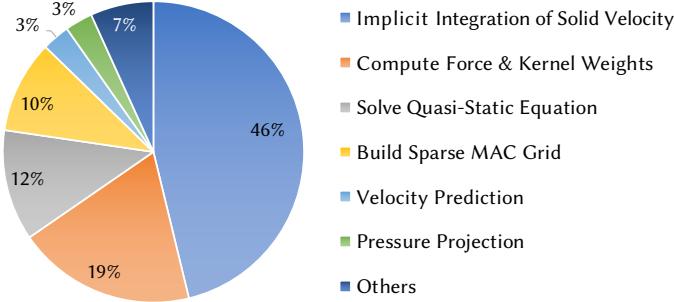


Figure 5.4: **Performance breakdown** for *Tighten the Towel* (Figure 5.1d).

also provide a detailed breakdown in Figure 5.4. The most time consuming part is for the calculation of the forces, plasticity and interpolation kernel weights. Throughout this part, we use a cell size of $\delta x = 0.288\text{cm}$, with an average distance between mesh vertices of 0.144cm . Since our grid is built only in a neighborhood around the solid vertices and liquid particles, its size is temporally variant. The number of cells varies between 50 and 250 in the largest dimension for all of our examples.

5.4 Surface Reconstruction and Rendering

We performed surface reconstruction in SideFX’s Houdini [214], which uses OpenVDB [171]. We adopted a *VDB from fluid particles* surface operator (SOP) to convert the liquid particles into a VDB grid, using a particle separation equal to the length of a simulation grid cell. To avoid flickering, we used a high-resolution VDB grid, where particles in a simulation cell are reconstructed with 8^3 VDB cells. We converted yarn strands into cylindrical tubes using the *PolyWire* SOP, with widths depending on the saturation of the yarns. These liquid tubes are then converted into a VDB using a *VDB from polygons* SOP. We combine the two VDB grids and use

Symbol	Physical quantity	Value	Unit
ϕ_0	rest volume fraction	0.098 ~ 0.88	n/a
d	fiber diameter	25.0 ~ 200.0	μm
r_b	capillary tube radius	15.0 ~ 122.0	μm
n_t	fabric thread count / square inch	0.1K ~ 7.4K	n/a
r_c	cloth half thickness or yarn radius	165.0 ~ 480.0	μm
λ	power of ϕ in effective stress	2.0	n/a
γ	surface tension coefficient	20.6 ~ 72.0	dyn/cm
η	liquid viscosity	0.22 ~ 81.0	centipoise (cP)
c	nonlinearity of drag coefficient	1.6	n/a
ρ_s	solid intrinsic density	0.25 ~ 4.0	g/cm^3
ρ_f	liquid intrinsic density	0.78 ~ 1.0	g/cm^3
θ	contact angle	40.8 ~ 90.0	degree

Table 5.2: **Range of physical parameters adopted throughout all examples.** Unless specified, we use fiber diameter and fabric thread count as input, and compute other parameters through their relationships given in table 5.3.

a dilate-smooth-erode operator [170] to create the smooth transition between them.

Besides affecting the bulk and surface liquid geometry, wet fabrics are usually darker and more specular [120]. We adopted a simple, customized shader to incorporate this effect, where the diffuse color, reflection, sheen, subsurface scattering, and roughness are modulated using linear functions of saturation S_r .

5.5 Fabric Parameters

In table 5.2 we list all the physics parameters used throughout this work, as well as their approximate ranges and units.

The rest volume fraction and capillary tube radius are computed through a simple geometric model: consider a piece of woven cloth or a piece of yarn in a knitted fabric composed of uniformly packed cylindrical fibers. The effective radius of the capillary tubes are computed from the empty volume between these fibers. By geometric

In\Out	d	r_b	n_t	ϕ_0
(r_b, ϕ_0)	$2r_b \sqrt{\frac{\phi_0}{1-\phi_0}}$	-	$\frac{2r_c s}{\pi r_b^2} (1 - \phi_0)$	-
(r_b, n_t)	$\sqrt{\frac{8r_c s}{\pi n_t} - 4r_b^2}$	-	-	$1 - \frac{\pi r_b^2 n_t}{2r_c s}$
(n_t, ϕ_0)	$\sqrt{\frac{8r_c s \phi_0}{\pi n_t}}$	$\sqrt{\frac{2r_c s}{\pi n_t} (1 - \phi_0)}$	-	-
(d, n_t)	-	$\sqrt{\frac{2r_c s}{\pi n_t} - \frac{d^2}{4}}$	-	$\frac{\pi d^2 n_t}{8r_c s}$
(d, ϕ_0)	-	$\frac{d}{2} \sqrt{\frac{1-\phi_0}{\phi_0}}$	$\frac{8r_c s \phi_0}{\pi d^2}$	-
(d, r_b)	-	-	$\frac{8r_c s}{\pi(4r_b^2 + d^2)}$	$\frac{d^2}{d^2 + 4r_b^2}$

Table 5.3: **Conversion between fabric parameters.** From any given pair of two parameters, the other two can be computed. We take $s = 2.54\text{cm/in}$ since the fabric thread count habitually taken in per square inch needs to be converted to the centimetergramsecond (CGS) system used throughout this part.

calculations, the relationship between different fabric parameters (fiber diameter d , radius of capillary tubes r_b , fabric thread count per square inch n_t , rest volume fraction ϕ_0 , and yarn radius or cloth half thickness r_c that is always given as user input) used throughout this part are presented in table 5.3.

Chapter 6

Limitations of the Wet Fabrics Simulator

We have presented a numerical model to animate liquid interactions with permeable cloth and yarn that is able to capture many key phenomena. We highlight below a few limitations imposed by our chosen assumptions, numerical methods, or experiments.

In Figure 3.2 we compared the diffusion simulated by our method with a laboratory experiment. While we have acquired a result close to the laboratory experiment through the manual specification of a volume fraction field, we found very difficult to match perfectly. This is because there are other factors that would affect the diffusion, such as the spatially varying fiber radii that changes the pore pressure and the abrasion of the textile sample that produces irregular bumps on the surface. In future work it would be worth investigating how to model and incorporate these textile “defects” for more realistic simulation.

Our fiber model makes assumptions about the dominant axes of the pore structure, which places limitations on the fidelity of our pore pressure and drag forces for general microstructures; for example, the yarn strands in a piece of terry cloth are assumed to attach perpendicularly to the cloth. For numerical efficiency, our drag model also relied on a lumping strategy that assumes shear and rotational effects are relatively

unimportant.

Since the liquid bridge geometry that causes wet cloth sticking is difficult to model, we adopted a fairly simple cohesion approach. Of course, there are situations in which cohesion has a very meaningful influence on the dynamics: consider the manner in which wet clothes adhere to one's body. Relatedly, we did not include surface tension in the bulk fluid flow, though adding an explicitly integrated approach would likely be straightforward.

More fundamentally, our system relies on mixture and porous flow theories, which themselves entail a variety of both limitations along with benefits. Principally, they assume continuum models of the phenomena and their interactions, for example abstracting away real fine-scale geometry of individual droplets and pores. In both engineering and animation this extreme level of detail is often superfluous, though not universally. For example, in the ring test, it is likely that we might recover *some* of the differences from the physical experiment with a more faithful coarse-scale model of the specific fabric geometry we used; however, certain small-scale heterogeneities, wrinkles, etc. seem likely to remain beyond the reach of our scheme.

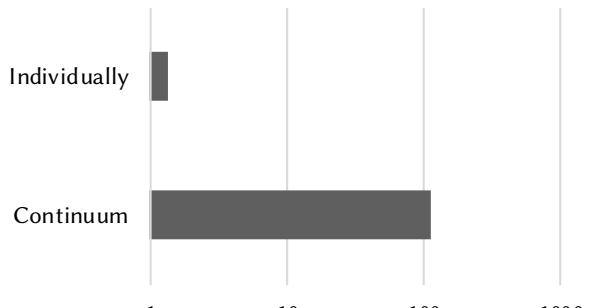
We adopted weak coupling through the drag force and do not enforce an exact matching of velocity at the interface. The liquid and solid are treated as a continuum mixture, and the drag force acts to pull the liquid velocity closer to that of the solid. The scale of the drag force depends on the solid permeability. As such, the solid velocity will only exactly match the liquid velocity in the limit of infinitely large drag

force (corresponding to the scenario of an impermeable cloth). We indeed observed artifacts when the discretization is too coarse, which is a limitation of continuum modeling.

At present, our method relies on a relatively fine grid resolution to achieve realistic results: the fabric thickness is not even a full order of magnitude smaller than a grid cell. Ideally, one would prefer a large gap to reduce the significant cost of the volumetric solve, although ensuring seamless interactions under such coarse conditions appears non-trivial. In a related vein, strictly speaking our model is semi-implicit which implies a time step restriction that could slow our results; a fully implicit solution remains an open question. However, in practice, the primary factor driving time step selection was collision-handling, rather than internal dynamics. For the sake of efficiency, we also adopted a free surface model that avoids simulating the bulk air volume; this offers a faster simulation but sacrifices air-dependent effects such as (grid-scale) bubbles.

Finally, although this framework works well for fabrics, we find it inefficient for stiff materials, such as hair strands. This issue is caused by the continuum assumption, which means all the hair strands, as a continuum, have to be

Avg. Computational Time for 4K Hairs
(secs / timestep, no collision)



solved as a whole system. Due to the scale of the system, one has to use an iterative solver. For stiff materials, however, an iterative solver usually needs thousands of

iterations to converge. As shown in the figure, simulating 4K hairs as a continuum, on average, is more than $100\times$ slower than solving the dynamics of each hair individually. In the next two chapters, we will introduce a new framework to couple the strands with liquid, where the strands are no longer simulated as a continuum but as discrete elements. In **Part II**, we will introduce the cohesion between strands, and a simple Newtonian surface flow on the surface of strand; in **Part III**, we will generalize the surface flow to shear-dependent fluid, and more importantly, we will derive a principled coupling strategy between wet discrete elements and a continuum phase.

Part II

Multi-Scale Model for Simulating Liquid-Hair Interactions

Chapter 7

From Wet Fabrics to Wet Hairs

In **Part I**, we have developed a multi-scale model simulating the liquid-fabric interactions, where we treat the liquid and fabric both as a continuum. Nevertheless, generalizing that model for hair strands may incur numerical difficulties, as explained in §6: it is more practical and efficient to simulate the hair strands individually.

Furthermore, coupling hairs with liquid requires that we develop a new model. For example, when removed from the liquid, hair inevitably retains a certain amount of liquid on its surface, which then flows along the hair strands and eventually drips off. The flow on hairs suffered from drastic movements can be fast and has the inertial effect, which cannot be captured by a Darcy-like diffusion model — where the inertia of flow is neglected — proposed in **Part I**. Meanwhile, the liquid on the surface of one hair also interacts with the liquid on the surface of another hair by forming a liquid bridge, whose surface energy would cause the cohesion of hair strands.

To effectively capture the phenomena for wet hairs simulated individually, in this part, we develop a framework with the following novel components:

- A surface liquid model for flow along an individual hair strand, which reduces

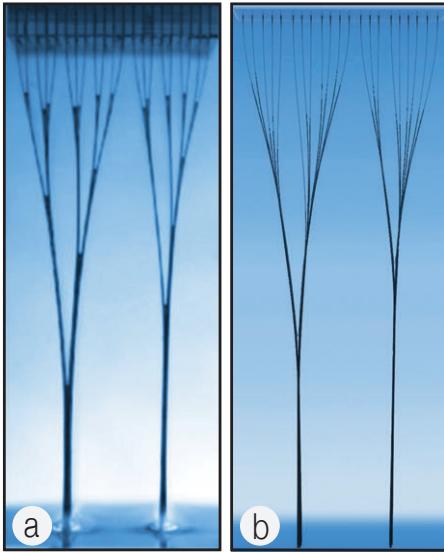


Figure 7.1: Recursive structures in wet hair coalescence. (a) A picture of physical laboratory experiment of Bico et al. [2004]. (b) Our simulated result with a similar setup.

by one dimension the shallow-water equations by invoking rotational symmetry

and introduces inertial forces due to acceleration of the hair centerline;

- A model for surface tension-induced cohesion forces between wet hairs;
- A model for dripping of reduced liquid off the hair strands;
- A model for capturing volumetric liquid onto the hair strands.

We will demonstrate that, collectively, these features enable a higher fidelity simulation (e.g., Figure 7.1) of the compelling dynamics of wet hairs than achieved by existing techniques.

Chapter 8

Wet Hair Physics

We begin by describing our physical models in the continuous setting, which are specifically tailored to capture hair-liquid dynamics at different scales and in different configurations. We derive a new surface tension model to compute forces between hairs that are bridged by liquid (§8.1); we derive a dimensionally-reduced model of liquid flow on the hair surface, in a manner analogous to shallow water equation [205], to simulate liquid flows along and between hair strands (§8.2).

8.1 Force between Hairs

We derive a new force model that computes the surface tension force between two wet hairs in proximity. This force tends to pull the hairs together, but is balanced by collision forces when two hairs contact each other. We also extend our surface tension model to compute adhesive forces between wet hairs and objects.

Cohesion due to Surface Tension

When two wet hairs are in proximity, a liquid bridge is formed (Figure 8.2). Microscopically, liquid molecules attract each other due to their cohesive forces, which



Figure 8.1: Water flowing over and through a sloped mat of fur. Straight hair initially clump and then become flattened by the weight of water captured by the strands.

creates a pressure imbalance at the surface causing the liquid to contract towards a minimal area configuration, subject to conservation of volume. Our surface tension model uses a formulation based on the surface potential energy, the energy needed to form liquid surface area. The surface tension force is the differential of this surface potential. Consider a 2D cross section of two hairs connected by a small drop of liquid, as depicted in Figure 8.2. With the notation introduced therein, the surface potential over an infinitesimal volume near the cross section is proportional to its surface area, expressed as

$$dE_s = \sigma [l_A(s) + \cos \theta l_S(s)] ds, \quad (8.1)$$

where σ is the surface tension coefficient of the liquid-air interface (i.e., 71.97dyn/cm for water at a room temperature of 298.15K). The equilibrium contact angle, θ , at which the liquid-air interface meets the liquid-hair interface is a constant that depends on the material properties of liquid, air, and hair. The arc lengths of the liquid-air and hair-air boundary are given by l_A and l_S , respectively (Figure 8.2-c), and ds is

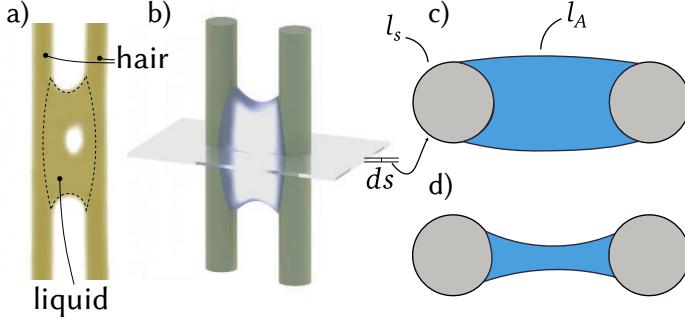


Figure 8.2: **A liquid bridge.** (a) Labeled microscopic photo of liquid held by hairs, courtesy of Wang et al. [243]; (b) rendered figure of the geometry; (c) cross section of a slice of the geometry; (d) another possible cross section with less liquid.

an infinitesimal length along the centerline of the hairs. Since dE_s can vary along the hair's centerline, the total surface potential is an integral over the length Γ of the hairs (i.e., $\int_{\Gamma} dE_s$). The expression (8.1) can be derived using Young's equation [260] as below.

Surface Potential of Wet Hair We can derive the surface potential energy used for the cohesion force as follows. Given the surface tension coefficient between the liquid-air interface σ , the solid-air interface σ_{SA} , and the liquid-solid interface σ_{LS} , the surface tension energy is the surface area times the corresponding coefficients

$$dE_s = [\sigma l_A(s) + \sigma_{SA}l_S(s) + \sigma_{LS}l_{LS}(s)] ds, \quad (8.2)$$

Applying the Young's equation [260],

$$\sigma_{SA} - \sigma_{LS} - \sigma \cos \theta = 0 \quad (8.3)$$

where θ is the equilibrium contact angle, we have

$$dE_s = [\sigma (l_A(s) + l_S(s)\cos\theta) + \sigma_{LS} (l_S(s) + l_{LS}(s))] ds. \quad (8.4)$$

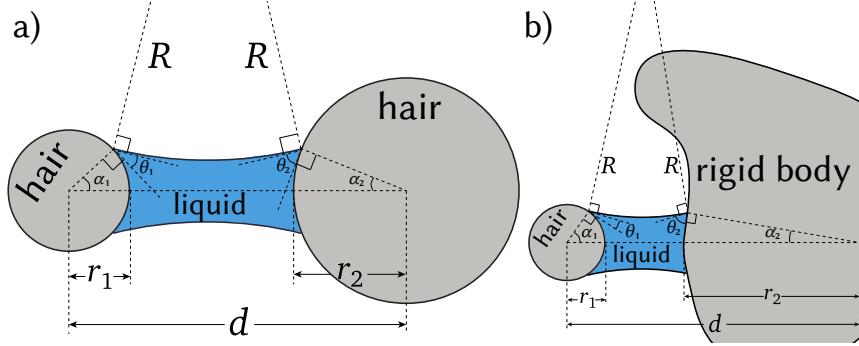


Figure 8.3: Notation for a liquid bridge cross section.

The sum of the length of the solid-air and liquid-solid interfaces is exactly the length of the solid interface, which is a constant value and does not affect the potential gradient. For our purpose we can simply set $\sigma_{LS} = 0$. Hence we have the form given in (8.1). ■

Our cohesion model builds on the investigation of Liu et al. [150]. We summarize their argument and then indicate our point of departure. First, when two hairs are close, surface tension dominates over gravity. If we ignore gravity, we notice that the top and bottom liquid-air interfaces must be symmetric, consisting of two circular arcs. This is because the surface energy contributed by molecular cohesive forces is uniformly distributed over the interfaces. Following the notation in Figure 8.3, the cross-sectional surface potential is expressed as

$$dE_s = \sigma [l_A(s) + \cos \theta_1 l_{S1}(s) + \cos \theta_2 l_{S2}(s)] ds, \quad (8.5)$$

where l_{S1} and l_{S2} are the arc lengths of the two hair-air boundaries, and θ_1 and θ_2 are the contact angles. (In this case, they are a known constant, $\theta_1 = \theta_2 = \theta$, as

discussed earlier.) Let r_1 and r_2 be the radii of the two hairs, and R be the radius of the circle corresponding to the liquid surface arcs. Since the liquid is bounded by circular arcs, the length of the liquid-air interface is given simply by

$$l_A(s) = R [\pi - (\theta_1 + \alpha_1 + \theta_2 + \alpha_2)] \quad (8.6)$$

and the length of the hair-air interface by

$$l_{Si}(s) = 2r_i(\pi - \alpha_i), \quad i = 1, 2. \quad (8.7)$$

However, what we would ultimately like to know is the dependence of the energy on the distance d between hairs. Therefore, we next need to find the dependence on d of R and α_i above.

Geometric arguments can be used to show that the distance d can be expressed as

$$d = R \sum_{i=1,2} \cos(\theta_i + \alpha_i) + \sum_{i=1,2} r_i \cos \alpha_i, \quad (8.8)$$

and the cross-sectional area of the liquid region of the bridge, A_L , as

$$\begin{aligned} A_L = -\pi R^2 + \sum_{i=1,2} & \left[\frac{1}{2} r_i^2 \sin 2\alpha_i + 2r_i R \sin \alpha_i \cos(\theta_i + \alpha_i) \right] + \\ & \sum_{i=1,2} \left[R^2 \left(\theta_i + \alpha_i + \frac{1}{2} \sin(2\theta_i + 2\alpha_i) \right) - \alpha_i r_i^2 \right]. \end{aligned} \quad (8.9)$$

Liu et al. [150] apply a variation of these equations to an analysis of capillary rise

between cylinders, subject to fixed boundary conditions (hair positions).

In a point of departure, we assume that the hair positions are variable, and employ these equations to determine the dependence of the surface energy on the distance d between hair centerlines, thereby yielding the inter-hair surface tension forces as the gradient of this energy.

Liu et al. [150] also showed that the arc lengths of the solid-liquid interfaces are equal, i.e., $r_1\alpha_1 = r_2\alpha_2$. Substituting this relationship into (8.8) and (8.9) above and differentiating both sides yields (dropping the subscript on α and using $\alpha \equiv \alpha_1$ for brevity)

$$\begin{bmatrix} \frac{\partial A_L}{\partial R} & \frac{\partial A_L}{\partial d} \\ \frac{\partial d}{\partial R} & \frac{\partial d}{\partial d} \end{bmatrix} \begin{bmatrix} \frac{\partial R}{\partial d} \\ \frac{\partial \alpha}{\partial d} \end{bmatrix} = \begin{bmatrix} \frac{dA_L}{dd} \\ \frac{dd}{dd} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (8.10)$$

where $\frac{dA_L}{dd}$ vanishes in the second equivalence. Since the longitudinal dimension is much larger than the transverse dimensions for hairs, we approximate the area of the liquid region to be constant as d changes. In turn, the fluid incompressibility is respectively enforced in the longitudinal and transverse dimensions. The left-hand-side Jacobian matrix can be analytically computed from (8.8) and (8.9). We then solve this 2×2 system for $\frac{\partial R}{\partial d}$ and $\frac{\partial \alpha}{\partial d}$. At last, we can evaluate the (magnitude of) cross-sectional surface tension force using

$$f_s = \frac{d}{dd} dE_s = \frac{\partial dE_s}{\partial R} \frac{\partial R}{\partial d} + \frac{\partial dE_s}{\partial \alpha} \frac{\partial \alpha}{\partial d}, \quad (8.11)$$

where both derivatives $\frac{\partial dE_s}{\partial R}$ and $\frac{\partial dE_s}{\partial \alpha}$ will be computed by plugging (8.6) and (8.7)

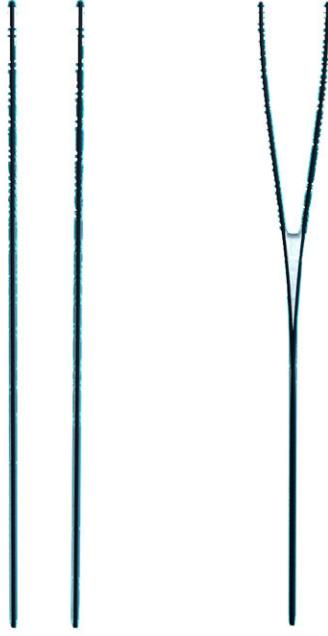


Figure 8.4: Two wet hairs held close and then pulled apart. Left: Hair strands freely separate without cohesion. **Right:** Cohesion forces cause hair strands to adhere. **Note:** for better visibility of small-scale illustrative examples in paper images (only), the radius of hair and surface liquid have both been scaled up.

into (8.5) and differentiating analytically. Details of the method of evaluating f_s are deferred to §9.1.

As depicted in Figure 7.1, our liquid-bridge based model of cohesion is able to reproduce the tree structures observed in the laboratory experiments of Bico et al. [33].

Adhesive Forces between Hairs and Solids. We can straightforwardly extend our surface tension model to determine the adhesive force between a hair and a solid object resulting from liquid bridges. The local geometry of a solid object can be approximated by a sphere whose curvature agrees with the local mean curvature of the solid object (Figure 8.3b). This approximation allows us to apply the surface tension model developed above, with two modifications: 1) using different contact angles to account for the solid object's material being different from the hair, and 2) using the radius of the local sphere that approximates the solid during the computation of the hair-body force (i.e., the reciprocal of the body's local curvature replaces r in the

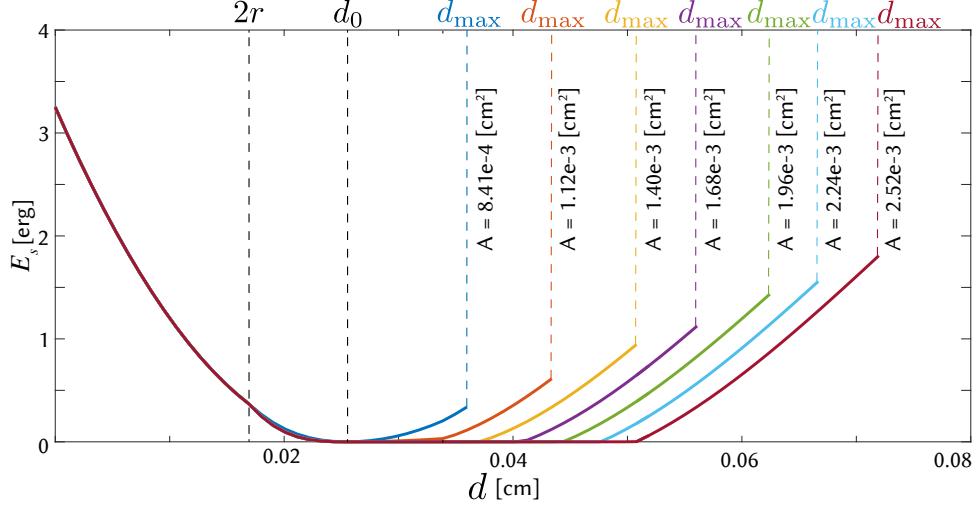


Figure 8.5: **Plot of the potential energy of the combined cohesion/repulsion force vs. distance between two hairs with unit length.** Curves are cut off at d_{\max} .

equations above).

Collision Forces

We adopt a simple penalty model to treat collisions and contact. Consider two hairs, each with radius r , having a distance d between their centerlines. Both surface tension and contact forces between the hairs depend solely on the distance d , but each dominates at a different d : penalty forces dominate when hairs overlap one another ($d < 2r$, where r is the hair radius), while surface tension dominates when hairs are separated but at a distance smaller than d_{\max} , a critical value at which the liquid bridge between the hairs breaks. Physical experiments have shown that d_{\max} depends on the equilibrium contact angle θ and the cross-sectional liquid area A_L

(defined in (8.9)) through an empirical relationship [147],

$$d_{\max} \equiv (1 + 0.5\theta)\sqrt{A_L}. \quad (8.12)$$

At some distance d_0 between $2r$ and d_{\max} , the penalty force balances surface tension, and the net force vanishes.

Surface tension and collision penalty forces by nature conflict with each other. Later, in our numerical simulation, if we compute them separately in each simulation step, we need either many iterations or tiny time steps until both forces reach a balance. To avoid this difficulty, we reconcile them in a single force model described as a piecewise function,

$$f(d) \equiv \begin{cases} k(d - d_0), & \text{if } d < 2r, \\ \frac{d-d_0}{B-d_0}\tilde{f}_s(d), & \text{if } 2r \leq d < d_0, \\ \frac{d-d_0}{B-d_0}f_s(B), & \text{if } d_0 \leq d < B, \\ f_s(d), & \text{if } B \leq d < d_{\max}. \end{cases} \quad (8.13)$$

Here k is the stiffness of the penalty force. The distance value B is chosen to allow a smooth (linear) transition from $f(d_0) = 0$ to the surface tension force $f(B) = f_s(B)$. In practice, we use $B \equiv 2d_0 - 2r$. The function $\tilde{f}_s(d)$ interpolates values between $f(2r)$ and $f(d_0) = 0$, expressed as

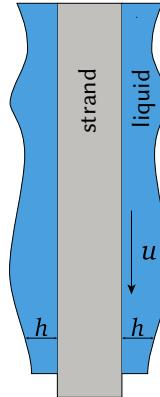
$$\tilde{f}_s(d) \equiv \frac{(d - d_{\min})f_s(B) + k[(d + B - d_0)d_0 - dB]}{d_0 - d_{\min}}. \quad (8.14)$$

Figure 8.5 plots a typical force profile with respect to d . We use $d_0 \equiv 3r$ throughout our implementation. Note that in (8.12), if the hair is dry, $A_L = 0$ and thus $d_{\max} = 0$; we therefore simply ignore the two formulas in the bottom of (8.13) and thus naturally handle the case of dry hair.

This combined model for contact and hair adhesion forces is straightforward to implement, reasonably efficient, and may avoid the stability issues that arise for staggered/decoupled treatments under moderate cohesion.

8.2 Liquid Flows over the Hair Surface

Flow along hair Wet hairs are covered by a thin layer of liquid flowing on the hair surface. Simulating the surface flows using the standard Eulerian approach is intractable, because the thin liquid layer and the long hair length relative to the tiny hair radius would demand an excessively fine Eulerian grid. However, we can exploit this disparity in length scales to approximate the liquid flow using a dimension-reduced model as in the shallow liquid equations [128, 241, 242]. In this way, the liquid's motion can be described in the reference frame of the hair; however, we must also consider the effects of the motion of this non-inertial reference frame itself. Thus the surface liquid's velocity has two components: the *intrinsic* hair-surface liquid velocity, i.e., velocity of shallow liquid-like flow with respect to the hair; and the *extrinsic* hair-surface liquid velocity, i.e., the velocity of the liquid due to the motion of the hair that carries it.



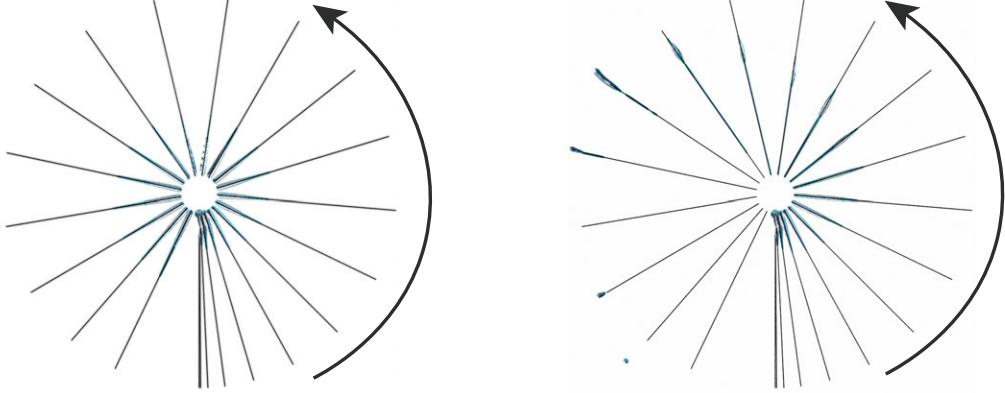


Figure 8.6: **Inertial force.** When a hair rotates about a central point, centrifugal forces cause the liquid to flow towards and off the tip. We compare our simulation without (**left**) and with (**right**) the inertial force. Time-slices are shown in order proceeding counter-clockwise from the bottom.

Additionally, because of the small hair radius, the transversal liquid flow *around* a hair strand occurs on a much smaller time scale in comparison to longitudinal flows. Thus, we assume a quasi-static liquid layer thickness that distributes uniformly at a transversal cross section of the hair, while the thickness along the hair can vary. We apply this approximation to all hairs, even in the presence of close contact between different hairs. This choice allows us to derive a 1D reduced-dimensional model in a manner analogous to the shallow water equations (see below for the derivation details), and strikes a balance between strict physical validity and practical computational expense. We model the 1D longitudinal intrinsic reduced velocity u_τ and liquid layer cross sectional area A_τ with the governing equations

$$\frac{\partial u_\tau}{\partial t} = -u_\tau \frac{\partial u_\tau}{\partial x} - \frac{1}{\rho_f} \frac{\partial p}{\partial x} + a, \quad (8.15)$$

$$\frac{\partial A_\tau}{\partial t} + \frac{\partial}{\partial x} (A_\tau u_\tau) = 0, \quad (8.16)$$

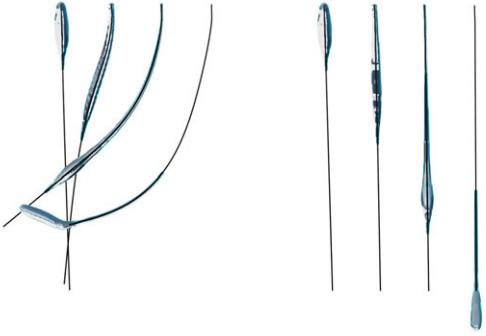


Figure 8.7: A droplet sliding down a single vertical hair as it moves under uniform horizontal velocity. Under (only) downward gravitational acceleration, the hair should remain vertical, in the absence of external horizontal forces. We compare na'ive local momentum rescaling (**left**) versus our momentum update transport (**right**).

where u_τ , A_τ , and p are all 1D functions along the hair, parameterized by the coordinate x . The variable p is the hydrostatic pressure value over a cross section, ρ_f is the liquid density, a is the acceleration introduced by external forces (such as gravity and inertial forces). The cross sectional area is computed as $A_\tau \equiv (h_\tau + r)^2 - r^2$ where r is the hair radius, and h_τ is the liquid layer thickness. Physically, (8.15) is derived from conservation of momentum; (8.16) is derived from conservation of mass, and describes the advection of liquid as the temporal evolution of the cross sectional area (and correspondingly, layer thickness h). We defer the computation of the inertial force (Figure 8.6) to the discrete setting, in §9.2.

Derivation of the Surface Liquid Model on Hair The derivation for our 1D surface liquid model generally follows the strategy used for the classical shallow water equation [205], with differences that account for the cylindrical geometry. We first parameterize the variables on the tangential direction of a hair segment.

$$u_\tau = u_\tau(x), h_\tau = h_\tau(x), A_\tau = A_\tau(x) \quad (8.17)$$

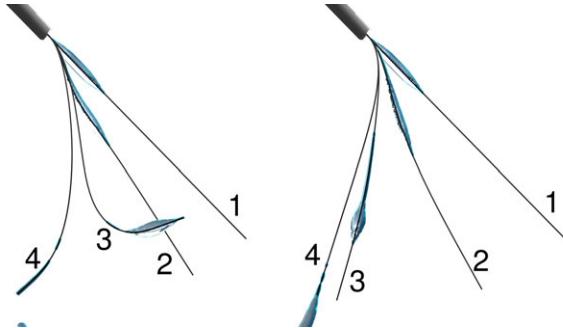


Figure 8.8: A droplet sliding down a single hair, causing it to swing. **Left:** Naïve rescaling leads to severe spurious bending. **Right:** Our momentum transports gives a natural motion.

where u_τ is the velocity in the frame of a hair segment, and A_τ is the area of the cross section.

Consider the mass flux passing through a point on the hair segment. For a point x the mass flux is $\rho_f A_\tau(x) u(x) dt$, and the flux passing through its neighbor coordinate $x + dx$ is $\rho_f A_\tau(x + dx) u_\tau(x + dx) dt$. Hence we have the equations about the flow's mass m_τ where

$$\begin{aligned} dm_\tau &= \rho_f A_\tau(x) u_\tau(x) dt - \rho_f A_\tau(x + dx) u_\tau(x + dx) dt \\ \frac{dm_\tau}{dt} &= -\rho_f \frac{A_\tau(x + dx) u_\tau(x + dx) - A_\tau(x) u_\tau(x)}{dx} \end{aligned} \quad (8.18)$$

Since $dm = \rho_f dA_\tau dx$, the factors of $\rho_f dx$ cancel on both sides leaving us with

$$\frac{dA_\tau}{dt} = -\frac{A_\tau(x + dx) u_\tau(x + dx) - A_\tau(x) u_\tau(x)}{dx} \quad (8.19)$$

Taking the limit of $dx \rightarrow 0$ and $dt \rightarrow 0$, we have

$$\frac{\partial A_\tau}{\partial t} = -\frac{\partial(A_\tau u_\tau)}{\partial x} \quad (8.20)$$

■

Momentum Transport along Hair The thin layer of liquid weighs down the hair by an additional mass density of $\pi(h^2 + 2hr)$. Consequently, the evolution of h —transport of liquid along the hair—alters the effective mass distribution of the hair. This is critical to consider because the Lagrangian hair is itself a dynamic system; redistributions of mass along the hair, without redistributions of velocity, would lead to sudden changes in the hair momentum, and corresponding artifacts in motion.

To properly account for evolution of the hair state due to surface liquid, we therefore transport hair *momentum* according to the same surface liquid equations. After updating the (intrinsic) surface liquid velocity with (8.15) and the height field with (8.16), we also solve the additional transport equations,

$$\frac{\partial}{\partial t} (\mathbf{u}_s A_\tau) + \frac{\partial}{\partial x} (\mathbf{u}_s A_\tau u_\tau) = 0, \quad (8.21)$$

$$\frac{\partial}{\partial t} (\omega_s A_\tau) + \frac{\partial}{\partial x} (\omega_s A_\tau u_\tau) = 0, \quad (8.22)$$

where $i \in \{x, y, z\}$ are the labels for three components of the Lagrangian hair velocity, \mathbf{u}_s , and ω_s is the angular velocity in the degree of freedom for hair twisting. The momentum transport only happens on the surface liquid. Hence, after the new \mathbf{u}_s and ω_s are computed (denoted as $\tilde{\mathbf{u}}_s$ and $\tilde{\omega}_s$) with these equations, we combine them with the velocity before momentum transport (denoted as $\hat{\mathbf{u}}_s$ and $\hat{\omega}_s$) using

$$\mathbf{u}_s \leftarrow (\rho_s \pi r^2 + \rho_f A_\tau)^{-1} (\rho_s \pi r^2 \hat{\mathbf{u}}_s + \rho_f A_\tau \tilde{\mathbf{u}}_s), \quad (8.23)$$

$$\omega_s \leftarrow (\rho_s \pi r^2 + \rho_f A_\tau)^{-1} (\rho_s \pi r^2 \hat{\omega}_s + \rho_f A_\tau \tilde{\omega}_s). \quad (8.24)$$

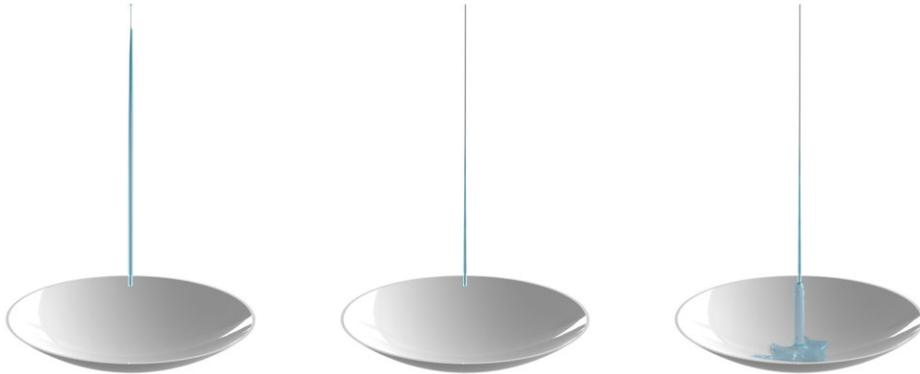


Figure 8.9: **Surface liquid flow and dripping on a single strand.** **Left:** No surface liquid flow. **Center:** Surface liquid flow without dripping; the liquid flows towards the tip and disappears. **Right:** Surface liquid with dripping enabled.

This approach is more costly than evolving mass alone, but it avoids the artifacts of ignoring momentum conservation. For example, a na^{ive} alternative might be to *locally* preserve momentum by rescaling the local hair velocity based on its updated mass; however, this ignores the momentum of the liquid flowing along the hair.

In Figures 8.7 and 8.8, we show two scenarios of a droplet sliding on a single hair, illustrating that our method produces physically meaningful results, while na^{ive} rescaling produces severe motion artifacts in the hair.

Remark: In this part we just simply agree with the statement above, and explicitly modify the strand's velocity after solving (8.21) and (8.22). A more rigorous justification will be presented in **Part III**, where we also discover that solving these transport equations is equivalent to adding an additional inertial term to the momentum equation of strands, and implicitly solving the latter provides better stability than the method presented in this part.



Figure 8.10: **Left-Top:** Neither surface liquid flow nor dripping enabled. **Left-Bottom:** With surface liquid flow enabled the liquid accumulates in the center and the hair sags. **Right:** Enabling dripping lets the liquid flow off the hair.

Dripping from hairs For liquid on the tip of a hair, the amount dripping off is simply determined by the flux of velocity off the tip (or zero, if the velocity is directed away from the tip). However, since an individual hair also cannot support an arbitrary amount of liquid at any given point along its length, we must determine a threshold for the local liquid volume allowed, and a mechanism to treat the excess liquid that drips off at that point (see Figure 8.9 and Figure 8.10).

We use a threshold determined from theory of capturing droplets with thin fibers [151]: given a region filled with N fibers, the averaged intensity of acceleration $a_{\hat{n}}$ applied on the normal directions of flows (perpendicular to the tangential direction of the hair), the surface tension coefficient σ , the radii of a thin fiber r , and the density of liquid ρ_L , the surface tension force applied is $4\pi r\sqrt{N}\sigma$. To capture the liquid we need the surface tension force to balance the other forces applied on the liquid, which is $\rho_L a_{\hat{n}} V$, where V is the volume of liquid attached to the small segments of hairs. Hence the maximal radius of droplet that can be held by the hair segments is computed as

$$r_{\max} = \left(\frac{3r\sigma\sqrt{N}}{\rho_L a_{\hat{n}}} \right)^{1/3}. \quad (8.25)$$

N is computed by counting how many hairs intersect with a given grid voxel. After

determining the maximum radius of liquid droplet that can be held by these N hairs, we remove the extra volume of liquid from the hairs and release them as APIC particles in accordance with Equ. 8.25, where any liquid outside the extent of r_{\max} is free to drip away from the hairs.

Chapter 9

Numerical Simulation of Wet Hairs

Having described our physical models, we now present the numerical methods we use to discretize and couple these models together, and thereby simulate hair-liquid interactions.

Method Overview. The simulation consists of three interacting components: hair strands simulated using the discrete elastic rods (DER) method [26, 28]; bulk liquid simulated using the affine particle-in-cell (APIC) method [122]; and hair surface liquid modeled using a novel formulation.

At the beginning of each time step, the hair, bulk- and surface liquid are advected separately. We then transfer the bulk- and surface liquid velocities onto a shared staggered background grid, which unifies the liquid momentum associated to the two representations, and serves to exchange momentum, drag and pressure forces in APIC style. Each grid face stores a (normal) velocity flux and density, sufficient to reconstruct the momentum flux across the face.

Figure 9.1 provides a visual overview of our algorithm and the flow of data through it for a single time step. The individual steps are:

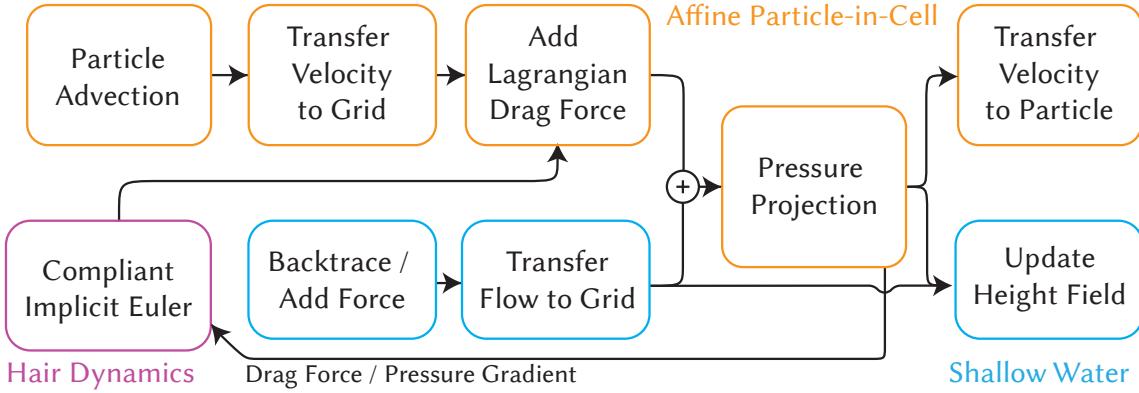


Figure 9.1: Algorithm overview and data flow for simulating wet hairs.

1. Advect the APIC particles.
2. Solve the hair dynamics, incorporating the drag force using velocities sampled from the grid, adhesive/repulsive forces between hairs, and the pressure impulse from the liquid stored from the previous step.
3. Advect and apply forces to the surface liquid on the hair.
4. Transfer the velocity of the APIC particles onto the grid.
5. Apply the Lagrangian drag force from the hair onto the grid in an Eulerian style.
6. Transfer the velocity of the reduced liquid onto the grid.
7. Solve the Poisson equation and perform pressure projection (standard incompressible fluid solver).
8. Transfer the grid velocity back to the particles with APIC.
9. Update the velocity of the reduced liquid due to surrounding pressure.
10. Exchange liquid volume between APIC and surface liquid domains through capture and dripping, accounting for conservation of mass.

9.1 Hair Simulation

We discretize a hair as a sequence of vertex masses endowed with the viscoelastic stretching, bending, and twisting forces of the discrete elastic rod model based on Kirchhoff rod theory [26, 28].

Discretization of Cohesion and Contact Model

When wet, proximate hairs adhere due to a strong cohesive force (8.13). Because this force is strongly nonlinear in the inter-hair distance, and vanishes at distance d_{\max} , a naive evaluation strategy is prone to spatiotemporal discontinuities with attendant instabilities and “popping” artifacts.

To enforce spatiotemporal continuity of the force we employ the adaptive quadrature method depicted in Figure 9.2, which employs a single-point quadrature at the centroid of the *adapted subdomain* consisting of distances closer than d_{\max} ; the crucial aspect is that the boundary (and therefore centroid) of the adapted subdomain varies smoothly with vertex positions, ensuring that the force is first-order continuous. To simplify the computation, we make the gross approximation that inter-hair closest-point distances vary linearly along an edge (in actuality, they are piecewise linear/quadratic). Therefore, our closest-point distances are only accurate at the vertices, but since the approximation still guarantees smoothly varying adapted subdomains, our central concern is heeded.

We build a graph between the hair segments in two phases, as depicted in Fig-

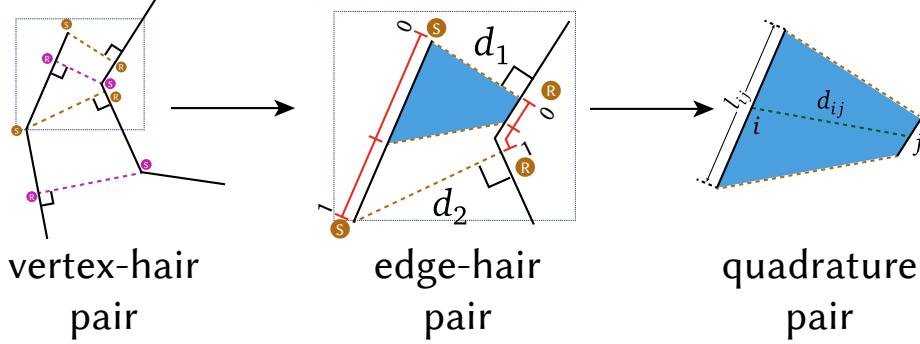


Figure 9.2: **An adaptive quadrature to evaluate the cohesion forces.** From left to right: potentially-connected vertex-hair pairs, connected edge-hair pairs, and quadrature pairs.

ure 9.2. In the following, recall that the cohesion cut-off distance d_{\max} is a function (8.12) of the cross-sectional liquid area.

Phase 1: Identify potentially-connected vertex-hair pairs For each hair (*source*) vertex, we consider all other hairs as candidate *receivers*. If the closest point on a candidate receiver is within distance $\frac{3}{2}d_{\max}(A_{\max})$, we say that the *source vertex* is *potentially connected* to the *receiving hair*. To be conservative in finding all potential pairs, we assume the maximum area A_{\max} that can be stored at the vertex, from (8.25). For each source vertex, we thus accumulate *all* those receivers that are proximate. This phase is accelerated by broad-phase culling that accounts for the threshold $\frac{3}{2}d_{\max}(A_{\max})$.

Phase 2: Identify connected edge-hair pairs For each vertex-hair pair, we independently examine the (one or two) edges incident on the source vertex to determine if this *source edge* is *connected* to the receiving hair.

Consider one such source edge. By definition, at least one of its endpoints is a

source vertex. We compute a threshold distance d_{\max} using the sum of the liquid areas at the source vertex and receiving point. If the source-receiver distance computed in Phase 1 is below d_{\max} , the source vertex is a *subdomain delimiter*. We likewise determine if the other endpoint is a delimiter.

If neither edge endpoint is a subdomain delimiter, we discard the edge; otherwise, we have identified a *source edge* that is *connected* to a *receiving hair*.

To establish a quadrature rule for this connected edge-hair pair, we first identify the quadrature subdomain on the source edge: if both edge endpoints are delimiters, the subdomain covers the entire edge, or barycentric interval $[0, 1]$; if one source vertex is a delimiter, it remains to find the other boundary of the subdomain. Suppose that the first vertex is a delimiter with distance $d_1 < d_{\max_1}$ and the second vertex is a non-delimiter with distance $d_2 > d_{\max_2}$. We set the barycentric coordinate of the second boundary at $d = d_{\max}$ by linearly interpolating from the edge endpoints via $\alpha = (d_{\max_1} - d_1)/(d_{\max_1} - d_1 - d_{\max_2} + d_2)$, yielding the subdomain with barycentric interval $[0, \alpha]$.

While the quadrature subdomain on the source edge covers at most one edge, the corresponding receiving subdomain in general covers some contiguous region of the centerline, not contained within an edge. To identify this region, we first map each source edge endpoint to its *receiving endpoint*, which lies on the centerline but generally not at a vertex. Treating the two receiving endpoints as the delimiters of the real interval $[0, 1]$, we map the barycentric interval $[0, \alpha]$ into corresponding

positions on the receiving centerline.

Once the source and receiving subdomains are identified, their corresponding midpoints are chosen as the evaluation points for our single-point quadrature method.

Force Computation. The quadrature subdomain established above allows us to discretize the cohesion force (8.5). Given a matched pair of subdomain midpoints indexed i and j , the adhesive force is computed as

$$\mathbf{f}_{s,ij} = -w_{\text{FF}} \frac{\partial dE_{s,ij}}{\partial d_{ij}} \hat{\mathbf{n}}_{ij} \quad (9.1)$$

where $\hat{\mathbf{n}}_{ij}$ is a unit vector pointing from point i to j . The factor $w_{\text{FF}} \in [0, 1]$ acts to smoothly disable the cohesion force if the quadrature pair lies beneath the bulk liquid surface; it is computed as $w_{\text{FF}} = 1 - \text{clamp}(\tilde{m}_f/\hat{m}_f, 0, 1)$ where \tilde{m}_f is the liquid particle mass interpolated from the background grid (refer to §9.3 for the computation of liquid particle mass distributed on background grid), and \hat{m}_f is a small positive threshold corresponding to average particle mass just beneath the free surface, e.g., $\hat{m}_f = \sqrt{2}\pi/6 * \rho_f(\Delta x)^3$, so that $\text{clamp}(\tilde{m}_f/\hat{m}_f, 0, 1)$ is a mollified submersion indicator function.

Let l_{ij} be the length of the source subdomain (see Figure 9.2). Discretizing (8.5),

the surface energy associated to the subdomain is given by

$$\begin{aligned} & dE_{s,ij}(d_{ij}, A_{L_{ij}}) \\ &= \omega_{ij} l_{ij} \sigma \left[l_A(d_{ij}, A_{L_{ij}}) + \sum_{k=1,2} \cos \theta_k l_{Sk}(d_{ij}, A_{L_{ij}}) \right], \end{aligned} \quad (9.2)$$

for the i -th point connected with neighbor point j . Since we often find a connection from one hair to another in both directions, we halve the force in such cases to avoid double-counting; this is expressed in the factor ω_{ij} which is set to 0.5 if a connection exists in the opposite direction and 1 otherwise. Thus we ensure the symmetry of forces by taking the average.

The length of the liquid-air interface is computed as

$$l_A(d_{ij}, A_{L_{ij}}) = R(d_{ij}, A_{L_{ij}}) \left[\pi - \sum_{k=1,2} (\theta_k + \alpha_k(d_{ij}, A_{L_{ij}})) \right] \quad (9.3)$$

and the length of the hair-air interface is

$$l_{Sk}(d_{ij}, A_{L_{ij}}) = 2r_k [\pi - \alpha_k(d_{ij}, A_{L_{ij}})], \quad k = 1, 2. \quad (9.4)$$

Precomputation for Cohesion Model Since $R(d_{ij}, A_{L_{ij}})$ and $\alpha_k(d_{ij}, A_{L_{ij}})$ are implicitly defined by d_{ij} and $A_{L_{ij}}$, we precompute $\partial dE_{s,ij}/\partial d$ and store its values into a table for efficiency. At run-time we can interpolate from the table to get $\partial dE_{s,ij}/\partial d$.

The first step of precomputation is building the table of $A_{L_{ij}}(d_{ij}, \alpha_k)$ for all combinations of a uniformly sampled set of values of d_{ij} and α_k , following (8.9). Since

$d_{ij} < d_{\max}$, where the latter is computed with (8.12), the range of possible d_{ij} values is bounded.

After we have the table of A_L , we uniformly discretize the range of resulting A_L values to compute a mesh grid for α_1 and α_2 : for each sample (d, A_L) on the mesh grid, we search for the closest A_L in the table with binary search and linearly interpolate to get the inverse mapping $\alpha(A_L, d)$. Then we solve (8.10) for each sample of (d, A_L) to get $\partial R(d, A_L)/\partial d$ and $\partial \alpha(d, A_L)/\partial d$.

Finally, using the derivatives of (9.3) and (9.4), we can find the gradient of (9.2) as

$$\nabla_d E_T(d, A_L) = \sigma \left[\frac{\partial l_A(d, A_L)}{\partial d} + \sum_{k=1,2} \cos \theta_k \frac{\partial l_{Sk}(d, A_L)}{\partial d} \right], \quad (9.5)$$

which gives us the table of $\nabla_d E_T(d, A_L)$ for each sample of (d, A_L) . ■

Time Integration

We integrate the elastic rods in time using the stable constraint-based solver of Tournier et al. [234]. This accounts for the internal forces (stretching, bending, twisting) and external forces of cohesion/repulsion, drag, and bulk pressure. (The details of bulk pressure force are presented in §9.3). During integration, the effective mass of hair above liquid is calculated as the actual mass of hair plus the mass of reduced liquid; while the effective mass of the hair underwater is just the actual mass of the hair. For stable integration, we modulate the surface liquid mass with w_{FF} (see *Force computation* above) to enable a smooth transition for the effective hair mass around

the liquid-air interface.

Preconditioning via Local Solves. We accelerate our solver using a novel preconditioner. When many hairs adhere to one another, a large number of constraints are created between individual hairs, and the resulting global interactions give rise to linear systems that are difficult to solve efficiently. Furthermore, even the construction of the stiffness matrix (jacobian matrix of the forces) \mathbf{H}_s can have a substantial performance overhead.

Observing that the adhesive force between hairs is the only type of constraint that causes off-diagonal blocks to appear in the stiffness matrix, we developed a preconditioning strategy for the conjugate gradient method that exploits this fact.

Before solving the large globally coupled system, we first build and solve the linear system corresponding to each hair strand independently of other strands, and ignoring adhesion. Observe that the stiffness matrix can be decomposed into a sum of individual hair stiffness matrices and a cohesion stiffness matrix, such that for any vector \mathbf{x} ,

$$\mathbf{H}_s \mathbf{x} = \sum_i^N \mathbf{H}_{s,i} \mathbf{S}_i \mathbf{x} + \mathbf{H}_{s,G} \mathbf{x}, \quad (9.6)$$

where N is the number of hairs, and S_i is a diagonal *selection matrix* whose j -th term is one if the j -th degree of freedom belongs to the i -th hair and zero otherwise. This effectively pulls out the entries of \mathbf{x} for a single hair into a shorter vector matching the dimensions of the stiffness matrix $\mathbf{H}_{s,i}$ corresponding to the i -th hair. The matrix $\mathbf{H}_{s,G}$ encodes inter-hair coupling components. Therefore in $\mathbf{H}_{s,i}$ only a few degree of

freedoms are involved, and these smaller systems can be solved in parallel for all the hairs.

Using the locally-solved velocity as an initial guess, we begin the preconditioned conjugate gradient method (PCG), using the local matrices for preconditioning. Each local matrix is small, banded, and remains fixed between PCG iterations; it can therefore be easily factored at the beginning of the PCG loop, and solved in parallel with a fast banded solver. To describe in detail the full algorithm for this process requires also to summarize the method and notation of Tournier et al. [234], as below.

Preconditioned Time Integration of Hair. For the dynamics of elastic rods, we extend the work of Tournier et al. [234] who proposed a stable and efficient constrained solver. We employ this solver since it is linear, stable at moderate time step, and is effective for constraints with a large range of stiffnesses.

In the following we first present our extended version of their method, which supports viscous drag and damping forces, and a novel preconditioner to boost the solver's efficiency for large systems with many inter-hair constraints.

We use φ to denote the constraints on positions (for example $\varphi_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\| - l_0$ for a spring with rest-length l_0), where \mathbf{q} refers to the configuration (position). $\dot{\varphi}$ for viscous constraints on velocity, for example $\varphi_v = \dot{\mathbf{q}} - \dot{\mathbf{q}}_0$ for the drag force whose target velocity is $\dot{\mathbf{q}}_0$; \mathbf{C} for the diagonal positional compliance matrix, which is the inverted stiffness matrix for positional constraints; and \mathbf{C}_v for the diagonal viscous compliance matrix, which stores the inverted viscous drag coefficients.

The energy of positional constraints is

$$E = \frac{1}{2} \boldsymbol{\varphi}^T \mathbf{C}^{-1} \boldsymbol{\varphi}, \quad (9.7)$$

and of viscous constraints is

$$E_v = \frac{1}{2} \boldsymbol{\varphi}_v^T \mathbf{C}_v^{-1} \boldsymbol{\varphi}_v. \quad (9.8)$$

We denote the states at the next time step with a subscript “+”. By linearization we have

$$\boldsymbol{\varphi}_+ \approx \boldsymbol{\varphi} + h \mathbf{J} \dot{\mathbf{q}} \quad (9.9)$$

where h is the time step and $\mathbf{J} \equiv \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{q}}$, and

$$\boldsymbol{\varphi}_{v,+} \approx \boldsymbol{\varphi}_v + h \mathbf{J}_{xv} \dot{\mathbf{q}}_+ + \mathbf{J}_v (\dot{\mathbf{q}}_+ - \dot{\mathbf{q}}) \quad (9.10)$$

where $\mathbf{J}_{xv} := \frac{\partial \boldsymbol{\varphi}_v}{\partial \mathbf{q}}$ and $\mathbf{J}_v \equiv \frac{\partial \boldsymbol{\varphi}_v}{\partial \dot{\mathbf{q}}}$.

Since the constraint forces are conservative, they arise as the negative gradient of the corresponding potentials, giving the form

$$\begin{aligned} \mathbf{f}_c &= -\frac{\partial E^T}{\partial \mathbf{q}} = -\frac{\partial \boldsymbol{\varphi}^T}{\partial \mathbf{q}} \frac{\partial E^T}{\partial \boldsymbol{\varphi}} = \mathbf{J}^T \boldsymbol{\lambda} \\ \mathbf{f}_{cv} &= -\frac{\partial E^T}{\partial \dot{\mathbf{q}}} = -\frac{\partial \boldsymbol{\varphi}_v^T}{\partial \dot{\mathbf{q}}} \frac{\partial E_v^T}{\partial \boldsymbol{\varphi}_v} = \mathbf{J}_v^T \boldsymbol{\lambda}_v \end{aligned} \quad (9.11)$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_v$ are the Lagrange multipliers of the positional and viscous constraints.

By the elastic constitutive law of the constraint forces [138, 212], $\mathbf{C}\boldsymbol{\lambda} = -\boldsymbol{\varphi}$, we

have

$$\boldsymbol{\lambda}_+ = -\mathbf{C}^{-1}(\boldsymbol{\varphi} + h\mathbf{J}\dot{\mathbf{q}}_+) \quad (9.12)$$

and

$$\boldsymbol{\lambda}_{v,+} = -\mathbf{C}_v^{-1} [\boldsymbol{\varphi}_v + h(\mathbf{J}_{xv} + h^{-1}\mathbf{J}_v)\dot{\mathbf{q}}_+] + \mathbf{C}_v^{-1}\mathbf{J}_v\dot{\mathbf{q}}. \quad (9.13)$$

Then with linearized implicit Euler [19] we have

$$(\mathbf{M}_s - h^2\mathbf{H}_s)\dot{\mathbf{q}}_+ = \mathbf{M}\dot{\mathbf{q}} + h [\mathbf{f}_e - \mathbf{J}^T\mathbf{C}^{-1}\boldsymbol{\varphi} - \mathbf{J}_v^T\mathbf{C}_v^{-1}(\boldsymbol{\varphi}_v - \mathbf{J}_v\dot{\mathbf{q}})] \quad (9.14)$$

where \mathbf{f}_e contains the constant external forces (gravity, etc.) and velocity impulses from fluid pressure (§9.3), and $\mathbf{H}_s = \frac{\partial(\mathbf{f}_e + \mathbf{f}_{cv})}{\partial \mathbf{q}} + \frac{\partial \mathbf{f}_{cv}}{\partial \dot{\mathbf{q}}}$ is the stiffness matrix. \mathbf{H}_s can be re-formulated by substitution into the form

$$\mathbf{H}_s = - \underbrace{[\mathbf{J}^T\mathbf{C}^{-1}\mathbf{J} + \mathbf{J}_v^T\mathbf{C}_v^{-1}(\mathbf{J}_{xv} + h^{-1}\mathbf{J}_v)]}_{\text{material stiffness}} + \underbrace{\frac{\partial \mathbf{J}^T}{\partial \mathbf{q}} : \boldsymbol{\lambda} + \left(\frac{\partial \mathbf{J}_v^T}{\partial \mathbf{q}} + h^{-1} \frac{\partial \mathbf{J}_v^T}{\partial \dot{\mathbf{q}}} \right) : \boldsymbol{\lambda}_v}_{\text{geometric stiffness}} \quad (9.15)$$

where the notation “:” denotes the tensor product in the dimension of the number of constraints. The *material stiffness* terms represent the change of magnitude of constraint forces, while the *geometric stiffness* terms encode the transverse variation in force direction.

During each time step, we first compute \mathbf{H}_s with (9.15) using $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_v$ computed from previous steps. We then solve (9.14). After $\dot{\mathbf{q}}_+$ is obtained, we update the Lagrange multipliers with (9.12) and (9.13).

Remark: For the adhesive/repulsive force of (9.1), we divide its intensity by the distance of the point-point pair to get the inverse compliance, where the material stiffness is then computed as

$$\mathbf{J}_{ij}^T \mathbf{C}_{ij}^{-1} \mathbf{J}_{ij} = \hat{\mathbf{n}}_{ij} \frac{\|\mathbf{f}_{s,ij}(d_{ij})\|}{d_{ij} - r_i - r_j} \hat{\mathbf{n}}_{ij}^T, \quad (9.16)$$

and similarly for the geometric stiffness.

In (9.6), we partition the stiffness matrix into individual hairs and a cohesive term. Since each $\mathbf{H}_{s,i}$ only a few degree of freedoms are involved, and these smaller systems can be solved in parallel for all the hairs. Using the locally-solved velocity as an initial guess, we begin the preconditioned conjugate gradient method (PCG), using the local matrices $\mathbf{M}_i - h^2 \mathbf{H}_{s,i}$ for preconditioning. Each local matrix is small, banded, and remains fixed between PCG iterations; it can therefore be easily factored at the beginning of the PCG loop, and solved in parallel with a fast banded solver (see Pseudocode 1).

In the pseudocode, we have denoted the left hand side of (9.14) as \mathbf{A} , the left hand side of (9.14) considering only the constraints inside the i -th hair as \mathbf{A}_i^* , the right hand side of (9.14) as \mathbf{b} , and the right hand side of (9.14) considering only the constraints inside the i -th hair as \mathbf{b}_i^* . We use the notation $[\cdot]$ for the assembly of the local vectors into a global vector. Finally, we have defined the updated generalized velocities for a particular hair i via $\dot{\mathbf{q}}_{+,i} := \mathbf{S}_i \dot{\mathbf{q}}_+$.

We compare our PCG with other techniques (refer to §10 and Figure 10.6), namely,

Algorithm 1 Locally-Preconditioned Solve

```
1: for all  $i$  do in parallel
2:   Solve  $\mathbf{A}_i^* \dot{\mathbf{q}}_{+,i} = \mathbf{b}_i^*$ .
3: end for
4:  $\mathbf{r} \leftarrow \mathbf{b} - \mathbf{A}[\dot{\mathbf{q}}_{+,i}]$ 
5: for all  $i$  do in parallel
6:   Solve  $\mathbf{A}_i^* \mathbf{z}_i = \mathbf{S}_i \mathbf{r}$ .
7: end for
8:  $\mathbf{z} \leftarrow [\mathbf{z}_i]$ 
9:  $\mathbf{p} \leftarrow \mathbf{z}$ 
10: while  $\mathbf{r}^T \mathbf{r} > \epsilon$  do
11:    $\mathbf{w} \leftarrow \mathbf{M} \dot{\mathbf{q}}_+ - h^2 (\sum_i^N \mathbf{H}_{s,i} \mathbf{S}_i \dot{\mathbf{q}}_+ + \mathbf{H}_{s,G} \dot{\mathbf{q}}_+)$ 
12:    $\gamma \leftarrow \mathbf{r}^T \mathbf{z}$ 
13:    $\alpha \leftarrow \frac{\mathbf{r}^T \mathbf{z}}{\mathbf{p}^T \mathbf{w}}$ 
14:    $\dot{\mathbf{q}}_+ \leftarrow \dot{\mathbf{q}}_+ + \alpha \mathbf{p}$ 
15:    $\mathbf{r} \leftarrow \mathbf{r} - \alpha \mathbf{w}$ 
16:   for all  $i$  do in parallel
17:     Solve  $\mathbf{A}_i^* \mathbf{z}_i = \mathbf{S}_i \mathbf{r}$ .
18:   end for
19:    $\mathbf{z} \leftarrow [\mathbf{z}_i]$ 
20:    $\beta \leftarrow \frac{\mathbf{z}^T \mathbf{r}}{\gamma}$ 
21:    $\mathbf{p} \leftarrow \mathbf{z} + \beta \mathbf{p}$ 
22: end while
```

a conjugate gradient solver preconditioned with the inversed diagonal terms of matrix K and initialized with the locally-solved velocity (denoted as *Initialized DPCG*), a regular conjugate gradient solver with the last velocity as an initial guess (denoted as *CG*), and the sparse LDLT solver in the Eigen library [94] (denoted as *LDLT*). We show that our method is more efficient than the others in terms of both iteration counts and timing.

9.2 Surface Liquid Simulation on Strands

We compute the flow of liquid clinging to the hair surface using the surface liquid model developed in §8.2. By assuming that the thin liquid is always rotationally-

symmetric about the hair centerline, the state is captured by two scalar fields, liquid depth $h_\tau(x)$ and velocity $u_\tau(x)$, as a function of hair centerline arclength parameter x .

Since the hair centerline accelerates over time, these local surface liquid coordinates live on a non-inertial reference frame, giving rise to inertial forces acting on the surface liquid system.

The inertial force at a position x of the hair appears as an additional acceleration, $-\mathbf{t}^T(x)\mathbf{a}(x)$, on the right hand side of (8.15). The force opposes the acceleration of the hair $\mathbf{a}(x)$ as projected onto the centerline unit tangent $\mathbf{t}(x)$.

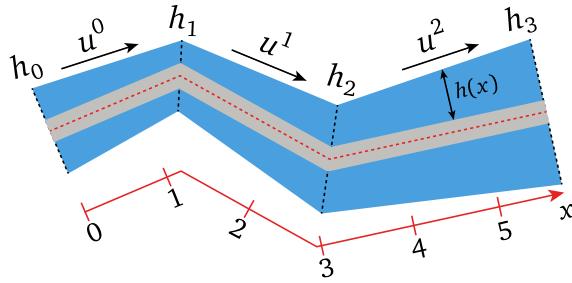


Figure 9.3: Discretized surface flow along hair.

Discretization. Consider a single hair represented by M vertices. Adopting the notation of Bergou et al. [26], we distinguish between primal quantities, associated with vertices and decorated with lower indices, from dual quantities, associated with edges and decorated with upper indices. We discretize $u_\tau(x)$ using a piecewise constant approximation over the centerline edges $1, \dots, M - 1$ with edge-based coefficients $\{u_\tau^1, u_\tau^2, \dots, u_\tau^{M-1}\}$. We discretize $h_\tau(x)$ via a piecewise linear approximation over the centerline edges with vertex-based coefficients $\{h_{\tau,1}, h_{\tau,2}, \dots, h_{\tau,M}\}$ (see Fig. 9.3). The vertex-based length $l_{\tau,i}$ for hair vertex i is computed as the averaged edge length

of its neighbor edges (a.k.a. $l_{\tau,i} = (l_{\tau}^{i-1} + l_{\tau}^i)/2$, where l_{τ}^i is the length of edge i).

Velocity Update. Denoting the time step by a superscript (k) , we solve the momentum equation of (8.15) with a semi-Lagrangian technique [224]. We backtrace the velocity *along the centerline*, sampling the velocity upstream

$$\tilde{u}^{\tau,j,(k)} = u_{\tau}(x_{\tau}^j - u_{\tau}^{j,(k)}h), \quad (9.17)$$

where x_{τ}^j is the arclength parameter for the midpoint of edge j , and h is the time step.

We evaluate a surface-tension induced vertex-based pressure $p_i = \sigma(Lh)_i$, where L is the univariate second finite difference operator and $\sigma(Lh)_i$ accounts for the surface tension energetic preference for linearly varying height fields [239].

The gravitational acceleration *per edge* along the tangential direction of edge i is denoted as $\mathbf{g} \cdot \mathbf{t}_i$. The velocity is then explicitly updated as

$$u_{\tau}^{j,(k+1)} = \tilde{u}_{\tau}^{j,(k)} - \frac{h}{\rho_f}(\text{grad } p)^{j,(k)} + h(\mathbf{g} \cdot \mathbf{t}) \quad (9.18)$$

where grad is the staggered grid gradient operator from vertex- to edge-based quantities.

Evolving Liquid Depth. Next, we solve the conservation law of mass (8.16). First, we compute the cross sectional area $A_{\tau,j} = \pi((h_{\tau,j} + r_j)^2 - r_j^2)$ on each hair vertex j ,

where r_j is the constant hair radius at the vertex j . Then, (8.16) is discretized into

$$\frac{A_{\tau,j}^{(k+1)} - A_{\tau,j}^{(k)}}{h} + \langle \text{grad } u_\tau \rangle_j^{k+1} A_{\tau,j}^{k+1} + \langle u_\tau^{i,(k+1)} (\text{grad } A_\tau)^{i,(k+1)} \rangle_j^i = 0, \quad (9.19)$$

where $(\text{grad } u_\tau)_j$ is the staggered grid gradient of the edge-based quantity u_τ , producing vertex-based gradient value. $(\text{grad } A_\tau)^i$ is the staggered grid gradient of vertex-based quantity A , producing edge-based gradient value. The operator $\langle \cdot \rangle_j^i$ converts edge-based quantities into vertex-based quantities. In particular, if the edge i and $i+1$ shares the vertex j , then an arbitrary edge-based quantity v^i is converted into a vertex-based quantity defined at the vertex j , using a weighted average with the edge lengths l_τ^i and l_τ^{i+1} ,

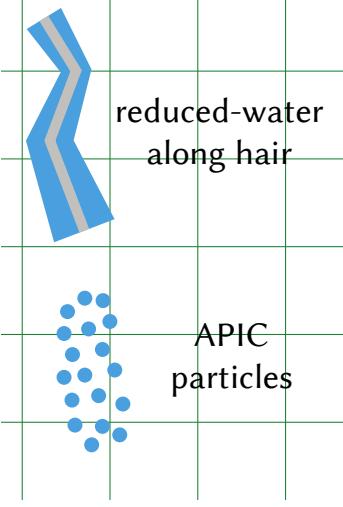
$$\langle v \rangle_j^i = \frac{1}{l_{\tau,i} + l_{\tau,i+1}} (v^i l_\tau^i + v^{i+1} l_\tau^{i+1}).$$

We apply absorbing boundary conditions for free hair tips so that liquid flows freely "off" (i.e., to be converted to dripping particles). For hair that stems from a solid object, we apply reflecting boundary conditions at the solid.

9.3 Bulk Fluid Simulation

To simulate bulk volumes of liquid not attached to the hair itself, we adopt the affine particle-in-cell (APIC) method [122], which offers reduced velocity dissipation and good preservation of linear and angular momentum.

At each step the fluid mass and momentum are transferred back and forth between a background grid and a set of fluid particles, allowing forces to be computed on the Eulerian grid and advection to be done in a Lagrangian fashion. This facilitates coupling between our system's components (recall Fig. 9.1) for both drag and pressure. APIC's pervasive use of fluid particles also conveniently enables us to treat the volume of hair surface liquid as a secondary *continuous* set of particles that likewise exchange mass and momentum with the grid, when determining the influence of pressure.



Transfer: Particles to Grid. Following the APIC scheme, we first compute mass and momentum on grid faces in the x , y , and z directions; these quantities must be transferred from the nearby Lagrangian representations at each step. Following the standard APIC notation, we have

$$m_{g,i}^n = \sum_j m_{p,j} w_{i,j}^n \quad \text{and} \quad (9.20)$$

$$m_{g,i}^n u_{f,g,i}^n = \sum_j m_{p,j} w_{i,j}^n [e_i^T \mathbf{u}_{f,p,j}^n + (\mathbf{c}_{p,j,i}^n)^T (\mathbf{x}_{g,i} - \mathbf{x}_{p,j}^n)],$$

where the first equation accumulates mass, and the second accumulates momentum. The superscript n is the index of the time step, p indicates a particle-based variable, g indicates a grid-face-based variable, $\mathbf{u}_{f,p,j}^n$ is the j -th liquid particle velocity, $\mathbf{x}_{g,i}$ is the position of i -th grid face center or particle depending on the subscript, and e_i is

(a unit vector) of the direction of the grid face. We use $w_{i,j}^n$ to denote the trilinear interpolation weight that transfers the information on particle j to grid face i (or vice versa), and $c_{p,j,i}^n$ is the affine strain rate of j -th particle in the direction of i -th grid face. These equations precisely follow Jiang et al. [122].

Drag Force. The motion of underwater hair is affected by fluid drag forces, and applied numerically during time integration of the hair §9.1, in which we define the effective mass of hair underwater as the actual mass of hair so that the drag force acts on the correct amount of mass. Otherwise, if the surface liquid mass is also counted underwater during integration, the drag force will be decreased. Besides, since the bulk liquid velocity has been extrapolated for several layers into the air, we modulate the drag force applied with $1 - w_{FF}$ to prevent the hairs from being dragged by the air.

Naturally, Newton’s third law dictates that an opposing force must be applied to the liquid, whose momentum is now stored in the grid. Our approach will be to rasterize the force that the hair exchanges with the fluid due to drag. We start with determining the surface liquid force per grid face. Viewing the hair strands as *continuous* particle sets, we compute the relevant force by integrating along the portion of each hair strand within the given cell. We index hair segments with t , the averaged drag force on a particular fluid grid face is found by summing over all

segments t using

$$\mathbf{f}_{\text{d,g},i}^n = \left[\sum_j \int_{l_j} w_{i,j}^n(s) ds \right]^{-1} \sum_j \left[\mathbf{e}_i^T \int_{l_j} w_{i,j}^n(s) \mathbf{f}_{\text{d,e},j}^n(s) ds \right],$$

where l_j indicates the integration path along the hair segment j inside the cell, and $w_{i,j}^n(s)$ is the interpolation weight from a point s on the j -th hair segment to the grid face i , and $\mathbf{f}_{\text{d,e},j}^n(s)$ is the drag force evaluated for a point s on the j -th hair segment. Each hair segment is first clipped against the relevant (staggered) cell, and the integration is approximated with two-point Gauss quadrature on each remaining segment or portion thereof.

The accumulated force is applied onto the velocity of the grid face as

$$u_{\text{f,g},i}^{n+1} = u_{\text{f,g},i}^n + h\rho_f^{-1}(\Delta x)^{-3} \mathbf{f}_{\text{d,g},i}^n.$$

where h is the time step.

Remark. The drag force presented in this section is simple, and is similar to the drag force used in CFD-DEM method [264], where both their and our drag force first homogenize the drag computed from discrete elements and then is applied to the continuum phase. However, the drag force presented above still lacks of a principled justification, as well as a detailed form to compute $\mathbf{f}_{\text{d,e},j}^n$. In §13.5 of **Part III**, we will deliver a more principled form of the drag force applied to liquid, accompanied with the detailed form of the drag applied on a single strand element, i.e., $\mathbf{f}_{\text{d,e},j}^n$.

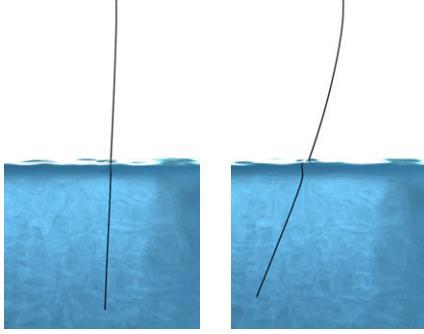


Figure 9.4: A hair suspended by its tip moving left to right through liquid. The drag force is disabled (left) or enabled (right).

Transfer: Reduced Liquid to Grid. Having applied the drag force to the bulk fluid, we now transfer the surface liquid mass and momentum onto the grid as well, before proceeding to make the full fluid volume divergence-free. We compute the necessary surface liquid masses per face,

$$m_{g,i}^n = \sum_j m_{\tau,j} \int_{l_j} w_{i,j}^n(s) ds.$$

where $m_{\tau,j}$ is the liquid mass on hair segment j , and then we simply add them to the appropriate per-face masses accumulated in (9.20).

With the same notation and approach, the momentum contributions from the surface liquid are

$$T_{g,i}^n = \sum_j m_{\tau,j} \left[\mathbf{e}_i^T \int_{l_j} w_{i,j}^n(s) \mathbf{u}_j^n(s) ds + \int_{l_j} (\mathbf{c}_{v,j,i}^n(s))^T (\mathbf{x}_{g,i} - \mathbf{x}_{v,j}^n(s)) ds \right], \quad (9.21)$$

where $\mathbf{c}_{v,j,i}^n(s)$ is a vector analogous to $\mathbf{c}_{p,j,i}^n$ in (9.20) for preserving the affine velocity field. The position and velocity of a point on the segment j parameterized by s are denoted by $\mathbf{x}_j^n(s)$ and $\mathbf{u}_j^n(s)$, respectively. To find $\mathbf{x}_j^n(s)$ and $\mathbf{u}_j^n(s)$ at arbitrary points along the segment we linearly interpolate from the segment's endpoints. Note that

$\mathbf{u}_j^n(s)$ should be understood to be as the sum of the extrinsic (hair velocity $\mathbf{u}_{s,j}^n$) and intrinsic ($u_{\tau,j}^n$) components of the surface liquid velocity, computed as $\mathbf{u}_{s,j}^n + u_{\tau,j}^n \mathbf{t}_j$.

We add these surface liquid momentum contributions to the corresponding grid face momenta accumulated in (9.20).

Pressure Projection. Given the grid velocities $u_{f,i}^{n+1}$ after applying drag, we perform the pressure projection step of the fluid simulation which yields the divergence-free velocity field $\tilde{u}_{f,i}^{n+1}$.

Transfer: Grid to Particles. Next, we transfer velocities from grid faces back to particles using the standard APIC rule

$$\mathbf{u}_{p,j}^{n+1} = \sum_i w_{i,j}^n \tilde{u}_{f,i}^{n+1} \mathbf{e}_i.$$

Transfer: Grid to Intrinsic Surface Liquid Velocity on Hairs. The pressure gradient from the grid should also be applied to the surface liquid on hairs as an impulse. We first sample the interpolated pressure gradient ∇p from the grid at the center of mass of the hair segment. We then use it to update the velocity of the surface liquid, considering only the component along the direction of hair segment i , using the simple update rule

$$u'_\tau = u_\tau - h\rho_f \mathbf{t}_i^T \nabla p.$$

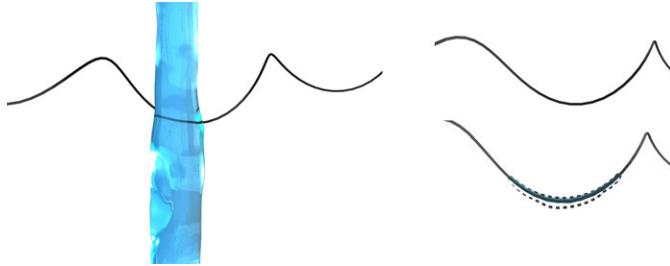


Figure 9.5: Capture of liquid (highlighted by dashed lines) from a flowing stream. **Left:** Water pours over a single curly hair. **Right:** A hair with capture turned off (**above**) and on (**below**).

Transfer: Grid to Hair Vertices. Lastly, the pressure force should be applied to the hair itself. We apply it on the next time integration step of the hair, so we retain the pressure field for that purpose. While this implies a weak/staggered coupling, the time integration of this force is nevertheless applied implicitly in the sense that the pressure gradient in the normal direction is sampled at the unknown predicted position of hair vertices. This is done using

$$\mathbf{f}_{\nabla p}^n = \frac{m_s}{\rho_s} \nabla p.$$

where m_s is the mass of a hair vertex including its surface liquid, and ρ_s is the hair mass density.

9.4 Surface Liquid Capture and Dripping

Another critical element of our model is the conservative exchange of the *volume* and associated *momentum* of liquid between the reduced and bulk fluid representations. We carry this out using new mechanisms for capture and dripping, described below. Our approach yields particles of varying sizes, since the amount of liquid entering and leaving a hair strand can be quite small; we use a particle-merging scheme similar to

Ando et al. [10] to allow such particles to coalesce into larger ones. If N_p is the target average particle number per cell, we use a default particle radius of $\sqrt{2}\Delta x/N_p$, and we delete any particles that fall below a radius of 1×10^{-7} cm in our implementation.

In each time step we first capture liquid from nearby particles and transform it into appropriate surface liquid height fields on hair segments. We implement this in the following readily parallelizable fashion:

1. For each hair segment we find its neighboring particles within the maximal radius of attachable liquid droplets, r_{\max} (see (8.25)), and pair the segment with each such particle. With each pair i we store the arclength coordinate of the point $x_{c,i}$ on the segment which is closest to the particle.
2. For each hair vertex we gather the liquid volume and momentum from the particle-segment pairs associated with its two incident hair segments. The new volume of liquid $V_{\tau,v,j}^{n+1}$ on vertex j and the net fluid velocity $\mathbf{u}_{v,j}^{n+1}$ are computed as

$$\begin{aligned} V_{\tau,v,j}^{n+1} = & \min \left(V_{\tau,\max}, V_{\tau,v,j}^n + \sum_{i \in \text{pairs}} V_{f,p,i}^{n+1} w(\|x_{c,i} - x_j\|) \right), \\ (\rho_f V_{\tau,v,j}^{n+1} + \rho_s V_{s,v,j}) \mathbf{u}_{v,j}^{n+1} = & \rho_s V_{s,v,j} \mathbf{u}_{s,v,j}^n + \rho_f V_{\tau,v,j}^n (\mathbf{u}_{s,v,j}^n + u_{\tau,v,j}^n \mathbf{t}_{v,j}) \\ & + \rho_f \sum_{i \in \text{pairs}} V_{p,i}^{n+1} \mathbf{u}_{f,p,i}^{n+1} \end{aligned} \quad (9.22)$$

where $w(\|x_{c,i} - x_j\|)$ is the linear interpolation kernel based on distance between the point $x_{c,i}$ and the vertex x_j , ρ_s is the density of hair, ρ_f is the density of

liquid, $V_{s,v,j}$ is the volume occupied by the j -th hair vertex computed from its radius and Voronoi length, $\mathbf{u}_{s,v,j}^n$ is the velocity of hair vertex j (and by association the *extrinsic* surface liquid velocity), $u_{\tau,v,j}^n$ is the intrinsic surface liquid velocity of liquid on the hair, and $\mathbf{t}_{v,j}$ is the average tangent vector at hair vertex j . The corresponding amount of liquid volume transferred to the hair is deducted from the particles in the relevant particle-segment pairs.

3. After gathering, we project the net fluid velocity $\mathbf{u}_{v,j}^{n+1}$ onto the tangent $\mathbf{t}_{v,j}$ and use it to replace the intrinsic surface liquid velocity; we also update the hair's total mass and the normal velocity of the hair:

$$\begin{aligned} m_{v,j}^{n+1} &= m_{v,j}^n + \rho_f V_{\tau,v,j}^{n+1}, \\ m_{v,j}^{n+1} \mathbf{u}_{s,v,j}^{n+1} &= m_{v,j}^n \mathbf{u}_{s,v,j}^n + \rho_f V_{\tau,v,j}^{n+1} (\mathbf{I}_3 - \mathbf{t}_{v,j} \mathbf{t}_{v,j}^T) \mathbf{u}_{v,j}^{n+1}. \end{aligned} \quad (9.23)$$

After the capture process, we allow liquid to leave the hair by carrying out a complementary dripping process as follows:

1. For each cell of the grid we gather the hair vertices inside it, and sum up their volume of liquid. If the hair vertex is on a hair tip, we gather the amount of liquid flowing out of the hair.
2. We determine if the total volume is larger than the specified threshold (§8.2) dictated by r_{\max} .
3. We convert the volume difference between the total volume and the threshold into liquid particles of $1/4$ the default particle size, and release them into the grid. We assign them (total) velocity of the liquid on the hair. Their position

on the hair is chosen uniformly at random from within the (one-dimensional) Voronoi region of the associated vertex.

4. We calculate the ratio of the new volume of liquid on the hair to its past volume; for each hair vertex in the cell we rescale the height of the liquid and the corresponding mass accordingly.

With this approach, the hair easily and conservatively exchanges its associated liquid mass and momentum with the surrounding bulk flow, as seen in Figure 9.5.

Chapter 10

Simulated Results of Wet Hairs

Our results can be roughly divided into two categories: first, a set of didactic examples intended to validate individual components of our system in relative isolation, and second, a set of several more general examples of water-hair interaction demonstrating the synthesis of the complete system.

10.1 Validation Examples

Surface Water Flow. We demonstrate our hair surface liquid model by assigning an even distribution of water to a length of curly hair in Figure 8.10, left. Upon releasing the hair, the water flows toward the lowest point and collects, causing it to sag more than the corresponding hair without flow.

Dripping. Taking the scenario above and enabling dripping causes the collected liquid to pour off as APIC droplets in Figure 8.10, right. This reduces the hair effective mass so that its internal forces more easily overcome gravity and pull the hair upwards again. Performing a similar test on a vertically suspended straight hair shows that water can also pour smoothly (and conservatively) off the hair tips (Figure 8.7).

Inertial Forces. The role of inertial forces cannot be ignored if we wish to have plausible flows along hair. Figure 8.6 presents a straight hair swung in a circle at a sufficient rate to cause the liquid to flow towards the tip; with inertial forces disabled the reduced-water remains stationary despite its changing reference frame.

Momentum Transport. In Figure 8.8 we use a slanted “hair pendulum” to compare the effects of rescaling momentum vs. proper momentum transport for updating the momentum of the hair to reflect the movement of reduced-water sliding along it. Only our approach captures the expected smooth oscillation behavior without artifacts.

Liquid Capture. Figure 9.5 shows a stream of APIC liquid falling past and around a hair; as it does so, water particles are captured and transformed into a thin layer of reduced-water on the hair surface.

Cohesion and Coalescence. Our cohesion model is demonstrated in Figure 8.4, in which two wet hairs are placed in close proximity and pulled apart from the top. Without cohesion the hairs separate instantly; with cohesion, they remain connected until gravity and the increasing separation distance eventually breaks the connection. In Figure 7.1 we show this same cohesion effect in a more general scenario with many hairs in a row pulled out of water. The structure of coalescence in the clumping hairs provides a good qualitative match with a comparable laboratory experiment [33]. Since the cohesion force is stiff, it can be difficult to solve with a regular integrator. In Figure 10.6 left, we compare the convergence behavior of different iterative solvers

on the linear system. We show that even for two hairs, our preconditioner (yellow curve) has a convergence rate that is several orders of magnitude faster than the other schemes we tried, namely the locally-presolved diagonally-preconditioned CG (blue curve) or the regular CG solver (green curve, mostly hidden by the blue curve).

Drag Force. Pulling a single hair through a pool of water illustrates the influence of our drag force: the fluid bends the hair in opposition to its movement (Figure 9.4). Without drag, the hair hangs vertically, oblivious to the presence of the liquid pool.

10.2 Large-Scale Examples

Wringing Out Hair. We can demonstrate several features of our method in action by pouring water onto a collection of horizontally suspended hairs to wet them, and then twisting them to effectively wring the water out (Figure 10.1). When the hairs are twisted, they become more closely packed. The attendant reduction in carrying capacity (8.25) leads to the release of bulk liquid from the hairs.

Hair Whipping. Figure 10.2 demonstrates cohesion between hairs and solid objects with different surface curvatures, for example, a straight wall and a ball. After the wet hair strands are pulled away from the vertical wall, some of the strands slide on the surface due to cohesion forces between the hair strands and the surface. Similarly, many strands of hair adhere to the ball throughout their relative motion.



Figure 10.1: Wringing out water from drenched hair.



Figure 10.2: Wet hairs are whipped onto a wall and then pulled off. Cohesion results between hairs and objects in the scene.

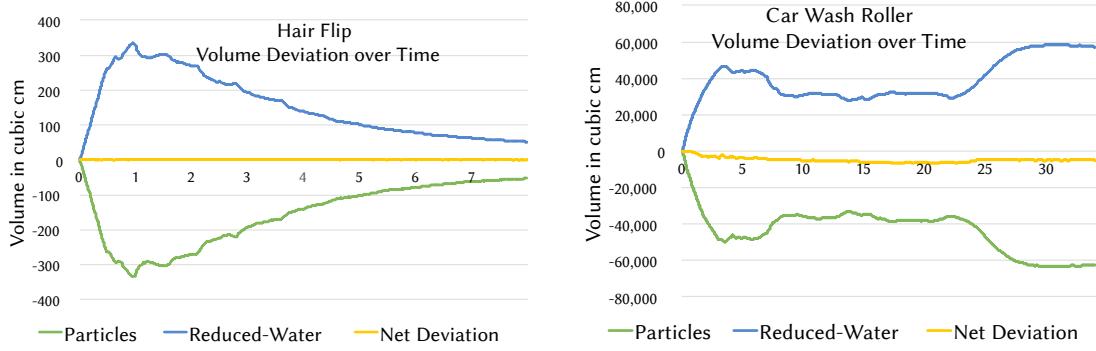


Figure 10.3: **Volume conservation** is demonstrated by plotting the deviation of fluid volume in particles (green), reduced-water (blue), and their total (yellow). **Left:** Hair flip example. **Right:** Car wash example.

Shaking “Dog”. As many pet owners can attest, dogs and other mammals often shake themselves in an alternating rotational motion to rapidly remove excess water from their bodies (and occasionally splash their owners). We simulate this process with a shaggy-haired cylindrical dog, which we pour water over and rotate rapidly side to side (Figure 10.7). The initial pouring causes the hair to become heavier and matted at the points of contact. Subsequent shaking helps to eject liquid off of the cylinder. Despite significant acceleration provided by our preconditioning strategy, time integration of the hair dynamics was consistently the most costly component, due to the large number of hairs and the complexity of their interactions.

Wet Hair Flip. Another familiar wet hair effect occurs when a head of long hair is pulled out of water and rapidly whipped forward (Figure 10.7-top).

In Figure 10.6 right, we compare among different solvers for different sizes of the scene. We show that even as the number of hairs becomes larger, our PCG solves the linear equations several orders of magnitude more efficiently than other schemes. For

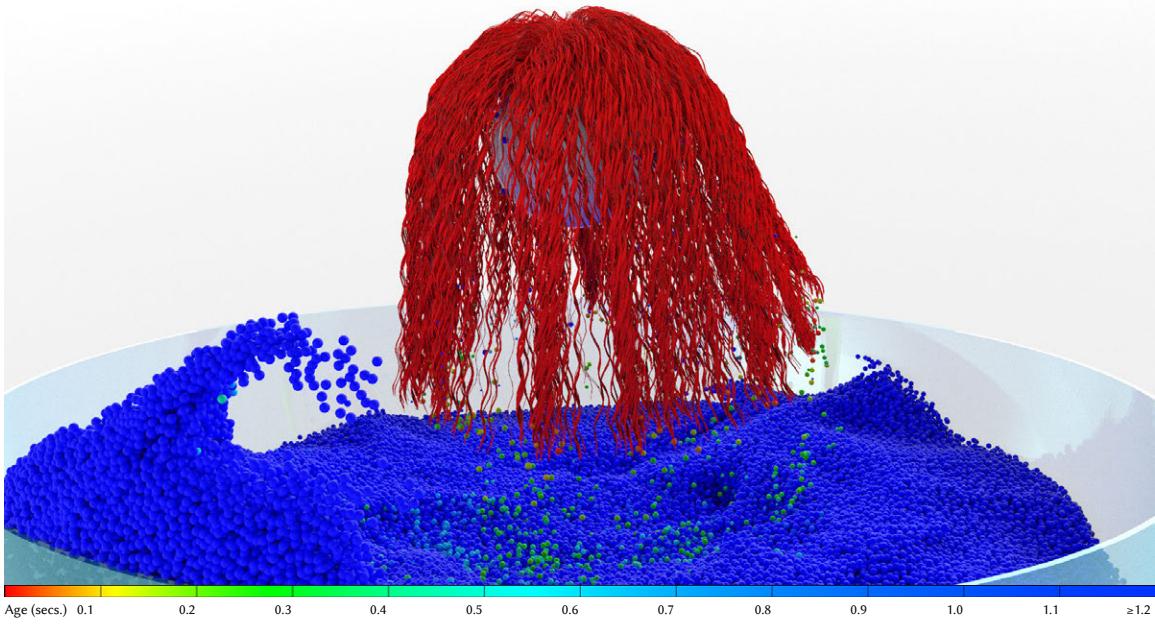


Figure 10.4: Color-coded visualization of the age of fluid particles as they transition from reduced liquid flow to APIC particles.

example, when there are 8K hairs, our method (yellow bars) is two orders faster than the locally-presolved diagonally-preconditioned CG (blue bars), three orders faster than a regular CG solver (green bars), and four orders faster than Eigen’s sparse LDLT solver (red bars). We only record comparisons up to 0.8 simulated seconds since performing one linear solve with LDLT in the 8K example can exceed half a day.

We also measure the total volume of reduced-hair liquid and bulk liquid over the course of the simulation. The volume of the bulk liquid is calculated as the sum of the spherical volumes of liquid associated with each APIC particle, according to each particle’s radius. The volume of the hair-surface flow is calculated as the sum of the volume of the annular cylinders of liquid around the hair segments. Figure 10.3, left, shows that the net increase in hair-reduced water (blue curve) was offset by the net



Figure 10.5: Water is poured over four fur mats of the same hair density but of differing hair lengths, revealing increased clumping effects. From left to right: 2cm, 3cm, 4cm, and 5cm.

decrease in bulk liquid (green curve), yielding remarkable conservation of total liquid volume (yellow curve).

In Figure 10.4 we visualize the age of particles for the frame in Figure 10.7-top to give a clearer view of how fluids drip from the hairs and change their discretization from reduced flow to particles. For a clearer view of the particles we render them with $0.7\times$ of their actual radius. The reduced flow on each hair segment is rendered as a red cylinder with the actual radius.

Car Wash Roller. As a particularly grueling test of our method with regards to capture and release of fluid by strands, we dunk a car wash-style roller brush into a large volume of fluid and begin spinning it at a progressively faster rate. The resulting centrifugal forces whip water out of the tips at high speed, while the spinning bristles slap back into the bulk volume and collect yet more fluid. Nonetheless, we again observe (Figure 10.3, right) that the net increase in reduced-water over time (blue curve) was closely offset by the net decrease in bulk liquid (green curve).

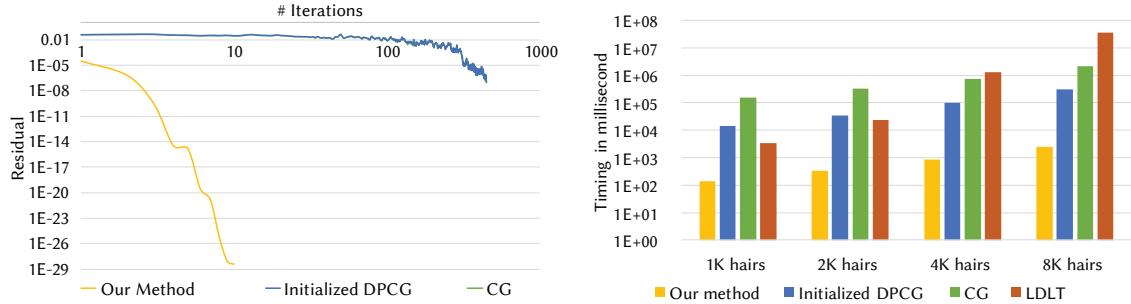


Figure 10.6: **Comparison between different solvers for the hair dynamics.** **Left:** Convergence rate at 0.25 simulated seconds in *Cohesion and coalescence* (Figure 8.4). **Right:** Timing for performing one linear solve, averaged from the beginning to 0.8 simulated seconds in *Wet hair flip* (Figure 10.7-top) .

Water on Mat of Fur. In the Figure 8.1 example we pour water onto and over a sloped stationary mat of fur. The initially raised dry hair is quickly weighed down and flattened by both the flowing and captured water, clumping effects form, and reduced-water drips off the hair tips. Figure 10.5 demonstrates the changing clumping behavior as we increase the hair length of the simulated fur.

10.3 Performance Numbers

We collected extensive timing data to evaluate the computational cost of our method and its various components on our large-scale examples, for which a detailed breakdown is included in Table 11.1. For the *Car Wash Roller*, its statistics are collected on a workstation with two sockets of Xeon E5-2620 v3 CPUs running at 2.4GHz each of which has six cores and twelve threads. For the other examples, their statistics are collected on a workstation with two sockets of Xeon E5-2687W CPUs running at 3.1GHz each of which has eight cores and sixteen threads. These examples employed fluid grid resolutions ranging from 64^3 to 128^3 , and hair counts ranging from 4K,

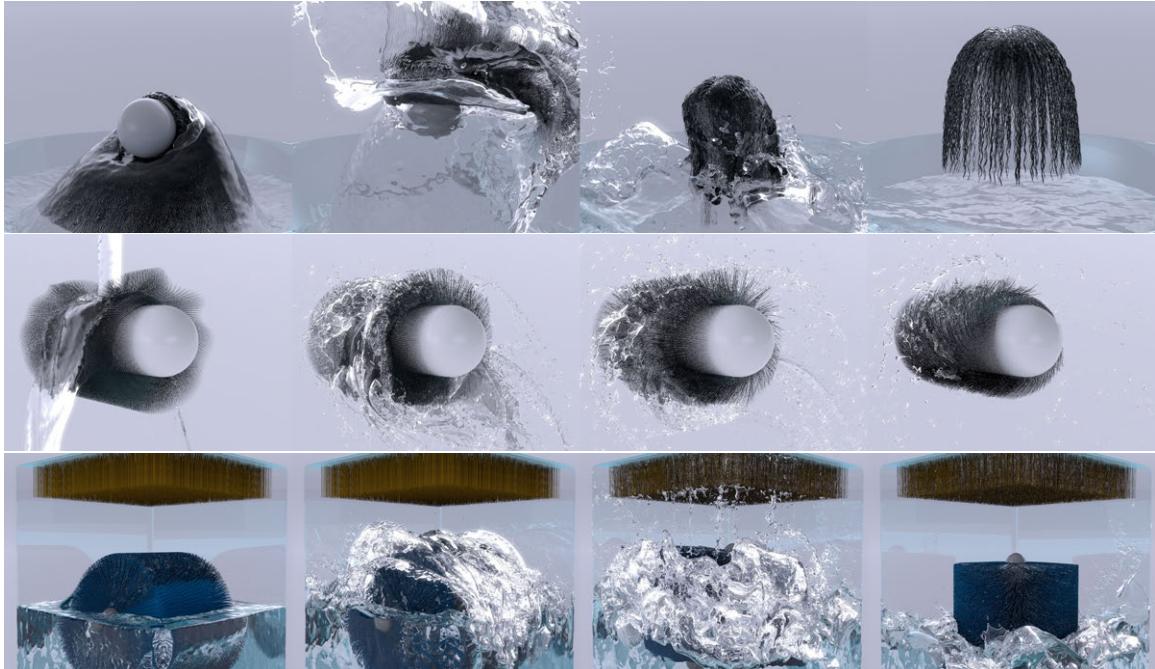


Figure 10.7: Large examples of wet hair simulation. **Top:** A ball is flipping its hairs, where a substantial volume of water is carried out along with the hair and is splashed into the air. **Middle:** A cylindrical pseudo-dog has water poured on its back; it shakes rapidly side-to-side to remove the excess water at high speed. **Bottom:** A car wash roller brush spins at increasing speeds through a pool, capturing water, splashing sheets into the air, and causing the bulk volume to swirl as well.

for the wringing and fur examples, to $32K$ for the dog, and car wash scenarios. The average cost per individual time step varied between 3 and 57 seconds, while the total simulation time varied between 3.5 and 57.8 hours. The simulated time span of the examples varied between 4 seconds for the *water on mat of fur* and 30 seconds for the *car wash roller*.

Chapter 11

Future Works on Wet Hairs

We have adopted or adapted a variety of physical models from prior work which are generally well-supported by experimental or analytical evidence. However, they are typically based on situations where competing factors have been isolated to focus on a single effect. In our pursuit of a very general model for animation it is possible or even likely that we have pushed these models far beyond their range of strict applicability (e.g. from single hairs to ten of thousands, or vice versa). Our calibration of the models in the full system has also been largely based on visual interpretation of the observed results, rather than carefully controlled experiments. Nevertheless, we believe our results achieve a high degree of realism.

Our capillary flow model for liquid transport between hairs is fairly na[']ive, since we consider only the graph structure and distance between hairs, and not an accurate structure/volume of channels between them. We found that it was much less effective when adjacent hairs are discretized in a highly non-uniform fashion. While such scenarios tend to be less common, this suggests there is room for improvement. Moreover, this effect typically occurs over longer time scales, so it may be entirely unimportant for many scenarios, including the large-scale ones we considered.

We found that the capture and dripping mechanisms can lead to a dramatic range of particles sizes spanning a few orders of magnitude, which presents a severe challenge to the reconstruction of a smooth and temporally coherent surface. This suggests two avenues. First, one could pursue improved surface reconstruction methods under such conditions, which we view as beyond the scope of the current work. Secondly, it could also be helpful to explore the development of strategies to enforce a narrower variation in particle size. The main source of arbitrary particle sizes is our capture mechanism; while precisely conservative in terms of the water volume the hair extracts from a particle, no quantization is guaranteed on the potentially quite small particle volume left behind.

While we have sought to preserve volume throughout, our (discretized) surface liquid model is not strictly conservative and leads to a small accumulating volume loss over time, particularly in the presence of large thickness gradients. Challenges of this nature remain open in computational physics as well [36], where the transport equation may have negative solutions, and direct clamping to tends to break the divergence-free condition. Alternative options are to smooth large variations with an artificial viscosity, or to solve a least squares system for transport under a positivity constraint. In preliminary experiments in this direction, the former was only partially effective while the latter struggled to converge.

Inter-hair forces dominate runtime costs, echoing the typical bottleneck in handling dry inter-hair contacts. Compared with the same elastic rod model by Kaufman et al. [129], our simple linearized solver suffers less cost per time step, although it also

Examples	s/step hair	(avg.) liquid	# vertices of strands	grid size dimensions	Δx (cm)
Shaking “Dog”	29.95	11.83	524.2K	$128 \times 128 \times 256$	1.0
Wet Hair Flip	49.65	6.91	335.8K	$128 \times 128 \times 128$	0.673
Car Wash Roller	4.60	23.14	462.8K	$96 \times 144 \times 96$	1.333
Wringing Out Hair	2.26	14.42	167.9K	$192 \times 192 \times 64$	0.052
Fur Mat (5cm)	2.32	1.18	65.5K	$64 \times 64 \times 64$	0.25

Table 11.1: **Timings and storage statistics.**

requires smaller time steps for the stiff repulsive/cohesive constraints between hairs.

Overall the cost of our hair dynamics are comparable with the dry contact timings reported in prior works that handles per-strand collisions. With a larger drag force, however, our linearized solver needs very small time step for maintaining stability.

In **Part III**, we would replace the strand solver with a nonlinear one, similar to the work by Kaufman et al. [129], and demonstrate a new pipeline to couple nonlinear hair strands with liquid with much larger drag force.

Our current model solves degrees of freedom for all hair strands, which is slow and memory-intensive for massive numbers of hairs: for example, human heads on average have 100k hairs at the scalp, which is three times larger than the largest example presented in this part. This approach differs from another family of methods for hair dynamics that use hierarchical structures, such as hair clusters and strips [50, 51, 245, 246, 248], and can achieve interactive or even real-time performance with sophisticated collision handling. While our method focuses on the multi-scale physics of liquid-hair coupling, these prior works are targeted at level-of-detail effects for hair simulation. Combining multi-scale physics interactions for wet hairs with such reduced models for high computational performance is a potentially promising direc-

tion for future research, and we hope that our work spurs extensions to such reduced or guide-hair settings.

Our simple, yet explicit coupling strategy between the liquid and the submerged hairs, presented in this part, cannot correctly simulate the buoyancy effect. In **Part III**, we will derive a new framework from the first principle for the coupling between the liquid and hairs, where the liquid pressure would be more accurate, and the buoyancy of strands can be correctly handled. Also, we will formally introduce the source of momentum transport along the hair, and show that the transport can be implicitly integrated by adding impulse to the strands, which provides stable simulation. Additionally, we have not revealed the specific formula of the drag force for the discrete elastic rods submerged in a liquid, which would be introduced in **Part III**.

Finally, we have only considered non-viscous, Newtonian liquid in this part. In **Part III**, however, we will generalize the methods introduced in this part for shear-dependent liquid, where we will couple strands with oil paint, mud, hot chocolate, and pasta sauce.

Part III

Multi-Scale Model for Coupling Strands with Shear-Dependent Liquid

Chapter 12

From a Newtonian Liquid to a Shear-Dependent Liquid

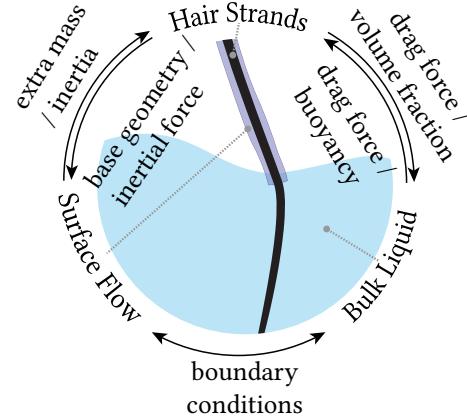
In **Parts I** and **II**, we have developed models for the cohesion between strands, the surface liquid along a fiber or a strand, and a simple two-way coupling scheme between the liquid and the submerged fabrics or hairs.

These models, however, only account for Newtonian and incompressible liquids. Proper treatment of interactions with non-Newtonian and compressible liquids requires a fundamental reconsideration of the underlying model components and their interactions.

For instance, a shear-dependent liquid can deform elastically or plastically. Such deformation is not captured by the prior surface liquid model, presented in **Part II**. Therefore, a new method is necessary if we are to simulate these detailed shear-dependent flows economically.

Another influential effect is the friction between wet strands, which is often a more significant factor in the context of shear-dependent liquid due to the influence of larger viscous and/or elastic forces. This fact suggests the need for an accurate, combined cohesive-frictional contact model to achieve a realistic behavior.

Finally, the models used for the strands, the bulk liquid, and the reduced surface flow must be coupled together in a *principled* framework so that we can stably, and plausibly simulate the drag and buoyancy behavior of the strands submerged. We identify the following requirements



for successful coupling: **1)** Since the strands act as the base geometry of the surface flow, the surface flow should contribute extra mass and inertia to the strands. **2)** The bulk liquid should exchange momentum with the submerged strands through drag forces and pressure gradients (buoyancy), which depend on the volume fraction of the submerged strands. **3)** The surface flow and the bulk liquid should maintain consistent velocities and pressures where they meet, through appropriate boundary conditions. These requirements apply to both Newtonian and non-Newtonian fluids. However, some non-Newtonian fluids (e.g., shaving cream or pasta sauce) are compressible and can, therefore, change their volume during deformation, which leads to a more complicated calculation for pressure.

Our contribution in this part is the development of a multi-scale framework capturing the interactions between strands and compressible shear-dependent fluid (where the incompressible or Newtonian fluid serves as a special case), which includes:

- a reduced model for shear-dependent liquids flowing on the strands that accounts for elastic and plastic deformation;

- three-way coupling between discrete elastic rods, continuum bulk liquid, and reduced surface flow in a principled framework, which accounts for a wide range of rheologies and degrees of compressibility;
- an extension to the existing shifted second-order Coulomb cone model [2, 65, 66, 87, 119, 125, 130, 195, 230] for stable cohesive and frictional effects when strands collide, by introducing contact hysteresis on both distance and relative velocity (extending the contact hysteresis where the cone is only shifted when two colliders are separating for the first time [119]), and a cohesive force designed for viscoplastic materials; and
- a stable semi-implicit solver for shear-dependent fluid, with a more efficient semi-analytical formulation of plastic flow.

Chapter 13

Physical Models for Coupling Strands with a Shear-Dependent Liquid

We begin by firstly introduce the background knowledge on shear-dependent liquid, which forms the basis of the simulation of the bulk liquid, the surface flow, their coupling, and the cohesion between strands. Following this introduction, we present three primary physical models to support simulating strands interacting with non-Newtonian liquids: **i)** a reduced non-Newtonian surface flow model, **ii)** a discrete strand model driven by strand-liquid interaction, and **iii)** a model for bulk liquid mixed with submerged strands. Similar to prior chapters, in describing the three physical models, our notation will be to use a subscript τ to indicate a quantity related to the surface flow (e.g., u_τ for surface flow velocity), a subscript s for the strand's quantities, and a subscript f for the bulk liquid's quantities.

13.1 Shear-Dependent Liquid

Below we summarize the theory behind the shear-dependent liquid, namely, the \mathcal{J}_2 liquid theory developed by Simo et al. [215]. Due to its simplicity and accuracy, this model is extensively used in prior works on simulating foams [261] or grains [262],

and is employed here as the constitutive model.

The deformation gradient of a shear-dependent liquid is a second-order tensor defined over the liquid domain Ω , denoted as $\mathbf{F} = \frac{\partial \Psi}{\partial \mathbf{x}} : \Omega \rightarrow \mathbb{R}^{d \times d}$, where Ψ is the deformation and d is the number of dimensions, i.e., $d = 2$ for 2D and $d = 3$ for 3D. Some shear-dependent liquids are compressible, and thus we need to consider their volume change, denoted as $J \equiv \det \mathbf{F}$, and we have [38]

$$\rho_f = J^{-1} \rho_{f,0} \quad (13.1)$$

where ρ_f is the liquid's (dynamic) mass density and $\rho_{f,0}$ is the mass density at rest.

It is convenient to decompose the deformation gradient \mathbf{F} into parts associated to the elastic \mathbf{F}^E and plastic \mathbf{F}^P deformation via the decomposition [21, 116, 124, 217, 251]

$$\mathbf{F} = \mathbf{F}^E \mathbf{F}^P. \quad (13.2)$$

According to the experimental observations [41, 42], the volume change is often reversible even the liquid is under a pressure up to 3×10^{10} dyne/cm². In other words, the plastic deformation is usually volume preserving (or isochoric), i.e., $J^P = \det \mathbf{F}^P = 1$ and $J^E = \det \mathbf{F}^E = J$. Below we ignore the difference between J and J^E , and only deal with the volume change due to elastic deformation.

The elastic energy depends on the rotation-free *left Cauchy-Green tensor* $\mathbf{b} \equiv \mathbf{F}\mathbf{F}^T$ and, especially, its elastic part $\mathbf{b}^E \equiv \mathbf{F}^E\mathbf{F}^{ET}$. The total energy density is then

decomposed into

$$W = W_v(J) + W_s(\mathbf{b}^E) \quad (13.3)$$

where W_v is the energy density resisting any volumetric change, and W_s is the shear-dependent energy density. Similar to prior work [261], we adopt a modified neo-Hookean model [215] and Rivlin's shear-dependent energy density [199] for W_v and W_s , respectively. We have the following constitutive formulas

$$W_v(J) = \frac{1}{2}\kappa\left(\frac{1}{2}(J^2 - 1) - \ln J\right), \quad (13.4)$$

and

$$W_s(\mathbf{b}^E) = \frac{1}{2}\mu\left(J^{E-\frac{2}{d}}\text{tr}\mathbf{b}^E - d\right). \quad (13.5)$$

where κ and μ are the *bulk modulus* and the *shear modulus*.

The Kirchhoff Stress. After the energy densities are defined, the Kirchhoff stress tensor $\boldsymbol{\tau} \in \mathbb{R}^{d \times d}$ can be derived, as following

$$\boldsymbol{\tau} \equiv \frac{\partial W}{\partial \mathbf{F}^E} \mathbf{F}^{ET} = \frac{\kappa}{2}(J^2 - 1)\mathbf{I}_d + \mu J^{E-\frac{2}{d}}\text{dev}[\mathbf{b}^E]. \quad (13.6)$$

where $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ is the d -dimension identity matrix, $\text{dev}[\mathbf{x}] \equiv \mathbf{x} - \frac{\text{tr}[\mathbf{x}]}{d}\mathbf{I}_d$ is the deviatoric operator. In (13.6), the first part is known as the *dilational Kirchhoff stress*, while the second part is the *shear Kirchhoff stress*. The Cauchy stress tensor is then computed with $\boldsymbol{\sigma} \equiv \boldsymbol{\tau}/J$.

Pressure. We first compute the *dilational Cauchy stress* (which equals to the dilational Kirchhoff stress divided by J), as following [38]

$$\begin{aligned}\sigma_v &\equiv \frac{1}{J} \frac{\partial W_v}{\partial \mathbf{F}^E} \mathbf{F}^{ET} = \frac{1}{J} \frac{\partial W_v}{\partial J^E} \frac{\partial J^E}{\partial \mathbf{F}^E} \mathbf{F}^{ET} \\ &= \frac{1}{J} \frac{\partial W_v}{\partial J^E} J^E \mathbf{F}^{E-T} \mathbf{F}^{ET} = \frac{1}{J} \frac{\partial W_v}{\partial J^E} J^E \mathbf{I}_d \\ &= \frac{\partial W_v}{\partial J^E} \mathbf{I}_d\end{aligned}\tag{13.7}$$

The negative value of this scalar applied on the d -dimensional identity matrix \mathbf{I}_d is then defined as the *pressure* [226], i.e.,

$$p \equiv -\frac{\partial W_v}{\partial J^E}\tag{13.8}$$

Shear Kirchhoff Stress. The shear Kirchhoff stress is defined as the deviatoric part of $\boldsymbol{\tau}$. Since (13.6) only contains diagonal terms in its first part, the shear Kirchhoff stress is equivalent to the second part of (13.6), where

$$\mathbf{s} \equiv \text{dev}[\boldsymbol{\tau}] = \mu J^{E-2/d} \text{dev}[\mathbf{b}^E].\tag{13.9}$$

Its scalar magnitude is then given as

$$s = \|\mathbf{s}\|\tag{13.10}$$

where $\|\cdot\|$ is the Frobenius norm, and we have also used the normalized deviatoric stress tensor defined as $\hat{\mathbf{s}} \equiv \mathbf{s}/s$.

Plasticity. Once the shear stress is larger than some threshold, the liquid will yield to the shear stress and its elastic deformation will irreversibly convert into plastic deformation, i.e., there will be a plastic flow. We adopt the simple and efficient *von Mises yield condition* [164] as the threshold for the onset of a plastic flow. This condition is written in the terms of the material-dependent yield stress τ_Y , as following

$$\Phi(s) = s - \sqrt{\frac{2}{3}}\tau_Y \leq 0. \quad (13.11)$$

For simplicity, we neglect any hardening or softening effects since they are not observable for the materials we considered [58, 250].

When the yield condition is violated, we compute the plastic flow according to the *yield excess* $\Phi(s)$ to estimate the excessive elastic strain that becomes the plastic strain. The temporal derivative of \mathbf{b}^E is given as [215, 216]

$$\frac{d\mathbf{b}^E}{dt} = \nabla \mathbf{u}_f \mathbf{b}^E + \mathbf{b}^E \nabla \mathbf{u}_f^T - \frac{2}{d} \text{tr}[\mathbf{b}^E] \gamma(s) \hat{\mathbf{s}} \quad (13.12)$$

where $\mathbf{u}_f \in \mathbb{R}^{d \times 1}$ is liquid velocity. The first two terms capture the change due to the flow field itself, while the last term captures the change due to plastic flow with flow rate denoted as γ (with physical unit s^{-1}).

We adopt the Herschel-Bulkley model [102] since it has been validated for a wide

range of materials. The flow rate formula for γ is therefore

$$\gamma(s) = \max \left(0, \frac{\Phi(s)}{\eta} \right)^{1/n}, \quad (13.13)$$

where η is the *flow consistency index* with physical unit $\text{Ba} \cdot \text{s}^n$ (or $\text{Pa} \cdot \text{s}^n$ in SI units), and n is the unitless *flow behavior index*. The liquid is pseudoplastic (shear-thinning) when $n < 1$, Newtonian when $n = 1$, and dilatant (shear-thickening) when $n > 1$.

The flow consistency index η indicates how slow the liquid would “forget” its elastic deformation. Liquid with a smaller η would become free from the elastic deformation more quickly. In the limit of $\eta \rightarrow 0$, any elastic deformation would immediately become plastic (Bingham plastics). If the yield stress is also zero, the liquid then becomes inviscid.

Remark: Connection with a Newtonian Liquid. By definition, a Newtonian liquid has a negligible elastic strain. We can then rewrite the elastic Cauchy-Green strain as $\mathbf{b}^E = \mathbf{I}_d + \epsilon \frac{d\mathbf{b}^E}{dt}$ where $\epsilon \ll 1$ is a tiny positive perturbation variable. We also have $\tau_Y = 0$ and $n = 1$. (13.12) then becomes

$$\epsilon \frac{d^2 \mathbf{b}^E}{dt^2} = \nabla \mathbf{u}_f \left(\mathbf{I}_d + \epsilon \frac{d\mathbf{b}^E}{dt} \right) + \left(\mathbf{I}_d + \epsilon \frac{d\mathbf{b}^E}{dt} \right) \nabla \mathbf{u}_f^T - 2 \left(1 + \frac{\epsilon}{d} \text{tr} \left[\frac{d\mathbf{b}^E}{dt} \right] \right) \mathbf{s} \eta^{-1} \quad (13.14)$$

After some algebraic manipulation, we have a shear stress \mathbf{s} equivalent to the viscous tensor for the Newtonian fluid, which proves that the flow consistency index plays

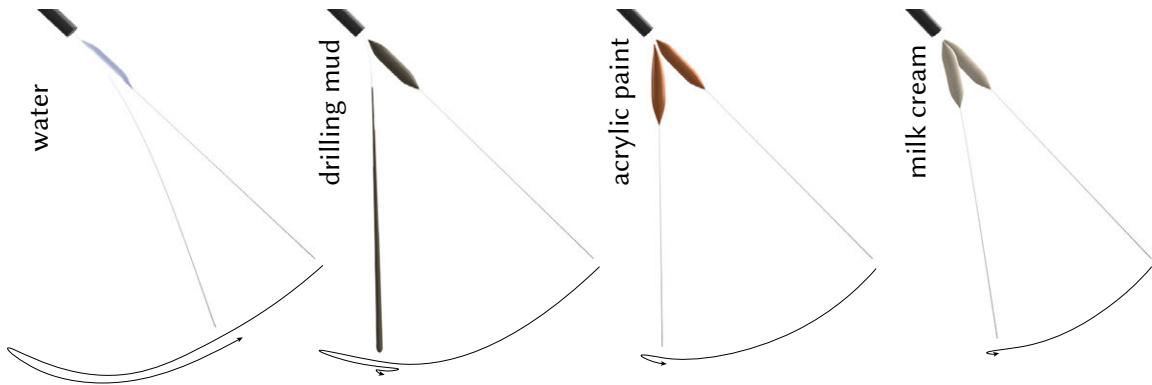


Figure 13.1: **Surface flows of various materials.** Different material properties yield drastically different behaviors. The beginning and ending frames are shown overlaid, with the curved arrows indicating the motion of strands.

the same role as the viscosity coefficient in a Newtonian liquid

$$\mathbf{s} = \frac{\eta}{2} (\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T) + \mathcal{O}(\epsilon). \quad (13.15)$$

13.2 Non-Newtonian Strand Surface Flow

After the background knowledge introduced, the first component to be developed is a model for non-Newtonian liquid flowing on the surface of a strand. While potentially imperceptible on a single thin strand, surface flow can significantly change the collective dynamics (and thus appearance) of many strands interacting and agglomerating together (Figure 1.3). Unfortunately, capturing the shallow depth of such a surface flow with a grid-based fluid solver would require an excessively refined background grid, making them a poor choice in this case. Different than **Part II** where we directly introduced the surface liquid model for Newtonian liquid (§8), in the following

sections we first *justify our assumptions*, so that we would have a *more principled* derivation of the surface liquid model for non-Newtonian liquid.

Kinematics of the Shear-Dependent Surface Liquid on a Strand

To present our surface liquid model, we start with the surface flow kinematics described using the cylindrical coordinate frames. These coordinate frames are aligned along a strand's centerline. In these frames, the surface liquid velocity is denoted as $\mathbf{u}_\tau = (u_\tau, v_\tau, \omega y)^T$, where u_τ and v_τ are liquid velocities along the centerline's tangential and radial directions, respectively, and ω is the angular velocity around the strand (see Figure 13.2 for an illustration of these notations). Our reduced surface flow model is established with a few assumptions on the velocity field:

1. The flow thickness h_τ and strand radius r are much smaller than the length L of the strand. Thus, we only need to consider the *average* longitudinal velocity u_τ across its depth and orientation, and the average velocity u_τ is invariant in $y-$ and $\theta-$ coordinates (i.e., $\partial u_\tau / \partial y = 0$ and $\partial u_\tau / \partial \theta = 0$).
2. The bulk modulus of a non-Newtonian fluid is often several orders of magnitude larger than its shear modulus¹. Given the surface flow layer is also thin, consequently, the volumetric change of the surface flow is negligible in comparison to its translation and shear motion. In other words, assuming the surface flow to

¹For example, shaving cream has a bulk modulus $1.09e6$ dyne/cm² while its shear modulus is $2.9e3$ dyne/cm². Similar materials (e.g., pasta sauce) or materials that have higher bulk but lower shear modulus (e.g., paint and mud) are used in this part. See Table 15.2 for detailed parameters.

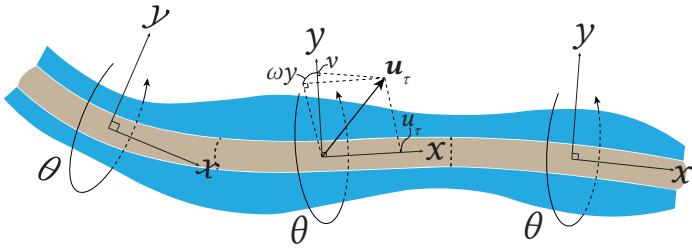


Figure 13.2: **Coordinate system defined along the strand centerline.** At each point of the strand’s centerline, the x -axis, i.e., longitudinal axis, of the frame is along the centerline’s tangential direction; the y -axis, i.e., radial axis, is the strand’s radial direction; and the θ -axis is the angular direction around the strand. The local velocity field (in the cross-section at each point) of the non-Newtonian surface fluid is $\mathbf{u}_\tau = (u_\tau, v_\tau, \omega_y)^T$, where u_τ and v_τ are fluid velocities along the centerline’s tangential and radial directions, respectively, and ω is the angular velocity around the strand.

be isochoric (incompressible) will not affect its visual appearance: $\nabla \cdot \mathbf{u}_\tau = 0$

and $\det \mathbf{F} = 1$, where \mathbf{F} is the deformation gradient.

3. Because the strand radius is small and the surface flow layer is thin, the fluid has relatively strong surface tension — $\mathcal{O}((h_\tau + r)^{-1})$ according to the Young-Laplace equation [260] — that keeps the fluid uniform around the strand centerline. Therefore, we ignore the angular motion of the surface flow (i.e., we assume $\omega = 0$).

Assumptions (i) and (iii) ensure that the surface liquid has no strain in the angular direction: its 3D strain tensor is reduced to a 2D strain tensor—a 2D symmetric matrix having three independent elements. The two diagonal elements indicate the normal strains separately in the longitudinal and radial directions, while the off-diagonal element involves both directions.

The 2D strain tensor can be reduced further. Since the strand’s radius r is much

smaller than its length L , the surface flow behavior in the strand cross-section differs from that in the longitudinal direction by a scaling factor $\epsilon \equiv r/L$ where $\epsilon \ll 1$. Through a multiscale asymptotic analysis with respect to ϵ (later we will present a detailed derivation), we find that the off-diagonal element of the strain tensor has only a second-order (i.e., ϵ^2) contribution to the deviatoric part of the strain, which is the strain component determining the shear stress and plastic flow. This observation suggests that we can ignore the off-diagonal element.

Moreover, due to incompressibility (i.e., assumption (ii) above), the strains in the longitudinal and radial directions are related: compression in one direction leads to an expansion in the other. Based on this condition, we further reduce the 2D strain to a single scalar, which we call the *reduced Cauchy-Green strain*,

$$c_\tau^E \equiv b_{xx} - b_{yy}, \quad (13.16)$$

where b_{xx} and b_{yy} are the principal strains in the longitudinal and radial directions (i.e., the two diagonal elements).

Dynamics of the Shear-Dependent Surface Liquid on a Strand

Momentum Equation. Incorporating this reduced strain leads to a modified form of the momentum equation for non-Newtonian fluids. As derived in a later section,

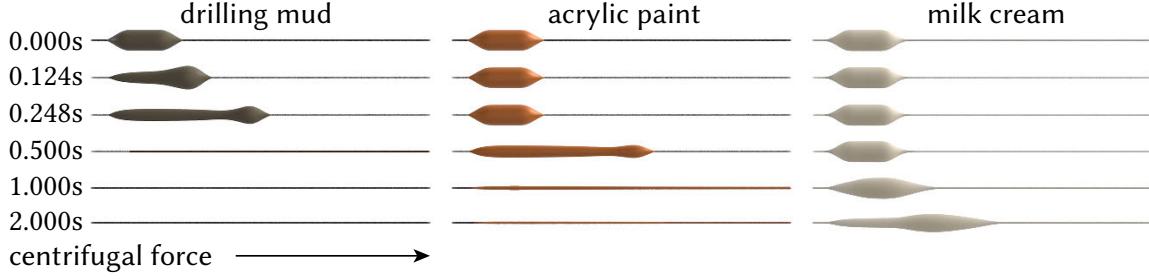


Figure 13.3: **Surface flows under increasing centrifugal force.** Three strands with surface flows of different materials are rotated with increasing angular velocity over time to apply increasing centrifugal force. The camera is aligned with the strands' tangential direction for legibility. Compared with the mud, the paint and cream only yield and start flowing under sufficiently high stress, where the cream has especially large yield stress (1200dyne/cm^2).

the standard 3D Navier-Stokes momentum equation becomes a simpler 1D equation,

$$A_\tau \rho_f \frac{D_{u_\tau} u_\tau}{Dt} = A_\tau f_{\text{ext},x} + \frac{\partial}{\partial x} (\mu A_\tau c_\tau^E) - C u_\tau, \quad (13.17)$$

where μ is the fluid's shear modulus, ρ_f is the liquid density, $A = \pi h_\tau (h_\tau + 2r)$ is the annular cross-sectional area of the flow with a thickness h_τ , and $f_{\text{ext},x}$ indicates external force density in the tangential direction. Similar to (8.15) in the last part, the external forces include gravity, inertial force due to strand motion, as well as the coupling forces among the surface flow, strands, and bulk fluid. The last term $C u_\tau$ in (13.17) is the friction force, which we will discuss shortly.

The reduced strain c_τ^E is not only used to model the shear force in (13.17), but is also time-varying and obeys a *flow rule*. For non-Newtonian fluids, the strain rate is determined by both elastic deformation and plastic flow. Their specific contributions,

emerging from a multiscale asymptotic analysis of the deviatoric strain, are given by

$$\frac{D_{u_\tau} c_\tau^E}{Dt} = \underbrace{2 \frac{\partial u_\tau}{\partial x} \sqrt{c_\tau^{E2} + 4}}_{\text{elastic deformation}} - \underbrace{\sqrt{2} \gamma(s_\tau) \left(c_\tau^E + \sqrt{c_\tau^{E2} + 4} \right) \text{sgn}(c_\tau^E)}_{\text{plastic flow}}, \quad (13.18)$$

where $s_\tau \equiv 2^{-1/2} \mu |c_\tau^E|$ is the magnitude of the shear stress, and $\gamma(s)$ is the flow rate function (13.13).

Derivation of the 1D Surface Flow

Before walking through the friction and mass conservation, we present a detailed, principled derivation for the momentum equation (13.17) and the strain derivative (13.18) introduced above. We take the plane-strain conditions, where we can safely ignore the velocity in the angular direction around the strand. Thus we can derive a 1D surface flow model from the 2D theory of shear-dependent liquid. For legibility, in the following discussion, we use the label of the axis to represent the corresponding row or column in the subscript of a strain or stress tensor, i.e., x for the first row or column, y for the second row or column.

Parameter Scaling. We denote the height of flow as h , and we have the velocity in the y -axis, i.e., radial axis, denoted as $v = \partial h / \partial y$. The velocity of the 2D flow is then denoted as $\mathbf{u}_f \equiv (u_\tau, v)$. From our assumptions (see §13.2), the surface flow is thin in height, where we can define a scaling parameter $0 < \epsilon \equiv r/L \ll 1$, and we define

$$h \equiv \epsilon H, y \equiv \epsilon Y, v \equiv \epsilon V, b_{xy}^E \equiv \epsilon B_{xy}^E, \quad (13.19)$$

Additionally, we have $\partial/\partial y = \epsilon^{-1}\partial/\partial Y$ as an deduction.

To expand a scalar to a matrix, we adopt these notations below

$$[*]_e \equiv [*; 0; 0, 0] \in \mathbb{R}^{2 \times 2}, [*]_s \equiv [0, *; *, 0] \in \mathbb{R}^{2 \times 2}. \quad (13.20)$$

Reduced Cauchy-Green Strain. The 2D left Cauchy-Green elastic strain tensor \mathbf{b}^E can be then written as

$$\mathbf{b}^E = [c_\tau^E]_e + b_{yy}^E \mathbf{I}_2 + \epsilon[B_{xy}^E]_s \quad (13.21)$$

where $c_\tau^E \equiv b_{xx}^E - b_{yy}^E$ is called the *reduced left Cauchy-Green strain*. In the following derivation, we will discover its temporal derivative, which is used to evolve the liquid's elastic and plastic deformation, and the momentum equation, which uses c_τ to compute shear stress.

By applying the deviatoric operator to both sides of (13.21), we have

$$\text{dev}[\mathbf{b}^E] = \text{dev}[[c_\tau^E]_e] + \epsilon[B_{xy}^E]_s \quad (13.22)$$

The definition of shear stress and its norm have been introduced in (13.9) and (13.10).

We can rewrite (13.10) using the scaled variables, as following

$$s = \mu J^{E^{-1}} \|\text{dev}[\mathbf{b}^E]\| = \mu J^{E^{-1}} \sqrt{\frac{1}{2} c^2 + 2\epsilon^2 B_{xy}^{E,2}}. \quad (13.23)$$

The derivative of c_τ^E is then computed as

$$\begin{aligned}\dot{c}_t^E &= \dot{b}_{xx}^E - \dot{b}_{yy}^E \\ &= 2 \left(\frac{\partial u_\tau}{\partial x} - \frac{\sqrt{2} c_\tau^E \gamma(s)}{\sqrt{c_\tau^{E^2} + 4\epsilon^2 B_{xy}^{E^2}}} \right) (c_\tau^E + b_{yy}^E) + 2 \left(\frac{\partial u_\tau}{\partial y} - \frac{\partial v}{\partial x} \right) \epsilon B_{xy}^E - 2 \frac{\partial v}{\partial y} b_{yy}^E\end{aligned}\tag{13.24}$$

We only consider the *average* longitudinal velocity u_τ across its depth, i.e., $\partial u_\tau / \partial y = 0$ (assumption (1)). With the scaling proposed in (13.19), we can approximate \dot{c}^E by neglecting the terms multiplied with ϵ^2 , as following:

$$\dot{c}_t^E = 2 \left(\frac{\partial u_\tau}{\partial x} - \sqrt{2} \gamma(s) \operatorname{sgn}(c_\tau^E) \right) (c_\tau^E + b_{yy}^E) - 2 \frac{\partial v}{\partial y} b_{yy}^E + \mathcal{O}(\epsilon^2)\tag{13.25}$$

We then use the incompressible condition (assumption (2))

$$\frac{\partial u_\tau}{\partial x} + \frac{\partial v}{\partial y} = 0, \det[\mathbf{b}^E] = 1,\tag{13.26}$$

to eliminate v and b_{yy}^E . For the latter we have

$$\det[\mathbf{b}^E] = b_{yy}^{E^2} + c_\tau^E b_{yy}^E + \mathcal{O}(\epsilon^2) = 1.\tag{13.27}$$

Since $b_{yy}^E \geq 0$, we then have

$$b_{yy}^E = \frac{\sqrt{c_\tau^{E^2} + 4 - \mathcal{O}(\epsilon^2)} - c_\tau^E}{2}\tag{13.28}$$

By substitution of (13.26), (13.28) into (13.25), and with the ϵ^2 terms neglected, we have the temporal derivative of c_τ^E in the same form as (13.18). ■

Shear Stress. After performing a decomposition to the 2D deviatoric Kirchhoff stress $\text{dev}[\boldsymbol{\tau}]$, we have

$$\text{dev}[\boldsymbol{\tau}] = \text{dev}[(\tau_{xx} - \tau_{yy})_e] + [\tau_{xy}]_s \quad (13.29)$$

From (13.6) we then have

$$\text{dev}[\boldsymbol{\tau}] = \mu \text{dev}[[c_\tau^E]_e] + \mu \epsilon [B_{xy}^E]_s \approx \mu \text{dev}[[c_\tau^E]_e]. \quad (13.30)$$

The 2D Cauchy stress tensor is therefore computed as (using the incompressibility assumption $J = 1$)

$$\boldsymbol{\sigma}_{2D} = \mu \text{dev}[[c_\tau^E]_e] - p \mathbf{I}_2. \quad (13.31)$$

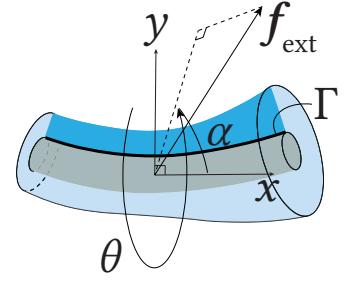
Since our flow is symmetric around the strand centerline, with the plane strain conditions the 3D stress tensor $\boldsymbol{\sigma}$ can be specified with this 2D stress tensor, where we have

$$\boldsymbol{\sigma} = [\boldsymbol{\sigma}_{2D,xx}, \boldsymbol{\sigma}_{2D,xy}, 0; \boldsymbol{\sigma}_{2D,xy}, \boldsymbol{\sigma}_{2D,yy}, 0; 0, 0, (\boldsymbol{\sigma}_{2D,xx} + \boldsymbol{\sigma}_{2D,yy})/2r]. \quad (13.32)$$

which will be used in the following derivation for a 1D reduced momentum equation.

Variational Form of the Momentum Equation. In the following we derive the 1D reduced momentum equation from the 3D Navier-Stokes momentum equation, which reads

$$\rho_f \frac{D\mathbf{u}_f}{Dt} - \nabla \cdot \boldsymbol{\sigma} = \mathbf{f}_{\text{ext}} + \mathbf{f}_\Lambda \quad (13.33)$$



where \mathbf{f}_Λ is the frictional force on the bottom of the flow; \mathbf{f}_{ext} is the external body force such as gravity and inertial force; and $\boldsymbol{\sigma}$ is the Cauchy stress tensor. We denote the interface between the reduced surface flow and the strand as Γ , and define a trial function $\Phi \equiv (\Phi_x, \Phi_y, \Phi_\theta)$ that is a vector defined in the cylindrical coordinate whose x -axis is aligned with the longitudinal direction of the strand. The weak formulation of (13.33) can be written as following

$$\begin{aligned} & \int_0^{2\pi} \int_r^{r+h_\tau} \int_{\Gamma} \left[\rho_f \left(\frac{\partial \mathbf{u}_f}{\partial t} + \mathbf{u}_f \cdot \nabla \mathbf{u}_f \right) \cdot \Phi + \boldsymbol{\sigma} : \boldsymbol{\Upsilon}(\Phi) \right] y \, dx \, dy \, d\theta \\ & + \int_0^{2\pi} \int_{\Gamma} \sigma_{yy} \Phi_y r \, dx \, dy \, d\theta = \int_0^{2\pi} \int_r^{r+h_\tau} \int_{\Gamma} (\mathbf{f}_\Lambda + \mathbf{f}_{\text{ext}}) \cdot \Phi \, y \, dx \, dy \, d\theta \end{aligned} \quad (13.34)$$

where σ_{yy} is the pressure applied to the flow by the strand surface Γ , $\boldsymbol{\Upsilon}$ is the rate of deformation tensor, i.e., $\boldsymbol{\Upsilon}(\Phi) \equiv (1/2) (\nabla \Phi + \nabla^T \Phi)$. In the following derivation we will use two deductions based on the assumptions made in §13.2: 1) due to the symmetry of the flow, we have $\mathbf{u}_f = (u_\tau, v, 0)$; and 2) due to the incompressibility of the flow, we have

$$\frac{\partial u_\tau}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (13.35)$$

Integrating the above equation over the y -axis from Γ to the free surface, and using the fact that $v|_{\Gamma} = 0$, i.e., the flow cannot penetrate the strand surface, we have

$$v = -y \frac{\partial u_{\tau}}{\partial x}. \quad (13.36)$$

Pressure of the Reduced Surface Flow. To derive the pressure σ_{yy} , we first choose a trial function by setting $\Phi_x = 0$ and $\Phi_{\theta} = 0$, i.e., only allowing $\Phi_y \neq 0$. With (13.30), (13.31), and (13.32) substituted into (13.34), and with the variables replaced with our scaled ones proposed in (13.19), we have

$$\begin{aligned} & \epsilon^3 \int_0^{2\pi} \int_r^{r+\epsilon H} \int_{\Gamma} \left[-H^3 Y^3 \rho_f \left(\frac{\partial^2 u_{\tau}}{\partial t \partial x} + \mathbf{u}_{\tau} \frac{\partial^2 u_{\tau}}{\partial x^2} - \left(\frac{\partial u_{\tau}}{\partial x} \right)^2 \right) \Phi_y \right. \\ & \quad \left. + \mu H^2 Y^2 B_{xy}^E \frac{\partial \Phi_y}{\partial x} \right] dx dy d\theta + \epsilon \int_0^{2\pi} \int_{\Gamma} \sigma_{yy} \Phi_y r H dx dy d\theta \\ &= \epsilon^2 \int_0^{2\pi} \int_r^{r+h_{\tau}} \int_{\Gamma} \|\mathbf{f}_{\text{ext}}\| \sin \alpha \cos \theta H^2 Y dx dy d\theta \end{aligned} \quad (13.37)$$

where α is the angle between the direction (in Euclidian space) of x -axis of the cylindrical coordinate and the direction of external force projected onto the xy -plane.

The right-hand side of (13.37) is zero since the external force perpendicular to the strand cancels after being integrated over $d\theta$. By dividing both sides with ϵ and discarding the remaining high-order terms multiplied with ϵ^2 , we have

$$\int_0^{2\pi} \int_{\Gamma} \sigma_{yy} \Phi_y r H dx dy d\theta = 0, \quad (13.38)$$

for arbitrary Φ_y , which simply indicates

$$\sigma_{yy} = 0. \quad (13.39)$$

Momentum Equation of the Reduced Surface Flow. To derive the reduced momentum equation about u_τ , we choose another trial function by setting $\Phi = (\Phi_x, -y\frac{\partial\Phi_x}{\partial x}, 0)$ — the middle term is set according to (13.36). After the variables in (13.34) replaced with our scaled ones, and (13.39) used, we have

$$\begin{aligned} & \epsilon^2 \int_0^{2\pi} \int_r^{r+\epsilon H} \int_{\Gamma} H^2 Y \left[\rho_f \left(\frac{\partial u_\tau}{\partial t} + u_\tau \frac{\partial u_\tau}{\partial x} \right) \Phi_x + \mu c_\tau^E \frac{\partial \Phi_x}{\partial x} \right] + \\ & \epsilon^4 \int_0^{2\pi} \int_r^{r+\epsilon H} \int_{\Gamma} \left[H^4 Y^4 \rho_f \left(\frac{\partial^2 u_\tau}{\partial t \partial x} + \mathbf{u}_\tau \frac{\partial^2 u_\tau}{\partial x^2} - \left(\frac{\partial u_\tau}{\partial x} \right)^2 \right) \frac{\partial \Phi_x}{\partial x} \right. \\ & \left. + \mu H^3 Y^3 B_{xy}^E \frac{\partial^2 \Phi_x}{\partial x^2} \right] dx dY d\theta = \epsilon^2 \int_0^{2\pi} \int_r^{r+\epsilon H} \int_{\Gamma} H^2 Y \\ & \left[(\|\mathbf{f}_{\text{ext}}\| \cos\alpha + f_\Lambda) \Phi_x - \epsilon \|\mathbf{f}_{\text{ext}}\| \sin\alpha \cos\theta H Y \frac{\partial \Phi_x}{\partial x} \right] dx dY d\theta \end{aligned} \quad (13.40)$$

where f_Λ is the magnitude of frictional force. The last term on the right hand side of (13.40) is again an external force perpendicular to the strand, which will be canceled after being integrated over $d\theta$. We then divide both sides with ϵ^2 and discard the remaining high-order terms containing ϵ^2 , which gives us

$$\begin{aligned} & \int_0^{2\pi} \int_r^{r+\epsilon H} \int_{\Gamma} \left[\rho_f \left(\frac{\partial u_\tau}{\partial t} + u_\tau \frac{\partial u_\tau}{\partial x} \right) \Phi_x + \mu c_\tau^E \frac{\partial \Phi_x}{\partial x} \right] H^2 Y dx dY d\theta \\ & = \int_0^{2\pi} \int_r^{r+\epsilon H} \int_{\Gamma} (\|\mathbf{f}_{\text{ext}}\| \cos\alpha + f_\Lambda) \Phi_x H^2 Y dx dY d\theta. \end{aligned} \quad (13.41)$$

Furthermore, the term $\int_0^{2\pi} \int_r^{r+\epsilon H} H^2 Y dY d\theta = \pi h_\tau (h_\tau + 2r)$ is the *cross-sectional*

area of the flow, and can be integrated individually since the other terms are independent of Y and θ . After reordering the multiple integrations in (13.41), we have

$$\begin{aligned} & \int_{\Gamma} \left[\rho_f \left(\frac{\partial u_{\tau}}{\partial t} + u_{\tau} \frac{\partial u_{\tau}}{\partial x} \right) \Phi_x + \mu c_{\tau}^E \frac{\partial \Phi_x}{\partial x} \right] \int_0^{2\pi} \int_r^{r+\epsilon H} H^2 Y dY d\theta dx \\ &= \int_{\Gamma} (\|f_{\text{ext}}\| \cos \alpha + f_{\Lambda}) \Phi_x \int_0^{2\pi} \int_r^{r+\epsilon H} H^2 Y dY d\theta dx. \end{aligned} \quad (13.42)$$

For brevity, we denote the cross section as A_{τ} . We then replace the variables in (13.42) back with the non-scaled version, which reads

$$\int_{\Gamma} \rho_f A_{\tau} \left(\frac{\partial u_{\tau}}{\partial t} + u_{\tau} \frac{\partial u_{\tau}}{\partial x} \right) \Phi_x + \mu A_{\tau} c_{\tau}^E \frac{\partial \Phi_x}{\partial x} = \int_{\Gamma} A_{\tau} (\|f_{\text{ext}}\| \cos \alpha + f_{\Lambda}) \Phi_x dx. \quad (13.43)$$

for arbitrary Φ_x . The corresponding momentum equation is then

$$\rho_f A_{\tau} \left(\frac{\partial u_{\tau}}{\partial t} + u_{\tau} \frac{\partial u_{\tau}}{\partial x} \right) - \mu \frac{\partial A_{\tau} c_{\tau}^E}{\partial x} = A_{\tau} (f_{\text{ext},x} + f_{\Lambda}). \quad (13.44)$$

where $f_{\text{ext},x} \equiv \|f_{\text{ext}}\| \cos \alpha$. As f_{Λ} substituted with the friction model proposed below, i.e., $A_{\tau} f_{\Lambda} \equiv -C u_{\tau}$, we have exactly the form of (13.17). ■

Friction between the Surface Flow and the Strand. Because of its viscosity, the non-Newtonian surface flow experiences friction on the strand surface. Microscopically, the friction is caused by intermolecular interactions that prevent the viscous liquid from either separating from or slipping along the solid surface [220]. We adapt a widely used model of this viscosity-induced friction that assumes a linear relationship between the slip velocity and the friction on the surface—the *Navier boundary condition*

tion [140]. Concretely, the friction force can be modeled by the $C u_\tau$ term in (13.17), where, according to [89], the coefficient C is related to the surface flow thickness h_τ by

$$C = A_\tau \kappa \left(h_\tau + \frac{\kappa h_\tau^2}{3\tilde{\eta}} \right)^{-1}. \quad (13.45)$$

Here, κ , defined as $\kappa \equiv \tilde{\eta}/b$, is the friction coefficient depending on the effective fluid viscosity $\tilde{\eta}$ [11], and the *Navier slip length*, b . The latter is a constant depending on the specific fluid and solid materials [45, 185]. This model, albeit simple, can closely match experimental results for various non-Newtonian fluids [208].

When using the Herschel-Bulkley model for non-Newtonian fluid, we can derive its effective viscosity using the *generalized Newtonian fluid* model [115] and obtain

$$\tilde{\eta} = \sqrt{\frac{2}{3}\tau_Y} \left(\frac{|u_\tau|}{h_\tau} \right)^{-1} + \eta \left(\frac{|u_\tau|}{h_\tau} \right)^{n-1}, \quad (13.46)$$

where τ_Y is the *yield stress*, η is the *flow consistency index*, and n is the *flow behavior index*. With the effective viscosity estimated, we use (13.45) in combination with the strand's annular cross-sectional area A_τ to obtain the formula for C in our surface flow momentum equation (13.17):

$$C = \pi \frac{h_\tau + 2r}{b + h_\tau/3} \tilde{\eta}. \quad (13.47)$$

Mass Conservation. Lastly, mass conservation for the surface flow follows (8.20)

in the prior chapter:

$$\frac{D_{u_\tau} A_\tau}{Dt} + A_\tau \frac{\partial u_\tau}{\partial x} = 0. \quad (13.48)$$

In summary, our non-Newtonian reduced surface flow model consists of a momentum equation (13.17), a mass conservation law (13.48), a time evolution equation for the reduced Cauchy-Green strain (13.18), as well as the Herschel-Bulkley flow rate function, i.e., (13.13) in §13.1.

13.3 Strand Dynamics

A strand's motion is heavily influenced by the liquid layer flowing on its surface. If the liquid were simply fixed to the strand surface (without any relative motion), then its only effect would be to cause the *effective* mass of the strand to become the sum of the strand mass, m_s , and the liquid mass, m_τ . However, surface flow motion relative to the strand will induce an additional inertial force on the strand.

This intuition can be formalized by writing out the total inertia of the strand and its surface fluid. We model a strand as a discrete elastic rod [26], and use \mathbf{u}_s to denote the strand velocity at a discrete strand vertex. The surface fluid flows along the strand with a velocity u_τ relative to the strand. Then, the absolute velocity of the surface flow is $\mathbf{u}_s + \mathbf{t}u_\tau$, where \mathbf{t} is the tangential direction of the strand. The

total inertia is

$$m_s \frac{D_{\mathbf{u}_s} \mathbf{u}_s}{Dt} + m_\tau \frac{D_{\mathbf{u}_s + t u_\tau} \mathbf{u}_s}{Dt} = (m_s + m_\tau) \frac{D_{\mathbf{u}_s} \mathbf{u}_s}{Dt} + m_\tau u_\tau \nabla_\tau \mathbf{u}_s. \quad (13.49)$$

The right hand side expands and regroups terms on the left, revealing the extra inertial force as the last term. With this total inertia, the momentum equation for the strands is

$$\tilde{m}_s \frac{D_{\mathbf{u}_s} \mathbf{u}_s}{Dt} = \mathbf{f}_{\text{int},s} + \mathbf{f}_{\text{ext},s} + \tilde{m}_s \mathbf{g} - m_\tau u_\tau \nabla_\tau \mathbf{u}_s, \quad (13.50)$$

where the effective mass $\tilde{m}_s \equiv m_s + m_\tau$, \mathbf{g} is the gravitational constant, and the last term comes from that of (13.49). The term $\mathbf{f}_{\text{int},s}$ represents the internal forces of the discrete elastic rod, including stretching, bending and twisting forces (see [26] for their details). The external force, $\mathbf{f}_{\text{ext},s}$, includes strand-strand (and strand-solid) contact forces as well as the coupling forces with the bulk fluid. These forces will be elaborated later.

Remark. In §8.2, we have also considered the inertial force contribution by surface water flows. However, there we explicitly transport the extra inertia along strand through (8.21) and modify the strand's velocity at the end of each time step. By contrast, in this part, we derived the force term in a more principled manner and incorporated it into the momentum equation (13.50), allowing for *implicit* integration (see §14). As demonstrated in Figure 13.4, explicitly adding the extra inertial force (as in the last part) impairs simulation stability, whereas using implicit integration approach makes the simulation quite stable.

Proof of the equivalence of the additional inertia To be more rigorous, we also provide a proof to show that, when using explicit integration, the momentum transport presented in the last part is equivalent to the (rightmost) additional inertia term in (13.50), when both are integrated explicitly.

We begin from the momentum transfer equation (8.21), which can be re-written through the product rule, as

$$A_\tau \left(\frac{\partial \mathbf{u}_s}{\partial t} + u_\tau \frac{\partial \mathbf{u}_s}{\partial x} \right) + \mathbf{u}_s \left(\frac{\partial A_\tau}{\partial t} + \frac{\partial}{\partial x}(A_\tau u_\tau) \right) = 0. \quad (13.51)$$

According to the mass conservation (13.48) (or equivalently, (8.16) or (8.20)) of the surface flow, the second term multiplied on \mathbf{u}_s is zero, i.e.,

$$\frac{\partial A_\tau}{\partial t} + \frac{\partial}{\partial x}(A_\tau u_\tau) = 0 \quad (13.52)$$

Therefore, as long as the strand is wet, i.e., $A_\tau > 0$, we have

$$\frac{\partial \mathbf{u}_s}{\partial t} = -u_\tau \nabla \mathbf{u}_s. \quad (13.53)$$

In Chapter 2, we firstly solve the momentum transfer (8.21), which, according to the derivation above, is equivalent to solving (13.53). With an explicit integration of (13.53), we have

$$\tilde{\mathbf{u}}_s \leftarrow \hat{\mathbf{u}}_s - h u_\tau \nabla \mathbf{u}_s, \quad (13.54)$$

where $\hat{\mathbf{u}}_s$ and $\tilde{\mathbf{u}}_s$ denote the strands' velocities before and after the momentum transfer

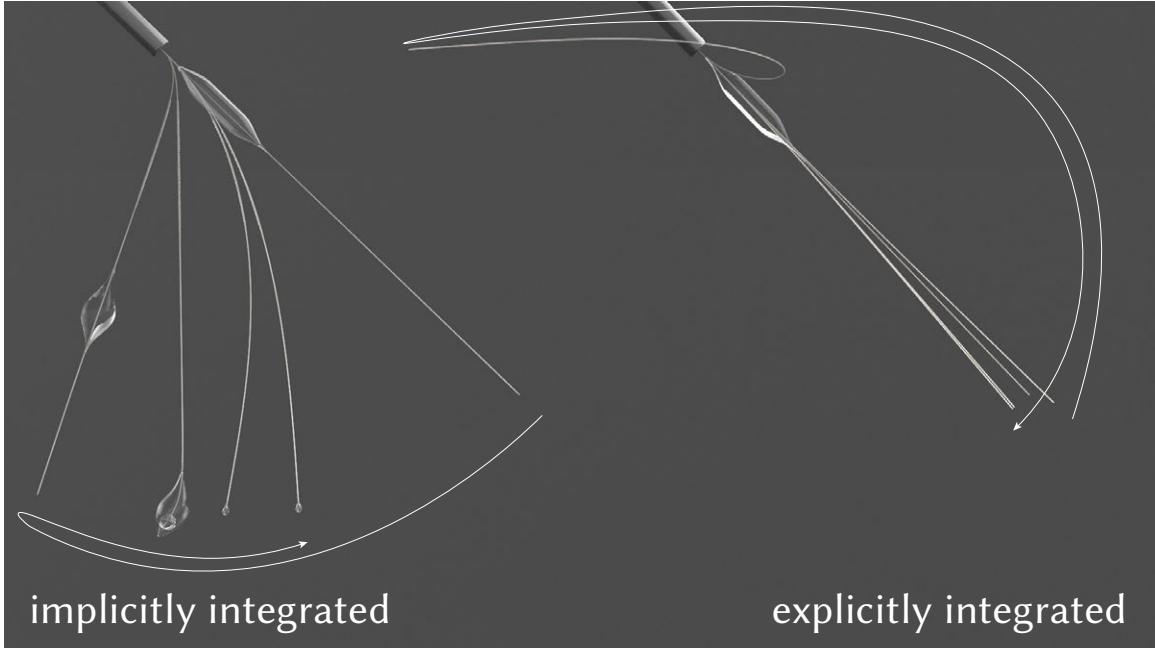


Figure 13.4: **Comparison between different methods to integrate the additional inertia.** A heavy droplet (tetrachloroethylene with mass density $1.622\text{g}/\text{cm}^3$) flows on a strand. **Left:** With the principled momentum equation for the coupled flow, presented in this part, the extra momentum of strand caused by the motion of surface flow can be stably integrated. **Right:** The explicit modification to the strand momentum proposed in last part causes instability.

is done, and h is the time step. We then modify the strands' velocity through (8.23).

By substituting (13.54) into (8.23) and rearranging the terms, we have

$$(m_s + m_\tau)\mathbf{u}_s = (m_s + m_\tau)\hat{\mathbf{u}}_s - hm_\tau u_\tau \nabla \mathbf{u}_s, \quad (13.55)$$

which is exactly the explicit discretization of the strands' momentum equation (13.50)

with all the other forces on its right hand side integrated into $\hat{\mathbf{u}}_s$. Similar derivation can be done for the angular velocity, beginning from the angular momentum transfer equation (8.22). ■

13.4 Bulk Liquid in a Mixture

An effective way of modeling bulk non-Newtonian liquid is to treat it as a continuum, as most non-Newtonian simulation methods have done [114]. In our scenarios, the bulk liquid will inevitably interact with submerged strands, which are modeled as *discrete* rod elements. Thus we are confronted with two contradictory simulation approaches. We must somehow reconcile these disparate views if we are to enforce mass and momentum conservation for both the liquid and the submerged strands in a unified framework.

A natural idea is to homogenize the discrete elements (i.e., strands in our case) and treat the combination of liquid mixed with discrete elements as a porous medium, whose behavior can be described using classic mixture theory [8, 39]. Indeed, the CFD-DEM method [264, 265] is based on this premise, but focuses on incompressible Newtonian fluids [85, 230]. Here we extend this methodology to the simulation of compressible, non-Newtonian fluids.

Volume Fraction for Mixture with Discrete Elements

We begin by homogenizing the volume of the strands. Suppose that the i -th strand element has a volume V_i . Then, in a porous medium where strands are mixed with liquid, the local *volume fraction* of the strand at any point \mathbf{x} can be estimated using a kernel-weighted average [85]:

$$\phi_s(\mathbf{x}) = \frac{1}{V^*} \sum_i w_{R,i}(\mathbf{x}) V_i. \quad (13.56)$$

Here the summation is taken over a small region of the continuum; in practice we use one grid cell. The total volume of the region is denoted V^* , and $w_{R,i}(\mathbf{x})$ is the kernel function centered at the i -th element with a kernel radius R . Similar to the prior work [85], we use a quadratic B-spline as w_R .

We can homogenize the velocity of the discrete strand elements in a similar fashion. Let $\mathbf{u}_{s,i}$ denote the velocity of the i -th strand element. The homogenized strand velocity $\bar{\mathbf{u}}_s(\mathbf{x})$ at position \mathbf{x} is defined as

$$\phi_s(\mathbf{x})\bar{\mathbf{u}}_s(\mathbf{x}) = \frac{1}{V^*} \sum_i w_{R,i}(\mathbf{x}) V_i \mathbf{u}_{s,i}. \quad (13.57)$$

The strand volume fraction evolves over time as the liquid and strand move relative to one another. As we derive below, the material derivative of the strand volume fraction can be expressed as

$$\frac{D_{\mathbf{u}_f}\phi_s(\mathbf{x})}{Dt} = \frac{1}{V^*} \sum_i V_i \nabla w_{R,i} \cdot (\mathbf{u}_f(\mathbf{x}_i) - \mathbf{u}_{s,i}), \quad (13.58)$$

where $\mathbf{u}_f(\mathbf{x}_i)$ is the liquid velocity around the strand's i -th element position, \mathbf{x}_i .

Derivation of the Derivatives of Volume Fraction We begin our derivation from (13.56). By taking its spatial derivative, we have (with the location parameter \mathbf{x} dropped for brevity)

$$\nabla \phi_s = \frac{\partial \phi_s}{\partial \mathbf{x}} = \frac{\sum_i V_i \nabla w_{R,i}}{V^*}. \quad (13.59)$$

Similarly we take the divergence of (13.57), where we have

$$\nabla \cdot (\phi_s \bar{\mathbf{u}}_s) = \frac{\sum_i V_i \nabla \cdot (\mathbf{u}_{s,i} w_{R,i})}{V^*}. \quad (13.60)$$

Since we have assumed that each rod element is incompressible, we have $\nabla \cdot \mathbf{u}_{s,i} = 0$,

and thus the equation above can be rewritten as

$$\nabla \cdot (\phi_s \bar{\mathbf{u}}_s) = \frac{\sum_i V_i \mathbf{u}_{s,i} \cdot \nabla w_{R,i}}{V^*}. \quad (13.61)$$

In mixture theory [8], the continuity equation for a solid with constant mass density reads

$$\frac{\partial \phi_s}{\partial t} + \nabla \cdot (\phi_s \bar{\mathbf{u}}_s) = 0. \quad (13.62)$$

Using (13.62), the material derivative of the solid volume fraction ϕ_s advected along with liquid velocity \mathbf{u}_f is then derived as follows:

$$\frac{D \mathbf{u}_f \phi_s}{Dt} \equiv \frac{\partial \phi_s}{\partial t} + \mathbf{u}_f \cdot \nabla \phi_s \quad (13.63)$$

$$= \mathbf{u}_f \cdot \nabla \phi_s - \nabla \cdot (\phi_s \bar{\mathbf{u}}_s). \quad (13.64)$$

By replacing the terms defined in (13.59) and (13.61), we have

$$\frac{D \mathbf{u}_f \phi_s}{Dt} = \mathbf{u}_f \cdot \frac{\sum_i V_i \nabla w_{R,i}}{V^*} - \frac{\sum_i V_i \mathbf{u}_{s,i} \cdot \nabla w_{R,i}}{V^*} \quad (13.65)$$

$$= \frac{\sum_i V_i (\mathbf{u}_f - \mathbf{u}_{s,i}) \cdot \nabla w_{R,i}}{V^*}. \quad (13.66)$$

which matches (13.58). ■

Lastly, since the volume is a mixture of strands and liquid, the volume fraction of the liquid is related to that of the strands via $\phi_f = 1 - \phi_s$, and its material derivative is $D_{\mathbf{u}_f}\phi_f(\mathbf{x})/Dt = -D_{\mathbf{u}_f}\phi_s(\mathbf{x})/Dt$. This material derivative is needed for the mixture's mass conservation law—which we will present later—because a change of volume fraction at position \mathbf{x} leads to a change of the mixture's local effective density at \mathbf{x} .

Remark. In the derivation of (13.58), we assume that each strand element is incompressible: it can be stretched, bent, or twisted, but always preserves its volume. This assumption is justified by the fact that strands are often thin and stiff. We note that Selle et al. [210] also treated strands collectively as a continuum and assumed incompressible motion. The key difference of our approach is that we only assume each individual strand element is incompressible. The collective volume of strands can still disperse or contract as needed.

Momentum Equation

Using the notion of volume fraction and following classic mixture theory [190], we now have the momentum equation for bulk liquid:

$$\phi_f \rho_f \frac{D_{\mathbf{u}_f} \mathbf{u}_f}{Dt} = \phi_f \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}_{\text{ext},f} + \phi_f \rho_f \mathbf{g}, \quad (13.67)$$

where $\boldsymbol{\sigma}$ is the liquid's internal stress due to its volumetric stress (i.e., pressure p) and shear stress, namely,

$$\boldsymbol{\sigma} = -p\mathbf{I}_3 + \mu J^{-2/3} \text{dev}[\mathbf{b}^E]. \quad (13.68)$$

Here μ is the liquid's shear modulus, \mathbf{I}_3 is the 3×3 identity matrix, the deviatoric operator $\text{dev}[\cdot]$ is defined as $\text{dev}[\mathbf{x}] \equiv \mathbf{x} - (\text{tr}[\mathbf{x}]/3)\mathbf{I}_3$, the left Cauchy-Green elastic strain tensor is denoted \mathbf{b}^E , and J is the determinant of the liquid's deformation gradient as introduced in the opening section of this part.

The term $\mathbf{f}_{\text{ext,f}}$ in Eq. (13.67) represents the external forces applied on the liquid, including the drag force produced by relative motion between the liquid and submerged strands, and interaction forces between the bulk liquid and surface flow on the strands. Both forces will be elaborated later.

Mass Conservation

According to mixture theory [137, 190], the mass conservation law for bulk liquid in the mixture is

$$\frac{D\mathbf{u}_f\phi_f\rho_f}{Dt} + \phi_f\rho_f\nabla \cdot \mathbf{u}_f = 0, \quad (13.69)$$

where ρ_f is the non-Newtonian liquid's density and $\phi_f\rho_f$ is the liquid's effective density in the mixture. Both ρ_f and ϕ_f are spatially varying, though for brevity we do not indicate their dependence on \mathbf{x} in (13.69). Many non-Newtonian materials (such as milk cream) are compressible, and therefore $\rho_f(\mathbf{x})$ depends on the determinant

$J(\mathbf{x})$ of the liquid's deformation gradient, which measures how much an infinitesimal liquid region has compressed or expanded—that is, $J(\mathbf{x}) = \rho_{f,0}/\rho_f(\mathbf{x})$, where $\rho_{f,0}$ is the liquid's (rest) material density. Using J to express ρ_f in (13.69), we obtain the material derivative of J ,

$$\frac{D_{\mathbf{u}_f} J}{Dt} = J \left(\phi_f^{-1} \frac{D_{\mathbf{u}_f} \phi_f}{Dt} + \nabla \cdot \mathbf{u}_f \right), \quad (13.70)$$

describing how the liquid's local volume changes over time. The local volume change also causes a change in liquid pressure, which we will derive later by leveraging (13.70).

Remark: Consistency with non-mixture liquid. The momentum and mass conservation laws derived from mixture theory are general enough to describe non-mixture fluids as well. For example, when there is no strand in the mixture (i.e., $\phi_f = 1$ everywhere), the first term on the right hand side of (13.70) vanishes, and Eq. (13.70) becomes the standard mass conservation equation for a single-phase material.

13.5 Coupling Forces

We now focus on the coupling forces that allow bulk liquid, discrete strands, and surface flows to interact each other. In particular, we present force models between bulk liquid and submerged strands and between bulk liquid and the strands' surface flows. Note that the interaction force between strands and their surface flows has already been discussed, emerging as the inertial force in the last term of (13.50).

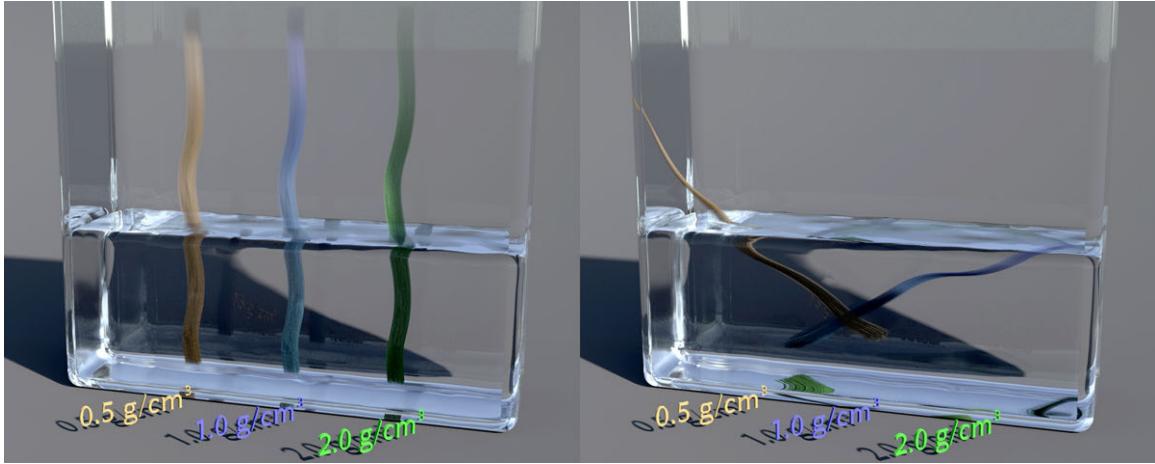


Figure 13.5: **Buoyancy**—left: 0.0s, right: 4.0s. With the pressure gradient computed using our method, we can correctly handle the buoyancy of strands in water (mass density $1.0\text{g}/\text{cm}^3$), where the light brown strands (mass density $0.5\text{g}/\text{cm}^3$) float, neutrally buoyant blue strands (mass density $1.0\text{g}/\text{cm}^3$) drift, and heavy green strands (mass density $2.0\text{g}/\text{cm}^3$) sink.

Pressure

First, we derive the pressure for a non-Newtonian mixture. An expression for pressure is needed for two reasons: i) pressure drives the bulk liquid’s motion by contributing to its stress in the momentum equation (13.68), and ii) the pressure gradient produces forces on the submerged strands, contributing to the external force term in (13.50).

Pressure is caused by volume change of the liquid. Formally, as explained in the opening section of this part, it is the negated derivative of dilational potential energy W_v with respect to elastic volume change J^E , i.e., $p \equiv -\partial W_v / \partial J^E$ — akin to how a spring force is related to the spring’s energy. There exist many models for the dilational potential energy, and our bulk liquid and strand models presented in earlier sections could incorporate any of them (through p). In our examples, we choose a modified neo-Hookean model [215], following the work of Yue et al. [261].

For the sake of numerical stability our simulation uses an implicit integration scheme which requires, for its implicit pressure solve, the pressure's material derivative,

$$\frac{D_{\mathbf{u}_f} p}{Dt} = -\frac{\partial^2 W_v}{\partial J^E \partial J^E} \frac{D_{\mathbf{u}_f} J^E}{Dt}. \quad (13.71)$$

Substituting (13.70) and the dilational potential energy (13.4) into (13.71) reveals that the pressure's material derivative is related to both the liquid and strand velocities in the mixture:

$$\frac{D_{\mathbf{u}_f} p}{Dt} = -\frac{\partial^2 W_v}{\partial J^E \partial J^E} J^E \left(\phi_f^{-1} \frac{D_{\mathbf{u}_f} \phi_f}{Dt} + \nabla \cdot \mathbf{u}_f - \frac{1}{J^P} \frac{D_{\mathbf{u}_f} J^P}{Dt} \right) \quad (13.72a)$$

$$= -\frac{\kappa}{2} J^* \left(\frac{1}{\phi_f V^*} \sum_i V_{s,i} \nabla w_{R,i}(\mathbf{x}) \cdot (\mathbf{u}_{s,i} - \mathbf{u}_f) + \nabla \cdot \mathbf{u}_f \right), \quad (13.72b)$$

where J^* is a shorthand for $J^* \equiv (J^E + (J^E)^{-1})$. The second equality utilizes the material derivative of the liquid volume fraction, $D_{\mathbf{u}_f} \phi_f / Dt$, estimated in (13.58) along with the fact that plastic flow is often isochoric, i.e., $J^P = 1$, even under very high pressure [41, 42]. Eq. (13.72b) will be discretized in §14 to solve for the pressure.

Pressure Force on Strands. When strands are mixed with the liquid, the spatial gradient of the pressure also induces a force $\mathbf{f}_{p,i}$ on each submerged strand element i through $\mathbf{f}_{p,i} = V_{s,i} \nabla p(\mathbf{x}_i)$, where $V_{s,i}$ is the volume of the discrete strand element i centered at position \mathbf{x}_i . The force $\mathbf{f}_{p,i}$ contributes in part to the external force $\mathbf{f}_{\text{ext},s}$ in (13.50). As a result, we are able to correctly capture buoyancy effects (see Figure 13.5).

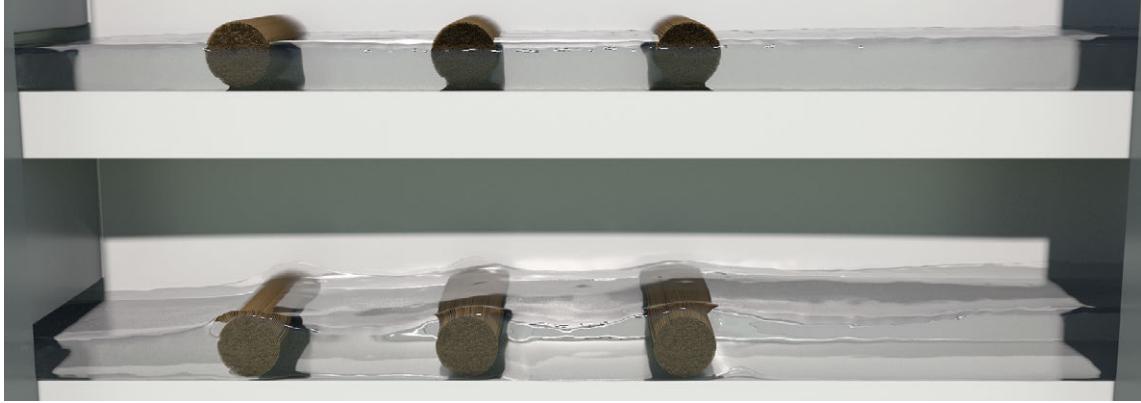


Figure 13.6: **Comparison between varying and fixed volume fraction (in the absence of drag).** Liquid flows from left to right through strands fixed in place. **Top: without the volume fraction considered.** The liquid flow does not change its volume despite part of the space being occupied by strands; **Bottom: with the volume fraction considered.** The liquid will expand naturally as it passes through strands.

Remark: Comparison to Single-Phase Liquid. Our pressure equation differs from that used in previous work (e.g., [226]) wherein only a single-phase liquid is considered. In particular, our model captures the mixture's effective volume change caused by the change of liquid volume fraction (as seen from the first term on the right hand side of (13.70)). This volume change in turn contributes to the pressure's material derivative (13.72a). In comparison to a single-phase liquid model, our mixture-theory-based model is able to capture richer liquid-strand interactions as illustrated in Figure 13.6.

Derivation: Comparison to Incompressible Mixture Models. In a prior work [85] and **Part I**, models for *incompressible* mixtures have been developed. Below we show that the incompressible mixture is a special case of our model when the material stiffness κ (e.g., used in (13.72b)) approaches infinity. Therefore, our model

is able to simulate liquids across a wider range of stiffnesses (see Figure 13.7).

Our derivation is not limited to a specific model. For an arbitrary non-zero dilational potential energy whose second-order derivative is denoted as $\kappa g(J^E)$ below, we may rewrite (13.72a) into the following form:

$$\frac{D_{\mathbf{u}_f} p}{Dt} = -\kappa g(J^E) \left(\phi_f^{-1} \frac{D_{\mathbf{u}_f} \phi_f}{Dt} + \nabla \cdot \mathbf{u}_f - \frac{1}{J^P} \frac{D_{\mathbf{u}_f} J^P}{Dt} \right) \quad (13.73)$$

For incompressible mixture we have the liquid material stiffness $\kappa \rightarrow \infty$ and $J^P = 1$.

By dividing both sides with κ and taking the infinite limit of κ , we have

$$\lim_{\kappa \rightarrow \infty} \left(\kappa^{-1} \frac{D_{\mathbf{u}_f} p}{Dt} \right) = 0 = -g(J^E) \left(\phi_f^{-1} \frac{D_{\mathbf{u}_f} \phi_f}{Dt} + \nabla \cdot \mathbf{u}_f \right) \quad (13.74)$$

or simply (since g is non-zero)

$$\frac{D_{\mathbf{u}_f} \phi_f}{Dt} + \phi_f \nabla \cdot \mathbf{u}_f = 0. \quad (13.75)$$

which can be rewritten by expanding the material derivative $D_{\mathbf{u}_f} \phi_f / Dt$, as following

$$\frac{\partial \phi_f}{\partial t} + \nabla \cdot (\phi_f \mathbf{u}_f) = 0. \quad (13.76)$$

This is exactly the continuity equation for liquid in mixture with constant mass density (see, e.g., [8]). After (13.62) is added with (13.76), we obtain (3.4) for the



Figure 13.7: **Simulated cream with a wide range of bulk moduli (κ).** Cream with a lower bulk modulus κ shrinks or dilates more easily during the simulation, and has a larger volume in the steady state.

incompressible mixture on which is focused in the prior work [85] and **Part I**.

$$\nabla \cdot (\phi_f \mathbf{u}_f + \phi_s \mathbf{u}_s) = 0. \quad (13.77)$$

where $\phi_f \equiv 1 - \phi_s$. ■

Remark: Generality. We obtain (13.72a) based only on compressible mixture theory; no specific properties of non-Newtonian fluids or the dilational potential model are needed. In other words, Eq. (13.72a) is sufficiently general to simulate other types of liquid materials (e.g., see [38, 219]). In the same vein, the discrete solids mixed into the liquid in our model need not necessarily be strands; other types of solids, such as gravel, sand, and clay, could be readily simulated as well.

Drag Force

Next, we present a model for computing drag forces between the liquid and submerged strands. The drag force contributes to the external force terms for both strands (13.50) and the liquid (13.67).

We choose to employ a popular drag model proposed by Di Felice [73]. This is a simple velocity-dependent model that is nevertheless flexible enough to support a range of materials and rheologies [139, 146]. Concretely, the drag force of a discrete strand element i centered at \mathbf{x}_i is expressed as

$$\mathbf{f}_{\text{drag,s},i} = \frac{1}{2}\rho_f(\mathbf{x}_i)C_d A_{\perp,i} \|\mathbf{u}_f(\mathbf{x}_i) - \mathbf{u}_{s,i}\|_2 \phi_f^{-\chi_i} (\mathbf{u}_f(\mathbf{x}_i) - \mathbf{u}_{s,i}), \quad (13.78)$$

where \mathbf{u}_f and $\mathbf{u}_{s,i}$ are liquid and strand element velocities (as defined previously), $A_{\perp,i}$ is the area of the strand element i projected on the plane perpendicular to the relative velocity vector $\mathbf{u}_f - \mathbf{u}_s$, C_d is the drag coefficient, and the parameter χ_i takes the empirical form

$$\chi_i = 3.7 - 0.65 \exp \left[-\frac{1}{2}(1.5 - \log \text{Re}_{p,i})^2 \right], \quad (13.79)$$

in which $\text{Re}_{p,i}$ is i -th strand element's *particle Reynolds number*, whose specific formula is given in (13.82).

A key parameter in this model is the drag coefficient C_d . Experiments have shown that C_d can be taken as a constant value if the solid elements are spherical particles

and the liquid has a low viscosity and a high Reynolds number [264]. While recent work on simulating wet sand [85] used this model with a constant C_d , we found that a constant C_d in (13.78) limits the model's generalizability to different liquid materials, as illustrated in Figure 13.9.

We adopt a formulation for C_d developed by Renaud [197]. It has been extensively verified for predicting drag forces between irregularly shaped solid elements and liquids, both Newtonian and non-Newtonian, with a Reynolds number up to 1500 (see [3, 54, 193]). Details are provided below, and its efficacy is demonstrated in Figure 13.9.

Drag Coefficient. The drag coefficient for rod element i has the following form [193]:

$$C_{d,i} = C_{d0,i} + \frac{A_{c,i}}{A_{\perp,i}} C_{d\infty} (C_{d0,i})^{2\beta} k \left[\frac{6Xb}{6Xb + C_{d0,i}} \right]^\beta + C_{d\infty} \left[\frac{6Xb}{6Xb + 128C_{d0,i}} \right], \quad (13.80)$$

where $A_{c,i}$ is the surface area of the i -th element, $A_{\perp,i}$ is the area of the i -th discrete element projected in the direction of relative velocity, and

$$C_{d0,i} \equiv \frac{24X}{Re_{p,i}}, \quad (13.81a)$$

$$C_{d\infty} \equiv 0.44, \quad (13.81b)$$

$$\alpha \equiv \frac{3}{n^2 + n + 1}, \quad (13.81c)$$

$$X \equiv 6^{(n-1)/2} \alpha^{n+1}, \quad (13.81d)$$

$$b \equiv \exp [3(\alpha - \ln 6)] , \quad (13.81e)$$

$$k \equiv \frac{3-\alpha}{6\alpha} \exp \left(\frac{3-\alpha}{2\alpha} \ln 3 \right) , \quad (13.81f)$$

$$\beta \equiv \frac{11}{48} \sqrt{6} \left[1 - \exp \left[\left(\frac{3-\alpha}{2\alpha} \right)^2 \ln \left(\frac{\sqrt{6}-1}{\sqrt{6}} \right) \right] \right] . \quad (13.81g)$$

where $\text{Re}_{p,i}$ is the *particle Reynolds number* (see below) of i -th element, and n is the flow behavior index.

The particle Reynolds number for a Herschel-Bulkley liquid. The drag coefficient proposed in (13.80) is originally developed for a power-law liquid. Nevertheless, Atapattu et al. [15] showed that a drag coefficient for a power-law liquid can also be generalized to a Herschel-Bulkley liquid by adopting a modified particle Reynolds number. Using the von Mises yield condition (13.11), the particle Reynolds number for i -th rod element reads [15, 73]:

$$\text{Re}_{p,i} \equiv \frac{\phi_f \rho_f d_{p,i}^n \| \mathbf{u}_f - \mathbf{u}_{s,i} \|^2}{\eta \| \mathbf{u}_f - \mathbf{u}_{s,i} \|^n + \sqrt{\frac{2}{3}} \tau_Y d_{p,i}^n} \quad (13.82)$$

where $d_{p,i}$ is the diameter of a circle that has the area equivalent to $A_{\perp,i}$, i.e., $d_{p,i} = 2\sqrt{A_{\perp,i}/\pi}$.

We plot the drag coefficient over its different parameters in Figure 13.8, where we can observe that the drag coefficient increases over the flow consistency index (or viscosity) η , the yield stress τ_Y , the flow behavior index n , and decreases over the relative velocity between liquid and strand element.

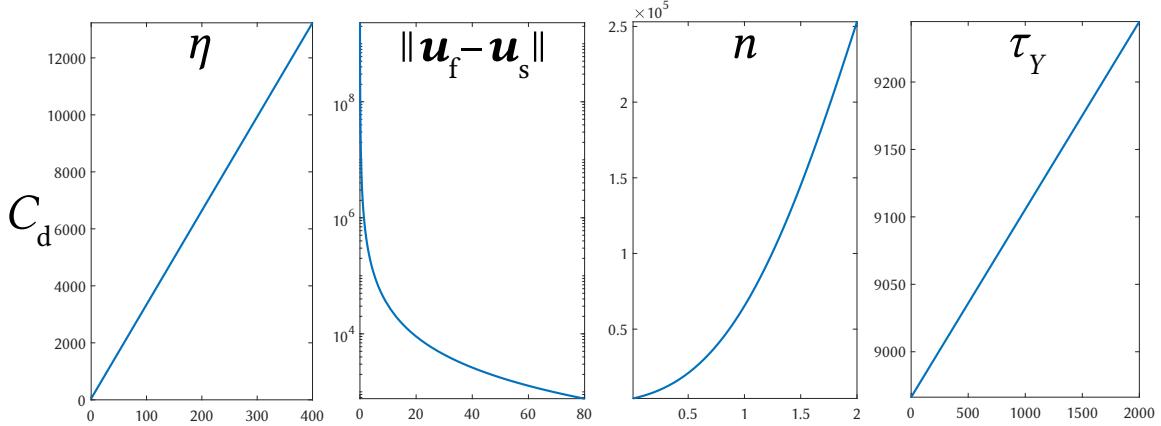


Figure 13.8: **Drag coefficient over its different parameters.** Data is acquired through varying one parameter and fixing the others with the parameters of shaving cream.

Remark: Degenerated to Drag Coefficient in Newtonian liquid. In a Newtonian liquid, the flow behavior index $n = 1$. Then in the equations above, $X = 1$, $\alpha = 1$, and $k = 1$, where (13.80) is precisely consistent with the drag coefficient for the irregular particles in a Newtonian liquid [161].

We conclude by homogenizing the drag forces experienced by the discrete strand elements to apply the corresponding force on the liquid. Homogenization is performed by taking a kernel-weighted average of the drag on the discrete elements, which is,

$$\mathbf{f}_{\text{drag,f}}(\mathbf{x}) = \frac{1}{V^*} \sum_i w_{R,i}(\mathbf{x}) \mathbf{f}_{\text{drag,s},i}. \quad (13.83)$$

Remark: Sanity Check through Dimensional Analysis. Dimensional analysis provides a useful sanity check for us. As mentioned in the opening section of this part, the flow consistency index has the physical unit $\text{Ba} \cdot \text{s}^n$, and the yield condition τ_Y has the physical unit of pressure (Ba). The divisor in (13.82) then has physical unit

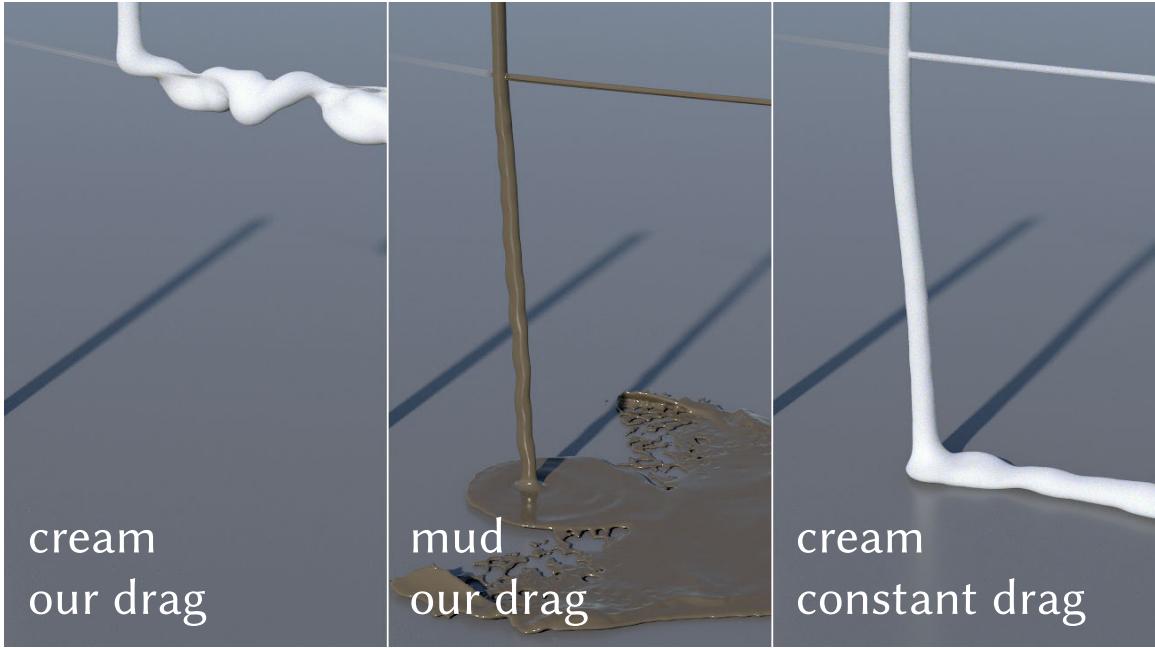


Figure 13.9: **Comparison of different drag coefficients.** Streams of shaving cream and mud are poured onto seven hanging strands. **Left and middle:** Our drag model yields very different interaction behavior depending on the liquid type; **Right:** A constant drag coefficient, on the other hand, produces a drag effect for the shaving cream similar to that of mud, since the yield stress and viscosity are not considered.

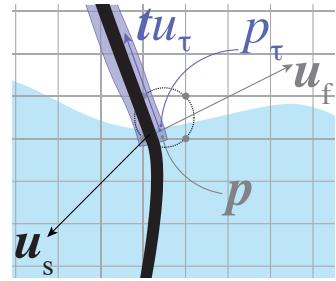
$\text{Ba} \cdot \text{cm}^n$, or $\text{g} \cdot \text{cm}^{n-1} \cdot \text{s}^{-2}$, which exactly cancels with the physical unit of the dividend.

Hence Re_p is indeed a unitless number. Obviously all the parameters in (13.81) are unitless, thence C_d is unitless. Furthermore, we have χ defined in (13.79) unitless. Therefore $\mathbf{f}_{\text{drag},s}$ has the physical unit $\text{g} \cdot \text{cm} \cdot \text{s}^{-2}$, or a dyne, which is precisely the unit of a force. In (13.83), the weighted sum also has the unit of a force. With the divisor V^* applied, $\mathbf{f}_{\text{drag},f}$ has the unit of a *force density*, which is exactly the unit of both sides of (13.67).

Constraints Between Bulk Liquid and Surface Flows

Lastly, we consider the interactions at the interface between the bulk liquid and the reduced surface flows. Mathematically, since the momentum conservation law is a second-order partial differential equation, it needs two boundary conditions on the interface to couple the surface flow with the bulk liquid.

Pressure Boundary Condition. The first condition requires that the pressure in the surface flow and the bulk liquid agree at the interface (see adjacent figure). As discussed in the prior section (see **Non-Newtonian Strand Surface**



Flow), the internal stress of the surface flow depends on the reduced shear strain c_τ^E . Thus, the surface flow pressure p_τ at the interface position \mathbf{x}_0 is $p_\tau(\mathbf{x}_0) = -\mu c_\tau^E(\mathbf{x}_0)$, where μ is the liquid's shear modulus. Let $p_f(\mathbf{x}_0)$ be the bulk liquid pressure at the same interface position. Then, the Dirichlet pressure condition is

$$p_f(\mathbf{x}_0) = p_\tau(\mathbf{x}_0). \quad (13.84)$$

This boundary condition will be used in our numerical pressure solve in §14.

Velocity Boundary Condition. The second condition is similar to the liquid-solid boundary condition in a typical liquid simulation, demanding velocity agreement at the interface, namely,

$$u_\tau(\mathbf{x}_0) = \mathbf{t}^T(\mathbf{u}_f(\mathbf{x}_0) - \mathbf{u}_{s,i}), \quad (13.85)$$

where the left hand side is the surface flow velocity, while the right hand side indicates the projected velocity difference (along strand direction \mathbf{t}) between the bulk liquid and the strand element i at \mathbf{x}_0 .

We enforce this boundary condition by applying penalty forces on discrete strand elements and the bulk liquid. This approach is similar to the classic immersed boundary method for simulating liquid-solid coupling [181]. The penalty force at a strand element i is computed as

$$\mathbf{f}_{\text{eq},i} = -\frac{1}{h}m_{\tau,i}(\mathbf{t}_i u_{\tau,i} + \mathbf{u}_{s,i} - \mathbf{u}_f), \quad (13.86)$$

where h is the time step size, $m_{\tau,i}$ is the mass of liquid on strand element i , and $u_{\tau,i}$ is the flow velocity at the strand element i . In the strand's momentum equation, (13.50), $\mathbf{f}_{\text{eq},i}$ serves as a part of the external force $\mathbf{f}_{\text{ext,s}}$. The corresponding penalty force on the liquid is the homogenization of the discrete element forces $\mathbf{f}_{\text{eq},i}$ (i.e., $\mathbf{f}_{\text{eq,f}}(\mathbf{x}) = -\sum_i w_{R,i}(\mathbf{x})\mathbf{f}_{\text{eq},i}$). The force $\mathbf{f}_{\text{eq,f}}$ contributes to the external force $\mathbf{f}_{\text{ext,f}}$ in the liquid's momentum equation (13.67).

13.6 Contact Between Wet Strands

Inter-strand contacts also significantly affect strand motion. In the context of wet strands, a straightforward approach to resolve contacts is through penalty forces. In **Part II** and prior works [148, 149], the penalty force can produce both repulsive and cohesive effects, because of the liquid bridge that forms between two strands. However,

penalty force methods suffer from several critical disadvantages: Wet strands, such as wet hair, are known to exhibit a strong frictional effect [32], but it remains unclear how to incorporate a principled friction model into penalty methods. Moreover, penalty forces on strands can become strongly cohesive due to the liquid bridge in between, quickly causing strand penetration during the simulation (see Figure 13.10). The nature of non-Newtonian liquids makes matters worse, as such liquids may produce even stronger strand cohesion due to non-Newtonian elasticity.

In light of these factors, we instead seek to resolve contacts through a constraint-based method that solves a second-order Coulomb cone (SOCC) problem for both contact and friction forces, following the prior works [30, 69, 118, 167]. This approach ensures contact resolution without strand penetration and ensures that the friction forces follow Coulomb’s law of friction precisely. However, it traditionally assumes that the contact force can be repulsive, but not cohesive, which is not an appropriate assumption for our setting. To address this limitation, we adopt a shifted cone [2, 65, 66, 87, 119, 125, 130, 195, 230] and specialize it to simulate strands covered with cohesive viscoplastic material.

More specifically, we augment the shifted cone in two aspects:

1. Two strands can approach until their surfaces touch each other, but won’t separate until the liquid bridge breaks. We treated this phenomenon by introducing contact hysteresis, where we used different distance criteria for collision detection according to the relative velocity of strands. This method is different with

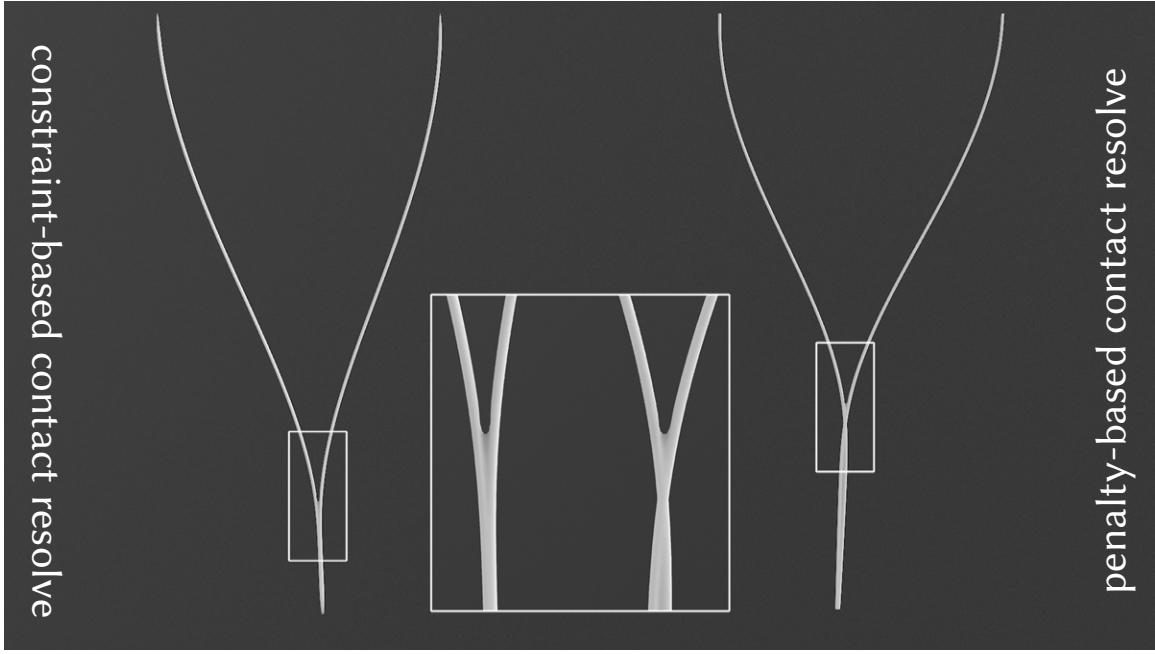


Figure 13.10: **Comparison between constraint- and penalty-based collision.** Even with relatively large cohesion forces (cream), our method (**Left**) can correctly handle contact with cohesion, without the tunneling problem seen in the penalty method (**Right**).

the prior work [119] only use a shifted cone when two colliders are separating for the first time.

2. The adhesion force includes two components: the capillary and the viscoplastic part. We calculate the capillary part through reusing the equation derived in **Part I**, and estimate the viscoplastic part, specifically for shear-dependent flow, by using the flow rate function in Herschel-Bulkley model [102].

A Shifted Second-Order Cone

Before introducing our augmentation, we first show the idea behind the shifted second-order coulomb cone following prior works [2, 65, 66, 87, 119, 125, 130, 195, 230], using the notation in the works of Daviet and Bertails-Descoubes [30, 69]. Given a contact

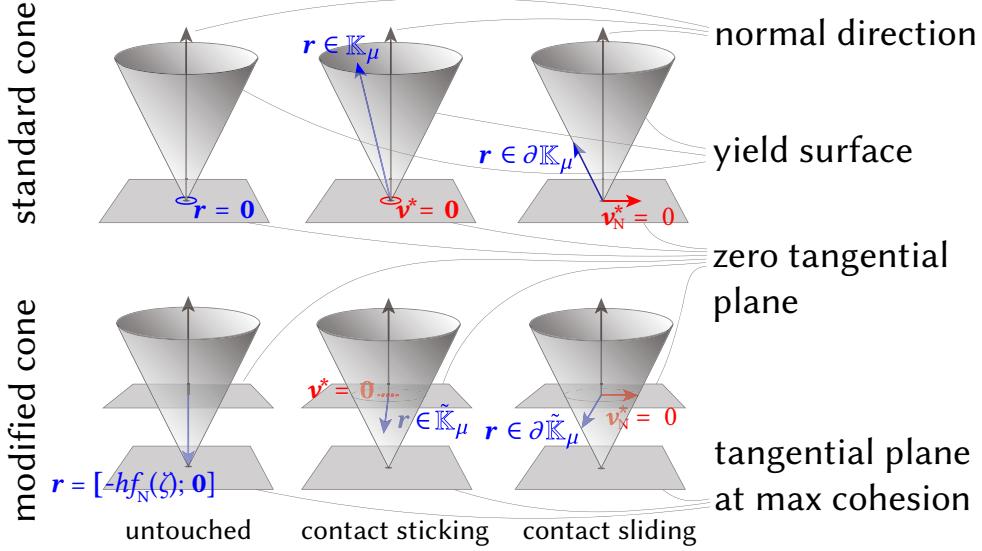


Figure 13.11: **Top: Regular second-order Coulomb cone (SOCC) when the strands are dry.** An impulse \mathbf{r} is constrained by the cone when two rods collide. For the tangential part of \mathbf{r} (denoted r_T) we have either $\mathbf{v}^* = 0$ when $r_T \in K_\mu$, producing static friction (sticking), or $\|\mathbf{v}_T^*\| > 0$ when r_T reaches the yield surface, i.e., $r_T \in \partial K_\mu$, producing dynamic friction (sliding). The relative velocity in the normal direction will be eliminated for either the sticking or sliding case. **Bottom: Modified SOCC applied when the strands are covered with liquid.** The impulse \mathbf{r} is offset and may yield cohesion: as long as the cohesive impulse is less than hf_N , i.e., $r_N > -hf_N$, we have $v_N^* = 0$. Due to the offset of the cone, an additional frictional effect will be induced, which, physically, corresponds to the shear stress of the cohesive material. Finally, when $f_N = 0$, the modified SOCC degenerates to a standard one [2, 65, 66, 87, 119, 125, 130, 195, 230].

point where the surface normal is denoted by \mathbf{n} , we consider the contact impulse (rather than force) across a time step size h . The contact impulse, including both the normal and frictional impulses, is denoted by \mathbf{r} . We use the subscripts N and T to denote the vector components along the normal direction and on the tangential plane, respectively. For example, $r_N = \mathbf{n}^T \mathbf{r}$ is the normal impulse (i.e., a scalar) and $\mathbf{r}_T = (\mathbf{I}_3 - \mathbf{n}\mathbf{n}^T)\mathbf{r}$ is the tangential (frictional) impulse (i.e., a vector). We also express \mathbf{r} explicitly using its two components as $[r_N; \mathbf{r}_T]$.

With this notation, the regular SOCC \mathbb{K}_μ , defined under a friction coefficient μ ,

is the set of vectors containing all possible contact impulses satisfying Coulomb's law of friction,

$$\mathbb{K}_\mu = \{[r_N; \mathbf{r}_T] \mid \mu x_N \geq \|\mathbf{x}_T\|_2\}, \quad (13.87)$$

which is visualized in the top row of Figure 13.11.

Next, we consider the shifted SOCC [2, 65, 66, 87, 119, 125, 130, 195, 230] which incorporates cohesive forces between wet strands. When a liquid bridge connects two wet strands, a cohesive force arises due to the liquid's surface tension and elasticity. Denoted as $f_N(\zeta)$, this force depends on the strand distance ζ and points along the normal direction \mathbf{n} , and its details will be discussed shortly. Now the total impulse \mathbf{r}_s includes both the contact impulse \mathbf{r} and the cohesive impulse $h f_N(\zeta)$, so

$$\mathbf{r}_s = \mathbf{r} + h [f_N(\zeta); \mathbf{0}]. \quad (13.88)$$

The normal and tangential components of the total impulse \mathbf{r}_s must satisfy Coulomb's law of friction, that is, \mathbf{r}_s must reside in \mathbb{K}_μ . This means that all possible contact impulses \mathbf{r} form a different SOCC (denoted as $\tilde{\mathbb{K}}_\mu$), that is like \mathbb{K}_μ but translated along the normal direction by $-h [f_N(\zeta); \mathbf{0}]$. The bottom row of Figure 13.11 shows a visualization of $\tilde{\mathbb{K}}_\mu$.

Given $\tilde{\mathbb{K}}_\mu$, there are several outcomes for the relative velocity \mathbf{v}^* of two strand elements after their contact is resolved, depending on where the contact impulse locates in the shifted SOCC [2, 65, 66, 87, 119, 125, 130, 195, 230]:

1. Untouched: $\zeta > \zeta_0$, $\mathbf{r}_s = 0$, and \mathbf{v}^* is not affected,
2. Contact sticking: $\zeta = \zeta_0$, $\mathbf{r}_s \in \mathbb{K}_\mu$, $\mathbf{r} \in \tilde{\mathbb{K}}_\mu$, and $\mathbf{v}^* = 0$,
3. Contact sliding: $\zeta = \zeta_0$, $\mathbf{r}_s \in \partial\mathbb{K}_\mu$, $\mathbf{r} \in \partial\tilde{\mathbb{K}}_\mu$, $v_N^* = 0$, and $\exists \alpha \in \mathbb{R}_+, \mathbf{v}_T^* = -\alpha \mathbf{r}_{s,T}$.

Here $\partial\mathbb{K}_\mu$ denotes the boundary of the SOCC \mathbb{K}_μ , and ζ_0 is a critical distance between two strand elements, indicating when the impulse \mathbf{r}_s occurs. The specific value of ζ_0 deserves some careful reasoning, as will be discussed next.

Contact Hysteresis on Both Colliding Distance and Relative Velocity

In the shifted SOCC model [2, 65, 66, 87, 119, 125, 130, 195, 230], two strand elements stop moving towards one another when their distance reaches ζ_0 . In reality, two strands can approach until their surfaces touch each other, regardless of their surface flow thickness. This observation suggests that ζ_0 should be set as $2r$ where r is the strand radius. On the other hand, when two strand elements move apart, the cohesive force persists until the liquid bridge between them breaks. This observation, by contrast, suggests that ζ_0 should be a value related to the surface flow thickness.

As for the contact hysteresis, the prior work [119] only uses shifted SOCC [2, 65, 66, 87, 119, 125, 130, 195, 230] when two colliders are separating for the first time to prevent cracked parts from re-cohesion, which targets at different application than ours and cannot resolve the paradox mentioned above. We resolve it by introducing

contact hysteresis based on both distance and relative velocity between two colliders, in which the critical distance ζ_0 for approaching strands is different from that of separating strands:

1. Approaching: when $v_N < 0$, $\zeta_0 = 2r$,
2. Separating: when $v_N \geq 0$, $\zeta_0 = 2r + (1 + 0.5\theta)\sqrt{A_L}$, where A_L is the total area of the cross section of the liquid bridge, and θ is the *contact angle* determined by the liquid and strand materials.

In case (b), ζ_0 is set to be the distance at which the liquid bridge breaks. To estimate this distance value, we follow the formula by Lian et al. [147]. The hysteretic contact force profile is illustrated in Figure 13.12.

Solver. Similar to the standard SOCC problem, the shifted SOCC problem [2, 65, 66, 87, 119, 125, 130, 195, 230] can be reformulated as a root-finding problem and solved by So-Bogus [30, 66, 69]. Next, we show the derivation following Kaufman et al. [129] as well as the implementation in ADONIS [130] and So-Bogus [30, 66, 69].

After adopting the change of variables proposed by De Saxcé and Feng [71] and Daviet et al. [69], we have following complementarity formulation of a self-dual cone $K \equiv K_{\mu=1}$:

$$K \ni \hat{\mathbf{v}} \perp \hat{\mathbf{r}} \in K \quad (13.89)$$

where

$$\hat{\mathbf{r}} \equiv [\mu r_{s,N}; \mathbf{r}_{s,T}], \hat{\mathbf{v}} \equiv \bar{m}_s[\tilde{v}_N; \mu \tilde{\mathbf{v}}_T], \quad (13.90)$$

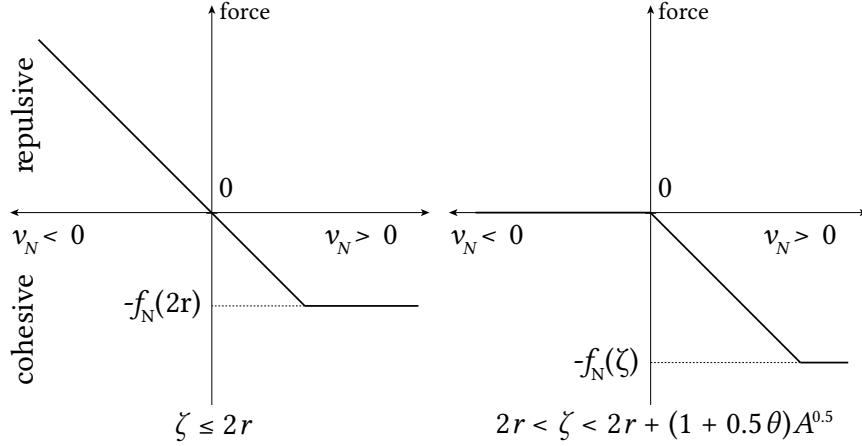


Figure 13.12: **The profile of the contact-cohesion force in the normal direction.** **Left:** the solid parts of the strands touch each other. When the two strands tend to approach each other with *negative* unconstrained normal velocity, there will be a repulsive force; when the two strands tend to depart from each other with *positive* unconstrained normal velocity, there will be a cohesive force. We evaluate the cohesive force f_N at $2r$ for the case $\zeta < 2r$ to avoid singularity. **Right:** the solid parts of the strands do not touch, but the liquid bridge exists. The two strands can freely move toward each other with *negative* unconstrained normal velocity, but a cohesive force will stick them together as long as they tend to separate with *positive* unconstrained normal velocity. In both cases, whenever there is a cohesive force, the relative normal velocity after applying constraint will be *zero* unless the maximal cohesive force is reached.

and

$$\tilde{\mathbf{v}} \equiv \mathbf{v} + \mu \|\mathbf{v}_T\| \mathbf{n}. \quad (13.91)$$

In (13.90), the mass $\bar{m}_s = (m_{s,1} + m_{s,2})/2$ is the averaged mass of the elements in contact, which scales the velocity so that $\bar{m}_s \hat{\mathbf{v}}$ has the same physical units as $\hat{\mathbf{r}}$.

Then solving for $\hat{\mathbf{r}}$ in (13.89) can be converted into a root-finding problem [69, 83]

$$K \ni \mathbf{x} \perp \mathbf{y} \in K \Leftrightarrow \mathbf{f}^{\text{MFB}}(x, y) = \mathbf{0} \quad (13.92)$$

where

$$f^{\text{MFB}}(\mathbf{x}, \mathbf{y}) \equiv \mathbf{x} \circ \mathbf{y} - (\mathbf{x} \circ \mathbf{x} + \mathbf{y} \circ \mathbf{y})^{\frac{1}{2}} \quad (13.93)$$

is known as the *modified Fischer-Burmeister* (MFB) function, and the operator \circ is the Jordan product defined as

$$\mathbf{x} \circ \mathbf{y} \equiv [\mathbf{x} \cdot \mathbf{y}; x_N \mathbf{y}_T + y_N \mathbf{x}_T]. \quad (13.94)$$

Remark. When two strands approach (i.e., $v_N < 0$), our contact hysteresis does not induce cohesive forces even if the liquid bridge has formed, unless the strand surfaces touch each other. However, the cohesive forces appear when they move away from each other. While not entirely accurate, our model nevertheless captures interesting wet strand behaviors, such as the formation of strand bundles. This is because when (sufficiently close) strands attempt to move apart, the cohesive force tends to prevent them from separating. The lack of cohesion as strands approach has an advantage in practice: the cohesion force does not act to accelerate the negative normal velocity and thereby increase the speed of the collision. Otherwise, the cohesive force would render the system much stiffer, making strand penetration or tunneling much harder to avoid.

Cohesive Force

Cohesion is usually caused by the capillary surface energy of the liquid, be it Newtonian or non-Newtonian. In addition, the non-Newtonian liquid may introduce an ex-

tra cohesive force: when two strands move apart and the liquid bridge gets stretched, the non-Newtonian liquid will experience a viscoplastic deformation until the bridge breaks. During a stretching motion, the elastic stress effectively produces an extra cohesive force on the strands. On the other hand, the non-Newtonian liquid bridge possesses a maximal stress, after which point the liquid begins to yield and convert its elastic strain into plastic strain. Thus the elastic stress is limited by the maximal stress.

Therefore, we model the cohesive force f_N with two components: the capillary part $f_{N,c}$ and viscoplastic part $f_{N,v}$, that is,

$$f_N = f_{N,v} + f_{N,c}. \quad (13.95)$$

The capillary part $f_{N,c}$ is the cohesive force proposed in **Part II**, computed through (8.11).

On the other hand, the viscoplastic part $f_{N,v}$ can be estimated from the flow rate function (13.13). In particular, following the simple relationship between a stress and a force [14], we compute $f_{N,v}$ by estimating the stress applied on the cross sectional area A_c of the liquid bridge:

$$f_{N,v} = A_c \left[\sqrt{\frac{2}{3}} \tau_Y + \eta \gamma^n \right]. \quad (13.96)$$

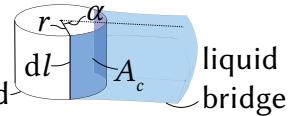
where τ_Y is the yield stress, η is the flow consistency index, and n is the flow behavior



Figure 13.13: Contact between strands with surface flows of different materials – **Left:** mud; **Right:** milk cream.

index.

To use (13.96) in practice, we estimate the flow rate $\gamma = \partial(\mathbf{n}^T \mathbf{u}_s)/\partial \mathbf{n}$ using finite differences, and A_c using the



area where the liquid contacts the strand, i.e., $A_c = 2r\alpha dl$ where dl is the wet length and α is the angle between the direction toward the liquid bridge and the direction toward the liquid/solid boundary.

Chapter 14

Discretization of the Strands and the Shear-Dependent Liquid

With our continuous physical models in hand, we can proceed to discretization. We adopt four types of Lagrangian variables to discretize the simulated geometry: 1) traditional MPM particles, or *particles* for brevity, are used to discretize the bulk liquid; 2) Lagrangian rod vertices, or *vertices*, are used to discretize the elastic rods; 3) Lagrangian rod elements (segments), or *elements*, are used to set up a staggered discretization of the surface flow on the rods. In addition, the bulk liquid is solved on an Eulerian staggered grid with velocities stored on cell *faces* and pressure stored on cell *centers*, similar to the *augmented material point method* (AMPM) [226]. Because our discretization shares similarities with other approaches for MPM [68, 85, 121, 123, 226, 230] and discrete elastic rods [26, 28, 129], in the following we briefly cover the common aspects while emphasizing the novel aspects of our approach in detail.

We adopt the notation N_p , N_v , N_e , N_k , N_g , and N_c to indicate the number of particles, strand vertices, strand elements, contacts, grid faces and grid centers. We adopt the superscript t to indicate variables that are known at the beginning of the current time step, i to indicate variables in the i -th Newton iteration, and $t + 1$ to

indicate variables to be solved for at the end of the time step. The time step size is denoted as h .

14.1 Discrete Constrained Dynamics

A naive but accurate approach would be to solve the strand, the bulk liquid, and the surface flow simultaneously. Unfortunately, since the velocities of these three components are tightly coupled into a stiff, non-smooth, and non-symmetric system, it can be very difficult to solve in practice, especially as sufficient nonlinearity for collisions is usually required for stability of the strands [129].

Therefore, in this part we adopt a staggered integrator and update the variables of different phases in an alternating fashion. One resulting benefit is greater ease of implementation: we can adopt existing methods for strand simulation (e.g., we adopt the method of Kaufman et al. [129]) and non-Newtonian liquids (e.g., we adopt the Herschel-Bulkley liquid model of Yue et al. [261]), and simply enhance them to support the coupling between the strands and the non-Newtonian liquid.

Strand Simulation Each time step begins with surface flow and strand simulation. We first apply semi-Lagrangian advection to the mass and velocity of the surface flow on the strand [224]. Then we integrate the strand dynamics, temporarily assuming the bulk liquid pressure is zero. The solved strand velocity is used for collision detection and as a prediction for the later pressure solve and additional surface flow dynamics.

Discretizing (13.50) leads to the discrete strand dynamics equation,

$$(\mathbf{M}_s^* + h\mathbf{D}_s)\mathbf{u}_s^{t+1} = \mathbf{M}_s^*\mathbf{u}_s^t + h(\mathbf{f}_{\text{int}}^{t+1} + \mathbf{M}_s\mathbf{g} + \mathbf{D}_s\mathbf{W}_{\text{gv}}\mathbf{u}_f^t) + \mathbf{M}_{\tau,v}^*(\mathbf{u}_s^t - \mathbf{u}_{s,-\mathbf{u}_{\tau,v}^t}^t), \quad (14.1)$$

where \mathbf{M}_s^* is the combined mass of the strand and the surface flow on it (*after* it has been advected for the current time step). The term $\mathbf{M}_{\tau,v}^*(\mathbf{u}_s^t - \mathbf{u}_{s,-\mathbf{u}_{\tau,v}^t}^t)$ is the extra rod inertia induced by the surface flow, which is also computed with semi-Lagrangian advection [224]: the backtraced velocity of the k -th vertex is defined as

$$\mathbf{u}_{s,-\mathbf{u}_{\tau,v}^t,k}^t \equiv \mathbf{u}_s(q_{s,\tau} - hu_{\tau,v}^t)_{\text{interp}} \quad (14.2)$$

where $q_{s,\tau}$ is the strand-space coordinate of the vertex, and the function $\mathbf{u}_s(x)_{\text{interp}}$ interpolates the value from \mathbf{u}_s at strand coordinate x .

To solve the discrete strand equation, we adopt a nonlinear Newton-Raphson solver [129]. For the i -th Newton iteration, the linearized equation is

$$\begin{aligned} \mathbf{C}_s \mathbf{u}_s^{i+1} &= \alpha^i (\mathbf{M}_s^*(\mathbf{u}_s^t - \mathbf{u}_s^i) + h(\mathbf{f}_{\text{int}}^t + \mathbf{M}_s^*\mathbf{g}) \\ &\quad + h\mathbf{D}_s\mathbf{W}_{\text{gv}}\mathbf{u}_f^t + \mathbf{M}_{\tau,v}^*(\mathbf{u}_s^t - \mathbf{u}_{s,-\mathbf{u}_{\tau,v}^t}^t)) + \mathbf{C}_s \mathbf{u}_s^i \end{aligned} \quad (14.3)$$

where α is the step length computed with backtracking line search [13], $\mathbf{C}_s = \mathbf{M}_s^* + h\mathbf{D}_s + h^2\mathbf{H}_s$ is the combination of mass and force Jacobian matrix, and $\mathbf{u}_{\tau,v}^t \equiv \mathbf{W}_{\text{ve}}^T \mathbf{u}_\tau^t$ is the surface flow velocity mapped to vertices.

After the pressure and surface flow have been solved, we re-integrate the dynamics

of the strand, this time using the solved pressure gradient and updated surface flow mass (hence the updated combined mass is denoted as \mathbf{M}^{t+1} below), where the i -th iteration of the momentum equation's Newton solve becomes

$$\begin{aligned}\mathbf{C}_s \mathbf{u}_s^{i+1} = & \alpha^i (\mathbf{M}_s^{t+1}(\mathbf{u}_s^t - \mathbf{u}_s^i) + h(\mathbf{f}_{\text{int}}^t + \mathbf{M}_s^{t+1}\mathbf{g}) \\ & + h\mathbf{D}_s \mathbf{W}_{\text{gv}} \mathbf{u}_f^t + \mathbf{M}_{\tau,v}^{t+1}(\mathbf{u}_s^* - \mathbf{u}_{s,-\mathbf{u}_{\tau,v}^{t+1}}^*) - h\mathbf{V}_s \mathbf{G}_{w,\text{cv}} \mathbf{p}) + \mathbf{C}_s \mathbf{u}_s^i.\end{aligned}\quad (14.4)$$

Since the surface flow velocity has been updated at this point, we compute the back-traced velocity with the updated surface flow velocity $\mathbf{u}_{\tau,v}^{t+1}$ and the (unconstrained) predicted solid velocity \mathbf{u}_s^* . We then use the solved velocity (denoted by \mathbf{u}_s^\dagger below) for contact resolution.

Finally, after the contact impulse \mathbf{r} is solved, we update the velocity with the impulse added to the right hand side, using

$$\begin{aligned}\mathbf{C}_s \mathbf{u}_s^{i+1} = & \alpha^i (\mathbf{M}_s(\mathbf{u}_s^t - \mathbf{u}_s^i) + h(\mathbf{f}_{\text{int}}^t + \mathbf{M}_s\mathbf{g}) \\ & + h\mathbf{D}_s \mathbf{W}_{\text{gv}} \mathbf{u}_f^t + \mathbf{M}_{\tau,v}(\mathbf{u}_s^\dagger - \mathbf{u}_{s,-\mathbf{u}_{\tau,v}^{t+1}}^\dagger) - h\mathbf{V}_s \mathbf{G}_{w,\text{cv}} \mathbf{p} + \mathbf{E}\mathbf{r}) + \mathbf{C}_s \mathbf{u}_s^i.\end{aligned}\quad (14.5)$$

Semi-Implicit Herschel-Bulkley Liquid. Explicitly integrated Herschel-Bulkley liquid is only stable with a time step that is two orders of magnitude smaller [261] than the implicitly integrated strands. To match the time step of the strands, the pressure and the shear stress must be implicitly integrated. Similar to prior work [226], we adopt a splitting scheme when integrating the shear stress and pressure for computational efficiency, which leads to a stable semi-implicit integrator. Figure 14.1 compares our method against explicit integration; our semi-implicit method can stably handle

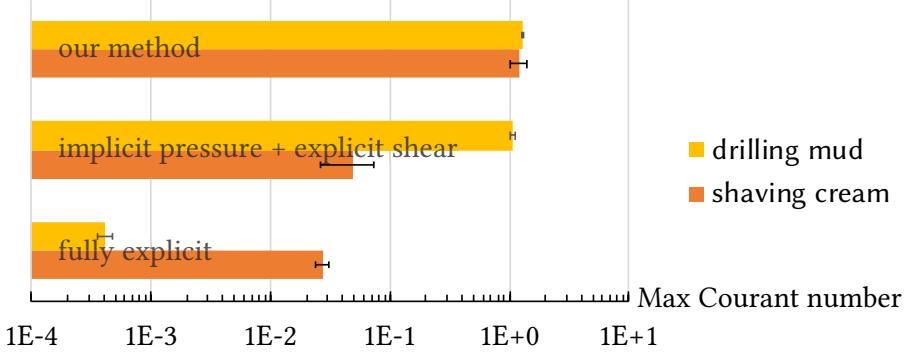


Figure 14.1: **Comparing implicit and explicit integration.** We compare the maximal Courant number allowed for stable simulation between different integrators for the scene in Figure 13.9. The error bars indicate standard deviation over different grid settings. We adopt logarithmic coordinates and put the origin at 10^{-4} for legibility.

much higher Courant numbers.

We first integrate the shear stress without the pressure applied by solving

$$\mathbf{C}_f \mathbf{u}_f^* = \mathbf{M}_f \mathbf{u}_f^t + h (\mathbf{f}_{MPM}^t + \mathbf{M}_f \mathbf{g} + \mathbf{D}_f \mathbf{u}_s^*) \quad (14.6)$$

where $\mathbf{C}_f = \mathbf{M}_f + \mathbf{D}_\tau + h \mathbf{D}_f + h^2 \mathbf{H}_f^t$, $\mathbf{D}_f = \text{diag}(\mathbf{W}_{gv}^T \text{vec}(\mathbf{D})) \in \mathbb{R}^{f \times f}$ is a diagonal matrix containing all drag coefficients interpolated on the grid, and \mathbf{H}_f^t is the Jacobian matrix of shear force evaluated at time step t . The operator $\text{vec}(\cdot)$ converts a diagonal matrix into a vector and $\text{diag}(\cdot)$ converts a vector into a diagonal matrix. Similarly $\mathbf{D}_\tau = \text{diag}(\mathbf{W}_{gv}^T \text{vec}(\mathbf{M}_\tau))$ is a diagonal matrix containing all the mass of the surface flow interpolated onto the grid. Here \mathbf{f}_{MPM} is the MLS-MPM discretization of the shear force [104] whose term on face i can be computed as

$$\mathbf{f}_{MPM,i}^t = - \sum_p \mu J_p^{-2/3} N_i(\mathbf{x}_p) V_p^0 \mathbf{D}_p^{-1} \mathbf{E}_i^T \text{dev}(\mathbf{b}_p^{E,t}) (\mathbf{x}_i - \mathbf{x}_p) \quad (14.7)$$

where μ is the shear modulus; $N_i(\mathbf{x}_p)$ is the B-spline kernel evaluated at the position \mathbf{x}_p of particle p ; $\mathbf{x}_{f,i}$ is the central position of grid face i ; \mathbf{E}_i is the normal direction of grid face i ; \mathbf{D}_p is the inertia tensor of the kernel function; and $\text{dev}(\mathbf{b}_p^{E,t})$ is the deviatoric part of the left Cauchy-Green strain tensor $\mathbf{b}_p^{E,t}$. We adopt a quadratic kernel where $\mathbf{D}_p = \frac{1}{4}\Delta x^2\mathbf{I}_3$ [123] and Δx is the grid spacing.

Using the new velocity with the shear stress applied, we solve the pressure equation (refer to **Pressure Solve** below for details). We then apply the pressure gradient onto the right hand side of (14.8) and perform another implicit solve, where

$$\mathbf{C}_f \mathbf{u}_f^{t+1} = \mathbf{M}_f \mathbf{u}_f^t + h (\mathbf{f}_{MPM}^t + \mathbf{M}_f \mathbf{g} + \mathbf{D}_f \mathbf{u}_s^* - \mathbf{V}_f \mathbf{G}_{cg} \mathbf{p}) \quad (14.8)$$

Remark. Directly interpolating the drag forces onto the grid can introduce poor conditioning due to the interpolation matrix [262]. To avoid this issue, we instead interpolate the drag *coefficients* onto the grid, and then use them to recompute a grid-based drag force and apply it to the liquid. This choice makes the drag matrix diagonal and avoids conditioning issues, without introducing apparent visual artifacts. Yue et al. [262] took a similar approach in the context of matching granular flow velocities solved with different models.

Semi-Analytic Plastic Flow. A Herschel-Bulkley liquid will yield and undergo plastic flow once its shear stress exceeds its yield stress, leading to a decrease of the shear stress. This fact must be considered when computing the Jacobian of the shear force, which necessitates the differentiability of the equation for plastic flow. However,

prior work on Herschel-Bulkley liquids [261] computed the plastic flow with bisection, making the process non-differentiable. Fortunately, we found that the plastic flow for Herschel-Bulkley liquids can in fact be computed analytically, as described below.

In the following discussion, we use a bar to denote *volume-preserving* variables, e.g., the *volume-preserving* left Cauchy-Green strain is $\bar{\mathbf{b}}^E \equiv J^{-2/d}\mathbf{b}^E$. Similar to prior work [261], we first compute an intermediate state for the updated normalized left Cauchy-Green strain as $\bar{\mathbf{b}}^{E,*} = \bar{\mathbf{f}}\bar{\mathbf{b}}^E\bar{\mathbf{f}}^T$ that accounts for the elastic deformation in (13.12), where $\mathbf{f} \equiv \mathbf{I}_d + h\nabla\mathbf{u}_f^t$ is the increment of the deformation gradient. The *updated* shear stress is then denoted as $\mathbf{s}^* \equiv \mu\text{dev}\bar{\mathbf{b}}^{E,*}$ and its norm as $s^* \equiv \|\mathbf{s}^*\|$.

Once plastic flow occurs, according to the von Mises yield condition (13.11), we can update the norm s^* as

$$s^{t+1} = \begin{cases} (s^* - \tilde{\sigma}_Y)e^{-\frac{2\hat{\mu}}{\eta}h} + \tilde{\sigma}_Y & n = 1 \\ \left[(s^* - \tilde{\sigma}_Y)^{\frac{n-1}{n}} - 2\hat{\mu}h \left(1 - \frac{1}{n}\right) \eta^{-\frac{1}{n}} \right]^{\frac{n}{n-1}} + \tilde{\sigma}_Y & n \neq 1 \end{cases} \quad (14.9)$$

where $\hat{\mu} \equiv \frac{\mu}{3}\text{tr}\bar{\mathbf{b}}^{E,*}$, $\tilde{\sigma}_Y \equiv \sqrt{\frac{2}{3}\sigma_Y}$, σ_Y is the yield stress, η is the flow consistency index, and n is the flow behavior index. We can then recover the volume-preserving Cauchy-Green strain after plastic flow as $\bar{\mathbf{b}}^{E,t+1} = \frac{s^{t+1}}{s^t}\text{dev}(\bar{\mathbf{b}}^{E,*}) + \hat{\mu}\mathbf{I}_3$.

Derivation of (14.9). We begin the derivation from the formulation of the temporal derivative of left Cauchy-Green strain, which is given in (13.12). Before solving the plastic flow, we have integrated the elastic deformation through $\bar{\mathbf{b}}^{E,*} = \bar{\mathbf{f}}\bar{\mathbf{b}}^E\bar{\mathbf{f}}^T$.

Therefore, we only need to consider the *plastic part* of $d\bar{\mathbf{b}}^E/dt$, which is denoted as $d\bar{\mathbf{b}}^{E,*}/dt$ and has the following formulation

$$\frac{d\bar{\mathbf{b}}^{E,*}}{dt} = -\frac{2}{3}\text{tr}(\bar{\mathbf{b}}^{E,*})\gamma(s^*)\hat{\mathbf{s}}^*. \quad (14.10)$$

where $s^* \equiv \|\mathbf{s}^*\|$ and $\mathbf{s}^* \equiv \mu\text{dev}\bar{\mathbf{b}}^{E,*}$ is the shear stress after the elastic deformation being integrated. We further define the normalized shear stress $\hat{\mathbf{s}}^* \equiv \mathbf{s}^*/s^*$.

With some algebraic manipulations, we can rewrite $\bar{\mathbf{b}}^{E,*}$ from the definition of $\hat{\mathbf{s}}^*$, as following

$$\bar{\mathbf{b}}^{E,*} = \frac{s^*}{\mu}\hat{\mathbf{s}}^* + \frac{1}{3}\text{tr}(\bar{\mathbf{b}}^{E,*})\mathbf{I}_3 \quad (14.11)$$

Taking the temporal derivative of both sides, we have

$$\frac{d\bar{\mathbf{b}}^{E,*}}{dt} = \mu^{-1} \left(\frac{ds^*}{dt}\hat{\mathbf{s}}^* + s^* \frac{d\hat{\mathbf{s}}^*}{dt} \right) + \frac{1}{3}\text{tr} \left(\frac{d\bar{\mathbf{b}}^{E,*}}{dt} \right) \mathbf{I}_3 \quad (14.12)$$

Since $\hat{\mathbf{s}}^*$ is deviatoric, $\text{tr}(\hat{\mathbf{s}}^*) = 0$, we then have

$$\text{tr} \left(\frac{d\bar{\mathbf{b}}^{E,*}}{dt} \right) = 0. \quad (14.13)$$

Besides, during the plastic flow, the shear stress \mathbf{s}^* would change in magnitude but *not in direction* due to the *principle of maximum plastic dissipation* [215], i.e.,

$$\frac{d\hat{\mathbf{s}}^*}{dt} = 0. \quad (14.14)$$

Therefore, we have

$$\frac{d\bar{\mathbf{b}}^{E,*}}{dt} = \mu^{-1} \frac{ds}{dt} \hat{\mathbf{s}}^* \quad (14.15)$$

In other words, the temporal derivative of $\mathbf{b}^{E,*}$ can be computed directly from the temporal derivative of the magnitude of shear stress s during the plastic flow.

Comparing (14.15) with (14.10), we discover that

$$\frac{ds^*}{dt} = -2\hat{\mu}\gamma(s^*), \quad (14.16)$$

where $\hat{\mu} \equiv \frac{\mu}{3} \text{tr} \bar{\mathbf{b}}^{E,*}$. Assuming $\Phi(s)$ in (13.13) will not change its sign during one time step, we can then integrate s^* from time step t to $t + 1$ through (14.16) analytically, which gives us the form of (14.9). ■

The key benefit of (14.9) is that plastic flow becomes differentiable over s^* and $\hat{\mu}$, making the shear stress differentiable. Hence we can compute the Jacobian matrix \mathbf{H}_f of the shear force and implicitly integrate the shear stress with plastic flow considered.

Jacobian of the Shear Force. In the augmented MLS-MPM method, the i -th row and j -th column of the Jacobian matrix \mathbf{H}_f of the shear force (defined in (14.7)) is computed as [104, 226]

$$\mathbf{H}_{f,ij} = \sum_p V_p^0 \mathbf{E}_i^T \mathbf{L}_{p,j} (\mathbf{F}_p^E)^T \mathbf{D}_p^{-1} N_i(\mathbf{x}_p) (\mathbf{x}_{f,i} - \mathbf{x}_p) \quad (14.17)$$

where

$$\mathbf{L}_{p,j} \equiv \frac{\partial^2 W_s}{\partial \mathbf{F} \partial \mathbf{F}} : \mathbf{D}_p^{-1} N_j(\mathbf{x}_p) \mathbf{E}_j (\mathbf{x}_{f,j} - \mathbf{x}_p)^T \mathbf{F}_p^E. \quad (14.18)$$

In the equations above, $N_i(\mathbf{x}_p)$ is the B-spline kernel evaluated at the position \mathbf{x}_p of particle p ; $\mathbf{x}_{f,i}$ is the central position of grid face i ; \mathbf{E}_i is the normal direction of grid face i ; \mathbf{D}_p is the inertia tensor of the kernel function; $\text{dev}(\mathbf{b}_p^{E,t})$ is the deviatoric part of the left Cauchy-Green strain tensor $\mathbf{b}_p^{E,t}$ (see the opening section of §13.1); and the operator $\mathbf{A} : \mathbf{B}$ denotes the tensor product between a fourth-order tensor \mathbf{A} and a second-order tensor \mathbf{B} . We then need to insert our Herschel-Bulkley model into these equations. Below we derive a general Jacobian matrix for 2D and 3D, with the number of dimensions denoted as d , i.e., $d = 2$ for 2D and $d = 3$ for 3D.

We begin our derivation from (13.6), where we have the derivative of shear energy over \mathbf{F}^E :

$$\frac{\partial W_s}{\partial \mathbf{F}_{ij}^E} = \mu \text{dev}(\bar{\mathbf{b}}^E) \mathbf{F}^{E-T}. \quad (14.19)$$

where a bar indicates normalized variables, and $\bar{\mathbf{b}}^E = J^{-2/d} \mathbf{F}^E \mathbf{F}^{E-T}$ is the normalized left Cauchy-Green strain tensor. We define a function λ to represent the plastic flow, i.e., rewriting (14.9) as $s^{t+1} = \lambda(s^*, \hat{\mu})$.

Below we drop the star and E superscripts for brevity. We have $\bar{\mathbf{b}} = \frac{\lambda(s, \hat{\mu})}{s} \text{dev}(\bar{\mathbf{b}}) + \hat{\mu} \mathbf{I}_3$ and thence $\text{dev}(\bar{\mathbf{b}}) = \frac{\lambda(s, \hat{\mu})}{s} \text{dev}(\bar{\mathbf{b}})$, and

$$\frac{\partial W_s}{\partial \mathbf{F}_{ij}} = \mu \tilde{\lambda} J^{-2/d} \left[\mathbf{F}_{ij} - \frac{1}{d} \text{tr}(\mathbf{F} \mathbf{F}^T) (\mathbf{F}^{-T})_{ij} \right]. \quad (14.20)$$

where $\tilde{\lambda} \equiv \frac{\lambda(s, \hat{\mu})}{s}$. Applying matrix calculus [182], we have the following derivatives:

$$\frac{\partial J^{-2/d}}{\partial \mathbf{F}_{uv}} = -\frac{2}{d} J^{-2/d} (\mathbf{F}^{-T})_{uv}, \quad (14.21a)$$

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{F}_{uv}} = \delta_{ui} \delta_{jv}, \quad (14.21b)$$

$$\frac{\partial}{\partial \mathbf{F}_{uv}} \text{tr}(\mathbf{FF}^T) = \mathbf{F}_{uv}, \quad (14.21c)$$

$$\frac{\partial}{\partial \mathbf{F}_{uv}} (\mathbf{F}^{-T})_{ij} = \frac{\partial}{\partial \mathbf{F}_{uv}} (\mathbf{F}^{-1})_{ji} = -(\mathbf{F}^{-1})_{ju} (\mathbf{F}^{-1})_{vi}. \quad (14.21d)$$

where δ_{ij} is the Kronecker delta, i.e., $\delta_{ij} = 1$ if and only if $i = j$.

By the chain rule, we can compute

$$\frac{\partial s}{\partial \mathbf{s}} = \frac{\mathbf{s}}{s}, \quad (14.22a)$$

$$\frac{\partial \mathbf{s}_{ij}}{\partial \mathbf{F}_{uv}} = \mu J^{-2/d} \left[-\frac{2}{d} (\mathbf{F}^{-T})_{uv} [\text{dev} \mathbf{b}]_{ij} + \delta_{iu} \mathbf{F}_{jv} - \frac{1}{d} \mathbf{F}_{uv} \delta_{ij} \right], \quad (14.22b)$$

$$\frac{\partial s}{\partial \mathbf{F}_{uv}} = -\frac{2s}{d} (\mathbf{F}^{-T})_{uv} + \mu \frac{(\text{dev} \bar{\mathbf{b}} \cdot \mathbf{F})_{uv} - \frac{1}{d} \text{tr}(\text{dev} \bar{\mathbf{b}})}{\|\text{dev} \bar{\mathbf{b}}\|}, \quad (14.22c)$$

$$\frac{\partial \hat{\mu}}{\partial \mathbf{F}_{uv}} = \frac{1}{d} \mu J^{-2/d} \left[-\frac{2}{d} (\mathbf{F}^{-T})_{uv} \text{tr} \mathbf{b} + \mathbf{F}_{uv} \right]. \quad (14.22d)$$

The Hessian of the shear energy then becomes

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{F}_{uv}} \left(\frac{\partial W_s}{\partial \mathbf{F}_{ij}} \right) \\ &= -\frac{2}{d} J^{-2/d} (\mathbf{F}^{-T})_{uv} \left[\mathbf{F}_{ij} - \frac{1}{d} \text{tr}(\mathbf{FF}^T) (\mathbf{F}^{-T})_{ij} \right] \\ &+ \mu J^{-2/d} \left[\delta_{ui} \delta_{jv} - \frac{1}{d} \left(\mathbf{F}_{uv} \mathbf{F}_{ij}^{-T} - \text{tr}(\mathbf{FF}^T) (\mathbf{F}^{-1})_{ju} (\mathbf{F}^{-1})_{vi} \right) \right] \\ &+ \left[\mu J^{-2/d} \left[\mathbf{F}_{ij} - \frac{1}{d} \text{tr}(\mathbf{FF}^T) (\mathbf{F}^{-T})_{ij} \right] \right] \end{aligned}$$

$$\left[\frac{\partial \tilde{\lambda}}{\partial s} \left(-\frac{2s}{d} (\mathbf{F}^{-T})_{uv} + \mu \frac{(\text{dev}\bar{\mathbf{b}} \cdot \mathbf{F})_{uv}}{\|\text{dev}\bar{\mathbf{b}}\|} \right) + \frac{\partial \tilde{\lambda}}{\partial \mu} \frac{\mu}{d} J^{-2/d} \left(-\frac{2}{d} (\mathbf{F}^{-T})_{uv} \text{tr}\mathbf{b} + \mathbf{F}_{uv} \right) \right]. \quad (14.23)$$

In addition, for an arbitrary matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$, under the Einstein notation we have [182]

$$\delta_{ui} \delta_{jv} \mathbf{B}_{uv} = \mathbf{B}_{ij}, \quad (14.24a)$$

$$\mathbf{F}_{uv} \mathbf{B}_{uv} = \text{tr}(\mathbf{F}^T \mathbf{B}), \quad (14.24b)$$

$$(\mathbf{F}^{-1})_{ju} (\mathbf{F}^{-1})_{vi} \mathbf{B}_{uv} = (\mathbf{F}^{-T} \mathbf{B}^T \mathbf{F}^{-T})_{ij}, \quad (14.24c)$$

$$\left(\frac{\partial^2 W_s}{\partial \mathbf{F} \partial \mathbf{F}} : \mathbf{B} \right)_{ij} \equiv \frac{\partial}{\partial \mathbf{F}_{uv}} \left(\frac{\partial W_s}{\partial \mathbf{F}_{ij}} \right) \mathbf{B}_{uv}. \quad (14.24d)$$

Using these equations and some algebra operations, we have the following formulation for multiplying the Hessian of shear energy with an arbitrary matrix \mathbf{B} :

$$\begin{aligned} \frac{\partial^2 W_s}{\partial \mathbf{F} \partial \mathbf{F}} : \mathbf{B} &= \mu \left[J^{-2/d} \mathbf{B} - \frac{2}{d} \text{tr}(\mathbf{F}^{-1} \mathbf{B}) \text{dev}(\bar{\mathbf{b}}) \mathbf{F}^{-T} \right. \\ &\quad \left. - \frac{1}{d} \mathbf{F}^{-T} (J^{-2/d} \text{tr}(\mathbf{F}^T \mathbf{B}) \mathbf{I}_d - \text{tr}(\bar{\mathbf{b}}) \mathbf{B}^T \mathbf{F}^{-T}) \right] \\ &\quad + \mu \text{dev}\bar{\mathbf{b}} \cdot \mathbf{F}^{-T} \left[\frac{\partial \tilde{\lambda}}{\partial s} \left(-\frac{2s}{d} \text{tr}(\mathbf{F}^{-1} \mathbf{B}) + \mu \frac{\text{tr}(\mathbf{F}^T \text{dev}\bar{\mathbf{b}} \cdot \mathbf{B})}{\|\text{dev}\bar{\mathbf{b}}\|} \right) \right. \\ &\quad \left. + \frac{\partial \tilde{\lambda}}{\partial \mu} \frac{\mu}{d} J^{-2/d} \left(-\frac{2}{d} \text{tr}(\mathbf{F}^{-1} \mathbf{B}) \text{tr}\mathbf{b} + \text{tr}(\mathbf{F}^T \mathbf{B}) \right) \right] \end{aligned} \quad (14.25)$$

Explicitly computing (14.25) and constructing a Jacobian matrix is not economically efficient. Instead, only computing the result of multiplying the Jacobian with a

vector is more effective, similar to prior works [104, 226] (e.g., in (14.6) that implicitly integrates the shear stress, the vector to be multiplied with is the velocity \mathbf{u}_f^* or some intermediate states in a conjugate gradient solver). Noticing that the deformation gradient \mathbf{F} can be canceled with or combined into $\bar{\mathbf{b}}$ when substituting (14.25) into (14.18) and (14.17), then the multiplication between the Jacobian matrix \mathbf{H}_f and an arbitrary vector \mathbf{q} (whose dimension matches the number of columns of \mathbf{H}_f) can be computed below.

Defining

$$\mathbf{v}_{j\alpha,p} \equiv N_{j\alpha}(\mathbf{x}_p)(\mathbf{x}_{f,j\alpha} - \mathbf{x}_p) \quad (14.26)$$

for direction $\alpha \in [0, d - 1]$, in 2D we define

$$\mathbf{B}_p \equiv \sum_j [\mathbf{v}_{j0,p} \mathbf{q}_{j0}, \mathbf{v}_{j1,p} \mathbf{q}_{j1}]. \quad (14.27)$$

and in 3D

$$\mathbf{B}_p \equiv \sum_j [\mathbf{v}_{j0,p} \mathbf{q}_{j0}, \mathbf{v}_{j1,p} \mathbf{q}_{j1}, \mathbf{v}_{j2,p} \mathbf{q}_{j2}]. \quad (14.28)$$

With all the variables defined above substituted into (14.17), and using \mathbf{B}_p to replace the arbitrary matrix \mathbf{B} in (14.25), the Jacobian matrix multiplied with an arbitrary vector \mathbf{q} is computed as

$$(\mathbf{H}_f \mathbf{q})_{i\alpha} = \sum_p \tilde{\mathbf{L}}_{p,\alpha*} \mathbf{v}_{i\alpha,p} \quad (14.29)$$

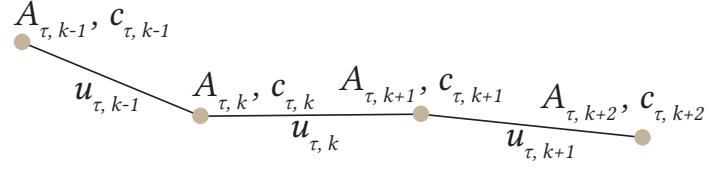


Figure 14.2: **A strand with staggered discrete flow variables:** the cross-sectional area A and the reduced elastic Cauchy-Green strain c_τ are defined on vertices, and the flow velocity u_τ is defined on edges.

where $\tilde{\mathbf{L}}_{p,\alpha*} \in \mathbb{R}^{1 \times d}$ is the α -row of $\tilde{\mathbf{L}}_p$, and

$$\begin{aligned} \tilde{\mathbf{L}}_p \equiv & \mu V_p^0 \mathbf{D}_p^{-2} \left[s^{t+1} \left(\mathbf{B}_p^T \bar{\mathbf{b}} - \frac{2}{d} \text{tr}(\mathbf{B}_p) \text{dev}(\bar{\mathbf{b}}) - \frac{1}{d} ((\mathbf{B}_p \odot \bar{\mathbf{b}}) \mathbf{I}_d - \text{tr}(\bar{\mathbf{b}}) \mathbf{B}_p) \right) \right. \\ & \left. + \text{dev}(\bar{\mathbf{b}}) \left(\frac{\partial \tilde{\lambda}}{\partial s} \left(\frac{\mu \text{tr}(\text{dev}(\bar{\mathbf{b}}) \mathbf{B}_p^T \bar{\mathbf{b}})}{\|\text{dev} \bar{\mathbf{b}}\|} - \frac{2s^{t+1}}{d} \right) + \frac{\partial \tilde{\lambda}}{\partial \mu} \frac{\mu}{d} \left(\text{tr}(\mathbf{B}_p^T \bar{\mathbf{b}}) - \frac{2}{d} \text{tr}(\mathbf{B}_p) \text{tr}(\bar{\mathbf{b}}) \right) \right) \right], \end{aligned} \quad (14.30)$$

where \odot denotes the Frobenius inner product, i.e., $\mathbf{A} \odot \mathbf{B} = \sum_i \sum_j A_{ij} B_{ij}$. ■

Pressure Solve. After integrating the shear stress, we compute the pressure \mathbf{p}

using

$$\mathbf{C}_p \mathbf{p} = \mathbf{K}_c^{-1} \mathbf{p}^t + h \mathbf{V}_{\text{c}}^{-1} \mathbf{G}_{\text{w,cv}}^T \mathbf{V}_{\text{s}} (\mathbf{u}_{\text{s}}^* - \mathbf{W}_{\text{gv}} \mathbf{u}_{\text{f}}^*) + h \mathbf{G}_{\text{cg}}^T \mathbf{u}_{\text{f}}^*, \quad (14.31)$$

where

$$\mathbf{C}_p = \mathbf{K}_c^{-1} + h^2 \mathbf{G}_{\text{cg}}^T \hat{\mathbf{C}}_{\text{f}}^{-1} \mathbf{V}_{\text{f}} \mathbf{G}_{\text{cg}}, \quad (14.32)$$

the diagonal matrix \mathbf{K}_c has i -th entry $\mathbf{K}_{c,i} \equiv \frac{\kappa}{2} \left(J_i^E + \frac{1}{J_i^E} \right)$, and $\hat{\mathbf{C}}_{\text{f}}^i$ is a diagonal matrix to approximate \mathbf{C}_{f} , i.e. $\hat{\mathbf{C}}_{f,kl}^i = \mathbf{C}_{f,kl}^i \delta_{kl}$. To preserve the symmetry of the pressure equation (14.31), we adopt explicit integration for the volume fraction change in (13.72b). We did not observe any instabilities resulting from this choice.

Surface Flow. Similar to the surface flow in **Part II**, we discretize the spatial derivatives in (13.17) with finite differences (Figure 14.2). At each time step we first advect both velocity u_τ and cross-sectional area A_τ with backtracing [224]. We then integrate the remaining terms in (13.17) and update the cross-sectional area A_τ by integrating the divergence of velocity in (13.48). For the area at vertex k we have

$$A_{\tau,k}^{t+1} = A_{\tau,k}^* \exp \left(-h \frac{u_{\tau,k} - u_{\tau,k-1}}{l_k} \right) \quad (14.33)$$

where $A_{\tau,k}^*$ is the area after advection, and l_k is the Voronoi length of vertex k . Finally we update the Cauchy strain c_τ^E by discretizing the spatial derivatives in (13.18) with finite differences, which produces a nonlinear equation that we solve via bisection.

Contact Handling. Similar to prior work [129], we loop over all contacts during the collision solve, updating each contact while the remaining contacts are fixed, in a Gauss-Seidel-like manner. By defining \mathbf{E} as the transformation from world space to contact space, the equation used to update the k -th contact is

$$\mathbf{u}_s^{t+1} = \mathbf{u}_s^\dagger + \mathbf{C}_s^{-1} (\mathbf{E}_k \mathbf{r}_{s,k} + \bar{\mathbf{E}}_k \bar{\mathbf{r}}_{s,k}) . \quad (14.34)$$

where $\bar{\mathbf{E}}_k$ and $\bar{\mathbf{r}}_{s,k}$ are respectively the complement matrix and vector formed by zeroing out the columns and entries used for \mathbf{E}_k and $\mathbf{r}_{s,k}$.

In the following we define the future relative velocity at the contact as $\mathbf{v} \equiv \mathbf{E}^T \mathbf{u}_s$.

Multiplying by \mathbf{E}^T on both sides of (14.34) gives

$$\mathbf{v}^{t+1} = \mathbf{E}^T (\mathbf{u}_s^\dagger + \mathbf{C}_s^{-1} \mathbf{E} (\mathbf{r} + \mathbf{r}_A)) . \quad (14.35)$$

Above, following the YacFS library [75] attached with ADONIS [129, 130], we have decomposed the total contact impulse \mathbf{r}_s into the unknown repulsive collision impulse \mathbf{r} and the known cohesive impulse $\mathbf{r}_A \in \mathbb{R}^{3k \times 1}$. The latter is a vector composed of cohesive forces in the normal directions of all contacts.

We can then reformulate the equations above as a second-order Coulomb cone problem (SOCCP) given by

$$\mathbf{S}\mathbf{r} = \mathbf{v} - \mathbf{E}^T \mathbf{u}_s^\dagger - \mathbf{S}\mathbf{r}_A, \quad (14.36a)$$

$$\forall k, (\mathbf{r}_{s,k}, \mathbf{v}_k) \in K_\mu \quad (14.36b)$$

where $\mathbf{S} = \mathbf{E}^T \mathbf{C}_s^{-1} \mathbf{E}$ is the *Delassus operator* [30]. We solve this with So-Bogus [69].

14.2 Algorithm

We summarize our resulting nonlinear mixture solver in Algorithm 2. For advection and mapping velocities between particles and the grid, we adopt the affine particle-in-cell (APIC) [122] method and the moving least-squares material point method (MLS-MPM) [104] for their simplicity and efficacy. As noted earlier, we rely on an underlying staggered (MAC) grid, similar to augmented MPM [226].

Algorithm 2 Algorithm solving the dynamics of the liquid-strand mixture.

```
 $\mathbf{u}_\tau^* \leftarrow \text{ADVECT}()$  for the surface flow  
 $\mathbf{u}_s^* \leftarrow \text{NONLINEARNEWTONSOLVE}()$  through (14.3) for the prediction of a new strand velocity  
 $\mathbb{K} \leftarrow \text{COLLISIONDETECTION}(\mathbf{u}_s^*, \mathbf{q}_s^t)$   
 $\mathbf{u}_f^* \leftarrow \text{SHEARSTRESSSOLVE}()$  for liquid (14.6)  
 $\mathbf{p}^{t+1} \leftarrow \text{PRESSURESOLVE}()$  for liquid (14.31)  
 $\mathbf{u}_f^{t+1} \leftarrow \text{SHEARSTRESSSOLVE}()$  with the pressure gradient applied (14.8) to update liquid velocity  
 $\mathbf{u}_s^{t+1} \leftarrow \text{NONLINEARNEWTONSOLVE}()$  through (14.4), for a new strand velocity with the pressure gradient applied  
 $\mathbf{u}_\tau^* \leftarrow \text{VELOCITYINTEGRATE}()$  of the surface flow with forces added  
 $\mathbf{A}_\tau^{t+1} \leftarrow \text{CROSSSECTIONALUPDATE}()$  for the surface flow with equation (14.33)  
 $j \leftarrow 0$   
while contact error >  $\epsilon_{\text{contact}}$  &  $j < j_{\max}$  do  
    for  $k \in \mathbb{K}$  do  
         $\mathbf{r}_k \leftarrow \text{CONTACTSOLVE}()$  with (14.36a) s.t.  $(\mathbf{r}_{s,k}, \mathbf{v}^{t+1}) \in K_\mu$   
         $\mathbf{r} \leftarrow \bar{\mathbf{r}}_k + \mathbf{r}_k$   
    end for  
     $j \leftarrow j + 1$   
end while  
 $\mathbf{u}_s^{t+1} \leftarrow \text{NONLINEARNEWTONSOLVE}()$  through (14.5) for the final strand velocity  
 $\mathbf{q}_s^{t+1} \leftarrow \mathbf{q}_s^{t+1} + h\mathbf{u}_s^{t+1}$   
return  $(\mathbf{u}_s^{t+1}, \mathbf{q}_s^{t+1}, \mathbf{u}_f^{t+1}, \mathbf{u}_\tau^{t+1}, \mathbf{A}_\tau^{t+1})$ 
```

A single step of our complete algorithm consists of the following sequence of operations:

1. **Transfer between surface and bulk liquid.** Bulk liquid is captured as surface flow for those strands crossing the interface; excess liquid from surface flow is converted into particles.
2. **Merge, split, and relax particles.** Following Winchenbach et al. [253], particles that are too small are merged with neighbor particles, while ones that are too large are split. In addition, we apply a pass of relaxation [9] to maintain the uniformity of the particle distribution.

3. **Map liquid particles to grid.** At the start of every time step we transfer the particles' mass, velocity, and volume change to the MAC grid, through the APIC method [122].
4. **Compute weighting matrices.** The matrices for mapping are computed with kernel weights, for both liquid and strands.
5. **Detect tearing regions.** Each particle's accumulated plasticity is examined to detect tearing [261].
6. **Solve for Mixture.** The velocities of the grid, surface flow, and strands are updated, following Algorithm 2. The shear equation (14.6) is solved with a Jacobi preconditioned conjugate gradient solver [204], and the pressure equation (14.31) is solved with an algebraic multigrid preconditioned conjugate gradient (AMGPCG) solver [263].
7. **Update liquid particles from grid.** We update each particle's velocity from the MAC grid via APIC [122].
8. **Update particle deformation info.** The deformation gradient, left Cauchy-Green strain, and volume change are updated through MLS-MPM [104].
9. **Update positions for particles and strands.** Positions are updated according to the velocities for liquid particles and strand vertices.
10. **Compute plasticity for bulk liquid.** The plastic flow of bulk liquid is computed from the deformation gradient, where excess elastic strain is converted to plastic strain.
11. **Compute plasticity for surface flow.** The strain in the surface flow is updated by (13.18), where excess elastic strain is converted to plastic strain.

12. Compute plastic recovery. The plasticity history is relaxed to account for the strengthening of bonds between bulk materials [261].

Chapter 15

Simulated Results of the Strands Coupled with Shear-Dependent Liquids

We divide our results into two classes: **i)** a group of didactic cases designed to validate individual components of our framework and **ii)** a set of more general scenarios of Herschel-Bulkley fluid interaction with strands that demonstrate the diversity of practical effects that can be achieved by our system.

15.1 Didactic Examples

Varying Volume Fraction. To show the importance of the volume fraction term in the pressure equation, we compare simulations of liquid flowing through hair with and without the volume fraction term used when solving the pressure equation (Figure 13.6). For ease of comparison drag is disabled in this scene.

Buoyancy. Our method introduced in this part can correctly handle materials with different mass densities, which is not considered in **Part II**. In Figure 13.5 we demonstrate the buoyancy behavior of hairs with different mass densities in water.

Examples	sec./step strand	min./frame liquid	hour/anim. strand	sec. liquid	# particles (max)	# vertices of strands	grid size max dimensions	Δx (cm)		
Shaking a Hairball	3.0	122.4	1.7	68.0	0.8	34.0	12.3 M	46.1K	$120 \times 128 \times 120$	0.5
Splashing Paint	14.9	36.1	8.3	20.1	4.1	10.0	2.1 M	119.4K	$328 \times 424 \times 328$	0.5
Chocolate "Dog"	20.2	57.3	11.2	31.8	5.6	15.9	3.0 M	688.1K	$192 \times 272 \times 680$	0.75
Soba with Oyster Sauce	32.4	9.3	45	12.9	22.5	6.5	492.7 K	97.0K	$128 \times 80 \times 136$	0.375

Table 15.1: **Timings and storage statistics.** The timings are averaged over all the steps or frames of one example. Each frame is 1/30s.

Drag Force. In Figure 13.9 we compare our material-specific drag coefficient against a constant drag coefficient for distinct liquids falling onto strands. In Figure 14.1 we use the same scenario to compare the maximal (unitless) Courant number (calculated with $u_{\max}h/\Delta x$ where u_{\max} is the maximal velocity across the whole domain, h is the time step, and Δx is the cell size) between different integration schemes. Compared with using an explicit integrator for the shear stress, or for both shear and pressure, our semi-implicit integrator is stable for both compressible shaving cream and incompressible drilling mud, and can handle more or less viscous liquids with moderate time steps (Courant number up to approximately 1.24).

Droplet Dripping. In Figure 13.1 large droplets of various liquids flow down thin strands, demonstrating the variety of material-dependent behavior that we capture. In Figure 13.4, we construct a similar scenario with a large, heavy tetrachloroethene droplet (mass density 1.622g/cm). When such a droplet flows on a thin strand, the flow can dramatically affect the strand's momentum. With the explicit inertia transfer method proposed in the last part, a huge correction will cause instability. On the other hand, our improved method can stably integrate the extra strand momentum induced by the large droplet.

High-Speed Rotation. To further demonstrate the behavior of surface flows for liquids with high viscosity or yield stress, we rapidly rotate strands with droplets of various materials on them (Figure 13.3). We observe that the mud and milk cream are flung out quickly after the simulation starts, while the milk cream starts to move only after the centrifugal force exceeds its yield stress. We also observed that the paint and milk cream flow more easily as their velocities are increased, demonstrating a shear-thinning behavior.

Cohesion and Coalescence. In Figure 13.10, two strands initially hang vertically with a distance of 0.011cm between their centerlines. We then gradually separate them. The milk cream between the two strands forms a strong cohesive force. Comparing our constraint-based model with the penalty-based model, the latter exhibits tunneling and locking artifacts when simulating the cohesion of strongly non-Newtonian materials.

Friction with Various Materials. To demonstrate how the liquid material affects friction between strands, we simulate two bundles of strands covered with cream and mud, respectively (Figure 13.13). Since the cream has higher viscosity and yield stress, the strands covered in cream move less readily compared with those covered in mud.

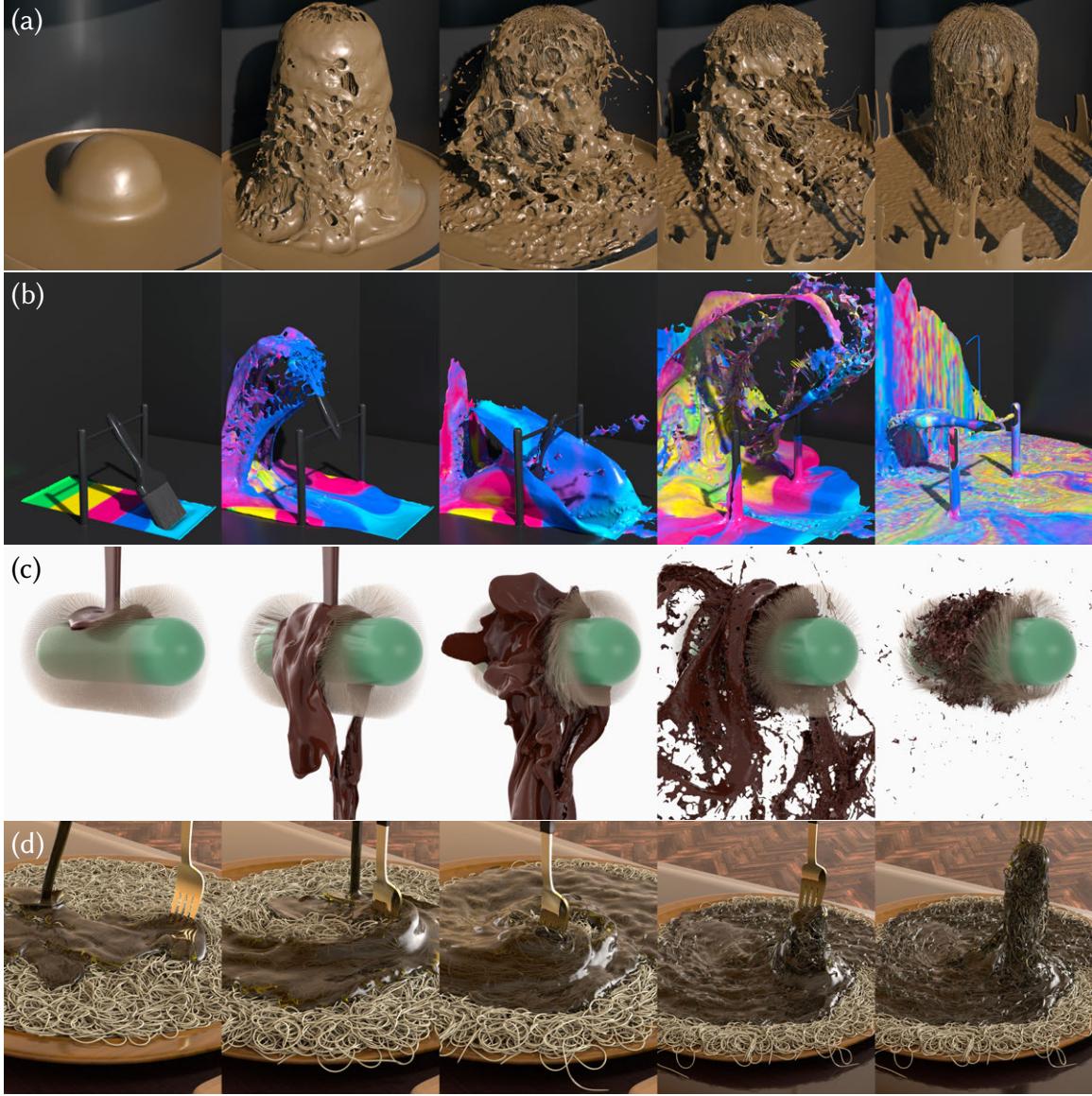


Figure 15.1: (a) *Shaking a Hairball*. A ball rises out of mud and shakes its hairs to throw off the mud. (b) *Splashing the Paint*. A rapidly rotating paint brush splashes oil paint everywhere. (c) *Chocolate “dog”*. Melted chocolate is poured onto a fluffy cylinder, which tries to shake the chocolate off by mimicking the mammal-shaking behavior. (d) *Soba with Oyster Sauce*. Oyster sauce is poured onto a plate of soba noodles, while a fork is used to stir and pull the noodles.

15.2 Large-Scale Examples

Shaking a Hairball. To illustrate the cohesive and frictional behavior of wet hairs, we simulate a hair-covered ball lifting out of a mud pool and shaking (Figure 15.1a).

When the motion stops, the hairs stick and tangle as expected.

Splashing Paint. Oil paint is another familiar shear-dependent fluid. We simulate the interaction between the brush bristles and a pool of colorful paint (Figure 15.1b). In contrast to examples from earlier work (e.g., [53, 266]) in which a brush moves slowly across paper and is one-way coupled with the paint, we show a brush that is two-way coupled with the paint during violent and rapid rotation, which causes large and dynamic splashing of the paint.

Chocolate “Dog”. In **Part II**, we simulated the process of pouring water over a rotating shaggy-haired cylinder, mimicking mammal shaking behavior [74]. To contrast water against more complex liquids, we revisit this scenario replacing the water with molten chocolate. Before the cylinder begins rotating, the falling chocolate forms thin sheets and tendrils; later, it separates into many chunky pieces. At the end of the rotation, much of the chocolate has adhered to the clumped hairs, as would be expected (Figure 15.1c).

Soba with Oyster Sauce. Moving beyond hair strands, our method can simulate the coupling between a plate of Soba (buckwheat) noodle and thick oyster sauce. As the fork is pulled up, due to the frictional effect induced by the strong viscosity of the sauce, the noodles naturally stick to the liquid and each other (Figure 15.1d).

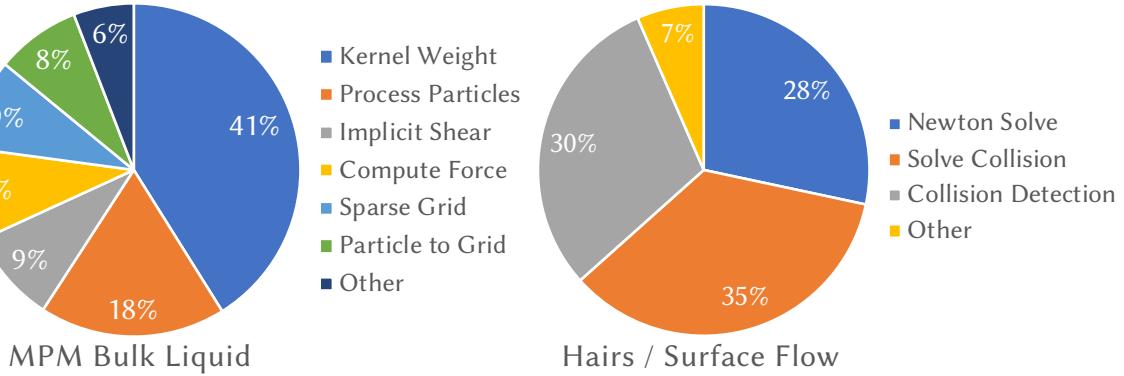


Figure 15.2: Timing statistics for the *Chocolate “Dog”*.

15.3 Performance Numbers

The timing data for our large scale examples is presented in Table 15.1, measured on a workstation with 2 sockets of Intel Xeon E5-2620 v3 CPUs running at 2.4GHz, each of which has 6 cores. We adopt a time step size 1×10^{-3} s for all examples, where the total lengths of examples vary between $0.75 \sim 8$ s. We also provide a detailed breakdown in Figure 15.2 for the *Chocolate “dog”* example.

15.4 Surface Reconstruction

When generating the liquid particles, we sample 64 particles in each cell occupied by the liquid, where we pre-compute a level-set to cull the particles sampled outside the generator. This amount of particles provides a smoother liquid surface during the reconstruction. We use the VDB [171] surface operators (SOPs) in Houdini [214] to perform the reconstruction. For each frame, we perform a *VDB from particle liquid* SOP to convert the particles into a level-set. To avoid incorrect holes or instability, we turn off the *rebuild* option and use a *Primitive* SOP to categorize the result VDB

as a level-set, which is followed by a *VDB renormalize* SOP to make sure the gradient of the level set is normalized. In the *VDB from particle liquid* SOP, we set the *particle separation* to be $0.5\Delta x$ where Δx is the cell size for simulation, and the *voxel size* (for reconstruction) is set to be 0.25, which means the resolution of the reconstruction grid is $8\times$ higher than the simulation grid. We then perform a dilation-smooth-erosion operation [170] to smooth the level-set and use a *Convert VDB* node to generate a polygonal surface mesh, where the smoothing method is set to *Mean Curvature Flow* so that the volume can be preserved during smoothing.

On the other hand, for the surface flow on strands, we first use a *PolyCut* SOP to remove the strand vertices that have zero flow height on a vertex itself and its neighborhood. Then we use a *Polywire* SOP to convert the height field on the remaining polylines into polygonal meshes.

We merge these two sets of polygons, and use a *VDB from polygons* SOP to convert the merged polygons back to a VDB with much higher resolution, with the *voxel size* set to 0.03 (cm). We then again perform a series of dilation-smooth-erosion operation [170] to smooth out the kinks around the connections between the bulk liquid and the surface flow, which in addition, also creates the liquid bridge between the flow on strands. Finally, we convert the level-set back into polygons for rendering with a *convert VDB* SOP.

15.5 Parameters

The physical parameters used in this part are taken from multiple pieces of the literature [12, 37, 131, 172, 261]. These parameters are given in the following table, where the water and tetrachloroethylene are *incompressible, Newtonian* liquids, the drilling mud, acrylic paint and oyster sauce are *incompressible, shear-thinning* liquids, the milk cream and shaving cream are *compressible, shear-thinning* liquids, and the milk chocolate is a *compressible, (almost) Bingham* liquid.

Materials	ρ (g/cm ³)	κ (dyne/cm ²)	μ (dyne/cm ²)	τ_Y (dyne/cm ²)	η (Ba · s ⁿ)	n (unitless)
water	1.0	2.0e10	0	0	8.9e-3	1.0
tetrachloroethylene	1.622	3.1e10	0	0	8.9e-3	1.0
drilling mud	1.22	2.0e10	1.0e3	16.813	6.496	0.5173
acrylic paint	0.95	1.35e9	4.0e3	9.6	173.56	0.3162
milk cream	0.275	1.09e6	1.6e4	1.2e3	50.0	0.27
shaving cream	0.2	1.09e6	2.9e3	3.19e2	2.72e2	0.22
oyster sauce	1.207	2e10	4.0e3	26.5	16.1	0.62
milk chocolate	0.95	4.28e6	4.0e3	3.0e2	28.0	0.98

Table 15.2: **Parameters of the materials used in this part.**

Chapter 16

Discussions and Limitations

We have presented a multi-scale framework that can couple strands with diverse shear-dependent liquids. The results in this part cover a wide range of materials, including incompressible or compressible, and Newtonian or non-Newtonian fluids.

Several of our developments benefit the coupling of strands to fluids in general, be they Newtonian or non-Newtonian. The pressure equation (13.72a) accommodates various constitutive models [123]. The momentum equation (13.50) of strands is agnostic to the fluid model. Indeed, (13.50) applies to any solid (not just strands) with surface flow, e.g., raindrops sliding over a glass pane. Di Felice's drag formula is equally applicable to both Newtonian and non-Newtonian fluid. The same is true of the boundary condition for the consistency between surface flow and bulk liquid.

Some of our other contributions are naturally more specific to shear-dependent fluids. The reduced Cauchy-Green strain and its dynamics are exclusive to shear-dependent surface flow. Aspects of the cohesive force and the semi-analytical formulation of the plastic flow (14.9) are only meaningful for Herschel-Bulkley fluids. Although Di Felice's drag formula is a comprehensive treatment, Renaud's drag coefficient is not, and is only required for coupling solids with non-Newtonian fluids [197,

229].

16.1 Limitations

Our framework still has a few limitations imposed by our assumptions, numerical methods, surface reconstructors, or selected experiments.

In the reduced surface flow model, surface tension in the longitudinal direction is neglected for ease of computation, and thus we cannot maintain the correct contact angle between the reduced surface flow and the strand. Also, large droplets that break the cylindrical assumption cannot be faithfully captured. We have not considered the fact that some strands may have anisotropic cross-sections. For some materials (e.g., milk cream), the surface liquid is slow in the absence of drastic strand motion, and hence the computation could be further simplified for better performance. Our liquid capturing process is purely geometric, and thus the liquid can exhibit some popping artifacts when the simulation replays slowly.

The pressure equation can be imprecise when the strands are compressible (e.g., rubber bands). We have not considered coupling strands with materials whose plastic flows are pressure-dependent, such as snow, sand, or rubber. The pressure solved in our staggered integration approach may not be consistent with the future velocity of the strands, although we have not observed visible artifacts caused by this fact. Similar to prior work [226], our bulk fluid model splits the integration of pressure from shear stress, which may affect the accuracy of the free surface [142].

Due to the limited resolution of the fluid grid, the drag force can affect the liquid motion over a wider region than expected if the grid resolution is insufficient. Increasing the resolution of the grid or using an adaptive grid [84] may alleviate this problem.

We adopted liquid parameters and drag coefficients from sources in the physics literature (§15.5). Nevertheless, with these measured parameters it remains difficult to accurately reproduce some real-life scenarios. For example, the behavior of pasta sauce will change drastically depending on variations in temperature, which is not considered in the cited references. Similarly, real pasta sauce may contain other ingredients (e.g., soy sauce, olive oil) that dramatically decrease its viscosity. Accurately capturing the physical parameters of liquids along with temperature change and ingredients remains for future work.

Some limitations are inherited from the sub-components adapted to build our framework. Augmented MPM spends more time computing (staggered) kernel weights than regular MPM. The hybrid iterative solver used for the SOCCP is not strictly guaranteed to converge, which may cause some penetrations (e.g., in *Soba with Oyster Sauce*, there are some penetrations between the noodle and the plate, which, however, are not observable explicitly). This is caused by the non-convergence of the SOCCP solver, which is a known problem for such a Gauss-Seidel solver. Neither is the stability of the elastic rods guaranteed: when two consecutive rod elements are bent to (almost) 180-degrees, the discrete curvature becomes (near-)singular, and the bending energy would rise to infinity. This issue can occur in practice when a large

drag force reshapes the strands into a problematic configuration.

Beyond the dynamics, it is difficult to perform an accurate surface reconstruction for particle-based non-Newtonian bulk liquid undergoing a drastic motion. Traditional geometric reconstruction methods largely ignore the particles' deformation history, such that thin sheets or tendrils may appear as separated droplets during reconstruction. Moreover, developing accurate shaders for rendering such non-Newtonian liquids can be difficult. For example, we tried using ketchup in *Soba with Oyster Sauce* but did not successfully obtain a ketchup shader with a realistic visual appearance. We believe many of the limitations described above can provide exciting fodder for future research.

Chapter 17

Conclusion

This dissertation focuses on the coupling between strands/clothes and different types of liquid. The purpose of this dissertation is to excavate the physics behind the amazing phenomena involved when strands and clothes interacting with liquid and, additionally, lay out a framework that can simulate these phenomena, including liquid dripping, capturing and dragging, as well as diffusing through fabric or flowing along the strands.

The phenomena involved in liquid-strand or liquid-fabric interaction can be very complicated. The main argument of this dissertation is that to plausibly recover these phenomena, a multi-scale model is necessary, where one model focuses on the scale of the radius of a strand or the thickness of fabrics and another model focuses on the surrounding bulk liquid. In this dissertation, we demonstrate the efficacy of our multi-scale models through multiple complicated examples, including tightening a wet towel in **Part I**, performing a hair flip in **Part II**, and spreading oil paint with a brush in **Part III**. Through these examples demonstrated in this dissertation, we argue that a diffusive/surface flow model is crucial for visual plausibility.

17.1 Summary of Contributions

To simulate the small-scale details economically, in **Part I**, we develop a convection-diffusion model derived from the mixture theory [8] for the liquid propagating through fabric; in **Part II**, we modify the shallow water equation [205] to simulate a Newtonian liquid flowing along a strand; and in **Part III**, we generalize the shallow water equation to simulate an elastoviscoplastic material flowing on a strand, where a Newtonian flow becomes a special case of our model. The capability of capturing these small-scale details makes the methods introduced in this dissertation significantly different from the prior works.

To simulate a complex scenario, we also need other supportive components. In **Part I**, we deliver a model to approximate the anisotropic fabric microstructure. In **Part II**, we deliver an analytical cohesion model, and in the same chapter, we also propose a liquid dripping and capturing scheme. In **Part III**, we derive an additional inertia term applied to the strands. These components are significant enhancements to our surface/diffusive flow model, enabling us to exchange mass and momentum between the surface flow and other phases (strands, fabrics, bulk liquid), as well as to stably and plausibly simulate cohesive and frictional effects between strands.

More importantly, we develop the coupling schemes between the bulk liquid and the strand/fabric submerged. In **Part I**, we treat the fabric as continuum porous media and couple it with bulk liquid using the *mixture theory*. However, upon discovering that it would be inefficient to simulate strands as a continuum, we turned

our attention to CFD-DEM, where the solid phase is instead modeled as discrete elements. In **Part II**, we adopt a simple coupling scheme, whereas in **Part III**, we formally build our coupling scheme from CFD-DEM, for the DERs, the liquid as a continuum, and the surface flow on strands.

Finally, to support all the physical models we presented in this dissertation, we also develop novel numerical techniques. In **Part I**, we split a stiff, non-symmetric, and nearly ill-conditioned system into three symmetric, positive and definite systems at the cost of introducing some indiscernible divergence; in **Part II**, we introduce a local-global splitting scheme making the conjugate gradient solver for hair dynamics to be several magnitudes faster; and in **Part III**, we derive an analytical form of the plastic flow in Herschel-Bulkley fluid, which, with a semi-implicit solver, enables moderate time-stepping for the shear-dependent bulk liquid.

17.2 Our Recommendation

During our exploration through these three parts, we have learned several conclusions about which way is better than the others for each specific phase of the simulation.

Although recently, the technique of simulating thin structures as a continuum has been significantly advanced (e.g., [95, 121]), we still recommend simulating stiff thin structures as discrete elements. This recommendation comes from both performance and quality considerations. When simulating these thin structures as discrete elements, each group of connected elements can be solved individually in parallel. Since

these groups contain only a small number of degree of freedom, a direct solver for each group is possible. This strategy is more efficient than simulating these strands as a continuum, where everything is solved as a whole system with an iterative solver. Also, solving the strands as discrete elements will avoid the “linearization” artifact raised in continuous representation, since the latter describes the motion of a group of strands only with a linear deformation gradient.

We also conclude that using a full-implicit integrator for simulating the thin structures is necessary, mainly when the scenario contains considerable drag, collision, or cohesive force. These external forces would quickly reshape the thin structures so that the latter would suffer from large stretching or bending energy. An explicit solver, or a linearized-implicit solver, does not fit for a system containing large non-linearity, and thus, is often unstable. This consideration further blocks the use of a continuum representation of the thin structures, because a full-implicit integrator for stiff continuum can be very expensive.

From the modeling side, we conclude that a reduced-dimensional model (e.g., a surface flow, or a convection-diffusion model) is crucial for producing plausible visual appearance. First, the reduced model itself would provide rich details of liquid flowing on the thin structures. Secondly, since the cohesion force depends on the thickness of liquid on the thin structures, a reduced-dimensional model also helps with correctly computing cohesion effects.

When the thin structures submerge into the liquid, most (if not all) interaction

effects between the thin structures and the liquid are contributed by the drag force and liquid pressure. Therefore, correctly computing these terms is crucial for plausible motion. Especially when considering liquid beyond water, using a material-specific drag coefficient would produce dramatically different effects than using a constant (or less carefully considered) drag coefficient. We have shown much difference in the *Drag Force* example of **Part III**.

Finally, we conclude that although a penalty-based contact model is simple, a constraint-based contact model is necessary for practice. With a large cohesion force, a penalty-based contact model would quickly produce tunneling artifact, unless the stiffness of collision is manually tuned individually for different parts of a simulation as well as for its different time, which seems to be a painful or even impossible task. On the other hand, a constraint-based contact model can be used without much tuning, where accurate frictional effect also comes natively.

Based on these considerations, we argue that at the time of the thesis's publication, the framework presented in **Part III** is so far the *most principled, stable, and effective* method that delivers a plausible simulation involving the coupling between liquid and strands. Although not been explored yet, extending the framework to clothes seems to be trivial, where one may easily replace the surface flow with the convection-diffusion model proposed in **Part I**.

17.3 Perspectives

As mentioned in the final sections of each chapter, the models and numerical methods in this dissertation still have many factors that need to be improved in future research: the lack of rigorous laboratory experiment for validation, the ignorance of correct contact angle between the liquid and strand/fabric surface, the lack of a principled physical model (and solver) for liquid capturing and dripping, the smearing of drag force due to limited grid resolution, and the fact that both the liquid and strands are not guaranteed to be stable for an arbitrarily large time step.

As observed in all the examples, the most unsatisfactory element is the surface reconstruction, but developing a stable, accurate, and efficient surface reconstructor is beyond the scope of this dissertation. Throughout the three chapters, we adopt Houdini’s [214] standard particle level set (PLS) method. This standard PLS method produces a popping artifact when the liquid particles vary greatly in their radii. For Newtonian liquids such as water, maintaining smoothness contradicts with preserving the details in splashes. For shear-dependent liquids such as creams, the method also cannot accurately reproduce the thin sheets or tendrils due to its ignorance to the deformation of a single particle.

Besides the issues on quality, our reconstruction workflow also consumes overwhelmingly large memory. Because we need to capture the liquid on strands, both the VDB grid and the resulting mesh inevitably have very high resolutions. For several large examples (e.g., a *Shaking “dog”*, the *Car wash roller* in **Part II**, and the

Chocolate “Dog” in **Part III**), producing high quality reconstruction requires more than 128 GB of physical memory during the surface reconstruction, whereas the generated mesh of one frame, even compressed, can be 1.8 GB or larger. This situation highlights the need for a new rendering algorithm that can render the liquid around strands or fabrics without requiring fine mesh reconstruction.

Despite these limitations, we hope this dissertation may deliver, or even deepen, the current understanding of the complex phenomena involved in the interactions between liquid and thin structures. We also believe many of the limitations described above can provide exciting fodder for future research. Beyond wet hairs or clothes, we hope the theories and methods in this dissertation can inspire studies on other porous media, such as wet sand or a wet sponge, and the fluidization of granular materials.

Bibliography

- [1] Keita Abe, Kenichi Soga, and Samila Bandara. “Material point method for coupled hydromechanical problems.” In: *Journal of Geotechnical and Geoenvironmental Engineering* 140.3 (2013), p. 04013033.
- [2] Vincent Acary and Bernard Brogliato. *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics*. Springer Science & Business Media, 2008.
- [3] Neha Agarwal and RP Chhabra. “Settling velocity of cubes in Newtonian and power law liquids.” In: *Powder Technology* 178.1 (2007), pp. 17–21.
- [4] KJ Ahn, JC Seferis, and JC Berg. “Simultaneous measurements of permeability and capillary pressure of thermosetting matrices in woven fabric reinforcements.” In: *Polymer Composites* 12.3 (1991), pp. 146–152.
- [5] Ömer Akgiray and Ahmet M Saatçι. “A new look at filter backwash hydraulics.” In: *Water Science and Technology: Water Supply* 1.2 (2001), pp. 65–72.
- [6] Nadir Akinci et al. “Coupling elastic solids with smoothed particle hydrodynamics fluids.” In: *Computer Animation and Virtual Worlds* 24.3-4 (2013), pp. 195–203.
- [7] SC Amico and C Lekakou. “Axial impregnation of a fiber bundle. Part 2: theoretical analysis.” In: *Polymer composites* 23.2 (2002), pp. 264–273.
- [8] T B Anderson and Roy Jackson. “Fluid mechanical description of fluidized beds. Equations of motion.” In: *Industrial & Engineering Chemistry Fundamentals* 6.4 (1967), pp. 527–539.
- [9] Ryoichi Ando, Nils Thurey, and Reiji Tsuruno. “Preserving fluid sheets with adaptively sampled anisotropic particles.” In: *IEEE Transactions on Visualization and Computer Graphics* 18.8 (2012), pp. 1202–1214.

- [10] Ryoichi Ando and Reiji Tsuruno. “A particle-based method for preserving fluid sheets.” In: *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*. ACM. 2011, pp. 7–16.
- [11] Nicolas Andreini, Gaël Epely-Chauvin, and Christophe Ancey. “Internal dynamics of Newtonian and viscoplastic fluid avalanches down a sloping bed.” In: *Physics of Fluids* 24.5 (2012), p. 053101.
- [12] Hesam Anvari Ardakani, Evan Mitsoulis, and Savvas G Hatzikiriakos. “Capillary flow of milk chocolate.” In: *Journal of Non-Newtonian Fluid Mechanics* 210 (2014), pp. 56–65.
- [13] Larry Armijo. “Minimization of functions having Lipschitz continuous first partial derivatives.” In: *Pacific Journal of Mathematics* 16.1 (1966), pp. 1–3.
- [14] Teodor M Atanackovic and Arđeshir Guran. *Theory of elasticity for scientists and engineers*. Springer Science & Business Media, 2012.
- [15] DD Atapattu, RP Chhabra, and PHT Uhlherr. “Creeping sphere motion in Herschel-Bulkley fluids: flow field and drag.” In: *Journal of non-newtonian fluid mechanics* 59.2-3 (1995), pp. 245–265.
- [16] Vinicius C Azevedo, Christopher Batty, and Manuel M Oliveira. “Preserving geometry and topology for fluid flows with thin obstacles and narrow gaps.” In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), p. 97.
- [17] P Bagchi and S Balachandar. “Effect of free rotation on the motion of a solid sphere in linear shear flow at moderate Re .” In: *Physics of Fluids* 14.8 (2002), pp. 2719–2737.
- [18] Samila Bandara and Kenichi Soga. “Coupling of soil deformation and pore fluid flow using material point method.” In: *Computers and geotechnics* 63 (2015), pp. 199–214.
- [19] David Baraff and Andrew Witkin. “Large Steps in Cloth Simulation.” In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’98. New York, NY, USA: ACM, 1998, pp. 43–54. ISBN: 0-89791-999-8.
- [20] C Barba et al. “Moisture sorption/desorption of protein fibres.” In: *Thermochimica acta* 552 (2013), pp. 70–76.
- [21] Adam W Bargteil et al. “A finite element method for animating large viscoplastic flow.” In: *ACM transactions on graphics (TOG)* 26.3 (2007), p. 16.

- [22] Christopher Batty, Florence Bertails, and Robert Bridson. “A Fast Variational Framework for Accurate Solid-fluid Coupling.” In: *ACM Transactions on Graphics (TOG)* 26.3 (July 2007). ISSN: 0730-0301.
- [23] Jacob Bear. *Dynamics of fluids in porous media*. Courier Corporation, 2013.
- [24] M Beaulne and E Mitsoulis. “Creeping motion of a sphere in tubes filled with Herschel–Bulkley fluids.” In: *Journal of non-newtonian fluid mechanics* 72.1 (1997), pp. 55–71.
- [25] A Bedford and D S Drumheller. “Theories of immiscible and structured mixtures.” In: *International Journal of Engineering Science* 21.8 (1983), pp. 863–960.
- [26] Miklós Bergou et al. “Discrete Elastic Rods.” In: *ACM Transactions on Graphics (SIGGRAPH)* 27.3 (2008), 63:1–63:12.
- [27] Miklos Bergou et al. “Discrete elastic rods.” In: *ACM Trans. Graph. (SIGGRAPH)* 27.3 (2008), p. 63.
- [28] Miklós Bergou et al. “Discrete viscous threads.” In: *ACM Transactions on Graphics (TOG)* 29.4 (2010), p. 116.
- [29] AN Beris et al. “Creeping motion of a sphere through a Bingham plastic.” In: *Journal of Fluid Mechanics* 158 (1985), pp. 219–244.
- [30] Florence Bertails-Descoubes et al. “A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies.” In: *ACM Transactions on Graphics (TOG)* 30.1 (2011), p. 6.
- [31] Florence Bertails et al. “Super-helices for predicting the dynamics of natural hair.” In: *ACM Transactions on Graphics (TOG)* 25.3 (2006), pp. 1180–1187.
- [32] Bharat Bhushan, Guohua Wei, and Paul Haddad. “Friction and wear studies of human hair and skin.” In: *Wear* 259.7-12 (2005), pp. 1012–1021.
- [33] José Bico et al. “Adhesion: elastocapillary coalescence in wet hair.” In: *Nature* 432.7018 (2004), p. 690.
- [34] José Bico, Étienne Reyssat, and Benoît Roman. “Elastocapillarity: When Surface Tension Deforms Elastic Solids.” In: *Annual Review of Fluid Mechanics* 50.1 (2018), pp. 629–659.
- [35] Maurice A Biot. “General theory of three-dimensional consolidation.” In: *Journal of applied physics* 12.2 (1941), pp. 155–164.

- [36] Pavel Bochev, Denis Ridzal, and Mikhail Shashkov. “Fast optimization-based conservative remap of scalar fields through aggregate mass transfer.” In: *Journal of Computational Physics* 246 (2013), pp. 37–57.
- [37] AA Bochkarev et al. “Formal rheological model of acrylic waterborne paints.” In: *Theoretical Foundations of Chemical Engineering* 43.1 (2009), pp. 100–107.
- [38] Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.
- [39] Ronaldo I Borja. “On the mechanical energy and effective stress in saturated and unsaturated porous continua.” In: *International Journal of Solids and Structures* 43.6 (2006), pp. 1764–1786.
- [40] Mario Botsch et al. *Polygon mesh processing*. CRC press, 2010.
- [41] Percy Williams Bridgman. “Linear Compressions to 30,000 kg/cm², including Relatively Incompressible Substances.” In: *Proceedings of the American Academy of Arts and Sciences* 77.6 (1949), pp. 189–234. ISSN: 01999818.
- [42] Percy Williams Bridgman. *The physics of high pressure*. London: Bells and Sons, 1949.
- [43] Robert Bridson. *Fluid simulation for computer graphics*. CRC Press, 2015.
- [44] Robert Bridson, Sebastian Marino, and Ronald Fedkiw. “Simulation of clothing with folds and wrinkles.” In: *ACM SIGGRAPH 2005 Courses*. ACM. 2005, p. 3.
- [45] F Brochard and PG De Gennes. “Shear-dependent slippage at a polymer/solid interface.” In: *Langmuir* 8.12 (1992), pp. 3033–3037.
- [46] Royal Harvard Brooks and Arthur Thomas Corey. “Hydraulic properties of porous media.” In: *Hydrology papers (Colorado State University)* 3 (1964).
- [47] Richard L Burden and J Douglas Faires. “2.2 Fixed-Point Iteration.” In: *Numerical Analysis (3rd ed.)*. PWS Publishers. ISBN 0-87150-857-5 (1985).
- [48] Philip Crosbie Carman. “Fluid flow through granular beds.” In: *Transactions-Institution of Chemical Engineers* 15 (1937), pp. 150–166.
- [49] Augustin Louis Baron Cauchy. *Exercices de mathématiques*. Vol. 3. Bure frères, 1828.

- [50] Menglei Chai, Changxi Zheng, and Kun Zhou. “A reduced model for interactive hairs.” In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), p. 124.
- [51] Menglei Chai, Changxi Zheng, and Kun Zhou. “Adaptive Skinning for Interactive Hair-Solid Simulation.” In: *IEEE Trans. on Vis. and Comp. Graph.* (2016).
- [52] Yujun Chen, Nadia Magnenat Thalmann, and Brian Foster Allen. “Physical simulation of wet clothing for virtual humans.” In: *The Visual Computer* 28.6-8 (2012), pp. 765–774.
- [53] Zhili Chen et al. “Wetbrush: GPU-based 3D painting simulation at the bristle level.” In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), p. 200.
- [54] Raj P Chhabra. *Bubbles, drops, and particles in non-Newtonian fluids*. CRC press, 2006.
- [55] Nelson S.-H. Chu and Chiew-Lan Tai. “MoXi: Real-time Ink Dispersion in Absorbent Paper.” In: *ACM Transactions on Graphics (TOG)* 24.3 (July 2005), pp. 504–511. ISSN: 0730-0301.
- [56] S Chwastiak. “A wicking method for measuring wetting properties of carbon yarns.” In: *Journal of Colloid and Interface Science* 42.2 (1973), pp. 298–309.
- [57] Gabriel Cirio et al. “Yarn-level simulation of woven cloth.” In: *ACM Transactions on Graphics (TOG)* 33.6 (2014), p. 207.
- [58] Philippe Coussot. *Mudflow rheology and dynamics*. Routledge, 2017.
- [59] R. V. Craster and O. K. Matar. “On viscous beads flowing down a vertical wire.” In: *J. Fluid Mech.* 553 (2006), pp. 85–105.
- [60] Cassidy J. Curtis et al. “Computer-generated Watercolor.” In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 421–430. ISBN: 0-89791-896-7.
- [61] Trong Dang-Vu and Jan Hupka. “Characterization of porous materials by capillary rise method.” In: *Physicochemical problems of mineral processing* 39 (2005), pp. 47–65.
- [62] Henry Philibert Gaspard Darcy. *Détermination des lois d’écoulement de l’eau à travers le sable*. 1856.

- [63] Brojeswari Das et al. “Effect of fibre diameter and cross-sectional shape on moisture transmission through fabrics.” In: *Fibers and Polymers* 9.2 (2008), pp. 225–231.
- [64] Brojeswari Das et al. “Moisture transmission through textiles.” In: *Part II: evaluation methods and mathematical modeling. Autex Res J* 7.3 (2007), pp. 194–216.
- [65] Gilles Daviet. “Modèles et algorithmes pour la simulation du contact frottant dans les matériaux complexes: application aux milieux fibreux et granulaires.” PhD thesis. 2016.
- [66] Gilles Daviet. *So-Bogus*. <https://bitbucket.org/gdaviet/so-bogus/src/master/>. 2013.
- [67] Gilles Daviet and Florence Bertails-Descoubes. “A semi-implicit material point method for the continuum simulation of granular materials.” In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), p. 102.
- [68] Gilles Daviet and Florence Bertails-Descoubes. “Simulation of Drucker–Prager granular flows inside Newtonian fluids.” working paper or preprint. Feb. 2017.
- [69] Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. “A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics.” In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 6. ACM. 2011, p. 139.
- [70] Reint De Boer. *Theory of porous media: highlights in historical development and current state*. Springer Science & Business Media, 2012.
- [71] G De Saxcé and Z-Q Feng. “The bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms.” In: *Mathematical and computer modelling* 28.4-8 (1998), pp. 225–245.
- [72] Crispin Deul et al. “Direct Position-Based Solver for Stiff Rods.” In: *Computer Graphics Forum* 37.6 (2018), pp. 313–324.
- [73] R Di Felice. “The voidage function for fluid-particle interaction systems.” In: *International Journal of Multiphase Flow* 20.1 (1994), pp. 153–159.
- [74] Andrew K Dickerson, Zachary G Mills, and David L Hu. “Wet mammals shake at tuned frequencies to dry.” In: *Journal of the Royal Society Interface* 9.77 (2012), pp. 3208–3218.
- [75] Weta Digital. *YacFS*. 2013.

- [76] C. Duprat et al. “Wetting of flexible fibre arrays.” In: *Nature* 482.7386 (2012), pp. 510–513.
- [77] Sabri Ergun. “Fluid flow through packed columns.” In: *Chem. Eng. Prog.* 48 (1952), pp. 89–94.
- [78] Yu Fang et al. “Silly Rubber: An Implicit Material Point Method for Simulating Non-equilibrated Viscoelastic and Elastoplastic Solids.” In: *ACM Transactions on Graphics (TOG)* 38.4 (2019).
- [79] Adolf Fick. “Ueber diffusion.” In: *Annalen der Physik* 170.1 (1855), pp. 59–86.
- [80] Paul Fillunger. “Der auftrieb in talsperren.” In: *Osterr. Wochenschrift fur den offentl. Baudienst* 19.32 (1913), pp. 532–555.
- [81] Philipp Forchheimer. “Wasserbewegung durch boden.” In: *Z. Ver. Deutsch. Ing* 45.1782 (1901), p. 1788.
- [82] Chuyuan Fu et al. “A Polynomial Particle-In-Cell Method.” In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), p. 222.
- [83] Masao Fukushima, Zhi-Quan Luo, and Paul Tseng. “Smoothing functions for second-order-cone complementarity problems.” In: *SIAM Journal on optimization* 12.2 (2002), pp. 436–460.
- [84] Ming Gao et al. “An adaptive generalized interpolation material point method for simulating elastoplastic materials.” In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), p. 223.
- [85] Ming Gao et al. “Animating fluid sediment mixture in particle-laden flows.” In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), p. 149.
- [86] Ming Gao et al. “GPU Optimization of Material Point Methods.” In: *ACM Transactions on Graphics (TOG)* 37.6 (2018).
- [87] Jorge Gascón, Javier S Zurdo, and Miguel A Otaduy. “Constraint-Based Simulation of Adhesive Contact.” In: *Symposium on Computer Animation*. 2010, pp. 39–44.
- [88] J Gaume et al. “Dynamic anticrack propagation in snow.” In: *Nature Communications* 9.1 (2018), p. 3047.
- [89] Jean-Frédéric Gerbeau and Benoît Perthame. “Derivation of Viscous Saint-Venant System for Laminar Shallow Water; Numerical Validation.” In: *Discrete and Continuous Dynamical Systems-Series B* 1.1 (2001), pp. 89–102.

- [90] Galen Gornowicz and Silviu Borac. “Efficient and stable approach to elasticity and collisions for hair animation.” In: *Proceedings of the 2015 Symposium on Digital Production*. ACM. 2015, pp. 41–49.
- [91] Mireille Grégoire and Elmar Schömer. “Interactive simulation of one-dimensional flexible parts.” In: *Computer-Aided Design* 39.8 (2007), pp. 694–707.
- [92] Eitan Grinspun et al. “Discrete Shells.” In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA ’03. San Diego, California: Eurographics Association, 2003, pp. 62–67. ISBN: 1-58113-659-5.
- [93] Eran Guendelman et al. “Coupling water and smoke to thin deformable and rigid shells.” In: *ACM Transactions on Graphics (TOG)* 24.3 (2005), pp. 973–981.
- [94] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. <http://eigen.tuxfamily.org>. 2010.
- [95] Qi Guo et al. “A material point method for thin shells with frictional contact.” In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), p. 147.
- [96] Sunil Hadap and Nadia Magnenat-Thalmann. “Modeling dynamic hair as a continuum.” In: *Computer Graphics Forum* 20.3 (2001), pp. 329–338.
- [97] Sunil Hadap et al. “Strands and hair - Modeling, simulation and rendering.” In: *SIGGRAPH Courses*. 2007, pp. 1–150.
- [98] Dongsoo Han and Takahiro Harada. “Tridiagonal matrix formulation for inextensible hair strand simulation.” In: *Proceedings of the Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association, 2013.
- [99] Fathi Hassine. “A review of depth-averaged models for dry granular flows: Savage-Hutter and $\mu(I)$ -rheology.” In: *arXiv preprint arXiv:1303.2838* (2013).
- [100] Henry Selby Hele-Shaw. “The flow of water.” In: *Nature* 58.1489 (1898), pp. 33–36.
- [101] JGI Hellström and TS Lundström. “Flow through porous media at moderate Reynolds number.” In: *International Scientific Colloquium: Modelling for Material Processing*. Vol. 2. 2006, pp. 129–134.

- [102] Winslow H Herschel and Ronald Bulkley. “Konsistenzmessungen von gummi-benzollösungen.” In: *Colloid & Polymer Science* 39.4 (1926), pp. 291–300.
- [103] JE Hilton, LR Mason, and PW Cleary. “Dynamics of gas–solid fluidised beds with non-spherical particle geometry.” In: *Chemical Engineering Science* 65.5 (2010), pp. 1584–1596.
- [104] Yuanming Hu et al. “A moving least squares material point method with displacement discontinuity and two-way rigid body coupling.” In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), p. 150.
- [105] Markus Huber, Bernhard Eberhardt, and Daniel Weiskopf. “Boundary handling at cloth–fluid contact.” In: *Computer Graphics Forum* 34.1 (2015), pp. 14–25.
- [106] Markus Huber, Simon Pabst, and Wolfgang Straßer. “Wet cloth simulation.” In: *ACM SIGGRAPH 2011 Posters*. ACM, 2011, p. 10.
- [107] Hayley Iben et al. “Artistic simulation of curly hair.” In: *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2013, pp. 63–71.
- [108] Markus Ihmsen et al. “Implicit incompressible SPH.” In: *IEEE Transactions on Visualization and Computer Graphics* 20.3 (2014), pp. 426–435.
- [109] Ioan R Ionescu. “Augmented Lagrangian for shallow viscoplastic flow with topography.” In: *Journal of Computational Physics* 242 (2013), pp. 544–560.
- [110] Ioan R Ionescu. “Onset and dynamic shallow flow of a viscoplastic fluid on a plane slope.” In: *Journal of non-newtonian fluid mechanics* 165.19-20 (2010), pp. 1328–1341.
- [111] Ioan R Ionescu. “Viscoplastic shallow flow equations with topography.” In: *Journal of Non-Newtonian Fluid Mechanics* 193 (2013), pp. 116–128.
- [112] Ioan R Ionescu and Oana Lupaşcu. “Modeling shallow avalanche onset over complex basal topography.” In: *Advances in Computational Mathematics* 42.1 (2016), pp. 5–26.
- [113] Ioan R Ionescu et al. “Viscoplastic modeling of granular column collapse with pressure-dependent rheology.” In: *Journal of Non-Newtonian Fluid Mechanics* 219 (2015), pp. 1–18.
- [114] Fridtjov Irgens. *Continuum mechanics*. Springer Science & Business Media, 2008.

- [115] Fridtjov Irgens. “Generalized Newtonian Fluids.” In: *Rheology and Non-Newtonian Fluids*. Cham: Springer International Publishing, 2014, pp. 113–124. ISBN: 978-3-319-01053-3. DOI: 10.1007/978-3-319-01053-3_6.
- [116] Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. “Invertible finite elements for robust simulation of large deformation.” In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2004, pp. 131–140.
- [117] M Khalid Jawed, Alyssa Novelia, and Oliver M O'Reilly. *A primer on the kinematics of discrete elastic rods*. Springer, 2018.
- [118] Michel Jean. “The non-smooth contact dynamics method.” In: *Computer Methods in Applied Mechanics and Engineering* 177.3-4 (1999), pp. 235–257.
- [119] Michel Jean, Vincent Acary, and Yann Monerie. “Non-smooth contact dynamics approach of cohesive materials.” In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 359.1789 (2001), pp. 2497–2518.
- [120] Henrik Wann Jensen, Justin Legakis, and Julie Dorsey. “Rendering of Wet Materials.” In: *Proceedings of the 10th Eurographics Conference on Rendering*. EGWR'99. Granada, Spain: Eurographics Association, 1999, pp. 273–282. ISBN: 3-211-83382-X.
- [121] Chenfanfu Jiang, Theodore Gast, and Joseph Teran. “Anisotropic elastoplasticity for cloth, knit and hair frictional contact.” In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 152.
- [122] Chenfanfu Jiang et al. “The affine particle-in-cell method.” In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), p. 51.
- [123] Chenfanfu Jiang et al. “The material point method for simulating continuum materials.” In: *ACM SIGGRAPH 2016 Courses*. ACM, 2016, p. 24.
- [124] Ben Jones et al. “Deformation embedding for point-based elastoplastic simulation.” In: *ACM Transactions on Graphics (TOG)* 33.2 (2014), p. 21.
- [125] Dirk Kadau et al. “Contact dynamics simulations of compacting cohesive granular systems.” In: *Computer Physics Communications* 147.1-2 (2002), pp. 190–193.
- [126] Jonathan M Kaldor, Doug L James, and Steve Marschner. “Efficient yarn-based cloth with adaptive contact linearization.” In: *ACM Transactions on Graphics (TOG)* 29.4 (2010), p. 105.

- [127] Jonathan M. Kaldor, Doug L. James, and Steve Marschner. “Simulating Knitted Cloth at the Yarn Level.” In: *ACM Transactions on Graphics (TOG)* 27.3 (Aug. 2008), 65:1–65:9. ISSN: 0730-0301.
- [128] Michael Kass and Gavin Miller. “Rapid, Stable Fluid Dynamics for Computer Graphics.” In: *SIGGRAPH Comput. Graph.* 24.4 (Sept. 1990), pp. 49–57. ISSN: 0097-8930.
- [129] Danny M Kaufman et al. “Adaptive nonlinearity for collisions in complex rod assemblies.” In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), p. 123.
- [130] Danny Kaufman. *ADONIS*. <https://bitbucket.org/nepluno/figarobogusfluids/src/columbia-DK-nonlinear-response/>. 2013.
- [131] VC Kelessidis et al. “Optimal determination of rheological parameters for Herschel–Bulkley drilling fluids and impact on pressure drop, velocity profiles and penetration rates during drilling.” In: *Journal of Petroleum Science and Engineering* 53.3-4 (2006), pp. 203–224.
- [132] HS Kim. “In-plane liquid distribution in nonwoven fabrics: Part 2–simulation.” In: *Int. Nonwoven J.* 12 (2003), pp. 29–33.
- [133] G Kirchhoff. “Ueber das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes.” In: *Journal für die reine und angewandte Mathematik* 56 (1859), pp. 285–313.
- [134] Erik Kiss. “Wetting and wicking.” In: *Textile Research Journal* 66.10 (1996), pp. 660–668.
- [135] Gergely Klár et al. “Drucker-prager elastoplasticity for sand animation.” In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), p. 103.
- [136] Tassilo Kugelstadt and Elmar Schömer. “Position and orientation based Cosserat rods.” In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association. 2016, pp. 169–178.
- [137] JAM Kuipers et al. “A numerical model of gas-fluidized beds.” In: *Chemical Engineering Science* 47.8 (1992), pp. 1913–1924.
- [138] Claude Lacourriere. “Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts.” In: (2007).
- [139] Hanna G Lagger et al. “Influence of hydrodynamic drag model on shear stress in the simulation of magnetorheological fluids.” In: *Journal of Non-Newtonian Fluid Mechanics* 218 (2015), pp. 16–26.

- [140] Horace Lamb. *Hydrodynamics*. Cambridge university press, 1993.
- [141] Mark Landeryou, Ian Eames, and A Cottenden. “Infiltration into inclined fibrous sheets.” In: *Journal of Fluid Mechanics* 529 (2005), pp. 173–193.
- [142] Egor Larionov, Christopher Batty, and Robert Bridson. “Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids.” In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 101.
- [143] Minjae Lee et al. “A skinned tetrahedral mesh for hair animation and hair-water interaction.” In: *IEEE Transactions on Visualization and Computer Graphics* 25.3 (2019), pp. 1449–1459.
- [144] C Lekakou and MG Bader. “Mathematical modelling of macro-and micro-infiltration in resin transfer moulding (RTM).” In: *Composites Part A: Applied Science and Manufacturing* 29.1-2 (1998), pp. 29–37.
- [145] Toon Lenaerts, Bart Adams, and Philip Dutré. “Porous flow in particle-based fluid simulations.” In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), p. 49.
- [146] Xingyue Li and Jidong Zhao. “Dam-break of mixtures consisting of non-Newtonian liquids and granular particles.” In: *Powder Technology* 338 (2018), pp. 493–505.
- [147] Guoping Lian, Colin Thornton, and Michael J Adams. “A theoretical study of the liquid bridge forces between two rigid spherical bodies.” In: *Journal of colloid and interface science* 161.1 (1993), pp. 138–147.
- [148] Wei-Chin Lin. “Boundary handling and porous flow for fluid–hair interactions.” In: *Computers & Graphics* 52 (2015), pp. 33–42.
- [149] Wei-Chin Lin. “Coupling hair with smoothed particle hydrodynamics fluids.” In: *Proceedings of the Workshop on Virtual Reality Interaction and Physical Simulation*. Ed. by Jan Bender et al. The Eurographics Association, 2014. ISBN: 978-3-905674-71-2.
- [150] T Liu, KF Choi, and Y Li. “Capillary rise between cylinders.” In: *Journal of physics D: Applied physics* 40.16 (2007), p. 5006.
- [151] Élise Lorenceau, Christophe Clanet, and David Quéré. “Capturing drops with a thin fiber.” In: *Journal of colloid and interface science* 279.1 (2004), pp. 192–197.

- [152] Lin Lou, Feng Ji, and Yiping Qiu. “Simulating adhesion of wet fabrics to water: surface tension-based theoretical model and experimental verification.” In: *Textile Research Journal* 85.19 (2015), pp. 1987–1998.
- [153] Lin Lou et al. “Simulating adhesion of wet fabrics to water: Gravity of liquid bridge-based theoretical model and experimental verification.” In: *Textile Research Journal* 87.7 (2017), pp. 769–779.
- [154] Lin Lou et al. “The influence of surface hydrophilicity on the adhesion properties of wet fabrics or films to water.” In: *Textile Research Journal* 88.1 (2018), pp. 108–117.
- [155] AEH Love. “The Small Free Vibrations and Deformation of a Thin Elastic Shell.” In: *Philosophical Transactions of the Royal Society of London. A* 179 (1888), pp. 491–546.
- [156] R Lucas. “Rate of capillary ascension of liquids.” In: *Kolloid Z* 23.15 (1918), pp. 15–22.
- [157] Hernán A Makse, David L Johnson, and Lawrence M Schwartz. “Packing of compressible granular materials.” In: *Physical review letters* 84.18 (2000), p. 4160.
- [158] Reza Masoodi and Krishna M Pillai. “A general formula for capillary suction-pressure in porous media.” In: *Journal of Porous Media* 15.8 (2012).
- [159] Reza Masoodi and Krishna M Pillai. *Wicking in porous materials: traditional and modern modeling approaches*. CRC Press, 2012.
- [160] Reza Masoodi, Krishna M Pillai, and Padma Prabodh Varanasi. “Role of hydraulic and capillary radii in improving the effectiveness of capillary model in wicking.” In: *ASME Summer Conference, Jacksonville, FL, USA*. 2008.
- [161] Evelyne Mauret and Maurice Renaud. “Transport phenomena in multi-particle systems – I. Limits of applicability of capillary model in high voidage beds-application to fixed beds of fibers and fluidized beds of spheres.” In: *Chemical Engineering Science* 52.11 (1997), pp. 1807–1817.
- [162] Aleka McAdams et al. “Detail preserving continuum simulation of straight hair.” In: *ACM Transactions on Graphics (TOG)* 28.3 (2009), p. 62.
- [163] José Meseguer and Angel Sanz. “Numerical and experimental study of the dynamics of axisymmetric liquid bridges.” In: *Journal of Fluid Mechanics* 153 (1985), pp. 83–101.

- [164] R v Mises. “Mechanik der festen Körper im plastisch-deformablen Zustand.” In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1913 (1913), pp. 582–592.
- [165] Rajat Mittal and Gianluca Iaccarino. “Immersed boundary methods.” In: *Annu. Rev. Fluid Mech.* 37 (2005), pp. 239–261.
- [166] Joe J Monaghan. “Simulating free surface flows with SPH.” In: *Journal of computational physics* 110.2 (1994), pp. 399–406.
- [167] Jean J Moreau. “Unilateral contact and dry friction in finite freedom dynamics.” In: *Nonsmooth Mechanics and Applications*. Springer, 1988, pp. 1–82.
- [168] Matthias Müller, David Charypar, and Markus Gross. “Particle-based fluid simulation for interactive applications.” In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 154–159.
- [169] Matthias Müller, Tae-Yong Kim, and Nuttapong Chentanez. “Fast simulation of inextensible hair and fur.” In: *Proceedings of the Workshop on Virtual Reality Interaction and Physical Simulation*. Vol. 12. 2012, pp. 39–44.
- [170] Ken Museth. “A flexible image processing approach to the surfacing of particle-based fluid animation (invited talk).” In: *Mathematical progress in expressive image synthesis I*. Springer, 2014, pp. 81–84.
- [171] Ken Museth. “VDB: High-resolution sparse volumes with dynamic topology.” In: *ACM Transactions on Graphics (TOG)* 32.3 (2013), p. 27.
- [172] Kentaro Nagasawa et al. “Mixing Sauces: A Viscosity Blending Model for Shear Thinning Fluids.” In: *ACM Transactions on Graphics (TOG)* 38.4 (2019).
- [173] Michael B Nielsen and Ole Østerby. “A two-continua approach to Eulerian simulation of water spray.” In: *ACM Transactions on Graphics (TOG)* 32.4 (2013), p. 67.
- [174] P Nithiarasu, KN Seetharamu, and T Sundararajan. “Natural convective heat transfer in a fluid saturated variable porosity medium.” In: *International Journal of Heat and Mass Transfer* 40.16 (1997), pp. 3955–3967.
- [175] Hilary Ockendon and John R Ockendon. *Viscous flow*. Vol. 13. Cambridge University Press, 1995.
- [176] Alexander Oron, Stephen H Davis, and S George Bankoff. “Long-scale evolution of thin liquid films.” In: *Reviews of modern physics* 69.3 (1997), p. 931.

- [177] Oktar Ozgen et al. “Underwater cloth simulation with fractional derivatives.” In: *ACM Transactions on Graphics (TOG)* 29.3 (2010), p. 23.
- [178] Claudio Paniconi, Alvaro A Aldama, and Eric F Wood. “Numerical evaluation of iterative and noniterative methods for the solution of the nonlinear Richards equation.” In: *Water Resources Research* 27.6 (1991), pp. 1147–1163.
- [179] Saket Patkar and Parag Chaudhuri. “Wetting of porous solids.” In: *IEEE transactions on visualization and computer graphics* 19.9 (2013), pp. 1592–1604.
- [180] Amalendu Patnaik et al. “Wetting and wicking in fibrous materials.” In: *Textile Progress* 38.1 (2006), pp. 1–105.
- [181] Charles S Peskin. “The immersed boundary method.” In: *Acta numerica* 11 (2002), pp. 479–517.
- [182] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. “The matrix cookbook.” In: *Technical University of Denmark* 7.15 (2008), p. 510.
- [183] Lena Petrovic, Mark Henne, and John Anderson. “Volumetric methods for simulation and rendering of hair.” In: *Pixar Animation Studios* 2.4 (2005).
- [184] Krishna M Pillai and Suresh G Advani. “Wicking across a fiber-bank.” In: *Journal of colloid and interface science* 183.1 (1996), pp. 100–110.
- [185] Remmelt Pit, Hubert Hervet, and Liliane Léger. “Friction and slip of a simple liquid at a solid surface.” In: *Tribology Letters* 7.2-3 (1999), pp. 147–152.
- [186] E Bruce Pitman and Long Le. “A two-fluid model for avalanche and debris flows.” In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 363.1832 (2005), pp. 1573–1601.
- [187] HM Princen. “Capillary phenomena in assemblies of parallel cylinders: I. Capillary rise between two cylinders.” In: *Journal of Colloid and Interface Science* 30.1 (1969), pp. 69–75.
- [188] HM Princen. “Capillary phenomena in assemblies of parallel cylinders: II. Capillary rise in systems with more than two cylinders.” In: *Journal of Colloid and Interface Science* 30.3 (1969), pp. 359–371.
- [189] HM Princen. “Capillary phenomena in assemblies of parallel cylinders: III. Liquid columns between horizontal parallel cylinders.” In: *Journal of Colloid and Interface Science* 34.2 (1970), pp. 171–184.

- [190] John W Pritchett. “A numerical model of gas fluidized beds.” In: *AICHE Symp.* 1978, pp. 134–148.
- [191] C. Py et al. “3D aggregation of wet fibers.” In: *Europhysics letters* 77.4 (2007), p. 44005.
- [192] Kumbakonam Ramamani Rajagopal and Lu Tao. *Mechanics of mixtures*. World Scientific, 1995.
- [193] P Rajitha et al. “Drag on non-spherical particles in power law non-Newtonian media.” In: *International Journal of Mineral Processing* 78.2 (2006), pp. 110–121.
- [194] Daniel Ram et al. “A material point method for viscoelastic fluids, foams and sponges.” In: *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM. 2015, pp. 157–163.
- [195] Michel Raous, Laurent Cangémi, and Marius Cocu. “A consistent model coupling adhesion, friction, and unilateral contact.” In: *Computer methods in applied mechanics and engineering* 177.3-4 (1999), pp. 383–399.
- [196] Bo Ren et al. “Multiple-fluid SPH simulation using a mixture model.” In: *ACM Transactions on Graphics (TOG)* 33.5 (2014), p. 171.
- [197] Maurice Renaud, Evelyne Mauret, and Rajendra P Chhabra. “Power-law fluid flow over a sphere: Average shear rate and drag coefficient.” In: *The Canadian Journal of Chemical Engineering* 82.5 (2004), pp. 1066–1070.
- [198] Lorenzo Adolph Richards. “Capillary conduction of liquids through porous mediums.” In: *Physics* 1.5 (1931), pp. 318–333.
- [199] RS Rivlin. “Large elastic deformations of isotropic materials IV. Further developments of the general theory.” In: *Philosophical Transactions of the Royal Society of London A* 241.835 (1948), pp. 379–397.
- [200] Avi Robinson-Mosher et al. “Two-way Coupling of Fluids to Rigid and Deformable Solids and Shells.” In: *ACM Transactions on Graphics (TOG)* 27.3 (Aug. 2008), 46:1–46:9. ISSN: 0730-0301.
- [201] Benoit Roman and José Bico. “Elasto-capillarity: deforming an elastic structure with a liquid droplet.” In: *Journal of Physics: Condensed Matter* 22.49 (2010), p. 493101.
- [202] W. Rungjirathananon, Y. Kanamori, and T. Nishita. “Wetting effects in hair simulation.” In: *Computer Graphics Forum* 31.7 (2012), pp. 1993–2002.

- [203] Witawat Rungjirathananon et al. “Real-time Animation of Sand-Water Interaction.” In: *Computer Graphics Forum* 27.7 (2008), pp. 1887–1893.
- [204] Yousef Saad. *Iterative methods for sparse linear systems*. Vol. 82. siam, 2003.
- [205] M de Saint-Venant. “Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l’introduction des marées dans leur lit.” In: (1871). Extrait des Comptes rendus de l’Académie des Sciences, tome LXXIII, séances des 17 et 24 juillet 1871, 11 p.
- [206] Pierre Saramito and Anthony Wachs. “Progress in numerical simulation of yield stress fluid flows.” In: *Rheologica Acta* 56.3 (2017), pp. 211–230.
- [207] Luc Scholtès et al. “On the capillary stress tensor in wet granular materials.” In: *International journal for numerical and analytical methods in geomechanics* 33.10 (2009), pp. 1289–1313.
- [208] WR Schowalter. “The behavior of complex fluids at solid boundaries.” In: *Journal of Non-Newtonian Fluid Mechanics* 29 (1988), pp. 25–36.
- [209] Aviv Segall, Orestis Vantzos, and Mirela Ben-Chen. “Hele-shaw Flow Simulation with Interactive Control Using Complex Barycentric Coordinates.” In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA ’16. Zurich, Switzerland: Eurographics Association, 2016, pp. 85–95. ISBN: 978-3-905674-61-3.
- [210] Andrew Selle, Michael Lentine, and Ronald Fedkiw. “A mass spring model for hair simulation.” In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), p. 64.
- [211] MT Senoguz et al. “Simulations and experiments on low-pressure permeation of fabrics: Part II – The variable gap model and prediction of permeability.” In: *Journal of composite materials* 35.14 (2001), pp. 1285–1322.
- [212] Martin Servin, Claude Lacoursiere, and Niklas Melin. “Interactive simulation of elastic deformable materials.” In: *SIGRAD 2006. The Annual SIGRAD Conference; Special Theme: Computer Games*. 019. Linköping University Electronic Press. 2006.
- [213] Toshihiro Shinohara et al. “Extraction of yarn positional information from a three-dimensional CT image of textile fabric using yarn tracing with a filament model for structure analysis.” In: *Textile Research Journal* 80.7 (2010), pp. 623–630.
- [214] SideFX. *SideFX Houdini*. <https://www.sidefx.com>. 2019.

- [215] Juan C Simo. “A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part I. Continuum formulation.” In: *Computer methods in applied mechanics and engineering* 66.2 (1988), pp. 199–219.
- [216] Juan C Simo. “A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. Part II: computational aspects.” In: *Computer methods in applied mechanics and engineering* 68.1 (1988), pp. 1–31.
- [217] Juan C Simo and Thomas JR Hughes. *Computational inelasticity*. Vol. 7. Springer Science & Business Media, 2006.
- [218] Kiran Singh, John R. Lister, and Dominic Vella. “A fluid-mechanical model of elastocapillary coalescence.” In: *J. Fluid Mech.* 745 (2014), pp. 621–646.
- [219] Breannan Smith, Fernando De Goes, and Theodore Kim. “Stable neo-hookean flesh simulation.” In: *ACM Transactions on Graphics (TOG)* 37.2 (2018), p. 12.
- [220] Taha Sochi. “Slip at fluid-solid interface.” In: *Polymer Reviews* 51.4 (2011), pp. 309–340.
- [221] Carlota Soler, Tobias Martin, and Olga Sorkine-Hornung. “Cosserat Rods with Projective Dynamics.” In: *Computer Graphics Forum* 37.8 (2018), pp. 137–147.
- [222] Xiaoyu Song and Ronaldo I Borja. “Mathematical framework for unsaturated flow in the finite deformation range.” In: *International Journal for Numerical Methods in Engineering* 97.9 (2014), pp. 658–682.
- [223] Jonas Spillmann and Matthias Teschner. “CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects.” In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer animation*. Eurographics Association. 2007, pp. 63–72.
- [224] Jos Stam. “Stable fluids.” In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 121–128.
- [225] Alexey Stomakhin et al. “A material point method for snow simulation.” In: *ACM Transactions on Graphics (TOG)* 32.4 (2013), p. 102.
- [226] Alexey Stomakhin et al. “Augmented MPM for phase-change and varied materials.” In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), p. 138.

- [227] Triantafylllos Stylianopoulos et al. “Permeability calculations in three-dimensional isotropic and oriented fiber networks.” In: *Physics of Fluids (1994-present)* 20.12 (2008), p. 123601.
- [228] Deborah Sulsky, Zhen Chen, and Howard L Schreyer. “A particle method for history-dependent materials.” In: *Computer methods in applied mechanics and engineering* 118.1-2 (1994), pp. 179–196.
- [229] Hervé Tabuteau, Philippe Coussot, and John R de Bruyn. “Drag force on a sphere in steady motion through a yield-stress fluid.” In: *Journal of rheology* 51.1 (2007), pp. 125–137.
- [230] Andre Pradhana Tampubolon et al. “Multi-species simulation of porous sand and water mixtures.” In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 105.
- [231] Karl von Terzaghi. “Die berechnung der durchlassigkeitsziffer des tones aus dem verlauf der hydrodynamischen spannungsscheinungen.” In: *Sitzungsberichte der Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Klasse, Abteilung IIa* 132 (1923), pp. 125–138.
- [232] Karl von Terzaghi. *Theoretical soil mechanics*. Vol. 18. Wiley Online Library, 1943.
- [233] Bernhard Thomaszewski et al. “Advanced Topics in Virtual Garment Simulation.” In: *Eurographics 2007 - Tutorials*. Ed. by Karol Myszkowski and Vlastimil Havran. The Eurographics Association, 2007.
- [234] Maxime Tournier et al. “Stable constrained dynamics.” In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), p. 132.
- [235] Kiwon Um, Xiangyu Hu, and Nils Thuerey. “Perceptual evaluation of liquid simulation methods.” In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 143.
- [236] Kiwon Um et al. “Porous deformable shell simulation with surface water flow and saturation.” In: *Computer Animation and Virtual Worlds* 24.3-4 (2013), pp. 247–254.
- [237] M Th Van Genuchten. “A closed-form equation for predicting the hydraulic conductivity of unsaturated soils.” In: *Soil science society of America journal* 44.5 (1980), pp. 892–898.
- [238] Orestis Vantzios, Saar Raz, and Mirela Ben-Chen. “Real-time Viscous Thin Films.” In: *ACM Trans. Graph.* 37.6 (Nov. 2018), 281:1–281:10.

- [239] Orestis Vantzos et al. “Functional Thin Films on Surfaces.” In: *IEEE transactions on visualization and computer graphics* 23.3 (2017), pp. 1179–1192.
- [240] Alexander Virozub, Nir Haimovich, and Simon Brandon. “Three-dimensional simulations of liquid bridges between two cylinders: forces, energies, and torques.” In: *Langmuir* 25.22 (2009), pp. 12837–12842.
- [241] C. B. Vreugdenhil. *Numerical methods for shallow-water flow*. 1994, p. 262.
- [242] Huamin Wang, Gavin Miller, and Greg Turk. “Solving general shallow wave equations on surfaces.” In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2007, pp. 229–238.
- [243] Qianbin Wang et al. “Chinese brushes: Controllable liquid transfer in ratchet conical hairs.” In: *Advanced Materials* 26.28 (2014), pp. 4889–4894.
- [244] Yongxin Wang, Stephen Michelsen, and Hoon Joo Lee. “Symmetric and asymmetric capillary bridges between a rough surface and a parallel surface.” In: *Langmuir* 29.35 (2013), pp. 11028–11037.
- [245] Kelly Ward, Nico Galoppo, and Ming C Lin. “Interactive virtual hair salon.” In: *Presence: Teleoperators and Virtual Environments* 16.3 (2007), pp. 237–251.
- [246] Kelly Ward and Ming C Lin. “Adaptive grouping and subdivision for simulating hair dynamics.” In: *Proceedings of 11th Pacific Conference on Computer Graphics and Applications*. IEEE. 2003, pp. 234–243.
- [247] Kelly Ward et al. “A survey on hair modeling: Styling, simulation, and rendering.” In: *IEEE TVCG* 13.2 (2007), pp. 213–234.
- [248] Kelly Ward et al. “Modeling hair using level-of-detail representations.” In: *Proceedings of 16th International Conference on Computer Animation and Social Agents*. IEEE. 2003, pp. 41–47.
- [249] Edward W Washburn. “The dynamics of capillary flow.” In: *Physical review* 17.3 (1921), p. 273.
- [250] Denis L Weaire and Stefan Hutzler. *The physics of foams*. Oxford University Press, 2001.
- [251] Martin Wicke et al. “Dynamic local remeshing for elastoplastic simulation.” In: *ACM Transactions on graphics (TOG)*. Vol. 29. 4. ACM. 2010, p. 49.

- [252] JG Williams, CEM Morris, and BC Ennis. “Liquid flow through aligned fiber beds.” In: *Polymer Engineering & Science* 14.6 (1974), pp. 413–419.
- [253] Rene Winchenbach, Hendrik Hochstetter, and Andreas Kolb. “Infinite continuous adaptivity for incompressible SPH.” In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 102.
- [254] Joshuah Wolper et al. “CD-MPM: Continuum Damage Material Point Methods for Dynamic Fracture Animation.” In: *ACM Transactions on Graphics (TOG)* 38.4 (2019).
- [255] Reinhard Woltmann. *Beiträge zur Hydraulischen Architectur*. Vol. 2. Dieterich, 1792.
- [256] Xiao Yan et al. “Multiphase sph simulation for interactive fluids and solids.” In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), p. 79.
- [257] Tao Yang et al. “A unified particle system framework for multi-phase, multi-material visual simulations.” In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), p. 224.
- [258] Tao Yang et al. “Fast multiple-fluid simulation using Helmholtz free energy.” In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), p. 201.
- [259] K Yazdchi and Stefan Luding. “Towards unified drag laws for inertial flow through fibrous materials.” In: *Chemical engineering journal* 207 (2012), pp. 35–48.
- [260] Thomas Young. “An essay on the cohesion of fluids.” In: *Philosophical Transactions of the Royal Society of London* 95 (1805), pp. 65–87.
- [261] Yonghao Yue et al. “Continuum foam: A material point method for shear-dependent flows.” In: *ACM Transactions on Graphics (TOG)* 34.5 (2015), p. 160.
- [262] Yonghao Yue et al. “Hybrid Grains: Adaptive Coupling of Discrete and Continuum Simulations of Granular Media.” In: *ACM Trans. Graph.* 37.6 (Nov. 2018), 283:1–283:19.
- [263] Xinxin Zhang. *A TBB Parallelized Liquid Solver Featuring Simple FLIP and AMGPCG Pressure Solver*. https://github.com/zhxx1987/tbb_liquid_amgpcg. 2015.

- [264] Wenqi Zhong et al. “DEM/CFD-DEM modelling of non-spherical particulate systems: theoretical developments and applications.” In: *Powder Technology* 302 (2016), pp. 108–152.
- [265] ZY Zhou et al. “Discrete particle simulation of particle–fluid flow: model formulations and their applicability.” In: *Journal of Fluid Mechanics* 661 (2010), pp. 482–510.
- [266] Bo Zhu et al. “Codimensional non-Newtonian fluids.” In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), p. 115.
- [267] Yongning Zhu and Robert Bridson. “Animating sand as a fluid.” In: *ACM Transactions on Graphics (TOG)* 24.3 (2005), pp. 965–972.

Appendix: Gradient and Hessians of the Discrete Curvatures in Discrete Elastic Rods

In this chapter we derive the gradient and Hessians of the discrete curvature used in discrete elastic rod (DER). Although very lengthy, the Hessian is necessary when one implicitly integrates the bending force of DERs.

Motivation. In the literature, there are multiple models for discrete elastic rods. We adopt the definition of the discrete curvatures (§17.3) following the *original work* of Bergou et al. [26]. In a following work, Bergou et al. [28] replaced these definition by projecting the curvature vector $\kappa\mathbf{b}$ to the neighbor material vectors \mathbf{m}^{i-1} and \mathbf{m}^i and combining the results. Although this latter form is simpler, i.e., the four curvatures used in their prior work [26] are reduced to two terms, it is problematic — mathematically, it is meaningless to combine the $\kappa\mathbf{b}$ projected into different frames. Hence, in this paper, we still follow the original definition of discrete curvatures [26], but replaced the space-parallel transport with time-parallel transport when computing the reference vector. The formulation of the discrete curvatures is the same as the one taken by Kaldor et al. [126]. Nevertheless, we follow the recent book by Jawed et al. [117] to present a rigorous derivation of the gradient and the Hessians of the discrete curvatures.

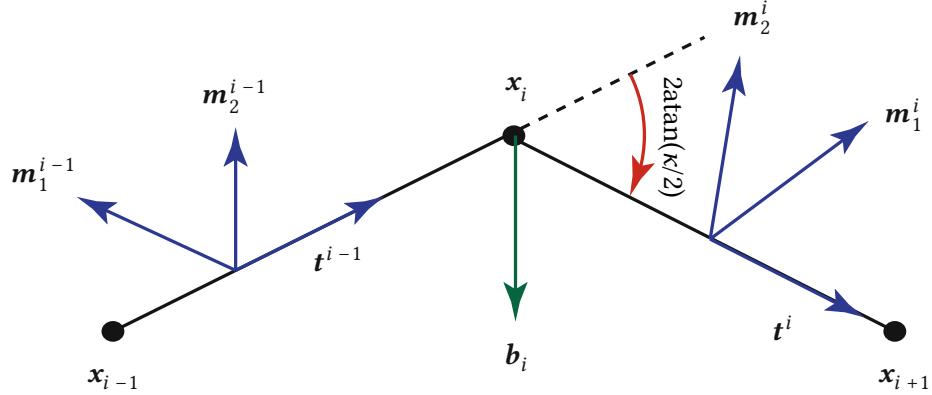


Figure A1: Discrete elastic rods, adapted from the book by Jawed et al. [117].

Integrated Curvature Vector

We begin with the definition of the curvature vector $\kappa \mathbf{b}$ at a vertex i . Similar as prior works [26, 126], we take the tangent of the half angle at vertex i as the discrete curvature.

$$(\kappa \mathbf{b})_i = \frac{2\mathbf{t}^{i-1} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} = \frac{2\mathbf{e}^{i-1} \times \mathbf{e}^i}{\|\mathbf{e}^{i-1}\| \|\mathbf{e}^i\| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i} \quad (\text{A1})$$

where \mathbf{t}^i is the normalized tangent vector at edge i , and \mathbf{e}^i is the edge vector itself (so that $\mathbf{t}^i = \mathbf{e}^i / \|\mathbf{e}^i\|$).

We then derive the variation of this curvature vector, which has the following form

$$\begin{aligned} \delta(\kappa \mathbf{b})_i &= \frac{2\delta\mathbf{e}^{i-1} \times \mathbf{e}^i}{\|\mathbf{e}^{i-1}\| \|\mathbf{e}^i\| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i} + \frac{2\mathbf{e}^{i-1} \times \delta\mathbf{e}^i}{\|\mathbf{e}^{i-1}\| \|\mathbf{e}^i\| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i} \\ &\quad - \frac{(\mathbf{e}^i + \|\mathbf{e}^i\| \mathbf{t}^{i-1}) \cdot \delta\mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\| \|\mathbf{e}^i\| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i} (\kappa \mathbf{b})_i - \frac{(\mathbf{e}^{i-1} + \|\mathbf{e}^{i-1}\| \mathbf{t}^i) \cdot \delta\mathbf{e}^i}{\|\mathbf{e}^{i-1}\| \|\mathbf{e}^i\| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i} (\kappa \mathbf{b})_i \end{aligned} \quad (\text{A2})$$

By dividing by the magnitude of the edge vectors we have

$$\begin{aligned} \delta(\kappa \mathbf{b})_i &= \frac{2\frac{\delta\mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} + \frac{2\mathbf{t}^{i-1} \times \frac{\delta\mathbf{e}^i}{\|\mathbf{e}^i\|}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \\ &\quad - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa \mathbf{b})_i \left(\frac{\delta\mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} + \frac{\delta\mathbf{e}^i}{\|\mathbf{e}^i\|} \right) \end{aligned} \quad (\text{A3})$$

Material Vectors

We compute the variation of the material vectors \mathbf{m}_1^i and \mathbf{m}_2^i for defined at edge i .

Following Jawed et al. [117], the temporal derivative of the material vectors are

$$\dot{\mathbf{m}}_1^i(t) = \dot{\gamma}^i(t)\mathbf{m}_2^i(t) - \left(\mathbf{m}_1^i(t) \cdot \dot{\mathbf{t}}^i(t)\right) \mathbf{t}^i(t), \quad (\text{A4a})$$

$$\dot{\mathbf{m}}_2^i(t) = -\dot{\gamma}^i(t)\mathbf{m}_1^i(t) - \left(\mathbf{m}_2^i(t) \cdot \dot{\mathbf{t}}^i(t)\right) \mathbf{t}^i(t). \quad (\text{A4b})$$

where γ is the angle between *reference* vector and material vector, and only depends on the twist of rods. So the first term reflects the change of twist along the complement material vector, and the second term reflects the change of direction of the edges. When the position of vertices are disturbed, the change of twist $\dot{\gamma}$ is zero. Actually, we have

$$\delta\mathbf{m}_1^i = \delta\gamma^i\mathbf{m}_2^i - \left(\mathbf{m}_1^i \cdot \delta\mathbf{t}^i\right) \mathbf{t}^i, \quad (\text{A5a})$$

$$\delta\mathbf{m}_2^i = -\delta\gamma^i\mathbf{m}_1^i - \left(\mathbf{m}_2^i \cdot \delta\mathbf{t}^i\right) \mathbf{t}^i. \quad (\text{A5b})$$

Discrete Curvatures

Our definition of discrete curvatures follows Kaldor et al. [126], where one vertex at i generates four terms regards to its previous and next edges

$$\kappa_{i,1}^{i-1} = \mathbf{m}_2^{i-1} \cdot (\kappa\mathbf{b})_i, \quad (\text{A6a})$$

$$\kappa_{i,1}^i = \mathbf{m}_2^i \cdot (\kappa\mathbf{b})_i, \quad (\text{A6b})$$

$$\kappa_{i,2}^{i-1} = -\mathbf{m}_1^{i-1} \cdot (\kappa\mathbf{b})_i, \quad (\text{A6c})$$

$$\kappa_{i,2}^i = -\mathbf{m}_1^i \cdot (\kappa\mathbf{b})_i. \quad (\text{A6d})$$

We compute the variation of these curvatures, where we have

$$\delta\kappa_{i,1}^{i-1} = \mathbf{m}_2^{i-1} \cdot \delta(\kappa\mathbf{b})_i + \delta\mathbf{m}_2^{i-1} \cdot (\kappa\mathbf{b})_i, \quad (\text{A7a})$$

$$\delta\kappa_{i,1}^i = \mathbf{m}_2^i \cdot \delta(\kappa\mathbf{b})_i + \delta\mathbf{m}_2^i \cdot (\kappa\mathbf{b})_i, \quad (\text{A7b})$$

$$\delta\kappa_{i,2}^{i-1} = -\mathbf{m}_1^{i-1} \cdot \delta(\kappa\mathbf{b})_i - \delta\mathbf{m}_1^{i-1} \cdot (\kappa\mathbf{b})_i, \quad (\text{A7c})$$

$$\delta\kappa_{i,2}^i = -\mathbf{m}_1^i \cdot \delta(\kappa\mathbf{b})_i - \delta\mathbf{m}_1^i \cdot (\kappa\mathbf{b})_i. \quad (\text{A7d})$$

When only the positions of vertices are changing, we know from equation A5a that $\delta\gamma^i = 0$ and $\delta\mathbf{m}^i$ is parallel with the tangential direction \mathbf{t}^i . Hence $\delta\mathbf{m}^i$ is orthogonal with $(\kappa\mathbf{b})_i$. Therefore, we have the terms $\delta\mathbf{m}_1^i \cdot (\kappa\mathbf{b})_i = 0$ and $\delta\mathbf{m}_2^i \cdot (\kappa\mathbf{b})_i = 0$. For similar reason, $\delta\mathbf{m}_1^{i-1} \cdot (\kappa\mathbf{b})_i$ and $\delta\mathbf{m}_2^{i-1} \cdot (\kappa\mathbf{b})_i$ are also zero. We then have the following variations of curvatures

$$\delta\kappa_{i,1}^{i-1} = \mathbf{m}_2^{i-1} \cdot \delta(\kappa\mathbf{b})_i - \delta\gamma^{i-1} \mathbf{m}_1^{i-1}(\kappa\mathbf{b})_i, \quad (\text{A8a})$$

$$\delta\kappa_{i,1}^i = \mathbf{m}_2^i \cdot \delta(\kappa\mathbf{b})_i - \delta\gamma^i \mathbf{m}_1^i(\kappa\mathbf{b})_i, \quad (\text{A8b})$$

$$\delta\kappa_{i,2}^{i-1} = -\mathbf{m}_1^{i-1} \cdot \delta(\kappa\mathbf{b})_i - \delta\gamma^{i-1} \mathbf{m}_2^{i-1}(\kappa\mathbf{b})_i, \quad (\text{A8c})$$

$$\delta\kappa_{i,2}^i = -\mathbf{m}_1^i \cdot \delta(\kappa\mathbf{b})_i - \delta\gamma^i \mathbf{m}_2^i(\kappa\mathbf{b})_i. \quad (\text{A8d})$$

To compute the derivatives we apply equation A3 to the variation of curvatures and set $\delta\mathbf{e}^{i-1}$, $\delta\mathbf{e}^i$, $\delta\gamma^{i-1}$ and $\delta\gamma^i$ to zero, respectively. We then have the following terms while the other terms are all zero

$$\frac{\partial\kappa_{i,1}^{i-1}}{\partial\mathbf{e}^{i-1}} \cdot \delta\mathbf{e}^{i-1} = \mathbf{m}_2^{i-1} \cdot \left[\frac{2\frac{\delta\mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa\mathbf{b})_i \cdot \frac{\delta\mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \right], \quad (\text{A9a})$$

$$\frac{\partial\kappa_{i,1}^i}{\partial\mathbf{e}^i} \cdot \delta\mathbf{e}^i = \mathbf{m}_2^i \cdot \left[\frac{2\mathbf{t}^{i-1} \times \frac{\delta\mathbf{e}^i}{\|\mathbf{e}^i\|}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa\mathbf{b})_i \cdot \frac{\delta\mathbf{e}^i}{\|\mathbf{e}^i\|} \right], \quad (\text{A9b})$$

$$\frac{\partial\kappa_{i,1}^{i-1}}{\partial\gamma^{i-1}} \cdot \delta\gamma^{i-1} = -\delta\gamma^{i-1} \mathbf{m}_1^{i-1}(\kappa\mathbf{b})_i, \quad (\text{A9c})$$

$$\frac{\partial \kappa_{i,1}^i}{\partial \mathbf{e}^{i-1}} \cdot \delta \mathbf{e}^{i-1} = \mathbf{m}_2^i \cdot \left[\frac{2 \frac{\delta \mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa \mathbf{b})_i \cdot \frac{\delta \mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \right], \quad (\text{A10a})$$

$$\frac{\partial \kappa_{i,1}^i}{\partial \mathbf{e}^i} \cdot \delta \mathbf{e}^i = \mathbf{m}_2^i \cdot \left[\frac{2 \mathbf{t}^{i-1} \times \frac{\delta \mathbf{e}^i}{\|\mathbf{e}^i\|}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa \mathbf{b})_i \cdot \frac{\delta \mathbf{e}^i}{\|\mathbf{e}^i\|} \right], \quad (\text{A10b})$$

$$\frac{\partial \kappa_{i,1}^i}{\partial \gamma^i} \cdot \delta \gamma^i = -\delta \gamma^i \mathbf{m}_1^i (\kappa \mathbf{b})_i, \quad (\text{A10c})$$

$$\frac{\partial \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^{i-1}} \cdot \delta \mathbf{e}^{i-1} = -\mathbf{m}_1^{i-1} \cdot \left[\frac{2 \frac{\delta \mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa \mathbf{b})_i \cdot \frac{\delta \mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \right], \quad (\text{A11a})$$

$$\frac{\partial \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^i} \cdot \delta \mathbf{e}^i = -\mathbf{m}_1^{i-1} \cdot \left[\frac{2 \mathbf{t}^{i-1} \times \frac{\delta \mathbf{e}^i}{\|\mathbf{e}^i\|}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa \mathbf{b})_i \cdot \frac{\delta \mathbf{e}^i}{\|\mathbf{e}^i\|} \right], \quad (\text{A11b})$$

$$\frac{\partial \kappa_{i,2}^{i-1}}{\partial \gamma^{i-1}} \cdot \delta \gamma^{i-1} = -\delta \gamma^{i-1} \mathbf{m}_2^{i-1} (\kappa \mathbf{b})_i, \quad (\text{A11c})$$

$$\frac{\partial \kappa_{i,2}^i}{\partial \mathbf{e}^{i-1}} \cdot \delta \mathbf{e}^{i-1} = -\mathbf{m}_1^i \cdot \left[\frac{2 \frac{\delta \mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa \mathbf{b})_i \cdot \frac{\delta \mathbf{e}^{i-1}}{\|\mathbf{e}^{i-1}\|} \right], \quad (\text{A12a})$$

$$\frac{\partial \kappa_{i,2}^i}{\partial \mathbf{e}^i} \cdot \delta \mathbf{e}^i = -\mathbf{m}_1^i \cdot \left[\frac{2 \mathbf{t}^{i-1} \times \frac{\delta \mathbf{e}^i}{\|\mathbf{e}^i\|}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} - \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} (\kappa \mathbf{b})_i \cdot \frac{\delta \mathbf{e}^i}{\|\mathbf{e}^i\|} \right], \quad (\text{A12b})$$

$$\frac{\partial \kappa_{i,2}^i}{\partial \gamma^i} \cdot \delta \gamma^i = -\delta \gamma^i \mathbf{m}_2^i (\kappa \mathbf{b})_i. \quad (\text{A12c})$$

By using the rule of triple product and other algebraic manipulations, we achieve the following equations:

$$\frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1}} = \frac{1}{\|\mathbf{e}^{i-1}\|} \left(-\kappa_{i,1}^{i-1} \tilde{\mathbf{t}} + \frac{2 \mathbf{t}^i \times \mathbf{m}_2^{i-1}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right), \quad (\text{A13a})$$

$$\frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^i} = \frac{1}{\|\mathbf{e}^i\|} \left(-\kappa_{i,1}^{i-1} \tilde{\mathbf{t}} - \frac{2 \mathbf{t}^{i-1} \times \mathbf{m}_2^{i-1}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right), \quad (\text{A13b})$$

$$\frac{\partial \kappa_{i,1}^{i-1}}{\partial \gamma^{i-1}} = -\mathbf{m}_1^{i-1} (\kappa \mathbf{b})_i, \quad (\text{A13c})$$

$$\frac{\partial \kappa_{i,1}^i}{\partial \mathbf{e}^{i-1}} = \frac{1}{\|\mathbf{e}^{i-1}\|} \left(-\kappa_{i,1}^i \tilde{\mathbf{t}} + \frac{2\mathbf{t}^i \times \mathbf{m}_2^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right), \quad (\text{A14a})$$

$$\frac{\partial \kappa_{i,1}^i}{\partial \mathbf{e}^i} = \frac{1}{\|\mathbf{e}^i\|} \left(-\kappa_{i,1}^i \tilde{\mathbf{t}} - \frac{2\mathbf{t}^{i-1} \times \mathbf{m}_2^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right), \quad (\text{A14b})$$

$$\frac{\partial \kappa_{i,1}^i}{\partial \gamma^i} = -\mathbf{m}_1^i (\kappa \mathbf{b})_i, \quad (\text{A14c})$$

$$\frac{\partial \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^{i-1}} = \frac{1}{\|\mathbf{e}^{i-1}\|} \left(-\kappa_{i,2}^{i-1} \tilde{\mathbf{t}} - \frac{2\mathbf{t}^i \times \mathbf{m}_1^{i-1}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right), \quad (\text{A15a})$$

$$\frac{\partial \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^i} = \frac{1}{\|\mathbf{e}^i\|} \left(-\kappa_{i,2}^{i-1} \tilde{\mathbf{t}} + \frac{2\mathbf{t}^{i-1} \times \mathbf{m}_1^{i-1}}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right), \quad (\text{A15b})$$

$$\frac{\partial \kappa_{i,2}^{i-1}}{\partial \gamma^{i-1}} = -\mathbf{m}_2^{i-1} (\kappa \mathbf{b})_i, \quad (\text{A15c})$$

$$\frac{\partial \kappa_{i,2}^i}{\partial \mathbf{e}^{i-1}} = \frac{1}{\|\mathbf{e}^{i-1}\|} \left(-\kappa_{i,2}^i \tilde{\mathbf{t}} - \frac{2\mathbf{t}^i \times \mathbf{m}_1^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right), \quad (\text{A16a})$$

$$\frac{\partial \kappa_{i,2}^i}{\partial \mathbf{e}^i} = \frac{1}{\|\mathbf{e}^i\|} \left(-\kappa_{i,2}^i \tilde{\mathbf{t}} + \frac{2\mathbf{t}^{i-1} \times \mathbf{m}_1^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i} \right). \quad (\text{A16b})$$

$$\frac{\partial \kappa_{i,2}^i}{\partial \gamma^i} = -\mathbf{m}_2^i (\kappa \mathbf{b})_i. \quad (\text{A16c})$$

where $\tilde{\mathbf{t}} \equiv \frac{\mathbf{t}^{i-1} + \mathbf{t}^i}{1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i}$.

Hessian of the Discrete Curvatures

Before deriving the Hessian of the curvatures, it is convenient to define several variables and compute their derivatives, as following (\otimes denotes the outer product, e.g., $\mathbf{a} \otimes \mathbf{b} \equiv \mathbf{a}\mathbf{b}^T$.)

$$\frac{\partial \mathbf{t}^i}{\partial \mathbf{e}^i} = \frac{1}{\|\mathbf{e}^i\|} (\mathbf{I}_3 - \mathbf{t}^i \otimes \mathbf{t}^i) \quad (\text{A17a})$$

$$\chi \equiv 1 + \mathbf{t}^{i-1} \cdot \mathbf{t}^i \quad (\text{A17b})$$

$$\frac{\partial \chi}{\partial \mathbf{e}^{i-1}} = \frac{1}{\|\mathbf{e}^{i-1}\|} (\mathbf{I}_3 - \mathbf{t}^{k-1} \otimes \mathbf{t}^{k-1}) \mathbf{t}^k \quad (\text{A17c})$$

$$\frac{\partial \chi}{\partial \mathbf{e}^i} = \frac{1}{\|\mathbf{e}^i\|} (\mathbf{I}_3 - \mathbf{t}^k \otimes \mathbf{t}^k) \mathbf{t}^{k-1} \quad (\text{A17d})$$

$$\frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^{i-1}} = \frac{1}{\chi \|\mathbf{e}^{i-1}\|} ((\mathbf{I}_3 - \mathbf{t}^{i-1} \otimes \mathbf{t}^{i-1}) - \tilde{\mathbf{t}} \otimes ((\mathbf{I}_3 - \mathbf{t}^{i-1} \otimes \mathbf{t}^{i-1}) \mathbf{t}^i)) \quad (\text{A17e})$$

$$\frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^i} = \frac{1}{\chi \|\mathbf{e}^i\|} ((\mathbf{I}_3 - \mathbf{t}^i \otimes \mathbf{t}^i) - \tilde{\mathbf{t}} \otimes ((\mathbf{I}_3 - \mathbf{t}^i \otimes \mathbf{t}^i) \mathbf{t}^{i-1})) \quad (\text{A17f})$$

Besides we have

$$\frac{\partial}{\partial \mathbf{e}^i} (\mathbf{a} \times \mathbf{b}) = [\mathbf{a}]_{\times} \cdot \frac{\partial \mathbf{b}}{\partial \mathbf{e}^i} - [\mathbf{b}]_{\times} \cdot \frac{\partial \mathbf{a}}{\partial \mathbf{e}^i} \quad (\text{A18})$$

for arbitrary vector \mathbf{a} and \mathbf{b} , where the notation $[\cdot]_{\times}$ denotes the cross product matrix such that $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \cdot \mathbf{b}$.

We then compute the Hessian of curvatures, where we have the following second derivative for the first line of equation A13a,

$$\begin{aligned} \frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \mathbf{e}^{i-1}} &= - \left(-\kappa_{i,1}^{i-1} \tilde{\mathbf{t}} + \frac{2\mathbf{t}^i \times \mathbf{m}_2^{i-1}}{\chi} \right) \otimes \frac{\mathbf{t}^{i-1}}{\|\mathbf{e}^{i-1}\|^2} \\ &\quad + \frac{1}{\|\mathbf{e}^{i-1}\|} \left(-\frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1}} \otimes \tilde{\mathbf{t}} - \kappa_{i,1}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^{i-1}} - \frac{2[\mathbf{m}_2^{i-1}]_{\times} \cdot \frac{\partial \mathbf{t}^i}{\partial \mathbf{e}^{i-1}}}{\chi} - \frac{2\mathbf{t}^i \times \mathbf{m}_2^{i-1}}{\chi^2} \otimes \frac{\partial \chi}{\partial \mathbf{e}^{i-1}} \right), \end{aligned} \quad (\text{A19})$$

Noticing that the first term of the above equation contains $\frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1}}$, and $\frac{\partial \mathbf{t}^i}{\partial \mathbf{e}^{i-1}} = 0$, after combining the terms, we have

$$\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \mathbf{e}^{i-1}} = - \frac{1}{\|\mathbf{e}^{i-1}\|} \left(\frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1}} \otimes \mathbf{t}^{i-1} + \tilde{\mathbf{t}} \otimes \frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1}} + \kappa_{i,1}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^{i-1}} + \frac{2\mathbf{t}^i \times \mathbf{m}_2^{i-1}}{\chi^2} \otimes \frac{\partial \chi}{\partial \mathbf{e}^{i-1}} \right). \quad (\text{A20})$$

The Hessian matrix for this part is symmetric. We then simplify it by combining the first and second terms. Also we define

$$\tilde{\mathbf{m}}_2^{i-1} \equiv \frac{2\mathbf{m}_2^{i-1}}{\chi}, \quad (\text{A21a})$$

$$\tilde{\mathbf{m}}_2^i \equiv \frac{2\mathbf{m}_2^i}{\chi}, \quad (\text{A21b})$$

$$\tilde{\mathbf{m}}_1^{i-1} \equiv \frac{2\mathbf{m}_1^{i-1}}{\chi}, \quad (\text{A21c})$$

$$\tilde{\mathbf{m}}_1^i \equiv \frac{2\mathbf{m}_1^i}{\chi}. \quad (\text{A21d})$$

Before going on, remember that we also need the Hessians over the twisting angle γ . To compute these terms we note

$$\begin{aligned} \delta \left(\frac{\partial \kappa_{i,1}^{i-1}}{\partial \gamma^{i-1}} \right) &= -\delta \mathbf{m}_1^{i-1}(\kappa \mathbf{b})_i - \mathbf{m}_1^{i-1} \delta(\kappa \mathbf{b})_i \\ &= -\delta \gamma^{i-1} \mathbf{m}_2^{i-1}(\kappa \mathbf{b})_i - \mathbf{m}_1^{i-1} \delta(\kappa \mathbf{b})_i \end{aligned} \quad (\text{A22})$$

We then derive other Hessians following a similar strategy for deriving (A20), and we use (A22) for the Hessians over γ . We have

$$\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \mathbf{e}^{i-1}} = -\frac{1}{\|\mathbf{e}^{i-1}\|} \text{sym} \left(\frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1}} \otimes (\mathbf{t}^{i-1} + \tilde{\mathbf{t}}) + \kappa_{i,1}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^{i-1}} + \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_2^{i-1}) \otimes \frac{\partial \chi}{\partial \mathbf{e}^{i-1}} \right), \quad (\text{A23a})$$

$$\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^i \partial \mathbf{e}^i} = -\frac{1}{\|\mathbf{e}^i\|} \text{sym} \left(\frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^i} \otimes (\mathbf{t}^i + \tilde{\mathbf{t}}) + \kappa_{i,1}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^i} + \frac{1}{\chi} (\mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_2^{i-1}) \otimes \frac{\partial \chi}{\partial \mathbf{e}^i} \right), \quad (\text{A23b})$$

$$\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^i \partial \mathbf{e}^{i-1}} = \left(\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \mathbf{e}^i} \right)^T \quad (\text{A23c})$$

$$= -\frac{1}{\|\mathbf{e}^{i-1}\|} \left(\tilde{\mathbf{t}} \otimes \frac{\partial \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^i} + \kappa_{i,1}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^i} + \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_2^{i-1}) \otimes \frac{\partial \chi}{\partial \mathbf{e}^i} + [\tilde{\mathbf{m}}_2^{i-1}]_\times \cdot \frac{\partial \mathbf{t}^i}{\partial \mathbf{e}^i} \right), \quad (\text{A23d})$$

$$\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \gamma^{i-1}} = \frac{1}{\|\mathbf{e}^{i-1}\|} (-\kappa_{i,2}^{i-1} \tilde{\mathbf{t}} - \mathbf{t}^i \times \tilde{\mathbf{m}}_1^{i-1}), \quad (\text{A24a})$$

$$\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \mathbf{e}^i \partial \gamma^{i-1}} = \frac{1}{\|\mathbf{e}^i\|} (-\kappa_{i,2}^{i-1} \tilde{\mathbf{t}} + \mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_1^{i-1}), \quad (\text{A24b})$$

$$\frac{\partial^2 \kappa_{i,1}^{i-1}}{\partial \gamma^{i-1} \partial \gamma^{i-1}} = -(\kappa \mathbf{b})_i \cdot \mathbf{m}_2^{i-1}, \quad (\text{A24c})$$

$$\frac{\partial^2 \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \mathbf{e}^{i-1}} = -\frac{1}{\|\mathbf{e}^{i-1}\|} \text{sym} \left(\frac{\partial \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^{i-1}} \otimes (\mathbf{t}^{i-1} + \tilde{\mathbf{t}}) + \kappa_{i,2}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^{i-1}} + \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_1^{i-1}) \otimes \frac{\partial \chi}{\partial \mathbf{e}^{i-1}} \right), \quad (\text{A25a})$$

$$\frac{\partial^2 \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^i \partial \mathbf{e}^i} = -\frac{1}{\|\mathbf{e}^i\|} \text{sym} \left(\frac{\partial \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^i} \otimes (\mathbf{t}^i + \tilde{\mathbf{t}}) + \kappa_{i,2}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^i} + \frac{1}{\chi} (\mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_1^{i-1}) \otimes \frac{\partial \chi}{\partial \mathbf{e}^i} \right), \quad (\text{A25b})$$

$$\frac{\partial^2 \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^i \partial \mathbf{e}^{i-1}} = \left(\frac{\partial^2 \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \mathbf{e}^i} \right)^T \quad (\text{A25c})$$

$$= -\frac{1}{\|\mathbf{e}^{i-1}\|} \left(\tilde{\mathbf{t}} \otimes \frac{\partial \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^i} + \kappa_{i,2}^{i-1} \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^i} - \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_1^{i-1}) \otimes \frac{\partial \chi}{\partial \mathbf{e}^i} - [\tilde{\mathbf{m}}_1^{i-1}]_\times \cdot \frac{\partial \mathbf{t}^i}{\partial \mathbf{e}^i} \right), \quad (\text{A25d})$$

$$\frac{\partial^2 \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^{i-1} \partial \gamma^{i-1}} = \frac{1}{\|\mathbf{e}^{i-1}\|} (\kappa_{i,1}^{i-1} \tilde{\mathbf{t}} - \mathbf{t}^i \times \tilde{\mathbf{m}}_2^{i-1}), \quad (\text{A26a})$$

$$\frac{\partial^2 \kappa_{i,2}^{i-1}}{\partial \mathbf{e}^i \partial \gamma^{i-1}} = \frac{1}{\|\mathbf{e}^i\|} (\kappa_{i,1}^{i-1} \tilde{\mathbf{t}} + \mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_2^{i-1}), \quad (\text{A26b})$$

$$\frac{\partial^2 \kappa_{i,2}^{i-1}}{\partial \gamma^{i-1} \partial \gamma^{i-1}} = (\kappa \mathbf{b})_i \cdot \mathbf{m}_1^{i-1}, \quad (\text{A26c})$$

$$\frac{\partial^2 \kappa_{i,1}^i}{\partial \mathbf{e}^{i-1} \partial \mathbf{e}^{i-1}} = -\frac{1}{\|\mathbf{e}^{i-1}\|} \text{sym} \left(\frac{\partial \kappa_{i,1}^i}{\partial \mathbf{e}^{i-1}} \otimes (\mathbf{t}^{i-1} + \tilde{\mathbf{t}}) + \kappa_{i,1}^i \frac{\partial \tilde{\mathbf{t}}}{\partial \mathbf{e}^{i-1}} + \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_2^i) \otimes \frac{\partial \chi}{\partial \mathbf{e}^{i-1}} \right), \quad (\text{A27a})$$

$$\frac{\partial^2 \kappa_{i,1}^i}{\partial e^i \partial e^i} = -\frac{1}{\|e^i\|} \text{sym} \left(\frac{\partial \kappa_{i,1}^i}{\partial e^i} \otimes (\mathbf{t}^i + \tilde{\mathbf{t}}) + \kappa_{i,1}^i \frac{\partial \tilde{\mathbf{t}}}{\partial e^i} + \frac{1}{\chi} (\mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_2^i) \otimes \frac{\partial \chi}{\partial e^i} \right),$$

(A27b)

$$\begin{aligned} \frac{\partial^2 \kappa_{i,1}^i}{\partial e^i \partial e^{i-1}} &= \left(\frac{\partial^2 \kappa_{i,1}^i}{\partial e^{i-1} \partial e^i} \right)^T \\ &= -\frac{1}{\|e^{i-1}\|} \left(\tilde{\mathbf{t}} \otimes \frac{\partial \kappa_{i,1}^i}{\partial e^i} + \kappa_{i,1}^i \frac{\partial \tilde{\mathbf{t}}}{\partial e^i} + \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_2^i) \otimes \frac{\partial \chi}{\partial e^i} + [\tilde{\mathbf{m}}_2^i]_\times \cdot \frac{\partial \mathbf{t}^i}{\partial e^i} \right), \end{aligned}$$

(A27c)

(A27d)

$$\frac{\partial^2 \kappa_{i,1}^i}{\partial e^{i-1} \partial \gamma^i} = \frac{1}{\|e^{i-1}\|} (-\kappa_{i,2}^i \tilde{\mathbf{t}} - \mathbf{t}^i \times \tilde{\mathbf{m}}_1^i),$$

(A28a)

$$\frac{\partial^2 \kappa_{i,1}^i}{\partial e^i \partial \gamma^i} = \frac{1}{\|e^i\|} (-\kappa_{i,2}^i \tilde{\mathbf{t}} + \mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_1^i),$$

(A28b)

$$\frac{\partial^2 \kappa_{i,1}^i}{\partial \gamma^i \partial \gamma^i} = -(\kappa \mathbf{b})_i \cdot \mathbf{m}_2^i,$$

(A28c)

$$\frac{\partial^2 \kappa_{i,2}^i}{\partial e^{i-1} \partial e^{i-1}} = -\frac{1}{\|e^{i-1}\|} \text{sym} \left(\frac{\partial \kappa_{i,2}^i}{\partial e^{i-1}} \otimes (\mathbf{t}^{i-1} + \tilde{\mathbf{t}}) + \kappa_{i,2}^i \frac{\partial \tilde{\mathbf{t}}}{\partial e^{i-1}} + \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_1^i) \otimes \frac{\partial \chi}{\partial e^{i-1}} \right),$$

(A29a)

$$\frac{\partial^2 \kappa_{i,2}^i}{\partial e^i \partial e^i} = -\frac{1}{\|e^i\|} \text{sym} \left(\frac{\partial \kappa_{i,2}^i}{\partial e^i} \otimes (\mathbf{t}^i + \tilde{\mathbf{t}}) + \kappa_{i,2}^i \frac{\partial \tilde{\mathbf{t}}}{\partial e^i} + \frac{1}{\chi} (\mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_1^i) \otimes \frac{\partial \chi}{\partial e^i} \right),$$

(A29b)

$$\begin{aligned} \frac{\partial^2 \kappa_{i,2}^i}{\partial e^i \partial e^{i-1}} &= \left(\frac{\partial^2 \kappa_{i,2}^i}{\partial e^{i-1} \partial e^i} \right)^T \\ &= -\frac{1}{\|e^{i-1}\|} \left(\tilde{\mathbf{t}} \otimes \frac{\partial \kappa_{i,2}^i}{\partial e^i} + \kappa_{i,2}^i \frac{\partial \tilde{\mathbf{t}}}{\partial e^i} - \frac{1}{\chi} (\mathbf{t}^i \times \tilde{\mathbf{m}}_1^i) \otimes \frac{\partial \chi}{\partial e^i} - [\tilde{\mathbf{m}}_1^i]_\times \cdot \frac{\partial \mathbf{t}^i}{\partial e^i} \right). \end{aligned}$$

(A29c)

(A29d)

$$\frac{\partial^2 \kappa_{i,2}^i}{\partial e^{i-1} \partial \gamma^i} = \frac{1}{\|e^{i-1}\|} (\kappa_{i,1}^i \tilde{\mathbf{t}} - \mathbf{t}^i \times \tilde{\mathbf{m}}_2^i),$$

(A30a)

$$\frac{\partial^2 \kappa_{i,2}^i}{\partial e^i \partial \gamma^i} = \frac{1}{\|e^i\|} (\kappa_{i,1}^i \tilde{\mathbf{t}} + \mathbf{t}^{i-1} \times \tilde{\mathbf{m}}_2^i),$$

(A30b)

$$\frac{\partial^2 \kappa_{i,2}^i}{\partial \gamma^i \partial \gamma^i} = (\kappa \mathbf{b})_i \cdot \mathbf{m}_1^i, \quad (\text{A30c})$$

where we use the notation $\text{sym}(\mathbf{A}) \equiv (\mathbf{A} + \mathbf{A}^T)/2$. The Hessian terms other than the ones above are all zero filled.

The total bending energy for a strand is then defined over curvatures

$$E_b = \frac{1}{4} \sum_i \sum_{j=0,1} [\kappa_{i,1}^{i-j} - \bar{\kappa}_{i,1}^{i-j}, \kappa_{i,2}^{i-j} - \bar{\kappa}_{i,2}^{i-j}] B_i [\kappa_{i,1}^{i-j} - \bar{\kappa}_{i,1}^{i-j}, \kappa_{i,2}^{i-j} - \bar{\kappa}_{i,2}^{i-j}]^T. \quad (\text{A31})$$

where $B_i \in \mathbb{R}^{2 \times 2}$ is the bending stiffness tensor at vertex i , and the variables with a bar denote the rest states. With the gradient and Hessians of curvatures given above, the bending force and its Jacobian can be trivially computed, following Kaldor et al. [126].