HAL
open science

# A Nonsmooth Newton Solver for Capturing Exact Coulomb Friction in Fiber Assemblies

Florence Bertails-Descoubes, Florent Cadoux, Gilles Daviet, Vincent Acary

## To cite this version:

Florence Bertails-Descoubes, Florent Cadoux, Gilles Daviet, Vincent Acary. A Nonsmooth Newton Solver for Capturing Exact Coulomb Friction in Fiber Assemblies. ACM Transactions on Graphics, 2011, 30 (1), pp.Article No. 6. 10.1145/1899404.1899410 . inria-00557706

**HAL Id: inria-00557706**

**https://inria.hal.science/inria-00557706**

Submitted on 24 Jan 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 Nonsmooth Newton Solver for Capturing Exact Coulomb Friction in Fiber Assemblies

FLORENCE BERTAILS-DESCOUBES and FLORENT CADOUX and GILLES DAVIET and VINCENT ACARY
INRIA, Grenoble, FRANCE

We focus on the challenging problem of simulating thin elastic rods in contact, in the presence of friction. Most previous approaches in computer graphics rely on a linear complementarity formulation for handling contact in a stable way, and approximate Coulombs's friction law for making the problem tractable. In contrast, following the seminal work by Alart and Curnier in contact mechanics, we simultaneously model contact and exact Coulomb friction as a *zero finding problem of a nonsmooth function*. A semi-implicit time-stepping scheme is then employed to discretize the dynamics of rods constrained by frictional contact: this leads to a set of linear equations subject to an equality constraint involving a non-differentiable function. To solve this one-step problem we introduce a simple and practical nonsmooth Newton algorithm, which proves to be reasonably efficient and robust for systems that are not over-constrained. We show that our method is able to finely capture the subtle effects that occur when thin elastic rods with various geometries enter into contact, such as stick-slip instabilities in free configurations, entangling curls, resting contacts in braid-like structures, or the formation of tight knots under large constraints. Our method can be viewed as a first step towards the accurate modeling of dynamic fibrous materials.

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

General Terms: Modeling, Simulation

Additional Key Words and Phrases: Contact, Coulomb friction, dynamics of thin elastic rods, constraint-based method, knot tying, hair simulation

©ACM, 2011. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version will be published in ACM Transactions on Graphics, January 2011.

## 1. INTRODUCTION

### 1.1 Motivation

Objects composed of thin deformable rods in contact are widely spread in the real world: hair, wool, entangled ropes or wires, knots in suture strands, etc., all fall into this category. Simulating such systems is particularly challenging, for three main reasons: first, finding a robust model for an individual strand that properly captures the important modes of deformation - bending and twisting - is known to be a difficult problem, mainly due to the stiff, high-order equations that characterize such a system. Second, resolving the multiple impacts and resting contacts occurring within a single entangled rope or an assembly of fibers is complex, and made even more difficult by the slender geometry of individual fibers. This calls for the use of extremely robust methods both for collision detection and response. Third, capturing the typical stick-slip effects, or tangles and knots that often occur in fibrous materials (see Figure 1), requires a realistic, nonsmooth model for friction.

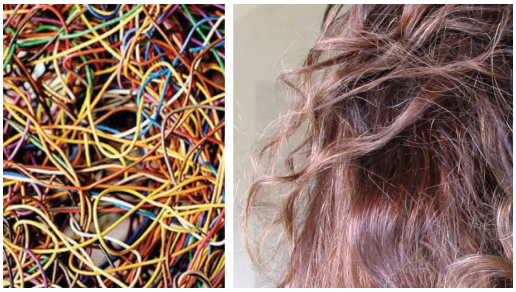

Fig. 1. Typical configurations observable in real assemblies of thin strands. Left: Tangled wires forming loops and knots. Right: hairs wisps sticking onto another hair layer due to dry friction.

Recently, a number of successful models for the dynamics of thin elastic rods (also referred to as "strands") were proposed in the computer graphics (CG) community [Bertails et al. 2006; Hadap 2006; Spillmann and Teschner 2007; Theetten et al. 2008; Bergou et al. 2008; Selle et al. 2008]. In this paper, we focus on the specific problem of the *contact and friction response* applied to thin elastic rods. This topic was hardly addressed in the past, because of the complexity of such a problem and the inability of classical methods to bring satisfying solutions. We propose here a first step towards the realistic modeling of dynamic rods subject to frictional contact.

### 1.2 Related work

We briefly review existing models for thin elastic rods before presenting the main approaches for simulating contact and friction in the general case of interacting (rigid or deformable) bodies. Finally, we summarize the different techniques that have been employed for simulating contact and friction in the case of thin elastic rods.

1.2.1 *Modeling thin elastic rods.* Models for thin elastic rods can be categorized into two distinct families: *maximal-coordinates* and *reduced-coordinates* models.

Maximal-coordinates models generally parameterize the centerline of the rod explicitly as a sequence of 3d space points, and formulate extra constraints to enforce the kinematics of the rod [Rosenblum et al. 1991; Lenoir et al. 2004; Choe et al. 2005; Spillmann and Teschner 2007; Bergou et al. 2008; Selle et al. 2008]. These approaches are popular because finding maximal-coordinates is simple and often leads to a sparse, block diagonal mass matrix. However, formulating the extra constraints can be challenging, especially when one wants to account for the inextensibility of the rod: adding stiff terms can lead to stability issues [Rosenblum et al. 1991; Spillmann and Teschner 2007; Selle et al. 2008]. Another

difficulty stems from the numerical drifts and energy loss when enforcing these constraints over time [Bergou et al. 2008].

In contrast, reduced-coordinates models parameterize a rod using its actual degrees of freedom [Hadap and Magnenat-Thalmann 2001; Hadap 2006; Bertails et al. 2006; Bertails 2009]. Such parameters often express a local rotation of the system, *e.g.*, joint quaternions in the case of the articulated rigid bodies system [Hadap and Magnenat-Thalmann 2001; Hadap 2006] and material curvatures and twist in the case of the Super-helix model [Bertails et al. 2006; Bertails 2009]. The parameterization is thus guaranteed to be minimal, with no redundant variables, and the kinematic constraints are always exactly maintained over time. In this kind of formulation however, the centerline of the rod has to be computed recursively from root to tip, as a nonlinear function of the parameters.

An argument that is often raised against reduced-coordinates models is the (apparent) difficulty to handle contact and friction with a non-explicit centerline [Bergou et al. 2008; Selle et al. 2008]. In this paper however, we show that reduced-coordinates models perfectly fit into our frictional contact formulation, with no additional cost. We present results using the Super-helix model as our rod model. To demonstrate the versatility of our approach, we have also performed simulations using maximal-coordinates rod models such as an implicit mass-spring system [Baraff and Witkin 1998] as well as the more recent CORDE model [Spillmann and Teschner 2007]. We demonstrate that the convergence of our contact algorithm is insensitive to the nature of parameterization of the rod models, but rather depends on the *level of constraining* of the dynamical system.

1.2.2 *Modeling contact and friction.* Modeling contact and friction has been an active research area in CG for several decades. Because of the large number of techniques employed and the wide spectrum of targeted applications (ranging from rigid bodies to 3D deformable objects, cloth, thin shells, and rods), it can be somewhat difficult for a non-expert reader to find a path into this intricate field. We try here to offer a synthetic view on the topic by classifying previous work into three main categories: *penalty-based*, *explicit constraint-based*, and *implicit constraint-based* methods. Note that we focus here on the modeling of the *response* to collision. For a recent survey on collision detection methods, we refer the reader to [Teschner et al. 2005].

**Penalty-based methods:** One common way of preventing interpenetration between colliding objects consists in adding mutual repulsive forces of the form $k\mathbf{f}(\delta)$, where $\delta$ is the penetration depth detected at current time step [Moore and Wilhelms 1988]. Though simple to implement and efficient, this method often fails to prevent excessive penetration if the stiffness factor $k$ is not large enough, unless barrier functions are used [Kaldor et al. 2008; Harmon et al. 2009]. However most barrier functions suffer from unbounded second derivatives which ruin the stability of the fixed step integration schemes. To circumvent this problem, Harmon et al. [2009] recently proposed the use of discrete penalty layers coupled with an asynchronous symplectic solver, at the price of losing computational efficiency. Another issue inherent to penalty-based approaches is the introduction of parasitical high frequencies due to the recourse to large stiffness values for preventing interpenetration [Baraff 1989]. Penalty-based approaches are thus generally not satisfying for robust contact handling, however they remain widely used in situations where time performance is a crucial criterion [Barbič and James 2007].

In the same vein, friction is often modeled using simple viscous forces, especially in the case of self-contact within deformable objects, for the sake of simplicity [Baraff and Witkin 1998; Choe et al. 2005; Spillmann and Teschner 2008; Kaldor et al. 2008]. Such a model cannot capture the threshold effect that characterizes friction between solids.

**Explicit constraint-based methods:** Instead of adding some forces that will hopefully lead to a collision-free state, an alternate way consists in deciding *in advance* what positions and velocities should be at next time step. To enforce the desired state, some authors propose to directly alter positions and/or velocities at the end of the time step [Baraff and Witkin 1998; Müller et al. 2007] while others compute the corresponding forces that will exactly lead to the desired, collision-free state [Spillmann et al. 2007; Spillmann and Teschner 2008]. Because of its simplicity and efficiency, this kind of approach has become very popular in the CG community for resolving contact in the case of systems parameterized by sample 3d points, such as 3d point-based deformable bodies [Müller et al. 2007], cloth [Baraff and Witkin 1998; Bridson et al. 2002], and hair [Choe et al. 2005; Selle et al. 2008]. However, this kind of technique is hardly applicable to systems that are subject to multiple impacts (in this case the desired change in velocity may be unclear), as well as to sophisticated models such as reduced-coordinates models, where the 3d positions at the surface of the system are not necessarily linear functions of the degrees of freedom.

In a similar spirit, Coulomb friction is often modeled explicitly, *i.e.*, the next state of the objects (take-off, stick, slip) and the sliding direction are determined in advance by considering the contact forces and relative velocities at the current time step [Baraff 1994; Bridson et al. 2002; Selle et al. 2008]. Again, this approach is simple, but as it does not guarantee that the Coulomb conditions are satisfied at the end of each time step, it may suffer from drifts over time and thus be inaccurate. Moreover, to be able to capture perfect dry friction, this method requires the ability to directly control the change in velocity and position of the contacting points of the system. For most systems, which are not controllable without altering their own dynamics, this is seldom achievable.

**Implicit constraint-based methods:** A last solution is to consider that the relative velocities and directions of friction at next time step are part of the *unknowns* of the problem. The dynamic system, augmented by contact variables, has then to be solved, traditionally using a fixed point method or an optimization algorithm. Computing contact and friction implicitly is the only way of strictly enforcing the proper conditions at the end of each time step, without the need for manual checking and readjustment.

Robust approaches have been provided for handling contact without friction, mostly relying on linear complementarity problems (LCP) [Baraff 1989; Baraff and Witkin 1992]. However, simulating contact with Coulomb friction in 3d remains a challenging issue because non-linearity has to be introduced into the equations. To overcome this issue, most approaches in CG propose to approximate Coulomb's friction law. Some authors model friction using a Tresca-like law where the sliding threshold is an arbitrary value independent of the normal force [Lötstedt 1984]. This model is not satisfying in many common situations. Consider for example two different solids lying on a horizontal plane. If an increasing, horizontal traction force is applied to them, both of them will start to slide exactly at the same time, no matter what their mass is. In [Kaufman et al. 2005], Coulomb friction is approximated by considering that the normal component of the reaction force

is known from a first computational step without friction. This procedure allows them to retrieve a standard convex QP and can be viewed as a Tresca-like approximation at each time-step. To compute hair-body frictional contact, Hadap [2006] replaces the Coulomb law with a new model based on an LCP. However, the model fails to capture dry friction properly: for a contact force in the interior of the friction cone, the relative velocity is not constrained to be zero. More recent approaches start with the Coulomb law equations and linearize the friction cone to come up with a set of LCP to be solved [Erleben 2007; Kaufman et al. 2008; Otaduy et al. 2009]. While this approximate formulation captures both dynamic and static friction properly, it introduces some anisotropy in the sliding direction unless a high number of facets is used [Acary and Brogliato 2008, Sec. 13.3.7]. A trade-off has thus to be made between accuracy and computational cost.

It is overall surprising that no method modeling exact Coulomb friction has become popular in the graphics community so far. Yet, such methods have been developed and successfully used for years in the contact mechanics community for accurately simulating multi-body systems involving frictional contact, such as granular materials [Moreau 1994; Jean 1999]. We believe the main reason why such methods have not caught on yet in the graphics community is the apparent difficulty of implementation for people who are not experts in nonsmooth optimization. In this paper, we demonstrate that the frictional contact problem can be elegantly expressed as a zero finding problem, and provided the system is not over-constrained, be simply solved using a Newton algorithm - without resorting to a complementarity formulation. Our algorithm is furthermore very easy to understand and implement, and does not rely on any black box optimization code. To help researchers test our approach and compare their methods with ours, we freely provide on our webpage the source code for our solver, coupled with the simple interactive double loop example illustrated in Figure 9.

We note that Duriez *et al.* [2006] made a first attempt to model exact Coulomb friction for the real-time simulation of frictional contact for haptics. Their approach relies on a so-called Gauss-Seidel algorithm that iteratively solves each single frictional contact while other contacts are frozen: for each contact, the contact force is computed by projection onto the (non-polyhedral) friction cone. The authors show that their method is more efficient than methods relying on a linearized friction cone for which the number of facets is sufficiently large to ensure a given precision. In their approach however, the projection step onto the friction cone is significantly simplified as the local Delassus operator is roughly approximated by a diagonal matrix. This approximation has no incidence when applied to isotropic objects such as spheres, but it may give rise to important drifts when dealing with general, anisotropic objects, for which the tangential velocity during dynamic friction not only depends on the applied external force, but also on the geometry of the object. Another concern of [Duriez et al. 2006] is that the formulations used for contact and friction laws are not velocity-based, but rely on the normal and tangential gaps between contacting elements. Such acceleration-force formulations have been shown to be inconsistent as they do not necessarily yield a solution [Baraff 1993; Stewart 2000]. In contrast, impulse-velocity formulations for frictional problems turn out to have better properties of convergence [Marques 1993; Stewart 2000], and as such, are often preferred.

Unlike [Duriez et al. 2006], our approach provides a velocity-based formulation for the frictional contact, and relies on a simple, functional formulation of the *exact* Coulomb friction model - initially due to Alart and Curnier [1991] - which is subsequently solved using a nonsmooth Newton algorithm. Moreover, our approach *simultaneously* solves all frictional contact problems, without relying on a Gauss-Seidel like algorithm. In our results section, we show that provided our dynamic system is not excessively constrained, we obtain satisfying convergence results that guarantee a good visual precision of our simulations.

1.2.3 *Self-contact and friction in rods.* Due to the number of applications involved, the study and simulation of self-contact in an assembly of thin elastic rods has increasingly drawn attention from researchers in both mechanics and computer graphics.

In both fields, most Lagrangian approaches for simulating contact between strands or groups of strands are penalty-based [Plante et al. 2001; Durville 2004; Choe et al. 2005; Bertails et al. 2006] or based on explicit constraints [Spillmann and Teschner 2007; 2008; Bergou et al. 2008; Selle et al. 2008]. In all these methods, friction is simply handled through a viscous term. These methods often suffer from instability or inaccuracy issues, and are unable to capture subtle frictional effects with threshold, such as stick-slip effects. To simplify the problem of self-contact in large assemblies of strands (*e.g.*, hair), some approaches resort to procedural modeling for capturing typical emerging phenomena, such as hair clumping, either by constraining hair to move as a predetermined set of wisps [Plante et al. 2001; Bertails et al. 2006], or by adding some attractive forces to capture stiction [Selle et al. 2008]. Hadap and Thalmann [2001] were the first to explore continuum laws for modeling hair self-contact. Very recently, [McAdams et al. 2009] combined a Lagrangian and an Eulerian representations for hair to bring some discontinuous details to the simulation. Although spectacular, their results do not capture the typical stick-slip effects that would be visible within a hair clump coming to rest.

From the state of the art, it actually turns out that whatever the scale of the targeted applications is (a knot in a single rod, hair with a hundred guide strands, hair with thousands of simulated strands), robust contact handling with accurate friction modeling remains a *constant missing element*. We propose a first solution to this challenging problem. Our method is related to implicit constraint-based models, which, to the best of our knowledge, were never explored for solving self-contacts within an assembly of fibers. Furthermore, we introduce a velocity-based model for simulating exact Coulomb friction. This allows us to design a pure physics-based model, able to capture for the first time typical contacting phenomena that occur within an assembly of strands.

## 1.3 Contributions

—We provide a compact formulation of frictional contact for thin elastic rods. Unlike previous approaches, our formulation exactly models the Coulomb friction law as a zero finding problem of a nonsmooth function, which turns out to be much simpler and efficient to solve compared to a nonlinear complementary formulation. Our approach is inspired by the theoretical work by Alart and Curnier [1991], which, to the best of our knowledge, was never exploited in CG to solve frictional contact.

—We give a practical implementation for solving the optimization problem resulting from our frictional contact model. Our method combines a nonsmooth Newton step with an accurate line search in order to stabilize the algorithm and accelerate its convergence. Although theoretical convergence properties are

Fig. 2. Various simulations of frictional contact in thin rods. From left to right and top to bottom: interactive formation of a tight knot and a plectoneme, simulation of 76 spaghetti falling in a plate, complex hair contacts with Coulomb friction in wavy hair (84 simulated strands).

lost in the nonsmooth case, we provide a simple experimental criterion, based on the degree of constraining of the system, to characterize the quality of convergence. When systems are not over-constrained, our solver yields fast convergence at tolerances suitable for graphics purposes and we show that many fiber systems typically lie in the favorable case.

—We demonstrate the versatility of our approach by performing simulations on various representative rod models: an implicit mass-spring system, the more elaborate CORDE model (maximal-coordinates), and the Super-helix model (reduced-coordinates). To some extent, we also illustrate the effectiveness of our approach on other dynamical systems, such as rigid bodies, while clearly defining the class of systems to which our contact method is the most adapted.

—We show that we can robustly simulate systems with large contact forces, such as tight knot tying, as well as capture subtle collective behaviors in entangled fibrous materials, such as hair, by using a pure Lagrangian method. To the best of our knowledge, this is one of the first attempts to accurately model contact and friction within free assemblies of fibers, regarding both CG and computational mechanics fields.

### 1.4 Organization

Our paper is organized as follows. Section 2 introduces a functional formulation for Coulomb's friction model, derived from the early work by Alart and Curnier. In Section 3, we formulate the discrete problem of the constrained dynamics that we wish to solve at each time step, and provide a practical solving algorithm in Section 4. Finally, results of our approach are presented in Section 5 and discussed in Section 6, before concluding.

## 2. MODELING CONTACT WITH PERFECT COULOMB FRICTION

The friction model proposed by Charles de Coulomb around 1770-1780 is a macroscopic model which captures a crucial physical phenomenon of friction: the threshold of sliding. This phenomenon is particularly visible on fibrous materials, such as hair, where fibers often get stuck onto each other, giving rise to static configurations with multiple fibers orientations, or to the spontaneous formation of entangled wisps (see Figure 1). Though simple in appearance, the Coulomb friction model is actually difficult to take into account exactly as it is a nonsmooth, multi-valued law which cannot be handled using classical mathematical tools of smooth analysis.

Formally, the Coulomb friction law can be described[1] as a set $C \subset \mathbb{R}^3 \times \mathbb{R}^3$ relating the relative velocity $\boldsymbol{u}^i$ and the contact force $\boldsymbol{r}^i$ at contact point $i$: the couple $(\boldsymbol{u}^i, \boldsymbol{r}^i)$ satisfies the Coulomb friction law if and only if $(\boldsymbol{u}^i, \boldsymbol{r}^i) \in C$. In the sequel, after defining some notations, we shall give two equivalent formulations of $C$: on the one hand, the classical disjunctive formulation, easy to understand but hardly tractable in practice; on the other hand, a zero finding formulation, initially due to Alart and Curnier [1991], which gives rise to a simple robust algorithm that we derive in Section 4.

### 2.1 Local forces and velocities

Let us consider a mechanical system in 3d space involving $n$ contact points at a given instant $t$. Each contact, labeled by $i \in 1, \dots n$, is assumed to involve two and only two contacting bodies, denoted by $A^i$ and $B^i$. We further assume that contacting surfaces are sufficiently smooth, so that a normal vector $\mathbf{e}^i$ (see Figure 3) can be defined. Taking body $B^i$ as a reference, consider the relative velocity $\boldsymbol{u}^i$ of $A^i$ with respect to $B^i$ and the force $\boldsymbol{r}^i$ applied by $B^i$ onto $A^i$. We note $x^i_{\mathrm{N}} = \boldsymbol{x}^i \cdot \boldsymbol{e}^i$ the (scalar) component of a vector $\boldsymbol{x}^i$, and $\boldsymbol{x}^i_{\mathrm{T}} = \boldsymbol{x}^i - x^i_{\mathrm{N}} \mathbf{e}^i$ the tangential (vector) part of $\boldsymbol{x}^i$.

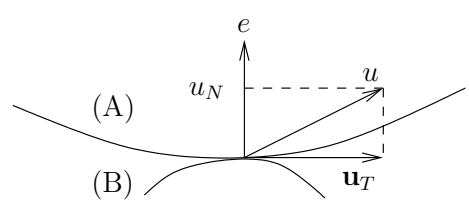


Fig. 3. Body $A^i$ and $B^i$ with tangent and normal spaces

### 2.2 Disjunctive formulation

Friction coefficients $\mu^i$ $(0 \leq \mu^i \leq \mu_{\max})$ being given, let us define the second-order cone with coefficient $\mu^i$ (the friction cone) by

$$K_{\mu^i} := \{\|\boldsymbol{x}_{\mathrm{T}}\| \leq \mu^i x_{\mathrm{N}}\} \subset \mathbb{R}^3. \quad (1)$$

[1]This formalism actually applies to all friction laws. For example, the viscous friction law can be formulated by defining the set $C$ as a vector space. In the case of Coulomb friction, $C$ obviously has a much more complex structure.

As illustrated in Figure 4, Coulomb's law was originally formulated as a *disjunction* of three cases: that is, $(\boldsymbol{u}^i, \boldsymbol{r}^i) \in C(\mathbf{e}^i, \mu^i)$ if and only if one of the three following cases[2,3] occurs for each $i \in 1, \dots n$:

—take-off : $\boldsymbol{r}^i = 0$ and $u^i_{\mathrm{N}} \geq 0$,

—stick : $\boldsymbol{r}^i \in K_{\mu^i}$ and $\boldsymbol{u}^i = 0$,

—slide : $\boldsymbol{r}^i \in \partial K_{\mu^i} \setminus 0,\ u^i_{\mathrm{N}} = 0,\ \boldsymbol{u}^i_{\mathrm{T}} \neq \mathbf{0}$ opposed to $\boldsymbol{r}^i_{\mathrm{T}}$:
$\exists \alpha^i > 0, \boldsymbol{r}^i_{\mathrm{T}} = -\alpha^i \boldsymbol{u}^i_{\mathrm{T}}$.

Coulomb's law can be interpreted as follows: if the first case, the normal relative velocity is nonnegative and the contact breaks (take-off case). Then there can be no force between the two bodies; this models dry friction without adherence. Note that rebound is handled separately through a restitution coefficient (Section 3). The two bodies can also remain in resting contact, then the contact force can lie anywhere in $K_{\mu^i}$. Finally, if the two bodies are sliding with respect to each other with tangent relative velocity, then the contact force must belong to the boundary of $K_{\mu^i}$ and the tangent force must be collinear to the relative velocity with the opposite direction (according to the "maximum dissipation principle" [Moreau 1988]). The value of $\mu^i$ depends on the characteristics of the contacting surfaces (from $\mu^i = 0$ for perfect contact without friction to $\mu^i = \mu_{\max}$ for rough surfaces).

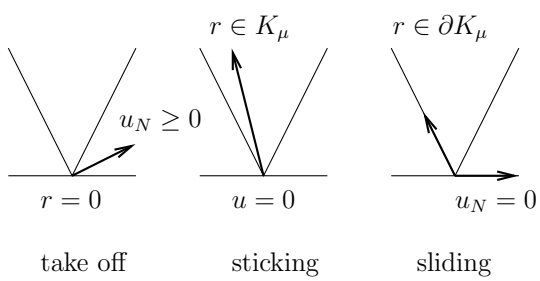


Fig. 4. The three cases of Coulomb's law

This disjunctive formulation is intuitive, but not very practical because of its combinatorial nature (there are $3^n$ cases to check, if the system contains $n$ contacts); in the next subsection, we give an equivalent, but more tractable, formulation of Coulomb's law.

### 2.3 Functional formulation

In this subsection we shall omit the upper script $i$ of contact variables, for the sake of clarity. Alart and Curnier [1991] were the first to demonstrate that it is actually possible to give a functional characterization of the Coulomb friction law, *i.e.*, to find a function $\boldsymbol{f}^{\mathrm{AC}}$ of $\mathbb{R}^3 \times \mathbb{R}^3$ in $\mathbb{R}^3$ such that

$$(\boldsymbol{u}, \boldsymbol{r}) \in C(\mathbf{e}, \mu) \iff \boldsymbol{f}^{\mathrm{AC}}(\boldsymbol{u}, \boldsymbol{r}) = \mathbf{0}. \tag{2}$$

The function $\boldsymbol{f}^{\mathrm{AC}}$ proposed by Alart and Curnier to satisfy this equivalence is defined as

$$\boldsymbol{f}^{\mathrm{AC}}(\boldsymbol{u}, \boldsymbol{r}) := \begin{bmatrix} f^{\mathrm{AC}}_{\mathrm{N}}(\boldsymbol{u}, \boldsymbol{r}) \\ \boldsymbol{f}^{\mathrm{AC}}_{\mathrm{T}}(\boldsymbol{u}, \boldsymbol{r}) \end{bmatrix} \tag{3}$$

where the so-called *normal* and *tangential* parts of $f$ respectively read

$$f^{\mathrm{AC}}_{\mathrm{N}} : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^3 \longrightarrow \mathbb{R} \\ (\boldsymbol{u}, \boldsymbol{r}) \quad \longmapsto \mathrm{P}_{\mathbb{R}^+}(r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) - r_{\mathrm{N}} \end{cases} \tag{4a}$$

and

$$\boldsymbol{f}^{\mathrm{AC}}_{\mathrm{T}} : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^3 \longrightarrow \mathbb{R}^2 \\ (\boldsymbol{u}, \boldsymbol{r}) \quad \longmapsto \mathrm{P}_{\boldsymbol{B}(\mathbf{0}, \mu r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}} - \rho_{\mathrm{T}} \boldsymbol{u}_{\mathrm{T}}) - \boldsymbol{r}_{\mathrm{T}} \end{cases}, \tag{4b}$$

where $\rho_{\mathrm{N}}$ and $\rho_{\mathrm{T}}$ are two positive constants ($\rho_{\mathrm{N}} = \rho_{\mathrm{T}} = 1$ for instance), $\boldsymbol{B}(\mathbf{0}, \delta) \subset \mathbb{R}^2$ is the ball centered in $\mathbf{0}$ of radius $\delta$, and $\mathrm{P}_K$ is the projection function (applying to any vector of dimension 1 or 2) onto the convex set $K$ (dim $K \leq 2$), with the convention that

$$\forall \mathbf{y} \in \mathbb{R}^2,\ \mathrm{P}_{\emptyset}(\mathbf{y}) = \mathbf{0}_{\mathbb{R}^2}.$$

We give the proof for equivalence (2) in appendix A. It is fairly simple and we invite the reader to check the equivalence by himself so that he can fully understand the meaning and the power of the Alart-Curnier function with respect to Coulomb's friction law.

## 3. FORMULATION OF THE ONE-STEP PROBLEM

Let us consider a set of $N_{\mathrm{s}}$ strands (modeled as implicit mass-spring chains, CORDEs, or Super-helices). The mass-matrix $\mathbf{M}$ of this system is diagonal-block where each block $i$ contains the mass-matrix $\mathbf{M}_i$ of the $i^{\mathrm{th}}$ strand. Let $m$ be the total number of degrees of freedom of the system, and $\dot{\mathbf{q}} \in \mathbb{R}^m$ the generalized velocities of the mechanical system, resulting from the concatenation of the generalized velocities of the $N_{\mathrm{s}}$ individual strands. Let $\mathbf{f}$ collect the set of (internal and external) forces applied onto the total system. $\mathbf{J}$ denotes the gradient that relates the local velocity of the mechanical system to its global velocity, and its calculation is explained in Section 4.1.

We wish to solve the following problem in the unknowns $(\dot{\mathbf{q}}, \mathbf{u}, \mathbf{r})$, where $\mathbf{u} \in \mathbb{R}^{3n}$ gathers all the relative 3d velocities $\boldsymbol{u}^i$ at contact points, and $\mathbf{r} \in \mathbb{R}^{3n}$ collects the 3d contact forces $\boldsymbol{r}^i$:

$$\begin{cases} \mathbf{M}(\mathbf{q}) \dfrac{\mathrm{d}\dot{\mathbf{q}}}{\mathrm{d}t} + \mathbf{f} = \mathbf{J}^\top \mathbf{r} \\ \mathbf{u} = \mathbf{J}\dot{\mathbf{q}} + \mathbf{w} \\ \forall i = 1 \dots n,\ (\boldsymbol{u}^i - c^i \mathbf{e}^i, \boldsymbol{r}^i) \in C(\mathbf{e}^i, \mu^i). \end{cases} \tag{5}$$

The set $C(\mathbf{e}^i, \mu^i)$ contains the couples $(\boldsymbol{u}^i, \boldsymbol{r}^i)$ satisfying Coulomb's law for a given value of the friction coefficient $\mu^i$ and a normal vector $\boldsymbol{e}^i$ at the $i$-th contact point. The constant $c^i \geq 0$ accounts for impacts: after a shock, Coulomb's law imposes that the normal part of $\boldsymbol{u}^i - c^i \mathbf{e}^i$ be zero, which implies that the norm of the normal relative velocity $u^i_{\mathrm{N}}$ takes the value $c^i$. By setting $c^i$ equal to a constant $k$ (with $k \in [0, 1]$) times the normal velocity *before* the shock, we

[2]Note that unlike most approaches, we compactly formulate a frictional contact law that both includes the pure contact law (namely the *Signorini conditions* [Moreau 1988]) and the pure Coulomb friction law.

[3]The Signorini law classically relates contact forces to interpenetration distances. Here we model contact at a higher order, using a velocity-based formulation. Moreau proved that in the continuous case, the velocity-based formulation of constraints implies the Signorini conditions (result known as the "viability lemma") [Moreau 1988]. As we already mentioned in the introduction, it has been experienced that in the numerical case, and when friction has to be accounted for, the velocity-based formulation conducts to more stable simulations [Stewart 2000].

obtain Newton's impact law with $k=0$ modeling inelastic impacts and $k=1$ modelling perfect rebound. In practice we model impacts between thin strands as merely inelastic, thus choosing $c^i=0$. For impacts between a heavy rod and a rigid body however, it may be suitable to set $c^i$ to a small non-zero value to allow for some rebound.

Solving the initial problem (5) amounts to solving

$$\begin{cases} \mathbf{M}(\mathbf{q})\,\dfrac{\mathrm{d}\dot{\mathbf{q}}}{\mathrm{d}t} - \mathbf{J}^\top\,\mathbf{r} + \mathbf{f} = \mathbf{0} \\ \boldsymbol{f}^{\mathrm{AC}}(\mathbf{J}\dot{\mathbf{q}} + \mathbf{w} - \mathbf{c}_e, \mathbf{r}) = \mathbf{0} \end{cases} \tag{6}$$

where $\mathbf{c}_e$ is a constant vector containing the terms $c^i\,\boldsymbol{e}^i$ from (5) and $\boldsymbol{f}^{\mathrm{AC}}$ collects the Alart-Curnier function (3) at all contact points. Using an arbitrary step length $\delta t$, we follow Moreau's time-stepping scheme for integrating nonsmooth problems [Moreau 1988]: the velocity derivative $\frac{\mathrm{d}\dot{\mathbf{q}}}{\mathrm{d}t}$ is replaced with the velocity jump between the end and the start of the timestep, the position derivative $\dot{\mathbf{q}}$ is approximated at first order, $\mathbf{r}$ and $\mathbf{u}$ are discretized using an *implicit* scheme (since brutal changes in their values are expected due to impacts) and the other terms[4] are discretized explicitly (since they are assumed to be smooth). We denote by $(\underline{\mathbf{v}}, \underline{\lambda})$ the discrete-time approximations of $(\dot{\mathbf{q}}, \mathbf{r})$, where $\underline{\lambda}$ is the discrete impulse $\mathbf{r}_{t+\delta t}\,\delta t$, and by $\underline{\mathbf{M}}$, $\underline{\mathbf{J}}$, $\underline{\mathbf{f}}$ and $\underline{\mathbf{w}}$ (accounting for $\mathbf{w}-\mathbf{c}_e$) the discretized data, where $\underline{\mathbf{f}} = \mathbf{f}_t\,\delta t - \underline{\mathbf{M}}\,\mathbf{v}_t$. We thus get the following one-step problem in the unknowns $(\underline{\mathbf{v}}, \underline{\lambda})$:

$$\begin{cases} \underline{\mathbf{M}}\underline{\mathbf{v}} - \underline{\mathbf{J}}^\top\underline{\lambda} + \underline{\mathbf{f}} &= \mathbf{0} \\ \boldsymbol{f}^{\mathrm{AC}}(\underline{\mathbf{J}}\underline{\mathbf{v}} + \underline{\mathbf{w}}, \underline{\lambda}) &= \mathbf{0}. \end{cases} \tag{7}$$

Our method consists in searching a solution to system (7) using Newton's algorithm (despite the fact that $\boldsymbol{f}^{\mathrm{AC}}$ is nonsmooth). We have formulated our initial problem in terms of impulses and velocities instead of forces and displacements. It is well–known that the impulse/velocity formulation is the only way to consistently integrate time systems with impacts. The price to pay is a formulation of unilateral constraints and Coulomb friction on the impulse level, which means with a time integral of forces when the evolution is smooth enough. For more details on the slight differences between an impulse model of friction and the standard one in forces, we refer to [Frémond 2002; Acary and Brogliato 2008].

## 4. COMPUTATION OF THE CONTACT FORCES

At each time step, we wish to solve the system of equations (7) in the unknowns $(\underline{\mathbf{v}}, \underline{\lambda})$. We first briefly explain the computation of $\underline{\mathbf{J}}$ before reducing the system as a zero finding problem of the function $\boldsymbol{f}^{\mathrm{AC}}$. The solving by Newton's method requires the computation of the Jacobian matrix of $\boldsymbol{f}^{\mathrm{AC}}$, which is described in detail in the following. The final algorithm for solving the frictional contacting problem, together with interpenetration correction and optimization, is finally presented.

[4] Actually, in the case of Super-helices and implicit mass-spring chains, internal elastic forces are computed implicitly. The actual matrix $\mathbf{M}$ of our discretized system (7) is thus $\mathbf{M}+dt^2\,\mathbf{K}$, where $\mathbf{K}$ is the total stiffness matrix of the system. We skip this detail in the description of the algorithm, for the sake of simplicity.

### 4.1 Computation of $\mathbf{J}$

As expressed in equation (5), the gradient matrix $\mathbf{J}$ of size $(3n, m)$ linearly relates the 3d relative velocities $\mathbf{u}$ at the $n$ contact points to the $m$ generalized velocities $\dot{\mathbf{q}}$ of the system. Let us consider the 3d row-block $\mathbf{J}_i$ corresponding to the $i^{\text{th}}$ contact point. As mentioned in Section 2.1, we assume that each contact involves no more than two bodies (or body parts), denoted $A^i$ and $B^i$. Let $\mathbf{r}^{A^i}(s^{A^i}, \mathbf{q}^{A^i})$ ($\mathbf{r}^{B^i}(s^{B^i}, \mathbf{q}^{B^i})$ respectively) be the centerline of the rod corresponding to body $A^i$ (resp. to body $B^i$): it gives the spatial coordinates of the point located at the curvilinear abscissa $s^{A^i}$ (resp. $s^{B^i}$) on the rod $A^i$ (resp. $B^i$) with generalized-coordinates $\mathbf{q}^{A^i}$ (resp. $\mathbf{q}^{B^i}$). At instant $t_c$, $\mathbf{J}_i$ thus reads

$$\mathbf{J}_i(t_c) = \frac{\partial \mathbf{r}^{A^i}}{\partial \mathbf{q}^{A^i}}(s_c^{A^i}, \mathbf{q}^{A^i}(t_c)) - \frac{\partial \mathbf{r}^{B^i}}{\partial \mathbf{q}^{B^i}}(s_c^{A^i}, \mathbf{q}^{A^i}(t_c))$$

where $s_c^{A^i}$ and $s_c^{B^i}$ respectively denote the curvilinear abscissa at contact point for rod $A^i$ and rod $B^i$ respectively. Note that in the case of self-contact, we have $A^i = B^i$ and $s_c^{A^i} \neq s_c^{B^i}$.

The term $\mathbf{J}_i$ can thus be computed by calculating the formal gradient $\frac{\partial \mathbf{r}}{\partial \mathbf{q}}$ of a rod with centerline $\mathbf{r}$ and generalized-coordinates $\mathbf{q}$, and evaluating it at the point of current contact $i$. Such a computation is straightforward for the mass-spring chain and the CORDE model. In the case of the Super-helix model, we relied on the Maple software [MapleSoft 2010] for deriving the gradient analytically.

### 4.2 Elimination of $\underline{\mathbf{v}}$

The problem (7) can be reduced by eliminating $\underline{\mathbf{v}}$ from the equations. Indeed, from (7) there holds

$$\underline{\mathbf{v}} = \underline{\mathbf{M}}^{-1}\underline{\mathbf{J}}^\top\underline{\lambda} - \underline{\mathbf{M}}^{-1}\underline{\mathbf{f}}$$

which implies

$$\underline{\mathbf{u}} = \underbrace{\underline{\mathbf{J}}\underline{\mathbf{M}}^{-1}\underline{\mathbf{J}}^\top}_{=:\underline{\mathbf{W}}}\underline{\lambda} + \underbrace{\underline{\mathbf{w}} - \underline{\mathbf{J}}\underline{\mathbf{M}}^{-1}\underline{\mathbf{f}}}_{=:\underline{\mathbf{b}}} = \underline{\mathbf{W}}\underline{\lambda} + \underline{\mathbf{b}}$$

where $\underline{\mathbf{W}}$ is the so-called Delassus operator. Its computation involves $\underline{\mathbf{M}}^{-1}$: it can be performed column by column using a conjugate gradient algorithm (then $\underline{\mathbf{M}}$ and $\underline{\mathbf{J}}$ need not be assembled), or using the Choleski decomposition of $\underline{\mathbf{M}}$ (which can be reused from a time-step to another if the mass matrix does not change).

After eliminating $\underline{\mathbf{v}}$, the problem becomes

$$\boldsymbol{f}^{\mathrm{AC}}(\underline{\mathbf{W}}\underline{\lambda} + \underline{\mathbf{b}}, \underline{\lambda}) = \mathbf{0}. \tag{8}$$

On the one hand, this new system is smaller than system (7) (size $3n$ instead of $m+3n$) and we exploit the fact that the linear part of system (7) does not change during Newton's iterations by computing $\underline{\mathbf{W}}$ once for all. On the other hand the computation of $\underline{\mathbf{W}}$ may be costly and $\underline{\mathbf{W}}$ may not be as sparse as system (7). The choice of performing this elimination or not is probably problem-dependent. In our experiments, we chose to use it since $\underline{\mathbf{W}}$ was sparse enough. Indeed, despite the fact that the mass matrix and the gradient $\mathbf{J}$ of individual rods are dense, they are sparse and block-structured for the whole system due to the connectivity of the contacts and the bodies (see Section 4.5).

### 4.3 Computing the gradient of $\boldsymbol{f}^{\text{AC}}$

The normal direction $\mathbf{e}^i$ at $i$-th contact point being known, choose $\mathbf{e}^i_S$ and $\mathbf{e}^i_{\text{T}}$ so that $(\mathbf{e}^i, \mathbf{e}^i_S, \mathbf{e}^i_{\text{T}})$ form an orthonormal basis. Define the matrix of the projection onto the normal direction by $\mathbf{P}_{\text{N}} := (\mathbf{e}^i)^\top$ and the matrix of the projection onto the tangent direction by $\mathbf{P}_{\text{T}} := [\mathbf{e}^i_S, \mathbf{e}^i_{\text{T}}]^\top$. We are now able to differentiate $\boldsymbol{f}^{\text{AC}}$. For the sake of simplicity, detailed calculations below are given in the case of a single contact.

**Normal part:** Function $f^{\text{AC}}_{\text{N}}$ defined by (4a) is continuous and affine by pieces : if $r_{\text{N}} - \rho_{\text{N}} u_{\text{N}} < 0$, then

$$f^{\text{AC}}_{\text{N}}(\boldsymbol{u}, \boldsymbol{r}) = -\mathbf{P}_{\text{N}}\boldsymbol{r}, \ \frac{\partial f^{\text{AC}}_{\text{N}}}{\partial \boldsymbol{u}} = \mathbf{0}_{1\times 3}, \ \frac{\partial f^{\text{AC}}_{\text{N}}}{\partial \boldsymbol{r}} = -\mathbf{P}_{\text{N}}$$

whereas, if $r_{\text{N}} - \rho_{\text{N}} u_{\text{N}} > 0$

$$f^{\text{AC}}_{\text{N}}(\boldsymbol{u}, \boldsymbol{r}) = -\rho_{\text{N}}\mathbf{P}_{\text{N}}\boldsymbol{u}, \ \frac{\partial f^{\text{AC}}_{\text{N}}}{\partial \boldsymbol{u}} = -\rho_{\text{N}}\mathbf{P}_{\text{N}}, \ \frac{\partial f^{\text{AC}}_{\text{N}}}{\partial \boldsymbol{r}} = \mathbf{0}_{1\times 3}.$$

**Tangential part:** Define the following function $g$ whose Jacobian matrix, where it exists, is computed in appendix B:

$$\mathbf{g}\colon \begin{cases} \mathbb{R}\times\mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ (\delta, \mathbf{y}) \ \longmapsto \mathrm{P}_{B(\mathbf{0},\delta)}(\mathbf{y}). \end{cases}$$

Then the tangential part $\boldsymbol{f}^{\text{AC}}_{\text{T}}$ of the Alart-Curnier function (4b) is

$$\boldsymbol{f}^{\text{AC}}_{\text{T}}(\boldsymbol{u}, \boldsymbol{r}) = g(\underbrace{\mu\mathbf{P}_{\text{N}}\boldsymbol{r}}_{=:\delta}, \underbrace{\mathbf{P}_{\text{T}}\boldsymbol{r} - \rho_{\text{T}}\mathbf{P}_{\text{T}}\boldsymbol{u}}_{=:\mathbf{y}}) - \mathbf{P}_{\text{T}}\boldsymbol{r}.$$

Differentiating $\boldsymbol{f}^{\text{AC}}_{\text{T}}$ using the chain rule, we obtain

$$\begin{cases} \frac{\partial \mathbf{f}_{\text{T}}}{\partial \boldsymbol{u}} = \frac{\partial \mathbf{g}}{\partial \delta}\frac{\partial \delta}{\partial \boldsymbol{u}} + \frac{\partial \mathbf{g}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \boldsymbol{u}} \\ \frac{\partial \mathbf{f}_{\text{T}}}{\partial \boldsymbol{r}} = \frac{\partial \mathbf{g}}{\partial \delta}\frac{\partial \delta}{\partial \boldsymbol{r}} + \frac{\partial \mathbf{g}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial r} - \mathbf{P}_{\text{T}} \end{cases}$$

that is to say, using the formulas of appendix B:

—If $\|\mathbf{y}\| \le \delta$, then

$$\begin{cases} \boldsymbol{f}^{\text{AC}}_{\text{T}}(\boldsymbol{u}, \boldsymbol{r}) &= -\rho_{\text{T}}\mathbf{P}_{\text{T}}\boldsymbol{u}, \\ \frac{\partial \boldsymbol{f}^{\text{AC}}_{\text{T}}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{r}) &= -\rho_{\text{T}}\mathbf{P}_{\text{T}}, \\ \frac{\partial \boldsymbol{f}^{\text{AC}}_{\text{T}}}{\partial \boldsymbol{r}}(\boldsymbol{u}, \boldsymbol{r}) &= \mathbf{0}_{2\times 3}. \end{cases}$$

—If $\|\mathbf{y}\| > \delta > 0$, then

$$\begin{cases} \boldsymbol{f}^{\text{AC}}_{\text{T}}(\boldsymbol{u}, \boldsymbol{r}) &= \lambda\frac{\mathbf{y}}{\|\mathbf{y}\|} - \mathbf{P}_{\text{T}}\boldsymbol{r}, \\ \frac{\partial \boldsymbol{f}^{\text{AC}}_{\text{T}}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{r}) &= -\rho_{\text{T}}\frac{\delta}{\|\mathbf{y}\|}(I_2 - \frac{\mathbf{y}\mathbf{y}^\top}{\|\mathbf{y}\|^2})\mathbf{P}_{\text{T}}, \\ \frac{\partial \boldsymbol{f}^{\text{AC}}_{\text{T}}}{\partial \boldsymbol{r}}(\boldsymbol{u}, \boldsymbol{r}) &= \mu\frac{\mathbf{y}}{\|\mathbf{y}\|}\mathbf{P}_{\text{N}} + \frac{\delta}{\|\mathbf{y}\|}(I_2 - \frac{\mathbf{y}\mathbf{y}^\top}{\|\mathbf{y}\|^2})\mathbf{P}_{\text{T}} - \mathbf{P}_{\text{T}}. \end{cases}$$

—If $\|\mathbf{y}\| \ge 0 \ge \delta$, then

$$\begin{cases} \boldsymbol{f}^{\text{AC}}_{\text{T}}(\boldsymbol{u}, \boldsymbol{r}) &= -\mathbf{P}_{\text{T}}\boldsymbol{r}, \\ \frac{\partial \boldsymbol{f}^{\text{AC}}_{\text{T}}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{r}) &= \mathbf{0}_{2\times 3}, \\ \frac{\partial \boldsymbol{f}^{\text{AC}}_{\text{T}}}{\partial \boldsymbol{r}}(\boldsymbol{u}, \boldsymbol{r}) &= -\mathbf{P}_{\text{T}}. \end{cases}$$

In limit cases, such as $\|\mathbf{y}\| = \delta$, the computation of the Jacobian matrix does not make sense. The notion of generalized gradients has to be introduced which numerically amounts to arbitrarily choosing one of the two possible formulas. For the normal part, for instance, one of the condition $r_{\text{N}} - \rho_{\text{N}} u_{\text{N}} > 0$ (or $r_{\text{N}} - \rho_{\text{N}} u_{\text{N}} > 0$) includes the equality case $r_{\text{N}} - \rho_{\text{N}} u_{\text{N}} = 0$. For more details on the computation of the generalized gradient and the theoretical aspects of generalized Newton's method, we refer to [Qi and Sun 1993]. The formulation of the function $\boldsymbol{f}^{\text{AC}}$ slightly differs from the original one presented in [Alart and Curnier 1991]. In particular, the tangential part $\boldsymbol{f}^{\text{AC}}_{\text{T}}$ uses the projection on the ball $\boldsymbol{B}(\mathbf{0}, \mu r_{\text{N}})$ rather than the projection on the modified ball $\boldsymbol{B}(\mathbf{0}, \mathrm{P}_{\mathbb{R}^+}(r_{\text{N}} - \rho_{\text{N}} u_{\text{N}}))$. In [Christensen et al. 1998], the same modification is done to simplify the computation of the function and its gradients; according to our experience, this variant does not change the behavior of the Newton method.

### 4.4 Newton step computation, line search and convergence

Once the computation of the Alart-Curnier function $\boldsymbol{f}^{\text{AC}}$ and its Jacobian matrix is implemented, the algorithm is simple: the unknowns are initialized (in the case of non-vanishing contacts, we use the values of the previous time step), and the Newton step is computed by solving the linearization of system (7). A simple Goldstein-Price line search [Bonnans et al. 2003] is then performed in the Newton direction to ensure that the least-square criterion $\|\boldsymbol{f}^{\text{AC}}\|^2$ decreases. We loop until $\|\boldsymbol{f}^{\text{AC}}\|^2$ goes below a given tolerance or, alternatively, until the maximum number of iterations is reached. The nonsmooth Newton method and the Goldstein-Price line search are respectively described with pseudo-code in algorithms 1 and 2. An important feature of our algorithm, compared to some previous works limited to the search for stationary points of optimization problems (not necessarily solutions of the frictional contact problem) is that our convergence error $\|\boldsymbol{f}^{\text{AC}}\|^2$ is a measure of the distance to an *actual* solution of our initial problem (7). Indeed, recall that cancelling $\boldsymbol{f}^{\text{AC}}$ is strictly equivalent to solving problem (7).

**Algorithm 1** Nonsmooth Newton Method.

**Require:** Initial guess $\mathsf{r}_0$
**Require:** tol tolerance, itermax maximum number of iteration
**Ensure:** Solution of $\mathsf{f}^{\text{AC}} = 0$
  /* *Initialization* */
  $\mathsf{k} \leftarrow 0$
  $\mathsf{r}_\mathsf{k} \leftarrow \mathsf{r}_0$
  /* *Start Newton loop* */
  **while** $\frac{1}{2}\|\mathsf{f}^{\text{AC}}(\mathsf{r}_\mathsf{k})\|^2 > \mathsf{tol}$ and $\mathsf{k} \le \mathsf{itermax}$ **do**
    /* *Compute* $\mathsf{f}^{AC}$ *and choose a sub–gradient* */
    $\mathsf{u}_\mathsf{k} \leftarrow \mathsf{W}\mathsf{r}_\mathsf{k} + \mathsf{b}$
    $\mathsf{b}_\mathsf{k} \leftarrow \mathsf{f}^{\text{AC}}(\mathsf{u}_\mathsf{k}, \mathsf{r}_\mathsf{k})$
    $\mathsf{A}_\mathsf{k} \leftarrow \frac{\partial \mathsf{f}^{\text{AC}}}{\partial \mathsf{u}}(\mathsf{u}_\mathsf{k}, \mathsf{r}_\mathsf{k})\mathsf{W} + \frac{\partial \mathsf{f}^{\text{AC}}}{\partial r}(\mathsf{u}_\mathsf{k}, \mathsf{r}_\mathsf{k})$
    /* *Compute Newton's direction* */
    Solve $\mathsf{A}_\mathsf{k}\mathsf{d}_\mathsf{k} = \mathsf{b}_\mathsf{k}$
    /* *Compute length with a line search procedure. See Algorithm 2.* */
    Compute $\alpha_\mathsf{k}$.
    /* *Update* $\mathsf{r}_\mathsf{k}$ */
    $\mathsf{r}_\mathsf{k} \leftarrow \mathsf{r}_\mathsf{k} + \alpha_\mathsf{k}\mathsf{d}_\mathsf{k}$
  **end while**

4.4.1 *Convergence, in theory.* Since $\|\boldsymbol{f}^{\text{AC}}\|^2$ decreases along the iterations, it eventually converges (unless the maximum number of iterations is reached), but not necessarily to a solution: $\|\boldsymbol{f}^{\text{AC}}\|^2$ may converge to a value above the given tolerance. Some further hints on the theoretical expectations of convergence are given in Cadoux's

**Algorithm 2** Goldstein–Price line search procedure

**Require:** $\alpha_{init}$ initial value of line length
**Require:** $[\alpha_\mathsf{l}, \alpha_\mathsf{u}]$ initial interval for $\alpha$
**Require:** $\alpha_{\max}$ maximum value of $\alpha$
**Require:** $\mathsf{m}_1 = 0.1, \mathsf{m}_2 = 0.9$ user defined parameters
**Ensure:** Optimal value of $\alpha$

```
α ← α_min
/* Compute the initial value of the merit function */
q_0 ← (1/2) ||f^AC(u_k, r_k)||2
/* Compute the initial value of the slope */
q'_0 ← (f^AC(u_k, r_k))^T ∂f^AC/∂r (u_k, r_k) d_k
success ← FALSE
while success = FALSE do
    r_trial ← r_k + α d_k
    u_trial ← W r_trial + b
    q_k ← f^AC(u_trial, r_trial)
    slope ← (q_t − q_0)/α
    if (slope ≥ m_2 q'_0 and slope ≤ m_1 q'_0) then
        success ← TRUE
    else
        if slope < m_2 q'_0 then
            α_l ← α
        else
            α_u ← α
        end if
        if α_u < α_max then
            α ← (1/2)(α_l + α_u)
        else
            α ← 10α
        end if
    end if
end while
```

PhD thesis [2009]. One the one hand, the dimension $nd \times m$ of matrix $\mathbf{J}$ (were $n$ is the total number of contacts, $m$ is the total number of degrees of freedom of the system, and $d$ is the dimension of space, typically $d = 3$) is deeply linked to the existence of a solution to the one-step problem (7): when $nd > m$, $\mathbf{J}$ is not surjective anymore, which does not guarantee the existence of a solution. On the other hand, the non-surjectivity of $\mathbf{J}$ implies $\mathbf{W}$'s singularity, and consequently, is likely to make the problem harder to solve. For more details, we refer the reader to [Cadoux 2009; Acary et al. 2010].

4.4.2 *Convergence, in practice.* In our experiments presented in Section 5, we shall see that the fraction

$$\nu = \frac{nd}{m}$$

plays an important role with respect to the quality and rate of convergence. In practice, the range of simulations that we are interested in mostly fall into the favorable case, namely the case when the criterion $\nu < 1$ is roughly satisfied. Provided the time step is chosen small enough (to overcome the difficulty when $\boldsymbol{f}^{\mathrm{AC}}$ is non-smooth close to the current iterate), convergence with reasonable speed will then be observed. Actually, $\nu$ can be viewed as a conditioning number for our simulations.

### 4.5 Collision detection and constraints partitioning

This paper focuses on the response due to impact and resting contact, and not on collision detection. However, in order to optimize the solving of the system resulting from the active constraints, we provide a constraints partitioning technique, described below.

The geometry of a rod is approximated by a set of bounding cylinders and for each pair of objects subject to collide we track the pairs of closest points that lie on the axes of the corresponding boundary cylinders over time. One contact is declared active when two distinct boundary cylinders are found to intersect at a given timestep, *i.e.*, once the distance between the two closest points is below the diameter of the rod. In practice we used around 1 cm-long cylinders, *i.e.*, 30 cylinders for a typical strand of 30 cm simulated in most of our examples. This simple algorithm proved sufficiently robust for our tests, even when using large time steps. For very high speed motion however, it may be preferable to use continuous detection algorithms [Redon et al. 2002] in order to capture every single occurring collision.

To save computing time, we partition the set of active constraints and the set of strands into $N_{\mathscr{P}}$ groups of *coupled* constraints and strands at each time step. This allows us to solve $N_{\mathscr{P}}$ independent systems of the form (7), instead of one single large system accounting for all strands of the scene. To determine the partition $\mathscr{P}$, let us represent the sets of constraints and strands as a graph, where nodes stand for strands and edges for constraints. An example is presented in Figure 5. Two different strands (resp. constraints) are said to be coupled if there exists a constraints path in the graph (resp. a strands path in the dual graph) connecting their corresponding nodes (resp. their corresponding nodes in the dual graph). Partitioning our initial set of constraints thus amounts to partitioning a graph into connected sets, which is achieved efficiently.

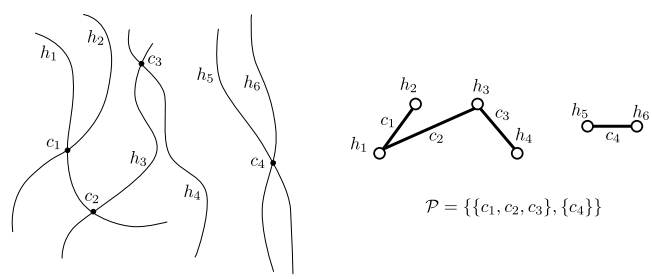


Fig. 5. Example of constraints partition using graph connectivity. Left: contacting strands; right: corresponding graph and partition ($N_{\mathscr{P}} = 2$).

Finally, we have accelerated the search for contacting pairs by employing a spatial hashing of the 3d space with a uniform grid, similarly to [Teschner et al. 2003]. In our simulation, collision detection took less than 10% of the total computational time, and thus never appeared to be the computational bottleneck in our simulations. See Section 5.3 for the detailed performance.

## 5. RESULTS

In this section, we accurately analyze our method in terms of 1) *realism* (*i.e.*, its ability to capture relevant emerging phenomena), 2) numerical *stability* (*i.e.*, its convergence properties), and 3) computational *efficiency* (*i.e.*, its time performance). Limitations and discussion are provided in the next section.

## 5.1 Realism: capture of relevant physical properties

5.1.1 *Contact robustness.* Figures 2 (left), 6, and the accompanying video demonstrate the ability of our method to address the challenging scenarios of plectoneme formation and tight knot tying. Because of the slender geometry of a thin rod, simulating self-contact and especially the formation of plectonemes or tight knots requires an extremely robust algorithm for contact. Unlike basic penalty-based approaches which fail to prevent interpenetration beyond a certain force/torque magnitude and thus may cause the self-crossing of the rod, our constraint-based algorithm robustly maintains contact whatever the amplitude of the force/torque applied onto the free end of the rod. With our method, the violation of constraints is prevented whatever the magnitude of the applied forces.

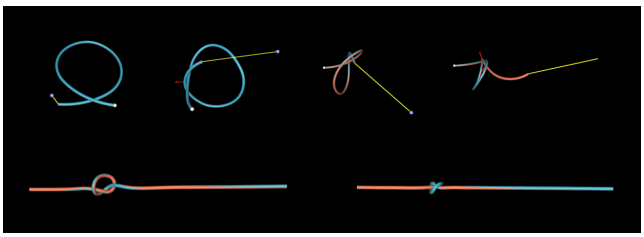

Fig. 6. Interactive making of a knot from a Super-helix of 20 elements ($\mu = 0$). With our method the user can tighten the knot as strongly as he wants, without causing contact to break.

5.1.2 *Capture of dynamic and static friction.* The accompanying video shows the effect of increasing the coefficient of friction $\mu$ on the simple example of a strand contacting a rotating sphere. When $\mu$ is large enough, dry friction is captured as expected. In the knot example, the user can easily perceive the influence of $\mu$ when virtually tying a knot: in the absence of friction this task is actually challenging since contact points are always sliding and thus cannot guide the user. With a nonzero friction coefficient however, static friction occurs and greatly helps the user as it maintains some fixed points of contact that can be used as pivots.

The typical stick-slip instabilities occurring inside fiber assemblies are illustrated in the following examples.

5.1.3 *Packing of fibers.* We have simulated the fall of a dense packing of 31 parallel spaghetti[5] (modeled as mass-spring chains with both ends free) on to a plate. Figure 7 and the video show the effects of varying the friction coefficient. In the case where there is no friction, the rods spread over the plate. When friction is activated between the rods and the plate, but not between the rods, the spaghetti form a stack but continue to slide against each other. When friction is active everywhere, the stack is higher and remains perfectly still.

In a second experiment depicted in the video, we have simulated a bunch of 7 spaghetti as CORDE rods with a straight shape at rest, but initialized with a nonzero curvature. After impact, the rods tend to recover their natural straight shape when there is no friction, whereas they remain curved in the frictional case as tangential friction forces compensate for elastic forces.

[5] As a matter of interest, modeling and animating spaghetti was amusingly listed by Blinn [1998] as one of the top 10 unsolved problems in computer graphics.

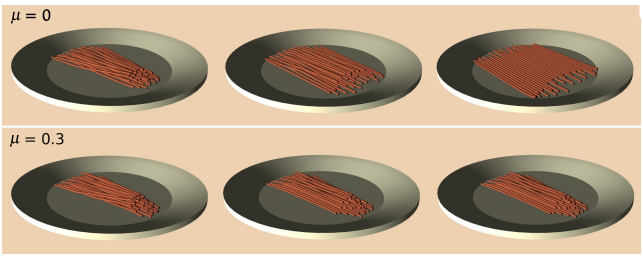


Fig. 7. Dense packing of 31 spaghetti in a plate. Top: without friction ($\mu = 0$), the rods just spread out over the plate. Bottom: in the presence of friction ($\mu = 0.3$), the rods stack in a stable way.

5.1.4 *Complex interactions in hair dynamics.* We have tested our method to resolve interactions in hair dynamics in the presence of friction. Figures 2 (right), 8, and the accompanying video show a wavy head of hair under motion where thin wisps are modeled as 84 contacting Super-helices, in the presence of friction ($\mu = 0.3$) or not ($\mu = 0$). Compared to previous approaches, our method is the first to properly capture subtle hair contacts occurring between hair curls, thus correctly preserving the *hair volume* as well as capturing *sticking* and *coherence of motion* in the presence of friction. In Figure 8, observe how hair wisps get entangled, forming complex and discontinuous patterns. On the right, note how friction plays a major role in the realism of motion: without friction, hair wisps just slide against each other and against the body, whereas in the presence of friction, typical stick-slip instabilities (especially visible on the shoulder of the character and on top of the head) occur.

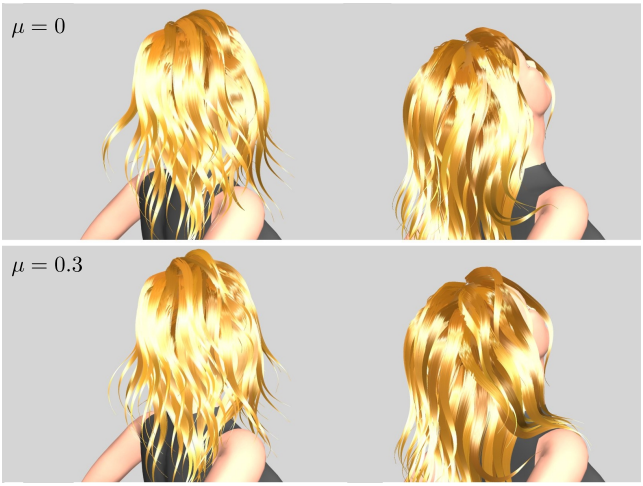


Fig. 8. Simulating frictional contact in a full head of hair (84 simulated strands). Top: without friction ($\mu = 0$). Bottom with friction ($\mu = 0.3$). On every picture, observe the retrieval of a realistic hair volume as well as the presence of discontinuous patterns caused by the proper solving of self-contacts. Left: during motion, hair subject to friction (bottom) exhibits less volume than the frictionless case (top) due to an enhanced coherence of motion. Right: at rest, when friction is applied (bottom), some hair wisps remain stuck over the shoulder thanks to the correct modeling of dry friction, whereas unnatural sliding occurs in the frictionless case (top). Observe the hair wisps that rest on top of another hair layer, similarly as in Figure 1. See also the accompanying video.

5.1.5 *Contact in rigid bodies.* In order to explore the versatility of our method we have attempted to solve frictional contact within other dynamic systems, such as rigid bodies. Figure 9 illustrates the loop benchmark (introduced by Kaufman *et al.* [2005]), extended to a double loop, where a ball rolls over two rails ($n = 2$) with a curly shape. The ball initially has some spin, but no translational velocity. Increasing the friction coefficient reduces sliding phases in favor of pure rolling (adherence), which causes the ball to go further through the looping rails (see the accompanying video).

Fig. 9. Real-time simulation of a ball in a double loop. The ball initially has some spin, and different friction coefficients between the ball and the rails are tested. When friction is high, the spin energy is converted into translational kinetic energy, which makes the ball roll through the loops.

In this simple example, convergence is very fast (less than 5 iterations) even when using a large time step (11 ms). However, as we shall see in Section 6.2, our method may not be adapted to handle a set of rigid bodies with stacking as the conditioning of the method will rapidly decrease due to the large number of contacts compared to the low number of degrees of freedom of the system (see also next section).

## 5.2 Stability: a criterion for fast convergence

We have carefully analyzed the convergence properties of our algorithm on a spaghetti simulation consisting of 76 flexible rods (modeled as mass-spring chains composed of 22 nodes with both ends free) subject to external contacts (with a plate) as well as self-contacts, in the presence of friction ($\mu = 0.3$). After a series of impacts, the assembly of rods rapidly stabilizes as expected, forming a stack subject to more than 1800 contacts (see Figure 2, right, and the accompanying video).

To evaluate the quality of convergence of the solver, we have plotted the time required for the solver to converge as a function of the number $\nu = \frac{nd}{m}$ introduced in Section 4.4. In this particular case, $d = 3$ and $m = 76 \times 22 \times 3 = 5016$, thus $\nu = \frac{n}{1672}$, where $n$ is the total number of contacts. Figure 10, top, gives the results of this plot. It clearly shows the influence of $\nu$ on the rate of convergence and identifies that the value $\nu = 1$ critically defines a threshold for proper convergence: when the criterion $\nu < 1$ is satisfied, convergence remains reasonably fast, whereas it grows exponentially as soon as $\nu > 1$. In practice, we have noted that the solver almost always manages to converge until $\nu$ reaches a value close to 1.5; however, the time taken to converge grows larger as $\nu$ increases. Figure 10, bottom, displays the relative density of convergence error as a function of $\nu$. It is noticeable that the solver converges in most cases to the desired precision (here, fixed to $10^{-5}$), even when $\nu$ is slightly superior to 1. Convergence occurred in roughly 100 iterations on average (the maximum number of iterations was set to 200). In 10% of the cases we studied, the solver did not strictly reach the required precision. Note however that the solver *always* converged to an acceptable solution, close to the target value 0 (the maximum recorded error was only of $10^{-2}$). In the $\nu < 1$ zone, we noticed that the solver happened to converge to such approximate solutions in situations involving large impacts (typically, when the spaghetti collide with the plate at maximum speed). We strongly suspect this issue may be due to the lack of a proper warm start in such cases, as contact forces and post-impact velocities are initialized with zero values when new contacts are set active. The solver thus starts from a configuration that is far from the solution when impacts are large, causing the Newton algorithm to fall into some local minima. In the near future, we plan to investigate how to properly estimate an initial guess of the solution in the specific case of impacts, based on the knowledge of relative velocities before impact. We note however that in all cases, such an approximate convergence did not produce any visually disturbing effects nor did it cause the solver to fail in the following time steps.

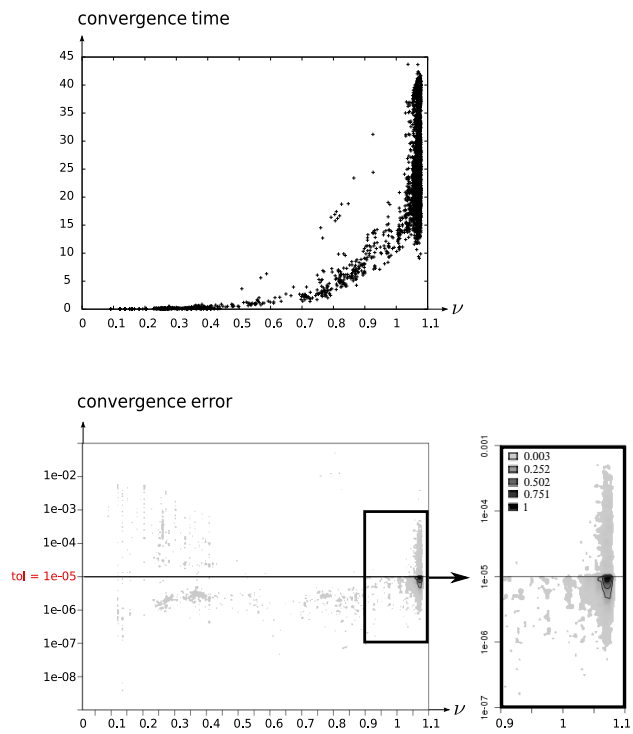


Fig. 10. Analysis of convergence for the 76 spaghetti demo. Top: convergence time of the solver (in seconds), as a function of $\nu$. Bottom: relative density of the convergence error of the solver (measured as $\|\boldsymbol{f}^{\mathrm{AC}}\|^2$), as a function of $\nu$. The tolerance error is set to $10^{-5}$ in our internal, dimensionless units. The dense zone inside the rectangle (in which most error values concentrate) is zoomed on the right to better depict the density distribution of error values (represented as greyscale values). Note that most error values lie beneath the tolerance.

We have plotted similar data for other rods experiments, and exactly observed the same properties for the convergence of the solver, as

depicted in Figure 11: the solver quickly converges to a solution when $\nu < 1$, whereas convergence still occurs for $\nu$ slightly superior to 1 (but is slowed down), and finally the solver fails to converge when the criterion is far from being satisfied (typically, when $\nu > 1.5$). This result confirms the observation previously made by Cadoux [2009] in the context of simpler experiments with a lower number of contacts, and enhances the fact that $\nu$ plays the role of a generic condition number.

In addition, by comparing the results obtained on different simulations, we have observed that for a constant $\nu$, the time of convergence grows as the number $n$ of contacts increases. Actually, the convergence time profile as a function of $\nu$ observed for two different simulations appears to be the same up to a scaling factor related to the mean number of contacts (see Figure 11).

Finally, although our problem is nonsmooth, we observe typical Newton quadratic convergence in some favorable cases where a high precision ($\|\boldsymbol{f}^{\mathrm{AC}}\|^2 < 10^{-18}$) is reached by the solver. This is for example the case for the simple simulation presented in Figure 9, where the solver reaches a precision of $10^{-35}$ in less than 5 iterations, at each time step! This observation is actually in line with the theoretical study made by Qi and Sun [1993] in the nonsmooth case (though they make more restrictive assumptions than we do).

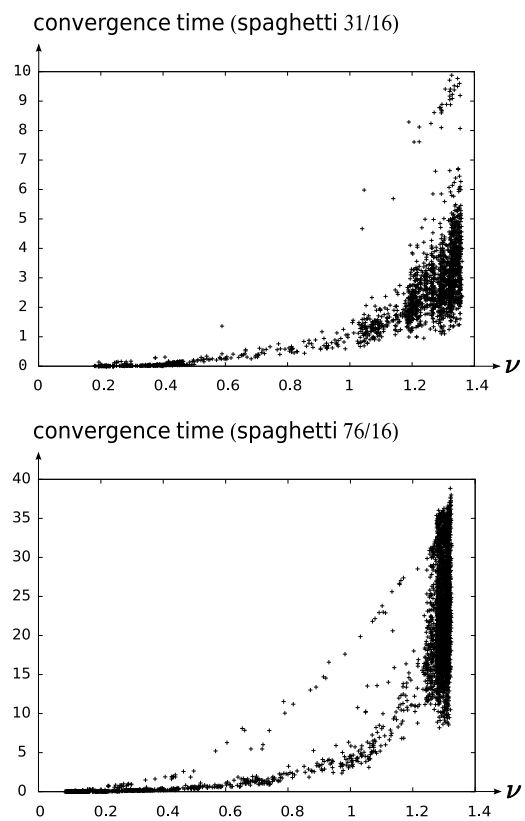


Fig. 11. Comparison of the solver convergence time profile (in seconds) between 2 different simulations: 31 mass-springs of 16 nodes (top) *vs.* 76 mass-springs of 16 nodes (bottom). Note that the profiles are similar up to a scaling factor directly related to the mean number of contacts.

### 5.3 Performance

Although time performance was not our primary goal, it turns out that our method is fast enough to be used for handling hundreds of frictional contacts within a few minutes per frame. This is mainly due to our sparse implementation of the algorithm, which is all the more valuable as the number of contacting strands is high. However, computational time quickly becomes the main bottleneck of our simulations, which currently limits the use of our technique to systems subject to a few thousands contacts.

All the simulations presented in the paper as well as in the accompanying video were processed on a single threaded application running on an Intel Core 2 Duo CPU at 2.1 GHz. Table I gives a summary of our computational results. The indicated mean computational time is based on computations during one elementary time step, including the dynamics of thin rods, collision detection and collision response. For large systems ($n > 100$) the computational time taken by our solver represents 80% to 90% of the total computational time, depending on the rod model used.

For every simulation, we used a tolerance on $\|\boldsymbol{f}^{\mathrm{AC}}\|^2$ of $10^{-5}$ in dimensionless units (*i.e*, physical quantities were divided by typical values of the rods system, in order to bring matrices coefficients close to 1), which in practice corresponds to a good trade-off between visual accuracy and computational cost[6]. We chose the time step between 0.1 and 11 ms, depending on the rod model used. Typically, the CORDE model, based on an explicit integration scheme, required a smaller time step to remain stable, compared to other rods models.

Table I. Performance results for our simulations on various models composed of $N_s$ rods and $N_e$ elements (or nodes) per rod.

| Rod model | Example ($N_s$, $N_e$) | $\mu$ | dt (ms) | $n$ max | $\nu$ max | mean time (s) |
|---|---|---|---|---|---|---|
| SH | Plectoneme (1, 20) | 0 | 11 | 18 | 0.9 | 0.04 |
| SH | Tight knot (1, 20) | 0 | 11 | 8 | 0.4 | 0.033 |
| SH | Wavy hair (84, 8) | 0 | 11 | 392 | 0.58 | 2.6 |
| SH | Wavy hair (84, 8) | 0.3 | 11 | 380 | 0.57 | 41.2 |
| C | Self-stacking (1, 256) | 0.3 | 0.25 | 303 | 1.18 | 0.31 |
| C | Spaghetti (7, 16) | 0.3 | 0.1 | 121 | 1.08 | 0.36 |
| MS | Spaghetti (31, 16) | 0.3 | 1 | 653 | 1.32 | 2.26 |
| MS | Spaghetti (76, 22) | 0.3 | 1 | 1806 | 1.08 | 19.1 |

SH: Super-helix; C: CORDE; MS: Mass-spring.

## 6. DISCUSSION

In this section, we provide comparisons with previous methods and discuss the strengths and current limitations of our approach.

### 6.1 Comparison with previous methods

6.1.1 *Smooth friction models.* Our results clearly demonstrate the advantage of using a nonsmooth solver. Dry friction is indeed a predominant phenomenon in contacting rods systems. Classical methods based on viscous friction would typically not be able to

[6]Alternatively, a relative non-square error measure such as $\varepsilon = \frac{\|\boldsymbol{f}^{\mathrm{AC}}\|}{1+\|\mathbf{r}\|}$ could be used. In our experiments, we have measured that our $10^{-5}$ tolerance is equivalent to a maximal relative error of $\varepsilon = 10^{-3}$.

recover the stick-slip effects depicted by our simulations. Hair is a crucial example where previous models (*e.g*, [Plante et al. 2001; Bertails et al. 2006; Hadap 2006; Selle et al. 2008; McAdams et al. 2009]) fail in capturing this typical, nonsmooth behavior.

6.1.2 *Ad-hoc, visually acceptable friction models.* Some approaches have attempted to modify Coulomb's friction model so as to stick to an easily solvable contact formulation [Milenkovic and Schmidl 2001; Kaufman et al. 2005]. In [Milenkovic and Schmidl 2001] Coulomb's friction model is adapted to fit into the quadratic program (QP) framework for resolving non-penetration. Frictional impacts are treated by restricting contact impulses to the linearized Coulomb friction cone, while resting contact is modelled by enforcing the relative acceleration to lie in the dual linearized friction cone and by subsequently projecting boundary-located accelerations onto the tangent space. While the former formulation is a reasonable approximation of Coulomb's model at impacts (see 6.1.3), the latter is more questionable: objects have to reach a certain normal acceleration before they are allowed to take off, which appears to be unphysical. Moreover, the authors do not fully resolve the resulting QP but rather stop after a few iterations only, with no bounded tolerance. Kaufman *et al.* [2005] derive an approximation of Coulomb friction based on the maximum dissipation principle, leading to the formulation of two separable QPs. Similarly as in [Milenkovic and Schmidl 2001], the problem is only partially solved for the sake of efficiency, using a limited number of iterations regardless of any convergence error.

Such simplified methods are appealing due to the fact they may scale favorably *w.r.t.* the number of contacts. In [Kaufman et al. 2005] computational time grows linearly with the number of contacts, which allows them to simulate a large number of rigid bodies subject to tens of thousands of contacts, in a reasonable amount of time. In contrast, due to its higher complexity, our approach is currently limited to the handling of a few thousand contacts. However, simplifications made on Coulomb friction and on the accuracy of resolution have a price to be paid, in terms of physical realism and stability: with our solver, as illustrated on the spaghetti and hair demos, all contacting systems remain *perfectly still* when they have come to rest, as one would expect. This is hardly achieved by the simulations presented in [Kaufman et al. 2005].

6.1.3 *Faceted Coulomb's friction cone.* The current trend in graphics to model Coulomb friction consists of solving an LCP generated by linearizing the Coulomb friction cone [Erleben 2007; Kaufman et al. 2008; Otaduy et al. 2009]. The main advantage of this approach is a theoretical result due to Stewart and Trinkle [1996], which both provides a criterion for the existence of at least one solution, and proves its computability *via* a pivoting technique such as the Lemke algorithm. While using an alternative technique – based on staggered projections – Kaufman *et al.* [2008] also report good convergence properties, even when using a very low convergence tolerance (down to $10^{-8}$, compared to a $10^{-1}$ tolerance said to be "sufficient to generate convincing frictional behavior"). Note that such quantitative measures of convergence are very sparse in the graphics literature when dealing with the frictional contact problem.

The faceting process suffers from a number of issues. Firstly, the faceting introduces an artificial anisotropy in the sliding plane yielding some artifacts in the mechanical behavior of the whole system. This induced anisotropy is generally difficult to avoid and to forecast with respect to the number of facets. The main reason is the non-monotonicity of the convergence of the error with the number of facets compared to the original second-order friction cone. As an illustration, a simple example of this issue where a sphere on a table is controlled by forces to draw a triangle is presented in [Acary and Brogliato 2008, Sec. 13.3.7]. Such a lack of precision may especially be disturbing for critical applications such as haptics or virtual prototyping. The second drawback of faceting the friction cone is the poor numerical efficiency beyond 1000 contacts. In most favorable cases, a pivoting method for an LCP such as Lemke's method is empirically of complexity $O(n^3)$ where $n$ is the number of unknowns. When the number of facets increases at each contact point, the number of unknowns grows linearly leading quickly to intractable problems. Last but not least, one major drawback of the faceting approach is the difficulty of its implementation, compared to a straightforward Newton method.

6.1.4 *Global and local approaches.* Our contact algorithm attempts to compute the contact forces by changing all forces at all contact points simultaneously. It can solve, in particular, problems involving only one contact, and this property can be used to devise another algorithm for the same problem, which solves a local problem contact-per-contact and loops over all contacts. This method is often called "Gauss-Seidel" due to its resemblance with the Gauss-Seidel algorithm to solve linear systems. The main interests of this algorithm are its good scalability *w.r.t.* the number of contacts, and the fact that it may be able, in some cases, to solve massive stacking problems of systems with a low number of degrees of freedom. Actually, this method requires very little implementation once the global method is available. In our rods experiments however, this method did not necessarily prove faster nor more robust than the global method: the method suffered from slow convergence together with cycling issues and sometimes failed to converge even in the favorable case where $\nu < 1$; moreover, its stopping criterion is less clear since there is no parameter (such as our $\|\boldsymbol{f}^{\mathrm{AC}}\|^2$) which monotonically controls convergence. As a consequence, we advocate the use of the global method when the criterion $\nu < 1$ is roughly satisfied (which was typically the case in our experiments), and of the local approach otherwise (typically, in the case of densely packed fibers or nonconvex rigid bodies stacking).

## 6.2 Limitations and future work

As mentioned above, our global approach is not adapted to overconstrained systems, *i.e.*, systems where the number of degrees of freedom is low compared to the number of contacts. This is typically the case in assemblies of rigid bodies. Note that in Figure 9 we have $d = 3$, $m = 6$ and $n = 2$, which makes this example fall into the favorable case where $\nu = 1$. However, the criterion is generally not verified in stacking systems of rigid bodies. Take for example a pile of cubes in 3d: if the number of contacts per cube is $n = 4$, then $\nu = 2$ and the criterion is not satisfied. Even after filtering the number of active contacts at the collision detection stage (leading to at least 3 contacts per face), $\nu$ would still be out of range. In practice, we indeed observed slow convergence and frequent occurrence of large convergence errors when trying to simulate this kind of problem. The worst case is actually obtained when simulating rigid bodies ($m = 6$) with a high number of self-contacts: typically, concave and convex rigid bodies that get entangled.

Our convergence criterion $\nu < 1$ is defined independently from the parameterization of the rod model used in the simulations. However, this criterion is obviously more favorable to systems whose

kinematics is loosely defined, *e.g.*, computed through energy functions (penalization terms) rather than hard constraints (reduced-coordinates or explicit hard constraints), since loose kinematics do not exactly remove degrees of freedom. However, in the case of deformable bodies, one acceptable solution may consist in increasing the resolution of the model at locations where the number of contacts is high. Note that if this is required in order to get good properties of convergence, this is also often desirable for increasing the quality of simulation, as high deformations are likely to occur near the contacting zones.

In the future, we would like to better address the difficult case where a high number of contacts is applied onto deformable objects without causing high deformations (such as a straight rod, with a low number of degree of freedom, lying on a flat surface): in this case, it would of course be preferable to model the rod using a low number of degrees of freedom (for instance, reduced-coordinates), which for the moment would be hardly solvable by our approach in the case when a high number of contacts is applied. Our goal would be to find a global approach that scales better in the number of contacts, while possessing better convergence properties than classical Gauss-Seidel like approaches. We note that in the specific case of dense structured stacking of rigid bodies with a faceted Coulomb cone, Erleben [2007] proposed a heuristic based on velocity-shock propagation to accelerate the rate of convergence of the Gauss-Seidel algorithm. In the same spirit, exploiting the peculiar structure of a dense packing of fibers may help us design a dedicated, optimized iterative scheme that would properly scale up for the efficient handling of thousands of fibers in contact.

## 7. CONCLUSION

We have introduced a new method for computing self-contacts in an assembly of thin elastic rods, in the presence of friction. Our method is simple to implement and can be applied to any rod model – maximal or reduced-coordinates. The only required information to provide, besides the dynamics, is the gradient $\mathbf{J}$. Furthermore, we provide the user with a simple criterion able to predict the quality of convergence for a given simulation. This criterion is often favorable in the case of deformable bodies. By testing our algorithm on various rods models, we showed our ability to capture subtle, nonsmooth effects due to contact and friction.

In the future, we would like to further investigate the scalability of our method to simulate very large assemblies of strands. Our main motivation is to design a reference model for hair material in order to better understand the process of wisps formation during motion, *e.g.*, to analyze the influence of hair geometry and hair friction on this process - such studies being very challenging to conduct experimentally [Bertails et al. 2006]. Such an accurate simulator would serve as a first step before designing a more efficient, macroscopic hair model that could further be validated against this reference model.

We would also like to extend our method to the robust handling of self-contact in cloth, which is crucial for capturing important features such as folds and wrinkles.

### REFERENCES


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

ACARY, V., CADOUX, F., LEMARECHAL, C., AND MALICK, J. 2010. A formulation of the linear discrete Coulomb friction problem via convex optimization. *ZAMM / Z angew Math Mech; Zeitschrift für Angewandte Mathematik und Mechanik*.

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.

BARAFF, D. 1989. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'89 conference)*. ACM, New York, NY, USA, 223–232.

BARAFF, D. 1993. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica 10*, 292–352.

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.

BARAFF, D. AND WITKIN, A. 1992. Dynamic simulation of non-penetrating flexible bodies. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'92 conference)*. 303–308.

BARAFF, D. AND WITKIN, A. 1998. Large steps in cloth simulation. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'98 conference)*. 43–54.

BARBIČ, J. AND JAMES, D. 2007. Time-critical distributed contact for 6-dof haptic rendering of adaptively sampled reduced deformable models. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'07)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 171–180.

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.

BERTAILS, F. 2009. Linear time super-helices. *Computer Graphics Forum (Proceedings of Eurographics'09) 28,* 2 (apr).

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. In *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'06 conference)*. ACM, 1180–1187.

BLINN, J. 1998. Ten more unsolved problems in computer graphics. *IEEE Computer Graphics and Applications (Keynote of the SIGGRAPH'98 Conference) 18,* 5, 86–89.

BONNANS, J. F., GILBERT, J. C., LEMARÉCHAL, C., AND SAGASTIZÁBAL, C. A. 2003. *Numerical Optimization*. Springer-Verlag, Berlin.

BRIDSON, R., FEDKIW, R., AND ANDERSON, R. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'02 conference 21,* 3, 594–603.

CADOUX, F. 2009. Méthodes d'optimisation pour la dynamique non-régulière. Ph.D. thesis, Université Joseph Fourier.

CHOE, B., CHOI, M., AND KO, H.-S. 2005. Simulating complex hair with robust collision handling. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'05)*. 153–160.

CHRISTENSEN, P., KLARBRING, A., PANG, J., AND STROMBERG, N. 1998. Formulation and comparison of algorithms for frictional contact problems. *Int. J. Numer. Methods Eng. 42,* 1, 145–173.

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.

DURVILLE, D. 2004. Modelling of contact-friction interactions in entangled fibrous materials. In *Procs of the Sixth World Congress on Computational Mechanics (WCCM VI)*.

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

FRÉMOND, M. 2002. *Non-Smooth Thermo-mechanics*. Springer-Verlag, Berlin-Heidelberg.

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.

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

HARMON, D., VOUGA, E., SMITH, B., TAMSTORF, R., AND GRINSPUN, E. 2009. Asynchronous contact mechanics. *ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH'09 conference)*.

JEAN, M. 1999. The non smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering 177*, 235–257.

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.

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.

LENOIR, J., MESEURE, P., GRISONI, L., AND CHAILLOU, C. 2004. A suture model for surgical simulation. *2nd International Symposium on Medical Simulation (ISMS'04)*, 105–113.

LÖTSTEDT, P. 1984. Numerical simulation of time-dependent contact friction problems in rigid body mechanics. *SIAM Journal of Scientific Statistical Computing 5,* 2, 370–393.

MAPLESOFT. 2010. Maple 14. http://www.maplesoft.com/.

MARQUES, M. M. 1993. *Differential inclusions in nonsmooth mechanical problems. Shocks and dry friction.* Progress in Nonlinear Differential Equations and their Applications. 9. Basel: Birkhäuser. x, 179 p. DM 98.00; S 764.40; sfr. 88.00 .

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.

MILENKOVIC, V. AND SCHMIDL, H. 2001. Optimization-based animation. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'01 conference)*. ACM, New York, NY, USA, 37–46.

MOORE, M. AND WILHELMS, J. 1988. Collision detection and response for computer animationr3. In *Computer Graphics Proceedings (Proceedings of the ACM SIGGRAPH'88 conference)*. 289–298.

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

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

MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation 18,* 2, 109–118.

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).

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)*. Computer Science. Springer, 139–148.

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

REDON, S., KHEDDAR, A., AND COQUILLART, S. 2002. Fast continuous collision detection between rigid bodies. In *Computer Graphics Forum (Proceedings of the Eurographics Conference)*. Vol. 21. 279–288.

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.

SPILLMANN, J., BECKER, M., AND TESCHNER, M. 2007. Non-iterative computation of contact forces for deformable objects. *Journal of WSCG 15*, 33–40.

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.

SPILLMANN, J. AND TESCHNER, M. 2008. An adaptive contact model for the robust simulation of knots. *Computer Graphics Forum 27,* 2. Proceedings of Eurographics'08.

STEWART, D. AND TRINKLE, J. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *ijnme 39,* 15.

STEWART, D. E. 2000. Rigid-body dynamics with friction and impact. *SIAM Rev. 42,* 1, 3–39.

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

TESCHNER, M., KIMMERLE, S., HEIDELBERGER, B., ZACHMANN, G., RAGHUPATHI, L., FUHRMANN, A., CANI, M.-P., FAURE, F., MAGNENAT-THALMANN, N., STRASSER, W., AND VOLINO, P. 2005. Collision detection for deformable objects. *Computer Graphics Forum 24,* 1 (March), 61–81.

THEETTEN, A., GRISONI, L., ANDRIOT, C., AND BARSKY, B. 2008. Geometrically exact splines. *Journal of Computer Aided Design 40,* 1, 35–48.

## APPENDIX

## A. EQUIVALENCE BETWEEN DISJUNCTIVE AND ZERO-FINDING FORMULATIONS FOR COULOMB'S LAW

We show here how to prove the equivalence $(\boldsymbol{u},\boldsymbol{r}) \in C \iff \boldsymbol{f}^{\mathrm{AC}}(\boldsymbol{u},\boldsymbol{r}) = \boldsymbol{0}$ where $\boldsymbol{f}^{\mathrm{AC}}(\boldsymbol{u},\boldsymbol{r})$ is the Alart-Curnier function defined by equation (4). This can be easily achieved by considering each case separately. Let us first demonstrate that $(\boldsymbol{u},\boldsymbol{r}) \in C \Rightarrow [f_{\mathrm{N}}^{\mathrm{AC}}, \boldsymbol{f}_{\mathrm{T}}^{\mathrm{AC}}](\boldsymbol{u},\boldsymbol{r}) = \boldsymbol{0}$. In the taking off case, $\boldsymbol{r} = \boldsymbol{0}$ and $u_{\mathrm{N}} \geq 0$ therefore $\mathrm{P}_{\mathbb{R}^+}(r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) = \mathrm{P}_{\mathbb{R}^+}(-\rho_{\mathrm{N}} u_{\mathrm{N}}) = 0 = r_{\mathrm{N}}$ and $\mathrm{P}_{\boldsymbol{B}(\boldsymbol{0},\mu r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}} - \rho_{\mathrm{T}} \boldsymbol{u}_{\mathrm{T}}) = \mathrm{P}_{\boldsymbol{B}(\boldsymbol{0},0)}(\dots) = \boldsymbol{0} = \boldsymbol{r}_{\mathrm{T}}$. In the sticking case, $\boldsymbol{u} = \boldsymbol{0}$ and $\boldsymbol{r} \in K$ therefore $\mathrm{P}_{\mathbb{R}^+}(r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) = \mathrm{P}_{\mathbb{R}^+}(r_{\mathrm{N}}) = r_{\mathrm{N}}$ and $\mathrm{P}_{\boldsymbol{B}(\boldsymbol{0},\mu r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}} - \rho_{\mathrm{T}} \boldsymbol{u}_{\mathrm{T}}) = \mathrm{P}_{\boldsymbol{B}(\boldsymbol{0},\mu r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}}) = \boldsymbol{r}_{\mathrm{T}}$. Finally, in the sliding case, $\mathrm{P}_{\mathbb{R}^+}(r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) = \mathrm{P}_{\mathbb{R}^+}(r_{\mathrm{N}}) = r_{\mathrm{N}}$ (because $u_{\mathrm{N}} = 0$ and $r_{\mathrm{N}} \geq 0$) and $\mathrm{P}_{\boldsymbol{B}(\boldsymbol{0},\mu r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}} - \rho_{\mathrm{T}} \boldsymbol{u}_{\mathrm{T}}) = \boldsymbol{r}_{\mathrm{T}}$ (because $\boldsymbol{r}_{\mathrm{T}} \in \partial K$ and $-\rho_{\mathrm{T}} \boldsymbol{u}_{\mathrm{T}} \in N_{\boldsymbol{B}(\boldsymbol{0},\mu r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}})$.

Conversely, if $[f_{\mathrm{N}}^{\mathrm{AC}}, \boldsymbol{f}_{\mathrm{T}}^{\mathrm{AC}}](\boldsymbol{u},\boldsymbol{r}) = \mathbf{0}$, let us show that $(\boldsymbol{u},\boldsymbol{r}) \in C$ holds. As $\boldsymbol{f}_{\mathrm{T}}^{\mathrm{AC}}(\boldsymbol{u},\boldsymbol{r}) = \mathbf{0}$ one has $\boldsymbol{r}_{\mathrm{T}} \in \boldsymbol{B}(\mathbf{0}, \mu\, r_{\mathrm{N}})$ therefore $\boldsymbol{r} \in K$. As $f_{\mathrm{N}}^{\mathrm{AC}}(u,r) = \mathbf{0}$, one has $r_{\mathrm{N}} \geq 0$ and we treat the two cases $r_{\mathrm{N}} = 0$ (*) and $r_{\mathrm{N}} > 0$ (**) separately. If $r_{\mathrm{N}} = 0$ (*) then $\boldsymbol{r}_{\mathrm{T}} = \mathbf{0}$ (because $\boldsymbol{r}_{\mathrm{T}} = \mathrm{P}_{\boldsymbol{B}(\mathbf{0},\,\mu\, r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}} - \rho_{\mathrm{T}}\boldsymbol{u}_{\mathrm{T}}) = \mathrm{P}_{\boldsymbol{B}(\mathbf{0},0)}(\dots) = \mathbf{0}$) and $u_{\mathrm{N}} \geq 0$ (because $0 = r_{\mathrm{N}} = \mathrm{P}_{\mathbb{R}^+}(-\rho_{\mathrm{N}}\, u_{\mathrm{N}})$) and one is in the taking of case. If $r_{\mathrm{N}} > 0$ (**) then $r_{\mathrm{N}} = \mathrm{P}_{\mathbb{R}^+}(r_{\mathrm{N}} - \rho_{\mathrm{N}}\, u_{\mathrm{N}}) = r_{\mathrm{N}} - \rho_{\mathrm{N}}\, u_{\mathrm{N}}$ therefore $u_{\mathrm{N}} = 0$. Again two different cases appear: $\boldsymbol{r}_{\mathrm{T}} \in \mathrm{int}\, \boldsymbol{B}(\mathbf{0},\, \mu\, r_{\mathrm{N}})$ and $\boldsymbol{r}_{\mathrm{T}} \in \partial\boldsymbol{B}(\mathbf{0},\, \mu\, r_{\mathrm{N}})$. In the first case, $\boldsymbol{r}_{\mathrm{T}} = \mathrm{P}_{\boldsymbol{B}(\mathbf{0},\,\mu\, r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}} - \rho_{\mathrm{T}}\boldsymbol{u}_{\mathrm{T}}) = \boldsymbol{r}_{\mathrm{T}} - \rho_{\mathrm{T}}\boldsymbol{u}_{\mathrm{T}}$ therefore $\boldsymbol{u}_{\mathrm{T}} = \mathbf{0}$, this corresponds to the sticking case. In the second case, $\boldsymbol{r} \in \partial K$ and $\boldsymbol{u}_{\mathrm{T}} \in N_{\boldsymbol{B}(\mathbf{0},\,\mu\, r_{\mathrm{N}})}(\boldsymbol{r}_{\mathrm{T}})$ therefore $\exists \alpha > 0$ , $\boldsymbol{r}_{\mathrm{T}} = -\alpha\, \boldsymbol{u}_{\mathrm{T}}$, this is the sliding case. This ends the proof.

## B. JACOBIAN MATRIX OF FUNCTION $G$

Distinguishing between three cases, the following formulas are easily derived.

If $\|y\| < \lambda$, then $g(\lambda, y) = y$, $\frac{\partial g}{\partial \lambda}(\lambda, y) = 0_{2\times 1}$, $\frac{\partial g}{\partial y}(\lambda, y) = I_2$.

If $\|y\| > \lambda > 0$, then $g(\lambda, y) = \lambda\, \frac{y}{\|y\|}$, $\frac{\partial g}{\partial \lambda}(\lambda, y) = \frac{y}{\|y\|}$, $\frac{\partial g}{\partial y}(\lambda, y) = \frac{\lambda}{\|y\|}(I_2 - \frac{yy^\top}{\|y\|^2})$.

If $\|y\| > 0 > \lambda$, then $g(\lambda, y) = 0_{2\times 1}$, $\frac{\partial g}{\partial \lambda}(\lambda, y) = 0_{2\times 1}$, $\frac{\partial g}{\partial y}(\lambda, y) = 0_{2\times 2}$.