ESAIM: PROCEEDINGS, **May 2009, Vol. 27, p. 54-69**
C. Besse, O. Goubet, T. Goudon & S. Nicaise, Editors

AN OPTIMIZATION-BASED ALGORITHM FOR COULOMB'S FRICTIONAL CONTACT

Florent Cadoux[1]

Abstract. The main goal of this paper is to propose a stable algorithm to compute friction forces governed by Coulomb's law in the course of the simulation of a nonsmooth Lagrangian dynamical system. The problem appears in computational mechanics, to simulate the dynamics of granular materials, robots, *etc.* Using a classical impulse-velocity formulation of Coulomb's law to model friction, and a semi-implicit time discretization scheme, we get a set of linear, non-linear and complementarity equations which has to be solved at each timestep. Two mutually dual parametric convex optimization problems coupled with a fixed point equation appear naturally. By solving (one of) these optimization problems iteratively within a damped nonsmooth-Newton algorithm, it is possible to decrease some infeasibility criterion and hopefully converge to a solution of the system. Numerical results are provided, which show that the number of iterations needed by the algorithm is very small in general and that the method is stable.

Résumé. Cet article propose un algorithme stable pour le calcul des forces de frottement régies par la loi de Coulomb lors de la simulation d'un système dynamique lagrangien non-régulier. Ce problème apparaît en mécanique numérique, pour simuler la dynamique de matériaux granulaires, robots, *etc.* En utilisant une formulation classique, dite impulsion-vitesse, de la loi de Coulomb pour modéliser le frottement, et un schéma de discrétisation semi-implicite, on obtient un ensemble d'équations linéaires, non-linéaires et de complémentarité, que l'on doit résoudre à chaque pas de temps. Deux problèmes d'optimisation convexes duaux l'un de l'autre, couplés à une équation de point fixe, apparaissent naturellement. En résolvant l'un ou l'autre de ces problèmes d'optimisation itérativement dans un algorithme de Newton avec recherche linéaire, on peut diminuer un certain critère d'infaisabilité avec l'espoir de converger vers une solution du système. Des résultats numériques sont fournis, qui montrent que le nombre d'itérations nécessaires est très petit en général, et que la méthode est stable.

Introduction

The simulation of the dynamics of a mechanical system involving friction at contact points is an active area of research with a broad range of applications, from civil engineering [13] to robotics [4] and computer graphics [11]. It can be used to compute or control the motion, discuss equilibria, and even access information unavailable to experiments : for instance, the simulation of large systems of dry granular material provides insight on the repartition of forces in a pile of stones or sand grains.

In the first place, physical laws must be chosen to model friction (what is the force between two bodies which rub against each other ?) and impacts (what is the velocity of two bodies after an impact, knowing the velocities before the event ?). We advocate the use of Coulomb's law of friction, described below in section 1, which is physically reasonable in many practical situations, along with a restitution coefficient to

[1] Inria, 655 av. de l'Europe, 38334 Montbonnot

Article published by EDP Sciences and available at http://www.edpsciences.org/proc or http://dx.doi.org/10.1051/proc/2009019

model rebound. Many other approaches modify the problem by regularizing the friction law (this is often called *Molecular Dynamics*) [21] or by somehow linearizing it, typically using a polyhedral approximation of the second-order cone involved in Coulomb's law [3]. By contrast, we use an exact impulse-velocity formulation (technique often referred to as *Contact Dynamics*).

Different sources of nonsmoothness make the problem much more difficult, both theoretically and practically [6], than classical (that is to say, smooth) Lagrangian mechanics where the motion takes place on a manifold. When unilateral contact is involved, collisions can take place during which the motion of the dynamical system can change very quickly. For this reason, jumps must be expected in the velocities in general. In this paper, we focus on numerical aspects and simulation ; from this viewpoint, we deal with nonsmoothness as follows:

- unilateral constraints (two elements cannot penetrate each other) are enforced via the impulse-velocity formulation (section 1),
- friction forces governed by Coulomb's law (a nonsmooth, multivalued law) are computed through an optimization-based algorithm, which is the main contribution of this paper,
- and jumps in velocities are handled by a time-stepping scheme (which, as opposed to event-driven schemes, does not consider every event, such as creation or disappearance of contact, separately).

An overview of algorithms and numerical methods used in frictional contact Mechanics can be found in [7]. A well-known exact formulation of Coulomb's law is the so-called *augmented Lagrangian* formulation, investigated by Alart-Curnier [1] (who solved the problem using a Newton-like algorithm, revisited in [8]), De Saxcé-Feng [9] (who used a solver based on Uzawa's algorithm) and Simo-Laursen [17]. Moreau [13,15], Jourdan-Alart-Jean [10] used a Gauss-Seidel like algorithm which solves the contact problem independently at each contact point, and then iterates until a global solution is found. In this paper, we propose a novel approach which handles all contacts globally and uses the structure of the problem to exhibit a natural optimization problem, namely a parametric second-order cone programming problem, coupled with a fixed point equation. The optimization problem can be solved iteratively until a solution to the fixed point problem, hence the contact problem, is found.

The paper is organized as follows. In section 1, we describe the physical meaning of Coulomb's law for dry frictional contact, and the two classical ways to formulate it as a complementarity constraint, either between the gap and the force or between the impulse and the velocity. In section 2, we recall the equations of motion for a Lagrangian mechanical system with contact forces governed by the impulse-velocity formulation of Coulomb's law, and we use a semi-implicit discretization scheme to derive a system of equations (the one-step problem) to be solved at each timestep. In section 3, we reformulate this sytem as a parametric optimization problem coupled with a fixed-point equation. The ability to practically solve and differentiate the parametric problem is finally used in section 4 in a damped nonsmooth Newton algorithm to minimize some infeasibility criterion, with the aim of converging to a solution of the one-step problem. Section 5 gathers some simple numerical experiments to assess the efficiency of the approach.

1. Coulomb's law of friction

Models used to describe friction depend on the nature of the materials in contact. The science of *tribology* is devoted to the design of different friction laws to model the various cases met by practitioners. We will focus on a simple yet often acceptable law proposed by Coulomb around 1770-1780, which depends only on a single coefficient and captures a crucial physical phenomenon of friction: the treshold of sliding.

1.1. **Local forces and velocities**

Let us consider a mechanical system in dimension d (typically $d = 2$ or $d = 3$) involving n punctual contact points at a given instant t. In this section only, we consider that t is fixed and the dependence on t is dropped. For each contact, labelled by $i \in 1, \dots n$ we denote by A^i and B^i the two contacting bodies and assume that the contacting surfaces are smooth enough for a tangent plane to be defined at contact point, so that terms like "tangential force" or "normal velocity" make sense. Taking body B^i as a reference, consider the relative

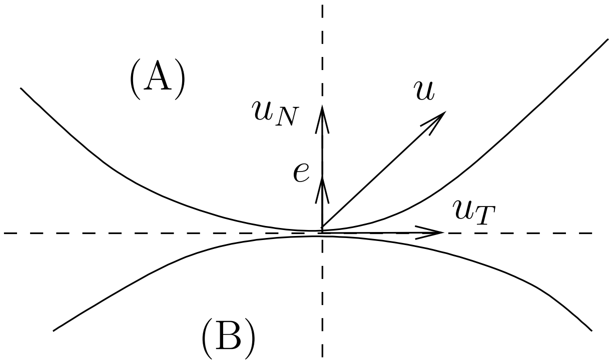

FIGURE 1. Body A^i and B^i with tangent and normal spaces

velocity[1] $\mathbf{u}^i$ of A^i with respect to B^i and the force $\mathbf{r}^i$ applied by B^i onto A^i. The normal and tangential components of a vector x are denoted x_N and x_T respectively.

1.2. **Disjunction formulation**

At each contact, we are given a coefficient of friction μ^i ($0 \le \mu^i \le +\infty$), whose physical meaning is discussed below in this subsection. Let us define the second-order cone with coefficient μ^i (the friction cone) as

$$K_{\mu^i} := \{\|x_T\| \le \mu^i x_N\} \subset \mathbf{R}^d, \tag{1}$$

whose dual cone is (with the convention that $\frac{1}{0} = \infty$ and $\frac{1}{\infty} = 0$)

$$K^*_{\mu^i} = K_{\frac{1}{\mu^i}} = \{\|x_T\| \le \frac{1}{\mu^i} x_N\} \subset \mathbf{R}^d. \tag{2}$$

The product L of all friction cones and its dual cone L^* are

$$L := \prod_{i=1}^n K_{\mu^i} \subset \mathbf{R}^{nd} \text{ and } L^* = \prod_{i=1}^n K^*_{\mu^i} \subset \mathbf{R}^{nd}. \tag{3}$$

Then Coulomb's law is usually formulated as a *disjunction*, that is, one of the three following cases must occur for each $i \in 1, \dots n$:

- take off : $\mathbf{r}^i = 0$ and $\mathbf{u}^i_N \ge 0$,
- sticking : $\mathbf{r}^i \in K_{\mu^i}$ and $\mathbf{u}^i = 0$,
- sliding : $\mathbf{r}^i \in \partial K_{\mu^i} \setminus 0$, $\mathbf{u}^i_N = 0$, $\mathbf{u}^i_T \ne 0$ opposed to $\mathbf{r}^i_T$: $\exists \alpha^i > 0$, $\mathbf{r}^i_T = -\alpha^i \mathbf{u}^i_T = -\alpha \mathbf{u}^i$.

The physical meaning of Coulomb's law is the following : the contact can cease (take off case), which means that the normal relative velocity is positive, but then there is no force between the two bodies : body B^i cannot push or pull onto body A^i when the contact breaks. In other words, there is no attractive force, no glue between the bodies (this models dry friction, without adherence) and no repulsive force either (a restitution coefficient will be added later to model rebound, but so far we are only discussing the contact force). The contact can also be maintained with zero relative velocity. In this case, the contact force can lie anywhere in its cone. Finally, if body A^i is sliding on body B^i, the relative velocity is tangent. In this case, the friction force must be on the boundary of its cone, and the tangent force must be collinear to the relative velocity with the opposite direction.

[1] As a general rule in this paper, bold letters will denote functions of time, whereas their discrete counterpart will be noted in normal font.

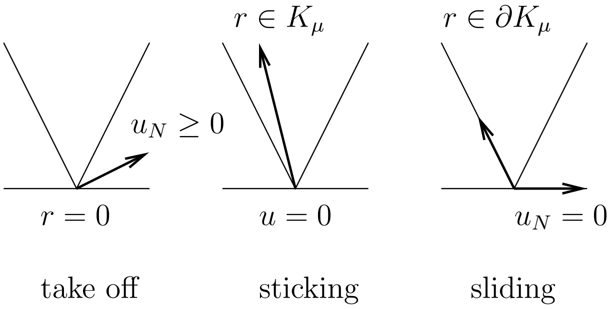

Figure 2. Three cases of Coulomb's law

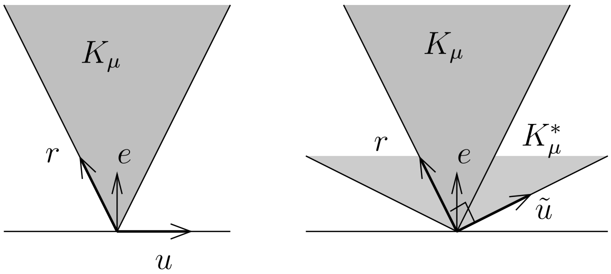

Figure 3. Change of variables $u \to \widetilde{u}$

This is often called the "maximimum dissipation principle" [15]. Note that the three cases are not mutually exclusive. For instance, the case where $\mathbf{r}^i = 0$ and $\mathbf{u}^i = 0$ can be seen either as a (slow) take off or as sticking. The value of parameter μ^i depends on the nature of the materials in contact (from $\mu^i = 0$ for a perfect contact with no friction at all, to $\mu^i = \infty$ for very rough and coarse surfaces).

This formulation as a disjunction does not seem convenient neither theoretically nor practically. It suggests algorithms based on a tree search to explore all possibilities (3^n if there are n contacts in the system) until one is found which satisfies both Coulomb's law and the equations of motions (derived below in section 2).

1.3. Complementarity

Fortunately, this apparently disparate set of three different rules has a lot of structure, as described for instance by De Saxcé-Feng [9]. Coulomb's law does not derive from a potential, even non differentiable; this can be checked using the cyclic-monotonicity condition by Rockafellar [16] as in [9]. However, it does derive from a so-called *bipotential.* The main trick is to change variables by setting

$$\widetilde{\mathbf{u}}^i = \mathbf{u}^i + \mu^i \|\mathbf{u}^i_T\| e^i \tag{4}$$

where $e^i \in \mathbb{R}^d$ is the unit normal vector from B^i to A^i at contact i. Coulomb's law can be expressed in a much more compact way as a second-order cone complementarity constraint:

$$\forall i \in 1, \dots n, : K_{\mu^i} \ni \mathbf{r}^i \perp \widetilde{\mathbf{u}}^i \in K^*_{\mu^i} \quad \iff \quad L \ni \mathbf{r} \perp \widetilde{\mathbf{u}} \in L^* \tag{5}$$

where $\mathbf{r} := (\mathbf{r}^1, \dots \mathbf{r}^n)$ and $\widetilde{\mathbf{u}} := (\widetilde{\mathbf{u}}^1, \dots \widetilde{\mathbf{u}}^n)$.

Indeed, (5) is equivalent to the following disjunction:

- either $\mathbf{r}^i = 0$, then automatically $\mathbf{r}^i \perp \widetilde{\mathbf{u}}^i$ for all $\widetilde{\mathbf{u}}^i \in K^*_{\mu^i}$; in turn, $\widetilde{\mathbf{u}}^i \in K^*_{\mu^i} \Leftrightarrow \mathbf{u}^i_N \geq 0$; this is the "take off" case.

- or $\widetilde{\mathbf{u}}^i = 0$ (so that $\mathbf{u}^i = 0$) and automatically $\mathbf{r}^i \perp \widetilde{\mathbf{u}}^i$ for all $\mathbf{r}^i \in K_{\mu^i}$; this is the "sticking" case
- or $\mathbf{r}^i$ and $\widetilde{\mathbf{u}}^i$ are both non-zero, then $\mathbf{r}^i \in \partial K_{\mu^i} \setminus 0$, $\widetilde{\mathbf{u}}^i \in \partial K^*_{\mu^i} \setminus 0$; so $\mu^i \widetilde{\mathbf{u}}^i_T = \widetilde{\mathbf{u}}^i_N$, and $\mathbf{u}^i_N = 0$ with $\mathbf{u}^i_T$ opposed to $\mathbf{r}^i_T$; this is the "sliding" case.

This simple idea of replacing $\mathbf{u}$ with $\widetilde{\mathbf{u}}$ in the formulation yields an elegant and compact formulation of Coulomb's law as a non-linear complementarity problem posed on two dual second-order cones. In the sequel, we will see that the complementarity formulation can be seen naturally as the optimality conditions (the Karush-Kuhn-Tucker system) of two mutually dual convex optimization problems coupled with a fixed-point equation.

1.4. Impulse-velocity versus gap-force formulation

The relation between $\mathbf{r}$ and $\mathbf{u}$ described in subsection 1.2 is called the impulse-velocity formulation of Coulomb's law, since complementarity occurs essentially between normal velocity $\mathbf{u}_N$ and force (or impulse) $\mathbf{r}$. It is adapted both to dynamic and quasi-static motion. On the other hand, the law is sometimes modelled using the so-called gap-force formulation, where complementarity occurs between the gap (defined as the distance, in some sense, between two potentially contacting objects) and the actual contact force between them. It is adapted to quasi-static motion, and has the advantage of enforcing non-penetrability explicitly, whereas the impulse-velocity formulation handles it implicitly through the fact that $\mathbf{u}_N$ is maintained non-negative. As a consequence, the non-penetration constraints can be slightly violated in the course of the time integration, when the impulse-velocity formulation is used[2]. On the other hand, more convergence results are available for the impulse-velocity formulation [12, 18] and it is considered to lead to more stable time-integration schemes.

2. Equations of motion

2.1. Smooth Lagrangian mechanical systems

Let us consider a mechanical system in dimension d with finitely many degrees of freedom parametrized by $\mathbf{q}(t) \in \mathbb{R}^m$. So far, only perfect (ie, without friction) bilateral constraints are allowed in the system (unilateral contact, possibly with friction, will be introduced later). Generalized velocities are defined as

$$\mathbf{v}(t) := \frac{d\mathbf{q}}{dt}(t)$$

and the equations of motion can be found using the formalism of Lagrangian Mechanics and the principle of virtual work ; they can be written in a general form as:

$$\mathbf{M}(\mathbf{q}(t))\frac{d\mathbf{v}}{dt}(t) + \mathbf{N}(\mathbf{q}(t), \mathbf{v}(t)) + \mathbf{F}_{\text{int}}(t, \mathbf{q}(t), \mathbf{v}(t)) = \mathbf{F}_{\text{ext}}(t), \tag{6}$$

where

- the matrix $\mathbf{M}(\mathbf{q})$, called the mass matrix, contains all the masses and the moments of inertia, in most applications one has $\mathbf{M}(\mathbf{q}) \in \mathbf{S}^{++}_n$ (ie, $\mathbf{M}(\mathbf{q})$ is a symmetric positive definite matrix),
- the vector $\mathbf{F}_{\text{int}} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is the nonlinear force between bodies, called also the internal forces which are not necessarily derived from a potential,
- the vector $\mathbf{F}_{\text{ext}} : \mathbb{R} \to \mathbb{R}^n$ collects all the external applied loads,
- and the vector
$$\mathbf{N}(\mathbf{q}, \mathbf{v}) = \left[\frac{1}{2}\sum_{k,l} \frac{\partial \mathbf{M}_{ik}}{\partial \mathbf{q}_l} + \frac{\partial \mathbf{M}_{il}}{\partial \mathbf{q}_k} - \frac{\partial \mathbf{M}_{kl}}{\partial \mathbf{q}_i}, i = 1 \ldots n\right] \tag{7}$$
collects the nonlinear inertial terms i.e., the gyroscopic accelerations.

[2] Non-penetrability can be retrieved a posteriori though, by projecting the solution onto the set of configurations with non-negative gap.

In the remaining part of the paper, the equation of motion (6) will be written in a more condensed form as

$$\mathbf{M}(\mathbf{q}(t))\frac{d\mathbf{v}}{dt}(t) = \mathbf{F}(t, \mathbf{q}(t), \mathbf{v}(t)), \tag{8}$$

where the vector $\mathbf{F} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ collects the terms $\mathbf{F}_{\text{int}}, \mathbf{F}_{\text{ext}}$ and $\mathbf{N}(\mathbf{q}, \mathbf{v})$.

2.2. Nonsmooth Lagrangian mechanical systems with unilateral contact and friction

Unilateral contraints can be introduced in the system and modelled using Lagrange multipliers, or using directly the contact forces written in the local frame at each contact. We will use the second option, which deals with variables that have a direct physical meaning. For each contact $i \in 1, \dots n$, we label arbitrarily body A^i and body B^i and we define $M_A^i = M_B^i$ as the contact point, with M_A^i attached to body A^i and M_B^i attached to body B^i. Then parameters $\mathbf{q}(t)$ can be used to find the position of M_A^i and M_B^i :

$$\exists f_A^i, f_B^i : \mathbb{R}^m \longrightarrow \mathbb{R}^d : M_A^i(t) = f_A^i(\mathbf{q}(t)), \; M_B^i(t) = f_B^i(\mathbf{q}(t))$$

and, differentiating with respect to time, their velocity

$$\frac{dM_A^i}{dt} = \text{Jac}[f_A^i](\mathbf{q}(t)) \; \mathbf{v}(t), \; \frac{dM_B^i}{dt} = \text{Jac}[f_B^i](\mathbf{q}(t)) \; \mathbf{v}(t)$$

(where Jac[$\cdot$] denotes the Jacobian operator) and finally their relative velocity $\mathbf{u}^i(t)$

$$\mathbf{u}^i(t) = \frac{dM_A^i}{dt} - \frac{dM_B^i}{dt} = (\text{Jac}[f_A^i] - \text{Jac}[f_B^i])(\mathbf{q}(t)) \; \mathbf{v}(t).$$

Gathering all local relative velocities $\mathbf{u}^i(t)(i = 1, \dots n)$ into $\mathbf{u}(t) = (u^1(t), \dots u^n(t))$, this can be written in matrix form as

$$\mathbf{u}(t) = \mathbf{H}(t)\,\mathbf{v}(t), \quad \mathbf{H}(t) := \begin{pmatrix} \text{Jac}[f_A^1] - \text{Jac}[f_B^1] \\ \text{Jac}[f_A^2] - \text{Jac}[f_B^2] \\ \vdots \end{pmatrix} (\mathbf{q}(t)) \tag{9}$$

so that the vector of local velocities $\mathbf{u}(t)$ is a linear function of the generalized velocities $\mathbf{v}(t)$ (it could be affine, in case of a time-dependent parametrization or moving reference frame, without changing the rest of the argument).

Introducing the contact force[3] $\mathbf{r}^i(t)$ from B^i to A^i at each contact i, and gathering them in vector $\mathbf{r}(t) = (\mathbf{r}^1(t), \dots \mathbf{r}^n(t))$, the principle of virtual work yields the new equation of motion as[4]

$$\mathbf{M}(\mathbf{q}(t))\frac{d\mathbf{v}}{dt}(t) = \mathbf{H}(t)^\top \mathbf{r}(t) + \mathbf{F}(t, \mathbf{q}(t), \mathbf{v}(t)). \tag{10}$$

Equation (10), along with Coulomb's law (4)-(5), form the continuous time problem. The model would be more complete with a restitution law (to allow for rebound) ; we will introduce it later in the discrete time model for convenience. Let us now turn to time discretization.

[3] Since the motion is nonsmooth, it should actually be an impulse and should be represented by a distribution, not a function, to allow for impacts.

[4] Once again, we write $\frac{d\mathbf{v}}{dt}$ as if $\mathbf{v}$ was a smooth function; this is abusive, since $\mathbf{v}$ is nonsmooth in general, but this is not the object of the paper to discuss the continuous time dynamics.

2.3. Time discretization

In what follows, we assume that a time step δt is fixed and that we know an approximation q_k, v_k of the configuration $\mathbf{q}(t_k), \mathbf{v}(t_k)$ at time t_k (for some k, the number of timesteps already performed). We want to compute an approximation q, v of the configuration $\mathbf{q}(t_{k+1}), \mathbf{v}(t_{k+1})$ at time $t_{k+1} := t_k + \delta t$. Assuming that $\mathbf{H}(\mathbf{q})$ and $\mathbf{F}(t, \mathbf{q}, \mathbf{v})$ vary smoothly, we use a semi-implicit time discretization scheme with implicit discretization for $\mathbf{r}(t)$ and explicit discretization for $\mathbf{H}(\mathbf{q})$ and $\mathbf{F}(t, \mathbf{q}, \mathbf{v})$. More precisely, we discretize (10) by

$$\mathbf{M}(q_k)(\mathbf{v}(t_{k+1}) - v_k) \approx \mathbf{H}(t_k)^\top \mathbf{r}(t_{k+1})\, \delta t + \mathbf{F}(t_k, q_k, v_k)\, \delta t. \tag{11}$$

To simplify the notation, we set

$$M := \mathbf{M}(q_k), \quad f := -\mathbf{M}(q_k)\, v_k - \mathbf{F}(t_k, q_k, v_k)\, \delta t, \quad H := \mathbf{H}(t_k)$$

and the discretized equation of motion writes

$$H^\top r = Mv + f \tag{12}$$

where $v \in \mathbb{R}^m$ and $r \in \mathbb{R}^{nd}$ are the unknown generalized velocity (approximation for $\mathbf{v}(t_{k+1})$) and contact impulse (approximation for $\mathbf{r}(t_{k+1})\ \delta t$) at time t_{k+1}, and the rest is data depending only on values which are known at t_k : M is an $m \times m$ symmetric positive definite matrix (again called the mass matrix), H is an $m \times nd$ matrix and $f \in \mathbb{R}^m$ is a vector. Following Moreau [13], more elaborate time discretization schemes are possible which result in the same equation (12), so that the same method can be used to solve them.

To discretize Coulomb's law (5), we also need a discrete version of the local velocities $\mathbf{u}(t)$; using the assumption that $\mathbf{H}$ varies smoothly, we replace $\mathbf{H}(t_{k+1})$ by $\mathbf{H}(t_k)$ in (9) and we define u, the approximation of $\mathbf{u}(t_{k+1})$, by

$$u := H\, v. \tag{13}$$

2.4. Restitution law

In addition to friction, we allow for rebound between particles. Following Moreau [13] again, we adopt a simple rule based on a single coefficient $\rho^i \in [0, 1]$ so that $\rho^i = 0$ stands for a completely soft impact and no rebound at all, whereas $\rho^i = 1$ stands for a perfect rebound. Coefficient ρ^i is introduced in the discretized model by replacing, in the expression of $\widetilde{u}$, the normal local velocity u^i_N at contact i by $u^i_N + \rho^i u^i_{N,k}$ (where $u^i_{N,k}$ is an approximation for $\mathbf{u}^i_N(t_k)$); in other words, we discretize (4) as follows

$$\widetilde{u}^i := u^i + \rho^i u^i_{N,k}\, e^i + \mu^i \|u^i\|\, e^i. \tag{14}$$

The reason why this change in the expression of $\widetilde{u}$ does model rebound is the following. Whenever $r^i \neq 0$, there holds $u^i_N + \rho^i u^i_{N,k} = 0$ (sticking or friction case in Coulomb's law), which means that

$$u^i_N = -\rho u^i_{N,k}.$$

This is the simplest restitution law (sometimes called Newton's law) , which states that the normal velocity after an impact is equal to minus the normal velocity before the impact amortized by ρ^i. It is considered valid for bodies with a geometry similar to spheres. On the contrary, for bodies with more complicated shape, it is known to be poor and can even entail *creation* of energy in the system [19], in contradiction with thermodynamics and common sense. For this reason, the chosen restitution law must be used with care and considered as a phenomenological rule more than a physical theory of impacts.

Let us introduce vector c of (known) damped normal velocities at t_k by

$$c := (\rho^1 u^1_{N,k}, \ldots \rho^n u^n_{N,k}) \in \mathbb{R}^n, \tag{15}$$

vector s of (unknown) sliding velocities at t_{k+1} by

$$s := (\|u_T^1\|, \ldots, \|u_T^n\|) \in \mathbb{R}^n, \tag{16}$$

and matrices

$$E := \begin{pmatrix} e^1 & 0 & \ldots \\ 0 & e^2 & \\ \vdots & & \ddots \end{pmatrix} \in \mathbb{R}^{nd\times n} \text{ and } D := \begin{pmatrix} \mu^1 & 0 & \ldots \\ 0 & \mu^2 & \\ \vdots & & \ddots \end{pmatrix} \in \mathbb{R}^{n\times n}. \tag{17}$$

Then, using (13), equation (14) can be written in matrix form as

$$\widetilde{u} = Hv + E(c + Ds). \tag{18}$$

Finally, we discretize Coulomb's law formulated as a complementarity constraint (5) by

$$L \ni r \perp \widetilde{u} \in L^*. \tag{19}$$

2.5. The one-step problem

Putting together the discretized equations of motion (12), the definition (18) of $\widetilde{u}$, the discretized version of Coulomb's law (19) and the definition (16) of s, we get the following set of equations:

$$\left\{ \begin{array}{lll} H^\top r = Mv + f & \text{[Newton's law]} & (a) \\ \widetilde{u} = Hv + E(c + Ds) & \text{[kinematics]} & (b) \\ L \ni r \perp \widetilde{u} \in L^* & \text{[Coulomb's law]} & (c) \\ s^i = \|\widetilde{u}_T^i\|, \ i \in 1, \ldots n & & (d) \end{array} \right. \tag{20}$$

where we used in (d) the fact that $\widetilde{u}_T = u_T$. Our goal is to find a solution (v, u, r, s) to this set of equations for given data H, M, f, E, c, D, L.

3. Optimization-based formulation

In this section, we will view (20)-(a-c) as the optimality conditions of a minimization problem, where s is considered as a fixed parameter, in order to obtain the above-mentionned decomposition of the one-step problem into a parametric optimization problem coupled with a fixed-point problem.

3.1. Optimality conditions of a conic problem

We first recall general sufficient optimality conditions for a convex optimization problem posed on a cone. Let $K \in \mathbb{R}^m$ be a closed convex cone, $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ a differentiable convex function, and $g : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ an affine function. If $(\bar{v}, \bar{r})$ solves the following system

$$\left\{ \begin{array}{l} \nabla J(v) = \mathrm{Jac}[g](v)^\top r \\ K \ni g(v) \perp r \in K^* \end{array} \right. \tag{21}$$

then $\bar{v}$ is an optimal solution to the following optimization problem

$$\left\{ \begin{array}{l} \min J(v) \\ g(v) \in K. \end{array} \right. \tag{22}$$

3.2. Application to the one-step problem

In (20), neglect equation (d) for the moment and consider s as a fixed parameter. The key observation is that, setting

$$J(v) := \frac{1}{2}v^\top Mv + f^\top v \quad \text{ and } \quad g(v) := Hv + E(c + Ds)$$

we observe that the remaining equations (20)-(a-c) in $(v, \tilde{u}, r)$ are the sufficient optimality conditions (22) for the minimization problem

$$\begin{cases} \min J(v) & \text{(quadratic, strictly convex)} \\ g(v) \in L^* & \text{(conic constraints).} \end{cases} \tag{23}$$

Unless this problem is infeasible, it has a unique solution v (M is positive definite). The corresponding $\widetilde{u}$ is then uniquely determined by (20)-(b) but observe that r is not unique if the rows of H are linearly dependent. In fact, setting

$$J^*(r) := \frac{1}{2}r^\top(HM^{-1}H^\top)r - (HM^{-1}f - E(c + Ds))r$$

in which $HM^{-1}H^\top$ may not be invertible, standard duality theory says that r is given as a solution of the dual problem of (23), namely

$$\begin{cases} \min J^*(r) & \text{(quadratic, convex)} \\ r \in L & \text{(conic constraints).} \end{cases} \tag{24}$$

Non-uniqueness of r is not surprising from a mechanical viewpoint, since a rigid body can be wedged and subject to non-unique contact forces ; the example of a bead in a corner below (subsection 5.3) shows such an example. As a consequence, the solution for velocities (v and $\widetilde{u}$) are unique or non-existing, but the solution for forces (or impulses) r must no be expected to be unique.

Using classical tricks [5], both problems can be reformulated into standard second order-cone programs (SOCP), that is to say optimization problems of the form

$$\begin{cases} \min c^\top x & \text{(linear objective function)} \\ Ax = b & \text{(affine constraints)} \\ x \in \prod_j K_j & \text{(conic constraints)} \end{cases}$$

where K_j are either second-order cones or polyhedral cones. The SOCP problem is well studied in the literature [2] and efficient algorithms and codes are available; we used the solver SeDuMi [20] for 3D problems ($d = 3$). For 2D problems ($d = 2$), the SOCP reduces to a quadratic program (QP) and any standard QP solver can be used. Finally, for each value of $s \in \mathbb{R}^n_+$, we can either find $v(s), \widetilde{u}(s), r(s)$ which satisfy (20)-(a-c) or prove that no such triplet exist, for the computational price of solving a SOCP problem. The idea developed in section 4 is then to solve iteratively equation (20)-(d) :

$$s^i = \|\widetilde{u}^i_T(s)\|, \ i \in 1, \dots n \tag{25}$$

which is a fixed point equation. We could try to solve it using fixed-point iterations, but

- simple examples and experiments showed that the fixed point function is not a contraction in general,
- it is actually possible to differentiate[5] function $s \longrightarrow \widetilde{u}(s)$ directionally, which allows use of Newton's algorithm to solve the fixed point equation ; in the experiments done so far, the cost of computing the Jacobian matrix is cheap compared to the solution of (23) or (24), so there is no reason not to take advantage of the ability to differentiate the fixed point function.

For these reasons, we used a nonsmooth Newton's algorithm to solve the fixed point equation (25). This algorithm is the object of the next section.

[5] This point needs a lot more theoretical explanations, which are beyond the scope of this paper. We only refer to subsection 4.1 to see how we perform the differentiation practically.

4. Nonsmooth Newton algorithm

We now turn to the problem of solving the fixed-point equation (25). This can be done using a damped Newton's algorithm : we first compute the Newton step, and then perform a line search in the Newton direction until the least-square function

$$\phi(s) := \frac{1}{2}\sum_i \bigl(s^i - \|\widetilde{u}_T^i(s)\|\bigr)^2$$

decreases.

4.1. **Differentiation**

In this subsection, we explain how we compute a generalized Jacobian matrix of $s \longrightarrow \widetilde{u}(s)$ practically, and we keep as a future research direction to understand in what sense it *is* indeed a generalized Jacobian matrix. In other words, we postpone the analysis of the nonsmooth function $s \longrightarrow \widetilde{u}(s)$. The idea, taken from the implicit functions theorem, is simply to derive the optimality conditions (20)-(a-c) with respect to s. Indeed, having solved (23), we obtain a list of those constraints in (1) and (2) which are active at $v(s)$ and $\widetilde{u}(s)$; then we write

$$G(v(s), \widetilde{u}(s), r(s), s) = 0 \Leftrightarrow G(w(s), s) = 0 \text{ with } w = (v, \widetilde{u}, r)$$

where G collects (20)-(a-b), the active constraints at $v(s)$ and $\widetilde{u}(s)$, and the orthogonality constraint $\widetilde{u} \cdot r = 0$. We then differentiate with respect to s, and get

$$\frac{\partial G}{\partial w}\frac{dw}{ds} + \frac{\partial G}{\partial s} = 0 \Rightarrow \frac{dw}{ds} = -\Bigl(\frac{\partial G}{\partial w}\Bigr)^{-1}\frac{\partial G}{\partial s}. \tag{26}$$

To compute $\frac{dw}{ds}$, hence $\frac{d\widetilde{u}}{ds}$, it suffices to solve a set of linear systems (26).

4.2. **Initialization**

Since we want to solve equation (25), we can restrict ourselves to $s \in \mathbb{R}^n_+$. However, for some $s \in \mathbb{R}^n_+$, the optimization problem (23) may be infeasible, so that function ϕ may not be defined. If the friction coefficients μ^i are all non-zero, however, it is always possible to find a feasible initial point for Newton's algorithm, by setting

$$s_0 := \max(-\frac{c_i}{\mu^i}, 0)$$

so that $v = 0$ is feasible in (23). If $\mu^i = 0$ for some i, it may not be possible to find an initial feasible point. For instance, if H is the null matrix, $c_i < 0$ and $\mu^i = 0$ for some i, it is clear that no feasible s exists.

4.3. **Computation of a Newton step**

To find a fixed point of $F(s) := (\|\widetilde{u}_T^i(s)\|)_{i=1,\dots n}$, where F can be directionally differentiated according to subsection 4.1, we choose an initial s_0 as in subsection 4.2, compute $\widetilde{u}(s)$ and a generalized Jacobian matrix $\text{Jac}[\widetilde{u}](s)$, then $F(s)$ and a generalized Jacobian matrix $\text{Jac}[F](s)$, and finally solve the linear equation

$$F(s) + \text{Jac}[F](s)\, ds = s + ds$$

to get the Newton step ds. Function $\widetilde{u}(s)$ is evaluated again at $s + ds$, and if the least-square criterion ϕ has decreased, we replace s with by $s + ds$ and loop. This is what happens almost all the time. Exceptionally however, the Newton step ds does not decrease the least-square criterion ϕ, or (23) is infeasible at $s + ds$ so that ϕ is undefined, or $s + ds \notin \mathbb{R}^n_+$. In these cases, we perform a line search.

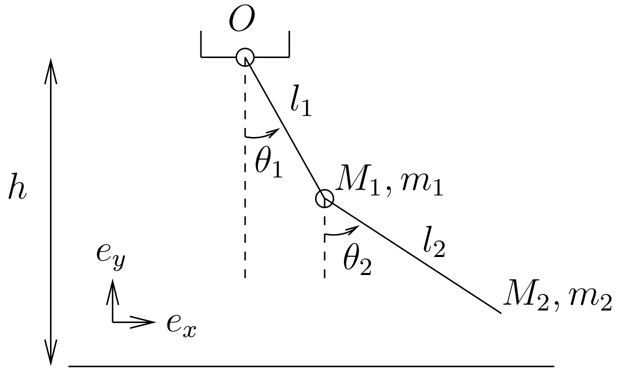

Figure 4. Double pendulum with friction

4.4. **Line-search**

We use the Armijo rule for updating the step ds : the Newton step ds is halved iteratively until ϕ is found to decrease. This happens necessarily, since

$$\nabla\phi(s)\cdot ds = (F(s)-s)^\top(\mathrm{Jac}[F](s)-I)(\mathrm{Jac}[F](s)-I)^{-1}(s-F(s)) = -\|F(s)-s\|^2$$

so that the Newton direction ds is a descent direction for ϕ, unless $F(s)=s$ which means that convergence already occured. Finally, the algorithm simply consists in minimizing ϕ over its (not known explicitly) domain, using a damped nonsmooth Newton algorithm, until we hopefully reach a global optimum with value 0. Since ϕ decreases at each iteration, the method is garanteed to be stable : it is not subject to cycling nor diverging brutally. However, it may stall and stop at a local minimum with non-zero value, even when a global optimum with value 0 exists. In this case, the method fails; such a counterexample is given in subsection 5.4.

5. Numerical examples

In this section, we give a few simple examples and counterexamples. The one-step problem (20) is written explicitly for the double pendulum with friction in two dimensions, to illustrate the approach. The algorithm is also run on random instances of problems, to see how it performs in various situations. Finally, counterexamples are given which show that the solution to (20) is not unique in general (example of a bead wedged in a corner) and that the algorithm is not garanteed to converge to an actual solution (global minimum of 0 for function ϕ) when one exists, since the damped Newton algorithm may stop at a local minimum of ϕ.

5.1. **Double pendulum with friction**

In this subsection, we illustrate the method by writing and solving numerically the problem for a double pendulum with friction, as depicted on figure 4. The pendulum is made of two punctual masses m_1 and m_2 in M_1 and M_2. M_1 is linked to the ceiling by a rigid rod of length l_1 with no mass and a perfect ball joint. M_2 is linked to M_1 by a rigid rod of length l_2 with no mass and a perfect ball joint. The only external load is the weight due to gravity $g\in\mathbb{R}^2$. The system has two degrees of freedom, we parametrize it by $q=(\theta_1,\theta_2)$. The kinetic energy is

$$T := \frac{1}{2}m_1\dot{M_1}^2 + \frac{1}{2}m_2\dot{M_2}^2 = \frac{1}{2}(m_1+m_2)l_1^2\dot{\theta_1}^2 + \frac{1}{2}m_2(2l_1l_2\cos(\theta_1-\theta_2)\dot{\theta_1}\dot{\theta_2} + l_2^2\dot{\theta_2}^2).$$

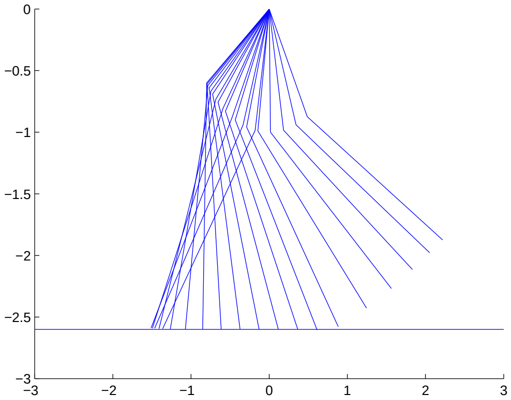

FIGURE 5. A trajectory for the double pendulum with friction

If $M_2 \cdot e_y \leq -h$, M_2 is in contact with the ground, and

$$\mathbf{H} = \frac{dM_2}{dq} = \begin{pmatrix} l_1 \cos\theta_1 & l_2 \cos\theta_2 \\ l_1 \sin\theta_1 & l_2 \sin\theta_2 \end{pmatrix}$$

(otherwise, if M_2 does not touch the ground, $\mathbf{H}$ is the empty matrix) and the Lagrange equations of motions write

$$\frac{d}{dt}\Big(\frac{\partial T}{\partial \dot{q}}\Big) - \frac{\partial T}{\partial q} = m_1 \frac{dM_1}{dq}^{\top} g + m_2 \frac{dM_2}{dq}^{\top} g + H^{\top} r$$

which, after some computations, is exactly in the form (10) with

$$\mathbf{M} = \begin{pmatrix} (m_1 + m_2) l_1^2 & m_2 l_1 l_2 \cos(\theta_1 - \theta_2) \\ m_2 l_1 l_2 \cos(\theta_1 - \theta_2) & m_2 l_2^2 \end{pmatrix}, \quad \mathbf{N} = m_2 l_1 l_2 \sin(\theta_1 - \theta_2) \begin{pmatrix} \dot{\theta_2}^2 \\ -\dot{\theta_1}^2 \end{pmatrix},$$

$$\mathbf{F}_{\text{ext}} = \begin{pmatrix} l_1 (m_1 + m_2) (\cos(\theta_1) g_x + \sin(\theta_1) g_y) \\ l_2 m_2 (\cos(\theta_2) g_x + \sin(\theta_2) g_y) \end{pmatrix} \text{ and } \mathbf{F}_{\text{int}} = 0.$$

For $m_1 = m_2 = 1$ (in kilograms), $l_1 = 1$, $l_2 = 2$, $h = 2.6$ (in meters) $\mu = 0.01$ (dimensionless) and $\rho = 0$ (dimensionless, no restitution), we obtain the trajectory drawn on figure 5. The pendulum swings from right to left (and then back) and its position is depicted every 0.2 seconds.

5.2. Random instances

In order to test the algorithm for various data, we performed experiments with random instances of the one-step problem. We used different values for n (number of contacts) and m (number of degrees of freedom), uniformly distributed values for the coefficients μ^i $(i = 1, \dots m)$ in the interval $[\mu_{min}, \mu_{max}]$, mass matrices M randomly generated by matlab's function *sprandsym*, and matrices H randomly generated by matlab's function *sprandn*. All instances where succesfully solved within a few seconds on a standard computer. The results are gathered in table 1 (in two dimensions) and 2 (in three dimensions). In two dimensions, it can be shown that function ϕ is actually piecewise quadratic; this implies that Newton's method, when started from a piece which contains a minimum, finds this minimum exactly in one iteration. This explains why the precision attained in table 1 is around machine accuracy. It may need several iterations to find the right piece, though. In three dimensions, function ϕ is not piecewise quadratic anymore, but table 2 shows that the method still converges in a few iterations, with satisfying precision. These results are encouraging, but one must keep in mind that each iteration is costly (one SOCP to solve) and that these random problems may not be typical of the situations met in real-world instances.

Table 1. Random instances in 2D

instance	n	m	μ_{min}	μ_{max}	#iter	$\phi(s_{initial})$	$\phi(s_{final})$
1	10	40	0.5	2	1	42.3	3.1e-30
2	20	80	0.5	2	2	38.8	8.4e-27
3	30	120	0.5	2	3	166.1	1.6e-28
4	40	160	0.2	3	3	184.6	1.2e-27
5	50	200	0.2	3	1	303.9	2.6e-27
6	60	240	0.2	3	4	610.8	7.8e-29

Table 2. Random instances in 3D

instance	n	m	μ_{min}	μ_{max}	#iter	$\phi(s_{initial})$	$\phi(s_{final})$
1	10	40	0.5	2	2	44.8	2.2e-11
2	20	80	0.5	2	2	90.3	1.6e-11
3	30	120	0.5	2	2	369.1	9.0e-10
4	40	160	0.2	3	2	414.4	7.4e-15
5	50	200	0.2	3	2	1105.9	1.7e-10
6	60	240	0.2	3	4	1287.3	2.7e-9

5.3. Bead in a corner

The example of a bead wedged in a corner shows that the one-step problem (20) may have several isolated solutions. Consider the situation of figure 6 : a bead is in contact with two walls, with friction at contact points, and an external force is used to pull the bead away from the corner. The bead is initially at rest (no movement) and we want to compute the first timestep, that is to say, to solve the first one-step problem. From a mechanical viewpoint, it is quite clear that two outcomes are possible :

- either there are contact forces, which counterbalance the applied external load ; the bead does not move, it is wedged by friction forces
- or there are no contact forces, the bead is pulled away and has nonzero velocities u_1 and u_2.

Both situations are physically acceptable, and indeed there are two isolated minima with value 0 in function ϕ depicted on figure 7. The origin ($s = (0, 0)$, no sliding velocity) corresponds to the case where the ball remains still, and the other minimum ($s = (1, 1)$ for the chosen values of the data) corresponds to the case where the bead takes off. Depending on the chosen initial point s in our damped Newton's algorithm, it can converge to one or the other solution. In practice, we initalize the sliding velocities s with the value they had at the previous timestep, so that when there are several solutions to the one-step problem (20), the algorithm chooses one which is close to the previous solution. In addition to providing a good starting point which helps convergence of the algorithm and decreases computation time, this strategy can be seen as a way to keep track of the history of the mechanical system. This is important when simulating granular material, since it is well-known that the repartition of load in a pile of grains at equilibrium depends not only on its configuration (vector $q(t)$) but also on the way it was prepared [14].

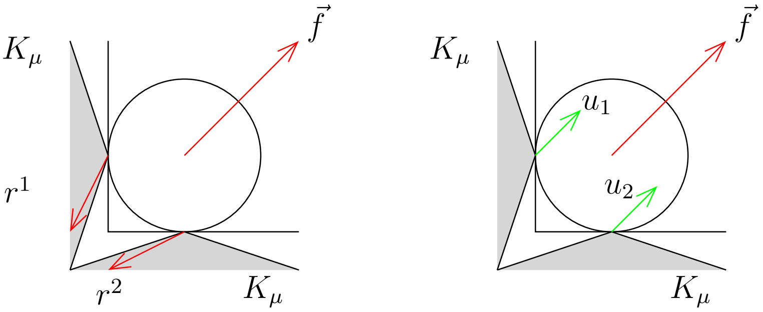

FIGURE 6. A bead wedged in a corner

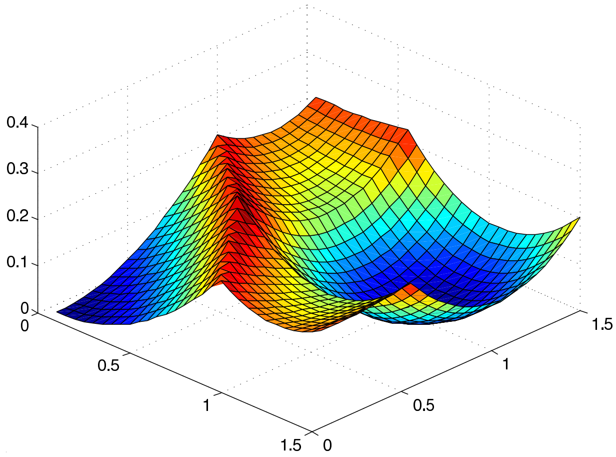

FIGURE 7. Piecewise quadratic function ϕ for the bead in a corner

5.4. Bad boy for Newton's algorithm

In this subsection, we give a counter-example which shows that even for the simplest case of one contact in two dimensions, the algorithm may converge to a point which is not a solution to the one-step problem (20). Consider the following data (for a mechanical system with two degrees of freedom):

$$M = \begin{pmatrix} 2 & -3 \\ -3 & 5 \end{pmatrix}, \quad f = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mu = 1, \quad c = 0$$

then $\phi : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ is the function depicted on figure 8. There is a local minimum for $s = 0$ with nonzero value, and a global minimum with zero value at $s = 5$. The function restricted to $[0, 2]$ or $[2, \infty[$ is convex (and quadratic, since $d = 2$). If our minimization algorithm for ϕ is initialized with $s_0 < 2$, it will converge to $s = 0$ and fail. If it is initialized with $s > 2$, it will converge to $s = 5$ and succeed. If one chooses exactly $s_0 = 2$, the function is not differentiable ; one of the two directional derivatives has to be chosen, and if we make the wrong choice the algorithm will converge to $s = 0$. This illustrates the fact that, as expected, function ϕ is nonsmooth in general, and that even with only one contact, two parameters and two dimensions, the one-step problem can be non trivial. Also note that this counterexample was built using a non-diagonal mass matrix, with a condition number around 50 (for a 2×2 matrix); experiments suggest that instances which make the method fail all have a significant amount of mass far from diagonal in the mass matrix, and a quite large condition number.

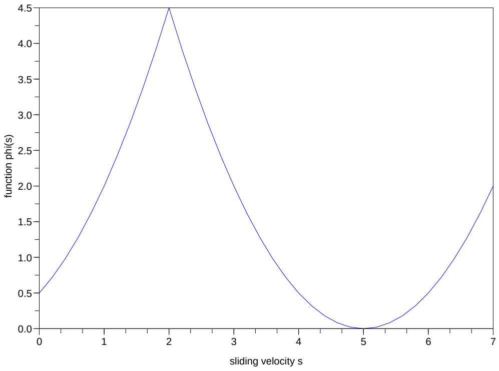

Figure 8. Piecewise quadratic function ϕ with a local minimum in 0 which is not a solution

Conclusion

Using a classical impulse-velocity formulation of Coulomb's law to model friction, a phenomenological rule to model restitution, and a semi-implicit time discretization scheme, we write the discretized equations of motion for a Lagrangian mechanical system subject to unilateral contact with friction. This results in a system of equations mixing linear, non linear and complementarity equations (the one-step problem), with interesting structure : it can be equivalently formulated as a parametric convex minimization problem coupled with a fixed point equation. This equivalent reformulation opens the way to a specific algorithm to compute a solution for the one-step problem. Numerical experiments illustrate the method and show the feasibility of the proposed algorithm. We keep as a research direction to better understand the properties of the nonsmooth function ϕ used for minimization, and the theoretical properties of the algorithm. In particular, it seems that although ϕ is nonsmooth, Newton's algorithm achieves superlinear convergence (as observed in the experiments). Future work will also include comparisons with existing techniques, based either on projected Gauss-Seidel like algorithm or on Newton's algorithm.

The author wishes to thank Vincent Acary, Claude Lemaréchal and Jérôme Malick for their help and support and the organizers of Canum 2008 for inviting him to publish this paper.

References

[1] P. Alart and A. Curnier. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput. Methods Appl. Mech. Eng.*, 92(3):353–375, 1991.

[2] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Math. Prog. Ser. B*, 95:3–51, 2003.

[3] M. Anitescu and F.A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dyn.*, 14(3):231–247, 1997.

[4] Brian Armstrong-Hélouvry. *Control of machines with friction.* The Kluwer International Series in Engineering and Computer Science. 128. Dordrecht: Kluwer Academic Publishers Group. VIII, 173 p. , 1991.

[5] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization. Analysis, algorithms, and engineering applications.* MPS/ SIAM Series on Optimization. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. Philadelphia, PA: MPS, Mathematical Programming Society, xvi, 488 p. , 2001.

[6] Bernard Brogliato. *Nonsmooth mechanics. Models, dynamics and control. 2nd ed.* Communications and Control Engineering Series. London: Springer. xx, 552 p., 1999.

[7] Bernard Brogliato and Vincent Acary. *Numerical methods for nonsmooth dynamical systems.* Springer, 2008.

[8] P.W. Christensen, A. Klarbring, J.S. Pang, and N. Stromberg. Formulation and comparison of algorithms for frictional contact problems. *Int. J. Numer. Methods Eng.*, 42(1):145–173, 1998.

[9] G. De Saxcé and Z. Q. Feng. The bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. *Math. Comput. Modelling*, 28(4-8):225–245, 1998.

[10] Franck Jourdan, Pierre Alart, and Michel Jean. A Gauss-Seidel like algorithm to solve frictional contact problems. *Comput. Methods Appl. Mech. Eng.*, 155(1-2):31–47, 1998.

[11] Danny M. Kaufman, Timothy Edmunds, and Dinesh K. Pai. Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.*, 24(3):946–956, 2005.

[12] Manuel D.P. Monteiro Marques. *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 , 1993.

[13] J-J. Moreau. Modélisation et simulation de matériaux granulaires. In *Actes du 35ème Canum*, 2003.

[14] J-J. Moreau. Modélisation et simulation de matériaux granulaires. In *Actes du 6ème Coll. Nat. Calc. Struct.*, 2003.

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

[16] R.T. Rockafellar. *Convex analysis.* Princeton, N. J.: Princeton University Press, XVIII, 451 p. , 1970.

[17] J.C. Simo and T.A. Laursen. An augmented Lagrangian treatment of contact problems involving friction. *Comput. Struct.*, 42(1):97–116, 1992.

[18] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Rev.*, 42(1):3–39, 2000.

[19] W.J. Stronge. Unraveling paradoxical theories for rigid body collisions. *J. Appl. Mech.*, 58(4):1049–1055, 1991.

[20] Jos F. Sturm. Using SeDuMi 1. 02, a MATLAB toolbox for optimization over symmetric cones. *Opt. Meth. Soft.*, 1999.

[21] Otis R. Walton. Numerical simulation of inelastic frictional particle-particle interactions. *Roco, M.C. (Ed.), Particulate Two-Phase Flow*, pages 884–911, 1993.