Preprint ANL/MCS-P1692-1109

A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics

A.Tasora [a] M.Anitescu [b]

[a] *Università degli Studi di Parma, Dipartimento di Ingegneria Industriale, 43100 Parma, Italy, tasora@ied.unipr.it*

[b] *Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA, anitescu@mcs.anl.gov*

Abstract

This paper proposes an iterative method that can simulate mechanical systems featuring a large number of contacts and joints between rigid bodies. The numerical method behaves as a contractive mapping that converges to the solution of a cone complementarity problem by means of iterated fixed-point steps with separable projections onto convex manifolds. Since computational speed and robustness are important issues when dealing with a large number of frictional contacts, we have performed special algorithmic optimizations in order to translate the numerical scheme into a matrix-free algorithm with $O(n)$ space complexity and easy implementation. A modified version, that can run on parallel computers is discussed. A multithreaded version of the method has been used to simulate systems with more than a million contacts with friction.

Key words: Large-scale multibody, contact, friction, cone complementarity problem

1. Introduction

Many engineering problems involve unilateral contacts between rigid bodies, for instance, in simulations of robotic cells and part feeders, in cam followers, in masonry stability analysis, and in packaging devices such as those depicted in Fig. 1.

The dynamical simulation of such systems is complicated by the nonsmooth nature of the frictional constraints. When the number of contacts between bodies increases to thousands or millions, as in the case of granular flows in silos or in rock-soil dynamics, the computational efficiency of traditional methods can become an issue even on supercomputers.

A straightforward approach to solve this class of problems may consist of regularization schemes, that transform the discontinuities into a stiff force field. This is, for example, the approach often adopted by discrete element schemes (DEMs) because it does not require major modifications to traditional solvers based on smooth ordinary differential equations (ODEs) [15,35,36,30]. Nevertheless, although successfully used to simulate granular flows with many contacts, the regularization approach requires small time steps to achieve numerical stability. Moreover, it forces the user to introduce artificial stiffness or heuristic parameters: actually, if the deformation of the parts is negligible, a method that can use large time steps and unconditionally rigid bodies would be more welcome.

These considerations encouraged our research on a fast, robust, and unified numerical scheme that can handle complex mechanical systems made of rigid bodies with an arbitrary number of contacts and joints. Such a scheme aims at simulating mechanical systems ranging from the simplest (articulated linkages with few bilateral kinematic pairs and motors) up to the most complex (for example, a bulldozer interacting with millions of particles of sand with the tracks and the blade).

In this context, the biggest challenge comes from the discontinuous nature of the adhesion constraints and non-interpenetration constraints; in fact, the simulation of rigid contacts embeds the solution of nonsmooth equations. To this end, the straightforward application of numerical methods for ODEs or differential algebraic equations (DAEs) is inefficient. In fact, a naive approach based on piecewise integrals is virtually impossible because it would require stopping and restarting the integrator at each discontinuity to change the active set of constraints. This could work only if there were a limited number of unilateral constraints [20,21]. Otherwise, the risk of combinatorial explosion could severely affect the computational efficiency to the point where the simulation would come to a halt [45].

The nature of nonsmooth dynamics requires the adoption of a deeper mathematical framework, where concepts like set-valued functions, inclusions, and complementarity conditions are used [33]. In particular, recent time-stepping approaches construct weak solu-

tions of the differential variational inequality (DVI) that describes the continuous time motion of rigid bodies with collision, contact, and friction. Earlier numerical methods based on differential variational inequalities can be found in [26,25,24], whereas the DVI formulation has been discussed in full generality and classified by differential index only recently, in [37,29].

Two main families of solvers spawn from the DVI formulation: those that lead to an acceleration-force complementarity problem [9,31,45] and that generate velocity-impulse, complementarity-based time-stepping methods [39,6,7]. The latter case results in schemes convergent to a vector measure differential inclusion, so named because it operates on vector measures or distributions [40]. It has the advantage that it can solve a class of problems with Coulomb friction that would be unsolvable in an acceleration-force context, as the Painlevé paradox [38].

In both cases, the introduction of inequalities in time-stepping schemes for DVI, together with a polyhedral approximation of the friction cone as a faceted pyramid, leads to linear complementarity problems (LCPs) [40], which are systems of complementary inequalities to be satisfied simultaneously [14]. Such LCPs, which are hard to solve because of their inherent nonlinear nature, must be solved at each time step in order to advance the integrator [24,40].

Most literature about this topic shows how, for a large number of contacts and rigid bodies, usual LCP solution schemes have significant limitations. In fact, classical approaches to the solution of LCP problems are based on *simplex methods*, also known as *direct* or *pivoting* methods, originating from the algorithms of Lemke and Dantzig [13]. These methods may exhibit an exponential worst-case complexity [10]. Our experience shows that, in spite of algorithmic optimizations [43], simplex methods still cannot practically handle multibody systems with more than one hundred colliding bodies.

Moreover, in the three-dimensional case, typical LCP solvers can be used only at the cost of approximating the Coulomb friction cone with faceted pyramids [40,45,6]. Not only does this expedient introduces artificial anisotropy in the friction phenomenon; it also impacts negatively the performance of LCP solvers, which is already critical in general, because the finite approximation of cones results in a much larger problem.

A precise description of the friction cone constraint in three-dimensional space would imply a nonlinear complementarity problem. This is a broader class of problems in mathematical programming, for which no off-the-shelf solvers are available. A custom method must be developed.

The above-mentioned limitations of the existing LCP approaches led us to develop a novel solution method based on a fixed-point iteration with projection on a convex set and presented in [8]. That method extended the seminal work on iterative LCP solvers by Mangasarian [28] to the LCP case with conical constraints, that is, a cone complementarity problem. In the same work we presented the convergence theory for the iteration; the scheme converges under certain conditions that do not include a small friction assumption. Applied to granular flow problems, our method demonstrated high performance and was able to solve benchmarks with up to a million dual variables.

The time-stepping scheme was proven to converge in a measure differential inclusion sense to the solution of the original continuous-time DVI [2].

In the present paper we extend our original formulation [8] in several ways.

(i) We enhance our approach to the case of both frictional contacts and bilateral constraints, either scleronomic or rheonomic (motors, imposed trajectories, etc.). This extension cannot be obtained with full theoretical guarantees of convergence for the algorithm in [8] by simply replacing the bilateral constraint with two unilateral constraints. The latter case allows for unbounded internal forces, for which the approach in [8, Corollary 1] does not apply (the resulting constraint cone is not *pointed*).

(ii) We present practical algorithmic details and optimizations that can be adopted to implement the method in a matrix-free, memory-efficient, reliable, fast, and robust way.

(iii) We demonstrate the performance of our approach for configurations that include both joint and contact-with-friction constraints.

A significant side-effect of the proposed method is that it proceeds monotonically toward the solution. If implemented in real-time applications such as virtual-reality and man-in-the-loop vehicle simulations, where the requirements on precision are less severe than those on the computational times, it can be stopped prematurely before the tolerance threshold is reached.

We believe that the CAD community will welcome the availability of this solver because of its ability of simulating generic mechanisms regardless of the number of parts, joints, and frictional contacts. To this end we developed a physics library based on this time-stepping method, written in C++ and called Chrono::Engine [41], that can be used by third parties to develop simulation software. Indeed, we used it to implement a 3D graphical interface for the interactive modeling and visualization of multibody systems. Moreover, we have already simulated many types of mechanical problems, ranging from robots to granular flows with hundreds of thousands of rigid bodies with friction.

Among the most complex tests, we simulated the granular flow of a fourth-generation pebble bed nuclear reactor. Thanks to the numerical scheme, the refueling motion of 170,000 uranium spheres was simulated on a single computer, whereas DEM methods required a supercomputer and much more CPU time [42].

2. The model

This section presents a formulation for the nonsmooth dynamics of multibody systems in the most general case of both bilateral constraints and frictional

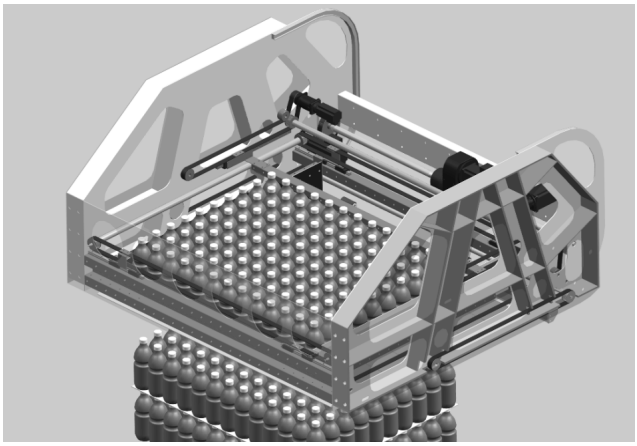

Fig. 1. Simulation of a palletizing device: a multibody problem with many frictional contacts.

contacts. We remark that frictional contacts embed also the case of unilateral constraints, since those represent a special case of contacts without friction.

2.1. *System state*

The position of the system at time t is represented by m_q generalized coordinates $\boldsymbol{q}(t) \in \mathbb{R}^{m_q}$. In the case of rigid bodies in three-dimensional space, these coordinates include the positions $\boldsymbol{x} \in \mathbb{R}^3$ of the centers of mass of all bodies, as well as the rotations of all body frames respect to the absolute frame.

We represent rotations by means of unimodular quaternions $\boldsymbol{\rho}(t) \in \mathsf{S}^3 \subset \mathbb{H}$. Since quaternions are four-dimensional numbers, each rigid body in our formulation requires 3+4=7 scalar coordinates plus one constraint $||\boldsymbol{\rho}|| = 1$ enforcing the unit length of the quaternion [1]. With this notation, we define the position vector to be $\boldsymbol{q} = \{\boldsymbol{x}^{1^T}, \boldsymbol{\rho}^{1^T}, \boldsymbol{x}^{2^T}, \boldsymbol{\rho}^{2^T}, ...\}$.

Generalized velocities are represented by the vector $\boldsymbol{v}(t) \in \mathbb{R}^{m_v}$, where for each body we consider the speed of the center of mass $\dot{\boldsymbol{x}} \in \mathbb{R}^3$ and the angular velocity $\boldsymbol{\omega}_l$, expressed in local body coordinates. Therefore a system with n bodies in three dimensions is represented by $m_v = 6n$ speed coordinates. With this notation, we define the speed vector to be $\boldsymbol{v} = \{\dot{\boldsymbol{x}}^{1^T}, \boldsymbol{\omega}_l^{1^T}, \dot{\boldsymbol{x}}^{2^T}, \boldsymbol{\omega}_l^{2^T}, ...\}^T$.

Given the angular velocity $\boldsymbol{\omega}_l$, one can obtain the time derivative $\dot{\rho}$ of the quaternion, if needed, by building the purely imaginary quaternion $\{0, \boldsymbol{\omega}_l\}$ and computing the quaternion product $\dot{\boldsymbol{\rho}} = \frac{1}{2}\boldsymbol{\rho}\{0, \boldsymbol{\omega}_l\}$. On this basis, we introduce $\dot{\boldsymbol{q}} = \Gamma(\boldsymbol{q}, \boldsymbol{v})$ as the linear map that gives the time derivative of the position:

$$\begin{aligned}\Gamma(\boldsymbol{q}, \boldsymbol{v}) = \dot{\boldsymbol{q}} &= \{\dot{\boldsymbol{x}}^{1^T}, \dot{\boldsymbol{\rho}}^{1^T}, \dot{\boldsymbol{x}}^{2^T}, \dot{\boldsymbol{\rho}}^{2^T}, ...\} = \\ &= \{\dot{\boldsymbol{x}}^{1^T}, \frac{1}{2}\boldsymbol{\rho}^1\{0, \boldsymbol{\omega}_l^1\}^T, \dot{\boldsymbol{x}}^{2^T}, \frac{1}{2}\boldsymbol{\rho}^2\{0, \boldsymbol{\omega}_l^2\}^T, ...\}^T.\end{aligned} \tag{1}$$

For the time integration of the system position, different options exist for the function $\boldsymbol{q}^{(t+\Delta t)} = \Lambda(\boldsymbol{q}^{(t)}, \boldsymbol{v}, \Delta t)$, the simplest one being the first-order explicit Euler formula $\boldsymbol{q}^{(t+\Delta t)} = \boldsymbol{q}^{(t)} + \Delta t \dot{\boldsymbol{q}}^{(t)}$. Indeed, for body positions a straightforward first-order differential approximation is used: $\boldsymbol{x}^{(t+\Delta t)} = \boldsymbol{x}^{(t)} + \Delta t \dot{\boldsymbol{x}}^{(t)}$. However, if the same approach is used for quaternions, as in $\boldsymbol{\rho}^{(t+\Delta t)} = \boldsymbol{\rho}^{(t)} + \Delta t \dot{\boldsymbol{\rho}}^{(t)}$, an annoying situation can happen: quaternions may slowly lose the unimodularity and drift away from the S^3 manifold, unless some stabilization is used to enforce $||\boldsymbol{\rho}|| = 1$. Therefore, we prefer to integrate the rotations using the following exponential map that preserves the unimodularity of the quaternions:

$$\boldsymbol{\rho}^{(t+\Delta t)} = \boldsymbol{\rho}^{(t)} e^{\{0, \frac{1}{2}\boldsymbol{\omega}_l \Delta t\}}, \tag{2}$$

so we get

$$\boldsymbol{\rho}^{(t+\Delta t)} = \Lambda(\boldsymbol{q}^{(t)}, \boldsymbol{v}, \Delta t) = \left\{ \begin{array}{l} \boldsymbol{x}^{1,(t)} + \Delta t \dot{\boldsymbol{x}}^{1,(t)} \\ \boldsymbol{\rho}^{1,(t)} e^{\{0, \frac{1}{2}\boldsymbol{\omega}_l^1 \Delta t\}} \\ \boldsymbol{x}^{2,(t)} + \Delta t \dot{\boldsymbol{x}}^{2,(t)} \\ \boldsymbol{\rho}^{2,(t)} e^{\{0, \frac{1}{2}\boldsymbol{\omega}_l^2 \Delta t\}} \\ ... \end{array} \right\}. \tag{3}$$

We can explicitly compute the second factor of the quaternion product thanks to the property $e^{\{0, \boldsymbol{u}\alpha\}} = \{\cos\alpha, \boldsymbol{u}\sin\alpha\}$ of quaternion exponentials; we obtain

$$\begin{aligned} e^{\{0, \frac{1}{2}\boldsymbol{\omega}_l \Delta t\}} &= e^{\{0, (\boldsymbol{\omega}_l/|\boldsymbol{\omega}_l|)\frac{1}{2}|\boldsymbol{\omega}_l|\Delta t\}} = \\ &= \{\cos\frac{1}{2}|\boldsymbol{\omega}_l|\Delta t, \frac{\boldsymbol{\omega}_l}{|\boldsymbol{\omega}_l|}\sin\frac{1}{2}|\boldsymbol{\omega}_l|\Delta t\}. \end{aligned} \tag{4}$$

Since (2) preserves the norm of the quaternions, large Δt time steps can be used (although, once in a while, it is safer to normalize all quaternions because numerical roundoff can accumulate small errors). Yet we can demonstrate that, for $\Delta t \to 0$, the formula (2) still corresponds to $\boldsymbol{\rho}^{(t+\Delta t)} = \boldsymbol{\rho}^{(t)} + \Delta t \dot{\boldsymbol{\rho}}$. In fact,

$$\dot{\boldsymbol{\rho}} = \lim_{\Delta t \to 0} \frac{\boldsymbol{\rho}^{(t+\Delta t)} - \boldsymbol{\rho}^{(t)}}{\Delta t}. \tag{5}$$

Hence, substituting (2) in (5), we can write

$$\dot{\boldsymbol{\rho}} = \lim_{\Delta t \to 0} \boldsymbol{\rho}^{(t)} \frac{\{\cos\frac{1}{2}|\boldsymbol{\omega}_l|\Delta t, \frac{\boldsymbol{\omega}_l}{|\boldsymbol{\omega}_l|}\sin\frac{1}{2}|\boldsymbol{\omega}_l|\Delta t\} - \{1, \boldsymbol{0}\}}{\Delta t}. \tag{6}$$

Applying the Hôpital theorem and simplifying, we obtain $\dot{\boldsymbol{\rho}} = \frac{1}{2}\boldsymbol{\rho}\{0, \boldsymbol{\omega}_l\}$, as expected.

2.2. *Bilateral constraints*

Most kinematic pairs, such as revolute joints, prismatic joints, and glyphs, can be expressed by means of holonomic constraints over the relative position of two bodies. In general, we introduce a set $\mathcal{G}_\mathcal{B}$ of scalar equations

$$\Psi^i(\boldsymbol{q}, t) = 0, \quad i \in \mathcal{G}_\mathcal{B}. \tag{7}$$

[1] Different methods can be used to represent the rotation. If one stores the 3x3 rotation matrix $A \in \mathsf{SO}(3, \mathbb{R})$, each body will require 3+9=12 scalar coordinates, which are highly redundant. On the other hand, storing only three angles (such as the three Euler angles or the three Cardano angles) could give problems of singularities. Not being affected by these limitations, quaternions are better suited for computational application.

The size of the set $\mathcal{G}_\mathcal{B}$ is the number of basic scalar bilateral constraints, not necessarily corresponding to complex three-dimensional mechanical joints [2].

We assume that $\Psi^i(\boldsymbol{q}, t)$ is smooth, so that it can be differentiated to obtain the Jacobian $\nabla_q \Psi^i = \left[\partial \Psi^i / \partial \boldsymbol{q}\right]^T$.

Constraints must be respected also at the velocity level: the full time-derivative of the ith constraint equation is

$$\frac{d\Psi^i(\boldsymbol{q}, t)}{dt} = \frac{\partial \Psi^i}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial \Psi^i}{\partial t} = \nabla_q {\Psi^i}^T \dot{\boldsymbol{q}} + \frac{\partial \Psi^i}{\partial t} = 0$$
$$\frac{d\Psi^i(\boldsymbol{q}, t)}{dt} = \nabla_q {\Psi^i}^T \Gamma(\boldsymbol{q}, \boldsymbol{v}) + \frac{\partial \Psi^i}{\partial t} = 0.$$

For simplicity, from now on we will define $\nabla {\Psi^i}^T = \nabla_q {\Psi^i}^T \Gamma(\boldsymbol{q}, \boldsymbol{v})$.

Note that the term $\frac{\partial \Psi^i}{\partial t}$ is nonzero only for rheonomic (time-dependent) constraints such as motors and imposed trajectories.

For each bilateral constraint, there exists a Lagrange multiplier $\gamma^i_\mathcal{B}$ such that the force acting on the system by the ith bilateral constraint is $\gamma^i_\mathcal{B} \nabla \Psi^i$ [20].

2.3. *Unilateral contact constraints*

Since rigid bodies cannot overlap each other, given the set of body shapes $\boldsymbol{\Omega} = \{\Omega^1, \Omega^2, ..., \Omega^n\}$, we assume that there exists a set of $\mathcal{G}_\mathcal{P}$ distance functions $\Phi(\boldsymbol{q}, \Omega)$ that must satisfy the unilateral constraint conditions:

$$\Phi^i(\boldsymbol{q}, \Omega) \geq 0, \quad i \in \mathcal{G}_\mathcal{P}. \tag{8}$$

An example of such a mapping is the signed distance function [23].

For convenience, we pose the problem in terms of contact points, since in most cases we can compute a minimal set of contact normals belonging to one of two neighboring bodies: distances $\Phi(\boldsymbol{q}, \Omega)$ are measured along those normals.

For example, in the case of spherical bodies with positions $\boldsymbol{x}^j_s$ and radius r_s, for all body pairs j, k the signed distance function is $\Phi = ||\boldsymbol{x}^j_s - \boldsymbol{x}^k_s|| - 2r_s$. Note, however, that, for bodies with generic shapes, finding a proper set of contact points (and defining their Φ^i distance functions) is not always trivial [3,4]. In fact, there could be multiple contact points, or it could even happen that defining a differentiable signed distance function is not possible, as in the case of concave shapes [3].

Nevertheless, since we are interested in enforcing non-penetration, what truly matters is that a signed distance function be defined up to some value of the penetration [4]. We thus assume that $\Phi(\boldsymbol{q}, \Omega)$ can be differentiably defined at least on a neighborhood of the set $\Phi(\boldsymbol{q}, \Omega) \geq 0$. Such an assumption does hold for smooth and strictly convex bodies, such as spheres [3].

[2] In the context of this work, bilateral constraints are always considered scalar, because complex mechanical joints can be modeled by using multiple basic scalar constraints. For example, kinematic pairs such as a ball joint require three scalar equations, a prismatic guide requires five scalar equations, and so on.

In addition, piecewise smooth bodies can be accommodated in a fixed-time-step framework, by decomposing the distance function in components attached to each pair of *features*, such as point and piecewise smooth surface or curved edge and curved edge, and using all normals attached to such pairs [19].

Also, we consider only a subset $\mathcal{G}_\mathcal{A}(\boldsymbol{q}, \Omega, \epsilon) \subset \mathcal{G}_\mathcal{P}$ of all potential contacts, that is, only those contacts whose surfaces are under a distance threshold ϵ:

$$\mathcal{G}_\mathcal{A}(\boldsymbol{q}, \Omega, \epsilon) = \left\{ i \;\middle|\; i \in \mathcal{G}_\mathcal{P}, \;\; \Phi^i(\boldsymbol{q}, \Omega) \leq \epsilon \right\}. \tag{9}$$

Special attention must be paid in implementing an efficient and robust collision algorithm for the generation of the $\mathcal{G}_\mathcal{A}(\boldsymbol{q}, \Omega, \epsilon)$ set of contact points.

A preliminary algorithm, called *broad-phase* collision detection, discards pairs of shapes that are farther than ϵ, in order to avoid a combinatorial waste of time if checking for collision points with all pairs of bodies. We use the *sweep and prune* (SAP) algorithm to this end [17].

The following *narrow-phase* step operates on the pairs of bodies that passed the broad-phase check: it finds the contact points and their normals. We adopt the GJK algorithm for this purpose because it features high efficiency and robustness even in the case of non-smooth surfaces [16].

Concave shapes, if any, undergo an off-line conversion into sets of convex shapes using a convex decomposition algorithm; a *middle-phase* AABB binary-tree traversal is used to check collisions between these compounds of shapes without running into superlinear time complexity.

A thin envelope is added around all shapes using the Minkowski sums in order to allow a small amount of interpenetration. If larger overlapping occurs, the Expanding Polytope (EPA) algorithm is used [11].

2.4. *Frictional constraints*

In the following section we introduce friction by means of conic constraints, which are an extension of complementarity models discussed in [6,40].

2.4.1. *The Coulomb friction model*

The original Coulomb model introduces static μ_s and kinetic μ_k friction coefficients as the only parameters to characterize the frictional phenomena at the surface. Although simple, this model was proven to be realistic and practical in many situations. Usually the kinetic coefficient is slightly lower than the static coefficient, but in this work we consider both to have the same value μ.

If a position $\boldsymbol{q}$ is feasible and the contact is active, that is, $\Phi(\boldsymbol{q}, \Omega) = 0$, then at the contact we have a normal force and a tangential force.

Let $\boldsymbol{n}$ be the normal at the contact pointing from the second body to the first body, and let $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ be the tangents at the contact. Here $\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2$ are mutually orthogonal vectors of length one in three dimensions. The vectors $\boldsymbol{n}$, $\boldsymbol{t}_1$, and $\boldsymbol{t}_2$ are a function of the position $\boldsymbol{q}$, but we ignore this fact until the end of this section.

The reaction force is impressed on the system by means of multipliers $\widehat{\gamma}_n \geq 0$, $\widehat{\gamma}_u$, and $\widehat{\gamma}_v$. The normal component of the force is $\boldsymbol{F}_N = \widehat{\gamma}_n \boldsymbol{n}$, and the tangential component of the force is $\boldsymbol{F}_T = \widehat{\gamma}_u \boldsymbol{t}_1 + \widehat{\gamma}_v \boldsymbol{t}_2$.

The Coulomb model consists of the following constraints:

$$\begin{aligned} \widehat{\gamma}_n &\geq 0, \quad \Phi(\boldsymbol{q}) \geq 0, \quad \Phi(\boldsymbol{q})\widehat{\gamma}_n = 0, \\ \mu\widehat{\gamma}_n &\geq \sqrt{\widehat{\gamma}_u^2 + \widehat{\gamma}_v^2}, \quad ||\mathbf{v}_T|| \left(\mu\widehat{\gamma}_n - \sqrt{\widehat{\gamma}_u^2 + \widehat{\gamma}_v^2}\right) = 0, \\ \langle \boldsymbol{F}_T, \boldsymbol{v}_T \rangle &= -||\boldsymbol{F}_T|| \, ||\mathbf{v}_T||, \end{aligned} \tag{10}$$

where $\mathbf{v}_T$ is the relative tangential velocity at contact. The effect of the friction over the dynamical system is defined by the friction coefficient $\mu \in \mathbb{R}^+$, which typically has a value between 0 and 1 for most materials.

The first part of the constraint can be restated as

$$\boldsymbol{F} = \boldsymbol{F}_N + \boldsymbol{F}_T = \widehat{\gamma}_n \boldsymbol{n} + \widehat{\gamma}_u \boldsymbol{t}_1 + \widehat{\gamma}_v \boldsymbol{t}_2 \in \mathcal{K},$$

where $\mathcal{K}$ is a cone in three dimensions, whose slope is $\arctan(\mu)$.

The constraint $\langle \boldsymbol{F}_T, \mathbf{v}_T \rangle = -||\boldsymbol{F}_T|| \, ||\mathbf{v}_T||$ requires that the tangential force be opposite to the tangential velocity. As a result, the reaction force is dissipative. In fact, an equivalent convenient way of expressing this constraint is by using the maximum dissipation principle [40,38,39],

$$(\widehat{\gamma}_u, \widehat{\gamma}_v) = \operatorname{argmin}_{\sqrt{\widehat{\gamma}_u^2 + \widehat{\gamma}_v^2} \leq \mu\widehat{\gamma}_n} (\widehat{\gamma}_u \boldsymbol{t}_1 + \widehat{\gamma}_v \boldsymbol{t}_2)^T \mathbf{v}_T.$$

These constraints are represented by mapping the vectors $\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2$ from contact coordinates to generalized coordinates [3].

We denote the generalized vector version of $\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2$ by $\boldsymbol{D}_n, \boldsymbol{D}_u, \boldsymbol{D}_v$.

In generalized coordinates, the Coulomb model becomes [8]

$$\boldsymbol{F}_N = \widehat{\gamma}_n \boldsymbol{D}_n, \quad \boldsymbol{F}_T = \widehat{\gamma}_u \boldsymbol{D}_u + \widehat{\gamma}_v \boldsymbol{D}_v, \tag{11}$$

$$\widehat{\gamma}_n \geq 0, \quad \Phi(\boldsymbol{q}) \geq 0, \quad \widehat{\gamma}_n \Phi(\boldsymbol{q}) = 0, \tag{12}$$

where the tangential multipliers $\widehat{\gamma}_u, \widehat{\gamma}_v$ are determined from the maximum dissipation principle

$$(\widehat{\gamma}_u, \widehat{\gamma}_v) = \operatorname{argmin}_{\sqrt{\widehat{\gamma}_u^2 + \widehat{\gamma}_v^2} \leq \mu\widehat{\gamma}_n} (\widehat{\gamma}_u \boldsymbol{D}_u + \widehat{\gamma}_v \boldsymbol{D}_v)^T \boldsymbol{v}.$$

The last relation is obtained from the identities $\boldsymbol{D}_u^T \boldsymbol{v} = \boldsymbol{t}_1^T \mathbf{v}_T$ and $\boldsymbol{D}_v^T \boldsymbol{v} = \boldsymbol{t}_2^T \mathbf{v}_T$.

2.5. *The Overall dynamical model*

The other dynamical data needed for the model are the mass matrix $M(\boldsymbol{q})$, which is symmetric positive definite; the external force $\boldsymbol{f}_e(t, \boldsymbol{q}, \boldsymbol{v})$; and the inertial force $\boldsymbol{f}_c(\boldsymbol{q}, \boldsymbol{v})$, containing the centrifugal and Coriolis forces.

We can define the total force

$$\boldsymbol{f}_t(t, \boldsymbol{q}, \boldsymbol{v}) = \boldsymbol{f}_e(t, \boldsymbol{q}, \boldsymbol{v}) + \boldsymbol{f}_c(\boldsymbol{q}, \boldsymbol{v}). \tag{13}$$

Assume now that we have multiple contact constraints $\Phi^i(\boldsymbol{q}, \Omega) \geq 0$, $i \in \mathcal{G}_\mathcal{A}$ and multiple bilateral constraints $\Psi^i(\boldsymbol{q}, t) = 0$, $i \in \mathcal{G}_\mathcal{B}$. Note that the unilateral condition $\widehat{\gamma}_n \geq 0, \Phi \geq 0, \Phi\widehat{\gamma}_n = 0$ can be written as a complementarity constraint $\widehat{\gamma}_n \geq 0 \perp \Phi \geq 0$.

The continuous model is a differential variational inequality [37]:

$$\begin{aligned} M(q^l)\frac{d\boldsymbol{v}}{dt} &= \sum_{i \in \mathcal{G}_\mathcal{A}} \left(\widehat{\gamma}_n^i \boldsymbol{D}_n^i + \widehat{\gamma}_u^i \boldsymbol{D}_u^i + \widehat{\gamma}_v^i \boldsymbol{D}_v^i\right) + \\ &\quad + \sum_{i \in \mathcal{G}_\mathcal{B}} \widehat{\gamma}_\mathcal{B}^i \nabla\Psi^i + \boldsymbol{f}_t(t, \boldsymbol{q}, \boldsymbol{v}) \\ \dot{\boldsymbol{q}} &= \Gamma(\boldsymbol{q}, \boldsymbol{v}) \\ \Psi^i(\boldsymbol{q}, t) &= 0 \quad i \in \mathcal{G}_\mathcal{B} \\ \widehat{\gamma}_n^i &\geq 0 \quad \perp \quad \Phi^i(\boldsymbol{q}, \Omega) \geq 0, \quad i \in \mathcal{G}_\mathcal{A} \\ \left(\widehat{\gamma}_u^i, \widehat{\gamma}_v^i\right) &= \operatorname{argmin}_{\mu^i\widehat{\gamma}_n^i \geq \sqrt{(\widehat{\gamma}_u^i)^2 + (\widehat{\gamma}_v^i)^2}} \quad i \in \mathcal{G}_\mathcal{A} \\ &\quad \boldsymbol{v}^T \left(\widehat{\gamma}_u \boldsymbol{D}_u^i + \widehat{\gamma}_v \boldsymbol{D}_v^i\right). \end{aligned} \tag{14}$$

Unfortunately, the introduction of the Coulomb friction model may lead to an inconsistent model. It is known [9] that paradoxical configurations exist for which such a model does not have a solution in terms of unknown accelerations and reaction forces. Such configurations are called Painlevé paradoxes [39]. Nevertheless, a weaker formulation of the problem can be solved in terms of vector measures, using a nonsmooth time-stepping scheme where reaction impulses are the unknowns at each time step [39].

To this end we define the following stepping scheme, with time step h, known positions $\boldsymbol{q}^{(l)}$, and velocity $\boldsymbol{v}^{(l)}$; the scheme is an equation problem with equilibrium constraints, where the unknowns are $\boldsymbol{q}^{(l+1)}$, $\boldsymbol{v}^{(l+1)}$, and constraint impulses $\gamma_n = h\widehat{\gamma}_n$, $\gamma_u = h\widehat{\gamma}_u$, $\gamma_v = h\widehat{\gamma}_v$, $\gamma_\mathcal{B} = h\widehat{\gamma}_\mathcal{B}$:

$$\begin{aligned} M^{(l)}(\boldsymbol{v}^{(l+1)} - \boldsymbol{v}^l) &= \sum_{i \in \mathcal{G}_\mathcal{A}} \left(\gamma_n^i \boldsymbol{D}_n^i + \gamma_u^i \boldsymbol{D}_u^i + \gamma_v^i \boldsymbol{D}_v^i\right) + \\ &\quad + \sum_{i \in \mathcal{G}_\mathcal{B}} \left(\gamma_\mathcal{B}^i \nabla\Psi^i\right) + h\boldsymbol{f}_t(t^{(l)}, \boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l)}) \end{aligned} \tag{15}$$

$$0 = \frac{1}{h}\Psi^i(\boldsymbol{q}^{(l)}) + \nabla\Psi^{i^T} \boldsymbol{v}^{(l+1)} + \frac{\partial\Psi^i}{\partial t}, \quad i \in \mathcal{G}_\mathcal{B} \tag{16}$$

$$\begin{aligned} 0 &\leq \frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}) + \nabla\Phi^{i^T} \boldsymbol{v}^{(l+1)} \\ &\quad \perp \quad \gamma_n^i \geq 0, \; i \in \mathcal{G}_\mathcal{A} \end{aligned} \tag{17}$$

$$\begin{aligned} \left(\gamma_u^i, \gamma_v^i\right) &= \operatorname{argmin}_{\mu^i\gamma_n^i \geq \sqrt{(\gamma_u^i)^2 + (\gamma_v^i)^2}} \quad i \in \mathcal{G}_\mathcal{A} \\ &\quad \left[\boldsymbol{v}^T(\gamma_u \boldsymbol{D}_u^i + \gamma_v \boldsymbol{D}_v^i)\right] \end{aligned} \tag{18}$$

$$\boldsymbol{q}^{(l+1)} = \Lambda(\boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l+1)}, h). \tag{19}$$

To simplify notation, we denoted $M(\boldsymbol{q}^l)$ by $M^{(l)}$.

In previous work, we have shown that the scheme is convergent, as the time step h goes to 0, to the solution of a measure differential inclusion [2].

For the special case of zero friction, the subproblem simplifies to (15-17), that is, a linear complementarity problem. Such problems can be solved by Lemke's algorithm [14,6]. Introducing the Coulomb friction (18), however, turns the problem into a nonlinear complementarity problem that poses more difficulties. If the nonlinear constraint cone (the Coulomb cone) is approximated by a piecewise linear cone, the subproblem (15-18) becomes again an LCP solvable by Lemke's algorithm [6]. Nevertheless, in [5] we have also demonstrated that, as the number of constraints in the prob-

lem increases, the computational cost of typical LCP solvers increases far faster than linearly with the size of the problem. Moreover, the approximation of friction cones by means of faceted pyramids would introduce unwanted anisotropy.

To overcome these difficulties, we modified the time-stepping scheme by relaxing the constraint (17) as

$$\begin{aligned} 0 \leq \frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}) + \nabla\Phi^{i^T}\boldsymbol{v}^{(l+1)} \\ -\mu^i\sqrt{(\boldsymbol{D}_u^{i,T}\boldsymbol{v})^2 + (\boldsymbol{D}_v^{i,T}\boldsymbol{v})^2} \perp \quad \gamma_n^i \geq 0,\ i \in \mathcal{G}_\mathcal{A}. \end{aligned} \tag{20}$$

This results in a cone complementarity problem that can be solved with a fixed-point iteration approach, as demonstrated in our earlier work [8].

2.6. *Cone complementarity formulation*

Developing the optimality conditions for the equilibrium constraint in (18), we obtain that there exists a Lagrange multiplier λ^i such that, for any $i \in \mathcal{G}_\mathcal{A}$,

$$\begin{aligned} \lambda^i\gamma_u^i = -\boldsymbol{D}_u^{i,T}\boldsymbol{v},\ \lambda^i\gamma_v^i = -\boldsymbol{D}_v^{i,T}\boldsymbol{v}, \\ \lambda^i \geq 0 \perp \mu^i\gamma_n^i - \sqrt{\left(\gamma_u^i\right)^2 + \left(\gamma_v^i\right)^2} \geq 0. \end{aligned} \tag{21}$$

The first two equations imply that $\lambda^i\sqrt{(\gamma_u^i)^2+(\gamma_v^i)^2} = \sqrt{(\boldsymbol{D}_u^{i,T}\boldsymbol{v})^2+(\boldsymbol{D}_v^{i,T}\boldsymbol{v})^2}$, while the complementarity constraint implies that

$$0 = \lambda^i\sqrt{\left(\gamma_u^i\right)^2 + \left(\gamma_v^i\right)^2}\left(\mu^i\gamma_n^i - \sqrt{\left(\gamma_u^i\right)^2 + \left(\gamma_v^i\right)^2}\right)$$

and, in turn, that

$$\mu^i\gamma_n^i\sqrt{(\boldsymbol{D}_u^{i,T}\boldsymbol{v})^2+(\boldsymbol{D}_v^{i,T}\boldsymbol{v})^2} = \lambda^i\left(\left(\gamma_u^i\right)^2 + \left(\gamma_v^i\right)^2\right). \tag{22}$$

We now define, for all potential contacts, the vectors

$$\boldsymbol{u}_\mathcal{A}^i = \left\{\frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}) + \nabla\Phi^{i^T}\boldsymbol{v}^{(l+1)}, \boldsymbol{D}_u^{i,T}\boldsymbol{v}, \boldsymbol{D}_v^{i,T}\boldsymbol{v}\right\}^T \tag{23}$$

$$\boldsymbol{\gamma}_\mathcal{A}^i = \left\{\gamma_n^i, \gamma_u^i, \gamma_v^i\right\}^T, \qquad i \in \mathcal{G}_\mathcal{A}. \tag{24}$$

We calculate the scalar product using (20),(21):

$$\begin{aligned} \left\langle \boldsymbol{u}_\mathcal{A}^i, \boldsymbol{\gamma}_\mathcal{A}^i \right\rangle &= \gamma_n^i\left(\frac{1}{h}\Phi^i + \nabla\Phi^{i^T}\boldsymbol{v}\right) + \gamma_u^i\boldsymbol{D}_u^{i,T}\boldsymbol{v} \\ &\quad + \gamma_v^i\boldsymbol{D}_v^{i,T}\boldsymbol{v} \\ &= \mu^i\gamma_n^i\sqrt{(\boldsymbol{D}_u^{i,T}\boldsymbol{v})^2+(\boldsymbol{D}_v^{i,T}\boldsymbol{v})^2} \\ &\quad - \lambda^i\left(\left(\gamma_u^i\right)^2 + \left(\gamma_v^i\right)^2\right) \\ &= 0 \quad \Rightarrow\ \boldsymbol{u}_\mathcal{A}^i \perp \boldsymbol{\gamma}_\mathcal{A}^i. \end{aligned} \tag{25}$$

We recall that the *dual cone* of a convex cone $\mathcal{K}$ is the set $\mathcal{K}^* = \{\boldsymbol{x} | \forall \boldsymbol{y} \in \mathcal{K}\ \langle \boldsymbol{y}, \boldsymbol{x}\rangle \geq 0\}$ and that the *polar cone* is defined as $\mathcal{K}^\circ = -\mathcal{K}^*$.

We now define the friction cone $\mathcal{FC}^i$ such that the Coulomb friction model is satisfied if $\boldsymbol{\gamma}_\mathcal{A}^i \in \mathcal{FC}^i$:

$$\mathcal{FC}^i = \left\{x, y, z \in \mathbb{R}^3 | \mu^i x \geq \sqrt{y^2 + z^2}\right\}.$$

Then, from (18), (20), and (25), the frictional contact constraints can be expressed by means of the following *cone complementarity constraints*:

$$-\boldsymbol{u}_\mathcal{A}^i \in \mathcal{FC}^{i\circ} \perp \boldsymbol{\gamma}_\mathcal{A}^i \in \mathcal{FC}^i, \quad i \in \mathcal{G}_\mathcal{A}. \tag{26}$$

To obtain a unified formalism, we can represent also the bilateral constraints (16) in terms of cone complementarity constraints. Of course, multipliers $\gamma_\mathcal{B}^i$, with $i \in \mathcal{G}_\mathcal{B}$, are not restrained into some special subset of $\mathbb{R}$, but even $\mathbb{R}$ itself is a convex cone. Thus we can introduce the scalar

$$u_\mathcal{B}^i = \frac{1}{h}\Psi^i(\boldsymbol{q}^{(l)}) + \nabla\Psi^{i^T}\boldsymbol{v}^{(l+1)} + \frac{\partial\Psi}{\partial t}, \quad i \in \mathcal{G}_\mathcal{B}, \tag{27}$$

which allows us to write the bilateral constraints (16) as

$$-u_\mathcal{B}^i \in \mathcal{BC}^{i\circ} \perp \gamma_\mathcal{B}^i \in \mathcal{BC}^i, \quad i \in \mathcal{G}_\mathcal{B}, \tag{28}$$

where $\mathcal{BC}^i = \{\mathbb{R}\}$ and $\mathcal{BC}^{i\circ} = \{0\}$, and $\langle u_b^i, \gamma_b^i\rangle = 0$ is always satisfied for $-u_\mathcal{B}^i \in \mathcal{BC}^{i\circ}$.

We now define the vector

$$\tilde{\boldsymbol{k}}^{(l)} = M^{(l)}\boldsymbol{v}^{(l)} + h\boldsymbol{f}_t(t^{(l)}, \boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l)}). \tag{29}$$

Then, equations (29), (28), and (26), together with (15) and the definition of $\boldsymbol{u}_\mathcal{A}^i$, $\boldsymbol{\gamma}_\mathcal{A}^i$, $u_\mathcal{B}^i$, and $\gamma_\mathcal{B}^i$, result in the following problem:

$$\begin{aligned} M^{(l)}v^{(l+1)} &= \sum_{i\in\mathcal{G}_\mathcal{A}}\left(\gamma_n^i\boldsymbol{D}_n^i + \gamma_u^i\boldsymbol{D}_u^i + \gamma_v^i\boldsymbol{D}_v^i\right) + \\ &\quad + \sum_{i\in\mathcal{G}_\mathcal{B}}\left(\gamma_\mathcal{B}^i\nabla\Psi^i\right) + \tilde{\boldsymbol{k}}^{(l)}, \\ &\quad - \boldsymbol{u}_\mathcal{A}^i \in \mathcal{FC}^{i\circ} \perp \boldsymbol{\gamma}_\mathcal{A}^i \in \mathcal{FC}^i, \quad i \in \mathcal{G}_\mathcal{A} \\ &\quad - u_\mathcal{B}^i \in \mathcal{BC}^{i\circ} \perp \gamma_\mathcal{B}^i \in \mathcal{BC}^i, \quad i \in \mathcal{G}_\mathcal{B}. \end{aligned} \tag{30}$$

If we want to obtain a cone complementarity problem as expressed in the typical form $-\mathcal{K}^\circ \ni \boldsymbol{f}(\boldsymbol{a}) \perp \boldsymbol{a} \in \mathcal{K}$, a more compact formulation of the problem (30) is necessary. To this end we denote by $n_\mathcal{A}$ and $n_\mathcal{B}$ the number of elements in the sets $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$, respectively. Then, we define the following vectors $\boldsymbol{b}_\mathcal{A} \in \mathbb{R}^{3n_\mathcal{A}}$, $\boldsymbol{\gamma}_\mathcal{A} \in \mathbb{R}^{3n_\mathcal{A}}$, $\boldsymbol{b}_\mathcal{B} \in \mathbb{R}^{n_\mathcal{B}}$, and $\boldsymbol{\gamma}_\mathcal{B} \in \mathbb{R}^{n_\mathcal{B}}$:

$$\begin{aligned} \boldsymbol{b}_\mathcal{A} &= \left\{\frac{1}{h}\Phi^{i_1}, 0, 0, \frac{1}{h}\Phi^{i_2}, 0, 0, \ldots, \frac{1}{h}\Phi^{i_{n_\mathcal{A}}}, 0, 0\right\}^T \\ \boldsymbol{\gamma}_\mathcal{A} &= \left\{\gamma_n^{i_1}, \gamma_u^{i_1}, \gamma_v^{i_1}, \gamma_n^{i_2}, \gamma_u^{i_2}, \gamma_v^{i_2}, \ldots, \gamma_n^{i_{n_\mathcal{A}}}, \gamma_u^{i_{n_\mathcal{A}}}, \gamma_v^{i_{n_\mathcal{A}}}\right\}^T \\ \boldsymbol{b}_\mathcal{B} &= \left\{\frac{1}{h}\Psi^1 + \frac{\partial\Psi^1}{\partial t}, \frac{1}{h}\Psi^2 + \frac{\partial\Psi^2}{\partial t}, \ldots, \frac{1}{h}\Psi^{n_\mathcal{B}} + \frac{\partial\Psi^{n_\mathcal{B}}}{\partial t}\right\}^T \\ \boldsymbol{\gamma}_\mathcal{B} &= \left\{\gamma_\mathcal{B}^1, \gamma_\mathcal{B}^2, \ldots, \gamma_\mathcal{B}^{n_\mathcal{B}}\right\}^T \\ \boldsymbol{u}_\mathcal{A} &= \left\{\boldsymbol{u}_\mathcal{A}^{1^T}, \boldsymbol{u}_\mathcal{A}^{2^T}, \ldots, \boldsymbol{u}_\mathcal{A}^{n_\mathcal{A}^T}\right\}^T, \boldsymbol{u}_\mathcal{B} = \left\{u_\mathcal{B}^1, u_\mathcal{B}^2, \ldots, u_\mathcal{B}^{n_\mathcal{B}}\right\}^T. \end{aligned} \tag{31}$$

It is useful to merge these vectors, joining data from both frictional constraints and bilateral constraints, obtaining vectors with $n_\mathcal{E} = 3n_\mathcal{A} + n_\mathcal{B}$ scalar elements:

$$\boldsymbol{b}_\mathcal{E} = \left\{\boldsymbol{b}_\mathcal{A}^T, \boldsymbol{b}_\mathcal{B}^T\right\}^T,\ \boldsymbol{\gamma}_\mathcal{E} = \left\{\boldsymbol{\gamma}_\mathcal{A}^T, \boldsymbol{\gamma}_\mathcal{B}^T\right\}^T,\ \boldsymbol{u}_\mathcal{E} = \left\{\boldsymbol{u}_\mathcal{A}^T, \boldsymbol{u}_\mathcal{B}^T\right\}^T. \tag{32}$$

For each frictional contact $i \in \mathcal{G}_\mathcal{A}$ we also define the following three-column matrix:

$$D^i = \left[\boldsymbol{D}_n^i | \boldsymbol{D}_u^i | \boldsymbol{D}_v^i\right]. \tag{33}$$

As before, it is useful to merge all Jacobians from both frictional constraints and bilateral constraints in a single, large matrix having $n_\mathcal{E}$ columns and m_v rows:

$$D_\mathcal{E} = \left[D^{i_1} | D^{i_2} | \ldots | D^{i_{n_\mathcal{A}}} | \nabla\Psi^1 | \nabla\Psi^2 | \ldots | \nabla\Psi^{n_\mathcal{B}}\right]. \tag{34}$$

From the definitions (23),(27), (31), (32), (33), and (34) one can see that

$$\boldsymbol{u}_{\mathcal{E}} = D_{\mathcal{E}}^T \boldsymbol{v}^{(l+1)} + \boldsymbol{b}_{\mathcal{E}}. \tag{35}$$

Also, premultiplying by ${M^{(l)}}^{-1}$ equation (15), one gets

$$\boldsymbol{v}^{(l+1)} = {M^{(l)}}^{-1} D_{\mathcal{E}} \boldsymbol{\gamma}_{\mathcal{E}} + {M^{(l)}}^{-1} \tilde{\boldsymbol{k}}. \tag{36}$$

Hence it is possible to substitute (36) into (35) to obtain

$$\boldsymbol{u}_{\mathcal{E}} = D_{\mathcal{E}}^T {M^{(l)}}^{-1} D_{\mathcal{E}} \boldsymbol{\gamma}_{\mathcal{E}} + D_{\mathcal{E}}^T {M^{(l)}}^{-1} \tilde{\boldsymbol{k}} + \boldsymbol{b}_{\mathcal{E}}. \tag{37}$$

To make the expressions more compact, we introduce the following:

$$N = D_{\mathcal{E}}^T {M^{(l)}}^{-1} D_{\mathcal{E}} \tag{38}$$

$$\boldsymbol{r} = D_{\mathcal{E}}^T {M^{(l)}}^{-1} \tilde{\boldsymbol{k}} + \boldsymbol{b}_{\mathcal{E}} \tag{39}$$

In this way, we can write

$$\boldsymbol{u}_{\mathcal{E}} = N \boldsymbol{\gamma}_{\mathcal{E}} + \boldsymbol{r}. \tag{40}$$

Consider the multidimensional cone obtained by performing the direct sum of all $\mathcal{FC}$ and $\mathcal{BC}$ cones and its embedding in the corresponding vector space direct sums:

$$\Upsilon = \left(\bigoplus_{i \in \mathcal{G}_{\mathcal{A}}} \mathcal{FC}^i \right) \bigoplus \left(\bigoplus_{i \in \mathcal{G}_{\mathcal{B}}} \mathcal{BC}^i \right). \tag{41}$$

From (41), (40), (31), the complementarity relationship in (30), and the property of convex cones [22], $\Upsilon = \bigoplus_i \Upsilon^i \Rightarrow \Upsilon^\circ = \bigoplus_i \Upsilon^{i,\circ}$, we can write the problem as a cone complementarity problem (CCP):

$$(N \boldsymbol{\gamma}_{\mathcal{E}} + \boldsymbol{r}) \in -\Upsilon^\circ \quad \perp \quad \boldsymbol{\gamma}_{\mathcal{E}} \in \Upsilon. \tag{42}$$

The separable structure of the cone will allow us to define an algorithm based on block matrices, with relatively small blocks (dimension no larger than 3 in the case of the contact problem).

2.7. *Physical effects of the relaxation*

In [2] we demonstrated that, for $h \to 0$, the solution of the time-stepping scheme with the relaxed constraints (20) will approach the solution of the same measure differential inclusion as the scheme that uses the unrelaxed constraints (17). In addition, iterates produced by the modified scheme approach the ones of the original scheme *even for one time step* at fixed h, provided that $\mu^i \gamma_n^i \sqrt{(\boldsymbol{D}_u^{i,T} \boldsymbol{v})^2 + (\boldsymbol{D}_v^{i,T} \boldsymbol{v})^2} << 1$, that is, with either low friction or low tangential speed [5]. We note that this regime happens frequently in granular flow applications, such as in the simulation of the refueling of pebble bed nuclear reactors [34] and dense packing of granular matter. On the other hand, the good behavior of our scheme occurs even beyond this regime, for reasons we now explain.

Note that the Φ/h term achieves constraint stabilization. When the term is positive and the constraint is active, it biases the normal impulse to be smaller than the one at exact contact and allows for "soft landing" projection onto the contact manifold. When the term is negative, it biases the normal impulse to be larger than the one at exact contact and allows restoration to the contact manifold. The square root term in (20), which can be written also as $\mu ||\mathbf{v}_T^i||$, does not appear in the original model and, as discussed in [2], does result in a larger value of the normal velocity. Therefore, it could potentially produce a departure from the prediction of the original model. As we discuss below, however, the constraint stabilization term substantially alleviates this effect. Indeed, the normal speed is $\mathbf{v}_T^i = \nabla \Phi^{i^T} \boldsymbol{v}$, so it follows that

$$\Phi^i(\boldsymbol{q}) = \mu^i ||\mathbf{v}_T^i|| h, \quad \mathbf{v}_N^i = 0, \tag{43}$$

satisfies the complementarity constraints (20) exactly. Hence, a solution to (20) that involves the term $\mu^i ||\mathbf{v}_T^i||$ need not result in an increase value of the normal velocity as long as the gap $\Phi^i(\boldsymbol{q})$ is about $\mu^i ||\mathbf{v}_T^i|| h$. All the meaningful simulations that we have carried out indicate that this is indeed what happens, and thus the solution approaches the solution of the original scheme insofar as normal velocity. Hence, the model effectively includes a separating boundary layer of size $\mu^i ||\mathbf{v}_T^i|| h$ that plays the role of zero-order effective compliance. Smaller values of $\Phi^i(\boldsymbol{q})$ result in higher than steady-state impulse, whereas larger values of $\Phi^i(\boldsymbol{q})$ result in smaller than steady-state impulse, both of which push the value of the gap function to the value $\mu^i \mathbf{v}_T^i$. This effect is immediately seen for one contact, which leaves from rest with zero tangential velocity or which lands in an inelastic fashion. This is harder to prove for multiple contacts, though it always held in our experiments. There is one caveat, however, as shown in [2]. That is, if one starts with $\Phi^i(\boldsymbol{q}) << \mu^i \mathbf{v}_T^i h$, then the normal component of the velocity is much larger than the one predicted by the original scheme and then what is needed to satisfy approximately (43) in a few steps, so the contact suffers an artificial takeoff. This corresponds to the only case in which we have seen our argument fail, the one of high initial tangential velocity at a contact. Since this case occurs rarely in practice, our relaxation approximates in most cases the solution that would have been obtained by the original, unrelaxed scheme.

3. The iterative method

To solve the CCP (42), we propose a fixed-point iteration with the following form:

$$\begin{aligned} \boldsymbol{\gamma}_{\mathcal{E}}^{r+1} = {} & \lambda \Pi_\Upsilon \left(\boldsymbol{\gamma}_{\mathcal{E}}^r - \omega B^r \left(N \boldsymbol{\gamma}_{\mathcal{E}}^r + \boldsymbol{r} + K^r \left(\boldsymbol{\gamma}_{\mathcal{E}}^{r+1} - \boldsymbol{\gamma}_{\mathcal{E}}^r \right) \right) \right) + \\ & + (1 - \lambda) \boldsymbol{\gamma}_{\mathcal{E}}^r, \\ & r = 0, 1, 2, \ldots. \end{aligned} \tag{44}$$

This iteration uses block matrices B^r and K^r. Matrix B is null except for blocks on the diagonal; in our implementation blocks that correspond to the ith frictional contact are scaled identity matrices $B_{\mathcal{A}}^i = \eta_{\mathcal{A}}^i I$, $B_{\mathcal{A}}^i \in \mathbb{R}^3$, while blocks that correspond to the ith bilateral constraint are scalars $B_{\mathcal{B}}^i = \eta_{\mathcal{B}}^i$, $B_{\mathcal{B}}^i \in \mathbb{R}$. Matrix K is an upper or lower block matrix, with null blocks on the diagonal corresponding to the blocks of B; in

both cases the method is similar to a projected block Gauss-Seidel. Another option is to use a null K matrix, so that the method is similar to a projected block Gauss-Jacobi.

Also, $\Pi_\Upsilon : \mathbb{R}^{n_\varepsilon} \mapsto \mathbb{R}^{n_\varepsilon}$ is the orthogonal projection operator, whose calculation, as we show in the subsequent sections, is considerably simplified by the separable cone structure (41).

For convergence of the scheme we need the following assumptions about the algorithm.

A1 The matrix N of the CCP problem is symmetric and positive semi-definite.

A2 There exists a positive number, $\alpha > 0$ such that, at any iteration r, $\quad r = 0, 1, 2, \ldots$, we have that $B^r \succ \alpha I$.

A3 There exists a positive number $\beta > 0$ such that, at any iteration r, $\quad r = 0, 1, 2, \ldots$, we have that $(\boldsymbol{\gamma}^{r+1} - \boldsymbol{\gamma}^r)^T \left((\lambda\omega B^r)^{-1} + K^r - \frac{N}{2} \right) (\boldsymbol{\gamma}^{r+1} - \boldsymbol{\gamma}^r) \geq \beta \left\| \boldsymbol{\gamma}^{r+1} - \boldsymbol{\gamma}^r \right\|^2$.

Note that the first assumption is always assured in multibody systems because the mass matrix is definite positive, B^r and α are under our control for satisfying the second assumption, and it is always possible to adjust the free ω and λ parameters in order to satisfy the third assumption. Our main convergence result [8, Corollary 1] is given in Theorem 1.

Theorem 1 *Assume that*

$$\mathbf{0} \neq \boldsymbol{\gamma}_\mathcal{E} \in \Upsilon \Rightarrow N\boldsymbol{\gamma}_\mathcal{E} \neq \mathbf{0}$$

(that is, there does not exist a choice of reaction forces whose net effect is zero; bodies do not get "stuck"). Then the algorithm (44) for the cone complementarity problem applied to (42) produces a bounded sequence, and any accumulation point results in the same velocity solution.

We note that, even if in [8] we have demonstrated the use of our algorithm for contacts only, Theorem 1 applies to the bilateral case as well. Nevertheless, the key observation is that we consider the bilateral constraints as linearized constraints for which we develop the additional algorithmic machinery in this paper. The direct application of the formulation in [8], by formulating a bilateral constraint as two unilateral constraints will violate the main assumption in Theorem 1. Therefore, we had to show that direct treatment of bilateral constraints will result in the same formal abstraction (42).

3.1. *Efficient computation of the projection onto the friction cone*

Because of the separable cone structure of the convex subspace Υ Π_Υ, the metric projection $\mathbb{R}^{n_\varepsilon} \rightarrow \mathbb{R}^{n_\varepsilon}$ is

$$\begin{aligned}\Pi_\Upsilon \;=\; \big\{ &\Pi_{\mathcal{FC}^1}(\boldsymbol{\gamma}_\mathcal{A}^1)^T, \Pi_{\mathcal{FC}^2}(\boldsymbol{\gamma}_\mathcal{A}^2)^T, \ldots, \Pi_{\mathcal{FC}^{n_\mathcal{A}}}(\boldsymbol{\gamma}_\mathcal{A}^{n_\mathcal{A}})^T, \\ &\Pi_{\mathcal{BC}^1}(\boldsymbol{\gamma}_\mathcal{B}^2)^T, \Pi_{\mathcal{BC}^2}(\boldsymbol{\gamma}_\mathcal{B}^2)^T, \ldots \Pi_{\mathcal{BC}^{n_\mathcal{B}}}(\boldsymbol{\gamma}_\mathcal{B}^{n_\mathcal{B}})^T \big\}^T,\end{aligned}$$

where $\Pi_{\mathcal{FC}^i}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ with $i \in \mathcal{G}_\mathcal{A}$, and $\Pi_{\mathcal{BC}^i}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ with $i \in \mathcal{G}_\mathcal{B}$. The projection operator must behave as $\Pi_\Upsilon(\boldsymbol{\gamma}) = \text{argmin}_{\zeta \in \Upsilon} ||\boldsymbol{\gamma} - \zeta||$ in order to be

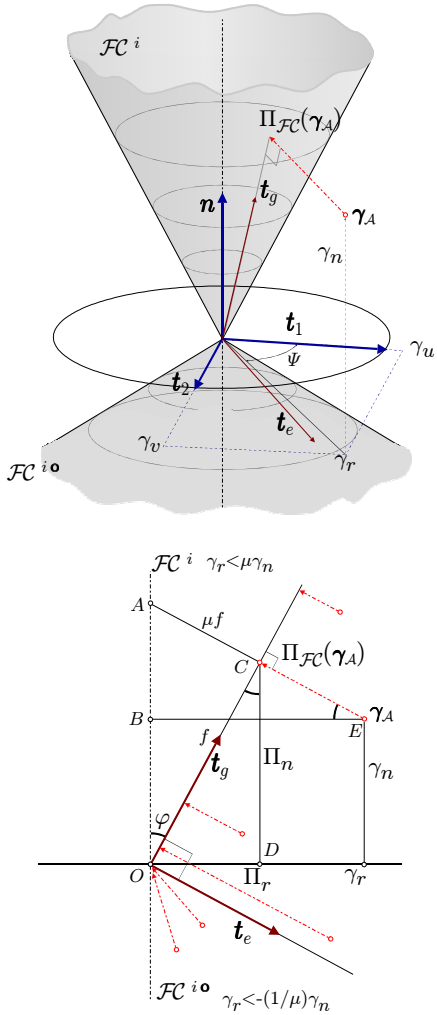

Fig. 2. A single friction cone and its dual cone, with examples of projections.

globally nonexpansive for projection on convex subspace Υ^i.

For the multipliers introduced by friction constraints, it is enough to introduce a straightforward mapping $\Pi_{\mathcal{FC}^i}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ to be applied multiple times on all the triplets of contact multipliers $\boldsymbol{\gamma}_\mathcal{A}^i, i \in \mathcal{G}_\mathcal{A}$. In Fig. 2 three subcases are detected:

- when $\boldsymbol{\gamma}_\mathcal{A}^i$ is inside $\mathcal{FC}^i$, the vector is left untouched;
- when $\boldsymbol{\gamma}_\mathcal{A}^i$ is inside the $\mathcal{FC}^{i^\circ}$ polar cone, it maps to the origin $\{0, 0, 0\}$;
- when $\boldsymbol{\gamma}_\mathcal{A}^i$ is projected to the nearest point on the $\mathcal{FC}^i$ cone.

Introducing $\gamma_r = \sqrt{\gamma_u^2 + \gamma_v^2}$, applying Pythagoras's theorem on triangle OCA, and using similarity between triangles OCA and OCD, we easily get

$$\Pi_n = \frac{\gamma_r \mu + \gamma_n}{\mu^2 + 1} \quad , \quad \Pi_r = \mu \Pi_n \quad , \quad \Pi_u = \gamma_u \frac{\Pi_r}{\gamma_r} \quad , \quad \Pi_v = \gamma_v \frac{\Pi_r}{\gamma_r}. \tag{45}$$

One can verify that, $\forall i \in \mathcal{G}_\mathcal{A}$, such a mapping exhibits $||\Pi_{\mathcal{FC}^i}(\boldsymbol{a}) - \Pi_{\mathcal{FC}^i}(\boldsymbol{b})|| \leq ||\boldsymbol{a} - \boldsymbol{b}||$; hence the nonextensibility property holds.

Including also the case of bilateral constraints $\mathcal{B}$, the projection operator becomes

$$\Pi_\Upsilon \quad \begin{cases} \forall i \in \mathcal{G}_\mathcal{B} & \Pi_{\mathcal{B}\mathcal{C}^i} = \gamma^i_\mathcal{B} \\ \forall i \in \mathcal{G}_\mathcal{A} & \\ \quad \gamma_r < \mu_i \gamma_n & \Pi_{\mathcal{F}\mathcal{C}^i} = \boldsymbol{\gamma}^i_\mathcal{A} \\ \quad \gamma_r < -\dfrac{1}{\mu_i}\gamma_n & \Pi_{\mathcal{F}\mathcal{C}^i} = \{0,0,0\}^T \\ \quad \gamma_r > \mu_i\gamma_n, \gamma_r > -\dfrac{1}{\mu_i}\gamma_n & \Pi_{\mathcal{F}\mathcal{C}^i_n} = \dfrac{\gamma_r\mu_i + \gamma_n}{\mu_i^2+1} \\ & \Pi_{\mathcal{F}\mathcal{C}^i_u} = \gamma_u \dfrac{\mu_i \Pi_{\mathcal{F}\mathcal{C}^i_n}}{\gamma_r} \\ & \Pi_{\mathcal{F}\mathcal{C}^i_v} = \gamma_v \dfrac{\mu_i \Pi_{\mathcal{F}\mathcal{C}^i_n}}{\gamma_r}. \end{cases} \tag{46}$$

4. Implementation

The CCP method proposed here can be applied to the simulation of multibody systems with a large number of parts and contacts because, where an upper limit on the number of iteration is enforced, the iteration (44) can run in $O(n)$ space and $O(n)$ time.

Previous sections showed that generic multibody problems with frictional contacts, expressed with the system (15)–(19), embed the cone complementarity problem (42), which can be solved by the iterative method (44).

Given (31), one can consider the final time-stepping scheme as a sequence of three main operations: a CCP problem that finds unknown reactions $\boldsymbol{\gamma}_\mathcal{E}$ (47a), an affine scaling (47b) that gives the new speeds $\boldsymbol{v}^{(l+1)}$, and a position update (47c):

$$(N\boldsymbol{\gamma}_\mathcal{E} + \boldsymbol{r}) \in -\Upsilon^o \quad \perp \quad \boldsymbol{\gamma}_\mathcal{E} \in \Upsilon \tag{47a}$$

$$\boldsymbol{v}^{(l+1)} = M^{-1}\left(\tilde{\boldsymbol{k}} + D_\mathcal{E}\boldsymbol{\gamma}_\mathcal{E}\right) \tag{47b}$$

$$\boldsymbol{q}^{(l+1)} = \Lambda(\boldsymbol{q}^{(l)}, \boldsymbol{v}^{(l+1)}, h). \tag{47c}$$

The biggest computational overhead is caused by the first problem, that is, the CCP (47a). In fact, (47c) is immediate, and (47b) can be computed quickly because in most cases the matrix M is diagonal and its inverse M^{-1} can be precomputed easily.

The convergence theory about the iterative scheme (44) leaves some degrees of freedom in choosing $\eta^i_\mathcal{A}, \eta^i_\mathcal{B}$ values that build the diagonal blocks of the iteration matrix B. A trivial choice could be to use the same $\eta^i_\mathcal{A} = \eta^i_\mathcal{B} = \xi$ value for all diagonal blocks, that is, $B = \xi I$, and then use the overrelaxation parameter ω to control the convergence. However, this may slow convergence in systems with large mass ratios, even with an optimal ω. A more practical approach, which copes better with systems affected by uneven masses, is inspired by the Gauss-Jacobi idea of using the inverse of the diagonal of the system matrix N, so we use $\eta^i_\mathcal{B} = \frac{1}{\nabla\Psi^{i,T} M^{-1} \nabla\Psi^i}$ and $\eta^i_\mathcal{A} = \frac{1}{\bar{g}_i}$, where $\bar{g}_i$ is the average of the diagonal values of the ith block of the N matrix. We note that $\bar{g}_i$ can be computed easily from the trace of the 3×3 matrix $D^{i,T} M^{-1} D^i$, as

$$\bar{g}_i = \frac{\text{Trace}(D^{i,T} M^{-1} D^i)}{3}. \tag{48}$$

We recall that the matrix N is a product of large matrices; $N = D_\mathcal{E}^T M^{-1} D_\mathcal{E}$, and it is full even if D and M are sparse. For systems with a large number of contacts, the size of N would be prohibitive and clearly would not satisfy the goal of $O(n)$ space complexity. To this end, direct multiplication of vectors and matrices in (44) must be avoided; otherwise the effort and the space requirement would be superlinear in the number of constraint.

For the reasons above, a scheme that does not need the explicit building of N, B, and K has been developed, exploiting the sparsity of M and D.

The K matrix in (44) can be chosen freely, within the convergence limits posed by assumptions [A1]–[A3]. Among the most noticeable options, we have the case $K = 0$, which results in a scheme like a projected Gauss-Jacobi, or the case where K is built by using the lower blocks of N.

4.1. *Optimized projected Gauss-Jacobi CCP*

Considering the case $K = 0$, and recalling Eq. (38), we can express the rth step of the iteration (44) as an inner loop with index $i = 1 \ldots n_\mathcal{A}$ on all $n_\mathcal{A}$ friction cones $\mathcal{F}\mathcal{C}^i$:

$$\boldsymbol{\delta}^{i,r+1}_\mathcal{A} = \boldsymbol{\gamma}^{i,r}_\mathcal{A} - \omega\eta^i_\mathcal{A}\left(D^{i,T} M^{-1}\left(\sum_{z=1}^{n_\mathcal{A}} D^z \boldsymbol{\gamma}^{z,r}_\mathcal{A} + \sum_{z=1}^{n_\mathcal{B}} \nabla\Psi^z \gamma^{z,r}_\mathcal{B} + \tilde{\boldsymbol{k}}\right) + \boldsymbol{b}^i_\mathcal{A}\right) \tag{49}$$

$$\boldsymbol{\gamma}^{i,r+1}_\mathcal{A} = \lambda\Pi_{\mathcal{F}\mathcal{C}^i}\left(\boldsymbol{\delta}^{i,r+1}_\mathcal{A}\right) + (1-\lambda)\boldsymbol{\gamma}^{i,r}_\mathcal{A}, \tag{50}$$

followed by an inner loop with index $i = 1 \ldots n_\mathcal{B}$ on all $n_\mathcal{B}$ bilateral constraints (which we do not report because it is like (49)-(50) except for $_\mathcal{B}$ instead of $_\mathcal{A}$ subscripts).

However, for each iteration, the previous loop $i = 1 \ldots n_\mathcal{A}$ would require quadratic time in terms of potential contacts $n_\mathcal{A}$ because of the presence of the summations $\sum_{z=1}^{n_\mathcal{A}}$. This major source of slow performance can be eliminated if one computes the algorithm in incremental form. In fact, from (36) it follows that

$$\boldsymbol{v}^r = M^{-1}\left(\sum_{z=1}^{n_\mathcal{A}} D^z \boldsymbol{\gamma}^{z,r}_\mathcal{A} + \sum_{z=1}^{n_\mathcal{B}} \nabla\Psi^z \gamma^{z,r}_\mathcal{B} + \tilde{\boldsymbol{k}}\right). \tag{51}$$

Substituting (51) into (49), we can write

$$\boldsymbol{\delta}^{i,r+1}_\mathcal{A} = \boldsymbol{\gamma}^{i,r}_\mathcal{A} - \omega\eta^i_\mathcal{A}\left(D^{i,T}\boldsymbol{v}^r + \boldsymbol{b}^i_\mathcal{A}.\right) \tag{52}$$

Considering the optimizations above, we can express the final CCP algorithm with the pseudocode of Algorithm 1.

In the proposed algorithm, for achieving high performance, some auxiliary data can be precomputed before starting the iteration. Specifically, we introduce the $m_v \times 3$ matrix $E^i_\mathcal{A} = M^{-1} D^i$ and the vector $\boldsymbol{E}^i_\mathcal{B} = M^{-1}\nabla\Psi^i$.

The iterations, usually stopped when an approximation threshold has been reached, can be also prematurely aborted when r exceeds a limit r_{max} on the

Algorithm 1: Solve complementarity - PGJ CCP

(1) *// Pre-compute some data for friction constraints*
(2) **for** $i := 1$ **to** $n_{\mathcal{A}}$
(3) $\quad E^i_{\mathcal{A}} = M^{-1}D^i$
(4) $\quad \eta^i_{\mathcal{A}} = \frac{3}{\mathrm{Trace}(D^{i,T}E^i_{\mathcal{A}})}$
(5) *// Pre-compute some data for bilateral constraints*
(6) **for** $i := 1$ **to** $n_{\mathcal{B}}$
(7) $\quad \boldsymbol{E}^i_{\mathcal{B}} = M^{-1}\boldsymbol{\nabla\Psi}^i$
(8) $\quad \eta^i_{\mathcal{B}} = \frac{1}{\boldsymbol{\nabla\Psi}^{i,T}\boldsymbol{E}^i_{\mathcal{B}}}$
(9)
(10) *// Initialize impulses*
(11) **if** warm start with initial guess $\boldsymbol{\gamma}^*_{\mathcal{E}}$
(12) $\quad \boldsymbol{\gamma}^0_{\mathcal{E}} = \boldsymbol{\gamma}^*_{\mathcal{E}}$
(13) **else**
(14) $\quad \boldsymbol{\gamma}^0_{\mathcal{E}} = \mathbf{0}$
(15)
(16) *// Initialize speeds*
(17) $\boldsymbol{v}^0 = \sum_{i=1}^{n_{\mathcal{A}}} E^i_{\mathcal{A}}\boldsymbol{\gamma}^{i,0}_{\mathcal{A}} + \sum_{i=1}^{n_{\mathcal{B}}} \boldsymbol{E}^i_{\mathcal{B}}\gamma^{i,0}_{\mathcal{B}} + M^{-1}\tilde{\boldsymbol{k}}$
(18)
(19) *// Main iteration loop*
(20) **for** $r := 0$ **to** r_{max}
(21) $\quad$ *// Loop on frictional constraints*
(22) $\quad$ **for** $i := 1$ **to** $n_{\mathcal{A}}$
(23) $\quad\quad \boldsymbol{\delta}^{i,r+1}_{\mathcal{A}} = \left(\boldsymbol{\gamma}^{i,r}_{\mathcal{A}} - \omega\eta^i_{\mathcal{A}}\left(D^{i,T}\boldsymbol{v}^r + \boldsymbol{b}^i_{\mathcal{A}}\right)\right)$;
(24) $\quad\quad \boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}} = \lambda\Pi_{\Upsilon}\left(\boldsymbol{\delta}^{i,r+1}_{\mathcal{A}}\right) + (1-\lambda)\boldsymbol{\gamma}^{i,r}_{\mathcal{A}}$;
(25) $\quad$ *// Loop on bilateral constraints*
(26) $\quad$ **for** $i := 1$ **to** $n_{\mathcal{B}}$
(27) $\quad\quad \delta^{i,r+1}_{\mathcal{B}} = \left(\gamma^{i,r}_{\mathcal{B}} - \omega\eta^i_{\mathcal{B}}\left(\nabla\boldsymbol{\Psi}^{i,T}\boldsymbol{v}^r + b^i_{\mathcal{B}}\right)\right)$;
(28) $\quad\quad \gamma^{i,r+1}_{\mathcal{B}} = \lambda\Pi_{\Upsilon}\left(\delta^{i,r+1}_{\mathcal{B}}\right) + (1-\lambda)\gamma^{i,r}_{\mathcal{B}}$;
(29) $\quad$ *// Update speeds*
(30) $\quad \boldsymbol{v}^{r+1} = \sum_{i=1}^{n_{\mathcal{A}}} E^i_{\mathcal{A}}\boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}} + \sum_{i=1}^{n_{\mathcal{B}}} \boldsymbol{E}^i_{\mathcal{B}}\gamma^{i,r+1}_{\mathcal{B}} + M^{-1}\tilde{\boldsymbol{k}}$
(31)
(32) **return** $\boldsymbol{\gamma}_{\mathcal{E}}$, $\boldsymbol{v}$

Algorithm 2: Solve complementarity - PGS CCP

(1) *// Pre-compute some data for friction constraints*
(2) **for** $i := 1$ **to** $n_{\mathcal{A}}$
(3) $\quad E^i_{\mathcal{A}} = M^{-1}D^i$
(4) $\quad \eta^i_{\mathcal{A}} = \frac{3}{\mathrm{Trace}(D^{i,T}E^i_{\mathcal{A}})}$
(5) *// Pre-compute some data for bilateral constraints*
(6) **for** $i := 1$ **to** $n_{\mathcal{B}}$
(7) $\quad \boldsymbol{E}^i_{\mathcal{B}} = M^{-1}\boldsymbol{\nabla\Psi}^i$
(8) $\quad \eta^i_{\mathcal{B}} = \frac{1}{\boldsymbol{\nabla\Psi}^{i,T}\boldsymbol{E}^i_{\mathcal{B}}}$
(9)
(10) *// Initialize impulses*
(11) **if** warm start with initial guess $\boldsymbol{\gamma}^*_{\mathcal{E}}$
(12) $\quad \boldsymbol{\gamma}^0_{\mathcal{E}} = \boldsymbol{\gamma}^*_{\mathcal{E}}$
(13) **else**
(14) $\quad \boldsymbol{\gamma}^0_{\mathcal{E}} = \mathbf{0}$
(15)
(16) *// Initialize speeds*
(17) $\boldsymbol{v} = \sum_{i=1}^{n_{\mathcal{A}}} E^i_{\mathcal{A}}\boldsymbol{\gamma}^{i,0}_{\mathcal{A}} + \sum_{i=1}^{n_{\mathcal{B}}} \boldsymbol{E}^i_{\mathcal{B}}\gamma^{i,0}_{\mathcal{B}} + M^{-1}\tilde{\boldsymbol{k}}$
(18)
(19) *// Main iteration loop*
(20) **for** $r := 0$ **to** r_{max}
(21) $\quad$ *// Loop on frictional constraints*
(22) $\quad$ **for** $i := 1$ **to** $n_{\mathcal{A}}$
(23) $\quad\quad \boldsymbol{\delta}^{i,r+1}_{\mathcal{A}} = \left(\boldsymbol{\gamma}^{i,r}_{\mathcal{A}} - \omega\eta^i_{\mathcal{A}}\left(D^{i,T}\boldsymbol{v}^r + \boldsymbol{b}^i_{\mathcal{A}}\right)\right)$;
(24) $\quad\quad \boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}} = \lambda\Pi_{\Upsilon}\left(\boldsymbol{\delta}^{i,r+1}_{\mathcal{A}}\right) + (1-\lambda)\boldsymbol{\gamma}^{i,r}_{\mathcal{A}}$;
(25) $\quad\quad \Delta\boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}} = \boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}} - \boldsymbol{\gamma}^{i,r}_{\mathcal{A}}$;
(26) $\quad\quad \boldsymbol{v} := \boldsymbol{v} + E^i_{\mathcal{A}}\Delta\boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}}$.
(27) $\quad$ *// Loop on bilateral constraints*
(28) $\quad$ **for** $i := 1$ **to** $n_{\mathcal{B}}$
(29) $\quad\quad \delta^{i,r+1}_{\mathcal{B}} = \left(\gamma^{i,r}_{\mathcal{B}} - \omega\eta^i_{\mathcal{B}}\left(\nabla\boldsymbol{\Psi}^{i,T}\boldsymbol{v}^r + b^i_{\mathcal{B}}\right)\right)$;
(30) $\quad\quad \gamma^{i,r+1}_{\mathcal{B}} = \lambda\Pi_{\Upsilon}\left(\delta^{i,r+1}_{\mathcal{B}}\right) + (1-\lambda)\gamma^{i,r}_{\mathcal{B}}$;
(31) $\quad\quad \Delta\gamma^{i,r+1}_{\mathcal{B}} = \gamma^{i,r+1}_{\mathcal{B}} - \gamma^{i,r}_{\mathcal{B}}$;
(32) $\quad\quad \boldsymbol{v} := \boldsymbol{v} + \boldsymbol{E}^i_{\mathcal{B}}\Delta\gamma^{i,r+1}_{\mathcal{B}}$.
(33)
(34) **return** $\boldsymbol{\gamma}_{\mathcal{E}}$, $\boldsymbol{v}$

maximum number of iterations if the simulation must meet hard-real-time requirements.

With minimal modifications to the $\Pi_{\Upsilon}(\cdot)$ operator, the proposed method can be easily adapted to the case of friction in 2D or the case of generic unilateral constraints.

In our simulations, we chose $\omega = 1$ and $\lambda = 1$, except for the $K = 0$ case, where we used $\omega = 0.2$. We cannot guarantee a priori that this will satisfy condition [A3], but it did for all our simulations. In addition, the matrix sequences K^r and B^r were constant. We can therefore claim that Theorem 1 does apply and, since the sequence did not diverge, any accumulation point is a solution of the cone complementarity problem (47a). In addition, our proofs of the theoretical results allow for similar conclusions if ω varies from iteration to iteration. Therefore, we could ensure that at some iteration the appropriate ω is chosen after decreasing its value a few times until assumption [A3] holds. It can be shown that if the value of ω is halved each time [A3] does not hold and the respective iteration is rejected, then [A3] will eventually be satisfied after a finite number of steps. In our experiments, however, the values we have chosen for ω and λ have worked for all iterations without need of further adjustment.

4.2. *Optimized projected Gauss-Seidel CCP*

Another option is to take K as the lower block structure of N, which results in a scheme similar to a projected Gauss-Seidel. The K matrix does not need to be explicitly built: its effect is that, as soon as computed, a reaction impulse γ^i will be used also for computing the following γ^{i+1} impulse, and so on for all i, without needing to finish a single iteration. In practical terms this means that the $\sum_{z=1}^{n_{\mathcal{A}}} D^z\boldsymbol{\gamma}^{z,r}_{\mathcal{A}}$ term in Eqs. (49) and (51) is split in $\sum_{z=1}^{i-1} D^z\boldsymbol{\gamma}^{z,r+1}_{\mathcal{A}} + \sum_{z=i}^{n_{\mathcal{A}}} D^z\boldsymbol{\gamma}^{z,r}_{\mathcal{A}}$, and the same for bilateral constraints; so the difference from Algorithm 1 is that after the update of a single multiplier we immediately update $\boldsymbol{v}^{(l+1)}$, as shown in Algorithm 2. Note that, to update the speeds, we avoid the full summation (36) and add only the contributions caused by the change of the single multiplier after the projection (50):

$$\Delta\boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}} = \boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}} - \boldsymbol{\gamma}^{i,r}_{\mathcal{A}}; \tag{53}$$

$$\boldsymbol{v}^{(l+1),i+1} = \boldsymbol{v}^{(l+1),i} + M^{-1}D^i\Delta\boldsymbol{\gamma}^{i,r+1}_{\mathcal{A}}. \tag{54}$$

Hence, the computational overhead is not different from Algorithm 1.

Numerical tests show that this scheme converges faster than the case of $K = 0$; moreover, $K = 0$ is more prone to slow convergence in case of redundant constraints. [3]

[3] Redundancy in constraints is frequent in simulations of degenerate contact situations, such as flat surface against flat surface, where the collision engine may create a large amount of superfluous contact points.

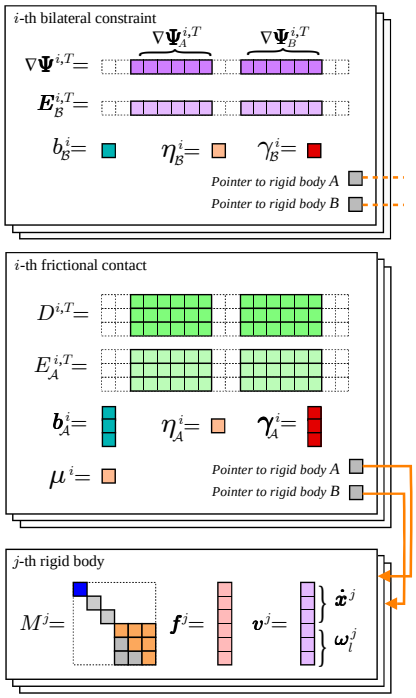

Fig. 3. Sparse data structures used by the solver.

5. Optimizations and improvements to the method

The proposed method can be further developed by introducing some algorithmic and theoretical improvements, obtaining different flavours of the original scheme. This section discusses the most significant optimizations.

5.1. *Transient data structures*

Figure 3 shows how the multibody model is represented by structures that are placed in memory. This transient data can be allocated on the heap during run-time, creating lists with unlimited numbers of constraints and rigid bodies.

Basically, each object that builds up the lists of bilateral constraints encapsulates the pointers to the two connected bodies, the Lagrange multiplier $\gamma^i_{\mathcal{B}}$, the constraint residual $b^i_{\mathcal{B}}$, the scalar value $\eta^i_{\mathcal{B}}$, and the Jacobian $\nabla\boldsymbol{\Psi}^i$.

The Jacobian should be a long row with m_v elements, but this would mean that the storage requirement for each constraint depends on the number of bodies, hence leading to an algorithm with superlinear space complexity. Instead, we store only the two small portions of $\nabla\boldsymbol{\Psi}^i$ that are not null, that is, the two parts $\nabla\boldsymbol{\Psi}^i_A$ and $\nabla\boldsymbol{\Psi}^i_B$ corresponding to the two connected bodies, hence requiring only a fixed number of 12 elements per Jacobian. When the $\nabla\boldsymbol{\Psi}^{i,T}\boldsymbol{v}^r$ multiplication must be performed, it is computed as $\nabla\boldsymbol{\Psi}^{i,T}_A\boldsymbol{v}^r_A + \nabla\boldsymbol{\Psi}^{i,T}_B\boldsymbol{v}^r_B$. Similar considerations apply to the data structure for frictional contacts, except that also μ^i coefficients must be stored and that Jacobians are made of two larger blocks, each with three rows. Thanks to this major optimization, the memory requirement per constraint and per contact is constant, so we obtain an optimal $O(n_{\mathcal{E}})$ space complexity. [4]

Looking at Algorithms 1 and 2, one can see that the solver can operate directly on these structures. Hence, no temporary matrices are necessary. Our method is thus a matrix-free method.

5.2. *Stabilization factor*

The stabilization terms $\frac{1}{h}\Psi^i(\boldsymbol{q}^{(l)})$ and $\frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)})$ in (16) and (17) are used to avoid constraint drifting during the time integration [4]. In fact, if these terms were missing, equations (16) and (17) would simply enforce the closure of constraints at the speed level, but errors might slowly accumulate in constraint positions after several integration steps. [5] .

If the model were linear (that is, the matrix $D_{\mathcal{E}}$ were constant in space, and, implicitly, in time), these terms would close constraint gaps, if any, in a single step. We experienced that, for bilateral constraints, this approach works well even if the time integration step h is small. However, this is not always the case when dealing with unilateral constraints, for the reason that follows. Because of numerical issues and nonlinearities, it is not possible to avoid some amount of penetration in contact constraints. For a contact with penetration, the term $\frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)})$ has a negative value. Hence, the inequality (17) requires that $\nabla\Phi^i\boldsymbol{v}^{(l+1)}$ (the speed of detachment of the contact) be large enough to bring the two surfaces at zero distance in a single step h. After the time integration step, at the next step the two surfaces might keep the $\frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)})$ separation speed, hence causing a bouncy behavior. Apart from being not predictable, this side effect gets worse if small timesteps h are used, because even the smallest interpenetration might cause macroscopic effects that can be perceived as bouncy motions even in contacts that should have no restitution.

We investigated different strategies to improve the original stabilization method. One idea was to scale the terms by K^i factors, with $0 < K^i < 1$:

$$\frac{K^i}{h}\Psi^i(\boldsymbol{q}^{(l)}) \quad , \quad \frac{K^i}{h}\Phi^i(\boldsymbol{q}^{(l)}).$$

This diminishes, but does not exclude, the risk of exceeding the interpenetration correction in a single step. Moreover, it has the drawback of creating artificial softness in stacked objects, since contact stabilization is somewhat delayed in successive frames.

A second approach, which we successfully used in many tests, involves using a clamping A as the maximum orthogonal speed for penetration recovery. This does not eliminate the risk of popping out from an intersecting contact, but at least the residual speed of separation, if any, is often negligible (comparable with

[4] We experienced that in many cases of granular flow simulations, the number of contacts tends to scale linearly with the number of rigid bodies n. Hence, under those circumstances the algorithm shows approximate linear space complexity $O(n)$ also in the number of bodies.

[5] This would happen either because of the numerical integration, whose finite precision cannot catch all geometric nonlinearities, or because of numerical truncation and errors.

the parameter A, which can be adjusted by the user) and independent of the time step h. Thus, the stabilization terms would become

$$\frac{1}{h}\Psi^i(\boldsymbol{q}^{(l)}) \quad , \quad \max\left(\frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}), -A\right).$$

Note that in this case, if the contact surfaces are separated, we have $\Phi^i > 0$ and the A clamping has no effect; thus it behaves as the original scheme with the $\frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)})$ term.

Until now we discussed how the stabilization term can correct the contact penetration errors when the surface distance is negative. However, it also has a useful side effect also when surfaces are not yet in contact. In fact, Eq. (17) can be interpreted as follows: the contact constraint is enforced only if the two surfaces are approaching fast enough to close the positive gap $\Phi^i(\boldsymbol{q}^{(l)})$ in a h time step. This allows us to include in the multibody system also contact constraints that are not yet in contact but simply within a "warning envelope"; later, the cone complementarity solver will do the rest.

The third approach is more expensive in terms of CPU effort, because it avoids constraint drifting by solving an additional complementarity problem over body positions, at each time step [12]. In this case, one solves the speed CCP problem using no stabilization terms on bilateral constraints, except for contacts that are not yet in contact, where it is useful to have

$$\max\left(\frac{1}{h}\Phi^i(\boldsymbol{q}^{(l)}), 0\right),$$

so that contacts with clearance are still allowed to approach until contact, because of positive $\frac{1}{h}\Phi^i$, while surfaces already in contact are forced to have a separation speed greater than or equal to 0, regardless of the amount of penetration. Later, after the time step advancement, one performs the following poststabilization step, that is a a linear complementarity problem that corrects the positions $\boldsymbol{q}$:

$$M\Delta\boldsymbol{q} = \sum_{i\in\mathcal{G}_\mathcal{A}} \left(\beta_n^i \boldsymbol{D}_n^i\right) + \sum_{i\in\mathcal{G}_\mathcal{B}} \left(\beta_b^i \nabla\Psi^i\right) \tag{55}$$

$$0 = \Psi^i(\boldsymbol{q}^{(l)}) + \nabla\Psi^{i^T}\Delta\boldsymbol{q}, \quad i \in \mathcal{G}_\mathcal{B} \tag{56}$$

$$0 \le \Phi^i(\boldsymbol{q}^{(l)}) + \nabla\Phi^{i^T}\Delta\boldsymbol{q} \perp \beta_n^i \ge 0, \qquad i \in \mathcal{G}_\mathcal{A} \tag{57}$$

This problem can be solved by using the same solver used for the speed CCP problem. The poststabilization idea performs better in the case of very large penetrations and ill-posed initial conditions.

In Fig. 4 a benchmark shows that the effort in reducing the maximum penetration error $||\epsilon_q||_\infty$ with poststabilization iterations is almost the same as that of the basic approach using only the stabilization coefficient in the speed CCP. However some iterations on the speed CCP still must be performed. For instance, in our complex simulations involving dense granular flow, the number of iterations of the poststabilization is comparable to the number of iterations for the speed CCP, so computational efforts are almost doubled.

A family of methods can be generated by balancing the total overhead toward the speed CCP iterations

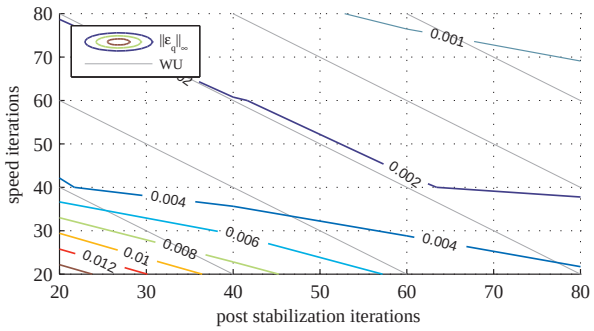

Fig. 4. Penetration error: tradeoff between speed iterations and poststabilization iterations, compared to WU overhead (GPU working units).

or the poststabilization iterations. In the extreme case where no iterations for the speed CCP are performed and iterations are performed only for the poststabilization, this method becomes similar to the *position based dynamics* algorithm, proposed in [27] as a robust method for real-time simulations.

Note that in this section we did not discuss the case of collisions with restitution and we assumed the simulation of contacts with fully plastic collisions. The reader interested in modifications to the method, for simulating also of the restitution phase, can read [1].

5.3. *Parallelization of the algorithm*

Algorithm 2 is inherently sequential, so Algorithm 1 can be easier to implement on computational architectures exploiting parallel processing because it does not feature data dependency for most of the inner loop.

In Algorithm 1, loops at rows 23 and 27 can be easily executed in parallel because they do not need to write at the same memory address at the same time. In the best scenario, one could run up to $n_\mathcal{A}+n_\mathcal{B}$ independent threads, each performing an update of its multiplier γ.

However, the sums at row 30 are more critical if parallelized in the same way, that is, on the basis of a thread per contact, since there is the risk that multiple threads will need to update the same element of the speed vector at the same time (this situation happens, for instance, if two contacts that refer to the same body are processed simultaneously). This problem can be solved in many ways, for example, by using a vector of Boolean mutexes, one per rigid body, which can prevent one thread to modify the speed of a rigid body if some other thread is writing into it. Otherwise, if the number of physical threads is very large as in GPU architectures, it is worth performing the sums in 30 using parallel-reduction algorithms.

We implemented this parallel algorithm on both a multicore processor and on an NVIDIA Tesla C870 GPU board featuring a massively multithreaded architecture, thanks to a stream processor that is capable of processing hundreds of threads in parallel. In the latter case, we experienced a speedup of 15 times respect to the serial implementation [44].

6. Examples

We present two benchmarks that we used to test the performance of our algorithm and an application to the simulation of the refueling of a nuclear reactor.

6.1. *Forklift truck simulator*

As a complex mechanical system, a forklift truck represents a significative benchmark for our algorithm because it entails most aspects that we discussed in this article: rigid bodies, bilateral constraints, applied forces, and unilateral contact constraints with friction. In detail, our truck model is made of seven rigid bodies (the frame, three wheels, the steering strut, the mast, the carriage with the forks) connected by six revolute and prismatic joints. The tilting of the mast, the lifting of the carriage, and the steering are obtained by introducing rheonomic bilateral constraints. This is a simplified approach, but more detailed models of the actuators, with hydraulic components and feedback controllers, are also possible within this framework. Motors and brakes provide torques to the wheels; motors, brakes, and actuators can be controlled in real time by the user, using a joystick or a keyboard, or by using automatic procedures.

In order to also test frictional constraints, a seventh rigid body is added: a wooden pallet of EUR/ISO1 type: the truck can pick and move such a pallet with the forks. Specific collision shapes have been defined to detect the contact points between the pallets, the environment, and the truck. Collision shapes have been used also for frame of the forklift and its overhead guard, so that we can simulate the roll-over of the vehicle and other hazardous events.

On our test system, a dual-core Centrino T2600 2.17 GHz with 2 GB of RAM, the proposed algorithm is able to simulate the truck in faster than real time, using a fixed timestep $h = 0.005$ s.

To assess the efficiency and capability of our matrixless approach, we simulated an increasing number of forklift trucks, up to 1,600 vehicles with 1,600 pallets, see Fig.5. In the largest simulation scenario, with 12,800 rigid bodies, the algorithm must handle 54,400 bilateral constraints and, on average, 19,000 frictional contacts: this means more than 110,000 primal and dual variables.

We used Algorithm 1 with a limit of 20 iterations. Table 1, averaged over 100 steps, shows that the CPU overhead grows almost linearly with the number of dual variables $3n_{\mathcal{A}} + n_{\mathcal{B}}$, that is, somehow proportional to the number of bodies.

Note that the frame rate in the case of hundreds of trucks, although not real time as for few trucks, is still fast and interactive.

Increasing the number of iterations results in an improvement of the precision: going from 20 iterations to 80 iterations, the largest errors in constraint position and in constraint velocity decrease, respectively, from 0.0310 mm and 0.024 m/s to 0.006 mm and 0.002 m/s. These results about precision are not dependent on the number of trucks, because the convergence of the solver is not affected by the increasing complexity of this type of benchmark, where the dynamics of each vehicle is uncoupled. On the other hand, our results indicate the efficiency of our algorithms, which is due to our customization to rigid-body dynamics structure. For example, the most efficient off-the-shelf algorithms in [32], which are either of either the interior-point or the projected gradient type, still need around 20 s per time step for about 10,000 dual variables. Those algorithms solve the same problem as here. If linear scaling would hold for those methods – a big assumption in their favor – our algorithm will still be more than 100 times faster.

Table 1
Time-step performance.

No. of Trucks	CCP Solve [ms]	Collision [ms]	Bodies	$3n_{\mathcal{A}} + n_{\mathcal{B}}$
1	0.22	0.1	8	70
400	135	39	3200	28000
800	268	79	6400	56000
1200	396	130	9600	84000
1600	550	200	12800	112000

How precision and convergence can be affected by systems with more stringent topology is investigated in the following example.

6.2. *Dense granular packing benchmark*

Dense stacking of granular material is one of the hardest problems involving nonsmooth rigid body dynamics. Indeed, the benchmark described in this section involves the simulation of the progressive stacking of many convex shapes in an empty box, with different settings.

The rigid bodies have a mass $m = 10$ kg, the moments of inertia are $I_{xx} = I_{yy} = I_{zz} = 10.24$ kgm^2, and the friction coefficient is $\mu = 0.4$. The horizontal section of the box measures 20 m$\times$20 m. We performed tests with different types of colliding shapes, but for the graphs presented here we used spheres with a radius $r = 1.6$ m. At the beginning of the simulation the box is filled by 220 spheres, with randomized positions and with an initial volume fraction of 0.4, on average. After the spheres have settled, we obtained the plots.

Figures 6 and 7 show a typical convergence pattern of the PGS CCP algorithm: ϵ_v is the violation of the constraints at the speed level. For increasing ω, the iteration shows a faster convergence. However, large ω values may lead to nonmonotonic behavior (that does not necessarily lead to divergence). We found that a good tradeoff between convergence speed and a monotonic nondivergent iteration, for scenarios involving equally sized spheres, is $\omega = 1$.

Figure 8 shows that the λ parameter acts like a smooting factor over the iteration. The higher the parameter the slower is the convergence, but the lower is the risk of nonmonotonic or divergent patterns. The optimal value depends on the type of simulation; we experienced that, on average, good default values can

Fig. 5. Benchmark: simulation of thousands of forklifts.

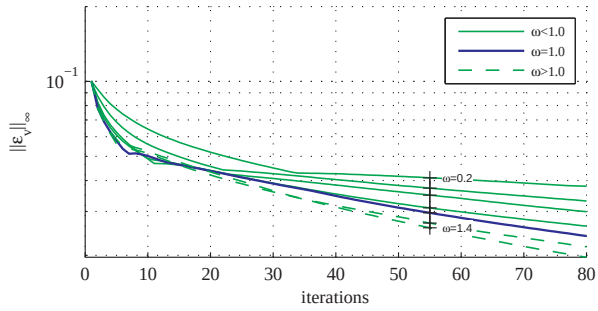

Fig. 6. Convergence of the residual for varying ω, for a sample time step in the 220-sphere benchmark.

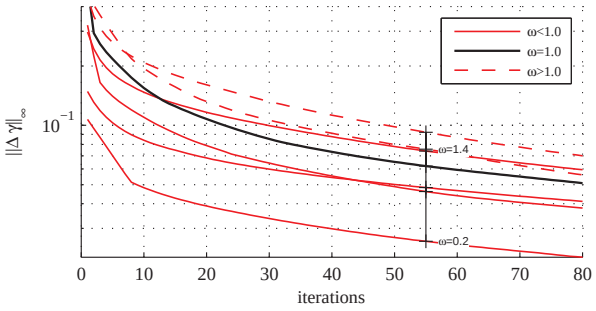

Fig. 7. Convergence of $\Delta\gamma$ for varying ω, for a sample time step in the 220-sphere benchmark.

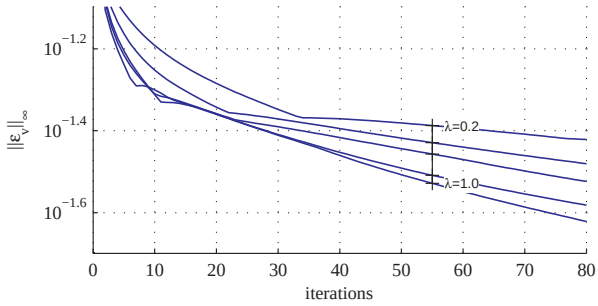

Fig. 8. Effect of the smoothing parameter λ, for fixed $\omega = 1$.

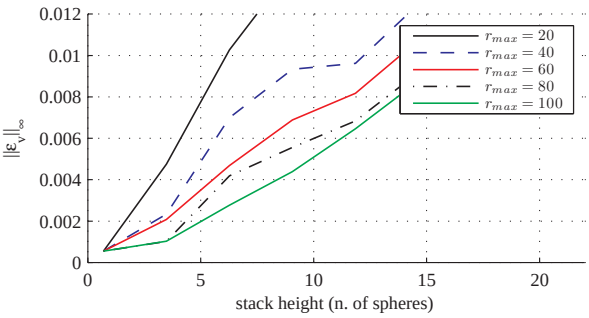

Fig. 9. Effect of the vertical size of the stack on the convergence, for varying number of iterations.

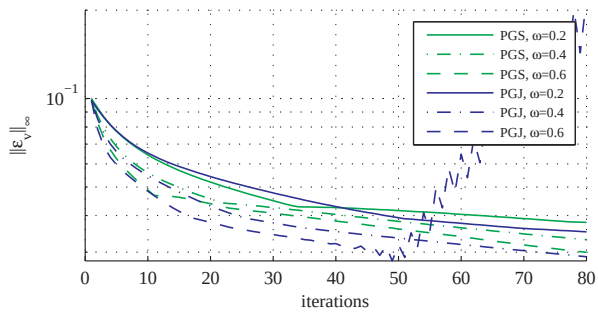

Fig. 10. Comparison of the convergence of the PGS and PGJ algorithms during the granular stacking benchmark, and example of divergence.

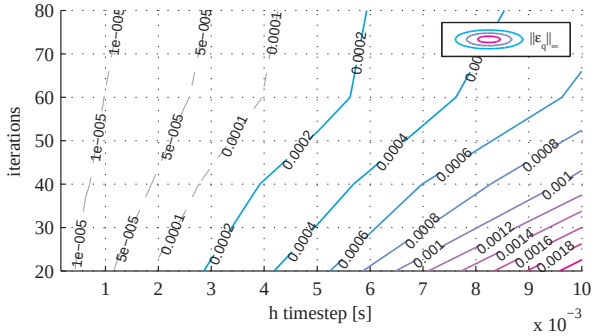

Fig. 11. Penetration error: combined effect of time step and number of iterations during the benchmark.

be chosen in the range $\lambda = 0.8 \div 1.0$.

The convergence can be largely affected by the topology of the mechanical system; the best-case scenario being many single objects on a flat plane, and the worst case being the objects stacked in a vertical row. This is shown in Fig. 9, where we performed simulations with the same number of spheres but with different sizes of boxes. The precision of the iteration, for a fixed number

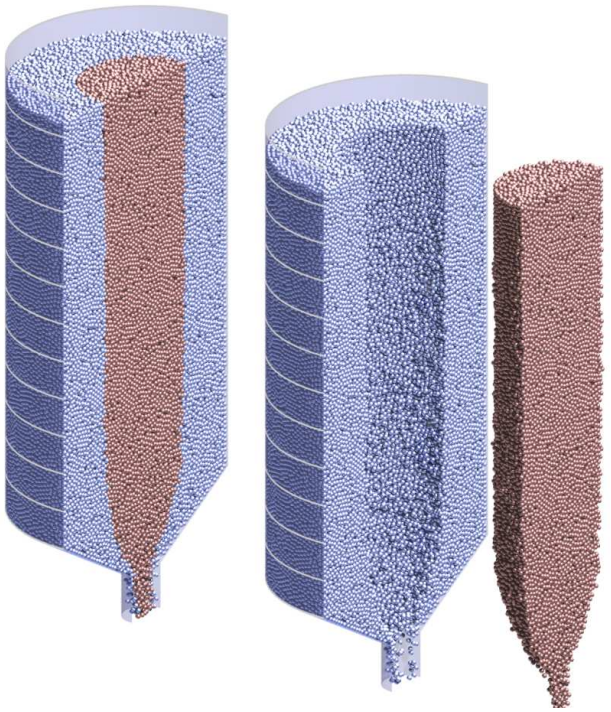

Fig. 12. Simulation of bidisperse granular flow in a PBR nuclear reactor (170,000 bodies). The inner column of graphite pebbles shows almost no dispersion in the surrounding fissile material.

of iterations, deteriorates proportionally to the height of the stack.

Figure 10 shows that, for low ω, the convergence of the PGJ CCP algorithm is comparable to the convergence of the PGS CCP algorithm. However the PGJ CCP method is more likely to run into divergence, especially in the case of redundant constraints. Therefore, it should be used with low ω, for example $\omega = 0.2$ in this case.

In Fig. 11 we show the results of various simulations with different time steps and iteration numbers, while monitoring the average penetration error in constraints, ϵ_p. The figure shows that a good precision in satisfying the constraints at the position level can be achieved both by increasing the number of iterations and by decreasing the time step h.

These plots are obtained for a sample time step, after 5 seconds from the beginning of the granular stacking simulation. Hence, previous time steps and different scenarios can lead to slightly different graphs. Although this is only a numerical benchmark, it is a worst-case scenario, and its results are indicative about the behavior of the solution method when dealing with practical engineering problems that share the same theoretical difficulties (masonry stability, soil compaction, etc.).

6.3. *Refueling cycle in a pebble bed nuclear reactor*

A significant application, which may benefit from the robustness and the speed of the method, is the simulation of the granular flow in the pebble bed nuclear reactor PBR [18].

The PBR reactor features a fourth-generation design based on a slow recirculation of fuel pebbles in a large silo: actinides are coated and packed with graphite moderator in the spherical pebbles, each with a typical diameter of 60 mm, while the helium coolant flows between the pebbles. To increase the efficiency, a central column of spheres could contain only graphite, to flatten the neutron flux. Spheres are slowly extracted from the bottom, reprocessed, and reinserted at the top. The simulation of the downward granular flow can be useful in estimating statistical parameters such as the void fraction or the dispersion of the vertical column, hence guiding more efficient designs that can maximize the burnup of the actinides. A past attempt at simulating a PBR reactor required one week on a 64-processor supercomputer at Sandia National Laboratories, using the discrete element method (DEM) [34]. Unlike DEM methods, our approach does not introduce stiff force fields, and larger time steps are allowed, so we could perform the simulation on a single laptop computer in few hours. On average, this problem involved 170,000 rigid bodies, more than 500,000 frictional contacts, leading to more than two millions of primal and dual variables (Fig. 12).

In [42] we presented results of these simulations and validation against experimental data. Because of the high stack of spheres, this example falls in the class of problems with slow convergence already discussed in the previous benchmark: this advocates for future research efforts that could improve the performance of the solver by leveraging on multiscale and domain decomposition implementations. Nevertheless, we point out that even under this circumstance we have shown in [42] that we correctly compute macroscopic parameters of the granular flow in the pebble bed reactor, such as porosity, while needing only a few hours on a laptop.

7. Conclusions

We presented a formulation for multibody systems with large amounts of bilateral constraints and frictional contacts, and we developed an iterative method for this purpose. Our approach poses the problem as a convex optimization that can be solved as a cone complementarity problem. The proposed fixed-point iteration has been tailored to feature high performance even in large simulation scenarios: special care has been devoted in optimizing algorithms and formulas and avoiding matrix storage, hence obtaining an $O(n)$ space complexity.

The method converges under standard assumptions on the configuration of the system, resulting in a robust algorithm that can simulate systems with millions of multipliers.

Improvements have been presented for the stabilization of contact constraints, resulting in matrix-free schemes that can correct large interpenetrations of rigid bodies without running into numerical problems.

We implemented a multibody system based on the method presented in this paper: the Chrono::Engine library [41]. Aiming at high performance, we expanded substantial efforts optimizing the C++ source code.

This software has been already used to simulate complex systems that were hardly tractable with other applications: granular flows in silos, interaction of wheels with sand and pebbles, size segregation devices, and other problems with a large number of bodies.

The method has been recently ported also on parallel stream-kernel GPU hardware, obtaining a remarkable computational efficiency [44].

Acknowledgments

We thank Erwin Coumans for hints about collision detection algorithms, and Dan Negrut for comments on an earlier version of the paper. Mihai Anitescu was supported by Contract DE-AC02-06CH11357 of the U.S. Department of Energy.

References

[1] M. Anitescu. A fixed time-step approach for multibody dynamics with contact and friction. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings*, volume 4, 2003.

[2] M. Anitescu. Optimization-based simulation of nonsmooth rigid multibody dynamics. *Math. Program.*, 105(1):113–143, 2006.

[3] M. Anitescu, J. F. Cremer, and F. A. Potra. Formulating 3d contact dynamics problems. *Mechanics of Structures and Machines*, 24(4):405–437, 1996.

[4] M. Anitescu and G. D. Hart. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. *International Journal for Numerical Methods in Engineering*, 60(14):2335–2371, 2004.

[5] M. Anitescu and G. D. Hart. A fixed-point iteration approach for multibody dynamics with contact and friction. *Mathematical Programming, Series B*, 101(1)(ANL/MCS-P985-0802):3–32, 2004.

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

[7] M. Anitescu, F. A. Potra, and D. Stewart. Time-stepping for three-dimensional rigid-body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 177:183–197, 1999.

[8] M. Anitescu and A. Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications*, pages Accepted for printing, DOI 10.1007/s10589–008–9223–4, 2008.

[9] D. Baraff. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica*, 10:292–352, 1993.

[10] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Computer Graphics (Proceedings of SIGGRAPH)*, pages 23–34, 1994.

[11] G. V. D. Bergen and G. J. A. Bergen. *Collision Detection in Interactive 3D Environments*. Morgan Kaufmann, 2004.

[12] M. Cline and D. Pai. Post-stabilization for rigid body simulation with contact and constraints. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 3744–3751, 2003.

[13] R. Cottle and G. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and Its Applications*, 1:103–125, 1968.

[14] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, Boston, 1992.

[15] B. R. Donald and D. K. Pai. On the motion of compliantly connected rigid bodies in contact: a system for analyzing designs for assembly. In *Proceedings of the Conf. on Robotics and Automation*, pages 1756–1762. IEEE, 1990.

[16] S. K. E.G. Gilbert, D.W. Johnson. A fast procedure for computing the distance between complex objects in three-dimensional space. *Robotics and Automation*, 4(2):193–203, 1988.

[17] C. Ericson. *Real-Time Collision Detection*. Elsevier, 2005.

[18] H. D. Gougar. *Advanced core design and fuel management for pebble-bed reactors*. Ph.D thesis, Penn State University, Department of Nuclear Engineering, 2004.

[19] G. D. Hart. *A Constraint-stabilized Time-stepping Approach for Piecewise Smooth Multibody Dynamics*. Ph.D thesis, University of Pittsburgh, Department of Mathematics, April 2007.

[20] E. J. Haug. *Computer Aided Kinematics and Dynamics of Mechanical Systems*. Allyn and Bacon, Boston, 1989.

[21] E. J. Haug, S. Wu, and S. Yang. Dynamic mechanical systems with coulomb friction, stiction, impact and constraint addition-deletion. *Mechanisms and Machine Theory*, 21(5):407–416, 1986.

[22] J.-B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Berlin, 1993.

[23] Y. J. Kim, M. C. Lin, and D. Manocha. Deep: Dual-space expansion for estimating penetration depth between convex polytopes. In *Proceedings of the 2002 International Conference on Robotics and Automation*, volume 1, pages 921–926. Institute for Electrical and Electronics Engineering, 2002.

[24] P. Lotstedt. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal of Applied Mathematics*, 42(2):281–296, 1982.

[25] M. D. P. Marques. *Differential Inclusions in Nonsmooth Mechanical Problems: Shocks and Dry Friction*, volume 9 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser Verlag, Basel, 1993.

[26] J. J. Moreau. Standard inelastic shocks and the dynamics of unilateral constraints. In G. D. Piero, F. Macieri, and S. Verlag, editors, *Unilateral Problems in Structural Analysis*, pages 173–221, New York, 1983. CISM Courses and Lectures no. 288.

[27] M. Müller, B. H. M. Hennix, and J. Ratcliff. Position based dynamics. In *Proceedings of Virtual Reality Interactions and Physical Simulations*, pages 71–80, 2006.

[28] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Helderman Verlag, Berlin, 1988.

[29] J. Pang and D. Stewart. Solution dependence on initial conditions in differential variational inequalities. *Mathematical Programming*, 116(1):429–460, 2009.

[30] J.-S. Pang, V. Kumar, and P. Song. Convergence of time-stepping method for initial and boundary-value frictional compliant contact problems. *SIAM J. Numer. Anal.*, 43(5):2200–2226, 2005.

[31] J.-S. Pang and J. C. Trinkle. Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with coulomb friction. *Math. Program.*, 73(2):199–226, 1996.

[32] C. Petra, B. Gavrea, M. Anitescu, and F. Potra. A computational study of the use of an optimization-based method for simulating large multibody systems. *Optimization Methods and Software*, 24(6):871–894, 2009.

[33] F. Pfeiffer and C. Glocker. *Multibody Dynamics with Unilateral Contacts*. John Wiley, New York City, 1996.

[34] C. Rycroft, G. Grest, J. Landry, and M. Bazant. Analysis of granular flow in a pebble-bed nuclear reactor. *Physical Review E*, 74, 021306, 2006.

[35] P. Song, P. Kraus, V. Kumar, and P. Dupont. Analysis of rigid-body dynamic models for simulation of systems with frictional contacts. *Journal of Applied Mechanics*, 68(1):118–128, 2001.

[36] P. Song, J.-S. Pang, and V. Kumar. A semi-implicit time-stepping model for frictional compliant contact problems. *International Journal of Numerical Methods in Engineering*, 60(13):267–279, 2004.

[37] D. Stewart and J.-S. Pang. Differential variational inequalities. *Mathematical Programming*, 113(2):345–424, 2008.

[38] D. E. Stewart. Convergence of a time-stepping scheme for rigid body dynamics and resolution of Painleve's problems. *Archive Rational Mechanics and Analysis*, 145(3):215–260, 1998.

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

[40] D. E. Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction. *International Journal for Numerical Methods in Engineering*, 39:2673–2691, 1996.

[41] A. Tasora. Chrono::Engine project, web page. *www.deltaknowledge.com/chronoengine*, 2006.

[42] A. Tasora and M. Anitescu. A convex complementarity approach for simulating large granular flow. *Journal of Computational Nonlinear Dynamics*, 2009. To appear.

[43] A. Tasora, E. Manconi, and M. Silvestri. Un nuovo metodo del simplesso per il problema di complementarit lineare mista in sistemi multibody con vincoli unilateri. In *Proceedings of AIMETA 05*, Firenze, Italy, 2005.

[44] A. Tasora, D. Negrut, and M. Anitescu. Large-scale parallel multi-body dynamics with frictional contact on the graphical processing unit. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 222(4):315–326, 2008.

[45] J. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with Coulomb friction. *Zeithschrift fur Angewandte Mathematik und Mechanik*, 77:267–279, 1997.