HAL
open science

A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics

Gilles Daviet, Florence Bertails-Descoubes, Laurence Boissieux

To cite this version:

Gilles Daviet, Florence Bertails-Descoubes, Laurence Boissieux. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. ACM Transactions on Graphics, 2011, Proceedings of SIGGRAPH Asia 2011, 30 (6), pp.139:1–139:12. 10.1145/2070781.2024173 . hal-00647497

HAL Id: hal-00647497

https://inria.hal.science/hal-00647497

Submitted on 2 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics

Gilles Daviet Florence Bertails-Descoubes Laurence Boissieux

INRIA, Grenoble, France*

Figure 1: *Comparison of the hair collective behavior between (top) real hair motion sequences and (bottom) our corresponding simulations, based on large assemblies of (up to 2,000) individual fibers with massive self-contacts and Coulomb friction. Our model retains typical emerging effects such as transient coherent motions or stick-slip instabilities. See the accompanying video for the full animations.*

Abstract

Dry friction between hair fibers plays a major role in the collective hair dynamic behavior as it accounts for typical nonsmooth features such as stick-slip instabilities. However, due the challenges posed by the modeling of nonsmooth friction, previous mechanical models for hair either neglect friction or use an approximate smooth friction model, thus losing important visual features. In this paper we present a new generic robust solver for capturing Coulomb friction in large assemblies of tightly packed fibers such as hair. Our method is based on an iterative algorithm where each single contact problem is efficiently and robustly solved by introducing a hybrid strategy that combines a new zero-finding formulation of (exact) Coulomb friction together with an analytical solver as a fail-safe. Our global solver turns out to be very robust and highly scalable as it can handle up to a few thousand densely packed fibers subject to tens of thousands frictional contacts at a reasonable computational cost. It can be conveniently combined to any fiber model with various rest shapes, from smooth to curly. Our results, visually validated against real hair motions, depict typical hair collective effects and greatly enhance the realism of standard hair simulators.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—[Animation]

Keywords: Hair simulation, hair contacts, Coulomb friction

Links: DL PDF WEB VIDEO

*e-mail: {Gilles.Daviet,Florence.Descoubes,Laurence.Boissieux}@inria.fr

1 Introduction

Hair is an important and distinctive human feature that has an interesting – often visually pleasing – dynamic behavior. Long hair flowing in air or swinging at the pace of a jogger is so common in the real world that one may think hair material arose the curiosity of scientists a long time ago, and that there is no mystery about it anymore. However, unlike fluids or granular materials which have been extensively studied for more than half a century, fibrous materials such as hair have long been neglected, mainly due to the lack of industrial applications at the time. With the recent advances and impacts of Computer Graphics applications in our everyday life, which stimulates the search for realistic models of virtual humans, the modeling and simulation of the hair dynamics has today become an important scientific challenge. Furthermore, this problem now also draws the attention of cosmetologists who intend to better understand and predict the hair mechanical behavior in order to design more adequate and demanding care products.

Modeling hair dynamics nevertheless remains a largely unsolved issue due to the inherent complexity of hair: human hair is composed of 150,000 individual fibers that tightly interact together, leading to a complex collective behavior. Due to the rough surface of the hair fibers, covered with microscopic scales, *dry friction* substantially alters contacts at the fiber level, and consequently greatly influences the hair dynamics at the *macroscopic* level. As illustrated in Figure 2 and in the accompanying video, we have identified three major hair visual features that directly emerge from those nonsmooth frictional contacts occurring at the fiber level:

1. Typical stick-slip instabilities during motion;
2. The spontaneous splitting of hair into multiple untidy wisps and "flyers" during strong motion *vs.* its spontaneous grouping into a few globally coherent locks during gentle motion;
3. The appearance of complex nonsmooth hair patterns that can remain perfectly still at the end of the motion.

We claim that accounting for these phenomena is essential for producing realistic and compelling hair animations.

Figure 2: *Three important features of the real hair collective behavior, emerging from nonsmooth friction: (1) Stick-slip instabilities, especially visible here between the hair and the shoulders; (2) Spontaneous splitting of hair into thin wisps and "flyers" during strong motion (left)* vs. *spontaneous grouping of hair into a few coherent locks during gentle motion (middle and right); (3) Appearance of complex, nonsmooth hair patterns that may remain stable at rest. Video courtesy of Sylvain Paris and Tilke Judd, MIT 2007.*

The first dynamical hair models in Computer Graphics were mostly designed at a coarse level, failing to reproduce the fine-scale details of real hair motion. More recently, simulating hair dynamics at the fiber level has gained some interest and resulted in much more detailed animations [Selle et al. 2008; McAdams et al. 2009]. However, nonsmooth friction was not considered and the three important visual effects mentioned above could not be accurately captured. More generally, in previous hair simulators the handling of hair self-contacts is approximated too much, and typical nonsmooth effects are lost. Additionally, the hair volume may not be properly preserved and simulations may suffer from instabilities when hair comes to rest.

In the same way as Selle et al. [2008] we think that simulating hair at the fiber lever is a promising research direction, and certainly a good start to better understand and capture the complexity of hair motion. In this paper, we show that dealing with nonsmooth frictional contact at the fiber level represents a both achievable and encouraging step towards the realistic modeling of hair dynamics. More precisely, we introduce a robust hair frictional contact solver that allows, for the first time, to capture all the desirable hair macroscopic features listed above. Our method allows us to greatly enhance the realism of standard hair simulators.

2 Prior Work

Simulating the motion of human hair has become an important research topic in Computer Graphics since the 90's. In parallel, inspired by some works performed in Computational Mechanics, multiple methods were designed to robustly handle massive contacts, especially in large assemblies of rigid bodies. We review here the works in hair simulation and contact handling that are the most relevant to our specific problem, and show how our approach actually bridges the gap between these two (disconnected) areas.

Modeling hair dynamics Due to performance limitations, the first works that attempted to simulate hair dynamics mostly neglected hair self-interactions [Rosenblum et al. 1991; Anjyo et al. 1992], or processed them at a coarse level between a small number of predefined interacting wisps using penalty forces [Plante et al. 2001; Choe et al. 2005; Bertails et al. 2006]. Alternatively, Hadap and Thalmann [2001] proposed a macroscopic model of the hair medium based on a fluid solver, observing that some fluid properties, such as incompressibility, could mimic the hair collective behavior well. Though interesting, their approach fails to capture the discontinuities emerging from large hair motions. Thanks to the design of realistic, robust and fast primitives for thin elastic rods [Pai 2002; Bertails et al. 2006; Hadap 2006; Spillmann and Teschner 2007; Bergou et al. 2008; Selle et al. 2008; Bergou et al. 2010], some approaches have been developed at the fiber lever in order to gain realism in hair simulations. Selle et al. [2008] designed an efficient mass-spring model for an individual fiber, allowing them to simulate up to 10,000 fibers in a reasonable computational time (from a few minutes up to one hour per frame) on a quad-core architecture. Unfortunately, many self-contacts were ignored, causing the fibers to penetrate each other, and thus failing to preserve the hair volume. Moreover, typical emerging effects such as the spontaneous sticking of neighboring fibers due to friction were modeled using an ad-hoc, procedural "stiction" model. To resolve these issues while still retaining some discontinuous details in the simulations, McAdams et al. [2009] recently proposed a hybrid Eulerian/Lagrangian hair model combining a fluid model together with the explicit treatment of fiber self-contacts. With this approach, the hair volume is properly preserved while detailed interactions at the fiber level yield nice visual effects. However, nonsmooth effects due to static friction, which play a major role in hair dynamics, are not captured. This is especially visible when hair comes to rest: hair fibers smoothly keep on sliding on each other instead of remaining stuck on top of each other.

To the best of our knowledge, other approaches in Computer Graphics that have attempted to model fiber assemblies have relied on penalty forces for modeling contact [Kaldor et al. 2008; Harmon et al. 2009; Ward et al. 2010; Kaldor et al. 2010], sometimes combined with an explicit treatment of static friction that is not accurate enough to automatically capture the subtle effects emerging from the true Coulomb friction model.

Modeling robust frictional contact The most robust and accurate methods for simulating frictional contact rely on constraint-based algorithms. They have been introduced in Computer Graphics for solving the problem of rigid body stacking [Baraff 1994; Kaufman et al. 2005; Erleben 2007], and are now starting to be extended to soft materials [Kaufman et al. 2008; Otaduy et al. 2009].

Inspired by the works performed for years in the simulation of granular materials by the computational mechanics community [Alart and Curnier 1991; Moreau 1994; Jourdan et al. 1998], Bertails et al. [2011] recently applied a nonsmooth Newton solver based on the Alart-Curnier formulation for resolving contact with exact Coulomb friction in fiber assemblies. Their analysis of convergence shows that the solver works well for systems with a limited number of contacts, but often fails to converge in the case of over-constrained systems such as densely packed fibers. In contrast, relying on a new hybrid method for Coulomb friction within a splitting algorithm, we manage here to tackle challenging hair scenarios involving thousands of tightly packed hair fibers.

3 Contributions

We focus on the *highly realistic* and *robust* simulation of long hair with various natural shapes (smooth to curly) subject to strong head motions. The hair material is modeled as a large assembly of thin elastic rods (up to 2,000 rods) that interact with each other. For sta-

bility reasons, we chose the super-helix model [Bertails et al. 2006] as our fiber model, but the method presented here is generic and can be combined with any fiber model in the literature.

We claim that high realism can only be achieved by properly taking into account nonsmooth friction in hair-body and hair-hair contacts, while robustness requires the design of some well-grounded and fail-safe numerical solver. Our contributions are twofold:

- We introduce a new functional formulation for (exact) Coulomb friction that extends the Fischer-Burmeister formulation for real complementarity problems [Fischer 1992] to our conical complementarity problem. Combined with a fail-safe based on an enumerative solver for a single contact, our new hybrid solver manages to always find a solution to the local problems arising in the Gauss-Seidel loop when it exists. As a result, our global solver turns out to be robust as well as highly scalable, allowing us to simulate the realistic motion of hundreds to a few thousand fibers subject to massive (up to tens of thousands) self-contacts in a reasonable computational time – a few minutes per frame – on a standard quad-core desktop computer.

- We thoroughly evaluate our method compared to previous approaches in terms of robustness and computational efficiency, and carefully validate our simulation results against real hair experiments. To the best of our knowledge, our method is the first, with respect to both Computer Graphics and Computational Mechanics fields, that allows for the simulation of thousands of tightly packed fibers subject to Coulomb friction, leading to an accurate capture of typical hair collective effects.

We first recall the Coulomb friction law and introduce the main notation in Section 4. We then describe in Section 5 our new zero-finding formulation for exact Coulomb friction and provide our full iterative hybrid solver for frictional contact in Section 6. Finally, results are presented and validated in Section 7 before concluding.

4 Coulomb Friction in Dynamical Systems

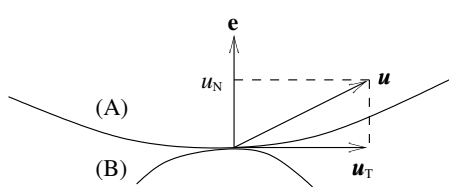

Figure 3: *Local contact basis, with normal and tangent subspaces.*

In this section we mainly follow the concepts and notation introduced in [Bertails-Descoubes et al. 2011] to formulate our Coulomb friction problem. We assume that one contact always occurs between exactly two bodies, and that the surface of contacting objects is sufficiently smooth so that we can define the contact normal $\mathbf{e}$, from which we build a local basis (see Figure 3). We can then express in this basis the relative velocity $\boldsymbol{u}$ between the two bodies, and the force $\boldsymbol{r}$ at the contact point. In the remainder of this paper, we will assume that the contact basis has been rotated so that $\mathbf{e} = (1,0,0)^\top$.

A classical way to describe Coulomb's frictional law in dimension d is to introduce the second-order cone (**SOC**) K_μ. It is defined as

$$K_\mu := \left\{ \mathbf{x} = [\mathrm{x_N}, \mathbf{x}_\mathrm{T}] \in \mathbb{R} \times \mathbb{R}^{d-1},\, \mu \mathrm{x_N} \geq \|\mathbf{x}_\mathrm{T}\| \right\},$$

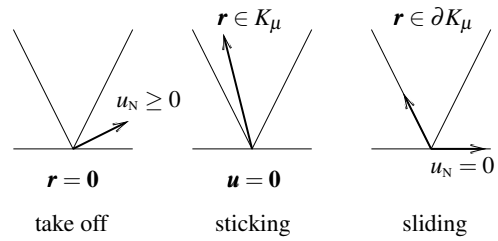

Figure 4: *Disjunctive formulation of the Coulomb law.*

where $\mathrm{x_N}$ denotes the (scalar) normal component of vector $\mathbf{x}$, and $\mathbf{x}_\mathrm{T}$ its tangential (vectorial) component.

The friction law can then be written in a disjunctive manner, as illustrated in Figure 4,

- Take-off case: $\boldsymbol{r} = \mathbf{0}$ and $u_\mathrm{N} \geq 0$
- Sticking case: $\boldsymbol{r} \in K_\mu$ and $\boldsymbol{u} = \mathbf{0}$
- Sliding case: $\boldsymbol{r} \in \partial K_\mu$, $u_\mathrm{N} = 0$ and $\exists \alpha \in \mathbb{R}_+^*$, $\boldsymbol{u}_\mathrm{T} = -\alpha \boldsymbol{r}_\mathrm{T}$.

Let $\mathscr{C}(\mu)$ be the set of vectors $(\boldsymbol{r}, \boldsymbol{u})$ in $\mathbb{R}^d \times \mathbb{R}^d$ satisfying Coulomb's law. We consider a smooth dynamical system that has been spatially discretized so that it can be fully described by m generalized coordinates $\boldsymbol{q} \in \mathbb{R}^m$, their time derivatives $\dot{\boldsymbol{q}} = \frac{d\boldsymbol{q}}{dt}$, and the equation of motion

$$\mathscr{M}(\boldsymbol{q}) \frac{d\dot{\boldsymbol{q}}}{dt} + \mathscr{F}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{0}, \tag{1}$$

where $\mathscr{M}$ is the mass matrix of the system and $\mathscr{F}$ collects the external and internal forces as well as the nonlinear inertial terms.

Let us now assume that the system is subject to n frictional contacts. To simplify notation, the n relative velocities $\boldsymbol{u}^i$ and contact forces $\boldsymbol{r}^i$ are now collected into $\boldsymbol{u}$ and $\boldsymbol{r}$, respectively. We define $\mathscr{H}(\boldsymbol{q}) := \frac{\partial \mathbf{u}}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q})$ the gradient matrix of dimension (nd, m) relating the relative spatial velocities at contact points $\boldsymbol{u}$ to the generalized velocities $\dot{\boldsymbol{q}}$. Let $\boldsymbol{u}_\mathrm{f}$ be the value of $\boldsymbol{u}$ when $\dot{\boldsymbol{q}} = \mathbf{0}$, which can be non-zero in the case of forced motion. The nonsmooth system including contact forces $\boldsymbol{r}$ can be written as

$$\begin{cases} \mathscr{M}(\boldsymbol{q}) \frac{d\dot{\boldsymbol{q}}}{dt} + \mathscr{F}(t, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathscr{H}(\boldsymbol{q})^\top \boldsymbol{r} \\ \boldsymbol{u} = \mathscr{H}(\boldsymbol{q}) \dot{\boldsymbol{q}} + \boldsymbol{u}_\mathrm{f}(t, \boldsymbol{q}) \\ \forall i = 1 \ldots n, (\boldsymbol{r}^i, \boldsymbol{u}^i) \in \mathscr{C}(\mu^i). \end{cases} \tag{2}$$

The notation $\frac{d\dot{\boldsymbol{q}}}{dt}$ is actually misused as the inclusion of contacts implies that we might observe jumps in the generalized velocities $\dot{\boldsymbol{q}}$. To circumvent this difficulty, we follow Moreau's time-stepping scheme and integrate the equations over an arbitrary timestep δt [Moreau 1988; Acary and Brogliato 2008]. We thus get a formulation over mechanical impulses instead of forces, and a velocity jump instead of an acceleration.

Let us set $\mathbf{r} := \int_{\delta t} \boldsymbol{r}\, dt$, and let $\mathbf{v}$ and $\mathbf{u}$ be the discrete approximations of the generalized velocity and of the relative velocity, respectively, at the end of the timestep. The constraint-free dynamics (1) is discretized using a θ-scheme and can be written at a given timestep as $\mathbf{M}\mathbf{v} = \mathbf{f}$. We further assume that $\mathscr{H}(\boldsymbol{q})$ and $\boldsymbol{u}_\mathrm{f}(t, \boldsymbol{q})$ are almost constant during the timestep, and approximate them with $\mathbf{H}$ and $\mathbf{u}_\mathrm{f}$, respectively. The discretized velocity/impulse formulation of (2), with unknowns $(\mathbf{v}, \mathbf{u}, \mathbf{r})$, then reads

$$\begin{cases} \mathbf{M}\mathbf{v} - \mathbf{f} = \mathbf{H}^\top \mathbf{r} \\ \mathbf{u} = \mathbf{H}\mathbf{v} + \mathbf{u}_\mathrm{f} \\ \forall i = 1 \ldots n, (\mathbf{r}^i, \mathbf{u}^i) \in \mathscr{C}(\mu^i). \end{cases} \tag{3}$$

To eliminate $\mathbf{v}$ from (3), we introduce the Delassus operator $\mathbf{W} := \mathbf{H}\mathbf{M}^{-1}\mathbf{H}^\top$ and set $\mathbf{b} := \mathbf{u}_\mathrm{f} + \mathbf{H}\mathbf{M}^{-1}\mathbf{f}$. We finally obtain the following reduced *one-step problem*

$$\begin{cases} \mathbf{W}\mathbf{r} + \mathbf{b} = \mathbf{u} \\ \forall i \left(\mathbf{r}^i, \mathbf{u}^i\right) \in \mathscr{C}(\mu^i). \end{cases} \tag{4}$$

Note that (4) is a *nonsmooth* problem due to the inclusion of $(\mathbf{r}^i, \mathbf{u}^i)$ in the set $\mathscr{C}(\mu^i)$. Furthermore, if $\operatorname{rank}\mathbf{H} < nd$, then $\mathbf{W}$ is not strongly positive-definite and the existence of solutions is not guaranteed. One typical example is the Painlevé paradox, described for instance in [Baraff 1991]. However, in our case where we simulate free hairstyles in absence of strong external constraints (such as a hand compressing a hair wisp), our friction problem is likely to possess a solution most of the time [Acary et al. 2011].

5 A New Functional Formulation for Exact Coulomb Friction

In this section, we introduce a new functional formulation of the Coulomb law that has sufficient smoothness properties for being conveniently resolved using a nonsmooth Newton algorithm. The key idea consists in splitting the Coulomb friction law into two parts: a frictionless law ($\mu^i = 0$), reformulated as a *linear complementarity problem* (**LCP**), and a pure frictional law ($\mu^i > 0$), reformulated as a *second-order cone complementarity problem* (**SOCCP**). Then, using the equivalence between an **LCP** and the Fischer functional formulation [Fischer 1992], and respectively between a **SOCCP** and a modified Fischer functional formulation [Fukushima et al. 2002], we manage to derive our new root-finding formulation of the Coulomb friction law that proves to be very convenient in practice.

5.1 A New Complementarity formulation

The disjunctive formulation of the Coulomb law implies that $\mathbf{r}^i$ lies inside the cone K_{μ^i} and $\mathbf{u}^i$ in the half-space $\mathbb{R}_+ \times \mathbb{R}^{d-1}$. Actually, this can be expressed as a cone complementarity problem by mapping this half-space to the dual cone of K_{μ^i}, which is $K_{\frac{1}{\mu^i}}$ (see Figure 5, (a) and (b)). Such a change of variables has been described by De Saxcé and Feng [1998],

$$\widetilde{\mathbf{u}}^i := \mathbf{u}^i + \mu^i \left\|\mathbf{u}^i_\mathrm{T}\right\| \mathbf{e}, \tag{5}$$

and leads to the following expression of the Coulomb law[1],

$$\left(\mathbf{r}^i, \mathbf{u}^i\right) \in \mathscr{C}(\mu^i) \iff K_{\frac{1}{\mu^i}} \ni \widetilde{\mathbf{u}}^i \perp \mathbf{r}^i \in K_{\mu^i}. \tag{6}$$

For frictionless contacts ($\mu^i = 0$), the complementarity problem (6) simply boils down to an **LCP** on the normal parts of $\mathbf{u}$ and $\mathbf{r}$.

Pure frictional contacts Let us now consider pure frictional contacts ($\mu^i > 0$). In order to fully benefit from the nonsmooth optimization theory, we would like to get a complementarity relationship similar to (6), but such that both variables belong to the *same* cone. Our idea consists in applying a second change of variables to our unknowns,

$$\hat{\mathbf{r}}^i := (\mu^i \mathrm{r}^i_\mathrm{N}, \mathbf{r}^i_\mathrm{T}) \qquad \text{and} \qquad \hat{\mathbf{u}}^i := (\rho \tilde{\mathrm{u}}^i_\mathrm{N}, \rho \mu^i \widetilde{\mathbf{u}}^i_\mathrm{T}) \tag{7}$$

where ρ is a positive constant which allows for $\hat{\mathbf{u}}$ and $\hat{\mathbf{r}}$ to share the same unit (we typically choose $\rho := 1$ kg). Our new variables $\hat{\mathbf{u}}^i$ and $\hat{\mathbf{r}}^i$ now belong to the d-dimensional **SOC** K,

$$\mathrm{K} := K_{\mu=1} = \left\{ [\mathrm{x}_\mathrm{N}, \mathbf{x}_\mathrm{T}] \in \mathbb{R} \times \mathbb{R}^{d-1}, \mathrm{x}_\mathrm{N} \geq \|\mathbf{x}_\mathrm{T}\| \right\}.$$

Note that K is independent of μ, and is a self-dual cone. See Figure 5 (c) for a graphical illustration. As a result, we can express the Coulomb friction law for pure frictional contacts as a **SOCCP**,

$$\left(\mathbf{r}^i, \mathbf{u}^i\right) \in \mathscr{C}(\mu^i) \iff \mathrm{K} \ni \hat{\mathbf{u}}^i \perp \hat{\mathbf{r}}^i \in \mathrm{K}.$$

Full complementarity formulation Finally, with our new formulation for pure frictional contacts, the full Coulomb law can be expressed as

$$\left(\mathbf{r}^i, \mathbf{u}^i\right) \in \mathscr{C}(\mu^i) \iff \begin{cases} \begin{cases} 0 \leq \mathrm{u}^i_\mathrm{N} \perp \mathrm{r}^i_\mathrm{N} \geq 0 \\ \mathbf{r}^i_\mathrm{T} = \mathbf{0} \end{cases} & \text{if } \mu^i = 0 \quad (8a) \\ \mathrm{K} \ni \hat{\mathbf{u}}^i \perp \hat{\mathbf{r}}^i \in \mathrm{K} & \text{if } \mu^i > 0\,. \quad (8b) \end{cases}$$

We note that Kanno et al. [2004] provided a *second-order cone linear complementarity problem* (**SOCLCP**) formulation for Coulomb's friction law, but they had to introduce an auxilliary variable $\lambda \in \mathbb{R}$ and express their problem on $\mathrm{K}_S := \mathbb{R}_+ \times \mathrm{K}$. More recently, Anitescu and Tasora [2008] and Zhang *et al.* [2011] also proposed **SOCLCP** formulations, but both of them only considered relaxed versions of the Coulomb law.

5.2 Modified Fischer-Burmeister function (MFB)

For the sake of simplicity, we consider here the case of a single frictional contact, involving the **SOC** K. The generalization to a product of **SOC**s $\mathrm{K} \times \ldots \times \mathrm{K}$ is straightforward.

It is well-known in nonsmooth optimization theory [Fischer 1992] that non-linear complementarity problems on $\mathbb{R}$ (**NCP**) can be formulated as the following root-finding problem,

$$0 \leq x \perp y \geq 0 \iff 0 = f^\mathrm{FB}(x,y) := x + y - \sqrt{x^2 + y^2}, \tag{9}$$

where f^FB is called the *Fischer-Burmeister* complementarity function. In the field of contact mechanics, it has already been used to solve **NCP** formulations of various frictional laws (*e.g.* [Silcowitz et al. 2009]). We can directly use this function to solve the frictionless case (8a) of our formulation, which is an **LCP**.

Now, in the frictional case, we would like to have an analogous formulation for our **SOCCP** (8b), that is, find a complementarity function f that is suitable for *conical* constraints,

$$\mathrm{K} \ni \hat{\mathbf{u}} \perp \hat{\mathbf{r}} \in \mathrm{K} \iff f(\hat{\mathbf{u}}, \hat{\mathbf{r}}) = \mathbf{0}. \tag{10}$$

Moreover, we would like this function f to satisfy desirable properties so that a Newton algorithm would perform well on the root-finding problem $f(\hat{\mathbf{u}}, \hat{\mathbf{r}}) = \mathbf{0}$.

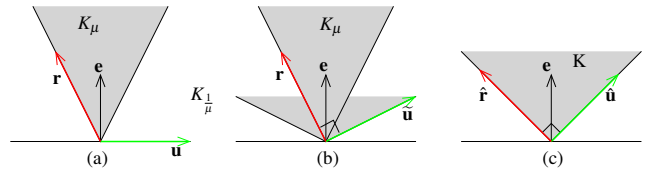

Figure 5: *Successive changes of variables applied onto (a) the relative velocity* $\mathbf{u}$ *and the contact force* $\mathbf{r}$*, in order to exhibit (b) the complementarity of* $\widetilde{\mathbf{u}}$ *and* $\mathbf{r}$ *proposed by De Saxcé and Feng [1998], and (c) our new complementarity relationship between* $\hat{\mathbf{u}}$ *and* $\hat{\mathbf{r}}$ *on the self-dual cone* K *for a pure frictional contact* ($\mu > 0$)*.*

[1] The $\mathbf{x} \perp \mathbf{y}$ notation expresses the orthogonality condition $\mathbf{x}^\top \mathbf{y} = 0$.

The key is to consider the theoretical work by Fukushima et al. [2002], who manage to extend some **NCP** complementarity functions to **SOCCP**s using the formalism of Jordan algebras. They first introduce the following Jordan product,

$$\forall \mathbf{x} = [\mathrm{x_N}, \mathbf{x_T}] \in \mathbb{R} \times \mathbb{R}^{d-1}, \forall \mathbf{y} = [\mathrm{y_N}, \mathbf{y_T}] \in \mathbb{R} \times \mathbb{R}^{d-1},$$
$$\mathbf{x} \circ \mathbf{y} = [< \mathbf{x}|\mathbf{y} >, \mathrm{x_N}\mathbf{y_T} + \mathrm{y_N}\mathbf{x_T}]$$

where $< \cdot|\cdot >$ denotes the Euclidian scalar product.

Then $(\mathbb{R}^d, +, \cdot, \circ)$ is an $\mathbb{R}$-algebra, tightly associated with the **SOC** K through the following properties:

- If $\mathbf{x} \in \mathrm{K}$, $\exists! \mathbf{s} \in \mathrm{K}$ s.t. $\mathbf{x} = \mathbf{s} \circ \mathbf{s}$; $\mathbf{s}$ is called the square root of $\mathbf{x}$ and is denoted $\mathbf{s} = \mathbf{x}^{\frac{1}{2}}$.
- $\forall \mathbf{x} \in \mathbb{R} \times \mathbb{R}^{d-1}$, $\mathbf{x} \circ \mathbf{x} \in \mathrm{K}$.

This means that $\forall (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^d$, $\mathbf{x} \circ \mathbf{x} + \mathbf{y} \circ \mathbf{y}$ is in K and possesses a unique square root. We can therefore define the *modified Fischer-Burmeister* function

$$f_{\mathrm{SOC}}^{\mathrm{FB}} : \begin{cases} \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}^d \\ (\mathbf{x}, \mathbf{y}) \quad \longmapsto \mathbf{x} + \mathbf{y} - (\mathbf{x} \circ \mathbf{x} + \mathbf{y} \circ \mathbf{y})^{\frac{1}{2}}, \end{cases} \tag{12}$$

and from [Fukushima et al. 2002] we get the equivalence

$$\mathrm{K} \ni \mathbf{x} \perp \mathbf{y} \in \mathrm{K} \iff f_{\mathrm{SOC}}^{\mathrm{FB}}(\mathbf{x}, \mathbf{y}) = \mathbf{0}. \tag{13}$$

Solving our frictional contact problem (4) with $\mu > 0$ therefore amounts to solving the root-finding problem $f_{\mathrm{SOC}}^{\mathrm{FB}}(\hat{\mathbf{u}}, \hat{\mathbf{r}}) = \mathbf{0}$, which is equivalent to solving the following minimization problem,

$$\min_{r \in \mathbb{R}^d} \Phi(\mathbf{r}) \quad \text{with} \quad \Phi(\mathbf{r}) := \frac{1}{2} \| f_{\mathrm{SOC}}^{\mathrm{FB}}(\hat{\mathbf{u}}(\mathbf{r}), \hat{\mathbf{r}}(\mathbf{r})) \|^2, \tag{14}$$

where $\hat{\mathbf{r}}(\mathbf{r})$ is defined in Equation (7) and $\hat{\mathbf{u}}(\mathbf{r})$ can be computed as $\hat{\mathbf{u}}(\widetilde{\mathbf{u}}(\mathbf{u}(\mathbf{r})))$ using Equations (7), (5), and (4).

From [Chen et al. 2008] we know that even if $f_{\mathrm{SOC}}^{\mathrm{FB}}$ is nonsmooth, $(\mathbf{x}, \mathbf{y}) \mapsto \frac{1}{2} \| f_{\mathrm{SOC}}^{\mathrm{FB}}(\mathbf{x}, \mathbf{y}) \|^2$ possesses the *SC1* property: it is continuously differentiable and its gradient is semismooth. Therefore, we have a good theoretical basis to use a nonsmooth Newton algorithm [Qi and Sun 1993] on formulation (14).

Detailed steps to compute $f_{\mathrm{SOC}}^{\mathrm{FB}}(\hat{\mathbf{u}}(\mathbf{r}), \hat{\mathbf{r}}(\mathbf{r}))$ and its Jacobian are provided in Appendix A.1, and we outline in Appendix A.2 a Newton algorithm solving our one-step problem (4). Let us now go ahead with the full n-contacts problem.

6 A Hybrid Gauss-Seidel Solver

In the last decades, splitting algorithms have been widely applied to friction-related problems [Baraff 1991; Moreau and Jean 1996; Jourdan et al. 1998; Duriez et al. 2006]. The basic idea, referred to as a *Gauss-Seidel* strategy, consists in iteratively solving each contact while updating the right-hand side $\underline{\mathbf{b}}_i$ of the local system to account for the neighboring contacts. Given enough computation time, this approach proved to be able to handle hundreds of thousands of contacts in granular materials [Dubois and Jean 2006].

While the general structure of the Gauss-Seidel algorithm is quite rigid, there is still room for experimentation with the code at its heart, *i.e.*, with the local Coulomb friction solver. As will be pointed out in Section 7.3, previous approaches do not prove robust or fast enough to handle our hair simulations. We hereafter present a novel local solver for the difficult problem of contact with exact Coulomb friction in large rod assemblies. This new solver manages to be both efficient and robust, thanks to a hybrid strategy.

6.1 A new hybrid local solver

The first step to solve our local problem is to use a nonsmooth Newton algorithm on our modified Fischer-Burmeister formulation introduced in Section 5.2. When achieving convergence, this algorithm is fast, especially on our 2×2 or 3×3 matrices. However, if it is initialized too far from the solution, it might fail to solve our local system. This happens in a non-negligible percentage of cases, generally between 0.001% and 0.1% of the cases raised by our rod problems (see Section 7). Unfortunately, we have learned from our experiments that whatever local solver is used, failing to converge at a reasonable precision for just one single call may be fatal to the global convergence of the Gauss-Seidel loop, leading to a crash of the simulation. Hence we aim at strongly *securing* our local solver.

One natural way for strengthening our solver consists in equipping it with a line-search. However, we found that using classical line-searches, such as the Armijo or Goldstein-Price rules, hardly increased the convergence properties of the algorithm while considerably degrading performance at the same time. Therefore, we chose instead to use a pure Newton strategy, and devised another fall-back strategy that proved to be much more efficient.

The key idea is to use an analytical, enumerative solver as a fail-safe. More precisely, we have implemented the α-formulation described in [Bonnefon and Daviet 2011]. This kind of solver works by successively checking the existence of solutions in the take-off case of Coulomb's law, then in the sticking case, and finally in the sliding case. The last case is the most difficult to solve: the problem amounts to finding the roots of a degree-four polynomial. While slower[2] than the Newton approach, this solver possesses an interesting feature: it gives the precise structure of the solution subspace. Consequently, we are able to find out when there is no solution to the problem[3]. This contrasts with optimization-based methods where the objective function would still have a finite value. Conversely, when solutions do exist, we are able to determine them up to the machine precision.

The outline of the enumerative algorithm is given in Appendix B.2. Note that though possible in theory, the generalization of this method to the n-contacts problem yields polynomials of high degree (4^n), which rapidly makes the problem intractable.

6.2 Full algorithm

The pseudo-code for our full solver is provided in Algorithm 1. For the sake of efficiency, additional optimization steps are included as described below.

The error of a given iterate $\mathbf{r}^k$ is evaluated as $\| f_{\mathrm{SOC}}^{\mathrm{FB}} \|_\infty$. Note that $f_{\mathrm{SOC}}^{\mathrm{FB}}$ is here computed in a global way, *i.e.*, using the full matrix $\mathbf{W}$ to derive the relative velocities $\mathbf{u}$ from the contact forces $\mathbf{r}$. We use the infinity norm to ensure that a large error at a given contact cannot be compensated by the sheer number of other contacts — which could happen with scaled norms such as $\frac{1}{n} \| f_{\mathrm{SOC}}^{\mathrm{FB}} \|$.

Contact freezing It has been shown in computational mechanics [Radjai et al. 1998] that, especially with stacking problems, prominent chains of stress rapidly appear in the Gauss-Seidel iterative process. These chains bear the greatest part of the structure, while local networks of much weaker forces ensure the stabilization of individual objects. As long as the rest of the structure is not yet stabilized, those local networks are prone to subsequent variations from one iteration to the other. To focus our computational power

[2] In practice, roughly three times slower; see Table 2 in section 7.3.

[3] Even one contact-problems may have no solution if $\mathbf{W}$ is singular.

Algorithm 1 Our hybrid Gauss-Seidel algorithm.

Require: Problem in its reduced form: $\mathbf{W}$, $\mathbf{b}$
Require: Initial guess $\mathbf{r}^0$
Ensure: Solution of $\mathrm{f}^{\mathrm{FB}}_{\mathrm{SOC}} = \mathbf{0}$
  Split $\mathbf{W}$ into n^2 $d \times d$ blocks $\mathbf{W}_{\alpha\beta}$, $(\alpha,\beta) \in [1,n]^2$
  $\mathrm{k} \leftarrow \mathbf{0}$
  while $\mathrm{k} \leq \mathrm{N}_{\mathrm{itmax}}$ and $\|f^{\mathrm{FB}}_{\mathrm{SOC}}(\mathbf{r}^k)\|_\infty > \mathrm{tol}$ **do** // *Gauss-Seidel loop*
    for $\mathrm{i} = 1 \ldots \mathrm{n}$ **do** // *For each contact*
      $\underline{\mathbf{b}}_\mathrm{i} \leftarrow \mathbf{b}_\mathrm{i} - \sum_{\mathrm{j}<\mathrm{i}} \mathbf{W}_{\mathrm{ij}}\,\mathbf{r}^\mathrm{k}_\mathrm{j} - \sum_{\mathrm{j}>\mathrm{i}} \mathbf{W}_{\mathrm{ij}}\,\mathbf{r}^{\mathrm{k}-1}_\mathrm{j}$
      // *Solve local friction problem*
      $\mathbf{r}^\mathrm{k}_\mathrm{i} \leftarrow solveNewtonFB(\mathbf{W}_{ii}, \underline{\mathbf{b}}_i, \mathbf{r}^{k-1}_i)$
      if $\|f^{\mathrm{FB}}_{\mathrm{SOC}}(\mathbf{r}^\mathrm{k}_\mathrm{i})\| > \mathrm{tol}_{\mathrm{local}}$ **then**
        $\mathbf{r}^\mathrm{k}_\mathrm{i} \leftarrow solveEnumerative(\mathbf{W}_{ii}, \underline{\mathbf{b}}_i)$
        if $\|f^{\mathrm{FB}}_{\mathrm{SOC}}(\mathbf{r}^\mathrm{k}_\mathrm{i})\| > \mathrm{tol}_{\mathrm{local}}$ **then**
          $\mathbf{r}^\mathrm{k}_\mathrm{i} \leftarrow \mathbf{0}$
        end if
      end if
      if $\|\mathbf{r}^\mathrm{k}_\mathrm{i} - \mathbf{r}^{\mathrm{k}-1}_\mathrm{i}\| < \varepsilon_{\mathrm{smallChange}}$ or $\|\mathbf{r}^\mathrm{k}_\mathrm{i}\| < \varepsilon_{\mathrm{weakForce}}$ **then**
        Stop solving contact i for $\mathrm{N}_{\mathrm{disableiters}}$ iterations
      end if
    end for
    $\mathrm{k} \leftarrow \mathrm{k}+1$
  end while

on the relevant parts, we temporarily stop solving contacts when their resulting force remains roughly constant or is too small.

Assembly of W The cost of assembling $\mathbf{W}$ can become prohibitive when the size of the system is large, as it grows in $O(n^2 + nm)$. Otaduy *et al.* [2009] proposed a splitting method to reduce this complexity, assuming that individual objects have an "almost" diagonally dominant mass-matrix. However, this assumption does not hold in general, and especially not in the case of reduced-coordinates models.

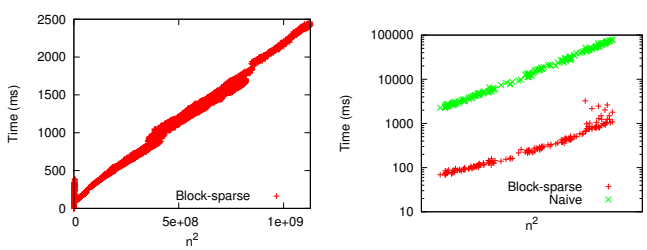

Figure 6: *Computational time for assembling* $\mathbf{W}$.
Left: In the full simulation "Curly", the time to compute $\mathbf{W}$ *grows steadily with* n^2.
Right: Computation times of block-sparse vs. *classical sparse aware algorithms on the batch of problems of Section 7.3.*

In our generic framework, we can still get reasonable computation times by exploiting the block-diagonal structure of $\mathbf{M}$ and the block-sparse structure of $\mathbf{H}$ which allows us to easily locate the non-zero $d \times d$ blocks of $\mathbf{W}$. Let us denote by η the number of these non-zero blocks, N the number of objects and M the maximum number of degrees of freedom per object. The cost of the computation then becomes $O(NM^3 + \eta M)$. Moreover, as $\mathbf{W}$ is symmetric, we only have to do half of the work.

As shown in Figure 6, for large systems – assuming that M remains bounded – the dominant term is $O(\eta)$, where η grows as $O(n^2)$. In practice however, the constant factor is small and we can account for around $20,000$ contacts in less than one second. Moreover, since the cost of one Gauss-Seidel iteration is also $O(\eta)$, there is not much to gain by further optimizing the assembly of $\mathbf{W}$.

7 Results

In this section, we present and evaluate our results in terms of realism, numerical robustness, and computational efficiency. All our simulations are shown in the accompanying video.

7.1 Framework

Fiber models We have tested the coupling of our solver on three different rod models: the super-helix model [Bertails et al. 2006], an implicit version of the CoRdE model [Spillmann and Teschner 2007], and the Discrete Elastic Rod [Bergou et al. 2008] with the fast-projection algorithm from [Goldenthal et al. 2007] and explicit bending stresses. Overall we found the super-helix model yields the most stable simulations. This is not only due to the implicit formulation of bending stresses, but also to the reduced kinematics which exactly preserves inextensibility and allows for a straightforward coupling to the contact solver. In contrast, for the Discrete Elastic Rod where inextensibility is enforced through additional nonlinear constraints, we found that intermingling fast-projection steps with iterations of our frictional contact solver often results in a degradation of convergence. For these reasons we have been using the super-helix model in all our hair simulations.

Collision detection To perform collision detection, we compute a linear spatial discretization of the rods and embed them into a set of bounding cylinders. We first use an adapted version of the spatial hashing method presented in [Teschner et al. 2003] to extract a rough subset of colliding pairs, then post-process them more finely to eliminate the false-positives. To facilitate the task of the contact solver, we track the evolution of contacts between timesteps so that we can provide a good initial guess of the forces for the contacts that are already active. This trick helps speed up the convergence of the solver in "static" phases of the simulation, such as stable stacking.

Since performing continuous detection would be too time-consuming, we avoid missing collisions by using an adaptative time-stepping scheme that automatically reduces the timestep during dynamic phases and resets the timestep to its default value when the hair motion gets gentler. Note that the overall time taken by collision detection steadily stays under 10% of the whole simulation step, and therefore never becomes the bottleneck of our framework.

Frictional contact solvers In the following results, we compare our method against a variety of frictional contact solvers,

- **MFB**: Our Newton method based on the modified Fischer-Burmeister formulation. It can be used either as a global solver or as a local solver within a Gauss-Seidel loop.
- **PAC** and **DAC**: Respectively pure and damped Newton methods based on the Alart-Curnier formulation used in [Bertails-Descoubes et al. 2011]. Both can be used as either global or local solvers.
- **Duriez06** and **Duriez08**: Local solvers from respectively [Duriez et al. 2006] and [Duriez 2008]. The latter consists of successive iterations of the former embedded in a fixed point loop. Unlike the [2006] approach, it solves the exact Coulomb friction problem — if convergence is achieved.
- **4sides**: The local solver from [Otaduy et al. 2009]. It approximates the Coulomb friction cone with a four-sided pyramid.
- **Enum**: The quartic enumerative local solver from [Bonnefon and Daviet 2011].

7.2 Qualitative validation

Exact Coulomb friction To illustrate the influence of the choice of the local friction formulation, we created a very simple experiment: the free end of a fiber is dropped on a ball that is rotating with sinusoidal oscillations, and a non-zero friction coefficient ($\mu = 1$) is set between the ball and the fiber. Figure 7 (top) shows the phase plots of the free end once it has reached its periodic orbit.

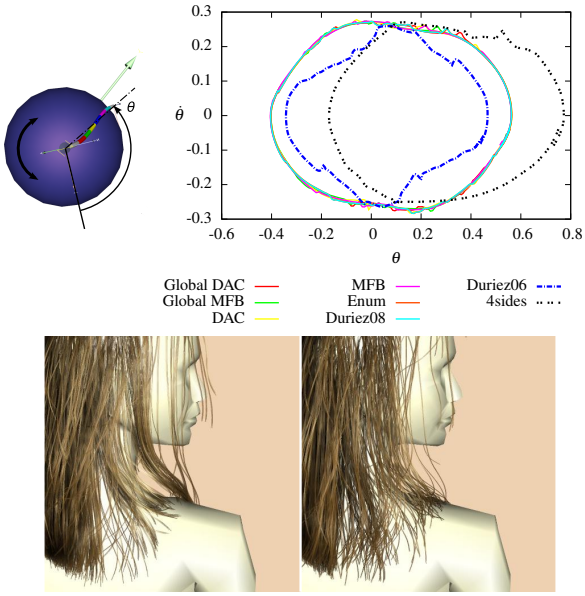

Figure 7: *Exact* vs. *approximate model for Coulomb friction. Top: Periodic orbit of the free end of a rod resting on top of a rotating sphere. Frictional contact is simulated using our eight different solvers. Solvers that model exact Coulomb friction all reach the same orbit, no matter the choice of the error function. Other models (dashed lines) reach completely different ones. Bottom: Visual comparison of a full hair simulation with exact (left) and linearized (right) Coulomb friction, for equal computation time. With exact Coulomb friction, spontaneous hair clumping emerges in a more visible way.*

We tried two global Newton methods, **DAC** and **MFB**, and six local solvers within a Gauss-Seidel loop: **DAC** and **MFB** again, as well as **Duriez06**, **Duriez08**, **4sides** and **Enum**. For **DAC** and **MFB** based solvers, the stopping criterion relied on the norm of the corresponding objective functions, while the four remaining methods were stopped as soon as the step size between two Gauss-Seidel iterations got below 5%.

No matter the choice of the stopping criterion, all solvers that model exact Coulomb friction converged towards the same orbit. For the two others (**Duriez06** and **4sides**), the Gauss-Seidel algorithm still converged quickly to a fixed point, but the trajectories substantially differed. This both experimentally validates our **MFB** solver and highlights the fact that the convergence of the Gauss-Seidel iterates does not necessarily mean that the problem has been accurately solved: a proper objective function should be computed.

Finally, we show in Figure 7 (bottom) and in the supplemental video that choosing an exact rather than a faceted Coulomb friction solver does have a visual impact on large hair simulations: the clumping of neighboring strands is more apparent. Note furthermore that on large hair problems, using the faceted solver **4sides** – as well as any solver other than **MFB** – always resulted in a crash after computing a few frames of simulation (see Section 7.3).

Full-scale simulations To evaluate the effectiveness of our solver on hair simulations, we ran three kinds of experiments. All are using the super-helix model, with 16 degrees of freedom per rod. The character was animated under 3ds Max [Autodesk 2009] by reproducing a real video-captured motion that serves as a reference (see Figure 1 and the accompanying video). The hair simulation entitled "Free" models a full, unconstrained haircut, and consists of about 2000 simulated rods. It is divided into two parts, "A" and "B", featuring a head rotation motion and a head leaning motion, respectively. The third hair simulation, entitled "Pony", contains only a few more than 300 rods, but those are tightly packed into a ponytail. All of these experiments include smooth as well as rough head motions. Finally, a last motion, "Curly', illustrates the fact that our method can also easily handle curly hair (see Figure 10). It is based on the "Free/A" head motion.

Figure 8: *Simulation of a fast head movement without (top) and with (bottom) Coulomb friction. In the latter case, hair remains much more coherent and the results are visually more pleasing.*

Visual results and comparisons to real hair motions are presented in Figures 1, 8, 9, 10, and in the accompanying video. Final rendering was performed under 3ds Max. Our method allows us to preserve the hair volume and to capture subtle phenomena such as stick-slip instabilities or the spontaneous appearance of transient coherent movements in hair. We also observe that a lot of energy is dissipated by Coulomb friction. Capturing it accurately is essential to achieve realism. Figure 8 shows the effect of increasing the hair/hair friction coefficient on the collective hair behavior. Without friction, hair looks artificially clean and light. In contrast, in the presence of friction, motion looks coherent and properly damped, while simultaneously featuring complex details at the fiber level.

7.3 Evaluation of performance and robustness

Measure of performance and robustness All our simulations were run on a desktop machine featuring a Core-i7 processor with 8 GB of memory. Numerical results for the three experiments described above are given in Table 1. The tolerance for the global Gauss-Seidel was chosen so that we do not observe any visual disturbance. In our internal units[4], the tolerance ranges from 10^{-4} when the motion is fast to 10^{-6} in long static phases. Though the solver sometimes fails to strictly reach the requested precision (in less than 2% of the cases), large errors never occur: the solver always gives an approximate solution with which the simulation can continue without exhibiting any artifact. To improve the convergence of the global algorithm, the local tolerance is always set to a lower value, typically 10^{-7}.

[4]This roughly translates into an average dimensionless relative error on $\mathbf{r}$ ranging from 10^{-3} to 10^{-4}.

Table 1: *Physical properties and performance results for our hair simulations.*

Name	N_{rods}	$\mu_{\text{hair}}/\mu_{\text{body}}$	$\overline{N_{\text{contacts}}}$	$\max(N_{\text{contacts}})$	$\%_{>\text{tol}}$ [1]	$\max(err)$	$\overline{\text{iters}}$	$\overline{T_{\text{GS}}}$ / $\overline{T_{\text{solv}}}$ [2] (s)	$\overline{T_{dt}}$ [3] (s)	$\overline{T_{\text{frame}}}$ (min)	T_{tot} (hours)
Free/A	1920	0.3 / 0.5	24659	46915	0.056	0.004	136	2.60 / 4.06	7.15	2.09	25
Free/B	1920	0.2 / 0.3	28153	36287	0.13	0.037	160	5.3 / 7.0	11.1	3.02	36
Pony	334	0.3 / 0.5	9850	16613	0.42	0.003	301	4.43 / 5.25	6.2	2.47	30
Curly	1920	0.3 / 0.5	21425	33578	0.013	0.003	118	2.26 / 3.39	7.54	4.15	49

Overlines indicate averaged quantities.
[1] $\%_{>\text{tol}}$: Percentage of one-step problems that did not reach (global) tolerance
[2] T_{GS} / T_{solv}: Time for the Gauss-Seidel loop (Algorithm 1) alone / Total time for the contact solver (including the assembly of **W**)
[3] T_{dt}: Total time for one simulation timestep (dt ranges from 1 to 4ms)

Figure 9: *Our simulations (right) capture the complex patterns emerging from static friction in real hair (left).*

Each simulation was run at a rate of a few minutes per frame, that is, 25 seconds of video in about 48 hours.

Comparisons against other local solvers We could not get the full-scale simulations to complete with any local solver other than our hybrid method **MFB+Enum**. All other solvers led to the divergence of the fiber model at some point of the simulation. Still, to quantitatively evaluate the performance of our solver, we saved about three hundred one-step problems from our least challenging simulation, "Pony". We then successively ran on this batch all the local solvers presented in Section 7.1, with or without the help of the enumerative solver.

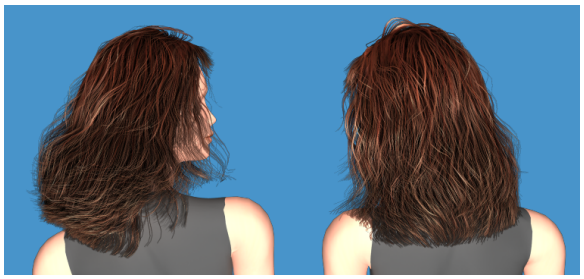

Figure 10: *Our method can also handle curly hair. Note how the volume of the hairstyle is preserved throughout the simulation, and how static friction is properly captured near the top of the head.*

Table 2: *Performance comparison of various local solvers on a set of 306 one-step problems.*

Local solver	$\%_{\text{alt}}$ [1]	$\%_{\text{fail}}$ [2]	$\%_{>\text{tol}}$ [3]	$\overline{\text{iters}}$ [4]	$\overline{T_{\text{GS}}}$ [6] (ms)
4sides	–	0	0	575	6497
Duriez08	–	2.45	50	267	4336
PAC	–	0.43	19.3	163	1265
DAC	–	0.26	0.33	60	874
MFB	–	0.13	4.9	72	484
Enum	–	0.0005	1	67	1044
4sides + Enum	38.7	0.001	0	47	763
Duriez08 + Enum	51	0.0007	0.33	90	1447
PAC + Enum	0.1	10^{-5}	0	62	543
DAC + Enum	0.09	10^{-6}	0	57	789
MFB + Enum	**0.07**	$\mathbf{10^{-6}}$	**0**	**41**	**312**

[1] $\%_{\text{alt}}$: Percentage of calls to fail-safe
[2] $\%_{\text{fail}}$: Percentage of local problems that did not reach tolerance
[3] $\%_{>\text{tol}}$: Percentage of one-step problems that were not solved to (global) tolerance (tol $= 10^{-6}$ except for **4sides**: 5×10^{-2})
[4] $\overline{\text{iters}}$: Mean number of Gauss-Seidel iterations
[5] $\overline{T_{\text{GS}}}$: Mean time in Gauss-Seidel algorithm

Convergence plots are shown in Figure 11 and numerical results are given in Table 2. Note that **Duriez06** is missing from this table as it always led to prompt divergence on these problems. We used the Fischer-Burmeister error measure for all our tests, except for the pure **4sides** solver which does not attempt to solve exact Coulomb friction[5].

From these numerical results we first note that using a linearized cone does not necessarily bring better time performance, despite a lower cost per call of the local solver. The **4sides** method was actually the one that required the greatest number of Gauss-Seidel iterations to reach sufficient accuracy. We also observed that the contact freezing policy was not of much benefit to the **4sides** solver which, as a result, had to process a higher number of local problems per Gauss-Seidel iteration than exact friction solvers. Using a finer Coulomb friction model thus does *not necessarily* imply more costly simulations, quite the reverse in our case.

A second interesting point deals with the analysis of the role played by the **Enum** solver. For all the local solvers we tested, using the **Enum** solver as a fail-safe improved both the success rate and the computation time. In such a configuration, it turns out to be unnecessary to use a very robust primary solver. Indeed, the **MFB**, the **PAC** and more surprisingly the **4sides** method all outperformed the **DAC** solver. We also noted that the **Enum** solver requires a large number of iterations to reach the global tolerance, despite a very low rate of unsolved local problems. This is due to problems

[5] In this case, we resorted to a measure of the iterates length. To avoid introducing any bias, we made sure that the tolerance was such that the timings of **MFB+Enum** were roughly the same under both error measures.

that do not admit an analytical solution, but for which an approximate solution with a low numerical error can still be found. While optimization-based solvers do find such an acceptable solution, the **Enum** solver remains stuck, thus spoiling the global convergence of the Gauss-Seidel algorithm. A ε-tolerance can be defined for such cases, but this does not fully eliminate this difficulty. When using the **Enum** solver as a fail-safe, the rate of remaining problems without solution fortunately becomes very low, and in these rare cases, simply reseting the force to zero proved to be sufficient.

Overall, these results confirm our claim that our hybrid method **MFB+Enum** is both robust *and* efficient. The **MFB** solver alone turns out to be both faster and more robust than **PAC**, and, although slightly less robust, far much faster than **DAC**. The robustness issue becomes irrelevant when the solvers are combined with **Enum**.

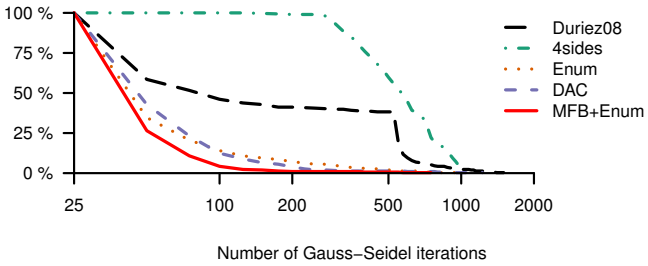

Figure 11: *Percentage of one-step problems (Y axis) requiring more than a given number of Gauss-Seidel iterations (X axis) to converge, for various local solvers. Our hybrid method (solid red) induces better global convergence than previous approaches.*

7.4 Limitations

Even though our hair simulations appear more convincing than those generated by standard methods, they do not perfectly match the real movement depicted in the reference videos. This is partly due to inaccuracies in the hair styling process and to the difficulty of precisely identifying the true air/hair and hair/hair friction coefficients. Another reason is related to the actual complexity of hair interactions. While Coulomb friction is a key ingredient, further effects such as anisotropic friction, adhesion, or electrostatics are likely to influence the hair motion, depending on the hair state (clean, dirty, wet) and external conditions (dry or wet atmosphere).

Finally, for now we have tested our solver on hair examples consisting of a few thousands fibers only – that is, 50 times smaller than a real human hair. Although we feel our solver could possibly resist some further scaling up in terms of convergence quality, we know that computational time will rapidly become the main bottleneck. Using a parallelization strategy would be an interesting option for gaining one more order of magnitude while keeping reasonable computational timings.

8 Conclusion

We have introduced a robust iterative solver for computing the dynamics of a large assembly of hair fibers in the presence of Coulomb friction. Relying on a new hybrid local solver that almost always converges to the desired precision for each nonsmooth problem, our method allows us to capture the subtle effects emerging from real hair collective behavior. In the future we would like to explore massive parallelization schemes so as to increase again the resolution of our simulations and get closer to the true complexity of a full head of hair. Another interesting direction of research deals with the improvement of the parameter setting process in hair simulations. When trying to reproduce a given hair motion, approaching the real friction coefficients and other physical parameters with tests and trials can be very tedious. Instead, we would like to automatically exploit captured data through an inverse dynamics process.

Acknowledgements

We would like to thank Sylvain Paris and Tilke Judd for sharing with us their video captures of real hair, and Nathan Krislock for carefully proofreading our paper.

References

ACARY, V., AND BROGLIATO, B. 2008. *Numerical methods for nonsmooth dynamical systems*, vol. 35 of *Lecture Notes in Computational and Applied Mechanics*. Springer.

ACARY, V., CADOUX, F., LEMARECHAL, C., AND MALICK, J. 2011. A formulation of the linear discrete Coulomb friction problem via convex optimization. *ZAMM / Z angew Math Mech; Zeitschrift für Angewandte Mathematik und Mechanik 91* (02), 155–175.

ALART, P., AND CURNIER, A. 1991. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput. Methods Appl. Mech. Eng. 92*, 3, 353–375.

ANITESCU, M., AND TASORA, A., 2008. An iterative approach for cone complementarity problems for nonsmooth dynamics.

ANJYO, K., USAMI, Y., AND KURIHARA, T. 1992. A simple method for extracting the natural beauty of hair. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'92 conference)*, 111–120.

AUTODESK, 2009. 3ds Max Hair & Fur. http://usa.autodesk.com/.

BARAFF, D. 1991. Coping with friction for non-penetrating rigid body simulation. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'91 conference)*, ACM, 31–40.

BARAFF, D. 1994. Fast contact force computation for nonpenetrating rigid bodies. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'94 conference)*, ACM, New York, NY, USA, 23–34.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'08 conference) 27*, 3, 1–12.

BERGOU, M., AUDOLY, B., VOUGA, E., WARDETZKY, M., AND GRINSPUN, E. 2010. Discrete viscous threads. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'10 Conference)*.

BERTAILS-DESCOUBES, F., CADOUX, F., DAVIET, G., AND ACARY, V. 2011. A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Transactions on Graphics 30* (February), 6:1–6:14.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'06 conference) 25* (July), 1180–1187.

BONNEFON, O., AND DAVIET, G. 2011. Quartic formulation of Coulomb 3D frictional contact. Tech. rep., INRIA - Laboratoire Jean Kuntzmann.

CHEN, J.-S., SUN, D., AND SUN, J. 2008. The SC1 property of the squared norm of the SOC Fischer-Burmeister function. *Operations Research Letters 36*, 3, 385 – 392.

CHOE, B., CHOI, M., AND KO, H.-S. 2005. Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, SCA '05, 153–160.

DUBOIS, F., AND JEAN, M. 2006. The non smooth contact dynamic method: recent LMGC90 software developments and application. In *Analysis and Simulation of Contact Problems*, P. Wriggers and U. Nackenhorst, Eds., vol. 27 of *Lecture Notes in Applied and Computational Mechanics*. Springer Berlin / Heidelberg, 375–378. 10.1007/3-540-31761-9-44.

DURIEZ, C., DUBOIS, F., KHEDDAR, A., AND ANDRIOT, C. 2006. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 1*, 12, 36–47.

DURIEZ, C. 2008. Rendering of frictional contact with deformable environments. In *Haptic Rendering: Foundations, Algorithms and Applications*. A. K. Peters, Ltd., Natick, MA, USA, 421–442.

ERLEBEN, K. 2007. Velocity-based shock propagation for multibody dynamics animation. *ACM Transaction on Graphics 26*, 2.

FISCHER, A. 1992. A special Newton-type optimization method. *Optimization: A Journal of Mathematical Programming and Operations Research 24*, 3, 269 – 284.

FUKUSHIMA, M., LUO, Z.-Q., AND TSENG, P. 2002. Smoothing functions for second-order-cone complementarity problems. *SIAM J. on Optimization 12* (February), 436–460.

G. DE SAXCÉ, G., AND FENG, Z.-Q. 1998. The bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. *Math. Comput. Modelling 28*, 4-8, 225–245.

GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. In *ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, SIGGRAPH '07.

HADAP, S., AND MAGNENAT-THALMANN, N. 2001. Modeling dynamic hair as a continuum. *Computer Graphics Forum 20*, 3, 329–338. Proceedings of Eurographics'01.

HADAP, S. 2006. Oriented strands - dynamics of stiff multi-body system. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'06)*, ACM-EG SCA, 91–100.

HARMON, D., VOUGA, E., SMITH, B., TAMSTORF, R., AND GRINSPUN, E. 2009. Asynchronous contact mechanics. ACM, New York, NY, USA.

JOURDAN, F., ALART, P., AND JEAN, M. 1998. A Gauss-Seidel like algorithm to solve frictional contact problems. *Computer Methods in Applied Mechanics and Engineering 155*, 31–47.

KALDOR, J., JAMES, D., AND MARSCHNER, S. 2008. Simulating knitted cloth at the yarn level. In *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'08 conference*, ACM, New York, NY, USA, 1–9.

KALDOR, J., JAMES, D., AND MARSCHNER, S. 2010. Efficient yarn-based cloth with adaptive contact linearization. In *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'10 conference*, ACM, New York, NY, USA, vol. 29, 105:1–105:10.

KANNO, Y., MARTINS, J. A. C., AND COSTA, A. P. D., 2004. Second-order cone linear complementarity formulation of quasi-static incremental frictional contact problem.

KAUFMAN, D., EDMUNDS, T., AND PAI, D. 2005. Fast frictional dynamics for rigid bodies. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'05 conference) 24*, 3, 946–956.

KAUFMAN, D., SUEDA, S., JAMES, D., AND PAI, D. 2008. Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH Asia'08 conference) 27*, 5, 164:1–164:11.

MCADAMS, A., SELLE, A., WARD, K., SIFAKIS, E., AND TERAN, J. 2009. Detail preserving continuum simulation of straight hair. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'09 conference) 28*, 3, 1–6.

MOREAU, J.-J., AND JEAN, M. 1996. Numerical treatment of contact and friction: the contact dynamics method. In *The 1996 3rd Biennial Joint Conference on Engineering Systems Design and Analysis, ESDA*, 201–208.

MOREAU, J.-J., 1988. Unilateral contact and dry friction in finite freedom dynamics. Nonsmooth mechanics and applications, CISM Courses Lect. 302, 1-82 (1988).

MOREAU, J. 1994. Some numerical methods in multibody dynamics: Application to granular materials. *European Journal of Mechanics - A/Solids supp.*, 4, 93–114.

OTADUY, M., TAMSTORF, R., STEINEMANN, D., AND GROSS, M. 2009. Implicit contact handling for deformable objects. *Computer Graphics Forum (Proceedings of Eurographics'09) 28*, 2 (apr).

PAI, D. 2002. Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum 21*, 3, 347–352. Proceedings of Eurographics'02.

PLANTE, E., CANI, M.-P., AND POULIN, P. 2001. A layered wisp model for simulating interactions inside long hair. In *EG workshop on Computer Animation and Simulation (EG CAS'01)*, Springer, Computer Science, 139–148.

QI, L., AND SUN, J. 1993. A nonsmooth version of Newton's method. *Mathematical Programming 58*, 353–367.

RADJAI, F., WOLF, D., JEAN, M., AND MOREAU, J.-J. 1998. Bimodal character of stress transmission in granular packings. *Phys. Rev. Lett. 80* (Jan), 61–64.

ROSENBLUM, R., CARLSON, W., AND TRIPP, E. 1991. Simulating the structure and dynamics of human hair: Modeling, rendering, and animation. *The Journal of Visualization and Computer Animation 2*, 4, 141–148.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'08 conference) 27*, 3, 1–11.

SILCOWITZ, M., NIEBE, S., AND ERLEBEN, K. 2009. Nonsmooth Newton Method for Fischer function reformulation of contact force problems for interactive rigid body simulation. 105–114.

SPILLMANN, J., AND TESCHNER, M. 2007. CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic

objects. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'07)*, ACM-EG SCA, 63–72.

TESCHNER, M., HEIDELBERGER, B., MÜLLER, M., POMERANTS, D., AND GROSS, M. 2003. Optimized spatial hashing for collision detection of deformable objects. In *Vision, Modeling, Visualization (VMV 2003)*, 47–54.

WARD, K., SIMMONS, M., MILNE, A., YOSUMI, H., AND ZHAO, X. 2010. Simulating Rapunzel's hair in Disney's Tangled. In *ACM SIGGRAPH 2010 Talks*, ACM, New York, NY, USA, SIGGRAPH '10, 22:1–22:1.

ZHANG, H., LI, J., AND PAN, S. 2011. New second-order cone linear complementarity formulation and semi-smooth Newton algorithm for finite element analysis of 3d frictional contact problem. *Computer Methods in Applied Mechanics and Engineering 200*, 1-4, 77 – 88.

A Modified Fischer-Burmeister Solver

A.1 Computation of $f^{\mathsf{FB}}_{\mathsf{SOC}}$ and of its Jacobian

Let $\mathbf{z} := \hat{\mathbf{r}} \circ \hat{\mathbf{r}} + \hat{\mathbf{u}} \circ \hat{\mathbf{u}}$, and let us define

$$\begin{array}{rcl} \boldsymbol{\alpha} & := & \hat{\mathrm{r}}_{\mathrm{N}}\hat{\mathbf{r}}_{\mathrm{T}} + \hat{\mathrm{u}}_{\mathrm{N}}\hat{\mathbf{u}}_{\mathrm{T}} \\ \lambda_i & := & \|\hat{\mathbf{r}}\|^2 + \|\hat{\mathbf{u}}\|^2 + 2(-1)^i\|\boldsymbol{\alpha}\| \\ \boldsymbol{\omega}_i & := & \begin{cases} \frac{1}{2}\left[1, (-1)^i \frac{\boldsymbol{\alpha}}{\|\boldsymbol{\alpha}\|}\right] & \text{if } \boldsymbol{\alpha} \neq \mathbf{0} \\ \frac{1}{2}\left[1, (-1)^i \boldsymbol{\beta}\right] & \text{if } \boldsymbol{\alpha} = \mathbf{0} \end{cases} \end{array}$$

with $\boldsymbol{\beta}$ any unitary vector in $\mathbb{R}^{d-1}$. λ_1 and λ_2 are called the *eigenvalues* of $\mathbf{z}$, and $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ are their associated *eigenvectors*. They satisfy $\mathbf{z} = \lambda_1\boldsymbol{\omega}_1 + \lambda_2\boldsymbol{\omega}_2$, and $\mathbf{z}^{\frac{1}{2}} = \sqrt{\lambda_1}\boldsymbol{\omega}_1 + \sqrt{\lambda_2}\boldsymbol{\omega}_2$. We can therefore directly compute

$$f^{\mathrm{FB}}_{\mathrm{SOC}}(\hat{\mathbf{r}}, \hat{\mathbf{u}}) = \hat{\mathbf{r}} + \hat{\mathbf{u}} - \left(\sqrt{\lambda_1}\boldsymbol{\omega}_1 + \sqrt{\lambda_2}\boldsymbol{\omega}_2\right). \tag{15}$$

To derive the gradient of $f^{\mathrm{FB}}_{\mathrm{SOC}}$, let us recall two properties of the **SOC** algebra:

- If $\mathbf{x} \in \mathrm{intK}$, $\mathbf{x}$ is invertible[6] and $\mathbf{x}^{-1} \in \mathrm{intK}$;
- $\mathbf{x} \in \mathbb{R} \times \mathbb{R}^{d-1}$ is invertible *iff* $\det \mathbf{x} = \mathrm{x}_{\mathrm{N}}^2 - \|\mathbf{x}_{\mathrm{T}}\|^2 \neq 0$.

Since we always have $\mathbf{z} \in \mathrm{K}$ and $\det \mathbf{z} = \lambda_1\lambda_2$, we get $\mathbf{z}$ invertible $\iff \mathbf{z} \in \mathrm{intK} \iff \lambda_1\lambda_2 \neq 0$. We can then distinguish three cases:

- If $\lambda_2 = 0$, *i.e.*, $\hat{\mathbf{r}} = \hat{\mathbf{u}} = \mathbf{0}$, we arbitrarly choose one element of the generalized Jacobian of $f^{\mathrm{FB}}_{\mathrm{SOC}}$ at this point, for example, $\frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{r}}}(\mathbf{0}, \mathbf{0}) = \frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{u}}}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$.
- Otherwise, if $\lambda_1 = 0$, then $\mathbf{z} \in \partial\mathrm{K} \setminus \mathbf{0}$, and we also have to choose one element of the generalized Jacobian of $f^{\mathrm{FB}}_{\mathrm{SOC}}$. We take[7]:

$$\frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{r}}}(\hat{\mathbf{r}}, \hat{\mathbf{u}}) = 1 - \frac{\hat{\mathrm{r}}_{\mathrm{N}}}{\hat{\mathrm{r}}_{\mathrm{N}}^2 + \hat{\mathrm{u}}_{\mathrm{N}}^2} I_d \text{ and } \frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{u}}}(\hat{\mathbf{r}}, \hat{\mathbf{u}}) = 1 - \frac{\hat{\mathrm{u}}_{\mathrm{N}}}{\hat{\mathrm{r}}_{\mathrm{N}}^2 + \hat{\mathrm{u}}_{\mathrm{N}}^2} I_d.$$

- Otherwise, we have $\mathbf{z} \in \mathrm{intK}$, and both $\mathbf{z}^{-1}$ and the Jacobian of $f^{\mathrm{FB}}_{\mathrm{SOC}}$ are uniquely defined. For $\mathbf{x} \in \mathbb{R}^d$, let us denote $L_{\mathbf{x}}$ the $d \times d$ matrix that satisfies $\forall \mathbf{y} \in \mathbb{R}^d\ L_{\mathbf{x}}\mathbf{y} = \mathbf{x} \circ \mathbf{y}$. We have

$$L_{\mathbf{z}^{-1}} = \frac{1}{\lambda_1\lambda_2}\begin{pmatrix} \mathrm{z}_{\mathrm{N}} & -\mathbf{z}_{\mathrm{T}}^{\top} \\ -\mathbf{z}_{\mathrm{T}} & \frac{\lambda_1\lambda_2}{\mathrm{z}_{\mathrm{N}}} I_{d-1} + \frac{\mathbf{z}_{\mathrm{T}}\mathbf{z}_{\mathrm{T}}^{\top}}{\mathrm{z}_{\mathrm{N}}} \end{pmatrix}.$$

$$\text{Then} \quad \frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{r}}}(\hat{\mathbf{r}}, \hat{\mathbf{u}}) = I_d - L_{\mathbf{z}^{-1}}L_{\hat{\mathbf{r}}} \text{ and } \frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{u}}}(\hat{\mathbf{r}}, \hat{\mathbf{u}}) = I_d - L_{\mathbf{z}^{-1}}L_{\hat{\mathbf{u}}}.$$

Finally, using the chain rule,

$$\frac{d f^{\mathrm{FB}}_{\mathrm{SOC}}}{d\mathbf{r}}(\mathbf{r}) = \frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{r}}}(\hat{\mathbf{r}}, \hat{\mathbf{u}})\frac{d\hat{\mathbf{r}}}{d\mathbf{r}} + \frac{\partial f^{\mathrm{FB}}_{\mathrm{SOC}}}{\partial \hat{\mathbf{u}}}(\hat{\mathbf{r}}, \hat{\mathbf{u}})\frac{d\hat{\mathbf{u}}}{d\tilde{\mathbf{u}}}\frac{d\tilde{\mathbf{u}}}{d\mathbf{u}}\frac{d\mathbf{u}}{d\mathbf{r}}.$$

Note that $\frac{d\tilde{\mathbf{u}}}{d\mathbf{u}}$ is not strictly defined when $\mathbf{u}_{\mathrm{T}} = \mathbf{0}$, because $\|\mathbf{u}_{\mathrm{T}}\|$ is not differentiable at $\mathbf{0}$. We can however take any element of its generalized Jacobian, such as I_d.

[6] With respect to the Jordan product, *i.e.*, $\exists \mathbf{y}$ s.t. $\mathbf{x} \circ \mathbf{y} = \mathbf{y} \circ \mathbf{x} = [1, \mathbf{0}]$.

[7] $\hat{\mathrm{r}}_{\mathrm{N}}^2 + \hat{\mathrm{u}}_{\mathrm{N}}^2 > 0$. Otherwise we would have $\hat{\mathrm{r}}_{\mathrm{N}} = \hat{\mathrm{u}}_{\mathrm{N}} = 0$, which would mean that $\mathbf{z}_T = \mathbf{0}$ and either $\mathbf{z} = \mathbf{0}$ or $\mathbf{z} \in \mathrm{intK}$.

A.2 Algorithm

Require: Problem in its reduced form: $\mathbf{W}$, $\mathbf{b}$, μ
Require: Initial guess $\mathbf{r}^0$
Require: $\sigma > 0$ (typically 10^{-3})
Ensure: Solution of $\mathrm{f}^{\mathrm{FB}}_{\mathrm{SOC}} = \mathbf{0}$ or $\mathsf{N}_{\mathsf{itermax}}$ reached
  $\mathsf{k} \leftarrow \mathbf{0}$; $\Phi_{\mathsf{best}} \leftarrow +\infty$
  while $\mathsf{k} \leq \mathsf{N}_{\mathsf{itermax}}$ **do** *// Newton loop*
    // Compute f^{FB}_{SOC} and its Jacobian at $\mathbf{r}^{\mathsf{k}}$ (see Appendix A.1)
    $(\mathsf{f}_{\mathrm{FB}}, \mathsf{J}_{\mathrm{FB}}) \leftarrow \mathsf{computeFBandJacFB}(\mathbf{W}, \mathbf{b}, \mu, \mathbf{r}^{\mathsf{k}})$
    $\phi^{\mathsf{k}} \leftarrow \frac{1}{2}\|\mathsf{f}_{\mathrm{FB}}\|^2$
    if $\phi^{\mathsf{k}} < \phi_{\mathsf{best}}$ **then** *// Check quality of current estimate*
      if $\phi^{\mathsf{k}} \leq \mathsf{tol}$ **then**
        return $\mathbf{r}^{\mathsf{k}}$
      end if
      $\mathbf{r}_{\mathsf{best}} \leftarrow \mathbf{r}^{\mathsf{k}}$; $\phi_{\mathsf{best}} \leftarrow \phi^{\mathsf{k}}$
    end if
    $\mathsf{d}\Phi \leftarrow \mathsf{J}_{\mathrm{FB}}^{\top}\,\mathsf{f}_{\mathrm{FB}}$ *// Compute gradient of objective function*
    // Compute new step $\mathsf{d}\mathbf{r}$
    if $\det \mathsf{J}_{\mathrm{FB}} \neq 0$ **then**
      $\mathsf{d}\mathbf{r} \leftarrow -\mathsf{J}_{\mathrm{FB}}^{-1}\,\mathsf{f}_{\mathrm{FB}}$
      // Check if new step sufficiently decreases the objective function
      if $\mathsf{d}\mathbf{r}.\mathsf{d}\Phi > -\sigma\|\mathsf{d}\mathbf{r}\|\|\mathsf{d}\Phi\|$ **then**
        $\mathsf{d}\mathbf{r} \leftarrow -\mathsf{d}\Phi$
      end if
    else
      $\mathsf{d}\mathbf{r} \leftarrow -\mathsf{d}\Phi$
    end if
    $\mathbf{r}^{\mathsf{k}+1} \leftarrow \mathbf{r}^{\mathsf{k}} + \mathsf{d}\mathbf{r}$
    $\mathsf{k} \leftarrow \mathsf{k} + 1$
  end while
  return $\mathsf{r}_{\mathsf{best}}$

B Enumerative solver

B.1 Quartic polynomial $P(\alpha)$

Let us decompose $\mathbf{W}$ and $\mathbf{b}$ as

$$\mathbf{W} := \begin{pmatrix} \mathrm{w}_{\mathrm{N}} & \mathbf{w}_{\mathrm{T}} \\ \mathbf{w}_{\mathrm{T}} & \mathbf{W}_{\mathrm{T}} \end{pmatrix} \quad \text{and} \quad \mathbf{b} := (\mathrm{b}_{\mathrm{N}}, \mathbf{b}_{\mathrm{T}})^{\top}.$$

We define

$$\begin{array}{llllll} \overline{\mathbf{W}} := & \mathbf{W}_{\mathrm{T}} - \frac{1}{\mathrm{w}_{\mathrm{N}}}\mathbf{w}_{\mathrm{T}}\mathbf{w}_{\mathrm{T}}^{\top} & \mathrm{A} := & \mathbf{tr}\overline{\mathbf{W}} - \mathbf{w}_{\mathrm{T}}^{\top}\overline{\mathbf{b}} & \mathrm{C} := & \det\overline{\mathbf{W}} - \mathbf{w}_{\mathrm{T}}^{\top}\mathrm{B} \\ \overline{\mathbf{b}} := & \frac{\mathbf{b}_{\mathrm{T}}}{\mathrm{b}_{\mathrm{N}}} - \frac{\mathbf{w}_{\mathrm{T}}}{\mathrm{w}_{\mathrm{N}}} & \mathbf{B} := & \left(\mathrm{adj}\overline{\mathbf{W}}\right)\overline{\mathbf{b}} & \mathrm{D} := & \left(\frac{\mathrm{w}_{\mathrm{N}}}{\mu}\right)^2. \end{array}$$

Then we have

$$P(\alpha) = \alpha^4 + 2\mathrm{A}\alpha^3 + \left(2\mathrm{C} + \mathrm{A}^2 - \mathrm{D}\overline{\mathbf{b}}^2\right)\alpha^2 + 2\left(\mathrm{CA} - \mathrm{D}\overline{\mathbf{b}}^{\top}\mathbf{B}\right)\alpha + \mathrm{C}^2 - \mathrm{D}\mathbf{B}^2. \tag{16}$$

B.2 Algorithm

Require: Problem in its reduced form: $\mathbf{W}$, $\mathbf{b}$, μ
Ensure: $\mathsf{r}^{\mathsf{sol}}$ is a solution of the problem, if any exists
  if $\mathrm{b}_{\mathrm{N}} \geq 0$ **then** *// Check take-off case*
    $\mathsf{r}^{\mathsf{sol}} \leftarrow \mathbf{0}$
  else *// Sticking case*
    $\mathsf{G} \leftarrow \left\{\mathbf{r} \in \mathbb{R}^{\mathrm{d}} \text{ s.t. } \mathbf{W}\mathbf{r} = -\mathbf{b}\right\}$
    if $\exists \mathbf{r}^* \in \mathsf{G} \cap \mathbf{K}_{\mu}$ **then**
      $\mathsf{r}^{\mathsf{sol}} \leftarrow \mathbf{r}^*$
    else *// Sliding case*
      if $\mu = 0$ **then** *// Handle frictionless case*
        $\mathsf{r}^{\mathsf{sol}}_{\mathrm{T}} \leftarrow \mathbf{0}$; $\mathsf{r}^{\mathsf{sol}}_{\mathrm{N}} \leftarrow -\frac{\mathrm{b}_{\mathrm{N}}}{\mathrm{W}_{00}}$
      else *// Sliding case, with friction*
        $\mathsf{P} \leftarrow$ quartic polynomial (see Eq 16)
        $\mathsf{S} \leftarrow \{\alpha \in \mathbb{R}^{+*}\ \mathsf{P}(\alpha) = 0\}$
        if $\mathsf{S} = \emptyset$ **then**
          return NO_SOLUTION
        end if
        Choose $\alpha \in \mathsf{S}$ and compute $\mathsf{r}^{\mathsf{sol}}$
      end if
    end if
  end if
  return SOLUTION_FOUND