Supplemental: A Multi-Scale Model for Simulating Liquid-Fabric Interactions

YUN (RAYMOND) FEI, Columbia University, USA
CHRISTOPHER BATTY, University of Waterloo, Canada
EITAN GRINSPUN and CHANGXI ZHENG, Columbia University, USA

ACM Reference Format:
Yun (Raymond) Fei, Christopher Batty, Eitan Grinspun, and Changxi Zheng. 2018. Supplemental: A Multi-Scale Model for Simulating Liquid-Fabric Interactions. *ACM Trans. Graph.* 37, 4, Article 51 (August 2018), 2 pages. https://doi.org/10.1145/3197517.3201392

1 DERIVATION OF EQUATION 24

Explicit integration of the solid and liquid dynamics yields the following equations:

$$(\mathbf{M}_\mathrm{s} + h[\mathbf{C}]\mathbf{V}_c)\,\boldsymbol{u}_\mathrm{s}^{n+1} - h[\mathbf{C}]\mathbf{V}_c\boldsymbol{u}_\mathrm{f}^{n+1} + h\mathbf{V}_\mathrm{s}\mathbf{G}\boldsymbol{p}^{n+1} = h\boldsymbol{f}_\mathrm{s} + \mathbf{M}_\mathrm{s}\boldsymbol{u}_\mathrm{s}^{n}, \tag{1}$$

$$(\mathbf{M}_\mathrm{f} + h[\mathbf{C}]\mathbf{V}_c)\,\boldsymbol{u}_\mathrm{f}^{n+1} - h[\mathbf{C}]\mathbf{V}_c\boldsymbol{u}_\mathrm{s}^{n+1} + h\mathbf{V}_\mathrm{f}\mathbf{G}\boldsymbol{p}^{n+1} = h\boldsymbol{f}_\mathrm{f} + \mathbf{M}_\mathrm{f}\boldsymbol{u}_\mathrm{f}^{n}. \tag{2}$$

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

$$\mathbf{M}_\mathrm{s}\boldsymbol{u}_\mathrm{s}^{n+1} + \mathbf{M}_\mathrm{f}\boldsymbol{u}_\mathrm{f}^{n+1} + h\,(\mathbf{V}_\mathrm{s} + \mathbf{V}_\mathrm{f})\,\mathbf{G}\boldsymbol{p}^{n+1} = h\boldsymbol{f}_\mathrm{s} + h\boldsymbol{f}_\mathrm{f} + \mathbf{M}_\mathrm{s}\boldsymbol{u}_\mathrm{s}^{n} + \mathbf{M}_\mathrm{f}\boldsymbol{u}_\mathrm{f}^{n}. \tag{3}$$

Substituting $\boldsymbol{u}_\mathrm{s}^{n+1}$ in equation (2) with equation (3), and combining the terms yields

$$\begin{aligned}&\left((\mathbf{I} + h[\mathbf{C}]\mathbf{V}_c\mathbf{M}_\mathrm{s}^{-1})\mathbf{M}_\mathrm{f} + h[\mathbf{C}]\mathbf{V}_c\right)\boldsymbol{u}_\mathrm{f}^{n+1}\\ &\quad + h\left(h[\mathbf{C}]\mathbf{V}_c\mathbf{M}_\mathrm{s}^{-1}(\mathbf{V}_\mathrm{s} + \mathbf{V}_\mathrm{f}) + \mathbf{V}_\mathrm{f}\right)\mathbf{G}\boldsymbol{p}^{n+1}\\ &= (\mathbf{I} + h[\mathbf{C}]\mathbf{V}_c\mathbf{M}_\mathrm{s}^{-1})(\mathbf{M}_\mathrm{f}\boldsymbol{u}_\mathrm{f}^{n} + h\boldsymbol{f}_\mathrm{f}) + h[\mathbf{C}]\mathbf{V}_c(\boldsymbol{u}_\mathrm{s}^{n} + h\mathbf{M}_\mathrm{s}^{-1}\boldsymbol{f}_\mathrm{s}).\end{aligned} \tag{4}$$

From this point on $\boldsymbol{u}_\mathrm{f}^{n+1}$ has been decoupled from $\boldsymbol{u}_\mathrm{s}^{n+1}$. Multiplying both sides of equation (4) with $(\mathbf{M}_\mathrm{s} + h[\mathbf{C}]\mathbf{V}_c)^{-1}\mathbf{M}_\mathrm{s}$ yields

$$\begin{aligned}(\mathbf{M}_\mathrm{f} + \mathbf{Q}\mathbf{M}_\mathrm{s})\boldsymbol{u}_\mathrm{f}^{n+1} &+ h(\mathbf{V}_\mathrm{f} + \mathbf{V}_\mathrm{s}\mathbf{Q})\mathbf{G}\boldsymbol{p}^{n+1}\\ &= \mathbf{M}_\mathrm{f}\boldsymbol{u}_\mathrm{f}^{n} + h\boldsymbol{f}_\mathrm{f} + \mathbf{Q}(\mathbf{M}_\mathrm{s}\boldsymbol{u}_\mathrm{s}^{n} + h\boldsymbol{f}_\mathrm{s}),\end{aligned} \tag{5}$$

where

$$\mathbf{Q} = (\mathbf{M}_\mathrm{s} + h[\mathbf{C}]\mathbf{V}_c)^{-1}h[\mathbf{C}]\mathbf{V}_c. \tag{6}$$

Similarly, substituting $\boldsymbol{u}_\mathrm{f}^{n+1}$ in equation (1) with equation (3), and multiplying both sides with $(\mathbf{M}_\mathrm{f} + h[\mathbf{C}]\mathbf{V}_c)^{-1}\mathbf{M}_\mathrm{f}$ yields

$$\begin{aligned}(\mathbf{M}_\mathrm{s} + \mathbf{P}\mathbf{M}_\mathrm{f})\boldsymbol{u}_\mathrm{s}^{n+1} &+ h(\mathbf{V}_\mathrm{s} + \mathbf{V}_\mathrm{f}\mathbf{P})\mathbf{G}\boldsymbol{p}^{n+1}\\ &= \mathbf{M}_\mathrm{s}\boldsymbol{u}_\mathrm{s}^{n} + h\boldsymbol{f}_s + \mathbf{P}(\mathbf{M}_\mathrm{f}\boldsymbol{u}_\mathrm{f}^{n} + h\boldsymbol{f}_\mathrm{f}),\end{aligned} \tag{7}$$

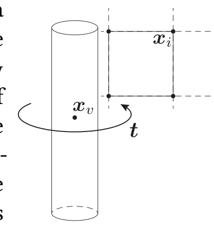
where

$$\mathbf{P} = (\mathbf{M}_\mathrm{f} + h[\mathbf{C}]\mathbf{V}_c)^{-1}h[\mathbf{C}]\mathbf{V}_c. \tag{8}$$

Combining equations (5) and (7) with equation (21), and introducing notations $\mathbf{D}_\mathrm{s} = \mathbf{M}_s + \mathbf{P}\mathbf{M}_f$ and $\mathbf{D}_\mathrm{f} = \mathbf{M}_\mathrm{f} + \mathbf{Q}\mathbf{M}_\mathrm{s}$, we have the form given by equation (24).

2 DISTRIBUTION OF TORQUE FROM VERTEX TO GRID

For simplicity of presentation, consider a yarn segment along the Y-direction (see the adjacent figure). If there is a torque $\boldsymbol{t} = t_\mathbf{Y}\boldsymbol{d}_\mathbf{Y}$ applied with respect to the centerline of the yarn to twist the yarn (where $t_\mathbf{Y}$ is the strength of the torque and $\boldsymbol{d}_\mathbf{Y}$ is the tangential direction of the yarn), then, when we distribute the torque $\boldsymbol{t}$ to a grid node i, this node receives a twisting force produced by the torque weighted by the (scalar) kernel function,



$$I_v^{-1}w_{v,i}[\boldsymbol{t} \times (\boldsymbol{x}_i - \boldsymbol{x}_v)] = [\boldsymbol{t} \times w_{v,i}I_v^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_v)], \tag{9}$$

where $w_{v,i}$ is the kernel function for the contribution of vertex v at the grid node i, and I_v is analogous to an inertia tensor defined as

$$I_v = \sum_i w_{v,i}(\boldsymbol{x}_i - \boldsymbol{x}_v)^*(\boldsymbol{x}_i - \boldsymbol{x}_v)^{*T}$$

where $\boldsymbol{x}^*$ is the cross-product matrix associated with vector $\boldsymbol{x}$. When $w_{v,i}$ is a trilinear function, then the relationship $w_{v,i}I_v^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_v) = \nabla w_{v,i}$ holds, as noted in [Jiang et al. 2015]. Then, the right-hand side of (9) can be simplified into

$$\boldsymbol{t} \times \nabla w_{v,i}.$$

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

3 A SIMPLE COHESION MODEL

We introduce a simple model to approximate the cohesion force. We assume that the surface tension appears when the distance between two wet cloth is less than a small constant a. Also the water-air surface is assumed perpendicular to the cloth surface. Under these

Authors' addresses: Yun (Raymond) Fei, Columbia University, Computer Science, New York, NY, 10027, USA; Christopher Batty, University of Waterloo, Computer Science, Waterloo, ON, N2L 3G1, Canada; Eitan Grinspun; Changxi Zheng, Columbia University, Computer Science, New York, NY, 10027, USA.

0730-0301/2018/8-ART51 $15.00
https://doi.org/10.1145/3197517.3201392

simplifications, the surface tension energy becomes $E_T = \int_\Omega \gamma d \mathrm{d}l$, 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

$$\mathrm{d}f_T = \gamma \mathrm{d}l. \tag{10}$$

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. [2007].
(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 $\mathrm{d}l$ in (10) using the effective length of the average triangle area. The force is then distributed evenly to the triangle element vertices.

4 SURFACE RECONSTRUCTION AND RENDERING

We performed surface reconstruction in SideFX's Houdini [2018], which uses OpenVDB [Museth 2013]. 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 a dilate-smooth-erode operator [Museth 2014] to create the smooth transition between them. Besides affecting the bulk and surface liquid geometry, wet fabrics are usually darker and more specular [Jensen et al. 1999]. 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 FABRIC PARAMETERS

In table 1 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 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 paper are presented in table 2.

Table 1. **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 2.

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

Table 2. **Conversion between fabric parameters.** From any given pair of two paramters, the other two can be computed. We take $s = 2.54$cm/in since the fabric thread count habitually taken in per square inch needs to be converted to the centimeter–gram–second (CGS) system used throughout this paper.

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)	-	-	$\frac{8r_c s}{\pi(4r_b^2+d^2)}$	$\frac{d^2}{d^2+4r_b^2}$

REFERENCES

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A Fast Variational Framework for Accurate Solid-fluid Coupling. *ACM Transactions on Graphics (TOG)* 26, 3, Article 100 (July 2007).

Henrik Wann Jensen, Justin Legakis, and Julie Dorsey. 1999. Rendering of Wet Materials. In *Proceedings of the 10th Eurographics Conference on Rendering (EGWR'99)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 273–282.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 51.

Ken Museth. 2013. VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics (TOG)* 32, 3 (2013), 27.

Ken Museth. 2014. A flexible image processing approach to the surfacing of particle-based fluid animation (invited talk). In *Mathematical progress in expressive image synthesis I*. Springer, 81–84.

SideFX. 2018. SideFX Houdini. https://www.sidefx.com. (2018).