Sag-Free Initialization for Strand-Based Hybrid Hair Simulation

JERRY HSU*, University of Utah & LightSpeed Studios, USA
TONGTONG WANG, LightSpeed Studios, China
ZHERONG PAN, LightSpeed Studios, USA
XIFENG GAO, LightSpeed Studios, USA
CEM YUKSEL, University of Utah & Roblox, USA
KUI WU, LightSpeed Studios, USA

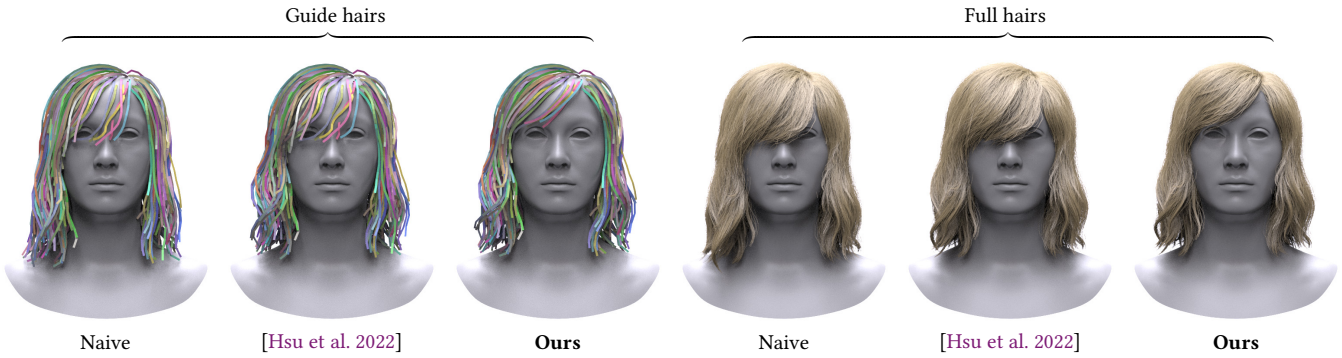

Fig. 1. *On an example wavy hairstyle "Long", we compare three techniques applicable to the Lagrangian/Eulerian hybrid strand-based hair simulation techniques: the naive initialization that treats the given initial shape as the rest shape, the previous sag-free initialization approach [Hsu et al. 2022], and ours. While both naive and previous sag-free initializations lead to hairs sliding down with gravity, our initialization can preserve the given initial hairstyle shape by treating it as the intended shape in static equilibrium under gravity. The left is the guide hairs used for the actual initialization and simulation, while the right shows the full hairs interpolated from the guide hairs.*

Lagrangian/Eulerian hybrid strand-based hair simulation techniques have quickly become a popular approach in VFX and real-time graphics applications. With Lagrangian hair dynamics, the inter-hair contacts are resolved in the Eulerian grid using the continuum method, i.e., the MPM scheme with the granular Drucker–Prager rheology, to avoid expensive collision detection and handling. This fuzzy collision handling makes the authoring process significantly easier. However, although current hair grooming tools provide a wide range of strand-based modeling tools for this simulation approach, the crucial sag-free initialization functionality remains often ignored. Thus, when the simulation starts, gravity would cause any artistic hairstyle to sag and deform into unintended and undesirable shapes.

This paper proposes a novel four-stage sag-free initialization framework to solve stable quasistatic configurations for hybrid strand-based hair dynamic systems. These four stages are split into two global-local pairs. The first one ensures static equilibrium at every Eulerian grid node with additional inequality constraints to prevent stress from exiting the yielding surface. We then derive several associated closed-form solutions in the local stage to compute segment rest lengths, orientations, and particle deformation gradients in parallel. The second global-local step solves along each hair strand to ensure all the bend and twist constraints produce zero net torque on every hair segment, followed by a local step to adjust the rest Darboux vectors to a unit quaternion. We also introduce an essential modification for the Darboux vector to eliminate the ambiguity of the Cosserat rod rest pose in both initialization and simulation. We evaluate our method on a wide range of hairstyles, and our approach can only take a few seconds to minutes to get the rest quasistatic configurations for hundreds of hair strands. Our results show that our method successfully prevents sagging and has minimal impact on the hair motion during simulation.

CCS Concepts: • **Computing methodologies → Physical simulation**.

Additional Key Words and Phrases: Hybrid MPM-Hair, Inverse Modeling, Sag-Free Simulation

ACM Reference Format:
Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. 2023. Sag-Free Initialization for Strand-Based Hybrid Hair Simulation. *ACM Trans. Graph.* 42, 4, Article 1 (August 2023), 14 pages. https://doi.org/10.1145/3592143

1 INTRODUCTION

Hair simulations have long been an active research topic in computer graphics that plays a critical role in the appearance and animation of humans and animals. Among various physically-based

*Part of this work was done when Jerry Hsu was an intern at LightSpeed Studios.

Authors' addresses: Jerry Hsu, jerry060599@gmail.com, University of Utah & LightSpeed Studios, Salt lake city, UT, USA; Tongtong Wang, tongttwang@tencent.com, LightSpeed Studios, Shenzhen, Guangzhou, China; Zherong Pan, zrpan@global.tencent.com, LightSpeed Studios, Seattle, WA, USA; Xifeng Gao, xifgao@global.tencent.com, LightSpeed Studios, Seattle, WA, USA; Cem Yuksel, cem@cemyuksel.com, University of Utah & Roblox, Salt lake city, UT, USA; Kui Wu, kwwu@global.tencent.com, LightSpeed Studios, Los Angeles, CA, USA.

0730-0301/2023/8-ART1 $15.00
https://doi.org/10.1145/3592143

models, Lagrangian/Eulerian hybrid strand-based hair simulation techniques [Fei et al. 2021a; Han et al. 2019; Jiang et al. 2017] have quickly become very popular in both VFX and real-time applications, achieving a superb balance between realism and performance over the sheet- and volumetric-based methods [Koh and Huang 2001; Wu and Yuksel 2016]. These hybrid techniques use Cosserat rod theory [Kugelstadt and Schömer 2016] to model individual strand deformations while relying on the continuum method, e.g., the MPM scheme with the granular Drucker–Prager rheology, to handle contacts between strands. With the prevalence of these simulations, several hair grooming tools, such as Maya XGen, Yeti, and nHair, have been introduced to aid artists in the creation of strand-based visual hair models. However, since these tools do not model any internal or external forces, the designed visual hair models are rarely simulation-ready. Indeed, as soon as the simulation starts, hair strands suffer from sagging (Fig. 1 left), i.e., unintended sliding and deformation due to gravity ruins carefully handcrafted hairstyle.

Although there has been a successful attempt toward sag-free hair modeling [Derouet-Jourdan et al. 2013], their method targets a purely Lagrangian hair simulation system and solves for static equilibrium configuration under Lagrangian contact forces. However, expensive hair-hair contact handling renders it impossible to achieve real-time Lagrangian simulations. On the other hand, extending their method to Lagrangian/Eulerian hybrid hair systems can be non-trivial, since the hybrid method handles the frictional contacts using the drastically different particles using the granular Drucker–Prager rheology MPM scheme.

Recently, Hsu et al. [2022] introduced a general framework for sag-free initialization of deformable simulations. This method utilizes an efficient global-local optimizer to fine-tune rest configurations that cancel out external forces (such as gravity and contacts). While this is a highly-promising and flexible approach that works with various simulation systems, hybrid hair simulations are not among them (Fig. 1 middle). This is owing to the fact that the Cosserat rod model tightly couples point-wise translational, segment-wise rotational, and grid-wise contact degrees of freedom. Simply decomposing this highly non-linear constrained problem into multiple small problems, as in the previous work, is non-trivial. Canceling out the forces and torques for such an intricate system requires substantially restructuring the global-local optimization approach. This further applies to stability-related concerns exacerbated by twisting and particle slippage, for which the prior work presents no solutions. We see this as a great missed opportunity as the fuzzy contact handling of hybrid hair simulations naturally lends it to practical and useful sag-free initialization.

In this paper, we propose a novel four-stage sag-free initialization framework that is the first to solve stable quasistatic configurations for hybrid strand-based hair systems (Fig. 1 right). We first introduce an essential modification for the Darboux vector to eliminate the ambiguity of the Cosserat rod rest pose in both initialization and simulation. We further propose a four-stage sag-free initialization algorithm, which is split into two global-local pairs. The first global-local pair ensures zero net force at every grid node, followed by a second global-local pair that ensures zero net torque at every hair segment. We enforce additional inequality constraints in the first global stage to prevent stress from exiting the yielding surface and then derive several associated closed-form solutions and constraints in the latter local stages to ensure computational efficiency and simulation stability. To cancel out the torque produced by the first global-local step, the second global-local step solves along each hair strand separately to ensure zero net torque on every hair segment, followed by a novel local quaternion adjustment to ensure rest Darboux vectors to be unit quaternions. We show that decomposing the highly non-linear quasi-static problem into multiple small constrained linear problems is possible. And we prove that our four-stage algorithm is guaranteed to yield feasible rest poses with zero net force and torque.

We validate our method on an efficient hybrid GPU hair simulator, which is built off of several existing techniques, including the hybrid MPM method [Han et al. 2019], position-based Cosserat rods [Kugelstadt and Schömer 2016], and the ASFLIP scheme [Fei et al. 2021a]. We highlight our method in a row of benchmarks with various hairstyles. As illustrated in Fig. 1, our method can find sag-free rest configurations for the given wavy hairstyle, for which our simulator yields stable animations.

2 BACKGROUND

This section first briefly reviews related work, and then introduces the hybrid hair simulation method that our approach is based on.

2.1 Related Work

Hair Dynamic Models. Hair simulation is computationally intensive due to the large number of hair strands (with about 100K strands on an average human). In practical applications, engineers typically use a number of representative primitives to simulate hair dynamics rather than simulating every hair. Over the years, a row of simplified hair representations has been proposed to approximate large bundles of hairs at a low cost. These include 2D strips [Koh and Huang 2001], cubic lattice representations [Volino and Magnenat-Thalmann 2006], short hair strips [Guang and Zhiyong 2002], and volumetric representations [Lee et al. 2019; Wu and Yuksel 2016]. These representations can be converted to dense hair models at render time. In comparison to these approaches, a more prevalent modeling technique is to simulate a small number of guiding hair strands and then recover the dense hair model via interpolation. Early works model hair strands using mass-spring chains [Rosenblum et al. 1991] or rigid multi-body chains [Anjyo et al. 1992; Chang et al. 2002], but ghost particles [Umetani et al. 2015] or altitude springs [Selle et al. 2008] are needed to take twisting effects into account for modeling curly hairs. Recently, researchers have devised more physically accurate models of hair strands with bending and twisting, such as the super-helix model [Bertails et al. 2006], Cosserat rod elements (CoRdE) [Spillmann and Teschner 2007], discrete elastic rods (DER) [Bergou et al. 2008], damped exponential time integrator (DETI) model [Michels et al. 2015], and position-based Cosserat rods [Kugelstadt and Schömer 2016].

Inter-Strand Contact Models. With a large number of hairs also comes a massive number of complex inter-strand collisions and contacts. Although the frictional contact between hairs can be resolved using explicit hair geometries [Daviet 2020; Daviet et al. 2011; Kaufman et al. 2014], their exhaustive contact resolution is too costly

for real-time applications. In view of this, Hadap and Magnenat-Thalmann [Hadap and Magnenat-Thalmann 2001] proposed a fluid-inspired, hybrid formulation to resolve collisions on a background Eulerian grid. Their result exhibits a significant speedup using a grid resolution much coarser than the characteristic length of hair strands. This idea has been extended to model volume-preserved gross behavior [McAdams et al. 2009], frictional contacts [Jiang et al. 2017], and two-way coupling with fluids [Fei et al. 2017]. A common issue with hybrid formulation lies in the numerical viscosity caused during particle-grid transfers. Han et al. [2019] proposed a hybrid particle-grid solver to alleviate this issue. Recently, Fei et al. [2021a] introduced the Affine-augmented Separable FLIP (ASFLIP) scheme that utilizes both velocity and positional adjustments to break the positional trap. Leveraging the power of GPU, the latest GPU implementation can simulate hundreds of thousands of MPM particles in real-time [Fei et al. 2021b]. In this work, we build our efficient GPU-based hair simulator by combining ideas from position-based Cosserat rods [Kugelstadt and Schömer 2016], hybrid particle-grid solver using the granular Drucker–Prager rheology for contacts [Han et al. 2019], and the ASFLIP scheme [Fei et al. 2021a]. Our forward hybrid hair simulation framework is capable of simulating thousands of visually convincing hair strands in real-time.

Sag-free Initialization. Artists oftentimes design physical scenarios by assuming they are in static force equilibrium. However, an exact force equilibrium cannot be achieved without carefully tuning the object's material properties and relative rest configurations. Without such fine-tuning, objects will gradually deviate from their designed configurations under gravity, an artifact known as sagging. A myriad of research has been devoted to automatically fine-tuning physical scenarios to achieve force equilibrium. A large class of techniques considers purely elastic objects and optimizes their rest shapes. This method has been applied to design balloons [Skouras et al. 2012], soft characters with actuators [Skouras et al. 2013], deformable 3D printable objects [Chen et al. 2014; Mukherjee et al. 2018; Wang et al. 2015], rod structures [Miguel et al. 2016; Pérez et al. 2015], and garments [Bartle et al. 2016]. To incorporate contact models, Twigg and Kačić-Alesić [2011] formulated the static equilibrium problem as a complex nonlinear global optimization to solve for rest configurations, such as the rest length of springs, for multiple deformable objects modeled as mass-spring systems. Ly et al. [2018] presented a sag-free initialization method for elastic shells with frictional contact handling. The vertices in contact are first constrained as static, and the results are then projected onto a convex frictional cone. Hsu et al. [2022] solve the nonlinear global optimization problem using a two-stage global-local optimization with a generalized formula for handling various dynamic systems, including FEM, MPM, and Position-Based Dynamics (PBD) with frictional contacts. Our proposed approach uses the same decomposition strategy but supports Cosserat rods while constraining the solution to avoid plastic flow for the MPM scheme, among other stability enhancements.

Unlike prior works, our initialization method can handle hybrid Cosserat-MPM hair dynamics. A key challenge for sag-free hair modeling is incorporating thousands of hair-body and hair-hair collisions in the optimization formulation. Existing hair initialization technique relies on exhaustive hair simulation [Lee and Ko 2001] or additional spring forces and constraints [Iben et al. 2019]. It has been shown that, for a single hair strand modeled as a multi-body chain, the static equilibrium state can be found using inverse dynamics [Hadap 2006]. Derouet-Jourdan et al. [2010] proposed an initialization approach for 2D super-helix model. Unfortunately, all of these approaches initialize each strand separately without considering contacts. Derouet-Jourdan et al. [2013] solves for the inverse static equilibrium configuration with Lagrangian contact forces as a well-posed constrained optimization problem. Their method can initialize thousands of hair strands with contacts in the order of a few seconds. Unfortunately, their method is designed for the pure Lagrangian case with explicit frictional contacts. In comparison, our approach also targets sag-free hair initialization but for a completely different forward simulation system that is more suitable for real-time applications. To our best knowledge, we propose the first contact-aware sag-free initialization approach for hybrid Lagrangian/Eulerian hair dynamics.

2.2 Hybrid Cosserat-MPM Hair Simulation

In this section, we describe the hybrid hair simulation method used. Our algorithm is very similar to the prior hybrid work [Han et al. 2019] and takes the Lagrangian point of view and models each hair strand as a Cosserat rod with time integration via extended position-based dynamics (XPBD). The vertices of each hair strand are further used as material particles in conjunction with the MPM method to handle collisions on an auxiliary Eulerian grid. Although such a hybrid framework has been proposed, we introduce essential modifications to allow stable sag-free initialization, which will be presented in the next section.

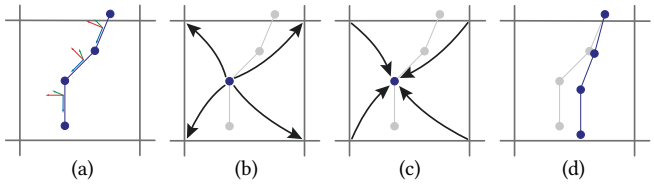

Fig. 2. ***Forward simulation pipeline:*** *(a) integrate Cosserat rod using XPBD scheme to predict per-particle Lagrangian force, (b) compute and transfer particle stress with predicted Lagrangian force to grid nodes and update velocity at each grid node, (c) transfer velocity back to the particle, and (d) advance particles based on the new velocity from G2P.*

We use subscript i as grid node indices, subscript p, p_+, and p_{++} as consecutive particle indices along the hair strand, and subscript s as indices of hair segment between consecutive particles pair p and p_+. We use superscript n as timestep index. $\alpha = \{1, 2, 3\}$ indicates the orthonormal vectors. To model twisting and bending, Cosserat theory attaches a frame with each strand segment, denoted as a quaternion $\mathbf{q}$. $\overline{\mathbf{q}}$ indicates the conjugate of a quaternion $\mathbf{q}$ and $\Im(\mathbf{q})$ indicates the imaginary component of $\mathbf{q}$. $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}_0$ indicate the Darboux vector at the current and rest pose. As shown in Fig. 2, our forward hair simulator uses a time-splitting scheme consisting of the following steps:

2.2.1 Cosserat Rod Integration. We adopt the PBD framework [Kugelstadt and Schömer 2016] to model each hair strand as Cosserat rods and extend it to XPBD [Macklin et al. 2016]. This step is performed independently for each hair strand. Cosserat rods introduce a frame $\mathbf{q}_s$ for each hair segment. As such, the α'th axis of the frame is expressed as $\mathbf{d}_{s,\alpha} = \mathbf{q}_s \mathbf{e}_\alpha \bar{\mathbf{q}}_s$, where $\mathbf{e}_\alpha$ are orthonormal basis. The Cosserat dynamic is modeled with two constraints. The *stretch and shear constraint* $\mathbf{C}_s^{\text{ss}}$ ensures that the rod segment s is nearly length-preserved and normal with the rod's cross-section:

$$\mathbf{C}_s^{\text{ss}}(\mathbf{x}_p, \mathbf{x}_{p_+}, \mathbf{q}_s) = \frac{1}{l_0}(\mathbf{x}_p - \mathbf{x}_{p_+}) - \mathbf{d}_{s,3}, \tag{1}$$

with rest length l_0. To model the bending and twisting behaviors, an additional *bend and twist constraint* $\mathbf{C}_s^{\text{bt}}$ is introduced between two pairs of consecutive particles, quaternion of which denoted as $\mathbf{q}_s$ and $\mathbf{q}_{s_+}$, to penalize the difference between the Darboux vector $\boldsymbol{\Omega}$ at current and rest pose, which is given by:

$$\begin{aligned} \mathbf{C}_s^{\text{bt}}(\mathbf{q}_s, \mathbf{q}_{s_+}) &= \Omega(\mathbf{q}_s, \mathbf{q}_{s_+}) - \phi\boldsymbol{\Omega}_0 \\ \phi &= \begin{cases} +1 & \text{for } \|\boldsymbol{\Omega} - \boldsymbol{\Omega}_0\|^2 \le \|\boldsymbol{\Omega} + \boldsymbol{\Omega}_0\|^2 \\ -1 & \text{for } \|\boldsymbol{\Omega} - \boldsymbol{\Omega}_0\|^2 > \|\boldsymbol{\Omega} + \boldsymbol{\Omega}_0\|^2 \end{cases}, \end{aligned} \tag{2}$$

where ϕ is used to always move $\boldsymbol{\Omega}$ towards the nearest pole, since both $\boldsymbol{\Omega}_0$ and $-\boldsymbol{\Omega}_0$ correspond to the same angle.

The energy potential E is further specified in terms of constraint functions $E_s^\star = \frac{1}{2} k^\star \mathbf{C}_s^{\star T} \mathbf{C}_s^\star$, where $\star$ indicates constraint type and $k^\star$ indicates the corresponding stiffness. We can use the XPBD time integrator to approximately solve each particle's orientation $\mathbf{q}_s^{n+1}$, angular velocity $\boldsymbol{\omega}_s^{n+1}$, temporary position $\tilde{\mathbf{x}}_p^{n+1}$, and temporary velocity $\tilde{\mathbf{v}}_p^{n+1}$ via the optimization:

$$\begin{aligned} \tilde{\mathbf{x}}_p^{n+1}, \tilde{\mathbf{v}}_p^{n+1}, \mathbf{q}_s^{n+1}, \boldsymbol{\omega}_s^{n+1} \approx \underset{\mathbf{x}_p, \mathbf{v}_p, \mathbf{q}_s, \boldsymbol{\omega}_s}{\text{argmin}} & \sum_p \frac{m_p}{2\Delta t^2} \|\mathbf{v}_p - \mathbf{v}_p^n\|^2 + \\ & \sum_s \frac{1}{2\Delta t^2} \|\boldsymbol{\omega}_s - \boldsymbol{\omega}_s^n\|_{\mathbf{I}_s}^2 + \sum_s E_s^{\text{ss}} + \sum_s E_s^{\text{bt}}, \end{aligned} \tag{3}$$

where m_p is the particle mass and $\mathbf{I}_s$is the segment inertial tensor.

To solve the optimization problem, an XPBD scheme is used to calculate the stretch and shear iterants as:

$$\begin{aligned} \Delta\boldsymbol{\lambda}^{\text{ss}} &= \frac{l_0 \left(\mathbf{x}_p - \mathbf{x}_{p_+} + \mathbf{q}_s \mathbf{e}_3 \bar{\mathbf{q}}_s l_0 - \tilde{\alpha}^{\text{ss}} \boldsymbol{\lambda}^{\text{ss}} l_0\right)}{w_p + w_{p_+} + l^2 \left(4 w_s + \tilde{\alpha}^{\text{ss}}\right)}, \\ \Delta\mathbf{x}_p &= +\frac{\Delta\boldsymbol{\lambda}_s}{l_0}, \quad \Delta\mathbf{x}_{p_+} = -\frac{\Delta\boldsymbol{\lambda}_s}{l_0}, \quad \Delta\mathbf{q}_s = -2\Delta\boldsymbol{\lambda}_s \mathbf{q}_s \bar{\mathbf{e}}_3. \end{aligned}$$

and the bend and twist iterants as:

$$\begin{aligned} \Delta\boldsymbol{\lambda}^{\text{bt}} &= \frac{-\left(\bar{\mathbf{q}}_s \mathbf{q}_{s_+} - s\Omega^0\right) - \tilde{\alpha}^{\text{bt}} \boldsymbol{\lambda}^{\text{bt}}}{w_s + w_{s_+} + \tilde{\alpha}^{\text{bt}}}, \\ \Delta\mathbf{q}_s &= \mathbf{q}_{s_+} \overline{\Delta\boldsymbol{\lambda}^{\text{bt}}}, \quad \Delta\mathbf{q}_{s_+} = \mathbf{q}_s \Delta\boldsymbol{\lambda}^{\text{bt}}, \end{aligned}$$

where $\boldsymbol{\lambda}^\star$ is the Lagrange multiplier and $\tilde{\alpha}^\star$ is a block diagonal compliance matrix corresponding to inverse stiffness. At the end of each iteration, the rotation quaternion is normalized. We refer readers to the previous work [Kugelstadt and Schömer 2016] for more details on the XPBD integration of Cosserat rods.

2.2.2 Particle Stress Computation. Instead of explicitly detecting and resolving inter-hair contacts using Lagrangian geometries, we follow the previous hybrid hair method [Fei et al. 2021a; Han et al. 2019] to resolve the contacts on the Eulerian grid with the elastoplastic MPM scheme. Like traditional MPM methods, besides position $\mathbf{x}_p$ and velocity $\mathbf{v}_p$, each particle carries affine coefficients $\mathbf{C}_p$ and deformation gradient $\mathbf{F}_p$. We treat each particle as elastoplastic material and follow standard MPM derivation [Klár et al. 2016] to compute the per-particle stress tensor:

$$\boldsymbol{\sigma}_p^n = \frac{1}{\det(\mathbf{F}_p^n)} \left(\frac{\partial \psi}{\partial \mathbf{F}}(\mathbf{F}_p^n)\right) (\mathbf{F}_p^n)^T, \tag{4}$$

with the material model:

$$\psi(\mathbf{F}) = \mu \operatorname{tr}((\ln \Sigma)^2) + \frac{1}{2}\lambda \operatorname{tr}(\ln \Sigma)^2, \tag{5}$$

where $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^T$ is the singular value decomposition, and μ, λ are the Lamé parameters. It is worth noting that, for real-time performance, we only resolve the contacts on the Eulerian grid, without using explicit Lagrangian geometric correction for collision as in [Han et al. 2019].

2.2.3 Particle-to-Grid (P2G) and Grid Update. After adjusting particle velocities and stress, the momentum $\mathbf{P}_i$ is then transferred from particles to the grid nodes, where explicit force computation is used to update grid velocity:

$$\mathbf{v}_i^n = \mathbf{P}_i^n / \sum_p w_{ip} m_p. \tag{6}$$

2.2.4 Grid-to-Particle (G2P). The final step updates particles' velocity $\mathbf{v}_p^{n+1}$, position $\mathbf{x}_p^{n+1}$, affine coefficient $\mathbf{C}_p^{n+1}$, and deformation gradient $\mathbf{F}_p^n$ using the following ASFLIP scheme [Fei et al. 2021a]:

$$\begin{aligned} \mathbf{v}_p^{n+1} &= \sum_i \mathbf{v}_i^{n+1} w_{ip} + \alpha(\mathbf{v}_p^n - \sum_i \mathbf{v}_i^n w_{ip}) \\ \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \Delta t(\sum_i \mathbf{v}_i^{n+1} w_{ip} + \beta_p \alpha(\mathbf{v}_p^n - \sum_i \mathbf{v}_i^n w_{ip})) \\ \mathbf{C}_p^{n+1} &= \frac{4}{\Delta x^2} \sum_i \mathbf{v}_i^{n+1} (\mathbf{x}_i - \mathbf{x}_p)^T w_{ip} \\ \tilde{\mathbf{F}}_p^{n+1} &= \mathbf{F}_p^n + \Delta t \nabla(\mathbf{v}^n)_p \mathbf{F}_p^n, \end{aligned} \tag{7}$$

where α is a velocity adjustment and β_p is the parameter to avoid the position trap caused by the numerical viscosity during particle-grid transfers. Here, w_{ip} is a standard B-spline quadratic kernel. Finally, we project $\tilde{\mathbf{F}}_p^{n+1}$ to derive $\mathbf{F}_p^{n+1}$ satisfying the following Drucker–Prager yielding condition:

$$c_F \operatorname{tr}(\boldsymbol{\sigma}_p^{n+1}) + \left\| \boldsymbol{\sigma}_p^{n+1} - \frac{\operatorname{tr}(\boldsymbol{\sigma}_p^{n+1})}{d} \right\|_{\mathbf{F}} \le 0, \tag{8}$$

where $d = 2, 3$ as in 2D and 3D, and $c_F \ge 0$ increases with amount of friction between grains. We refer readers to the previous work [Klár et al. 2016] for more details on this step.

3 METHOD

This section introduces our essential modifications to the hybrid hair simulator that enables sag-free initialization and then describes our four-stage sag-free solver with regularization techniques.

3.1 Modifications to the Forward Simulator

We have briefly described the hybrid strand-based hair simulator in Sec. 2.2; however, the existing framework contains ambiguities during computation, which impedes the stability of sag-free Initializations.

3.1.1 Augmented Darboux Vector. The following discrete Darboux vector suggested by Kugelstadt and Schömer [2016]:

$$\mathbf{\Omega} = 2/l_0 \mathfrak{I}(\bar{\mathbf{q}}_s \mathbf{q}_{s+}), \tag{9}$$

only takes the imaginary components and leads to an ambiguity in the sign of the real part of $\bar{\mathbf{q}}_s \mathbf{q}_{s+}$. As such, both the positive and negative values of the real part become valid rest poses. Problematically, this does not represent the same angle but an axis-inverted configuration instead and can be exceptionally prone to creating instantaneous inversions of two axes as the angle between segments approaches zero for highly bent hair. In order to resolve this ambiguity, we modify the definition of the discrete Darboux vector as follows:

$$\mathbf{\Omega} = 2/l_0 (\bar{\mathbf{q}}_s \mathbf{q}_{s+}), \tag{10}$$

dictating the real and imaginary parts to completely remove ambiguity and allow the hair to return to a unique rest shape reliably.

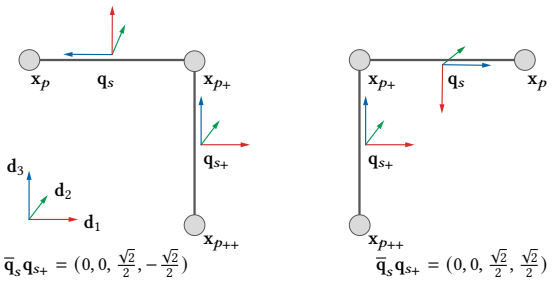

Fig. 3. *Those two configurations have the same discrete Darboux vector under the definition of [Kugelstadt and Schömer 2016] (ignoring the real component), which introduces ambiguity. In practice, when the angle between segments gets smaller, this manifests as an instantaneous inversion of two axes when the segments crossover.*

Fig. 3 demonstrates ambiguous discrete Darboux vector suggested by Kugelstadt and Schömer [2016], which only takes the imaginary components. This leads to an ambiguity in the sign of the real part of $\bar{\mathbf{q}}_s \mathbf{q}_{s+}$, while our augmented Darboux vector resolves this ambiguity and significantly improves the simulation quality of twisting. In summary, there are two ambiguities at play for the discrete Darboux vector. The first is the ambiguity between the equivalent angles of q and $-q$ within the bending and twisting constraint. The second ambiguity lies in the sign of the missing real component in the case of the discrete Darboux vector. Our constraint formulation flips the sign of q to resolve the first ambiguity and we augment the Darboux vector to resolve the second ambiguity.

3.1.2 Lagrangian/Eulerian Coupling. Unlike previous works that directly update position $\mathbf{x}_p^{n+1}$ and velocity $\mathbf{v}_p^{n+1}$ after Cosserat rod integration for the next frame, we use it as the temporary velocity $\tilde{\mathbf{v}}_p^{n+1}$ to predict per-particle Lagrangian force $\mathbf{f}_p^n$ as:

$$\mathbf{f}_p^n = m_p(\tilde{\mathbf{v}}_p^{n+1} - \mathbf{v}_p^n)/\Delta t \,,$$

which is then transferred to the grid for the actual position and velocity update. This modification has two advantages. First, it prevents hairs from exhibiting penetration during the Lagrangian update step. Second, this later allows the contacts which, in this case, correspond to our MPM forces to fully respond to the forces introduced by the Lagrangian step. Note that, due to the relatively small hair radius, we assume the hair segment's angular velocities and rotations do not impact MPM-hybridization. Thus, we updates $\mathbf{q}_p^{n+1}$ and $\boldsymbol{\omega}_p^{n+1}$ directly to their configurations.

In P2G, we also account for additional per-particle Lagrangian forces $\mathbf{f}_p$ computed by the previous step using XPBD. Summing up the contributions, the linear momentum $\mathbf{P}_i^n$ of each grid node takes the following form:

$$\begin{aligned}\mathbf{P}_i^n = \Delta t \sum_p \mathbf{f}_p^n w_{ip} - \Delta t \sum_p V_p^0 \det(\mathbf{F}_p^n) \boldsymbol{\sigma}_p^n \nabla w_{ip} + \\ \sum_p w_{ip} m_p (\mathbf{v}_p^n + \mathbf{C}_p^n (\mathbf{x}_i^n - \mathbf{x}_p^n)) + \Delta t \mathbf{g} \sum_p w_{ip} m_p,\end{aligned} \tag{11}$$

where the first term accounts for Lagrangian forces, the second term is the traction force, the third term uses the affine-corrected velocity [Jiang et al. 2015], and the last term is the gravitational force.

3.2 Sag-free Initialization

Assuming that the initial state of the hair system is quasistatic, i.e., $\mathbf{v}_p^0$, $\boldsymbol{\omega}_p^0$, and $\mathbf{C}_p^0$ are zero, the goal of the sag-free initialization is to solve for stable rest configurations including $\mathbf{q}_p$, $\mathbf{F}_p$, l_0, and $\mathbf{\Omega}_0$, such that all future $\mathbf{v}_p^n, \boldsymbol{\omega}_p^n, \mathbf{v}_i^n$ are zero under gravity and the system remains quasistatic. This is a challenging problem because our hybrid system chains these variables together into a complex nonlinear pipeline spanning multiple systems. Our method follows a similar idea as the two-stage approach introduced in [Hsu et al. 2022] and applies induction on the condition that $\mathbf{v}_p^n = \boldsymbol{\omega}_p^n = \mathbf{v}_i^n = 0$ to avoid deformation. Under this condition, all the variables become intransient, and we can omit the superscript n without confusion. We propose a four-stage approach:

(1) **Global Force Step** computes the Lagrangian force $\mathbf{f}_p$ and the volume-scaled stress $\tilde{\boldsymbol{\sigma}}_p = \det(\mathbf{F}_p)\boldsymbol{\sigma}_p$ that cancel out gravitational forces on each grid node.
(2) **Local Force Step** computes the rest length l_0 for each stretch and shear constraint, quaternion $\mathbf{q}$ for each hair segment, and deformation gradient $\mathbf{F}_p$ for each particle based on the result from the previous global step.
(3) **Global Torque Step** computes the rest Darboux vector $\mathbf{\Omega}_0$ for each bend and twist constraint to cancel out the torque produced by the quaternion $\mathbf{q}$ from the local-force step.
(4) **Local Torque Step** enforces that $\mathbf{\Omega}_0$ correspond to a stable unit quaternion under stability constraints.

We describe each of these four steps below and present our regularization techniques:

3.2.1 Global Force Step. Thanks to our Lagrangian force transfer scheme in Eq. 11, we can utilize the grid momentum to solve for the hair static equilibrium state. In particular, our inductive condition implies the third term in Eq. 11 evaluates to zero, and $\mathbf{P}_i = 0$; otherwise, the G2P step would produce non-zero velocities. Put together, and we have the following expression for each node i:

$$\mathbf{0} = \sum_p \mathbf{f}_p w_{ip} - \sum_p V_p^0 \det(\mathbf{F}_p)\boldsymbol{\sigma}_p \nabla w_{ip} + \mathbf{g}\sum_p w_{ip} m_p \,. \tag{12}$$

The Lagrangian force $\mathbf{f}_p$ on each particle is the summation of stretch and shear forces contributed by each neighboring segment as $\mathbf{f}_p = \sum_{p\in s} \mathbf{f}_s^{\text{ss}}$, where p can be either particle of segment s and stretch and shear force $\mathbf{f}_s^{\text{ss}} = -\partial E_s^{\text{ss}}(\mathbf{x}_p, \mathbf{x}_{p_+})/\partial \mathbf{x}_p$.

Among infinitely many solutions that can lead to quasistatic configurations, we encourage small Lagrangian force $\mathbf{f}_p$ and the volume-scaled stress $\tilde{\boldsymbol{\sigma}}_p$ under the quasistatic constraint since those forces/stresses are related to the user-input material model and should be unaltered as much as possible. More importantly, we cannot accept just any $\tilde{\boldsymbol{\sigma}}_p$ due to the Drucker-Prager yielding condition, and violating that would lead to plastic flow. Unfortunately, directly solving Eq. 12 with the above constraints for rest configurations in E^{ss} and deformation gradient $\mathbf{F}_p$ in $\boldsymbol{\sigma}_p$ is a non-convex optimization problem, due to the Cosserat rod model (Eq. 1, 2) and the hyperelastic material model (Eq. 5). Even worse, the optimization would be large-scale and involves all the hair segments, making a direct numerical optimization prohibitively costly.

We propose a novel convexification technique to boost performance significantly. Our idea is to leave the material model out of the optimization and use the per-segment force $\mathbf{f}_s^{\text{ss}}$ and per particle volume-scaled stress $\tilde{\boldsymbol{\sigma}}_p$ as the decision variables to solve the following Second-Order Conic Programming (SOCP):

$$\begin{aligned} \underset{\tilde{\boldsymbol{\sigma}}_p, \mathbf{f}_s^{\text{ss}}}{\text{argmin}} \; & \sum_s \left\|\mathbf{f}_s^{\text{ss}}\right\|^2 + \alpha \sum_p \|\tilde{\boldsymbol{\sigma}}_p\|^2 \\ \text{s.t.} \quad & \sum_p \mathbf{f}_p w_{ip} - \sum_p V_p^0 \tilde{\boldsymbol{\sigma}}_p \nabla w_{ip} + \mathbf{g}\sum_p w_{ip} m_p = 0 \quad \forall i \\ & c_F \text{tr}(\tilde{\boldsymbol{\sigma}}_p) + \left\| \tilde{\boldsymbol{\sigma}}_p - \frac{\text{tr}(\tilde{\boldsymbol{\sigma}}_p)}{d} \right\|_{\mathbf{F}} \le 0 \quad \forall p, \end{aligned} \tag{13}$$

where $\alpha = 1e{-}2$ is a scaling factor used to bias for contacts. Note that the second constraint is equivalent to the Drucker–Prager yielding condition (Eq. 8), where $\det(\mathbf{F}_p)$ is positive due to our hyperelastic material model. Although Eq. 13 is still large-scale, there exists an efficient splitting algorithm [O'Donoghue et al. 2016] to solve such a SOCP. Similar to the global-local algorithm [Hsu et al. 2022], O'Donoghue et al. [2016] factorizes the global matrix only once and then uses local steps to satisfy each conic constraint. In the Appendix A, we show that Eq. 13 is always feasible.

3.2.2 Local Force Step. Given the per-segment force $\mathbf{f}_s^{\text{ss}}$ from global step and per-particle volume-scaled stress $\tilde{\boldsymbol{\sigma}}_p$, we can solve the inverse problem to recover rest configurations for each hair segment and particle in parallel.

Local step for $\mathbf{f}_s^{ss}$. Given the $\mathbf{f}_s^{\text{ss}}$, the local step aims to find the rest length l_0 for the stretch and shear constraint so that it can produce a matching force. However, only tuning l_0 would expose one degree of freedom, so the force can be only along the segment direction, while $\mathbf{f}_s^{\text{ss}}$ can be arbitrarily given by the optimization in Eq. 13. To match the degree of freedom, we must put the rest orientation $\mathbf{q}_s$ as the additional tunable parameter. To find l_0 and $\mathbf{q}_s$ such that:

$$\mathbf{f}_s^{\text{ss}} = \frac{k^{\text{ss}}}{l_0}\left(\frac{\mathbf{x}_p - \mathbf{x}_{p_+}}{l_0} - \mathbf{d}_{s,3}\right),$$

we notice that $\mathbf{d}_{s,3}$ must be a unit vector, i.e.:

$$\|\mathbf{d}_{s,3}\|^2 = \left\| \frac{l_0 \mathbf{f}_s^{\text{ss}}}{k^{\text{ss}}} - \frac{\mathbf{x}_p - \mathbf{x}_{p_+}}{l_0} \right\|^2 = 1. \tag{14}$$

The above equation is bi-quadratic in l_0 , which could lead to multiple solutions for l_0. We choose the solution that is closest to the current segment length. After l_0 is computed, we solve for $\mathbf{q}_s$ that rotates $\mathbf{e}_{s,3}$ to $\mathbf{d}_{s,3}$. Unfortunately, given $\mathbf{f}_s^{\text{ss}}$, there is no guarantee that Eq. 14 has a solution since the magnitude of compressive forces generated by Eq. 14 is upper-bounded by definition. To ensure the solvability, we could introduce the following hard constraint to Eq. 13 that ensures the discriminant of Eq. 14 is positive:

$$\left(\frac{(\mathbf{x}_p - \mathbf{x}_{p_+})^T \mathbf{f}_s^{\text{ss}}}{(k^{\text{ss}})^2} + 1\right)^2 - 4\frac{||\mathbf{x}_p - \mathbf{x}_{p_+}||^2 ||\mathbf{f}_s^{\text{ss}}||^2}{(k^{\text{ss}})^2} \ge 0 \quad \forall s. \tag{15}$$

Although the above hard constraint is a rotated quadratic cone and can be readily handled by the splitting solver, enforcing it in the global step often leads to no solution depending on the given initial shape. Hence, we do not enforce that constraint in the global step. Instead, we adjust the stiffness in the local step to ensure the discriminant of Eq. 14 is positive and then solve for l_0.

Local step for $\tilde{\boldsymbol{\sigma}}_p$. Given $\tilde{\boldsymbol{\sigma}}_p$, we can recover $\mathbf{F}_p$ based on the hyperelastic material model defined in Eq. 5. Given the SVD decomposition of $\tilde{\boldsymbol{\sigma}}_p = \mathbf{U}\tilde{\Sigma}_p\mathbf{V}$, the $\mathbf{U}, \mathbf{V}$ component must be the same as those of $\mathbf{F}_p$. We analytically solve for the diagonal component of $\ln(\Sigma)$ via the following linear system:

$$\text{diag}(\ln \Sigma) = \left[2V_p^0 \mu \mathbf{I} + V_p^0 \lambda \mathbf{1}\mathbf{1}^T\right]^{-1} \text{diag}(\tilde{\Sigma}_p),$$

where $\mathbf{1}$ is the all-one vector. We can then recover $\mathbf{F}_p$ as $\mathbf{F}_p = \mathbf{U}\exp(\ln\Sigma)\mathbf{V}^T$.

3.2.3 Global Torque Step. While the node force is canceled out in the local-force step, additional torque can be induced by stretch and shear energy E_s^{ss}, denoted as $\boldsymbol{\tau}_s^{\text{ss}}$. Here we use 4D generalized torque associated with an energy $E_s^\star$ defined as: $\boldsymbol{\tau}_s^\star = -\partial E^\star / \mathbf{q}_s$. To prevent the hair segment from rotating, we solve the second global step on each hair strand to achieve torque equilibrium. In particular, we solve the temporary rest orientation $\tilde{\Omega}_{0,s}$ for segment s, such that stretch and shear torque is canceled out by the twist and bend torque for each hair segment s

$$\boldsymbol{\tau}_s^{\text{ss}} + \sum_s \boldsymbol{\tau}_s^{\text{bt}}(\tilde{\Omega}_0) = 0 \quad \forall s, \tag{16}$$

where $\boldsymbol{\tau}_s^{\text{bt}}$ denotes the torque from the bend and twist constraint, which is linearly related to $\tilde{\Omega}_0$ (we ignore s for brevity). The generalized torque we solve in this step is temporary because it does not satisfy the unity constraint: $\|l_0\Omega_0/2\| = 1$. In the next step, we will show how to adjust $\tilde{\Omega}_0$ to yield unit Ω_0 without violating Eq. 16.

3.2.4 Local Torque Step. In the local stage, we then need to adjust $\tilde{\Omega}_0$ to Ω_0 by ensuring $\|l_0\Omega_0/2\| = 1$. Furthermore, this needs to be done while maintaining the zero net torque condition. Interestingly, for the consecutive segments s and s_+, any 4D generalized torque along $\bar{\mathbf{q}}_s\mathbf{q}_{s_+}$ would generate zero 3D torque. Hence, there is always an orthogonal decomposition of $\Omega_0 = \Omega_{0,\parallel} + \Omega_{0,\perp}$ such that $\Omega_{0,\parallel} = t\bar{\mathbf{q}}_s\mathbf{q}_{s_+}$ does not generate any torque for any real scaler t. For the derivation, please refer to Appendix B.

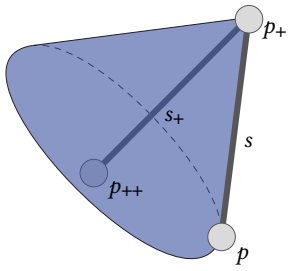

Fig. 4. *We illustrate the cone (blue), in which $s_{0,+}$ must stays to avoid flipping of elements.*

Therefore, we could choose a scalar t such that Ω_0 is a unit quaternion. However, an arbitrary Ω_0 can lead to an unstable dynamic system in two cases. First, C^{bt} relies on a non-smooth jump function ϕ, which changes abruptly when the difference between $\|\Omega - \Omega_0\|$ and $\|\Omega + \Omega_0\|$ is close to zero. To avoid such abrupt change, we need to choose Ω_0 that leaves enough margin between the two values. Thus, we solve for the most stable Ω_0 via the following local optimization:

$$\begin{aligned} &\underset{\|\Omega_0\|=1,t}{\text{argmax}} \ \left|\|\Omega - \Omega_0\|^2 - \|\Omega + \Omega_0\|^2\right| \\ &\text{s.t.} \quad \phi\Omega_0 = \tilde{\Omega}_{0,\perp} + t\bar{\mathbf{q}}_s\mathbf{q}_{s_+}, \end{aligned} \tag{17}$$

where t is an auxiliary variable. A second unstable case happens when the segment s_+ at its current- and rest-pose lies on two sides of segment s. To keep these two poses on the same side of segment s, we need the following additional constraint as shown in Fig. 4:

$$\|\Omega_\perp\| \le \min\left(\Re(\Omega), \frac{2(1-\epsilon_b)}{l_0}\right), \tag{18}$$

for some small constant margin ϵ_b. Unfortunately, given the $\tilde{\Omega}_0$ solved from the global-torque step, no remaining degree of freedom can be used to satisfy Eq. 18. Hence, we enforce the above constraint by raising the bending stiffness locally until Eq. 18 is met. Then, we can use the new $\|\Omega_\perp\|$ to solve Eq. 17 for the final unit quaternion Ω_0. In Appendix B, we derive the constraint in detail, provide a closed-form solution for Eq. 17, and show that it is always feasible given sufficiently large stiffness k^{bt}.

3.2.5 Regularization. In practice, the simple regularization from Eq. 13 can yield unsatisfactory solutions. This is because, for the Lagrangian force, we would like to penalize bending forces more than stretching forces since relying on bending force often leads to undesirable rest configuration as shown in Fig. 5. Regarding stretching force, we prefer tension forces over compression forces because compression forces can result in follow-up bending along indefinite directions. But the stretching stiffness of hair is often several magnitudes larger than the bending stiffness, which results in a relatively small pocket of stability in the case of compression. To alleviate this issue, we design an asymmetric regularization technique for the global force step.

Initial shape Ours w/o regularization Ours w/ regularization

Fig. 5. *Given the initial rod (right), without Lagrangian force regularization, we will get a rest configuration that relies on bending forces to support itself, which is demonstrated by simulating the rod without gravity (middle). Unfortunately, this bent shape is on the saddle point and easy to deform. By penalizing bending forces more than stretching forces, we get a more stable rest shape (right).*

We first decompose $\mathbf{f}_s^{ss}$ into tangent and normal components:

$$\mathbf{f}_s^{ss} = \mathbf{t}_s z_s^t + \mathbf{n}_s z_s^n, \tag{19}$$

where $\mathbf{n}_s = (\mathbf{x}_p - \mathbf{x}_{p_+})/\|\mathbf{x}_p - \mathbf{x}_{p_+}\|$ is the tangent direction and $\mathbf{t}_s$ are the two normal directions. We then decompose z_s^n into positive and negative components via the following constraints:

$$z_s^n = z_s^{n+} + z_s^{n-} \wedge z_s^{n+} \ge 0 \wedge z_s^{n-} \le 0. \tag{20}$$

For similar reasons, it would be better to use hydrostatic stress $\mathbf{z}_p^d$ (diagonal part) over deviatoric stress $\mathbf{z}_d^o$ (off-diagonal part) of $\tilde{\boldsymbol{\sigma}}_p$, so we use the following decomposition:

$$\tilde{\boldsymbol{\sigma}}_p = \mathbf{z}_p^d + \mathbf{z}_p^o. \tag{21}$$

To enforce asymmetric regularization, we add Eq. 19, 20, 21 to Eq. 13 and replace the objective function with:

$$\begin{aligned} w^t \sum_s (z_s^{n+})^2 + w^c \sum_s (z_s^{n-} - z_s^{n-\star})^2 + w^b \sum_s \|z_s^t\|^2 \\ + w^d \sum \|\mathbf{z}_p^d\|^2 + w^o \sum \|\mathbf{z}_p^o\|^2, \end{aligned} \tag{22}$$

where z_s^t, $z_s^{n\pm}$, $\mathbf{z}_p^d$, and $\mathbf{z}_p^o$ are our new decision variables, and we use $w^t = 1$, $w^c = 10^4$, $w^b = 10^4$, $w^d = 10^3$, $w^o = 10^5$ in all our experiments. We also regulate using a bias value $z_s^{n-\star}$. Empirically, we find setting $z_s^{n-\star}$ to be half the weight of a hair strand to yield good results.

4 RESULTS

We implement our sag-free initialization via C++ using the Splitting Conic Solver (SCS) [O'Donoghue 2021] and our forward hair simulation based on a CUDA MPM implementation [Gao et al. 2018]. All timings are measured on a desktop machine with a 3.4 GHz AMD Ryzen 9 5950X CPU and an NVIDIA GeForce RTX 3090 GPU with 24 GB of memory, except forward simulation time and sag-free initialization time for the "Curly wig" hairstyle measured on a desktop machine with a 3.7 GHz AMD Ryzen Threadripper 3970X CPU and an NVIDIA GeForce RTX 3090 GPU with 256 GB of memory.

To verify the practicality of our method, we export four hairstyles from MetaHuman in Unreal Engine. Given the hairstyle (full hairs), we first randomly pick either 256 or 512 hair strands and uniformly sample 16 particles per hair strand as guide hairs for sag-free initialization and forward simulation. Meanwhile, we build an interpolation weight map between guide hairs and full hairs based on their initial relative position. With the weight map, we reconstruct the

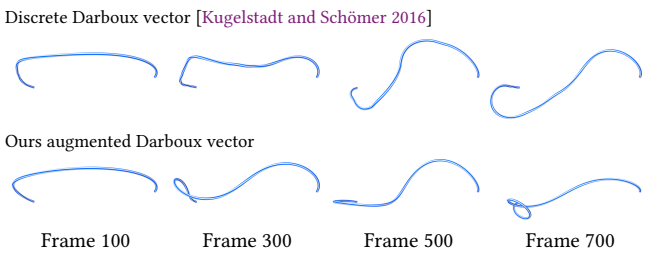

Fig. 6. ***Twisting rod:*** *Comparison between discrete Darboux vector (top) and our augmented Darboux vector (bottom). Our Darboux vector definition uses both the real and imaginary parts to remove ambiguity and improve curvature smoothness.*

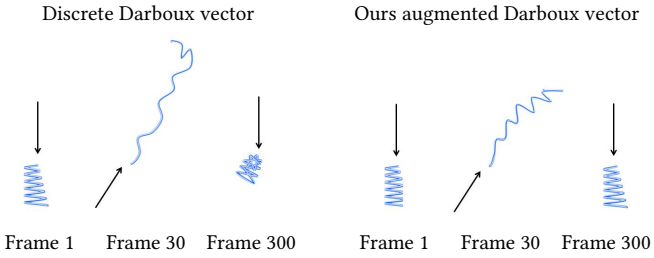

Fig. 7. ***Zig-zag rod:*** *Our method can initialize the sag-free rod based on either the discrete Darboux vector (left) or our augmented Darboux vector (right) in bend and twist constraints. The Zig-Zag rod that uses the discrete Darboux vector cannot recover after being blown by the wind due to the ambiguity in the Darboux vector computation. The arrow indicates the force direction.*

full hair at render time based on the new guide hair positions. Fig. 1 shows an example hairstyle with both guide and full hairs.

In the forward simulation, we use 64 iterations for each XPBD step with a time step size of $\Delta t = 10^{-3}$. Thanks to our GPU implementation and efficient inter-hair contact handling over the grid, it only takes 0.9 ms per timestep, including 64 XPBD iterations and one MPM collision step, which makes the strand-based hair simulation accessible for real-time applications. Note that since our vertices and grid size are relatively small, the reported simulation time is dominated by the CUDA kernel launch overheads and can be subject to further optimizations.

Fig. 6 shows that our augmented Darboux vector can significantly improve the simulation quality of the twisting rod. Furthermore, Fig. 7 demonstrates the ambiguity caused by the discrete Darboux vector for sag-free initialization. The Zig-zag rods are modeled by the bend and twist constraints based on either the discrete Darboux vector (left) or our augmented Darboux vector (right). Although our method can initialize both rods with a sag-free initial pose under gravity, the rod with the discrete Darboux vector cannot recover its initial shape after being blown by the wind due to the ambiguity in the Darboux vector definition.

Fig. 8 demonstrates the importance of enforcing yielding conditions during initializing a cluster of hairs lying on another bunch initially. With either naive initialization or if we solve static equilibrium state without considering Drucker–Prager yielding condition, hair strands slide down after the simulation starts.

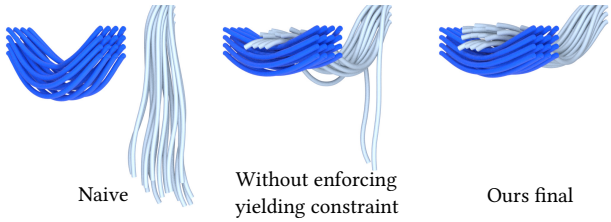

Fig. 8. ***Drape:*** *A cluster of hair strands laying on another cluster initially. With either naive initialization (left) or if we solve the static equilibrium state without considering Drucker–Prager yielding condition (middle), hair strands slide down when the simulation starts, while our initialization (right) can correctly solve inter-hair frictional contacts and preserve the initial shape under the gravity.*

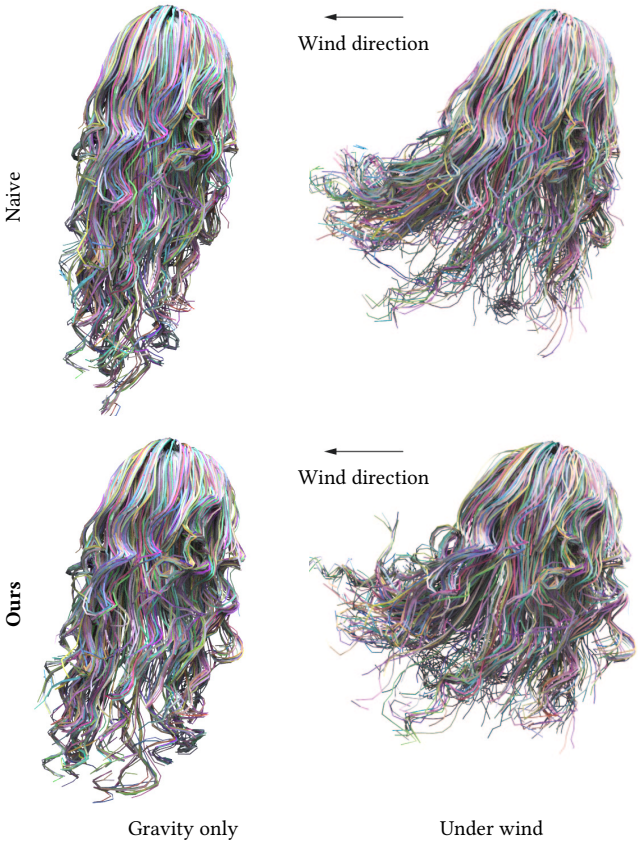

Fig. 9. ***Curly wig:*** *We evaluate our method on the "Curly wig" hairstyle extracted from the same data used in the prior work [Derouet-Jourdan et al. 2013] with a similar number of hair strands. Using the naive initialization (top row), the hairstyle breaks as soon as the simulation starts due to the large sagging artifacts. Our sag-free initialization (bottom row) can preserve the designed hairstyle while having similar hair dynamics under blowing wind (right).*

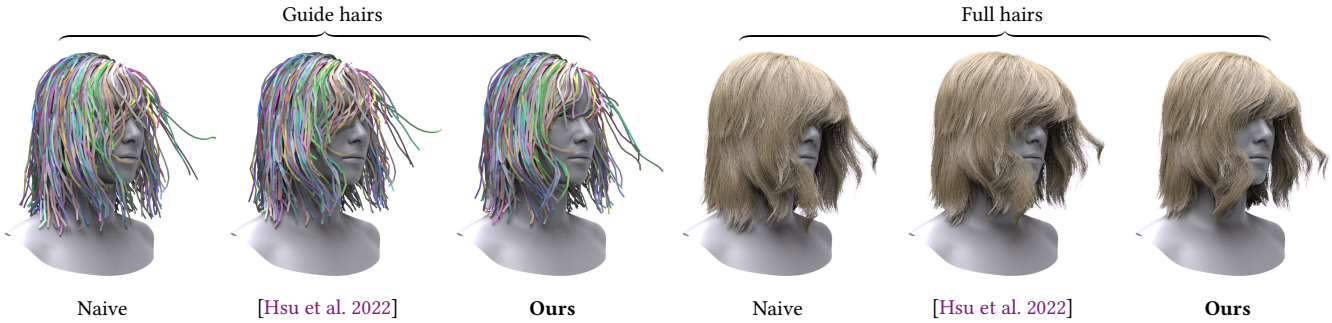

Fig. 10. ***Long:*** *The "Long" hairstyles initialized with the naive method, the previous sag-free initialization approach [Hsu et al. 2022], and ours behave very similarly under the wind from the left. The left is the guide hairs used for the actual initialization and simulation, and the right shows the full hairs interpolated from the guide hairs.*

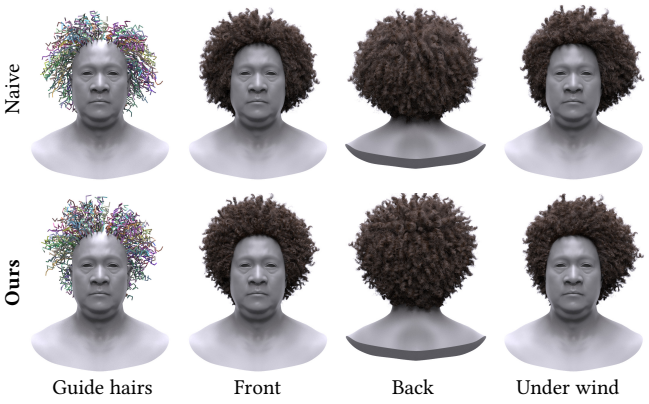

Fig. 11. ***Afro:*** *We show the guide hairs (leftmost), the front (second leftmost), and back (second rightmost) views of full hairs under gravity, where naive initialization causes significant volume loss while ours can preserve the given hair shape. Meanwhile, our result (rightmost) behaves very similarly to that of the naive method under the wind force blowing from the left.*

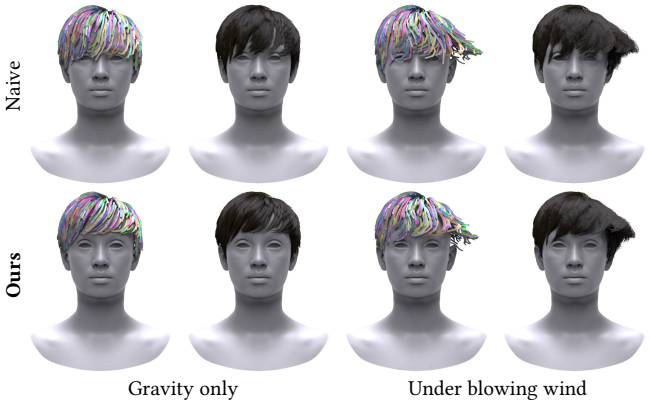

Fig. 12. ***Short:*** *We evaluate our method on one "Short" hairstyle. With the naive initialization (top row), the hairstyles break as soon as the simulation starts. Our sag-free initialization (bottom row) can preserve the designed hairstyles while having similar hair dynamics under blowing wind (right).*

To compare with the prior work [Derouet-Jourdan et al. 2013], we perform sag-free initialization on the same hairstyle (Fig. 9) with a similar number of hair strands (1024). Since their work uses explicit contact handling while ours uses an inter-hair Eulerian grid, there is no clear way to match the exact number of contacts. In our setup, we use 32 particles per strand with a grid size of 128^3, leading to 32555 occupied voxels. Since our system has six unknown variables for each deformation gradient per particle and seven unknown variables for constraints associated with per hair segment, we have a total of 323584 unknown variables to solve for in our initialization step, a much larger problem size than the prior work [Derouet-Jourdan et al. 2013]. Even with a larger system to solve, our method can still successfully initialize this challenging curly hairstyle without sagging in 606.4 seconds. More importantly, it only takes 1.3 ms per step to simulate the hybrid system with 1024 hair strands and 32768 particles, which shows the superiority of our frameworks over previous methods.

To compare with another previous work [Hsu et al. 2022], we use one step of the global-local solver to only solve for net force in the "Long" example. Unfortunately, since the prior method does not consider the net torque and cannot support bend and twist constraints with quaternions and the Darboux vector, the hairstyle initialized via the prior method breaks as soon as the simulation starts (Fig. 1) as it does in the naive initialization. Fig. 10 further demonstrates that all results present similar hair dynamics under the blowing wind.

We further evaluate our method on three additional hairstyles, including Afro (Fig. 11), Short (Fig. 12), and Middle (Fig. 14). Without the correct sag-free initialization, the hairstyles deform once the simulation starts and significantly change the avatars' appearances. Our sag-free initialization preserves the designed hairstyles. "Long" and "Middle" demonstrate our method with long hairstyles that contain large overhangs in both the front and the back. "Afro" demonstrates our approach can also work well for highly intertwined volumetric hair.

Additionally, we use two common human head motions, head-turning (Fig. 13) and dance (Fig. 14), to evaluate the simulation quality of "Long" and "Middle". As shown in Fig. 13, the hairstyle

Fig. 13. ***Head-turning:*** *We evaluate our method on the curly hairstyle with a moderate head-turning motion. Our sag-free initialization (top row) can produce similar visual results as the one with naive initialization (bottom row) does while preserving the bangs.*

Fig. 14. ***Dance:*** *We further evaluate our result on the "Middle" hairstyle with an aggressive dance motion of the head. The Leftmost demonstrates the guide hairs used for initialization and simulation, while the rest are the corresponding full hairs in the dance sequence. The hairs with naive initialization (top row) cannot preserve the initialization and lead to intertwined (second rightmost), while ours (bottom row) has a much better visual quality.*

initialized with our method allows us to preserve both the initial shape under static equilibrium and the dynamic motion under animation. The dynamics under motion remain visually similar, both with and without our initialization. Note that Fig. 14 shows that without our initialization, hairs become universally too soft to preserve the hairstyle and later excessively intertwine with each other due to the aggressive head motions. We list all physical parameters, input parameters, and their resulting initialization metrics in Table 1. We show the percentage of elements where material modifications take place. Generally, hairstyles with curly hair exhibit more significant local stiffening. In particular, any hairstyle or segment positioned horizontally to gravity requires additional stiffening. This is often required when sweeping strands on the forehead or extremely curly hair. We also list the equilibrium condition residual as force (in Newtons) mean squared error (MSE) in Table 1.

To evaluate the scalability of our method, we initialize the "Curly wig" with 128, 256, 512, and 1024 hair strands and 64^3, 64^3, 128^3, and 128^3 grid sizes, respectively. As shown in Fig. 15, the initialization takes 27, 91, 129, and 606 seconds for different numbers of hair strands, where we observe a roughly quadratic scaling in initialization time as the number of hair strands increases under the same grid size. In terms of the forward simulation, thanks to our GPU-friendly implementation, it only takes 0.75, 0.78, 1.16, and 1.30 ms per step, respectively. The forward simulation performance is highly related to the grid size as the MPM update dominates the computation performance, and the Lagrangian rod integration is reasonably fast. It is worth noting that our system can initialize up

Table 1. *Statistics and Performance of Initialization, and physical parameters and their resulting initialization metrics. * is measured on a desktop machine with a 3.7 GHz AMD Ryzen Threadripper 3970X CPU with 256 GB of memory.*

Model	Strand #	Vertex #	Grid size	Valid node #	Initialization	k^{ss}	k^{bt}	Friction coeff.	% stiffened	Force MSE
Drape	32	512	64^3	722	2.4 s	20.0	3.0	0.7	1.2%	$2.3e-4$
Middle	256	4096	64^3	2769	43.4 s	20.0	5.0	0.2	18.9%	$1.4e-3$
Short	256	4096	64^3	1199	63.9 s	20.0	5.0	0.2	20.3%	$1.4e-3$
Afro	256	4096	32^3	2580	83.8 s	20.0	5.0	0.2	7.3%	$1.6e-3$
Long	512	8192	128^3	2705	247.5 s	20.0	5.0	0.2	15.3%	$1.2e-2$
Curly wig*	1024	32768	128^3	32555	606.4 s	100.0	30.0	0.2	56.1%	$7.2e-2$
Knit	1	1024	64^3	2378	2.3 s	60.0	5.0	0.2	63.2%	$1.4e-2$

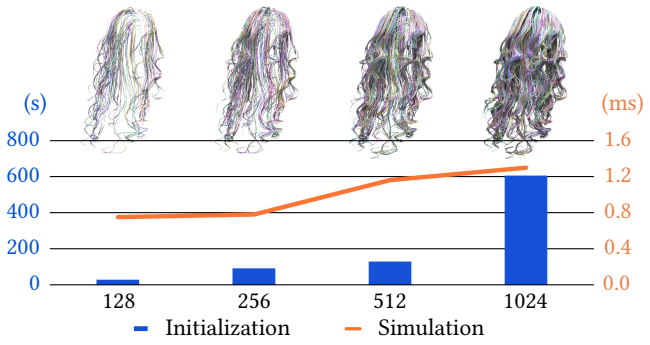

Fig. 15. ***Scalability:*** *We initialize the "Curly wig" hairstyle with 128, 256, 512, and 1024 hair strands, respectively. The blue bars show the initialization time in seconds, while the orange line indicates the per-step simulation performance in ms.*

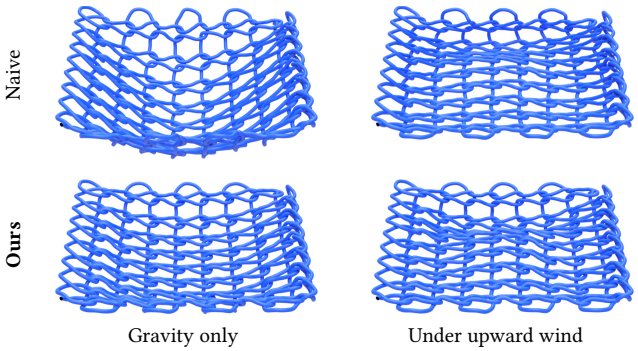

Fig. 16. ***Knit:*** *we initialize a 9×9 knit swatch with one intertwined yarn. Like hairs, the knit sags with naive initialization (top), while our sag-free initialization maintains its shape (bottom).*

to 32 thousand vertices (or 2048 - 4096 strands), which is limited by the direct matrix solver used in our implementation.

Depending on the number of hair strands used, our method can produce valid and stable solutions within several minutes for even our largest examples, as shown in Table 1, where valid nodes indicate the number of free grid nodes impacted by particles. The drape example shown in Fig. 8 only takes around 2 seconds to initialize for 32 guide hairs, and our largest example in Fig. 10 takes around 4 minutes to initialize for 512 guide hairs. In general, we observe a roughly quadratic scaling in initialization time as the number of guide hairs increases. However, as shown in Fig. 10, 512 strands are more than dense enough such that the individual guide hairs themselves could be rendered as hair strands without any bold spots.

Finally, we demonstrate an additional application of our method in Fig. 16, where a single intertwined yarn forms a 9×9 knitted swatch, and the yarn-yarn contacts keep the knit from unraveling. Like other hairstyles, the knit sags with a naive initialization, while our sag-free initialization maintains its original shape when the simulation starts. Note that even without explicit geometric contact detection, our forward simulator can still robustly capture yarn-yarn contacts and avoid yarn pass-through.

5 CONCLUSION

We propose a novel sag-free initialization technique for hybrid hair dynamics. Each hair strand is modeled as a Cosserat rod and inter-hair collisions are handled on an Eulerian grid. To this end, we first proposed a stabilized hair simulator with the augmented Darboux vector. Next, we propose a four-stage algorithm to compute quasistatic configurations and material parameters for both Lagrangian rods and Eulerian particles. Our first two stages compute collision forces and tune stretch and shear forces to cancel out external forces, while our last two stages use twist and bend torques to cancel out stretch and shear torques. We show that our algorithm is well-defined and always yields a solution. Our experiments show that our method can successfully compute the sag-free configuration for various hairstyles.

6 LIMITATIONS AND FUTURE WORK

As future work, we plan to design a joint optimization formulation to achieve sag-free initialization with fewer material parameter modifications. While some material modifications may be unavoidable in specific inputs, the ramifications of minimizing local stiffening during the optimization process remain unexplored.

Furthermore, as we focus heavily on real-time hair simulations, our work begs a natural extension to applications in high-fidelity offline applications. As we implement our algorithm using a direct matrix solver, we are unfortunately limited in the input size, and future work on a matrix-free solver is essential for extending our method to large-scale offline simulations. As part of our scope, we also leave as future work the extensions to wet hair simulations using wet sand material models.

ACKNOWLEDGMENTS

We would like to thank our colleagues from LightSpeed Studios, Fengquan Wang and Dong Li, for their project support. This project was supported in part by NSF grant #1956085.

REFERENCES

Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. 1992. A Simple Method for Extracting the Natural Beauty of Hair. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92)*. Association for Computing Machinery, New York, NY, USA, 111–120.

Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-Driven Pattern Adjustment for Direct 3D Garment Editing. *ACM Trans. Graph.* 35, 4, Article 50 (July 2016), 11 pages.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 63, 12 pages.

Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-Helices for Predicting the Dynamics of Natural Hair. *ACM Trans. Graph.* 25, 3 (jul 2006), 1180–1187.

Johnny T. Chang, Jingyi Jin, and Yizhou Yu. 2002. A Practical Model for Hair Mutual Interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Antonio, Texas) *(SCA '02)*. Association for Computing Machinery, New York, NY, USA, 73–80.

Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. 2014. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Trans. Graph.* 33, 4, Article 95 (July 2014), 11 pages.

Gilles Daviet. 2020. Simple and Scalable Frictional Contacts for Thin Nodal Objects. *ACM Trans. Graph.* 39, 4, Article 61 (aug 2020), 16 pages.

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. *ACM Trans. Graph.* 30, 6 (dec 2011), 1–12.

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot. 2013. Inverse Dynamic Hair Modeling with Frictional Contact. *ACM Trans. Graph.* 32, 6, Article 159 (Nov. 2013), 10 pages.

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot. 2010. Stable Inverse Dynamic Curves. *ACM Trans. Graph.* 29, 6, Article 137 (Dec. 2010), 10 pages.

Yun (Raymond) Fei, Qi Guo, Rundong Wu, Li Huang, and Ming Gao. 2021a. Revisiting Integration in the Material Point Method: A Scheme for Easier Separation and Less Dissipation. *ACM Trans. Graph.* 40, 4, Article 109 (jul 2021), 16 pages.

Yun (Raymond) Fei, Yuhan Huang, and Ming Gao. 2021b. Principles towards Real-Time Simulation of Material Point Method on Modern GPUs. *arXiv preprint* abs/2111.00699 (2021), 19 pages. arXiv:2111.00699 https://arxiv.org/abs/2111.00699

Yun (Raymond) Fei, Henrique Teles Maia, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2017. A Multi-Scale Model for Simulating Liquid-Hair Interactions. *ACM Trans. Graph.* 36, 4, Article 56 (jul 2017), 17 pages.

Ming Gao, Xinlei Wang, Kui Wu, Andre Pradhana, Eftychios Sifakis, Cem Yuksel, and Chenfanfu Jiang. 2018. GPU Optimization of Material Point Methods. *ACM Trans. Graph.* 37, 6, Article 254 (Dec. 2018), 12 pages.

Yang Guang and Huang Zhiyong. 2002. A method of human short hair modeling and real time animation. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.* IEEE, New York, NY, USA, 435–438.

Sunil Hadap. 2006. Oriented Strands: Dynamics of Stiff Multi-Body System. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vienna, Austria) *(SCA '06)*. Eurographics Association, Goslar, DEU, 91–100.

Sunil Hadap and Nadia Magnenat-Thalmann. 2001. Modeling Dynamic Hair as a Continuum. *Computer Graphics Forum* 20, 3 (2001), 329–338.

Xuchen Han, Theodore F. Gast, Qi Guo, Stephanie Wang, Chenfanfu Jiang, and Joseph Teran. 2019. A Hybrid Material Point Method for Frictional Contact with Diverse Materials. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 17 (July 2019), 24 pages.

Jerry Hsu, Nghia Truong, Cem Yuksel, and Kui Wu. 2022. A General Two-Stage Initialization for Sag-Free Deformable Simulations. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2022)* 41, 4, Article 64 (07 2022), 13 pages.

Hayley Iben, Jacob Brooks, and Christopher Bolwyn. 2019. Holding the Shape in Hair Simulation. In *ACM SIGGRAPH 2019 Talks* (Los Angeles, California) *(SIGGRAPH '19)*. ACM, New York, NY, USA, Article 59, 2 pages.

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.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The Affine Particle-in-Cell Method. *ACM Trans. Graph.* 34, 4, Article 51 (July 2015), 10 pages.

Danny M. Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Trans. Graph.* 33, 4, Article 123 (jul 2014), 12 pages.

Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-Prager Elastoplasticity for Sand Animation. *ACM Trans. Graph.* 35, 4, Article 103 (jul 2016), 12 pages.

Chuan Koon Koh and Zhiyong Huang. 2001. A Simple Physics Model to Animate Human Hair Modeled in 2D Strips in Real Time. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation* (Manchester, UK). Springer-Verlag, Berlin, Heidelberg, 127–138.

T. Kugelstadt and E. Schömer. 2016. Position and Orientation Based Cosserat Rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Zurich, Switzerland) *(SCA '16)*. Eurographics Association, Goslar, DEU, 169–178.

Doo-Won Lee and Hyeong-Seok Ko. 2001. Natural Hairstyle Modeling and Animation. *Graph. Models* 63, 2 (March 2001), 67–85.

Minjae Lee, David Hyde, Michael Bao, and Ronald Fedkiw. 2019. A Skinned Tetrahedral Mesh for Hair Animation and Hair-Water Interaction. *IEEE Transactions on Visualization and Computer Graphics* 25, 3 (2019), 1449–1459.

Mickaël Ly, Romain Casati, Florence Bertails-Descoubes, Mélina Skouras, and Laurence Boissieux. 2018. Inverse Elastic Shell Design with Contact and Friction. *ACM Trans. Graph.* 37, 6, Article 201 (dec 2018), 16 pages.

Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: Position-Based Simulation of Compliant Constrained Dynamics. In *Proceedings of the 9th International Conference on Motion in Games* (Burlingame, California) *(MIG '16)*. ACM, New York, NY, USA, 49–54.

Aleka McAdams, Andrew Selle, Kelly Ward, Eftychios Sifakis, and Joseph Teran. 2009. Detail Preserving Continuum Simulation of Straight Hair. *ACM Trans. Graph.* 28, 3, Article 62 (jul 2009), 6 pages.

Dominik L. Michels, J. Paul T. Mueller, and Gerrit A. Sobottka. 2015. A Physically Based Approach to the Accurate Simulation of Stiff Fibers and Stiff Fiber Meshes. *Comput. Graph.* 53, PB (dec 2015), 136–146.

Eder Miguel, Mathias Lepoutre, and Bernd Bickel. 2016. Computational Design of Stable Planar-rod Structures. *ACM Trans. Graph.* 35, 4, Article 86 (July 2016), 11 pages.

Rajaditya Mukherjee, Longhua Wu, and Huamin Wang. 2018. Interactive Two-Way Shape Design of Elastic Bodies. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 1, Article 11 (July 2018), 17 pages.

Brendan O'Donoghue. 2021. Operator Splitting for a Homogeneous Embedding of the Linear Complementarity Problem. *SIAM Journal on Optimization* 31 (August 2021), 1999–2023. Issue 3.

Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. 2016. Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding. *Journal of Optimization Theory and Applications* 169, 3 (June 2016), 1042–1068.

Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and Fabrication of Flexible Rod Meshes. *ACM Trans. Graph.* 34, 4, Article 138 (July 2015), 12 pages.

Robert E. Rosenblum, Wayne E. Carlson, and Edwin Tripp III. 1991. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation* 2, 4 (1991), 141–148.

Andrew Selle, Michael Lentine, and Ronald Fedkiw. 2008. A Mass Spring Model for Hair Simulation. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–11.

Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. 2012. Computational Design of Rubber Balloons. *Comput. Graph. Forum* 31, 2pt4 (May 2012), 835–844.

Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational Design of Actuated Deformable Characters. *ACM Trans. Graph.* 32, 4, Article 82 (July 2013), 10 pages.

J. Spillmann and M. Teschner. 2007. CoRdE: Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '07)*. Eurographics Association, Goslar, DEU, 63–72.

Christopher D. Twigg and Zoran Kačić-Alesić. 2011. Optimization for Sag-Free Simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vancouver, British Columbia, Canada) *(SCA '11)*. ACM, New York, NY, USA, 225–236.

Nobuyuki Umetani, Ryan Schmidt, and Jos Stam. 2015. Position-Based Elastic Rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark) *(SCA '14)*. Eurographics Association, Goslar, DEU, 21–30.

P. Volino and N. Magnenat-Thalmann. 2006. Real-time animation of complex hairstyles. *IEEE Transactions on Visualization and Computer Graphics* 12, 2 (2006), 131–142.

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation Capture and Modeling of Soft Objects. *ACM Trans. Graph.* 34, 4, Article 94 (July 2015), 12 pages.

Kui Wu and Cem Yuksel. 2016. Real-Time Hair Mesh Simulation. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Redmond, Washington) *(I3D '16)*. Association for Computing Machinery, New York, NY, USA, 59–64.

A WELL-DEFINEDNESS OF MPM STEPS

We show that Eq. 13 can always be solved for a valid $\mathbf{F}_p$ and $\mathbf{f}_s^{\text{ss}}$.

Lemma A.1. *Suppose each hair strand is one-end fixed. Given sufficiently large k^{ss}, Eq. 13 with additional constraint Eq. 15 is feasible, which can be solved for $\tilde{\boldsymbol{\sigma}}_p, \mathbf{f}_s^{\text{ss}}$. A valid non-invertible $\mathbf{F}_p$ can then be recovered from $\tilde{\boldsymbol{\sigma}}_p$.*

Proof. First note that Eq. 13 endows a trivial feasible solution of $\tilde{\boldsymbol{\sigma}}_p = 0$ and $\mathbf{f}_p = m_i\mathbf{g}$. Given $\mathbf{f}_p = m_i\mathbf{g}$, we can solve for $\mathbf{f}_s^{\text{ss}}$ via a linear system. The linear system is independent for each hair strand. Let us denote a hair strand as connected by a sequence of N particles with indices:

$$< \alpha(1), \alpha(2), \cdots, \alpha(N) >,$$

then the linear system takes the following form:

$$\begin{pmatrix} -\mathbf{I} & \mathbf{I} & & \\ & \ddots & & \\ & -\mathbf{I} & \mathbf{I} & \\ & & -\mathbf{I} & \end{pmatrix} \begin{pmatrix} \mathbf{f}^{\text{ss}}_{\alpha(1)\alpha(2)} \\ \mathbf{f}^{\text{ss}}_{\alpha(2)\alpha(3)} \\ \vdots \\ \mathbf{f}^{\text{ss}}_{\alpha(N-2)\alpha(N-1)} \\ \mathbf{f}^{\text{ss}}_{\alpha(N-1)\alpha(N)} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_{\alpha(2)} \\ \vdots \\ \mathbf{f}_{\alpha(N-1)} \\ \mathbf{f}_{\alpha(N)} \end{pmatrix},$$

where $\mathbf{f}^{\text{ss}}_{\alpha(1)\alpha(2)}$ indicates the stretch and shear force due to a segment connecting particles $\alpha(1)$ and $\alpha(2)$, the lefthand side is full-rank, and block tridiagonal, so we can always find a solution for $\mathbf{f}_s^{\text{ss}}$. Here we do not need any equation for $\mathbf{f}_{\alpha(1)}$ since the hair strand is one-end fixed. Therefore, Eq. 13's feasible domain is non-empty and must yield a solution $\mathbf{F}_p, \mathbf{f}_s^{\text{ss}}$.

However, to ensure our Local-force step has a solution for l_0, we need to add an additional constraint Eq. 15, with which Eq. 13 might not have a feasible solution. To ensure feasibility, we notice that Eq. 15 must hold when $k^{\text{ss}} \to \infty$, i.e., we can always increase the stiffness parameter to ensure feasibility.

Finally, we show that, given $\tilde{\boldsymbol{\sigma}}_p$, we can utilize the material model to recover $\mathbf{F}_p$. We assume the SVD decomposition of $\tilde{\boldsymbol{\sigma}}_p = \mathbf{U}\tilde{\Sigma}_p\mathbf{V}$, where the $\mathbf{U}, \mathbf{V}$ component must be the same as those of $\mathbf{F}_p$ (see [Klár et al. 2016] for detailed derivation of this result). By direct verification, we have:

$$\begin{aligned} \tilde{\boldsymbol{\sigma}}_p &= -V_p^0 \left(\frac{\partial \psi}{\partial \mathbf{F}_p}(\mathbf{F}_p(\mathbf{x})) \right) \mathbf{F}_p^T \\ \tilde{\Sigma}_p &= V_p^0 \left(2\mu \ln \Sigma + \lambda \text{tr}(\ln \Sigma)\mathbf{I} \right). \end{aligned}$$

We can solve for the diagonal component of $\ln(\Sigma)$ via the following linear system:

$$\text{diag}(\ln \Sigma) = \left[2V_p^0\mu\mathbf{I} + V_p^0\lambda\mathbf{1}\mathbf{1}^T \right]^{-1} \text{diag}(\tilde{\Sigma}_p),$$

where $\mathbf{1}$ is the all-one vector. We can then recover $\mathbf{F}_p$ as:

$$\mathbf{F}_p = \mathbf{U} \exp(\ln \Sigma)\mathbf{V}^T,$$

thus all is proven. □

In practice, we solve Eq. 13 without Eq. 15 and increase the k^{ss} that violates the constraints. This strategy sacrifices optimality for computational speed.

B WELL-DEFINEDNESS OF TORQUE STEPS

For the global-local step of torque, we use 4D generalized torque $\boldsymbol{\tau}_s$. We first reveal connection between generalized torque $\boldsymbol{\tau}_s$ and conventional torque $\tilde{\boldsymbol{\tau}}_s$. After an infinitesimal δt, the orientation is updated by:

$$\frac{\delta t}{2}\boldsymbol{\omega}_s\mathbf{q}_s,$$

so the energy $E_s^\star$ is updated by:

$$-\langle \boldsymbol{\omega}_s\delta t, \tilde{\boldsymbol{\tau}}_s \rangle = \delta E_s^\star = \left\langle \frac{\delta t}{2}\boldsymbol{\omega}_s\mathbf{q}_s, \frac{\partial E_s^\star}{\partial \mathbf{q}_s} \right\rangle.$$

Comparing the two sides of the above equation and we immediately have the following:

$$\tilde{\boldsymbol{\tau}}_s(\boldsymbol{\tau}_s) = \mathfrak{I}(\boldsymbol{\tau}_s\mathbf{q}_s)/2 - \mathfrak{R}(\mathbf{q}_s)\mathfrak{I}(\boldsymbol{\tau}_s). \tag{23}$$

Global-Torque Feasibility. Next, we show that Eq. 16 must have a solution.

Lemma B.1. *Suppose each hair strand is one-end fixed, the system Eq. 16 is always well-defined and feasible.*

Proof. By direct verification, we have the following analytic formula for $\boldsymbol{\tau}_s^{\text{bt}}$ and $\boldsymbol{\tau}_{s+}^{\text{bt}}$:

$$\begin{aligned} \boldsymbol{\tau}_{ss+}^{\text{bt}} &= -k^{\text{bt}} \frac{\mathbf{C}^{\text{bt}}}{\mathbf{q}_s}^T \mathbf{C}^{\text{bt}} = \frac{2k^{\text{bt}}}{l_0}\mathbf{q}_{s+}(\bar{\Omega} - s\bar{\Omega}_{0,ss+}) \\ \boldsymbol{\tau}_{s+s}^{\text{bt}} &= -k^{\text{bt}} \frac{\mathbf{C}^{\text{bt}}}{\mathbf{q}_{s+}}^T \mathbf{C}^{\text{bt}} = \frac{2k^{\text{bt}}}{l_0}\mathbf{q}_s(\Omega - s\Omega_{0,ss+}). \end{aligned}$$

Let us treat $\boldsymbol{\tau}_{ss+}^{\text{bt}}$ and $\boldsymbol{\tau}_{s+s}^{\text{bt}}$ as auxiliary variables, we must have:

$$\left(\frac{\mathbf{C}^{\text{bt}}}{\mathbf{q}_s} \right)^{-T} \boldsymbol{\tau}_{ss+}^{\text{bt}} = \left(\frac{\mathbf{C}^{\text{bt}}}{\mathbf{q}_{s+}} \right)^{-T} \boldsymbol{\tau}_{s+s}^{\text{bt}},$$

and it is easy to see that $\frac{\partial \Omega}{\partial \mathbf{q}_s}$ and $\frac{\partial \Omega}{\partial \mathbf{q}_{s+}}$ are full-ranked, so their inversion is well-defined and we can plug the following substitution:

$$\boldsymbol{\tau}_{s+s}^{\text{bt}} = \left(\frac{\partial \Omega}{\partial \mathbf{q}_{s+}} \right)^T \left(\frac{\partial \Omega}{\partial \mathbf{q}_s} \right)^{-T} \boldsymbol{\tau}_{ss+}^{\text{bt}} \triangleq \mathbf{T}_{ss+}\boldsymbol{\tau}_{ss+}^{\text{bt}},$$

into the first part of Eq. 16 to eliminate $\boldsymbol{\tau}_{s+s}^{\text{bt}}$. After the substitution, we can write the linear system in matrix form as

$$\begin{pmatrix} \mathbf{T}_{\alpha(1)\alpha(2)\alpha(3)} & \mathbf{I} & & \\ & \ddots & & \\ & & \mathbf{T}_{\alpha(N-3)\alpha(N-2)\alpha(N-1)} & \mathbf{I} \\ & & & \mathbf{T}_{\alpha(N-2)\alpha(N-1)\alpha(N)} \end{pmatrix}$$

$$\begin{pmatrix} \boldsymbol{\tau}^{\text{bt}}_{\alpha(1)\alpha(2)\alpha(3)} \\ \vdots \\ \boldsymbol{\tau}^{\text{bt}}_{\alpha(N-3)\alpha(N-2)\alpha(N-1)} \\ \boldsymbol{\tau}^{\text{bt}}_{\alpha(N-2)\alpha(N-1)\alpha(N)} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{\tau}^{\text{ss}}_{\alpha(2)\alpha(3)} \\ \vdots \\ \boldsymbol{\tau}^{\text{ss}}_{\alpha(N-2)\alpha(N-1)} \\ \boldsymbol{\tau}^{\text{ss}}_{\alpha(N-1)\alpha(N)} \end{pmatrix},$$

where the lefthand side is clearly full-ranked. Here the notation $\mathbf{T}_{\alpha(1)\alpha(2)\alpha(3)}$ indicates the generalized torque on the segment connecting particles $\alpha(1)$ and $\alpha(2)$, due to bending and twisting energy of two consecutive segments (one connecting $\alpha(1)$ and $\alpha(2)$ and the other one connectin $\alpha(2)$ and $\alpha(3)$). Note that we do not need an

equation to cancel out $\boldsymbol{\tau}^{\text{ss}}_{\alpha(1)\alpha(2)}$ because the hair strand is one-end fixed. Given the solution $\boldsymbol{\tau}^{\text{bt}}_{ss+}$ and $\boldsymbol{\tau}^{\text{bt}}_{s+s}$, we can recover Ω_0, which is denoted as $\tilde{\Omega}_0$ in our pipeline. This is clearly doable as the coefficient matrices $\frac{\partial \Omega}{\partial \mathbf{q}_s}$ and $\frac{\partial \Omega}{\partial \mathbf{q}_{s+}}$ are full-ranked. Since the two linear systems are equivalent (we use $\boldsymbol{\tau}^{\text{bt}}_s$ and $\boldsymbol{\tau}^{\text{bt}}_{s+}$ as decisions variables, while Eq. 16 use Ω_0), Eq. 16 is solvable. □

Torque Decomposition. Since the conventional torque $\tilde{\boldsymbol{\tau}}^{\text{bt}}_s$ is only 3D, we must have the following decomposition of Ω_0:

Lemma B.2. *Ω_0 can be uniquely decomposed into a parallel component $\Omega_{0,\parallel}$ with $\tilde{\boldsymbol{\tau}}_s(\Omega_{0,\parallel}) = 0$ and an orthogonal component $\Omega_{0,\perp}$ with $\tilde{\boldsymbol{\tau}}_s(\Omega_{0,\perp}) = \tilde{\boldsymbol{\tau}}_s(\Omega_0)$, such that $\Omega_0 = \Omega_{0,\parallel} + \Omega_{0,\perp}$.*

Proof. By direct verification, we can find that:

$$\tilde{\boldsymbol{\tau}}_s(\mathbf{q}_s) = 0.$$

This implies that any scalar multiple of $\mathbf{q}_s$ will generate zero torque. Therefore, we have the following result if Ω_0 is a scalar multiple of $\bar{\mathbf{q}}_s\mathbf{q}_{s_+}$:

$$\begin{aligned}\tilde{\boldsymbol{\tau}}_s(\delta\boldsymbol{\tau}^{\text{bt}}_s) =& -\frac{2k^{\text{bt}}\phi}{l_0}\tilde{\boldsymbol{\tau}}_s(\mathbf{q}_{s_+}\delta\Omega_0)\\ =& -\frac{2k^{\text{bt}}\phi}{l_0}\tilde{\boldsymbol{\tau}}_s(\mathbf{q}_{s_+}\overline{\bar{\mathbf{q}}_s\mathbf{q}_{s_+}}\delta)\\ =& -\frac{2k^{\text{bt}}\phi}{l_0}\tilde{\boldsymbol{\tau}}_s(\mathbf{q}_{s_+}\bar{\mathbf{q}}_{s_+}\mathbf{q}_s\delta)\\ =& -\frac{2k^{\text{bt}}\phi}{l_0}\tilde{\boldsymbol{\tau}}_s(\mathbf{q}_s\delta) = 0.\end{aligned}$$

Now since $\bar{\mathbf{q}}_s\mathbf{q}_{s_+}$ is a unit 4D vector, we immediately have:

$$\Omega_{0,\parallel} = \text{vec}(\bar{\mathbf{q}}_s\mathbf{q}_{s_+})\text{vec}(\bar{\mathbf{q}}_s\mathbf{q}_{s_+})^T\Omega_0 \quad \Omega_{0,\perp} = \Omega_0 - \Omega_{0,\parallel},$$

where $\text{vec}(\bullet)$ converts a quaternion into a 4D vector. □

Torque Constraint Derivation. We first show that our constraint can leave enough margin from the decision boundary of s:

Lemma B.3. *Suppose $\|l_0\Omega_{0,\perp}/2\| \le 1-\epsilon_b < 1$, then $|\|\Omega-\Omega_0\|^2 - \|\Omega+\Omega_0\|^2| \ge \frac{16}{l_0^2}\sqrt{1-(1-\epsilon_b)^2} > 0$.*

Proof. We have:

$$\begin{aligned}&|\|\Omega-\Omega_0\|^2 - \|\Omega+\Omega_0\|| = 4|\text{vec}(\Omega)^T\text{vec}(\Omega_0)|\\ =&4|\text{vec}(\Omega_{0,\parallel})^T\text{vec}(\Omega)| = \frac{8}{l_0}\|\Omega_{0,\parallel}\| = \frac{8}{l_0}\sqrt{\|\Omega_0\|^2 - \|\Omega_{0,\perp}\|^2}\\ =&\frac{16}{l_0^2}\sqrt{1-(1-\epsilon_b)^2} > 0,\end{aligned}$$

where we have used the orthogonal decomposition of Ω_0. □

Next, we show that our constraint can ensure the current- and rest-segment s_+ are on the same side of s.

Lemma B.4. *Suppose $\|\Omega_{0,\perp}\| \le \|\Im(\Omega)\|$, then the current- and rest-segment s_+ are on the same side of the plane passing through s with normal $s \times (s_+ \times s)$ (see Fig. 4).*

Proof. We refer readers to Fig. 4 for the derivation of this proof. In the current configuration, s_+ is rotated around s by the following angle:

$$\sin\frac{1}{2}\angle(s_+, s) = \|\Im(l_0\Omega/2)\|,$$

which is by the definition of our generalized Darboux vector. We then denote the rest vector s_+ of as $s_{0,+}$ and $s_{0,+}$ is rotated around s_+ by the following angle:

$$\begin{aligned}\cos\frac{1}{2}\angle(s_{0,+}, s_+) =& \Re(l_0^2\bar{\Omega}\Omega_0/4)\\ =& l_0^2/4\Re(\bar{\Omega}\Omega_0)\\ =& l_0^2/4\text{vec}(\Omega)^T\text{vec}(\Omega_0)\\ =& l_0/2\|\Omega_{0,\parallel}\|.\end{aligned}$$

In order for $s_{0,+}$ to stay inside the blue cone in the inset, we can let:

$$l_0/2\|\Omega_{0,\perp}\| = \sin\frac{1}{2}\angle(s_{0,+}, s_+) \le \sin\frac{1}{2}\angle(s_+, s) = \|\Im(l_0\Omega/2)\|,$$

and all is proven. □

Local-Torque Solution. In order to solve Eq. 17, we notice from Theorem B.3 that we need to make $\|\Omega_{0,\parallel}\|$ as large as possible, or $\|\Omega_{0,\perp}\|$ as small as possible. Therefore, we first solve the following problem:

$$\underset{\Omega_{0,\perp}}{\text{argmin}}\ \|\Omega_{0,\perp}\| \quad \text{s.t. } s\Omega_0 = t\Omega - \frac{l_0}{2k^{\text{bt}}}\bar{\mathbf{q}}_s\boldsymbol{\tau}^{\text{bt}}_{s_+},$$

which has a closed-form solution

$$\Omega_{0,\perp} = \left[\mathbf{I} - \text{vec}(\bar{\mathbf{q}}_s\mathbf{q}_{s_+})\text{vec}(\bar{\mathbf{q}}_s\mathbf{q}_{s_+})^T\right]\left[-\frac{l_0}{2sk^{\text{bt}}}\bar{\mathbf{q}}_s\boldsymbol{\tau}^{bt}_{s_+}\right].$$

If the second constraint of Eq. 17 is violated, then our problem is infeasible. Otherwise, we use $\Omega_{0,\parallel}$ to normalize Ω_0. When infeasibility is detected we increase k^{bt} until the second constraint is just satisfied. This is always possible because $\Omega_{0,\perp}$ can get arbitrarily small as k^{bt} increases.