

Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact

CHENFANFU JIANG, University of Pennsylvania and Jixie Effects

THEODORE GAST, University of California, Los Angeles and Jixie Effects

JOSEPH TERAN, University of California, Los Angeles and Jixie Effects

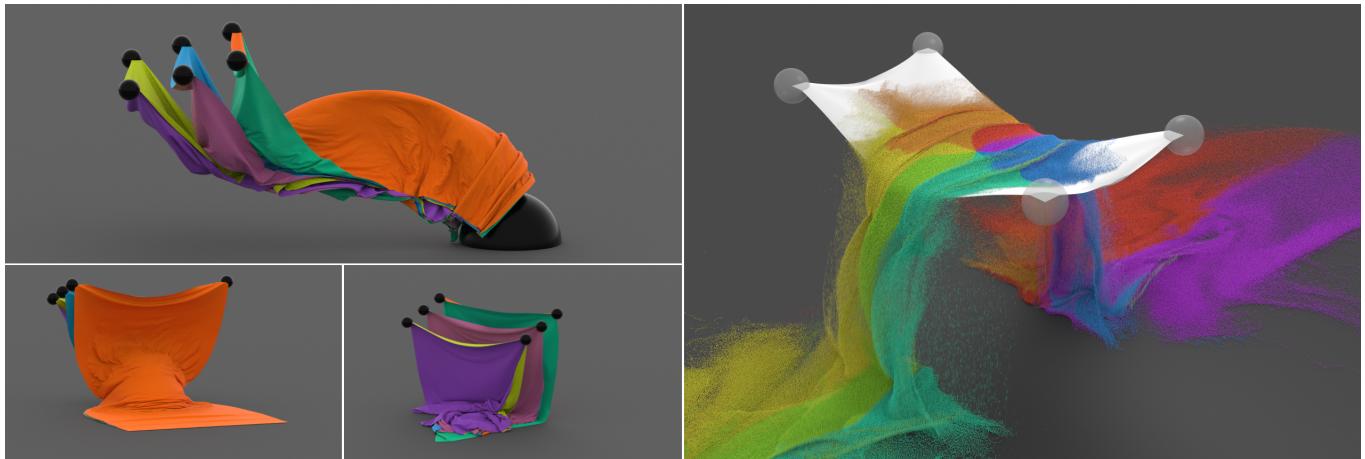


Fig. 1. The left figure shows three pieces of cloth with around 1.4M triangles pushed back and forth by a sphere, revealing intricate folds and contact. The right figure shows 7M colored sand grains coupled with elastic cloth, exhibiting beautiful flow patterns.

The typical elastic surface or curve simulation method takes a Lagrangian approach and consists of three components: time integration, collision detection and collision response. The Lagrangian view is beneficial because it naturally allows for tracking of the codimensional manifold, however collision must then be detected and resolved separately. Eulerian methods are promising alternatives because collision processing is automatic and while this is effective for volumetric objects, advection of a codimensional manifold is too inaccurate in practice. We propose a novel hybrid Lagrangian/Eulerian approach that preserves the best aspects of both views. Similar to the Drucker-Prager and Mohr-Coulomb models for granular materials, we define our collision response with a novel elastoplastic constitutive model. To achieve this, we design an anisotropic hyperelastic constitutive model that separately characterizes the response to manifold strain as well as shearing and compression in the directions orthogonal to the manifold. We discretize the model with the Material Point Method and a novel codimensional Lagrangian/Eulerian update of the deformation gradient. Collision intensive scenarios with millions of degrees of freedom require only a few minutes per frame and examples with up to one million degrees of freedom run in less than thirty seconds per frame.

CCS Concepts: • Computing methodologies → Physical simulation;

Additional Key Words and Phrases: MPM, elastoplasticity, friction, cloth, knit, hair

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2017/7-ART152 \$15.00

DOI: <http://dx.doi.org/10.1145/3072959.3073623>

ACM Reference format:

Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact. *ACM Trans. Graph.* 36, 4, Article 152 (July 2017), 14 pages.

DOI: <http://dx.doi.org/10.1145/3072959.3073623>

1 INTRODUCTION

Physically based animation of elastic surfaces and curves has been an essential aspect of computer graphics for nearly three decades. Whether it be different layers of clothing in a virtual garment, individual strands in a head of hair or even yarns in a knit garment, collision and contact phenomena of these materials are essential for the richness and realism provided by physics based simulation. Unfortunately the thin nature of these materials makes collision detection and resolution challenging. This process is often the bottleneck in modern visual effects. Building on recent methods that characterize frictional contact in granular materials via elastoplasticity [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Narain et al. 2010], we present a new approach for codimensional elasticity that uses a hybrid Lagrangian/Eulerian Material Point Method (MPM) [Sulsky et al. 1994] discretization to model frictional contact with a continuum view. Unlike traditional approaches, our elastoplastic description completely characterizes all collision/contact response in the continuum and requires no separate post-processing.

Codimensional elastic objects are naturally represented with a Lagrangian mesh. Such approaches were pioneered in graphics by Terzopoulos et al. [1988; 1987] and are still primarily used today. With the Lagrangian mesh model, individual particles are tracked and mesh polygons or segments are used to approximate the spatial

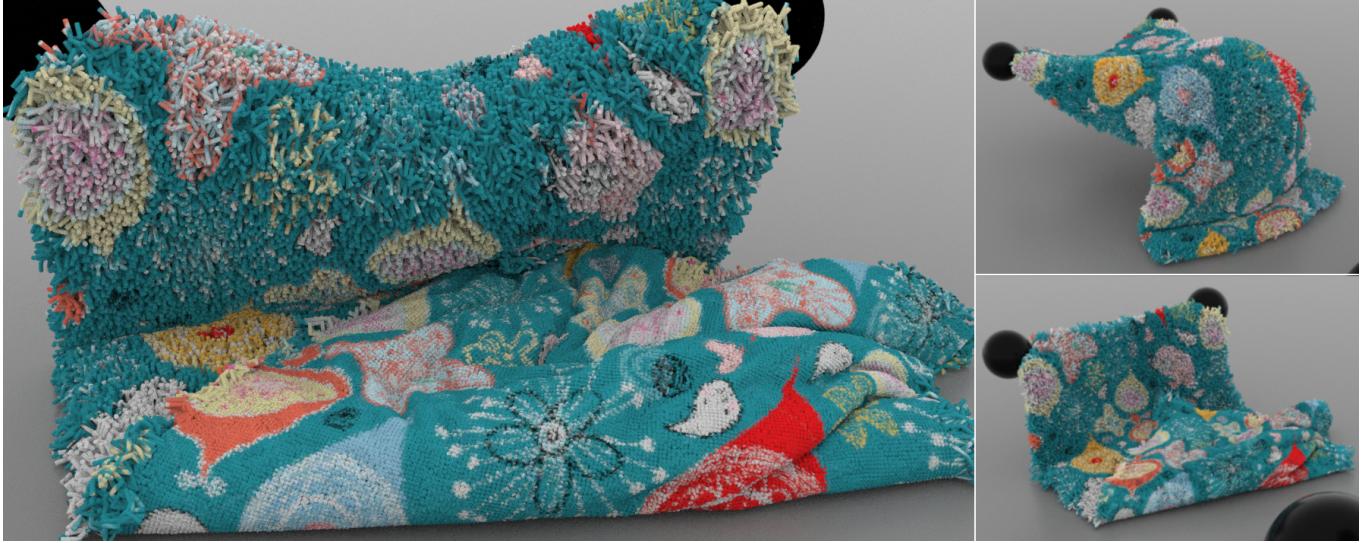


Fig. 2. A shag carpet with 1M particles is pushed by a sphere and folds back, showing detailed folds.

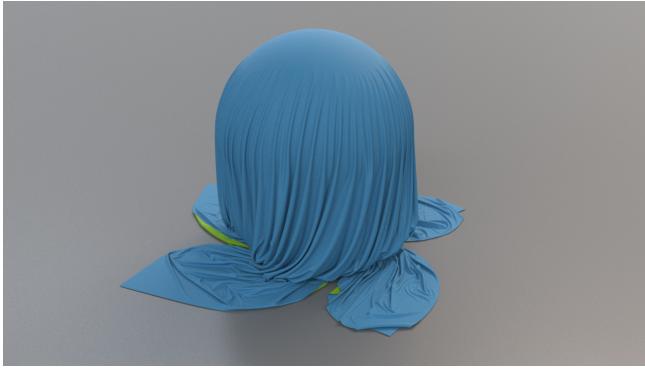


Fig. 3. A piece of cloth with 2M triangles falling onto a rotating sphere. Low friction on the ground and high friction on the sphere causes detailed wrinkles and folds.

derivatives of the material deformation mapping (often referred to as the deformation gradient). The mechanics of elasticity are naturally discretized with this Lagrangian view. However additional modeling is required to include the effects of self and external contact since these phenomena may cause interactions between distant regions in the mesh. For example, mesh facets like points and triangles or segment pairs must not pass through one another over a time step. These constraints must be satisfied by some means external to the elasticity modeling. This can be done in number of ways including repulsion penalties [Baraff and Witkin 1998; Choi and Ko 2002; Cirio et al. 2014; Harmon et al. 2009; Kaldor et al. 2008, 2010], impulsive change in momenta [Bridson et al. 2002; Harmon et al. 2008; McAdams et al. 2009; Sifakis et al. 2008; Tang et al. 2016], and linear complimentary formulations of the constrained dynamics

[Bertails-Descoubes et al. 2011; Daviet et al. 2011; Otaduy et al. 2009].

The Eulerian view is a useful alternative to the Lagrangian view. In the Eulerian view, discrete samples of the solution are computed on a stationary background grid as material advects through the domain. While Lagrangian methods require two different components of the algorithm: elasticity modeling and collision/contact resolution, Eulerian methods typically do not. The constitutive behavior of contact is expressed as that of the material itself. For example, for free-surface incompressible flow, no self collision model is imposed beyond the velocity divergence condition needed to enforce incompressibility [Bridson 2008]. Recently, various researchers have utilized aspects of Eulerian methods to design collision and contact treatment for elastic materials [Fan et al. 2013, 2014; Levin et al. 2011; Li et al. 2013; Teng et al. 2016]. For elastic objects accurate treatment of advection is needed to preserve an accurate rest state of the material, e.g. the impressive treatment for volumetric objects in Levin et al. [2011]. However, this is difficult for codimensional elasticity.

Hybrid Lagrangian/Eulerian approaches seek to combine the benefits of both views. For example Particle-In-Cell (PIC) [Harlow 1964] approaches like FLIP [Brackbill and Ruppel 1986] and MPM combine a Lagrangian form of advection with a regular grid momentum update. Various researchers have developed approaches of this type to simplify contact with Lagrangian methods or to improve the advection accuracy with Eulerian methods. For example, McAdams et al. [2009] use FLIP and Eulerian incompressibility to efficiently model self collision for hair. Petrovic et al. [2005] use a hybrid approach where Lagrangian hair velocities are smoothed on an Eulerian grid. Faure et al. [2007] also use a hybrid approach defining interaction penalties and constraints for Lagrangian objects after mapping mass,

momentum and density to an Eulerian grid. MPM has some similarities to this and allows for the simulation of a wide range of elastoplastic materials in contact without a separate model for collision [Stomakhin et al. 2013; Yue et al. 2015]. Jiang et al. [2015] simulate volumetric hyperelastic objects in contact with MPM. Recently, various authors have shown that frictional contact for sand can be modeled with a hybrid discretization of an appropriate plastic flow [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Narain et al. 2010; Zhu and Bridson 2005].

Our approach is a hybrid Lagrangian/Eulerian MPM method. We build on the recent approaches for contact and collision via elasto-plasticity of granular materials in [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Narain et al. 2010; Zhu and Bridson 2005] and via hybrid Lagrangian/MPM approach for volumetric objects in Jiang et al. [2015]. Standard MPM updates the deformation gradient on each particle independently with an Eulerian view. However, this leads to numerical plasticity and failure. While these phenomena are useful when simulating elastoplasticity with failure, they are difficult to prevent when simulating hyperelastic objects. To prevent this, we track the deformation of codimensional elastic objects in a Lagrangian way, as in Jiang et al. [2015]. However, while the approach in Jiang et al. [2015] allows for self-collision with volumetric

objects, it does not work for codimensional objects. This occurs because for non-volumetric objects, the Lagrangian update of deformation only tracks components in the manifold, it does not capture deformation in the orthogonal directions. It is the elastic response to deformation in these directions that allows for frictional contact with volumetric objects in MPM simulations. We remedy this by updating the orthogonal components of the deformation gradient in the standard MPM way. Furthermore, while relying on the elastic response alone was sufficient for self-collision simulation with volumetric objects in Jiang et al. [2015], it can lead to artificial cohesion (stickiness) and excessive friction for codimensional elasticity. We fix this with a novel plastic flow that enforces a Coulomb friction inequality between shear and normal stresses in the directions orthogonal to the surface or curve. The approach is similar to the approach for sand in Klar et al. [2016], but with a modification that prevents plasticity in the codimensional manifold, which should be purely elastic. We define the elasticity in an anisotropic way so that the response in the surface or curve cleanly relates to the response in the orthogonal directions.

Our approach is computationally efficient, largely due to the simplicity of handling contact and collision through the constitutive modeling alone. Furthermore, it naturally allows for coupling with multiple materials. We demonstrate this by simulating a range of coupled elastic surfaces and curves, fluids and granular materials with millions of degrees of freedom in a few minutes per frame. We investigated both implicit and explicit versions of the grid based momentum update in MPM. For explicit time stepping, we developed a novel damping model that does not penalize rigid motions and that unlike standard approaches like Rayleigh damping, does not impact the time step restriction. To summarize, we list our contributions below.

- Elasticity models that cleanly relate the elastic response in the surface or curve to the response in the orthogonal directions.
- Anisotropic plasticity models that characterize frictional contact with elastic surfaces and curves.
- A return mapping algorithm to temporally discretize the plastic flow.
- A hybrid Lagrangian/Eulerian MPM discretization of the deformation gradient.
- An explicit damping model that does not penalize rigid modes.

2 PREVIOUS WORK

2.1 Cloth

Clothing simulation has a long history in computer graphics. Thorough summaries can be found in Choi and Ko [2005], Nealen et al. [2006] and Thomaszewski et al. [2007]. Baraff and Witkin [1998] use dynamically defined stiff springs that prevent cloth/cloth penetration and apply an implicit integration scheme for efficiency. Choi and Ko [2002] develop a semi-implicit treatment that efficiently handles buckling instabilities. Bridson et al. [2002] process collisions for all point/triangle and edge/edge pairs in the mesh via impulsive response. Harmon et al. [2008] provide an improved alternative to the failsafe rigid impact zones of Provot [1997] and Bridson et al.

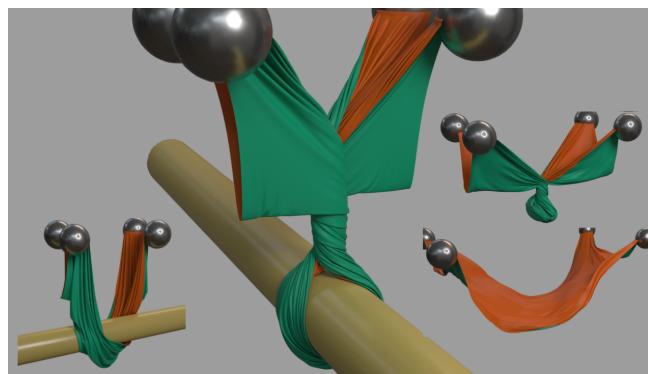


Fig. 4. A piece of cloth with 1.14M triangles is twisted by a cylinder. The simulation runs at 1.3 minutes per frame on average.

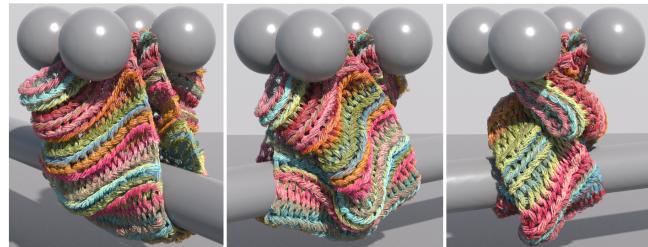


Fig. 5. Our method handles detailed contacts between individual yarn threads under tension as well as macroscopic collision between yarns as demonstrated by this knitted cloth being twisted. Complex per-yarn motion is well resolved through elastoplasticity.



Fig. 6. Our method is able to capture many intricate features of tearing apart a fibrous material consisting of 2.15M particles.

[2002]. In [2009] Harmon et al. combine asynchronous variational integration with kinetic data structures. Collision resolution is done with barrier potentials that they efficiently approximate with nested families of quadratic potentials. In Ainsley et al. [2012], they further accelerate this approach by two orders of magnitude. Otaduy et al. [2009] use an implicit treatment of complimentary collision constraints and a new solver for large mixed linear complementarity problems. Recent work has shown that implicit time stepping for cloth can be performed more efficiently via optimization with auxiliary variables representing strain [Bouaziz et al. 2014; Liu et al. 2013].

2.2 Yarn and knits

In the seminal work of Kaldor et al. [2008] it was shown that elastic curves representing individual yarns can be used with stunning effect to create knit garments. In Kaldor et al. [2010] they improve performance with infrequently updated locally co-rotated linear approximations of the barrier potential to provide a five fold speed up. Cirio et al. [2014] build on the work of Sueda et al. [2011] to resolve warp/weft intersections with a point that connects the two curves. Breen et al. [1994] also demonstrated yarn-level simulation.

2.3 Hair

A good survey of hair simulation in computer graphics is provided in Ward et al. [2007]. Many researchers have modeled hair with various elasticity models per strand [Bando et al. 2003; Choe et al. 2005; Kaufman et al. 2014; McAdams et al. 2009; Selle et al. 2008]. Others model hair as inextensible [Anjyo et al. 1992; Chang et al. 2002; Choe et al. 2005; Müller et al. 2012]. Bertails et al. use Kirchhoff elastic rod models, which naturally enforce inextensibility [2009; 2006]. Much recent work has shown the importance of including frictional effects [Bertails-Descoubes et al. 2011; Daviet et al. 2011; Derouet-Jourdan et al. 2013; Kaufman et al. 2014]. Kaufman et al. [2014] note that typical impulse based collision models for elastic rods can lead to instabilities via excitation of stretching modes and

that an improved, adaptively nonlinear model for collision leads to more robust and accurate simulations.

2.4 Parallel computing

Selle et al. [2009] developed a distributed memory, parallel version of Bridson et al. [2002] capable of simulating cloth meshes with millions of triangles. Tang et al. [2013] were the first to implement all components of a cloth solver on the GPU and achieve speed-ups of 10-14X over Selle et al. [2009]. They improve on this further in Tang et al. [2016] with an implicit GPU implementation that adopts the inelastic impact zone response from Harmon et al. [2008]. Other CPU/GPU level parallelism is explored in [Li et al. 2011; Schmitt et al. 2013]. Ni et al. [2015] accelerate the asynchronous contact mechanics approaches [Ainsley et al. 2012; Harmon et al. 2009] by up to 12 times with a 384 core Cray XC30.

2.5 Reduced and adaptive models

Recently, much progress has been made in reduced models that can simulate clothing at interactive rates in many scenarios [de Aguiar et al. 2010; Feng et al. 2010; Kavan et al. 2011; Kim et al. 2013; Wang et al. 2010; Zurdo et al. 2013]. For hair, Chai et al. [2014] use a reduced model driven by high-resolution simulation data to incredibly simulate up to 150 thousand strands in real-time. Adaptivity has also been shown to produce stunning detail at modest cost in [Koh et al. 2015; Lee et al. 2010; Narain et al. 2012].

2.6 Eulerian and continuum collision/contact

Goktekin et al. [2004] use an Eulerian approach to simulate viscoelastic materials without need for explicit contact modeling. Levin et al. [2011] advect the reference configuration in an Eulerian approach to volumetric elastic object simulation that drastically reduces the complexity of collision modeling. Fan et al. [2013; 2014] further improve this by reducing memory and computational cost. Teng et al. [2016] show that a similar approach handles hyperelastic solids coupled with incompressible fluids. Li et al. [2013] use an Eulerian view of hyperelastic surfaces to model skin/tissue contact. Continuum friction models have proven useful for collision models of crowd interactions [Golas et al. 2014; Narain et al. 2009]. The hybrid Lagrange/Eulerian PIC/MPM has been used for sand animation in [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Narain et al. 2010; Zhu and Bridson 2005]. Also, MPM has been used for various elastoplastic materials [Jiang et al. 2015; Stomakhin et al. 2013; Yue et al. 2015]. Müller et al. [2015] mesh the space surrounding elastic objects and enforce an incompressibility constraint on the mesh. Wu and Yuksel [2016] use hair volume meshes to simulate hair in real time. Others have experimented with meshes with dynamic connectivity for collision processing [Chang et al. 2002; Kim and Neumann 2000; Sifakis et al. 2008]. Macklin et al. [2014] model both constitutive response and collision via constraints. Hadap et al. [2001], Bando et al. [2003] and McAdams et al. [2009] use continuum fluid concepts to accelerate collision processing for hair.

3 MATHEMATICAL BACKGROUND

We represent codimensional objects as volumetric elastoplastic continua. While our objects are thin surfaces or even curves in 3D, it

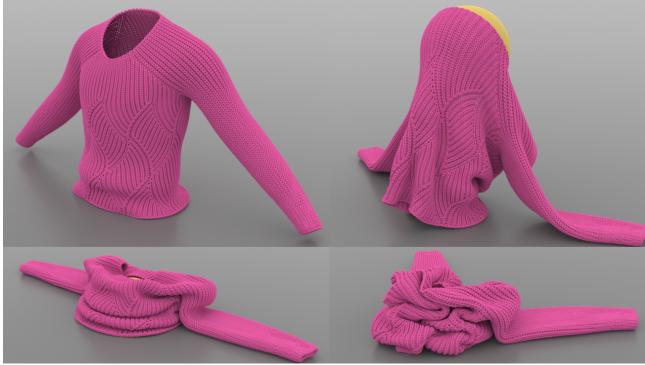


Fig. 7. A sweater dropped onto a rising sphere, stretched and then folded on the ground. The simulation takes 13 seconds per frame.

is useful to conceive of their dynamics as if they have appreciable thickness in a continuum, e.g. as is done with granular materials in [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Narain et al. 2010]. Given the similarities in our assumptions, we adopt the notation of Klár et al. [2016], where the state can be described at each location by its density $\rho(\mathbf{x}, t)$ and velocity $\mathbf{v}(\mathbf{x}, t)$. The governing equations come from conservation of mass and momentum (see [Bonet and Wood 2008; Gonzalez and Stuart 2008] for derivations)

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0, \quad \rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}. \quad (1)$$

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

3.1 Deformation gradient

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

Codimensional deformation is expressed via components of \mathbf{F} . We define material directions $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ at point \mathbf{X} . In the case of elastic surfaces \mathbf{D}_1 and \mathbf{D}_2 are tangent to the initial configuration of the surface and \mathbf{D}_3 is normal to the surface. In the case of curves, \mathbf{D}_1 is tangent to the curve and \mathbf{D}_2 and \mathbf{D}_3 are orthogonal to \mathbf{D}_1 . We define $\mathbf{d}_i = \mathbf{F}\mathbf{D}_i$, $i = 1, 2, 3$. With this convention, the \mathbf{d}_i represent the local direction and stretching of material in the \mathbf{D}_i direction under ϕ . Thus for curves, all deformation in the manifold is expressed via \mathbf{d}_1 and for surfaces, via \mathbf{d}_1 and \mathbf{d}_2 . The remaining \mathbf{d}_i represent the deformation of material normal to the manifold. Thus we can both account for deformation in the manifold, deformation normal to the manifold and shearing of material relative to the manifold in terms of the \mathbf{d}_i and their relation to one another. For example for surfaces, while \mathbf{D}_3 is normal to the initial surface, \mathbf{d}_3 will not be when there

is a shearing motion relative to the manifold. We visualize the case of a curve in 2D in Figure 8.

3.2 Codimensional anisotropic plasticity

Large strain elastoplasticity is modeled by factoring the deformation gradient into elastic and plastic parts as $\mathbf{F} = \mathbf{F}^E \mathbf{F}^P$. The plastic part \mathbf{F}^P represents deformation history that has been lost and will no longer be penalized elastically. The elastic part \mathbf{F}^E remains and is penalized elastically. We use a hyperelastic potential energy density that increases with increasing deformation in \mathbf{F}^E .

As in Klár et al. [2016], we model frictional contact with elastoplasticity. However, unlike for granular materials where collision can occur in any direction local to a grain, we only consider contact to occur in the direction orthogonal to the elastic surface or curve. Therefore, the plasticity should only modify the codimensional components of the deformation. The mechanical response to deformations in the manifold is purely elastic. This places an anisotropic constraint on the multiplicative decomposition. To guarantee that deformation in a curve is purely elastic, our plasticity model must satisfy $\mathbf{F}\mathbf{D}_1 = \mathbf{F}^E\mathbf{D}_1$ (equivalently $\mathbf{D}_1 = \mathbf{F}^P\mathbf{D}_1$). To guarantee that the deformation in a surface is purely elastic, we additionally require $\mathbf{F}\mathbf{D}_2 = \mathbf{F}^E\mathbf{D}_2$ (equivalently $\mathbf{D}_2 = \mathbf{F}^P\mathbf{D}_2$). By satisfying these constraints, we guarantee that all non-rigid deformation of the surface or curve is penalized elastically. Plasticity is only allowed to effect the components of the deformation that

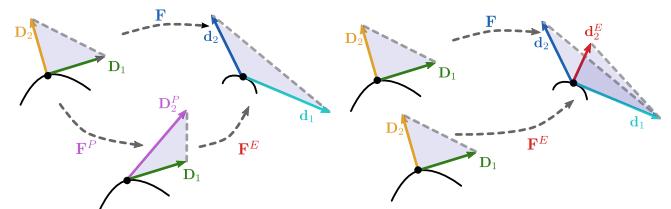


Fig. 8. Left: \mathbf{F} is decomposed into \mathbf{F}^P , the forgotten sliding and separation, and \mathbf{F}^E the remembered stretching, collision and shearing. Right: \mathbf{F}^E deforms \mathbf{D}_1 and \mathbf{D}_2 into \mathbf{d}_1 and \mathbf{d}_2^E .

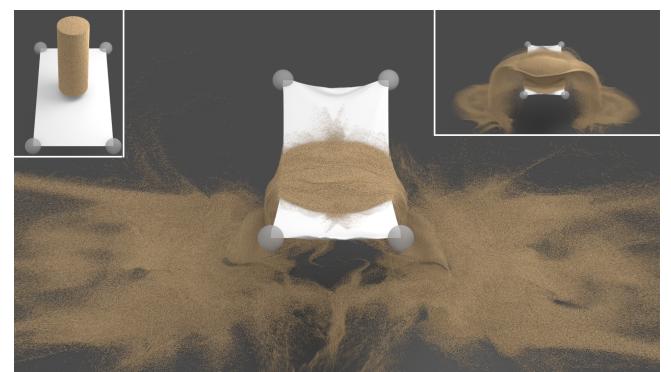


Fig. 9. Here we show two way coupling between 7M sand grains and a piece of elastic cloth.

provide compression, extension and shearing in directions normal to the manifold.

3.3 Codimensional anisotropic elasticity

As in [Klár et al. 2016; Stomakhin et al. 2013] we use a model that is hyperelastic in F^E , where the elastic potential energy density increases with deformation of the elastic part of the deformation gradient. The Cauchy stress in the material is $\sigma = \frac{1}{\det(F)} \frac{\partial \psi}{\partial F^E} F^{ET}$. Here $\psi(F^E)$ is the elastic energy density for penalizing non-rigid F^E . In our case, it serves two purposes. First, it expresses the resistance to deformations that change the shape of the surface or curve manifold. However, as in the Klar et al. [2016] approach to granular materials, it also defines the resistance to contact and frictional sliding. In the case of codimensional elasticity, this resistance is only to deformations normal to the manifold and we design an anisotropic energy density that takes this dual nature into account.

We define our potential with respect to the local material directions $D = [D_1, D_2, D_3]$ as $\psi(F^E, D)$. The elastic potential must be invariant under world space rotations, and so we are free to choose a convenient basis. One natural choice is given by orthogonalizing the vectors $d_i^E = F^E D_i$, the elastically deformed material directions, with the Gram-Schmidt process to obtain $Q = [q_1, q_2, q_3]$. Equivalently, $QR = F^E D$ is the QR-decomposition of $F^E D$. We then define $\psi(F^E, D) = \hat{\psi}(R)$. This is similar to the isotropic case where the additional material frame invariance allows one to choose u_i and v_i such that $F^E v_i = \sigma_i u_i$, and define $\psi(F^E) = \hat{\psi}(\Sigma)$.

Another common stress measure is the first Piola-Kirchhoff stress, defined as $P = \frac{\partial \psi}{\partial F^E} (F^P)^{-T}$, and related to σ with $\sigma = \frac{1}{J} P F^T$. If we further denote $F^E D$ with d^E , we have $\frac{\partial \psi}{\partial F^E} = \frac{\partial \psi}{\partial d^E} D^T$. We show in [Jiang et al. 2017a] that $\frac{\partial \psi}{\partial d^E} = Q (\mathcal{T}(K) + \mathcal{T}(K)^T - \mathcal{D}(K)) R^{-T}$, where $K = \frac{\partial \hat{\psi}}{\partial R} R^T$. Here \mathcal{T} and \mathcal{D} are operators on matrices that keep upper triangular part and diagonal part respectively.

3.3.1 Curves: We model elastic curves with rotational material symmetry around the fiber direction D_1 . In other words for our curves, the material directions D_2 and D_3 are regarded as essentially arbitrary and any choices which are mutually orthogonal with D_1 should give the same potential. Guided by this material symmetry, we decompose $R = R_3 R_2 R_1$ into a composition of three deformations, first (R_1) a stretching of the fibers, second (R_2) a shearing of the fibers and finally (R_3) a deformation of the cross section of the fibers

$$R_1 = \begin{bmatrix} r_{11} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_2 = \begin{bmatrix} 1 & r_{12} & r_{13} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix}$$

Based on this decomposition, we write our elastic potential for curves as a sum of 3 terms $\hat{\psi}(R) = f(R_1) + g(R_2) + h(R_3)$. The first term $f(R_1) = \frac{k}{2}(r_{11} - 1)^2$ penalizes change of fiber length, for the second term $g(R_2) = \frac{\gamma}{2}(r_{12}^2 + r_{13}^2)$ penalizes shearing along the fibers. In absence of other deformations, we would like a cross section of the fibers to behave like material in frictional contact, resisting compression and shearing. Thus for the third term $h(R_3)$, we use the two dimensional version of the elastic potential from Klar et al. [2016], since it naturally enforces such frictional contact (when combined with appropriate plasticity law). That is $h(R_3) = \mu(\epsilon_1^2 + \epsilon_2^2) + \frac{\lambda}{2}(\epsilon_1 + \epsilon_2)^2$, where ϵ_1 and ϵ_2 are the logarithms of the nontrivial singular values of R_3 .

$$\mu(\epsilon_1^2 + \epsilon_2^2) + \frac{\lambda}{2}(\epsilon_1 + \epsilon_2)^2, \text{ where } \epsilon_1 \text{ and } \epsilon_2 \text{ are the logarithms of the nontrivial singular values of } R_3.$$

3.3.2 Surfaces: For the surface energy, we think of the surface itself as being isotropic, i.e. the potential is invariant under rotation of the material directions D_1 and D_2 . Again we decompose $R = R_3 R_2 R_1$ into a composition of three deformations, first (R_1) an in-plane deformation of the surface, second (R_2) a shearing of surface normal and finally (R_3) a compression or stretching of the surface normal

$$R_1 = \begin{bmatrix} r_{11} & r_{12} & 0 \\ 0 & r_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_2 = \begin{bmatrix} 1 & 0 & r_{13} \\ 0 & 1 & r_{23} \\ 0 & 0 & 1 \end{bmatrix}, R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r_{33} \end{bmatrix}$$

Then we write the energy as a sum of three terms, $\hat{\psi}(R) = f(R_3) + g(R_2) + h(R_1)$. For the first term $f(R_3)$, we only penalize compression of the material in the normal direction, as we think of the surface as being mostly surrounded by empty space which can expand freely

$$f(R_3) = \begin{cases} \frac{k}{3}(1 - r_{33})^3 & r_{33} \leq 1 \\ 0 & r_{33} > 1 \end{cases}.$$

For the second term we use $g(R_2) = \frac{\gamma}{2}(r_{13}^2 + r_{23}^2)$ to penalize shearing of the normal to the surface. For the third term $h(R_1)$ we use the two dimensional version of the fixed corotated potential from Stomakhin et al. [2012] to penalize deformation of the surface, although any in-plane energy can be used (e.g., the in-plane surface energy from [Baraff and Witkin 1998]) with no change except to the stress and stress derivatives.

3.4 Friction and plastic yield condition

With Coulomb friction, the frictional force f_f must be smaller than a constant c_F , the coefficient of friction, times the normal force f_n : $f_f \leq c_F f_n$. In the continuum view, at a given point x in world space and a normal vector n , the traction vector t gives the local force per area that the material on one side of the plane with normal n exerts on the other side [Gonzalez and Stuart 2008]. The traction has normal and shearing components $f_n = -n^T t$ and $f_s = s(n, \theta)^T t$ respectively where $s(n, \theta)$ is an arbitrary vector in the plane with normal n , and θ indicates its direction in the plane. Note that we use the convention that compressive force f_n is positive as in the Coulomb model. If the material is in frictional contact in direction n , then $f_f = f_s$ is the component of the frictional response in the direction $s(n, \theta)$ and for all angles θ , the inequality $s(n, \theta)^T t \leq c_F f_n$ must be satisfied.

Expressing the traction in terms of the Cauchy stress σ as $t = \sigma n$, shows that the Coulomb model places constraints on the stress: $s(n, \theta)^T \sigma n + c_F n^T \sigma n \leq 0$ for all θ given frictional contact direction n . This is the idea behind the frictional contact model with Drucker-Prager for granular materials [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Narain et al. 2010]. However, in the case of a grain, contact is assumed to happen in all directions. For cloth or curves we only consider the directions orthogonal to the manifold to be in frictional contact. Together with our choice of elastic potential, this results in a less restrictive constraint that does not effect elastic deformation in the manifold. For frictional contact between surfaces, we only consider contact in the direction normal to the surface: $n = q_3$. For curves, there is a two dimensional normal space spanned

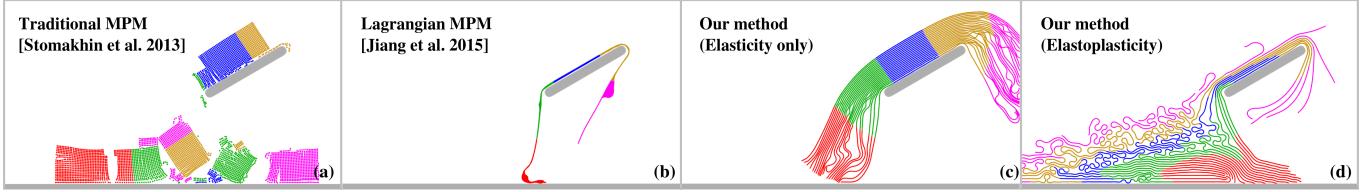


Fig. 10. We demonstrate the effect of different choices of deformation gradient discretization as well as the effect of plasticity in this fiber piling example. (a) shows standard MPM deformation gradient; (b) shows the codimensional Lagrangian deformation gradient from Jiang et al. [2015]; (c) shows our model without any plasticity; (d) is our full elastoplasticity model.

by \mathbf{q}_2 and \mathbf{q}_3 and thus we consider contact in all directions $\mathbf{n} = \cos(\hat{\theta})\mathbf{q}_2 + \sin(\hat{\theta})\mathbf{q}_3$.

In order to satisfy the stress constraints the material will yield. Physically this yielding manifests as the cloth or fibers sliding, and “forgetting” some of the shearing deformation. This “forgetting” occurs by some of the incremental deformation being stored in \mathbf{F}^P as opposed to \mathbf{F}^E , as the stress is a function of \mathbf{F}^E and not \mathbf{F}^P . We discuss these aspects in (§4.5).

4 DISCRETIZATION

Our MPM discretization is the key to translating from the theoretical continuum equations into an algorithm that resembles traditional approaches to codimensional elasticity with contact. PIC and therefore MPM are hybrid particle/grid methods. From the continuum point of view, the particles represent discrete samples of the continuous material and the grid is just a helper for computing their physical interactions. For example, with sand in [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Narain et al. 2010], the particles can be rendered as individual grains and the grid update, whose essential features are inherited from the continuum, can be seen as a means for processing their frictional contact interactions. However, for codimensional elasticity, we conceive of our continuum samples as coherent surface or curve meshes, rather than as unstructured particles. This aspect is the primary difference from traditional MPM approaches.

Because our MPM discretization has many similar features to most other MPM approaches in computer graphics [Daviet and Bertails-Descoubes 2016; Jiang et al. 2015; Klár et al. 2016; Stomakhin et al. 2013; Yue et al. 2015], we briefly cover aspects of the algorithm that are common to these approaches and address in more detail our novel modifications. These are primarily related to an improved version of the codimensional sampling in Jiang et al. [2015] that allows for self-collision (§4.3) and its ramifications in the grid momentum updates (§4.4), as well as the novel aspects of

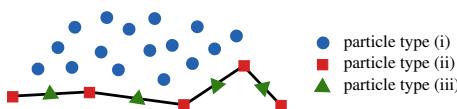


Fig. 11. Particle types. (i) Traditional MPM particles, (ii) Lagrangian mesh vertex particles, (iii) Lagrangian mesh element quadrature particles.

the associated elasticity and plasticity updates (§4.5). We summarize essential steps in the algorithm below.

- (1) **Transfer to grid:** Use APIC [Jiang et al. 2015] to transfer mass and momentum from particles to the grid. Divide grid momentum by grid mass to define grid velocity. (§4.2)
- (2) **Update grid momentum** Use either explicit symplectic Euler or backward Euler to update grid momentum. (§4.4)
- (3) **Transfer to particles:** Use APIC to transfer velocities and affine matrices from grid to particles [Jiang et al. 2015]. (§4.2)
- (4) **Update particles:** Update particles positions (§4.2) and deformation gradients (§4.3).
- (5) **Update plasticity:** Project the deformation gradient for plasticity, updating the elastic and plastic parts. (§4.5)

4.1 Lagrangian state

The Lagrangian particle state is the primary representation of material for MPM. We classify particles as either: (i) traditional MPM particles, (ii) Lagrangian mesh vertex particles or (iii) Lagrangian mesh element quadrature particles (Figure 11). Type (ii) particles are used to track the in-manifold deformation of the elements as in traditional FEM solvers. Type (iii) particles are used to track the codimensional deformation in an updated Lagrangian manner. The details of this deformation tracking are discussed in Section (§4.3). At time t^n , we store the particle positions \mathbf{x}_p^n , velocities \mathbf{v}_p^n , initial mass m_p , elastic deformation gradient $\mathbf{F}_p^{E,n}$, initial volume V_p^0 , affine velocity \mathbf{C}_p^n and undeformed material directions \mathbf{D}_p . \mathbf{D}_p is not stored for particles of type (i) and (ii). $\mathbf{F}_p^{E,n}$ is not stored for particles of type (ii). We set mass $m_p = \rho V_p^0$, and we set V_p^0 according to the total volume divided by the number of particles for type (i) particles. For particles of type (ii) and (iii), we divide the volume of each triangle/segment equally among the type (ii) and type (iii) particles incident to it and then add this to each particle’s V_p^0 .

In the following sections, we use the notation $\mathcal{I}^{(i)}$, $\mathcal{I}^{(ii)}$, $\mathcal{I}^{(iii)}$ to represent the sets of particle indices of types (i),(ii) and (iii) respectively. E.g. for any particle index p it is in exactly one of $\mathcal{I}^{(i)}$, $\mathcal{I}^{(ii)}$ or $\mathcal{I}^{(iii)}$, depending on its type.

4.2 Grid transfers

4.2.1 Particle to grid: We transfer particle mass and momentum to the grid using APIC [Jiang et al. 2015, 2017b]. As with standard PIC, we associate with each particle p and grid node i a weight

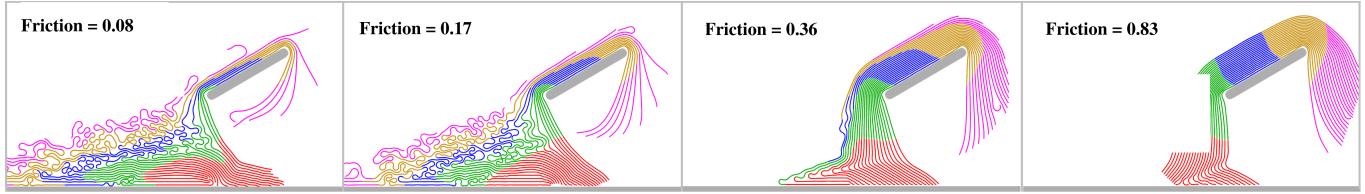


Fig. 12. We demonstrate the effect of increasing friction in this 2 dimensional piling example. Friction increases from left to right.

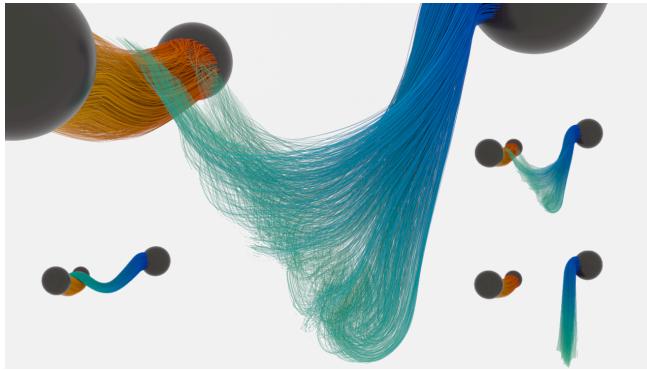


Fig. 13. A hair tube with 2655 strands is dropped onto another tube of hairs, showing detailed dynamics due to frictional contact between hairs. The simulation runs at 11 seconds per frame.

$w_{ip}^n = N(\mathbf{x}_p^n - \mathbf{x}_i)$ where \mathbf{x}_i is the location of the grid node and $N(\mathbf{x})$ are linear, quadratic or cubic b-spline kernels. These weights define the contribution of quantities on particle p to grid node i . The mass contribution from particle p to grid node i is $w_{ip}^n m_p$. The total mass on grid node i is the sum of the contributions from all particles $m_i^n = \sum_p w_{ip}^n m_p$.

With APIC, each particle also stores matrix C_p^n to define an affine velocity local to the particle. With this convention, the momentum contribution from particle p to node i is $w_{ip}^n m_p (\mathbf{v}_p^n + C_p^n(\mathbf{x}_i^n - \mathbf{x}_p^n))$. For each grid node i , we sum the contribution from all particles p to get grid momentum and then divide by the grid mass to get the grid velocity as $\mathbf{v}_i^n = \frac{1}{m_i^n} \sum_p w_{ip}^n m_p (\mathbf{v}_p^n + C_p^n(\mathbf{x}_i^n - \mathbf{x}_p^n))$.

4.2.2 Grid to particle: After the update of grid node momentum, we transfer grid velocities to particles. We use $\tilde{\mathbf{v}}_i^{n+1}$ to denote grid node velocities after the momentum updates (see Section (§4.4)). The grid to particle transfer with APIC is written as $\mathbf{v}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_i^{n+1}$ and $\tilde{\mathbf{C}}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_i^{n+1} ((\frac{6-b^d}{h^2})(\mathbf{x}_i^n - \mathbf{x}_p^n))^T$, where b^d is the b-spline degree ($b^d = 3$ for cubic b-spline interpolation, $b^d = 2$ for quadratic b-spline interpolation) and h is the Eulerian grid spacing. For the linear kernel, $\tilde{\mathbf{C}}_p^{n+1}$ reduces to the velocity gradient. In that case it can be computed as $\tilde{\mathbf{C}}_p^{n+1} = \sum_i \tilde{\mathbf{v}}_i^{n+1} (\nabla w_{ip}^n)^T$.

For traditional MPM particles and Lagrangian mesh vertex particles, we update their positions with $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}$. We enslave

the positions of Lagrangian mesh element quadrature particles to the barycenters of their corresponding Lagrangian elements to prevent drifting that would occur with a traditional MPM update of these positions.

4.2.3 Damping: The Rigid Particle-in-Cell (RPIC) introduced in Jiang et al. [2015] can be thought of as a reduced APIC scheme that damps out stretching and shearing motions while conserving rigid velocity modes. APIC degrades to RPIC if only the skew symmetric part of $\tilde{\mathbf{C}}_p^{n+1}$ is kept. More importantly, if we decompose $\tilde{\mathbf{C}}_p^{n+1}$ into the symmetric part $\tilde{\mathbf{C}}_p^{n+1,s}$ and the skew symmetric part $\tilde{\mathbf{C}}_p^{n+1,k}$, we can stably introduce damping to the system by scaling $\tilde{\mathbf{C}}_p^{n+1,s}$:

$$\mathbf{C}_p^{n+1} = \tilde{\mathbf{C}}_p^{n+1,k} + (1 - \nu) \tilde{\mathbf{C}}_p^{n+1,s}, \quad (2)$$

where $\nu \in [0, 1]$ is the damping coefficient. This allows us to control the damping on stretching and shearing without damping local or global rigid motions. In practice any value of ν may be chosen depending on how energetic a simulation needs to be. Similarly to Muller et al. [2007] and Kaldor et al. [2008], our damping scheme damps non-rigid motion in a stable way without introducing any additional time step restriction. Notably, this works well with explicit symplectic Euler time stepping, unlike many more standard approaches like Rayleigh damping.

4.3 Deformation gradient update

It is essential to conceive of our continuum samples as coherent surfaces or curves, rather than as unstructured particles. This was first observed for elastic volumes and surfaces (without self-collision) by Jiang et al. [2015]. For these meshes, they abandon the traditional Eulerian MPM update of the deformation gradient and use the mesh connectivity to compute the manifold components of the deformation gradient. We also use this approach for our triangulated surfaces and segmented curves. This allows for a Lagrangian update of the deformation gradient that does not suffer from the numerical plasticity or fracture of the Eulerian update. While their approach allowed for coupling of Lagrangian elastic meshes with standard MPM materials, self-contact with codimensional meshes is not well resolved. Our approach combines the method of Jiang et al. [2015], with the Eulerian deformation of the grid, to obtain the full rank deformation gradient, which allows for penalizing self-contact, while still preventing numerical plasticity and fracture. A comparison of these methods is shown in Figure 10.

The deformation gradient is updated in terms of the motion of the grid over a time step. We describe this in terms of \mathbf{x}_i , the Eulerian

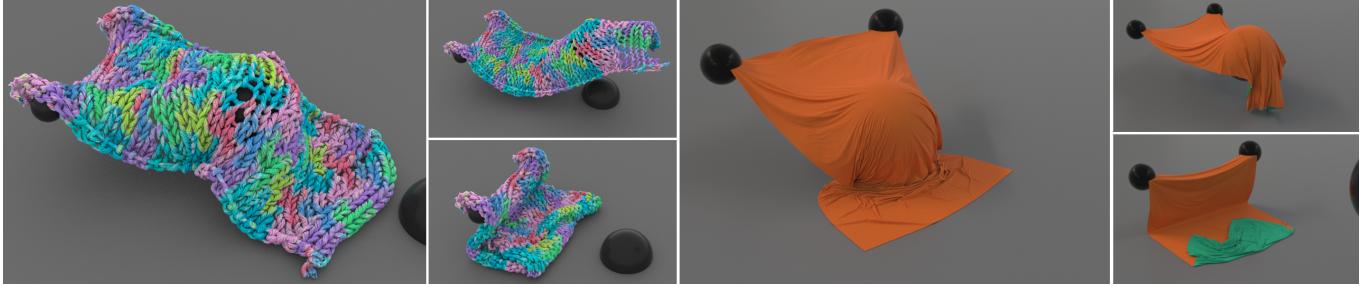


Fig. 14. Left: A knitted curtain is pushed by a sphere and folds back. Right: A cloth curtain with around 1.78M triangles is hit by a ball.

grid nodes at the beginning of the time step and $\hat{\mathbf{x}}_i$ their location after a grid update. Although we never deform the grid, it is useful to conceive of the grid momentum update in terms of the motion it would cause on the grid. In particular this allows us to define the elastic grid forces through the differentiation of a potential. This use of incremental grid node motion, is called the updated Lagrangian view, as we essentially view the motion of the grid as Lagrangian, albeit from the grid configuration at time t^n , instead of the rest configuration. Using $\hat{\mathbf{v}}_i^{n+1}$ to denote the grid node velocity after the momentum update (see (§4.4)), we have $\hat{\mathbf{x}}_i = \mathbf{x}_i + \Delta t \hat{\mathbf{v}}_i^{n+1}$. Using $\hat{\mathbf{x}}, \hat{\mathbf{v}}$ to denote the vector of all moved grid positions $\hat{\mathbf{x}}_i$ and velocities $\hat{\mathbf{v}}_i^{n+1}$ respectively we can define the deformation gradient update on particle p in terms of $\hat{\mathbf{F}}_p^E(\hat{\mathbf{x}})$.

For traditional MPM particles (type (i)), the elastic deformation gradient is updated as

$$\hat{\mathbf{F}}_p^E(\hat{\mathbf{x}}) = (\nabla \hat{\mathbf{x}})_p \mathbf{F}_p^{E,n}, \quad (3)$$

where $(\nabla \hat{\mathbf{x}})_p = \sum_i \hat{\mathbf{x}}_i (\nabla w_{ip}^n)^T$ and $\mathbf{F}_p^0 = \mathbf{I}$ assuming no initial deformation. For mesh vertex particles (type (ii)), we do not store the deformation gradient so nothing needs to be done. For a Lagrangian mesh element quadrature particle (type (iii)), we use



Fig. 15. By stretching, twisting and releasing a knitted cloth from different directions, we demonstrate the ability of our method in handling yarn level anisotropic characteristics.

$\hat{\mathbf{F}}_p^E(\hat{\mathbf{x}}) = \hat{\mathbf{d}}_p^E \mathbf{D}_p^{-1}$. Here $\hat{\mathbf{d}}_p^E(\hat{\mathbf{x}}) = [\hat{\mathbf{d}}_{p,1}^E(\hat{\mathbf{x}}), \hat{\mathbf{d}}_{p,2}^E(\hat{\mathbf{x}}), \hat{\mathbf{d}}_{p,3}^E(\hat{\mathbf{x}})]$ and $\mathbf{D} = [\mathbf{D}_{p,1}, \mathbf{D}_{p,2}, \mathbf{D}_{p,3}]$ are matrices whose columns are respectively the deformed elastic and initial material directions. For the ζ (1 for curves and 2 for surfaces) directions tangent to the manifold we choose $\mathbf{D}_{p,\beta} = \mathbf{X}_{\text{mesh}(p,\beta)} - \mathbf{X}_{\text{mesh}(p,0)}$, to be the undeformed mesh element edge vectors (where $\beta = 1, \dots, \zeta$), and then $\hat{\mathbf{d}}_{p,\beta}^E(\hat{\mathbf{x}}) = \hat{\mathbf{d}}_{p,\beta}(\hat{\mathbf{x}}) = \hat{\mathbf{x}}_{\text{mesh}(p,\beta)} - \hat{\mathbf{x}}_{\text{mesh}(p,0)}$ are the deformed edge vectors, where the deformed mesh particle positions are interpolated from the deformed grid node positions $\hat{\mathbf{x}}_q = \sum_i \hat{\mathbf{x}}_i w_{iq}^n$. Note that we do not need the superscript E for ζ directions $\hat{\mathbf{d}}_{p,\beta}$ tangent to the manifold since they will not experience plasticity. Also, we use the notation $\text{mesh}(p,\beta)$ for $\beta = 0 \dots \zeta$ to denote the mesh vertex particle (type (ii)) indices of the mesh element corresponding to Lagrangian mesh element quadrature particle p . We choose the remaining $3 - \zeta$ directions \mathbf{D}_β to be of unit length and normal to the manifold, and evolve them as in traditional MPM via

$$\hat{\mathbf{d}}_{p,\beta}^E(\hat{\mathbf{x}}) = (\nabla \hat{\mathbf{x}})_p \mathbf{d}_{p,\beta}^{E,n}. \quad (4)$$

In both cases, $\hat{\mathbf{F}}_p^E$ is updated ignoring any further plasticity. $\hat{\mathbf{F}}_p^E$ will be further processed for plasticity to obtain $\mathbf{F}_p^{E,n+1}$ in (§4.5).

4.4 Grid momentum update

Here we describe the update of the grid node momentum (i.e. velocities since the grid node masses do not change in the update): $\mathbf{v}_i^n \rightarrow \hat{\mathbf{v}}_i^{n+1}$. We use the hat in $\hat{\mathbf{v}}_i^{n+1}$ to distinguish this update from the velocities that would transferred to the grid in the next time step \mathbf{v}_i^{n+1} . As in (§4.3), we use $\hat{\mathbf{x}}, \hat{\mathbf{v}}$ to denote the vector of all grid moved positions $\hat{\mathbf{x}}_i$ and velocities $\hat{\mathbf{v}}_i^{n+1}$ respectively. We can denote the grid momentum update as

$$\hat{\mathbf{v}}_i^{n+1} = \mathbf{v}_i^n + \frac{\Delta t}{m_i^n} \mathbf{f}_i(\hat{\mathbf{x}}(\hat{\mathbf{q}})) + \Delta t \mathbf{g}, \quad (5)$$

where \mathbf{f} is the elastic force, \mathbf{g} is the gravitational acceleration and $\hat{\mathbf{q}} = \mathbf{0}$ for symplectic Euler or $\hat{\mathbf{q}} = \hat{\mathbf{v}}$ for backward Euler. We use the notation $\hat{\mathbf{x}}(\hat{\mathbf{q}})$ to emphasize the dependence of the moved grid nodes through $\hat{\mathbf{x}}_i = \mathbf{x}_i + \Delta t \mathbf{q}_i^{n+1}$.

The elastic force \mathbf{f} is defined in terms of the elastic potential energy. Given elastic energy density ψ , the total potential is computed as $\Phi = \int_\Omega \psi dV$ over material domain Ω . When discretely approximated on particles, we have

	ρ	E	η	γ	k	Φ_F	ν
Cloth Curtain (Fig. 14)	2	400	0.3	0	800	0°	0.8
Cloth Curtain x3 (Fig. 1)	2	400	0.3	0	800	0°	0.8
Cloth & Ball (Fig. 3)	2	200	0.3	0	100	0°	0.7
Cloth Twister (Fig. 4)	2	200	0.3	0	400	0°	0.1
Knit Sweater (Fig. 7)	4	500	0.3	100	500	10°	1.0
Knit Poncho (Fig. 17)	4	500	0.3	500	2000	40°	1.0
Knit Twister (Fig. 5)	4	500	0.3	500	2000	40°	1.0
Knit Curtain (Fig. 14)	4	500	0.3	500	500	20°	1.0
Knit Anisotropy (Fig. 15)	4	500	0.3	0	10000	0°	1.0
Shag Carpet (Fig. 2)	2	400	0.3	0	800	0°	0.8
Tearing Fiber (Fig. 6)	4	100	0.3	500	8000	40°	0.5
Hair Tubes (Fig. 13)	1	60	0.3	10	2000	15°	0.07
Curtain & Goo (Fig. 19)	2	400	0.3	0	40000	0°	1.0
Cloth & Slime (Fig. 18)	2.5	15000	0.35	0	20000	0°	0.85
Curtain & Sand (Fig. 19)	2	400	0.3	0	200000	0°	0.0
Cloth & Sand (Fig. 9)	2	200	0.3	0	40000	0°	0.0

Table 1. Material Parameters: Here ρ refers to the density, E to the Young's modulus, η to the Poisson's ratio, γ to the shearing stiffness, k to the stiffness, Φ_F to the internal friction angle, and ν to the damping coefficient. For surfaces, the friction coefficient $c_F = \tan(\Phi_F)$. For curves, the friction coefficient along the fiber is $\beta = \tan(\Phi_F)$. The one perpendicular to the fiber is $\alpha = \sqrt{\frac{2}{3} \frac{2 \sin(\Phi_F)}{3 - \sin(\Phi_F)}}$.

	s/Frame	Element #	Particle #	Δt	Δx	Mem(GB)
Cloth Curtain (Fig. 14)	120	1.78M	2.67M	7×10^{-5}	2.2×10^{-3}	2.36
Cloth Curtain x3 (Fig. 1)	123	1.39M	2.09M	7×10^{-5}	2.2×10^{-3}	2.16
Cloth & Ball (Fig. 3)	86	2.00M	3.00M	1×10^{-4}	1.5×10^{-3}	2.66
Cloth Twister (Fig. 4)	78	1.14M	1.71M	7×10^{-5}	3×10^{-3}	1.63
Knit Sweater (Fig. 7)	13	0.27M	0.54M	1×10^{-4}	1×10^{-2}	0.39
Knit Poncho (Fig. 17)	63	0.30M	0.61M	5×10^{-5}	1.6×10^{-2}	0.56
Knit Twister (Fig. 5)	10	0.31M	0.62M	2×10^{-4}	1.6×10^{-2}	0.45
Knit Curtain (Fig. 14)	17	0.31M	0.62M	1×10^{-4}	4×10^{-3}	0.99
Knit Anisotropy (Fig. 15)	38	0.31M	0.62M	5×10^{-5}	7×10^{-4}	0.46
Shag Carpet (Fig. 2)	26	0.48M	0.97M	1×10^{-4}	1×10^{-2}	0.65
Tearing Fiber (Fig. 6)	29	1.04M	2.15M	1×10^{-4}	3.2×10^{-2}	1.09
Hair Tubes (Fig. 13)	11	0.38M	0.76M	2×10^{-4}	7×10^{-3}	0.20
Curtain & Goo (Fig. 19)	62	0.07M	2.59M	2×10^{-4}	7.5×10^{-3}	0.83
Cloth & Slime (Fig. 18)	47	0.04M	1.88M	2×10^{-4}	1.2×10^{-2}	0.59
Curtain & Sand (Fig. 19)	85	0.06M	2.90M	2×10^{-4}	7.2×10^{-3}	1.36
Cloth & Sand (Fig. 9)	876	0.16M	7.23M	1×10^{-4}	5×10^{-3}	3.13

Table 2. All simulations are run on Intel Xeon E5-2690 V2 with 19 threads. Element # denotes number of triangles (for surfaces), number of segments (for curves) or the total count (for coupling simulations). Memory usage is measured in the end of the first frame. Δt here means maximum allowed time step. The actual running Δt is adaptive and may be restricted by CFL condition when the particle velocities are high. In all of our simulations we use a CFL number equal to 0.3, i.e., we don't allow particles to move further than $0.3\Delta x$ in a time step.

$$\Phi = \sum_{p \in \mathcal{I}^{(i)} \cup \mathcal{I}^{(iii)}} V_p^0 \psi(\hat{\mathbf{F}}_p^E),$$

where V_p^0 is the undeformed volume of particle p . To compute grid node forces, we differentiate Φ as a function of $\hat{\mathbf{x}}$, using the definitions from (§4.3) to express $\hat{\mathbf{F}}_p^E$ as a function of $\hat{\mathbf{x}}$. The force on node i is then given as

$$\mathbf{f}_i(\hat{\mathbf{x}}) = -\frac{\partial \Phi}{\partial \hat{\mathbf{x}}_i} = -\sum_{p \in \mathcal{I}^{(i)} \cup \mathcal{I}^{(iii)}} V_p^0 \frac{\partial \psi}{\partial \mathbf{F}^E}(\hat{\mathbf{F}}_p^E(\hat{\mathbf{x}})) : \frac{\partial \hat{\mathbf{F}}_p^E}{\partial \mathbf{x}_i}, \quad (6)$$

Note that $\frac{\partial \hat{\mathbf{F}}_p^E}{\partial \mathbf{x}_i}$ is a third order tensor, and does not depend on $\hat{\mathbf{x}}$ because $\hat{\mathbf{F}}_p^E$ is linear in $\hat{\mathbf{x}}_i$. This tensor has a convenient property that allows us to split the computation into three parts (see [Jiang et al. 2017a] for a derivation). The first part corresponds to standard MPM particles and we denote this as $\mathbf{f}_i^{(i)}(\hat{\mathbf{x}}) = -\sum_{p \in \mathcal{I}^{(i)}} V_p^0 \left(\frac{\partial \psi}{\partial \mathbf{F}^E}(\hat{\mathbf{F}}_p^E(\hat{\mathbf{x}})) \right)$:

$\frac{\partial \hat{\mathbf{F}}_p^E}{\partial \mathbf{x}_i}$. Next, by expressing \mathbf{F} using a material space coordinate frame which is aligned with the element, we can assume that the first ζ columns of \mathbf{F} correspond to the in manifold deformation [Jiang et al. 2017a]. This means that the first ζ columns of $\hat{\mathbf{F}}_p^E$ correspond to the stress response inside the manifold and these terms can be computed by first computing forces on Lagrangian mesh vertex particles (type (ii)) and then transferring them to the grid. Denoting these vertex forces with $\mathbf{f}_p^{(ii)}(\hat{\mathbf{x}})$, their corresponding contribution to grid node forces can be computed as $\mathbf{f}_i^{(ii)}(\hat{\mathbf{x}}) = \sum_{p \in \mathcal{I}^{(ii)}} w_{ip}^n \mathbf{f}_p(\hat{\mathbf{x}})$.

The last $3 - \zeta$ columns of $\frac{\partial \psi}{\partial \mathbf{F}^E}$ correspond to the normal space contribution. This part is computed by summing over all Lagrangian mesh element quadrature particles (type (iii)). Differentiating the last ζ columns of $\hat{\mathbf{F}}_p^E(\hat{\mathbf{x}})$ with respect to $\hat{\mathbf{x}}_i$ is done similarly to traditional MPM through $\mathbf{d}_{\beta}^E(\hat{\mathbf{x}}) = (\nabla \hat{\mathbf{x}})_p \mathbf{d}_{\beta}^{E,n}$. This contribution to the force is given as $\mathbf{f}_i^{(iii)}(\hat{\mathbf{x}}) = -\sum_{p \in \mathcal{I}^{(iii)}} \sum_{\beta=\zeta+1}^3 V_p^0 \frac{\partial \psi}{\partial \mathbf{F}^E} \mathbf{d}_{\beta}^T \nabla w_{ip}^n$, where \mathbf{F}_{β}^E denotes the β th column of \mathbf{F}^E . With this view, we have $\mathbf{f}_i(\hat{\mathbf{x}}) = \mathbf{f}_i^{(i)}(\hat{\mathbf{x}}) + \mathbf{f}_i^{(ii)}(\hat{\mathbf{x}}) + \mathbf{f}_i^{(iii)}(\hat{\mathbf{x}})$.

When using backward Euler, we solve the system similar to [Stomakhin et al. 2013] by utilizing the Hessian of Φ with respect to $\hat{\mathbf{x}}$ and ignoring the effects of plasticity during the solve. The action of this Hessian on an arbitrary increment $\delta \mathbf{u}$ can be expressed as

$$\delta f_{i\alpha} = \frac{\partial f_{i\alpha}}{\partial x_{j\lambda}} \delta u_{j\lambda} = -\sum_{p \in \mathcal{I}^{(i)} \cup \mathcal{I}^{(iii)}} V_p^0 \frac{\partial F_{\beta\zeta}^E}{\partial x_{i\alpha}} \frac{\partial^2 \psi}{\partial F_{\beta\zeta}^E \partial F_{\alpha\sigma}^E} \frac{\partial F_{\alpha\sigma}^E}{\partial x_{j\lambda}} \delta u_{j\lambda},$$

where we used implicit summation on repeated indices. We use Newton's method to solve the nonlinear system. Klar et al. [2016] recently investigated including the effects of plasticity in the implicit solve, but their approach has non-symmetric force derivatives that require GMRES when using Newton's method. Our lagged plasticity approach has symmetric force derivatives and thus we use MINRES to solve the linearized systems.

4.5 Return mapping

The return mapping is the discrete version of the yield condition and flow rule. It takes the form of a projection, which enforces the yield condition on the stress, with the direction of the flow rule, defining the direction of the projection. The return mapping is applied after the deformation gradient update so that the updated stress satisfies the yield condition. To perform the return mapping, first we compute the QR-decomposition of $\hat{\mathbf{d}}^E = \hat{\mathbf{F}}^E \mathbf{D} = \mathbf{Q} \hat{\mathbf{R}}$. Then we project \mathbf{R}^{n+1} from $\hat{\mathbf{R}}$ to satisfy the yield condition, and finally we construct $\mathbf{F}^{E,n+1} = \mathbf{Q} \mathbf{R}^{n+1} (\mathbf{D}^{-1})$. We want the stress to satisfy $\mathbf{s}(\mathbf{n}, \theta)^T \boldsymbol{\sigma} \mathbf{n} + c_F \mathbf{n}^T \boldsymbol{\sigma} \mathbf{n} \leq 0$, for every normal \mathbf{n} to the curve, and every tangential direction $\mathbf{s}(\mathbf{n}, \theta)$ perpendicular to \mathbf{n} . In the following discussion we use the convention that $\hat{\boldsymbol{\sigma}}$ denotes the stress evaluated at $\hat{\mathbf{R}}$ (the hypothetical stress if no plastic flow occurs), while $\boldsymbol{\sigma}$ denotes the stress evaluated at \mathbf{R}^{n+1} (the stress accounting for plasticity), and likewise for the \hat{f}_n and f_n , and other hatted and non-hatted variables.

4.5.1 Surfaces. For surfaces we fix the normal $\mathbf{n} = \mathbf{q}_3$, and further eliminate $\mathbf{s}(\mathbf{n}, \theta)$ by noting that maximum of $\mathbf{s}(\mathbf{n}, \theta)^T \mathbf{t}$ is obtained at the angle corresponding to the projection of \mathbf{t} into the

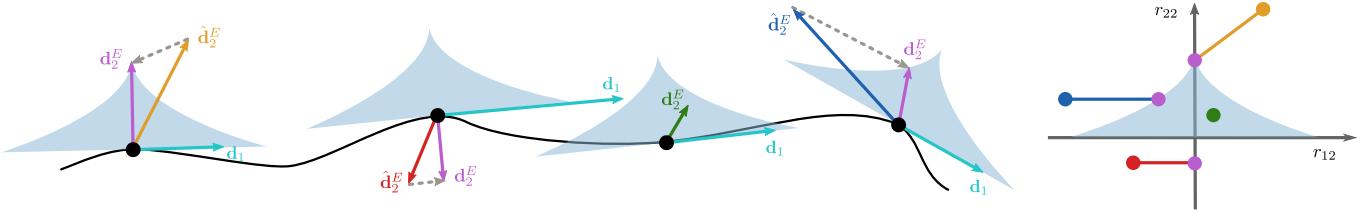


Fig. 16. The left figure visualizes the return mapping for 2D curve. \mathbf{d}_1^E is unchanged. Each shaded region corresponds to the feasible state of the deformed material normal \mathbf{d}_2^E , and is determined by the yield function. Right figure shows the same yield surface in $r_{12}r_{22}$ space. The tip of the yield surface corresponds to the world space manifold normal and is usually different from \mathbf{d}_2 . Return mapping takes a trial strain $\hat{\mathbf{d}}_2$ and results in the final \mathbf{d}_2^E . The green particle corresponds to $\hat{\mathbf{d}}_2$ that is inside the yield surface and exhibits an elastic response. The blue particle corresponds to $\hat{\mathbf{d}}_2^E$ that is under compression but experiences more shear than friction allows. Such a configuration is projected to the yield surface along the direction that avoids normal component change. The orange particle corresponds to $\hat{\mathbf{d}}_2^E$ that is experiencing tension and is projected to the tip of the yield surface. The corresponding stress is zero, allowing materials to separate freely. The red particle corresponds to $\hat{\mathbf{d}}_2^E$ that is inverted from the original side of material (due to discrete numerical step). In this case, we disable shearing by nullify the r_{22} component, and rely on elasticity to penalize such configuration. In practice we find that the red case rarely happens.

plane with normal \mathbf{n} : $\max_\theta \mathbf{s}(\mathbf{n}, \theta)^T \mathbf{t} = |\mathbf{t} + f_n \mathbf{n}|$. We first check if $\hat{r}_{33} > 1$, if so the normal is in extension and no friction should occur, so we set $r_{13} = r_{23} = 0$, further we set $r_{33} = 1$, as we are using it to penalize collisions and previous stretching in that direction should not be taken into account. If $\hat{r}_{33} \leq 1$, we check the condition, $|\hat{\mathbf{t}} - (\hat{\mathbf{t}}^T \mathbf{n})\mathbf{n}| + c_F \mathbf{n}^T \hat{\mathbf{t}} \leq 0$, where $\mathbf{n} = \mathbf{q}_3$. The traction is given by $\mathbf{t}(\mathbf{R}) = \frac{1}{r_{11}r_{22}}(yr_{13}\mathbf{q}_1 + yr_{23}\mathbf{q}_2 + k(r_{33} - 1)^2\mathbf{q}_3)$, see the supplemental technical document [Jiang et al. 2017a] for the derivation. Thus the magnitude of the normal traction is given by $f_n(\mathbf{R}) = \frac{k}{r_{11}r_{22}}(r_{33} - 1)^2$ and the magnitude of the tangential traction by $f_f(\mathbf{R}) = \frac{Y}{r_{11}r_{22}}\sqrt{r_{13}^2 + r_{23}^2}$. If $\hat{f}_f \leq c_F f_n$ is violated, we uniformly scale $r_{13} = c_f \frac{f_n}{f_f} \hat{r}_{13}$ and $r_{23} = c_f \frac{f_n}{f_f} \hat{r}_{23}$ so that $f_f = c_F f_n$. We only change the r_{13} and r_{23} components, as these represent shearing along the surface of the cloth. Furthermore only the magnitude of the tangential traction should change, and not the direction, which explains the uniform scale. Note that the return mapping is independent of the choice of in-plane energy $h(\mathbf{R}_1)$. E.g., one could use the in-plane surface energy from [Baraff and Witkin 1998] instead. See Figure 16, for an illustration of the return mapping.

4.5.2 Curves. Many fibrous materials will have anisotropic friction, for example most hair is smoother along the fiber direction than perpendicular to it. For this reason, we use an anisotropic friction law, that is stricter and moreover easier to apply. Let $\sigma_{ij} = \mathbf{q}_i^T \boldsymbol{\sigma} \mathbf{q}_j$ be the entries of $\boldsymbol{\sigma}$ in the \mathbf{Q} basis, and $J_2 = (\sigma_{22} - \sigma_{33})^2 + 4\sigma_{23}$. Then our conditions are $\sqrt{J_2} + \frac{\alpha}{2}(\sigma_{22} + \sigma_{33}) \leq 0$, which is Mohr-Coulomb in the $\mathbf{q}_2, \mathbf{q}_3$ plane, and $\sqrt{\sigma_{12}^2 + \sigma_{13}^2} + \frac{\beta}{2}(\sigma_{22} + \sigma_{33}) \leq 0$ which controls the friction along the fiber. Where α and β are material parameters, larger α increases friction perpendicular to the fibers, and larger β increases friction along the fiber.

To see that these stricter conditions imply that $\mathbf{s}(\mathbf{n}, \theta)^T \boldsymbol{\sigma} \mathbf{n} + c_F \mathbf{n}^T \boldsymbol{\sigma} \mathbf{n} \leq 0$, with $c_F = \alpha + \beta$. Observe that the direction $\mathbf{s}(\mathbf{n}, \theta)$, can be written as $c\mathbf{q}_1 + s\hat{\mathbf{q}}$, where $c^2 + s^2 = 1$, and $\hat{\mathbf{q}}$ is in the $\mathbf{q}_2, \mathbf{q}_3$ plane. Thus $f_f - c_F f_n \leq |\mathbf{q}_1^T \boldsymbol{\sigma} \mathbf{n}| + |\hat{\mathbf{q}}^T \boldsymbol{\sigma} \mathbf{n}| + c_F \mathbf{n}^T \boldsymbol{\sigma} \mathbf{n}$. Note that $|\hat{\mathbf{q}}^T \boldsymbol{\sigma} \mathbf{n}| + c_F \mathbf{n}^T \boldsymbol{\sigma} \mathbf{n} \leq \sqrt{J_2} + \frac{c_F}{2}(\sigma_{22} + \sigma_{33})$, from the derivation of

2D Mohr-Coulomb in Klar et al. [2016]. Therefore $f_f - c_F f_n \leq \sqrt{J_2} + \frac{\alpha}{2}(\sigma_{22} + \sigma_{33}) + \sqrt{\sigma_{12}^2 + \sigma_{13}^2} + \frac{\beta}{2}(\sigma_{22} + \sigma_{33})$, which is less or equal to 0 by assumption.

To enforce these constraints we first perform the 2D volume preserving return mapping from Klar et al. [2016] on $\hat{\mathbf{R}}_3$. This is easiest to describe in terms of $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$, the logarithms of the singular values of $\hat{\mathbf{R}}_3 = \mathbf{U} \exp(\hat{\boldsymbol{\epsilon}}) \mathbf{V}^T$, (with $\hat{\epsilon}_1 > \hat{\epsilon}_2$). First we check if $\hat{\epsilon}_1 + \hat{\epsilon}_2 \geq 0$, in which case the hair has expanded laterally and no friction should occur and $\epsilon_1 = \epsilon_2 = 0$. Otherwise we check $\sqrt{J_2} + \frac{\alpha}{2}(\hat{\sigma}_{22} + \hat{\sigma}_{33}) \leq 0$, if so no plasticity occurs and $\epsilon_1 = \hat{\epsilon}_1$ and $\epsilon_2 = \hat{\epsilon}_2$. Otherwise plastic flow occurs and we set $\epsilon_1 = \hat{\epsilon}_1 - \eta$, $\epsilon_2 = \hat{\epsilon}_2 - \eta$, which will preserve $\epsilon_1 + \epsilon_2 = \hat{\epsilon}_1 + \hat{\epsilon}_2$. We choose $\eta = \frac{\hat{\epsilon}_1 - \hat{\epsilon}_2}{2} + \frac{\alpha\lambda}{4\mu}(\hat{\epsilon}_1 + \hat{\epsilon}_2)$ to make $\sqrt{J_2} + \frac{\alpha}{2}(\sigma_{22} + \sigma_{33}) = 0$. We then construct $\mathbf{R} = \mathbf{U} \exp(\boldsymbol{\epsilon}) \mathbf{V}^T$. Then we check the condition $\sqrt{\hat{\sigma}_{12}^2 + \hat{\sigma}_{13}^2} + \frac{\beta}{2}(\hat{\sigma}_{22} + \hat{\sigma}_{33}) \leq 0$ for shearing tangent to the fibers, and if violated scale $r_{12} = \zeta \hat{r}_{12}$ and $r_{13} = \zeta \hat{r}_{13}$, to satisfy $\sqrt{\sigma_{12}^2 + \sigma_{13}^2} + \frac{\beta}{2}(\sigma_{22} + \sigma_{33}) = 0$, i.e. $\zeta = -\frac{\beta(\hat{\sigma}_{22} + \hat{\sigma}_{33})}{2\sqrt{\hat{\sigma}_{12}^2 + \hat{\sigma}_{13}^2}}$.

5 RESULTS

We list the runtime performance for all of our examples in Table 2. The parameter choices are given in Table 1. All simulations are run on Intel Xeon E5-2690 V2 with 19 threads.

5.1 Effect of friction

In Figure 12, we simulate 2D curves with different friction coefficients dropping on a block. Internal friction is naturally controlled via adjusting c_F .

5.2 Cloth

We demonstrate the efficacy of our surface elastoplasticity model by simulating cloth triangulated surfaces in various scenarios involving numerous self collisions (Figure 1, 4, 3, 14). Our elastoplasticity model successfully resolves frictional contacts with a modest computational cost. Notably, the run time does not typically grow with the number of colliding primitives. For most cloth examples, we

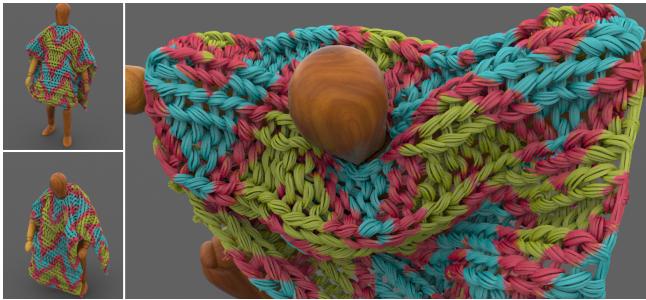


Fig. 17. A character wearing a knitted poncho performing a jumping motion. Our fiber elastoplasticity model captures detailed frictional contact behavior at the yarn level with multiple fiber threads per yarn. The simulation runs at 1 minute per frame on average.

choose zero friction ($c_F = 0$ and $\gamma = 0$) to enable smooth sliding between cloth pieces.

5.3 Knitted garments, hair and fibers

We demonstrate the fiber elastoplasticity with yarn level simulation of knitted fabric. We can even simulate multiple threads per yarn. We use the mesh models from Cirio et al. [2014] and Yuksel et al. [2012] and simulate different examples including dropping a sweater (Figure 7), a jumping character wearing a poncho (Figure 17), twisting a knitted cloth with a cylinder (Figure 5), and hitting a knitted cloth curtain with a ball (Figure 14). By accurately capturing frictional contact between yarns, our method naturally reproduces anisotropic stretching behaviors governed by knit patterns (Figure 15). Our fiber model is also suitable for simulating hair (Figure 13), shag carpet (Figure 2) and other fibrous materials (Figure 6). We simulate every single strand in these examples. Our method scales well regardless of the complexity of segment contacts. We use RPIC damping for stabilizing the simulations under large timesteps.

5.4 Two-way coupling

Like other MPM solvers, our method naturally handles multiple material coupling without requiring any additional treatment. The constitutive model is defined on each particle. A sand column with 7M particles couples with a piece of cloth, as shown in Figure 9 and Figure 1. In the attached video we selectively colored the sand particles white based on their location in the final frame to spell out the word SIGGRAPH. We simulate a cloth curtain getting hit by a ball while interacting with sand/googly eyes in Figure 19. In Figure 18, we drip viscous green slime into an elastic bear geometry. A relatively high collision stiffness (k) is set on the cloth to prevent sand from sticking to it.

5.5 Comparison with previous MPM methods

We compare the result of our method with other MPM approaches by simulating multiple curves dropping onto a block (Figure 10). Traditional meshless MPM as in Stomakhin et al. [2013] causes numerical fracture of the material. The method of Jiang et al. [2015] only models the manifold stress response, resulting in unnatural

clumping and numerical cohesion of curves. With our method, if we only apply elasticity, curves successfully separate from each other but exhibit excessive frictional effects. When we adopt the full elastoplasticity model, the dynamics becomes natural and realistic, and exhibits a smooth frictional sliding appearance.

6 LIMITATIONS AND FUTURE WORK

6.1 Limitations

Our method uses a regular grid to resolve material interactions. The apparent thickness of the surface/curve depends on the grid resolution and on the repulsion energy model. If the grid resolution is too large, there can be some artifacts, mostly manifest as numerical separation where particles tend to stay separated by an amount about half a grid cell width or numerical stickiness where the motion of the grid is not accurate enough to resolve the proper behavior. Therefore it is difficult to simulate perfectly frictionless materials and it is difficult to control the apparent thickness independent of the grid. Furthermore the method works best if the size of the surface or curve elements is at most the size of the grid spacing. A mesh that is too fine compared to the grid may cause persistent wrinkles. A mesh that is too coarse compared to the grid may cause self penetration because the material coverage is not resolved well enough on the grid. When penetration accidentally happens (due to Δt being too big), it has a chance of being resolved because elasticity will still penalize that inverted collision direction. In practice we try to prevent this from happening by enforcing the CFL condition when choosing Δt . We note that this often requires rather small time steps and increases the overall run time.

While clearly quite different from traditional purely Lagrangian cloth solvers, our method has many aspects that are analogous to their standard components. For example, when we transfer from particles to grid, we essentially determine local neighbor information. This is analogous to e.g. bounding box hierarchy proximity queries in a traditional approach. Furthermore, our penalty on compression in the normal direction is analogous to repulsion springs commonly used in Lagrangian approaches. Notably, we have no fail-safe to prevent self penetration when the repulsions fail. We note that in general for high grid resolution simulation, our approach likely requires smaller time steps since it does not have fail-safes. Interestingly though, our performance is still very competitive for the high resolution examples. E.g., for the cloth curtain example (Fig. 14), we achieve computation times of 2 minutes per frame on average for 1.8 million triangles. Selle et al. [2009] run the same benchmark with 1.7 million triangles with an average frame computation time of 45 minutes using 16 processors. Tang et al. [2016] are able to run the same example at 35 seconds per frame with 1 million triangles on the GPU. We are also competitive with adaptive approaches in Narain et al. [2012].

6.2 Future work

Our method tends to scale as well as the particle-grid transfers. Currently, we parallelize this process over multiple CPU threads. In the future, we believe significant performance gains can be achieved with a GPU implementation. Also, our discretization of surface elasticity although derived from a continuum, resembles a Mindlin

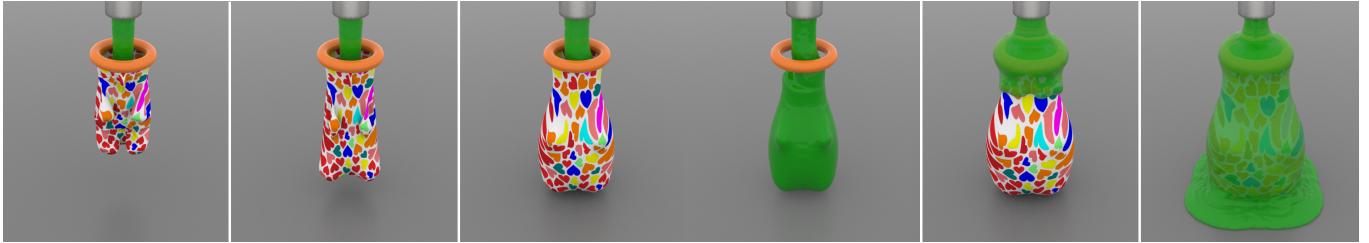


Fig. 18. Our method naturally handles two-way coupling of solids and fluids, as demonstrated with green slime and a cloth bear.



Fig. 19. Top: A cloth curtain is hit by a ball and couples with sand. Bottom: A cloth curtain is hit by a ball and couples with blue goo.

Reissner plate model [Mindlin 1951], in that material local to the surface moves in straight lines with the normal. Thus it would be interesting future work to investigate a continuum based bending model that works with this view.

ACKNOWLEDGMENTS

The authors would like to thank Gabriel Cirio, Doug James, Steve Marschner, Miguel Otaduy and Cem Yuksel for contributing their knitted meshes. The work is supported by DoD W81XWH-15-1-0147.

REFERENCES

- S. Ainsley, E. Vouga, E. Grinspun, and R. Tamstorf. 2012. Speculative Parallel Asynchronous Contact Mechanics. *ACM Trans Graph* 31, 6 (2012), 151:1–151:8.
- K. Anjyo, Y. Usami, and T. Kurihara. 1992. A Simple Method for Extracting the Natural Beauty of Hair. *SIGGRAPH Comp Graph* 26, 2 (1992), 111–120.
- Y. Bando, B. Chen, and T. Nishita. 2003. Animating hair with loosely connected particles. In *Comp Graph Forum*, Vol. 22. Wiley Online Library, 411–418.
- D. Baraff and A. Witkin. 1998. Large Steps in Cloth Simulation. In *Proc ACM SIGGRAPH (SIGGRAPH '98)*, 43–54.
- F. Bertails. 2009. Linear Time Super-Helices. In *Comp Graph Forum*, Vol. 28. Wiley Online Library, 417–426.
- F. Bertails, B. Audoly, M. Cani, B. Querleux, F. Leroy, and J. Lévéque. 2006. Super-helices for Predicting the Dynamics of Natural Hair. *ACM Trans Graph* 25, 3 (2006), 1180–1187.
- F. Bertails-Descoubes, F. Cadoux, G. Daviet, and V. Acary. 2011. A Nonsmooth Newton Solver for Capturing Exact Coulomb Friction in Fiber Assemblies. *ACM Trans Graph* 30, 1 (2011), 6:1–6:14.
- J. Bonet and R. Wood. 2008. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press.
- S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans Graph* 33, 4 (2014), 154:1–154:11.
- J. Brackbill and H. Ruppel. 1986. FLIP: A method for adaptively zoned, Particle-In-Cell calculations of fluid flows in two dimensions. *J Comp Phys* 65 (1986), 314–343.
- David E. Breen, Donald H. House, and Michael J. Wozny. 1994. Predicting the Drape of Woven Cloth Using Interacting Particles. In *Proc SIGGRAPH (SIGGRAPH '94)*. ACM, 365–372.
- R. Bridson. 2008. *Fluid simulation for computer graphics*. Taylor & Francis.
- R. Bridson, R. Fedkiw, and J. Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans Graph* 21, 3 (2002), 594–603.
- M. Chai, C. Zheng, and K. Zhou. 2014. A Reduced Model for Interactive Hairs. *ACM Trans Graph* 33, 4 (2014), 124:1–124:11.
- J. Chang, J. Jin, and Y. Yu. 2002. A Practical Model for Hair Mutual Interactions. In *Proc ACM SIGGRAPH/Eurographics Symp Comp Anim*. ACM, 73–80.
- B. Choe, M. Choi, and H. Ko. 2005. Simulating Complex Hair with Robust Collision Handling. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim*. ACM, 153–160.
- K. Choi and H. Ko. 2002. Stable but Responsive Cloth. *ACM Trans Graph* 21, 3 (2002), 604–611.
- K. Choi and H. Ko. 2005. Advanced topics on clothing simulation and animation. In *Int Conf Comp Grap Int Tech: ACM SIGGRAPH 2005 Course*.

- G. Cirio, J. Lopez-Moreno, D. Miraut, and M. Otaduy. 2014. Yarn-level Simulation of Woven Cloth. *ACM Trans Graph* 33, 6 (2014), 207:1–207:11.
- G. Daviet and F. Bertails-Descoubes. 2016. A Semi-implicit Material Point Method for the Continuum Simulation of Granular Materials. *ACM Trans Graph* 35, 4 (2016), 102:1–102:13.
- G. Daviet, F. Bertails-Descoubes, and L. Boissieux. 2011. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. *ACM Trans Graph* 30, 6 (2011), 139:1–139:12.
- E. de Aguiar, L. Sigal, A. Treuille, and J. Hodgins. 2010. Stable Spaces for Real-time Clothing. *ACM Trans. Graph.* 29, 4 (2010), 106:1–106:9.
- A. Derouet-Jourdan, F. Bertails-Descoubes, G. Daviet, and J. Thollot. 2013. Inverse Dynamic Hair Modeling with Frictional Contact. *ACM Trans Graph* 32, 6 (2013), 159:1–159:10.
- Y. Fan, J. Litven, D. Levin, and D. Pai. 2013. Eulerian-on-lagrangian Simulation. *ACM Trans Graph* 32, 3 (2013), 22:1–22:9.
- Y. Fan, J. Litven, and D. Pai. 2014. Active Volumetric Musculoskeletal Systems. *ACM Trans Graph* 33, 4 (2014), 152:1–152:9.
- F. Faure, J. Allard, and M. Nesme. 2007. *Eulerian Contact for Versatile Collision Processing*. Research Report RR-6203. INRIA, 23 pages.
- W. Feng, Y. Yu, and B. Kim. 2010. A Deformation Transformer for Real-time Cloth Animation. *ACM Trans. Graph.* 29, 4 (2010), 108:1–108:9.
- T. Goktekin, A. Bargteil, and J. O'Brien. 2004. A Method for Animating Viscoelastic Fluids. *ACM Trans Graph* 23, 3 (2004), 463–468.
- A. Golas, R. Narain, and M. Lin. 2014. Continuum modeling of crowd turbulence. *Phys Rev E* 90 (2014), 042816, Issue 4.
- O. Gonzalez and A. Stuart. 2008. *A first course in continuum mechanics*. Cambridge University Press.
- S. Hadap and N. Magnenat-Thalmann. 2001. Modeling dynamic hair as a continuum. In *Comp Graph Forum*, Vol. 20. Wiley Online Library, 329–338.
- F. Harlow. 1964. The particle-in-cell method for numerical solution of problems in fluid dynamics. *Meth Comp Phys* 3 (1964), 319–343.
- D. Harmon, E. Vouga, B. Smith, R. Tamstorf, and E. Grinspun. 2009. Asynchronous Contact Mechanics. *ACM Trans Graph* 28, 3 (2009), 87:1–87:12.
- D. Harmon, E. Vouga, R. Tamstorf, and E. Grinspun. 2008. Robust Treatment of Simultaneous Collisions. *ACM Trans Graph* 27, 3 (2008), 23:1–23:4.
- Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017a. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact: Supplementary Technical Document. (2017).
- C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. 2015. The Affine Particle-In-Cell Method. *ACM Trans Graph* 34, 4 (2015), 51:1–51:10.
- Chenfanfu Jiang, Craig Schroeder, and Joseph Teran. 2017b. An angular momentum conserving affine-particle-in-cell method. *J Comp Phys* 338 (2017), 137 – 164.
- J. Kaldor, D. James, and S. Marschner. 2008. Simulating Knitted Cloth at the Yarn Level. *ACM Trans Graph* 27, 3 (2008), 65:1–65:9.
- J. Kaldor, D. James, and S. Marschner. 2010. Efficient Yarn-based Cloth with Adaptive Contact Linearization. *ACM Trans Graph* 29, 4 (2010), 105:1–105:10.
- D. Kaufman, R. Tamstorf, B. Smith, J. Aubry, and E. Grinspun. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Trans Graph* 33, 4 (2014), 123:1–123:12.
- L. Kavan, D. Gerszewski, A. Bargteil, and P. Sloan. 2011. Physics-inspired Upsampling for Cloth Simulation in Games. *ACM Trans Graph* 30, 4 (2011), 93:1–93:10.
- D. Kim, W. Koh, R. Narain, K. Fatahalian, A. Treuille, and J. O'Brien. 2013. Near-exhaustive Precomputation of Secondary Cloth Effects. *ACM Trans Graph* 32, 4 (2013), 87:1–87:8.
- T. Kim and U. Neumann. 2000. A thin shell volume for modeling human hair. In *Proc Comp Anim*. IEEE, 104–111.
- G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran. 2016. Drucker-prager Elastoplasticity for Sand Animation. *ACM Trans Graph* 35, 4 (2016), 103:1–103:12.
- W. Koh, R. Narain, and J. O'Brien. 2015. View-Dependent Adaptive Cloth Simulation with Buckling Compensation. *IEEE Trans Vis Comp Grap* 21, 10 (2015), 1138–1145.
- Y. Lee, S. Yoon, S. Oh, D. Kim, and S. Choi. 2010. Multi-Resolution Cloth Simulation. In *Comp Graph Forum*, Vol. 29. 2225–2232.
- D. Levin, J. Litven, G. Jones, S. Sueda, and D. Pai. 2011. Eulerian Solid Simulation with Contact. *ACM Trans Graph* 30, 4 (2011), 36:1–36:10.
- D. Li, S. Sueda, D. Neog, and D. Pai. 2013. Thin Skin Elastodynamics. *ACM Trans Graph* 32, 4 (2013), 49:1–49:10.
- H. Li, Y. Wan, and G. Ma. 2011. A CPU-GPU hybrid computing framework for real-time clothing animation. In *2011 IEEE Int Conf Cloud Comp Intel Sys*. IEEE, 391–396.
- T. Liu, A. Bargteil, J. O'Brien, and L. Kavan. 2013. Fast Simulation of Mass-Spring Systems. *ACM Trans Graph* 32, 6 (2013), 209:1–7.
- M. Macklin, M. Müller, N. Chentanez, and T. Kim. 2014. Unified Particle Physics for Real-Time Applications. *ACM Trans Graph* 33, 4 (2014), 104.
- A. McAdams, A. Selle, K. Ward, E. Sifakis, and J. Teran. 2009. Detail Preserving Continuum Simulation of Straight Hair. *ACM Trans Graph* 28, 3 (2009), 62:1–62:6.
- R. Mindlin. 1951. Influence of rotary inertia and shear on flexural motions of isotropic elastic plates. 18 (1951), 31–38.
- M. Müller, N. Chentanez, T. Kim, and M. Macklin. 2015. Air Meshes for Robust Collision Handling. *ACM Trans. Graph.* 34, 4 (2015), 133:1–133:9.
- Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position Based Dynamics. *J Vis Comm Imag Represent* 18, 2 (2007), 109–118.
- M. Müller, T. Kim, and N. Chentanez. 2012. Fast Simulation of Inextensible Hair and Fur. *VRIPHYS* 12 (2012), 39–44.
- R. Narain, A. Golas, S. Curtis, and M. Lin. 2009. Aggregate Dynamics for Dense Crowd Simulation. *ACM Trans Graph* 28, 5 (2009), 122:1–122:8.
- R. Narain, A. Golas, and M. Lin. 2010. Free-flowing granular materials with two-way solid coupling. *ACM Trans Graph* 29, 6 (2010), 173:1–173:10.
- R. Narain, A. Samii, and J. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans Graph* 31, 6 (2012), 152:1–152:10.
- A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. 2006. Physically based deformable models in computer graphics. In *Comp Graph Forum*, Vol. 25. Wiley Online Library, 809–836.
- X. Ni, K. Laxmikant, and R. Tamstorf. 2015. Scalable Asynchronous Contact Mechanics Using Charm++. In *IEEE 29th Int Par Dist Proc Symp*. 677–687.
- M. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. 2009. Implicit Contact Handling for Deformable Objects. *Comp Graph Forum* 28, 2 (2009).
- L. Petrovic, M. Henne, and J. Anderson. 2005. Volumetric methods for simulation and rendering of hair. *Tech Report* (2005).
- X. Provot. 1997. *Collision and self-collision handling in cloth model dedicated to design garments*. Springer Vienna, 177–189.
- N. Schmitt, M. Knuth, J. Bender, and A. Kuijper. 2013. Multilevel Cloth Simulation using GPU Surface Sampling. *VRIPHYS* 13 (2013), 1–10.
- A. Selle, M. Lentine, and R. Fedkiw. 2008. A Mass Spring Model for Hair Simulation. *ACM Trans Graph* 27, 3 (2008), 64:1–64:11.
- A. Selle, J. Su, G. Irving, and R. Fedkiw. 2009. Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction. *IEEE Trans Vis Comp Graph* 15, 2 (2009), 339–350.
- E. Sifakis, S. Marino, and J. Teran. 2008. Globally Coupled Collision Handling Using Volume Preserving Impulses. In *Proc 2008 ACM SIGGRAPH/Eurographics Symp Comp Anim*. 147–153.
- A. Stomakhin, R. Howes, C. Schroeder, and J. Teran. 2012. Energetically consistent invertible elasticity. In *Proc Symp Comp Anim*. 25–32.
- A. Stomakhin, C. Schroeder, L. Choi, J. Teran, and A. Selle. 2013. A Material Point Method for snow simulation. *ACM Trans Graph* 32, 4 (2013), 102:1–102:10.
- S. Sueda, G. Jones, D. Levin, and D. Pai. 2011. Large-scale Dynamic Simulation of Highly Constrained Strands. *ACM Trans Graph* 30, 4 (2011), 39:1–39:10.
- D. Sulsky, Z. Chen, and H. Schreyer. 1994. A particle method for history-dependent materials. *Comp Meth App Mech Eng* 118, 1 (1994), 179–196.
- M. Tang, R. Tong, R. Narain, C. Meng, and D. Manocha. 2013. A GPU-based Streaming Algorithm for High-Resolution Cloth Simulation. *Comp Graph Forum* 32, 7 (2013), 21–30.
- M. Tang, H. Wang, L. Tang, R. Tong, and D. Manocha. 2016. CAMA: Contact-Aware Matrix Assembly with Unified Collision Handling for GPU-based Cloth Simulation. *Comp Graph Forum* 35, 2 (2016), 511–521.
- Y. Teng, D. Levin, and T. Kim. 2016. Eulerian Solid-fluid Coupling. *ACM Trans Graph* 35, 6 (2016), 200:1–200:8.
- D. Terzopoulos and K. Fleischer. 1988. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *SIGGRAPH Comp Graph* 22, 4 (1988), 269–278.
- D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. 1987. Elastically Deformable Models. *SIGGRAPH Comput Graph* 21, 4 (1987), 205–214.
- B. Thomaszewski, M. Wacker, W. Strasser, E. Lyard, C. Luible, P. Volino, M. Kasap, V. Muggeo, and N. Magnenat-Thalmann. 2007. Advanced Topics in Virtual Garment Simulation. In *Eurographics 2007 - Tutorials*.
- H. Wang, F. Hecht, R. Ramamoorthi, and J. O'Brien. 2010. Example-based Wrinkle Synthesis for Clothing Animation. *ACM Trans. Graph.* 29, 4 (2010), 107:1–107:8.
- K. Ward, F. Bertails, T. Kim, S. Marschner, M. Cani, and M. Lin. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Trans Vis Comp Graph* 13, 2 (2007), 213–234.
- K. Wu and C. Yuksel. 2016. Real-time Hair Mesh Simulation. In *ACM SIGGRAPH Symp Int 3D Graph Games*. ACM.
- Y. Yue, B. Smith, C. Batty, C. Zheng, and E. Grinspun. 2015. Continuum foam: a material point method for shear-dependent flows. *ACM Trans Graph* 34, 5 (2015), 160:1–160:20.
- C. Yuksel, J. Kaldor, D. James, and S. Marschner. 2012. Stitch Meshes for Modeling Knitted Clothing with Yarn-level Detail. *ACM Trans Graph* 31, 4 (2012), 37:1–37:12.
- Y. Zhu and R. Bridson. 2005. Animating sand as a fluid. *ACM Trans Graph* 24, 3 (2005), 965–972.
- J. Zurdo, J. Brito, and M. Otaduy. 2013. Animating Wrinkles by Example on Non-Skinned Cloth. *IEEE Trans Vis Comp Grap* 19, 1 (2013), 149–158.