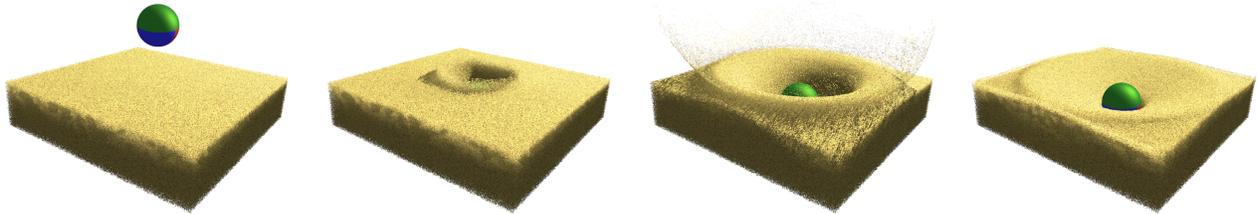


# A Semi-Implicit Material Point Method for the Continuum Simulation of Granular Materials

Gilles Daviet

Florence Bertails-Descoubes

INRIA and Laboratoire Jean Kuntzmann (Grenoble University, CNRS), France\*



**Figure 1:** A steel ball impacts a granular bed. Unlike previous approaches, our fully resolved (non-linearized) Drucker-Prager rheology allows us to retrieve a perfectly round crater.

## Abstract

We present a new continuum-based method for the realistic simulation of large-scale free-flowing granular materials. We derive a compact model for the rheology of the material, which accounts for the exact nonsmooth Drucker-Prager yield criterion combined with a varying volume fraction. Thanks to a semi-implicit time-stepping scheme and a careful spatial discretization of our rheology built upon the Material-Point Method, we are able to preserve at each time step the exact coupling between normal and tangential stresses, in a stable way. This contrasts with previous approaches which either regularize or linearize the yield criterion for implicit integration, leading to unrealistic behaviors or visible grid artifacts. Remarkably, our discrete problem turns out to be very similar to the discrete contact problem classically encountered in multibody dynamics, which allows us to leverage robust and efficient nonsmooth solvers from the literature. We validate our method by successfully capturing typical macroscopic features of some classical experiments, such as the discharge of a silo or the collapse of a granular column. Finally, we show that our method can be easily extended to accommodate more complex scenarios including two-way rigid body coupling as well as anisotropic materials.

**Keywords:** Granular materials, continuum modeling, material point method, Drucker-Prager yield criterion, nonsmooth rheology

**Concepts:** •Computing methodologies → Simulation by animation; Physical Simulation;

## 1 Introduction

Walking across sand dunes or pouring a teaspoon of sugar into the morning coffee are common natural scenes, in which granular materials (such as sand, soil, cereals, salt and sugar) are key components. In Computer Graphics, simulating the motion of such materials realistically is essential for capturing the rich interactions occurring between virtual characters and their environment.

Granular materials (see, e.g., [Andreotti et al. 2011] for a comprehensive description) commonly refer to a large collection of small solid grains larger than  $100\ \mu\text{m}$  in size — which typically distinguishes them from powders, made of much smaller grains. Considering this limit size, grain-grain interactions in granulars are mainly dictated by *contact* and *dry friction*, while air-grain interactions can be neglected. *Cohesion* between grains may furthermore be considered, typically in the case of wet materials.

Due to their discrete nature at macroscopic scale, granular materials are highly dissipative systems which exhibit various physical states, ranging from “liquid” (when the material continuously flows) to “solid” (when the material rigidifies due to internal friction and undergoes plastic deformations), or even “gaseous” (when grains are dispersed in air and interact mainly through impacts). Such a rich visual behavior, very distinctive from that of Newtonian fluids (e.g., water), explains why simulating granular matter has, beyond classical applications in Physics and Mechanical Engineering, gained increased attention in Computer Graphics for a few decades [Zhu and Bridson 2005; Alduán et al. 2009; Narain et al. 2010].

However, unlike Newtonian fluids or elastic materials for which robust physical models have been developed in the last centuries, granular matter still resists some comprehensive physical understanding: no universal equation exists yet for describing the collective behavior of such a peculiar material.

As a result, various methods have been developed for studying and simulating granular materials, following two main strategies. On the one hand, many approaches naturally rely upon a *discrete* model, where grains are represented individually as rigid bodies and grain-grain interactions are accounted for using a frictional contact model [Moreau 1994]. Because they explicitly model the material at the grain scale, those methods are able to capture the different states of the granular matter and their transitions quite accurately. However, considering that a simple teaspoon of sugar already consists of around  $10^5$  grains, one may easily imagine that such approaches rapidly suffer from scalability issues.

\*e-mail: {gilles.daviet,florence.descoubes}@inria.fr

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

SIGGRAPH '16 Technical Paper, July 24 - 28, 2016, Anaheim, CA,

ISBN: 978-1-4503-4279-7/16/07

DOI: <http://dx.doi.org/10.1145/2897824.2925877>

On the other hand, *continuum*-based methods, where granular matter is modeled as a viscoplastic fluid, have started to be investigated for a few decades. Though less accurate than discrete models, such methods have nevertheless demonstrated some good qualitative agreement with real scenarios in plastic and dense flowing regimes. In particular, a popular constitutive law is the  $\mu(I)$  rheology [Jop et al. 2006], relying upon a pressure-dependent yielding model — referred to as the *Drucker-Prager* yield criterion — with a non-constant friction coefficient. For the sake of tractability, existing numerical models however make some important simplifications, for instance assuming an incompressible flow, a uniform density, or a regularized or linearized Drucker-Prager yield criterion. Such approximations often result in a considerable loss of realism, for instance preventing a heap of sand from stabilizing after flowing, or introducing some artificial anisotropy in the flow.

In this paper we wish to capture the visual richness of large-scale 3D granular materials which may flow as a liquid material, while possibly transitioning to a solid behavior due to internal friction, or free-flowing as disperse grains in low density regions. To this aim, we opt for a continuum strategy and represent granular matter as a compressible viscoplastic fluid combined with a Drucker-Prager yield criterion and a unilateral compressibility constraint, similarly to [Narain et al. 2010]. Our main contribution is to propose the first semi-implicit discrete scheme fully accounting for this nonsmooth rheology, without regularizing nor linearizing it. Our results are validated against classical experimental laws reported in the literature, under controlled conditions. Furthermore, in the case of challenging graphics-like scenarios, our method demonstrates enhanced realism and reduced visual artifacts compared to previous work. We finally extend our method to the handling of two-ways rigid body coupling and anisotropy, and make a first step towards the implicit treatment of cohesion.

## 2 Related work

The last few decades have seen granular simulations find myriads of applications, ranging from geophysics and civil engineering to food-processing and pharmaceutical industries, as well as special effects. As a result, a large body of research has been devoted to the computational modeling of granular materials, mainly in Physics, Mechanical Engineering, and more recently in Computer Graphics. In the following we comment only on the most representative works from these various communities.

### 2.1 Discrete Models (DEM)

A natural way to model granular materials is by considering a large collection of individual grains. The overall motion of the material then results from the dynamics of each individual grain coupled with others through grain-grain interactions. Such a *discrete* modeling strategy is, depending on the communities, either referred to as a *particle-based* method, or to a *Discrete Element Method* (DEM). Among DEM, popular methods include the molecular dynamics approach [Bell et al. 2005; Alduán et al. 2009], where deformable grains interact through regularized contact and friction, and the nonsmooth contact dynamics approach [Moreau 1994; Jean 1999], where in contrast rigid grains are subject to non-penetration and Coulomb friction constraints.

The major drawback of DEM is the computational cost inherent to the individual treatment of grain-grain interactions, which becomes intractable in the case of large scale 3D scenarios containing billions of grains. Position Based Dynamics (PBD) methods such as [Side Effects Software 2015] have recently become popular in Computer Graphics to alleviate this cost. Such methods iteratively project grains positions to valid locations, without computing the interaction forces. As each pair of neighbouring grains still has to

be taken into account, PBD methods typically do not target global convergence but rather use a predetermined number of iterations. Their main drawback is that the control parameters are thus unphysical and non-scalable — the number of solver iterations determines the “stiffness of the material”. Quoting [Side Effects Software 2015], one “would primarily want to use [PBD] for simulating bouncy objects with elastic energy, or background objects”.

Alternatively, to scale up computations, one may instead consider that each particle or element represents a clump of material instead of a physical grain. Such a macroscopic view motivates the transition to continuum-based methods.

### 2.2 Continuous models

Continuous models for granular materials are derived upon macroscopic *yield criterions*, i.e., inequalities that should be satisfied by the principal stresses — the eigenvalues of the stress tensor — for the material to remain stable. The most well-known is the *Mohr-Coulomb* criterion. For cohesionless 3D materials such as dry sand, and with  $\sigma_3 \leq \sigma_2 \leq \sigma_1$  the principal stresses, it states

$$\sigma_1 - \sigma_3 \leq \sin \varphi (\sigma_1 + \sigma_3),$$

where  $\varphi$  is the so-called friction angle, which corresponds to the inclination of a stable heap of the granular material under gravity.

In the space of principal stresses, the Mohr-Coulomb criterion defines a cone with hexagonal basis, which is numerically unwieldy. It is thus often discarded in favor of the *Drucker-Prager yield criterion* on the second invariant  $J_2$  of the deviatoric stress tensor,

$$\sqrt{J_2} \leq \hat{\mu} \frac{\sigma_1 + \sigma_2 + \sigma_3}{3} \quad \text{where } J_2 := \frac{1}{6} \sum_{i \neq j} (\sigma_i - \sigma_j)^2,$$

which defines a second-order (ice-cream) cone in the space of principal stresses<sup>1</sup>. Note that in 2D, both criteria are equivalent. For a detailed explanation of how the friction coefficient  $\hat{\mu}$  relates to the friction angle  $\varphi$ , we refer the reader to Appendix A.

**Mesh-based approaches (FEM)** The recent introduction of the  $\mu(I)$  continuum rheology [Jop et al. 2006] has inspired several numerical simulation methods for dense granular flows, relying on finite elements (FEM) [Lagrée et al. 2011; Ionescu et al. 2015]. On well-studied 2D scenarios such as the granular column collapse, these approaches have been successfully compared to corresponding physical experiments as well as to DEM simulations.

Existing FEM techniques are however not directly applicable to complex 3D graphics scenarios. Indeed, they are making some strong assumptions regarding the dynamics of the flow — especially enforcing incompressibility — which precludes the simulation of free-flowing granular materials. In a recent work, Daviet and Bertails-Descoubes [2016] relaxed the incompressibility constraint and proposed a finite-element approach for dense yet dilatant flows, subject to the nonsmooth Drucker-Prager yield criterion. However, their approach is limited to 2D for performance reasons. Moreover, it considers that granular matter is always at its maximum compaction rate, and furthermore bound to a fixed-sized domain without air interface. In contrast, we consider here 3D free-flowing materials, where the air-grain mixture is modeled through a varying volume fraction of the grains.

<sup>1</sup>It could seem weird to replace a polyhedral yield surface with a quadratic one, while in the context of 3D Coulomb friction several authors argue that it is easier to use a polyhedral approximation of the friction cone. However, remember that the Mohr-Coulomb yield surface is only polyhedral in the space of principal stresses — the space defined by its eigenvectors — and not directly in the space of the tensor coefficients.



**Figure 2:** Final states after the collapse of a cylindrical column with exact (Frobenius norm, left) and linearized ( $\ell_\infty$ -norm, right) Drucker-Prager yield surfaces, both for  $\mu = 0.6$ .

Another drawback of FEM is inherent to the pure mesh-based representation. While such a representation is well-suited for modeling the averaged behavior of the flow, which is of prime interest for physicists, it is not sufficient for graphic applications where the capture of detailed geometry — in particular, the motion of individual grains in the diffuse air-grain interface — is of utmost importance. Far from being a mere visualization issue, this problem requires the accurate and temporally consistent tracking of geometric details, which can only be performed during the simulation stage. This brings a compelling argument for enriching the flow description with a particle representation.

**SPH methods** Smoothed-particle hydrodynamics, a fully meshless approach, has long been popular in Computer Graphics for simulating fluid flows, as it can capture details of the flow while being intrinsically free of grid artifacts.

In recent years, many authors have used this approach to model granular flows [Lenaerts and Dutré 2009; Alduán and Otaduy 2011; Chambon et al. 2011], using either some regularization or linearization of the yield criterion. However, as each particle affects its neighbors through its kernel function, satisfying the frictional constraints in a global manner may require several passes over the whole list of particles, inducing a prohibitive cost for very large numbers of particles. Ihmsen et al. [2013] reduce this cost by adaptively freezing particles, but cannot handle arbitrary scenarios.

**Hybrid methods** Mixing both previous approaches allows to keep the high-resolution details from particles while relying on a background mesh to reduce the number of degrees of freedom of the system, and therefore its computational cost.

Such hybrid methods were first applied to granular materials for Computer Graphics by Zhu and Bridson [2005]. Narain et al. [2010] extended this approach to free-flowing materials subject to a maximum volume fraction constraint, and used a staggered LCP method to account for the coupling between pressure and frictional stress. However, the use of a finite-difference staggered grid, which stores separately the components of the stress tensor, required the linearization of the yield criterion along the grid axis, eventually leading to visible artificial anisotropy (see [Narain et al. 2010, Figure 10, right] and our own comparison in Figure 2).

The Material Point Method (MPM) is a popular technique in Mechanical Engineering for simulating granular materials [Wieckowski et al. 1999; Mast et al. 2014; Wieckowski and Pawlak 2015; Dunatunga and Kamrin 2015]. In the aforementioned approaches, the solid regime is modeled with an elastic constitutive law, leading to numerically stiff problems, especially for dry sand which features a high Young modulus. As they rely upon explicit time-integration schemes, those techniques are thus bound to the use of very small timesteps, which considerably limits their applicability to graphics-like scenarios. In Computer Graphics, the elastoplastic model of [Stomakhin et al. 2013], originally devised for snow

simulation, has also been recently tuned to animate sand by Jiang et al. [2015]. However, the proposed approach lacks an implicit treatment of plasticity as well as a proper granular constitutive law.

In contrast, here we do not attempt to capture elasticity of the material, but rather claim that the solid regime may be just considered as purely rigid, without visual loss of realism. The main strength of our approach is our ability to enforce the resulting multivalued constitutive law in an *implicit* manner and over the whole domain, by leveraging tools from nonsmooth optimization. Such an implicit treatment yields stable results, even for large timesteps.

### 3 Contributions

In this paper we propose a stable and robust method for simulating 3D free-flowing granular flows, with enhanced realism compared to previous approaches. Our core contribution is the extension of the method of [Daviet and Bertails-Descoubes 2016], which simulates dense Drucker-Prager type flows, to the handling of complex 3D scenarios with a varying volume fraction of grains.

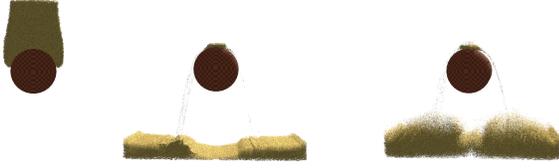
We first formulate a compact and spatially exact problem modeling our full rheology, which is then stably enforced at each time step in a semi-implicit manner (Section 4). In contrast to the finite-differences scheme used in [Narain et al. 2010], our spatial discretization relies on the Material Point Method, which allows us to use a finite-element formulation with collocated stress components (Section 5). Remarkably, our resulting discrete system may be recast into a discrete contact problem, which allows us to leverage robust and efficient nonsmooth solvers from multibody dynamics. We then extend this framework to include two-ways interactions with rigid bodies, and show how to incorporate anisotropic effects in our model (Section 6). A first step towards cohesive materials is also briefly presented. Finally, we validate our model against classical experiments of the literature and demonstrate the effects of its parameters on a simple scenario. We also carefully compare the performance of selected Coulomb friction algorithms on representative problems, before discussing the merits and limitations of our approach (Section 7).

### 4 A continuous medium

In this section we extend the usual continuous equations for granular materials (see, e.g., [Daviet and Bertails-Descoubes 2016]) to account for a grain volume fraction lower than 1. Our main result here is the construction of a compact and spatially exact problem modeling this new rheology constraint (System (7)), which is then enforced at each time step in a semi-implicit manner. The resulting one-step problem is, in turn, discretized spatially in Section 5.

**Notation** Let  $S_d$  be the space of symmetric  $3 \times 3$  tensors. For  $\sigma \in S_d$ , we denote by  $\text{Tr } \sigma$  its trace (normal part) and by  $\text{Dev } \sigma$  its deviatoric (tangential) part,  $\text{Dev } \sigma = \sigma - \frac{1}{3} \text{Tr } \sigma \mathbf{I}$ , where  $\mathbf{I}$  is the identity tensor. We shall use the norm  $|\cdot|$  associated to the scalar product  $\langle \sigma, \tau \rangle = \frac{\sigma : \tau}{2} = \frac{1}{2} \sum \tau_{ij} \sigma_{ij}$ . Note that  $|\cdot|$  amounts to the Frobenius norm scaled by  $\frac{1}{\sqrt{2}}$ .

Throughout this article, fields will be represented with upright letters and values with italic ones, e.g.,  $\mathbf{v} = \mathbf{v}(\mathbf{x})$  will denote the value  $v$  of the field  $\mathbf{v}$  at location  $\mathbf{x} \in \Omega$ , where  $\Omega \subset \mathbb{R}^3$  is our space domain. Expressions including upright letters without argument, such as  $\phi = \phi_{max}$ , will implicitly hold for the entire space  $\Omega$ , i.e., giving here  $\forall \mathbf{x} \in \Omega, \phi(\mathbf{x}) = \phi_{max}$ . The gradient of a scalar or vector field  $\phi$  will be written as  $\nabla \phi$ , and the divergence of a vector or tensor field  $\mathbf{v}$  as  $\nabla \cdot \mathbf{v}$ . Given a vector field  $\mathbf{v}$ , let  $D(\mathbf{v}) := \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^\top)$  be the symmetric part of its gradient and  $W(\mathbf{v}) := \frac{1}{2}(\nabla \mathbf{v} - (\nabla \mathbf{v})^\top)$  its skew-symmetric part. Finally,



**Figure 3:** Comparison of the stable heaps formed by (left) the collapse of a granular columns on top of a cylinder with (middle) our model that takes into account the varying volume fraction of grains and (right) the simpler  $\nabla \cdot \mathbf{u} \geq 0$  condition from [Daviet and Bertails-Descoubes 2016].

$\mathbf{u}$  will denote the velocity field of the grains, and we shall use the shorter notation  $\dot{\boldsymbol{\epsilon}} := \text{D}(\mathbf{u})$  for the so-called *strain rate* tensor. Note that  $\text{Tr} \dot{\boldsymbol{\epsilon}} = \nabla \cdot \mathbf{u}$ .

#### 4.1 Continuous equations

Let the adimensional scalar field  $\phi$  denote the local fraction of space occupied by the granular material — that is,  $\phi$  is zero where there are no grains, and  $\phi$  gets closer to 1 in densely packed zones.

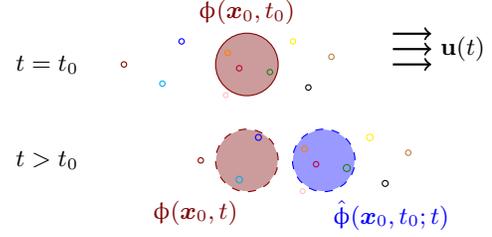
**Frictional contact stress** As we neglect the effect of surrounding air, the total stress of the grain-air mixture with volume fraction of grains  $\phi$  can be expressed as  $\boldsymbol{\sigma}_{tot} = \phi \boldsymbol{\sigma}_g$ , where  $\boldsymbol{\sigma}_g$  is the solid phase stress. As in [Daviet and Bertails-Descoubes 2016], we decompose the solid phase stress tensor as  $\boldsymbol{\sigma}_g := \eta \dot{\boldsymbol{\epsilon}} + \boldsymbol{\sigma}_c$ , where the first part corresponds to a standard Newtonian viscosity (dissipative term due to random collisions in the flowing material), and  $\boldsymbol{\sigma}_c$  is the additional stress due to the Coulomb interactions between individual grains. Note that  $\eta$  may vanish, in this case no internal stress is applied in the “gaseous” phase and we retrieve the constitutive law used in [Narain et al. 2010]. The contact normal stress is  $p = -\frac{1}{3} \text{Tr} \boldsymbol{\sigma}_c$ , such that  $\boldsymbol{\sigma}_c = \text{Dev} \boldsymbol{\sigma}_c - p \mathbf{I}$ . The second stress invariant can be computed through the identity  $J_2 = \frac{1}{2} \text{Tr}(\text{Dev} \boldsymbol{\sigma}_c^2) = |\text{Dev} \boldsymbol{\sigma}_c|^2$ . The Drucker-Prager yield criterion therefore reads  $|\text{Dev} \boldsymbol{\sigma}_c| \leq \hat{\mu} p$ .

Moreover, the maximum dissipation principle states that in the yielded regime, friction should be saturated and the frictional stress tensor should be colinear to the deviatoric part of the strain rate. The deviatoric part of  $\boldsymbol{\sigma}_c$  should thus satisfy one of the two regimes,

$$\begin{cases} \text{Dev} \boldsymbol{\sigma}_c = (\hat{\mu} p) \frac{\text{Dev} \dot{\boldsymbol{\epsilon}}}{|\text{Dev} \dot{\boldsymbol{\epsilon}}|} & \text{if } \text{Dev} \dot{\boldsymbol{\epsilon}} \neq 0 \text{ (yielded)} \\ |\text{Dev} \boldsymbol{\sigma}_c| \leq \hat{\mu} p & \text{if } \text{Dev} \dot{\boldsymbol{\epsilon}} = 0 \text{ (unyielded)}. \end{cases} \quad (1)$$

**Maximum volume fraction constraint** Let  $\phi_{max} \leq 1$  be the maximum packing fraction for the material, beyond which grains cannot be compacted anymore. For monodisperse spherical grains,  $\phi_{max}$  usually lies in the 0.55–0.65 range [Andreotti et al. 2011]. We used  $\phi_{max} = 0.6$  in all our simulations.

Note that Daviet and Bertails-Descoubes [2016] do not consider a volume fraction field. Instead, the fluid is considered to be dense — that is, already at its maximal volume fraction  $\phi_{max}$  — everywhere inside the simulation domain. The material is theoretically allowed to dilate, but not to compress, through the complementarity condition  $0 \leq \nabla \cdot \mathbf{u} \perp p \geq 0$ . However, in [Daviet and Bertails-Descoubes 2016], the use of a fixed domain precludes an actual dilatation of the material through time. More fundamentally, even if the volume fraction field were somehow tracked through time (e.g. with particles), the  $\nabla \cdot \mathbf{u} \geq 0$  condition would still prevent the



**Figure 4:** Eulerian (in red) and Lagrangian (in blue) points of view for defining the volume fraction of a compressible fluid.

material from recompacting after dilatating. As illustrated in Figure 3, sparse grain layers would remain stuck in mid-air instead of falling back all the way down to the ground. For this latter reason, we follow [Narain et al. 2010] to account for a non-constant  $\phi$ , and instead enforce the complementarity condition<sup>2</sup>

$$0 \leq \phi_{max} - \phi \perp p \geq 0. \quad (2)$$

Considering conditions (1) and (2), notice that our rheology boils down to the trivial case  $\boldsymbol{\sigma}_c = 0$  if  $\phi \neq \phi_{max}$ , meaning that contact (and thus friction) only applies at maximum compaction.

**Conservation equations** We consider that the only external force acting on the material is gravity. The mass and momentum conservation equations read

$$\frac{\partial \phi}{\partial t} + \nabla \cdot [\phi \mathbf{u}] = 0, \quad (3)$$

$$\rho \phi \frac{D\mathbf{u}}{Dt} - \nabla \cdot \underbrace{[\phi \eta \dot{\boldsymbol{\epsilon}} + \phi \boldsymbol{\sigma}_c]}_{\boldsymbol{\sigma}_{tot}} = \rho \phi \mathbf{g}. \quad (4)$$

For brevity of exposition, we consider homogeneous Dirichlet boundary conditions on the sides of our simulation domain  $\Omega$ , though other (e.g., no-slip) conditions could be derived. In Section 6.1, we also present a frictional boundary condition which can be used to couple our granular material with simulated rigid bodies.

#### 4.2 Reformulation of the rheology

We show here how Equations (1–2) can be recast into a compact problem with a similar structure to that of [Daviet and Bertails-Descoubes 2016]’s rheology, allowing us to leverage their numerical tools in the sequel. The key insight is to derive a first-order-in-time approximation of the vector field  $\phi$ , allowing us to express Constraint (2) at a given instant on the strain rate tensor  $\dot{\boldsymbol{\epsilon}}$ .

**Instantaneous volume fraction constraint** The mass conservation equation (3) describes how the volume fraction changes in time. Yet, as depicted in Figure 4, there are two ways to express this change. On the one hand, the Eulerian point-of-view looks at the change of  $\phi$  at a fixed location in space  $\mathbf{x}_0$ , which corresponds to the partial derivative  $\frac{\partial \phi}{\partial t}$  evaluated at  $\mathbf{x}_0$ . We have

$$\begin{aligned} \phi(\mathbf{x}_0, t) &= \phi(\mathbf{x}_0, t_0) + \int_{t_0}^t \frac{\partial \phi}{\partial t}(\mathbf{x}_0, s) ds \\ &= \phi(\mathbf{x}_0, t_0) - \int_{t_0}^t \nabla \cdot [\phi \mathbf{u}](\mathbf{x}_0, s) ds, \quad \text{using (3)}. \end{aligned}$$

<sup>2</sup>Narain et al. [2010] actually use a scaled volume fraction field (so that  $0 \leq \phi \leq 1$ ), and write the conservation of momentum using the mixture’s maximum volumic mass instead of the grains volumic mass. These minor changes only affect notation, and lead to the same equations as ours.

On the other hand, the Lagrangian point-of-view follows grains as they move through space. The volume fraction at such a tracked point is given by  $\hat{\phi}(\mathbf{x}_0, t_0; t) := \phi(X(\mathbf{x}_0, t_0; t), t)$ , where  $X(\mathbf{x}_0, t_0; t)$  is the *characteristics* function that gives at every instant  $t$  the position of the point that was at  $\mathbf{x}_0$  at time  $t_0$ . Chain rule w.r.t. time gives  $\frac{\partial \hat{\phi}}{\partial t}(\mathbf{x}_0, t_0; t) = \frac{D\phi}{Dt}(X(\mathbf{x}_0, t_0; t), t)$ , with

$$\begin{aligned} \frac{D\phi}{Dt} &:= \frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi \\ &= -\phi \nabla \cdot \mathbf{u}, \quad \text{using (3) and merging.} \end{aligned} \quad (5)$$

We then have

$$\begin{aligned} \hat{\phi}(\mathbf{x}_0, t_0; t) &= \hat{\phi}(\mathbf{x}_0, t_0; t_0) + \int_{t_0}^t \frac{D\phi}{Dt}(X(\mathbf{x}_0, t_0; s), s) ds \\ &= \phi(\mathbf{x}_0, t_0) - \int_{t_0}^t (\phi \nabla \cdot \mathbf{u})(X(\mathbf{x}_0, t_0; s), s) ds, \end{aligned} \quad (6)$$

using (5).

Constraint (2) can be expressed either on  $\phi$  or on  $\hat{\phi}$ . Studying the rate of change eventually leads to conditions on  $\nabla \cdot [\phi \mathbf{u}]$  or  $\phi \nabla \cdot \mathbf{u}$ , respectively. Both conditions are equivalent in the spatially continuous case, however they yield different discretizations. Still, at the grain scale the contact forces oppose the relative velocity of the particles, which motivates setting that the macroscopic pressure should oppose a function of the divergence of the velocity field.

With the Lagrangian framework of Equation (6) and anticipating on the time discretization (Section 4.3), we linearize the constraint  $\hat{\phi} \leq \phi_{max}$  on a time interval  $\Delta_t$  as  $(\phi(t) - \Delta_t \phi \nabla \cdot \mathbf{u}) \leq \phi_{max}$ . Equation (2) can thus be approximated at the first order as

$$0 \leq \phi(t) \nabla \cdot \mathbf{u} + \beta(\Delta_t; t) \perp p \geq 0,$$

where  $\beta(\Delta_t; t) := \frac{\phi_{max} - \phi(t)}{\Delta_t}$  is a scalar field expressing the maximum rate at which the material can compress during  $\Delta_t$ .

**Compact expression of the full rheology** We now perform a few mathematical manipulations that will reveal a problem structure similar to that of [Daviet and Bertails-Descoubes 2016].

Since the contact stress  $\sigma_c$  trivially vanishes when the material is not at maximum compaction, and in particular where there is no granular matter ( $\phi = 0$ ), one may rewrite (1) equivalently as

$$\begin{cases} \text{Dev } \sigma_c = (\hat{\mu} p) \frac{\phi \text{Dev } \dot{\boldsymbol{\epsilon}}}{|\phi \text{Dev } \dot{\boldsymbol{\epsilon}}|} & \text{if } \phi \text{Dev } \dot{\boldsymbol{\epsilon}} \neq 0 \\ |\text{Dev } \sigma_c| \leq \hat{\mu} p & \text{if } \phi \text{Dev } \dot{\boldsymbol{\epsilon}} = 0. \end{cases}$$

We now consider the friction coefficient  $\mu := \sqrt{\frac{2}{3}} \hat{\mu}$ , and the symmetric tensor fields  $\boldsymbol{\lambda} := -\sigma_c$  and  $\boldsymbol{\gamma} := \phi \dot{\boldsymbol{\epsilon}} + \frac{p}{3} \mathbf{I}$ . We have  $\phi \nabla \cdot \mathbf{u} + \beta = \text{Tr } \boldsymbol{\gamma}$ ,  $\phi \text{Dev } \dot{\boldsymbol{\epsilon}} = \text{Dev } (\boldsymbol{\gamma})$ , so that our whole rheology rereads

$$\begin{cases} \text{Dev } \boldsymbol{\lambda} = -\mu \frac{\text{Tr } \boldsymbol{\lambda}}{\sqrt{6}} \frac{\text{Dev } \boldsymbol{\gamma}}{|\text{Dev } \boldsymbol{\gamma}|} & \text{if } \text{Dev } \boldsymbol{\gamma} \neq 0 \\ |\text{Dev } \boldsymbol{\lambda}| \leq \mu \frac{\text{Tr } \boldsymbol{\lambda}}{\sqrt{6}} & \text{if } \text{Dev } \boldsymbol{\gamma} = 0 \\ 0 \leq \text{Tr } \boldsymbol{\gamma} \perp \text{Tr } \boldsymbol{\lambda} \geq 0. \end{cases} \quad (7)$$

We have thus obtained a rheology that is very similar to the one of [Daviet and Bertails-Descoubes 2016], the notable difference being that  $\dot{\boldsymbol{\epsilon}}$  has been replaced with  $\boldsymbol{\gamma} = \phi \dot{\boldsymbol{\epsilon}} + \frac{p}{3} \mathbf{I}$ . Using their

notation, we will denote by  $\mathcal{DP}(\mu) \subset S_d \times S_d$  the set of pairs of symmetric tensors  $(\boldsymbol{\lambda}, \boldsymbol{\gamma})$  that satisfy Problem (7).

In the supplemental document, Section 1, we prove that the full problem (3-4) combined with our  $\mathcal{DP}(\mu)$  rheology is, as desired, a dissipative system.

### 4.3 Semi-implicit time integration

**Overview** We consider a finite time-step, from  $t$  to  $t + \Delta_t$ . We use a first-order semi-implicit Euler integration scheme, where the volume fraction  $\phi$  is analogous to the position variable in Lagrangian dynamics, and  $\mathbf{u}$  to the velocity variable. More specifically,

1. We compute the end-of-step velocities  $\mathbf{u}(t + \Delta_t)$  and stresses  $\boldsymbol{\lambda}(t + \Delta_t)$  from the discrete-time momentum balance (9) (see below) using the current positions  $\phi(t)$  and with the implicit constraint that the  $\mathcal{DP}(\mu)$  rheology should be satisfied at the end of the timestep. That is, we ensure  $(\boldsymbol{\lambda}(t + \Delta_t); \boldsymbol{\gamma}(t + \Delta_t)) \in \mathcal{DP}(\mu)$ , with  $\boldsymbol{\gamma}$  computed as

$$\boldsymbol{\gamma}(t + \Delta_t) := \phi(t) \text{D}(\mathbf{u})(t + \Delta_t) + \frac{\beta(\Delta_t; t)}{3} \mathbf{I}. \quad (8)$$

2.  $\phi(t + \Delta_t)$  is deduced by moving the particles using the end-of-step velocities  $\mathbf{u}(t + \Delta_t)$ .

A detailed version of this algorithm is given in Section 5.3.

**Discrete-time momentum balance** As we are using a particle-based representation of our material, it is natural to use our Lagrangian framework to compute the material derivative  $\frac{D\mathbf{u}}{Dt}$ . With a first-order time discretization,

$$\frac{D\mathbf{u}}{Dt} \sim \frac{\mathbf{u}(t + \Delta_t) - \mathbf{u}^{p \rightarrow g}}{\Delta_t}$$

where  $\mathbf{u}^{p \rightarrow g}$  corresponds to the velocity recovered from the particles at the start of the timestep. Our discrete-time momentum balance reads

$$\begin{aligned} \frac{\rho \phi(t)}{\Delta_t} \mathbf{u}(t + \Delta_t) + \nabla \cdot [\phi(t) (\boldsymbol{\lambda} - \eta \dot{\boldsymbol{\epsilon}})(t + \Delta_t)] \\ = \rho \phi(t) \left( \mathbf{g} + \frac{\mathbf{u}^{p \rightarrow g}}{\Delta_t} \right). \end{aligned} \quad (9)$$

## 5 Material Point Method

The basic idea behind MPM is to consider that the whole mass of the material is condensed at a finite number of material points. The volume fraction field  $\phi$ , now seen as a mathematical distribution, can therefore be expressed as  $\phi(\mathbf{x}, t) = \sum_p V_p \delta(\mathbf{x} - \mathbf{x}_p(t))$ , where  $\delta$  is the Dirac distribution, and  $V_p, \mathbf{x}_p$  are the volume and position of the  $p^{\text{th}}$  particle, respectively. The total volume of granular reads  $V_{tot} = \int_{\Omega} \phi(t) = \sum_p V_p$ .

This representation is particularly adapted to the FEM formalism, where quantities are evaluated in weak-form, by multiplying them by a test function and integrating over a domain. For instance, for  $\mathbf{v}$  integrable over  $\Omega$  we have  $\int_{\Omega} \phi(t) \mathbf{v} = \sum_p V_p \mathbf{v}(\mathbf{x}_p(t))$ . In this sense, the Material Point Method can be seen as a quadrature rule for which the quadrature points are given by the particles positions, and the corresponding weights by the particles volumes.

First, we present in a general manner how this method can be applied to cast our discrete-time equations into a standard Coulomb friction problem. In a second part, we specify our choices of implementation regarding some technical points, such as grid-particle transfers.

## 5.1 Variational formulation

As in standard FEM, let us first expose a variational formulation of our problem. Let  $\mathcal{U} \subset (\mathcal{H}^1(\Omega))^3$  be a space of square-integrable vector fields on  $\Omega$  with square-integrable derivatives, and  $\mathcal{V}(0) \subset \mathcal{U}$  the subspace satisfying the homogeneous Dirichlet conditions. Let  $\mathcal{T} \subset (L^2(\Omega))^6$  be a space of square-integrable symmetric tensor fields on  $\Omega$ .

The variational formulation of our discrete-time momentum balance (9) subject to the  $\mathcal{DP}(\mu)$  rheology (7) at timestep  $n$  reads

Find  $\mathbf{u} \in \mathcal{V}(0)$ ,  $\boldsymbol{\lambda} \in \mathcal{T}$ ,  $\boldsymbol{\gamma} \in \mathcal{T}$  such that

$$m(\mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) = b(\boldsymbol{\lambda}, \mathbf{v}) + l(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{V}(0) \quad (10)$$

$$s(\boldsymbol{\gamma}, \boldsymbol{\tau}) = b(\boldsymbol{\tau}, \mathbf{u}) + k(\boldsymbol{\tau}) \quad \forall \boldsymbol{\tau} \in \mathcal{T} \quad (11)$$

$$(\boldsymbol{\lambda}; \boldsymbol{\gamma}) \in \mathcal{DP}(\mu) \quad (12)$$

with the bilinear and linear forms

$$m(\mathbf{u}, \mathbf{v}) := \sum_p \frac{\rho}{\Delta t} V_p (\mathbf{u}(\mathbf{x}_p^n) \cdot \mathbf{v}(\mathbf{x}_p^n))$$

$$a(\mathbf{u}, \mathbf{v}) := \sum_p V_p \eta D(\mathbf{u})(\mathbf{x}_p^n) : D(\mathbf{v})(\mathbf{x}_p^n)$$

$$b(\boldsymbol{\tau}, \mathbf{u}) := \sum_p V_p \boldsymbol{\tau}(\mathbf{x}_p^n) : D(\mathbf{u})(\mathbf{x}_p^n)$$

$$s(\boldsymbol{\gamma}, \boldsymbol{\tau}) := \int_{\Omega} \boldsymbol{\gamma} : \boldsymbol{\tau}$$

$$l(\mathbf{v}) := \sum_p \rho V_p \left( \left( \frac{\mathbf{u}^{p \rightarrow g}}{\Delta t} + \mathbf{g} \right) \cdot \mathbf{v}(\mathbf{x}_p^n) \right)$$

$$k(\boldsymbol{\tau}) := \int_{\Omega} \frac{\phi_{max}}{3\Delta t} \mathbf{I} : \boldsymbol{\tau} - \sum_p \frac{V_p}{3\Delta t} \mathbf{I} : \boldsymbol{\tau}(\mathbf{x}_p)$$

and where  $\mathbf{x}_p^n$  denotes the position of the  $p^{\text{th}}$  at timestep  $n$ . Those derivations are detailed in supplemental document, Section 2.

## 5.2 Spatial discretization

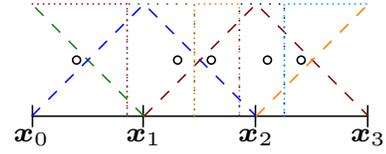
As usual with FEM, we have to choose a finite number of basis functions for discretizing our vector and tensor fields spaces. We use the notation of [Stomakhin et al. 2013] and denote by  $(N_i^v)$  and  $(N_p^t)$  these sets of basis functions for  $\mathcal{V}(0)$  and  $\mathcal{T}$ , respectively.

Note that for any symmetric tensor  $\boldsymbol{\tau}$ ,  $\boldsymbol{\tau} : \dot{\boldsymbol{\epsilon}}(\mathbf{x}_p^n) = \boldsymbol{\tau} : (\nabla \mathbf{u})(\mathbf{x}_p^n)$ , therefore  $b(\boldsymbol{\lambda}, \mathbf{v}_i) = \sum_p V_p \boldsymbol{\lambda}(\mathbf{x}_p^n) : \nabla N_i^v(\mathbf{x}_p^n)$ . We retrieve the expression of the nodal forces from [Stomakhin et al. 2013, Eq. (6)].

While the possibilities are endless, we choose for boths field spaces one of the simplest possible basis, that is trilinear functions on a regular grid. We discuss this choice below, though we will continue to present our method in a general framework. Exploring other discretizations is among our priorities for future work.

**Finite basis for vector fields** While one could discretize directly the subspace of velocities for which the boundary conditions are satisfied  $\mathcal{V}(0)$ , we choose instead to discretize the whole space  $\mathcal{U}$ , and enforce *a posteriori* the boundary conditions through a projection operator (see Section 5.2 below).

A popular choice in the MPM literature is to define  $(N_i^v)$  as diadic products of cubic splines [Stomakhin et al. 2013]. We choose instead products of linear functions, which yields more compact basis functions and therefore sparser matrices. A potential drawback of this approach is the discontinuity of  $\boldsymbol{\epsilon}$  at the grid elements faces, however we did not find this to be problematic in practice.



**Figure 5:** Comparison in 1D of our choice of grid-aligned stress basis function  $N_j^t$  (dashed) with the classical MPM particle-aligned basis functions  $N_p^t$  (dotted).

**Finite basis for tensor fields** Most MPM implementations map stresses to individual particles, which amounts to defining a set of  $p$  basis function  $(N_p^t)$  such that  $N_p^t(\mathbf{x}_p^n) = \delta_p^q$ . We do not retain this solution, because as we shall see in the following section, the cardinality of the set  $(N_p^t)$  determines the number of constraints in our system and therefore the computational time required to solve it. Since our goal is to handle a very high number of particles, the cost of having one basis function per particle could become prohibitive.

Instead we use once again trilinear interpolation functions on the same regular grid, that is, we take  $(N_j^t) = (N_i^v)$  (see Figure 5). Discontinuous Galerkin approximations would also make good candidates, yet increasing compacty at the cost of a higher cardinality of  $(N_j^t)$ .

**Decomposition of symmetric tensors** Thanks to our newly chosen basis functions, we can now express our vector and tensor fields as interpolated functions of a discrete number of values at nodes. However, we want to express our quantities as vectors of real coefficients, and therefore we need to choose basis for  $\mathbb{R}^3$  and  $S_d$  in which to decompose those nodal values. For  $\mathbb{R}^3$ , the canonical basis  $(\mathbf{e}_r^3)$  of  $\mathbb{R}^3$  is a natural choice. For  $S_d$  however, no obvious candidate comes up. In order to simplify the expression of our rheology constraint at the grid nodes  $(\boldsymbol{\lambda}(\mathbf{x}_i); \boldsymbol{\gamma}(\mathbf{x}_i)) \in \mathcal{DP}(\mu)$ , we choose to decompose our symmetric tensors on the basis  $(\chi(\mathbf{e}_s^6))$ , where  $\chi : \mathbb{R}^6 \rightarrow S_d$  is the isomorphism presented in [Daviet and Bertails-Descoubes 2016],

$$\chi : (a, b, c, d, e, f) \mapsto \begin{pmatrix} \frac{\sqrt{2a-c}}{\sqrt{3}} + b & d & e \\ d & \frac{\sqrt{2a-c}}{\sqrt{3}} - b & f \\ e & f & \frac{\sqrt{2a+2c}}{\sqrt{3}} \end{pmatrix}.$$

The isomorphism  $\chi$  is orthonormal for the euclidean spaces  $(\mathbb{R}^6, \|\cdot\|_2)$  and  $(S_d, |\cdot|)$ , i.e.,  $\forall(\mathbf{r}, \mathbf{u}) \in \mathbb{R}^6 \times \mathbb{R}^6$ ,  $\mathbf{r}^T \mathbf{u} = \chi(\mathbf{r}) : \chi(\mathbf{u})$ . In particular, for  $\mathbf{r} = (r_N; \mathbf{r}_T) \in \mathbb{R} \times \mathbb{R}^5$ , we have  $\sqrt{6} r_N = \text{Tr} \chi(\mathbf{r})$  and  $\|\mathbf{r}_T\|_2 = |\text{Dev} \chi(\mathbf{r})|$ .

Using these identities with the notation  $(\boldsymbol{\lambda}_N; \boldsymbol{\lambda}_T) := \chi^{-1}(\boldsymbol{\lambda})$ ,  $(\boldsymbol{\gamma}_N; \boldsymbol{\gamma}_T) := \chi^{-1}(\boldsymbol{\gamma})$ , the tensorial inclusion  $(\boldsymbol{\lambda}, \boldsymbol{\gamma}) \in \mathcal{DP}(\mu)$  can be expressed equivalently through a vector-valued law,

$$\begin{cases} \boldsymbol{\lambda}_T = -\mu \boldsymbol{\lambda}_N \frac{\boldsymbol{\gamma}_T}{\|\boldsymbol{\gamma}_T\|_2} & \text{if } \boldsymbol{\gamma}_T \neq 0 \\ \|\boldsymbol{\lambda}_T\|_2 \leq \mu \boldsymbol{\lambda}_N & \text{if } \boldsymbol{\gamma}_T = 0 \\ 0 \leq \boldsymbol{\gamma}_N \perp \boldsymbol{\lambda}_N \geq 0. \end{cases} \quad (13)$$

With simple algebraic manipulations, one can identify System (13) as the standard Coulomb-Signorini conditions arising in contact problems of multibody dynamics, the sole difference being that the tangential components lie in  $\mathbb{R}^5$  instead of the usual  $\mathbb{R}^2$ . We can thus use any complementarity function defined for Coulomb friction problems to express our rheology as a root-finding problem on  $\mathbb{R}^6$ . For instance, for  $\mathbf{u}, \mathbf{r}$  in  $\mathbb{R}^6 \times \mathbb{R}^6$ , we have

$$f_{\text{FB}}(\mathbf{r}, \mathbf{u}) = 0 \iff (\chi(\mathbf{r}), \chi(\mathbf{u})) \in \mathcal{DP}(\mu)$$

where  $f_{\text{FB}} : \mathbb{R}^6 \times \mathbb{R}^6 \rightarrow \mathbb{R}^6$  is the Modified Fischer-Burmeister function introduced in [Daviet et al. 2011].

With  $q$  the cardinality of  $(N_j^\tau)$  (i.e., the number of grid nodes), let  $\underline{\lambda}$  and  $\underline{\gamma}$  in  $\mathbb{R}^{6q}$  denote the vectors of coefficients corresponding to the decomposition of  $\lambda$  and  $\gamma$  in our chosen bases  $(N_j^\tau; \chi(e_s^6))$ . We can now express that our rheology should be satisfied at each interpolation node through the equivalence

$$f_{\text{FB}}(\underline{\lambda}, \underline{\gamma}) = 0 \iff (\lambda(\mathbf{x}_i); \gamma(\mathbf{x}_i)) \in \mathcal{DP}(\mu) \quad \forall 1 \leq i \leq q,$$

where  $f_{\text{FB}}$  is extended from  $\mathbb{R}^6$  to  $\mathbb{R}^{6q}$  by concatenation.

**Discrete problem** The discrete problem results from the assembly of the real matrices and vectors corresponding to the decomposition of the bilinear and linear forms  $a, b, m$  and  $l, k$  on the bases  $(N_i^\nu; e_r^3)$  and  $(N_j^\tau; \chi(e_s^6))$ . We can construct those matrices  $A, B, M$  and the coefficient vectors  $\underline{l}$  and  $\underline{k}$  as

$$\begin{aligned} M_{3(i-1)+r, 3(j-1)+s} &= m(N_i^\nu e_r^3, N_j^\nu e_s^3) \\ A_{3(i-1)+r, 3(j-1)+s} &= a(N_i^\nu e_r^3, N_j^\nu e_s^3) \\ B_{6(i-1)+r, 3(j-1)+s} &= b(N_i^\tau \chi(e_r^6), N_j^\nu e_s^3) \\ \underline{l}_{3(i-1)+r} &= l(N_i^\nu e_r^3) \\ \underline{k}_{6(i-1)+r} &= \max(0, k(N_i^\tau \chi(e_r^6))). \end{aligned}$$

Enforcing positive  $\underline{k}$  coefficients amounts to clamping the value of the volume fraction to  $\phi_{\text{max}}$ . This preserves the dissipativity property, but may lead to a loss of volume over time (see Section 7.5).

Note that we do not assemble the matrix corresponding to the bilinear form  $s$ . Indeed, as mentioned in [Daviet and Bertails-Descoubes 2016],  $s$  may introduce an unphysical asymmetry in the system. Fortunately, when using a trapezoidal rule to approximate this integral,  $s$  reduces to a node-by-node scaling. Since the constraint  $f_{\text{FB}}(\underline{\lambda}, \underline{\gamma}) = 0$  is invariant to a strictly positive scaling factor on  $\underline{\gamma}$ , we can safely discretize  $s$  as the identity matrix.

Let  $P_v$  be the matrix projecting a vector of velocity coefficients onto the Dirichlet boundary conditions. Our problem amounts to finding the coefficient vectors  $\underline{u}, \underline{\lambda}, \underline{\gamma}$  such that

$$\begin{aligned} (P_v(M + A) + (\mathbf{I} - P_v))\underline{u} &= P_v B^\top \underline{\lambda} + P_v \underline{l} \\ \underline{\gamma} &= B P_v \underline{u} + \underline{k} \\ \mathbf{0} &= f_{\text{FB}}(\underline{\lambda}, \underline{\gamma}), \end{aligned} \quad (14)$$

which is a standard Discrete Coulomb Friction Problem (DCFP), for which numerous algorithms exist (see 5.3) — though the constraints are 6-dimensional, instead of the usual 2D or 3D. Moreover, the DCFP (14) satisfies the criterion from [Acary et al. 2011] which guarantees the existence of a solution.

### 5.3 Implementation

The code that was used to generate all the results presented in this article is released as open-source alongside this document.

**Transferring particles velocities to the grid** The natural FEM way to obtain the grid velocities  $\mathbf{u}^{p \rightarrow g}$  from the particles at the beginning of each time step would be to solve the linear system

$$m(\mathbf{u}^{p \rightarrow g}, \mathbf{v}) = \sum_p V_p \mathbf{v}_p^n \mathbf{v}(\mathbf{x}_p^n), \quad \forall \mathbf{v} \in \mathcal{V}(0) \quad (15)$$

where  $\mathbf{v}_p^n$  is the velocity of the  $p^{\text{th}}$  particle at the beginning of the timestep.

However, when there are not enough particles in a non-empty simulation cell,  $m$  may become singular — thinking of them as quadrature points, we need enough particles for the integral to be well-posed. A popular alternative is to replace the  $m$  bilinear form with the so called *lumped* (or diagonalized) mass bilinear form  $\tilde{m}$  [Wieckowski et al. 1999]. Its matrix  $\tilde{M}$  is diagonal with coefficients

$$\tilde{M}_{3(i-1)+r, 3(i-1)+r} := \sum_p V_p N_i^\nu(\mathbf{x}_p^n). \quad (16)$$

When considering only nodes belonging to non-empty cells,  $\tilde{M}$  is always positive-definite, and the lumped version of (15) is trivial to solve. However, this approach may lead to a considerable loss of kinetic energy. Several methods have attempted to reduce this energy loss while retaining the robustness of the lumped-mass approach, one of the most famous being FLIP [Zhu and Bridson 2005]. We choose to use the recently proposed APIC [Jiang et al. 2015], which provides increased stability over FLIP. With our trilinear basis functions  $(N_i^\nu)$ , this amounts to storing information about the local velocity gradient on each particle before advecting them at the end of each time-step, then propagating it back to the right-hand-side of the velocity transfer equation (15). That is, we set

$$\begin{aligned} \underline{\mathbf{u}}^{p \rightarrow g} &= \tilde{M}^{-1} \underline{\mathbf{v}}^{\text{APIC}} \\ \underline{\mathbf{v}}_{3(i-1)+k}^{\text{APIC}} &= \sum_p V_p (\mathbf{v}_{p,k}^n N_i^\nu(\mathbf{x}_p^n) + \mathbf{c}_{p,k}^n (\mathbf{x}_i - \mathbf{x}_p^n)) \end{aligned} \quad (17)$$

where  $\mathbf{x}_i$  denotes the position of grid node  $i$  and  $\mathbf{c}_{p,k}$  the  $k^{\text{th}}$  row of  $\mathbf{c}_p^n$ , the  $3 \times 3$  velocity gradient matrix computed at the end of the previous step as

$$\mathbf{c}_p = (\nabla \mathbf{u})(t^n, \mathbf{x}_p^n) = \sum_i \mathbf{u}(t^n, \mathbf{x}_i) (\nabla N_i^\nu)(\mathbf{x}_p^n). \quad (19)$$

See [Jiang et al. 2015, Sections 5 and 6] for more details.

**Solving the Discrete Coulomb Friction Problem** The generalized stiffness matrix of the lumped system (14),  $\tilde{A} := (P_v(\tilde{M} + A) + (\mathbf{I} - P_v))$ , is sparse, but has a dense inverse for  $\eta > 0$ . In 2D, Daviet and Bertails-Descoubes [2016] solve a similar DCFP, which possesses also a dense inverse, using the Cadoux algorithm [Acary et al. 2011] and inner solvers that do not require the explicit evaluation of  $\tilde{A}^{-1}$ . However, their approach is mentioned to be computationally expensive, and likely intractable in 3D. Here, we avoid this problem by making the assumption that the Newtonian viscosity  $\eta$  is very small<sup>3</sup>, and that  $\tilde{A}$  is thus close to  $P_v \tilde{M}$ . We proceed in two steps as described hereafter.

We first solve the unconstrained momentum balance (20) using a conjugate-gradient algorithm,

$$\tilde{A} \underline{\mathbf{u}}^* = P_v \underline{\mathbf{l}}. \quad (20)$$

The frictional response of the material is then computed by neglecting the effect of the change in Newtonian stress due to the addition of contact stresses. That is, we write  $\underline{\mathbf{u}} = \underline{\mathbf{u}}^* + \underline{\Delta \mathbf{u}}$ , and solve a new DCFP: Find  $\underline{\Delta \mathbf{u}}, \underline{\lambda}, \underline{\gamma}$  s.t.

$$\begin{aligned} \underline{\Delta \mathbf{u}} &= \tilde{M}^{-1} P_v B^\top \underline{\lambda} \\ \underline{\gamma} &= B P_v \underline{\Delta \mathbf{u}} + B P_v \underline{\mathbf{u}}^* + \underline{\mathbf{k}} \\ \mathbf{0} &= f_{\text{FB}}(\underline{\lambda}, \underline{\gamma}). \end{aligned} \quad (21)$$

<sup>3</sup>Daviet and Bertails-Descoubes [2016] observed that for high values of  $\eta$  (low Reynolds number), the material behaves like a standard Newtonian fluid, losing its granular properties. In practice we use  $\eta = 1$  mPa.s.

Doing so, we fully preserve the impact of the Newtonian viscosity in freely flowing zones, while neglecting its effect within zones that are dominated by static friction. Our DCFP’s stiffness matrix is now diagonal, positive-definite and therefore trivial to inverse. This positively impacts the performance in two ways:

- We get a significant speedup over [Daviet and Bertails-Descoubes 2016] using the same algorithms, as the linear system solve at each iteration of the inner problem is now replaced with a single multiplication by a diagonal matrix.
- The Delassus operator of (21) is now sparse and easy to assemble, which opens the way to a wider range of algorithms.

We leverage this last remark and use the Gauss-Seidel algorithm from [Daviet et al. 2011] implemented in the open-source library So-bogus<sup>4</sup>. Note that this library does not provide a polynomial root-finding local solver for Coulomb friction in dimension 6, therefore we cannot use the hybrid algorithm described in [Daviet et al. 2011], but rather the pure Newton method on the  $f_{\text{FB}}$  function, without failsafe. However, thanks to the regularity and well-posedness of our grid-based problem, in practice we have not encountered the local-solver failures reported in the case of hair systems.

The So-bogus library also provides DCFP solvers based on the Cadoux algorithm [Acary et al. 2011]. We compare the performance of such methods on our 6D problems in Section 7.4.

**Splitting and merging** In order to keep the simulation particles well distributed, we follow the split/merge heuristics of [Narain et al. 2010, Section 3]. To each particle is associated an ellipsoid representing its occupied volume. Such an ellipsoid with semi-axis  $(\mathbf{a}_{p,i})_{i=1,2,3}$  defines a symmetric, positive-definite tensor  $F_p := \sum \mathbf{a}_{p,i} \mathbf{a}_{p,i}^\top$ , whose change in time satisfies

$$\frac{dF_p}{dt} = (\dot{\boldsymbol{\varepsilon}} + \mathbf{W}(\mathbf{u}))(\mathbf{x}_p^n)F_p + F_p(\dot{\boldsymbol{\varepsilon}} - \mathbf{W}(\mathbf{u}))(\mathbf{x}_p^n). \quad (22)$$

Particles are then split if one axis of the ellipsoid becomes much longer than the others. The reverse operation is performed when two close particles can be merged into a more isotropic one.

Still following [Narain et al. 2010], these frames are useful not only at simulation time, but also at the rendering stage. Indeed, they define volumes from which the passively-advected grain samples are not allowed to escape. This ensures that a rendering sample cannot drift to a place where no simulated particle is present, i.e., to a place where the velocity field has no physical meaning.

**Moving particles** Particles are then advected following

$$\mathbf{x}_p^{n+1} = \mathbf{u}(\mathbf{x}_p^n) \quad \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{v}_p^{n+1} \Delta t. \quad (23)$$

**Overview of a time-step** Our full algorithm can be summarized as follows:

1. Compute the lumped mass matrix  $\tilde{M}$  with Eq (16)
2. Recover nodal velocities  $\mathbf{u}^{p \rightarrow g}$  from System (17)
3. Assemble forms matrices  $A$  and  $B$  and vectors  $\mathbf{l}$ ,  $\mathbf{k}$  as in 5.2
4. Solve for unconstrained velocity  $\mathbf{u}^*$  with Eq (20)
5. Solve DCFP (21) and get the total velocity  $\mathbf{u} = \mathbf{u}^* + \Delta \mathbf{u}$
6. Update particles frames with Eq (22), then split or merge them according to Section 5.3
7. Compute the velocity gradient matrix  $\mathbf{c}_p$  using Eq (19)
8. Update particle positions and velocities as per Eq (23)
9. Proceed to next time-step.

The quantities that need to be stored on particles are therefore  $V_p$ ,  $\mathbf{x}_p$ ,  $\mathbf{v}_p$ ,  $\mathbf{c}_p$  and  $F_p$ . We also store the stress field  $\boldsymbol{\lambda}$  between timesteps, in order to warm-start the DCFP solver.

<sup>4</sup><http://gdaviet.fr/code/bogus>



**Figure 6:** A slightly tilted wheel, on which a constant torque is applied, is dropped on a sandy ground. Without wheel-sand friction ( $\mu_{\text{RB}} = 0$ ), the wheel rests down (left). Otherwise ( $\mu_{\text{RB}} > 0$ ), it first rolls for some short distance, before skidding on the ground (right).

## 6 Extensions

### 6.1 Rigid-bodies coupling and boundary conditions

Frictional boundary conditions are imposed through the means of rigid bodies with infinite inertia. Those are represented as a level set  $\phi_{\text{RB}}$ , such that  $\phi_{\text{RB}}(\mathbf{x}) > 1$  inside the rigid body.

**Volume fraction correction** We have to update the expression of the maximum compaction rate  $\beta$  of our material to account for the fraction of space taken by nearby rigid bodies, replacing the term  $\phi_{\text{max}}$  with  $(\phi_{\text{max}} - \phi_{\text{RB}}(t + \Delta t))$ . Linearizing in time, we have  $\phi_{\text{RB}}(t + \Delta t) = \phi_{\text{RB}}(t) + \Delta t \bar{\mathbf{v}}^\top \nabla \phi_{\text{RB}}$ , with  $\bar{\mathbf{v}} := \mathbf{J}(\mathbf{v}_{\text{RB}}) - \mathbf{u}$ , and where  $\mathbf{J}(\mathbf{v}_{\text{RB}})$  represents the velocity field spawned by a rigid body with linear and angular velocities  $\mathbf{v}_{\text{RB}}$  and center of mass  $\mathbf{x}_{\text{RB}}$ ,

$$\mathbf{J}(\mathbf{v}_{\text{RB}})(\mathbf{x}) = \mathbf{J} \begin{pmatrix} \mathbf{v}_{\text{RB}}^{\text{lin}} \\ \mathbf{v}_{\text{RB}}^{\text{ang}} \end{pmatrix} (\mathbf{x}) = \mathbf{v}_{\text{RB}}^{\text{lin}} + \mathbf{v}_{\text{RB}}^{\text{ang}} \times (\mathbf{x} - \mathbf{x}_{\text{RB}}).$$

**Boundary conditions** We consider an additional stress  $-\boldsymbol{\lambda}_{\text{RB}}$  inside the rigid body. The reaction force applied by the granular material through the boundary is given by  $\mathbf{r} = \boldsymbol{\lambda}_{\text{RB}} \mathbf{n}_{\text{RB}}$ , with  $\mathbf{n}_{\text{RB}} = \frac{\nabla \phi_{\text{RB}}}{\|\nabla \phi_{\text{RB}}\|}$ . We want  $\mathbf{r}$  and the relative velocity  $\bar{\mathbf{v}}$  to follow a Coulombic relationship. A convenient way to enforce this in our framework is to add the condition  $(\boldsymbol{\lambda}_{\text{RB}}, \boldsymbol{\gamma}_{\text{RB}}) \in \mathcal{DP}(\mu_{\text{RB}})$ , where  $\boldsymbol{\gamma}_{\text{RB}} := \frac{1}{2}((\bar{\mathbf{v}} \mathbf{n}_{\text{RB}}^\top + \mathbf{n}_{\text{RB}} \bar{\mathbf{v}}^\top))$  and the coefficient  $\mu_{\text{RB}}$  sets the intensity of friction between the granular material and the rigid body (see Figure 6). The rationale behind this constraint is exposed in our supplemental document, Section 3.

In practice, we create alongside our symmetric tensor basis  $(N_j^\top)$  a new set of basis functions  $(N_i^{\tau_{\text{RB}}})$ . We construct the matrices  $B_{\text{RB}}^1$  and  $B_{\text{RB}}^2$  associated to the following bilinear forms,

$$b_{\text{RB}}^1(\boldsymbol{\tau}, \mathbf{u}) := - \int_{\phi_{\text{RB}} > 0} \frac{1}{3} (\mathbf{u}^\top \nabla \phi_{\text{RB}}) \mathbf{I} : \boldsymbol{\tau}$$

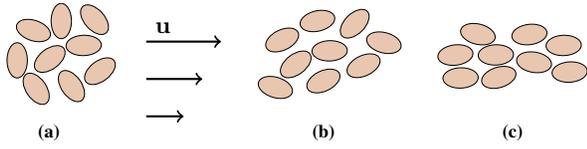
$$b_{\text{RB}}^2(\boldsymbol{\tau}_{\text{RB}}, \mathbf{u}) := - \int_{\phi_{\text{RB}} > 1} ((\nabla \phi_{\text{RB}}) \mathbf{u}^\top) : \boldsymbol{\tau}_{\text{RB}}.$$

We concatenate  $B_{\text{RB}}^1$  and  $B_{\text{RB}}^2$  to obtain the matrix  $B_{\text{RB}}$ . The volume fraction correction is given by the vector  $\mathbf{k}_{\text{RB}}$  associated to

$$k_{\text{RB}}(\boldsymbol{\tau}) := - \frac{1}{3 \Delta t} \int_{\Omega} \phi_{\text{RB}} \mathbf{I} : \boldsymbol{\tau}.$$

Finally, we update the DCFP (21) by extending  $\boldsymbol{\gamma}$  and  $\boldsymbol{\lambda}$  to include our new  $\boldsymbol{\gamma}_{\text{RB}}$  and  $\boldsymbol{\lambda}_{\text{RB}}$ , adding  $B_{\text{RB}}$  to  $B$  and  $(\mathbf{k}_{\text{RB}} = B_{\text{RB}} \mathbf{J} \mathbf{v}_{\text{RB}})$  to  $\mathbf{k}$ ,  $J$  being the matrix associated to the discretization of the operator  $\mathbf{J}$  at each grid node  $\mathbf{x}_i$ .

**Two-way coupling** Since now we know how to compute the force applied by the granular material on the rigid body, we can easily couple their dynamics. Indeed, the generalized force applied by the granular material on the rigid body’s center of mass is given



**Figure 7:** Rationale for our handling of anisotropy. (a) Random orientation of anisotropic particles still results in isotropic macroscopic friction. (b) A shear flow induces a privileged orientation. (c) Aligned particles yield anisotropic macroscopic friction: reduced in the horizontal direction, more intense in the vertical one.

by  $-J^\top B_{\text{RB}}^\top \underline{\lambda}$ . The increment on  $\underline{\gamma}$  corresponding to an increment  $\Delta \mathbf{v}_{\text{RB}}$  of the rigid body 6-velocities is given by  $-B_{\text{RB}} J \Delta \mathbf{v}_{\text{RB}}$ . Deconvolving, we obtain

$$\underline{\gamma} = B P_v \underline{\mathbf{u}} + \underline{\mathbf{k}} + B_{\text{RB}} J M_{\text{RB}}^{-1} J^\top \underline{\lambda}.$$

## 6.2 Anisotropy

Many granular materials are composed of anisotropic grains that are thinner along one direction. This is typically the case for corn flakes, or, more common in the Computer Graphics imaginary, piles of gold coins (see Figure 16). It is noteworthy that anisotropy at the grain scale does play a role on the collective granular behavior. Indeed, while there is no reason to favor any particular direction when the grains are randomly oriented, when all their normals are aligned the macroscopic friction becomes anisotropic: it has much less dissipative effect when grains are sliding on top of each other, rather than when their relative displacement is along their common normal (see Figure 7 for an illustration).

However, remember that each one of our particles does not represent a single grain, but a collection of them, and thus it may include different orientations. Rather than mapping a normal to each particle, we should instead store a probability distribution function (PDF)  $\psi(\mathbf{n})$ . For efficiency purposes we store only its second moment, the symmetric tensor  $\nu_2 = \int_{S^2} \mathbf{n} \mathbf{n}^\top \psi(\mathbf{n})$ . In the following, we construct an heuristic evolution equation for this tensor, then propose a way to model its influence on our  $\mathcal{DP}(\mu)$  rheology.

**Evolution of  $\nu_2$**  The evolution of the orientation of rigid ellipsoids in a shearing flow has been described by Jeffery [1922]. Integrating it over the unit sphere of all possible orientations (see e.g., [Folgar and Tucker 1984]) yields the evolution of the second moment tensor  $\pi_2$  of the orientation PDF,

$$\frac{D\pi_2}{Dt} = W(\mathbf{u})\pi_2 + \pi_2 W(\mathbf{u}) + \ell(\dot{\epsilon}\pi_2 + \pi_2\dot{\epsilon} - 2\dot{\epsilon} : \pi_4) \quad (24)$$

where  $0 \leq \ell \leq 1$  is a coefficient describing the elongation of the ellipsoid,  $\ell = \frac{L/W-1}{L/W+1}$ , and  $\pi_4$  the fourth moment of the orientation PDF. The parameter  $\ell$  affects the tendency of the ellipsoids to align with the flow. Note that this model is only valid for dilute suspensions. Many authors have postulated laws for extending it to higher particle concentrations (e.g., Folgar and Tucker [1984] added an additional dissipative term modeling random collisions between particles in the semi-concentrated regime).

We propose the following algorithm for the evolution of our normal orientation tensor  $\nu_2$ , which yields good enough results for our purposes — for high values of  $\ell$ , flat particles in a shearing flow tend to become parallel to each other, which corresponds to the expected behavior (see Figures 7 and 9).

1. Deduce  $\pi_2(t)$  from  $\nu_2(t)$  as  $\pi_2 = \frac{1}{2}P(\mathbf{I} - D)P^\top$ , where  $P$  and  $D$  are given by the eigen decomposition of  $\nu_2$ .

2. Explicitly compute  $\pi_2(t + \Delta t)$  using Equation (24) and the quadratic approximation of the  $\pi_4$  tensor,  $\pi_4 \sim \pi_2 \otimes \pi_2$  — the outer tensor product of  $\pi_2$  with itself.
3. Deduce  $\nu_2(t + \Delta t)$  from  $\pi_2(t + \Delta t)$ , and normalize it using  $\nu_2 \leftarrow \frac{\nu_2}{\|\nu_2\|_1}$ .

**Alteration of the rheology of  $\mathcal{DP}(\mu)$**  Anisotropy can be included in the rheology by simply replacing the norm  $|\cdot|$  with  $|\cdot|_N = |N \cdot N|$ , where  $N$  is a symmetric tensor. In order for the maximum dissipation principle to remain satisfied,  $\gamma$  should also be replaced with  $N\gamma N$  on the first line of Equation (7).

We choose to deduce the matrix  $N$  from the normal orientation tensor  $\nu_2$  as  $N^{-1} = (1 - \alpha)\mathbf{I} + 3\alpha\nu_2$ , where  $0 \leq \alpha \leq 1$  is a dimensionless coefficient parameterizing the amount of anisotropy in the frictional law. Using this formula, the effective friction coefficient will remain equal to  $\mu$  for isotropic orientations (the three eigenvalues of  $\nu_2$  equal to  $\frac{1}{3}$ ). If all the normals are oriented in the same direction (a single non-zero eigenvalue equal to 1), the effective friction coefficient will be  $(1 + 2\alpha)^2\mu$  in the normal direction, and  $(1 - \alpha)^2\mu$  in the tangential ones.

Instead of modifying our friction solver, we can introduce the linear operator  $\underline{\mathbf{N}} : \underline{\mathbf{N}}\underline{\lambda} \mapsto \frac{\text{Tr}\underline{\lambda}}{3}\mathbf{I} + N \text{Dev } \underline{\lambda}N$ , make the changes of variable  $\tilde{\underline{\lambda}} = N\underline{\lambda}N$ ,  $\tilde{\underline{\gamma}} = N\underline{\gamma}$  and replace the DCFP (21) with

$$\begin{aligned} \underline{\Delta}\mathbf{u} &= \tilde{M}^{-1} P_v B^\top \underline{\mathbf{N}}^{-1} \tilde{\underline{\lambda}} \\ \tilde{\underline{\gamma}} &= \underline{\mathbf{N}}^{-1} B P_v \underline{\Delta}\mathbf{u} + \underline{\mathbf{N}}^{-1} (B P_v \underline{\mathbf{u}}^* + \underline{\mathbf{k}}) \\ \mathbf{0} &= f_{\text{FB}}(\tilde{\underline{\lambda}}, \tilde{\underline{\gamma}}). \end{aligned}$$

## 6.3 Adding further ingredients to the rheology

**Varying friction coefficient  $\mu(I)$**  Jop et al. [2006] showed that granular flows tend to exhibit shear-thickening properties; the effective friction coefficient increases with the inertial number  $I$ ,

$$\mu(I) = \mu + (\mu_d - \mu) \frac{I}{I_0 + I} \quad I = \frac{d_p |\dot{\epsilon}|}{\sqrt{p/\rho}}$$

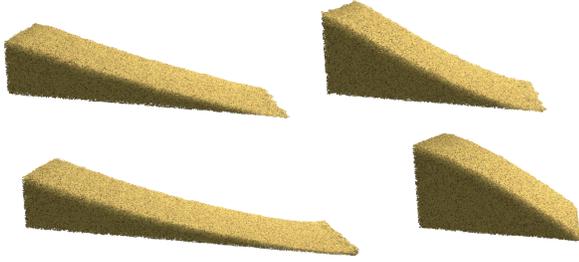
where  $\mu_d$  and  $I_0$  are material-dependent dimensionless coefficients. While the  $\mu(I)$  rheology does not affect the static friction coefficient, and thus the rest angle of the granular material, it influences visibly the dynamics of the flow — and typical features such as the run-out length of a column collapse (see Figure 8). We can implement this variable friction coefficient in an explicit fashion by evaluating the inertial number at each particle before advecting them, then reconstructing an averaged friction coefficient at each node.

**Cohesion** Cohesion fits in very nicely with the continuous model, but creates tremendous discretization issues. On the model side, adding cohesion amounts to making the change of variable  $\tilde{\underline{\lambda}} = \underline{\lambda} + c\mathbf{I}$ , and enforcing the rheology  $(\tilde{\underline{\lambda}}; \underline{\gamma}) \in \mathcal{DP}(\mu)$ . The momentum balance becomes

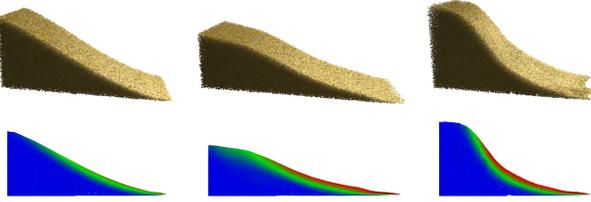
$$\rho\phi \frac{D\mathbf{u}}{Dt} - \nabla \cdot [\phi\eta\dot{\epsilon} - \phi\tilde{\underline{\lambda}}] = \rho\phi\mathbf{g} + \nabla [c\phi].$$

However, preserving the dissipativity of the system imposes  $\phi < \phi_{\text{max}} \implies c = 0$ . This makes physical sense: grains may only be subject to cohesion when they are contacting each other. We can therefore only obtain a stable cohesive medium when the particles are aligned with the underlying stress basis functions ( $N_j^t$ ).

This can still be useful to model initially standing structures, such as in Figure 15. Debonding was modeled using a simple decay model linear in the norm of the strain rate  $|\dot{\epsilon}|$ ,  $\frac{dc}{dt} = -\xi|\dot{\epsilon}|c$ .



**Figure 8:** Collapsed granular columns with different friction coefficients. Top left:  $\mu = 0.26$ ,  $\mu_d = 0.48$ . Bottom left:  $\mu = \mu_d = 0.26$ . Top right:  $\mu = \mu_d = 0.48$ . Bottom right:  $\mu = \mu_d = 0.48$ , high Newtonian viscosity  $\eta$ .



**Figure 9:** Snapshot and corresponding velocity field of the anisotropic column collapse with  $\mu = 0.48$  at  $t = 2.72$ . Left: isotropic friction. Middle:  $\alpha = 0.8$ , with random initial particle normals. Right:  $\alpha = 0.8$ , with upward initial particle normals.

## 7 Results and Discussion

All our simulations and benchmarks were run on 2010 Intel® Xeon® quad-core workstations. Sand scenes were rendered with the algorithm of [Narain et al. 2010], using rasterization instead of ray-tracing. Coins went through the same process, but were shaded with per-sample normals instead of the volume fraction gradient.

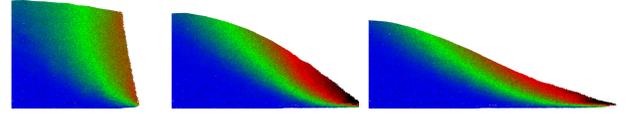
### 7.1 Influence of parameters and visual comparisons

Here we study the influence of several parameters of our model on the collapse of a rectangular column. We refer the reader to the accompanying video for a more comprehensive view of their dynamical effects.

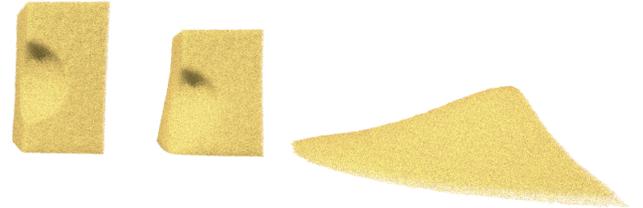
**Influence of friction coefficients** Figure 8 depicts the final, stable states following the collapse for the constant  $\mu$  and  $\mu(I)$  rheologies. While  $\mu$  is directly related to the slope of the final granular heap (higher  $\mu$  implies steeper slope),  $(\mu_d - \mu)$  has an effect on the dynamic regime, and therefore on the horizontal spread of the heap. Cranking up the Newtonian viscosity parameter  $\eta$  also reduces this spread, but gives a muddier look to the simulation.

**Influence of anisotropy** Figure 9 compares the profiles and velocities of the column collapse for different anisotropy settings. The anisotropic collapse with random initial orientations yields a longer run-out, as the particles align with the flow and minimize friction. The distinct frictional responses of the yielded and unyielded zones are even more visible when initial orientations are uniform.

**Comparisons** Continuum simulations using the  $\mu(I)$  rheology were recently validated against DEM simulations [Lagrée et al. 2011] and real experiments [Ionescu et al. 2015] of a 2D granular column collapse. We have reproduced the experiment of [Lagrée et al. 2011, Figure 9]. The friction parameters  $\mu = 0.26$ ,  $\mu_d = 0.48$  of our 3D simulation were chosen so as to match those



**Figure 10:** Orthographic views of collapsing column at dimensionless times  $\bar{t} = 0.66$ , 1.37 and 1.92, to be compared with [Lagrée et al. 2011, Figure 9]. Colors denote particles velocities (blue slowest, black fastest), and can be compared with a similar visualization in [Ionescu et al. 2015, Figure 2].



**Figure 11:** Revisiting [Zhu and Bridson 2005]'s column collapse.

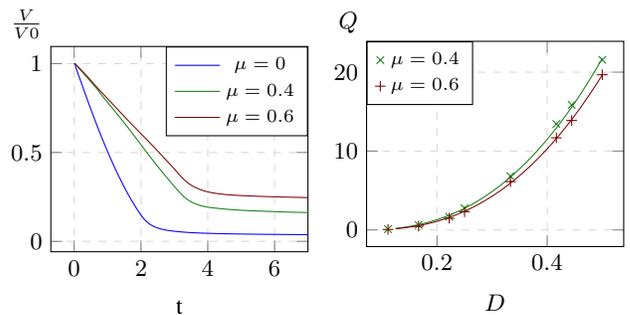
from [Lagrée et al. 2011] using the inscribed Drucker-Prager surface (see Appendix A). Figure 10 shows that we retrieve the correct profiles throughout time. Our final dimensionless height and length also match those from the 2D experiments.

Figure 11 depicts representative frames of our simulation of the sand column collapse scenario introduced in [Zhu and Bridson 2005]. Our results are comparable to those later obtained in [Narain et al. 2010], where the incompressibility constraint was relaxed. However, Figure 2 illustrates that replacing the Frobenius norm of the Drucker-Prager law with  $\ell_\infty$ , as done in [Narain et al. 2010], yields anisotropic artifacts, as the effective friction coefficient then ranges from  $\mu$  to  $\sqrt{5}\mu$  depending on the flow direction.

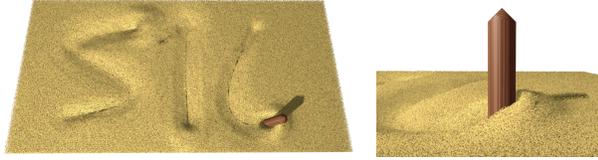
### 7.2 Retrieving the empirical laws of silo discharge

Being ubiquitous, granular silos have been described by macroscopic laws which we attempt to reproduce in our 3D simulations.

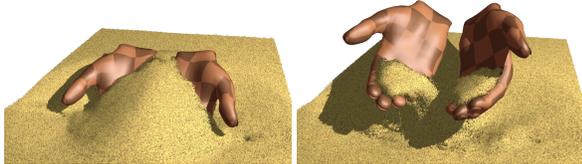
**Discharge curve** The distinctive feature of a granular silo (such as a sand hourglass) is that its discharge rate is linear in time — while for a clepsydra with a Newtonian liquid, the rate depends on the volume of material left inside the upper compartment [Staron et al. 2012]. Figure 12, left, shows that we retrieve this property.



**Figure 12:** Left: Normalized remaining volume over time for different friction coefficients. Right: Evolution of the simulated discharge rate with the outlet width (marks), and Beverloo fits (lines).



**Figure 13:** Letters drawn by dragging a stick in the sand. Right: a typical mound grows at the front of the stick.



**Figure 14:** Picking-up sand and letting it flow away.

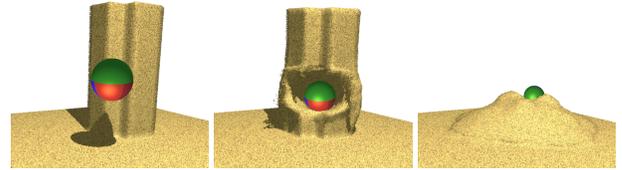
**Beverloo scaling** The Beverloo scaling is a phenomenological law relating the discharge rate  $Q$  of a granular silo to the diameter  $D$  of its outlet as  $Q = C(D - k)^{\frac{5}{2}}$  in 3D, where  $C$  and  $k$  are properties of the material and silo geometry. Figure 12, right, shows that we do retrieve this power  $\frac{5}{2}$  law. However due to our relatively rough discretization, the resulting  $k$  corresponds to about two grid cells – instead of a few grain diameters in reported experiments.

### 7.3 Complex scenarios

**Free-flowing material** We simulated two scenarios in which sand is manipulated through scripted rigid-body motions. The first one, reproduced in Figure 13, features a small cylinder being dragged across the ground. We capture the formation of a typical mound at the front of the cylinder, and the permanent marks that are due to the frictional nature of the material. In the second scenario (Figure 14), a handful of sand is picked up before being let to flow freely, illustrating the transitions between the gaseous, liquid and solid regimes of the material.

**Impacts and cohesion** We ran two simulations reproducing impacts of a fully-coupled rigid sphere on granular media, with and without cohesion. While Narain et al. [2010] were able to capture the dynamics of the impact of a two-inch tungsten ball on a granular bed<sup>5</sup>, their simulation suffered from visible artifacts due to their linearization of the Drucker-Prager law. In contrast, our simulation of the same scenario yields a perfectly round crater (Figure 1), matching much more closely that of their reference video<sup>6</sup>. Figure 15 reproduces the destruction of a granular tower initially standing thanks to cohesion.

**Anisotropy** Scenes featuring large piles of gold coins are common in Computer Graphics — think for example of the daily money swim of *Uncle Scrooge*, *Aladdin* making his way through the cave filled with treasure, or *Smaug* waking up under mounds of gold coins in the *Hobbit* trilogy. Inspired by the latter, we have simulated the Stanford bunny waking up under a large heap of coins, triggering anisotropic avalanches (see Figure 16).



**Figure 15:** A sphere impacting a sand tower initially standing up thanks to a high cohesion coefficient.



**Figure 16:** Treasures made of gold coins are a typical example of highly anisotropic granular materials.

### 7.4 Performance

**Simulation timings** Table 1 provides the size and simulation timings for all our examples. The target framerate (FPS) is dictated by the desired playback speed of our accompanying video — simulation frames were then subdivided using either a fixed number of substeps, or an adaptive criterion based on the fastest particle speed to grid cell size ratio. The tolerance for the DCFP solutions was set to  $10^{-3}$  times the typical stress  $\rho g L$ , and was typically reached within 25 to 250 iterations of the Gauss-Seidel algorithm warm-started with the solution of previous time step.

The computational bottleneck of our method lies in the DCFP solving. Note that with our trilinear stress basis functions, this cost increases with the number of active grid nodes, and has very little dependence on the number of simulated particles. We can therefore use a high numbers of particles per cell, such as in the Silo simulation, without incurring too much overhead.

**Comparisons between Coulomb friction solvers** As mentioned in 5.3, we use the free So-bogus library, which implements a few algorithms for solving DCFP, either tackling it directing using the Gauss-Seidel (GS) algorithm with the MFB local solver from [Daviet et al. 2011], or decomposing it as a sequence of optimization problems using the Cadoux algorithm [Acary et al. 2011]. In the latter case, inner problems can be solved using another GS algorithm, or a few variants of the Projected-Gradient (PG) algorithm, including the APGD algorithm [Mazhar et al. 2015] and a line-search-free implementation of the SPG method [Tasora 2013].

**Table 1:** Sizes and simulation time of our examples

Example	FPS	Grid	$n_P$ <sup>1</sup>	$n_N$ <sup>1</sup>	$\bar{t}$ <sup>2</sup>	% <sub>b</sub>	% <sub>s</sub> <sup>3</sup>
<b>Collapse</b>	96	40×10×20	$5.0 \cdot 10^4$	2755	1.47	19	72
<b>ZB05</b>	96	100×100×50	$5.1 \cdot 10^5$	34112	23.5	21	71
<b>Wheel</b>	96	42×28×14	$6.5 \cdot 10^4$	10152	9.62	4	92
<b>Silo</b>	96	36×36×72	$1.3 \cdot 10^6$	49774	61.5	41	41
<b>Writing</b>	60	96×48×24	$4.9 \cdot 10^5$	61864	48.4	18	65
<b>Cohesion</b>	240	80×80×40	$3.3 \cdot 10^5$	42205	6.8	32	52
<b>Digging</b>	96	64×64×32	$2.2 \cdot 10^6$	75283	32.3	23	65
<b>Crater</b>	600	75×75×50	$8.3 \cdot 10^5$	110887	89.6	19	65
<b>Treasure</b>	96	70×70×50	$3.3 \cdot 10^6$	121143	83.6	27	59

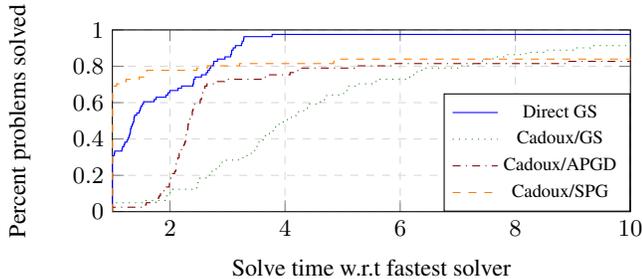
<sup>1</sup> Maximum numbers of particles ( $n_P$ ) and active grid nodes ( $n_N$ )

<sup>2</sup> Average simulation time in seconds per rendering frame  $\bar{t}$

<sup>3</sup> Percentage of time for building and solving the DCFP

<sup>5</sup> <https://youtube.com/watch?v=ZoZ0ZAzr6eg#t=90>

<sup>6</sup> <http://dsc.discovery.com/videos/time-warp-deep-impact>



**Figure 17:** Performance profiles for a variety of solvers. The line  $x = 1$  gives the solvers that were the most often the fastest, while  $y = 1$  shows the most robust ones.

PG Algorithms do not have to explicitly assemble the Delassus matrix, and we indeed found that they performed better when using only an implicit expression of this operator.

Figure 17 shows the performance profiles for those solvers on a variety of problems extracted from our simulations — that is, the percentage of problems solved under a certain multiple of the time taken by the best-performing solver for each problem. To avoid favoring a given algorithm, we used the Alart-Curnier complementarity function to evaluate all the residuals equally. Overall, we found the direct GS to perform the best on our quad-core setup and for our chosen tolerance. Note however that PG algorithms should theoretically scale more easily to a higher number of processors.

**Comparison with [Narain et al. 2010]** We have compared Narain et al. [2010]’s reference code to ours on the same hardware. Using similar grid size, number of particles, and physical parameters, we have found our method to run most of the time in the range 1 to 2 times slower than that of [Narain et al. 2010]. Note however that in contrast to [Narain et al. 2010] where two iterations of the staggered projections algorithm are performed unconditionally, our method relies upon a residual-based convergence criterion which is enforced at each time step, and in practice satisfied in most cases.

## 7.5 Limitations

While our method is applicable to various challenging scenarios, it still suffers from a number of limitations.

First, due to our continuum approximation of granulars, the range of possible simulations is restricted to homogeneous materials where all grains share common features, such as their size. Our method is thus not appropriate for simulating polydisperse media with various grain shapes. Moreover, as the result of an averaging process, our  $\mathcal{DP}(\mu)$  rheology cannot model specific arrangements of grains, such as the formation of arches clogging the silo outlet.

Other limitations are inherent to our choice of discretization for the continuous model. In particular, as mentioned earlier, cohesive materials have to be aligned with the underlying grid in order to remain stable, which drastically limits possible scenarios including cohesion. Moreover, disturbing artifacts may happen when visually disjoint clusters of particles in neighboring cells react together. For instance, a heavy cluster of grains falling nearby a few isolated grains may send them flying, which makes sense for a continuum but not for the visible granular material.

Finally, enforcing the rheology to be satisfied only at a few discrete points in space may induce an overall loss of volume. Another solution may be brought by the recently proposed power-particles method of [de Goes et al. 2015].

## 8 Conclusion

We have presented a new method for simulating a nonsmooth viscoplastic granular rheology using the Material-Point Method. Compared to [Narain et al. 2010], our method significantly reduces visual artifacts. It also benefits from a more compact and well-posed system, which is resolved accurately under controlled convergence. We have validated our approach by retrieving typical macroscopic behaviors of granular matter, such as the Beverloo scaling effect for a silo discharge. Furthermore, we have shown that our method could easily be coupled with rigid bodies and extended to the handling of more complex media such as anisotropic materials.

In the future we intend to improve the discretization of velocity and stress fields. Better quality-to-cost ratios could be achievable using higher discretization orders, or adaptive meshes. We would also like to investigate massive parallelization of the method. Currently, most computer-intensive operations lie at the core of the friction solver, and correspond in the Cadoux/PG settings to a very large number of sparse matrix/vector products. Such operations could largely benefit from GPU or many-cores architectures. Finally, the dynamics of light and fine powders such as flour is heavily influenced by the interactions with the surrounding air, which we have disregarded here. Designing a fully coupled model to simulate such materials would be a natural extension of our work.

## Acknowledgements

We would like to thank P.-Y. Lagrée for his help in reviewing the literature on granular simulation, R. Narain for sharing his code, and anonymous reviewers for their insightful feedbacks. This work was supported in part by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissement d’Avenir.

## A Yield surfaces

The Drucker-Prager (DP) criterion is defined regardless of the space dimension as  $\sqrt{J_2} \leq \hat{\mu}p$ , with  $p$  the mean of the principal stresses  $\sigma_i$  and  $J_2$  the second invariant of the deviatoric stress tensor. We can relate this coefficient  $\hat{\mu}$  to the material’s friction angle  $\varphi$  (see Figure 18) by comparing the DP and Mohr-Coulomb yield surfaces.

In 2D, the Mohr-Coulomb (MC) criterion states, for  $\sigma_1 \geq \sigma_2$ ,

$$\frac{1}{2}(\sigma_1 - \sigma_2) = \frac{1}{2} \sin \varphi (\sigma_1 + \sigma_2).$$

Since we also have  $p = \frac{1}{2}(\sigma_1 + \sigma_2)$  and  $\sqrt{J_2} = \frac{1}{2}(\sigma_1 - \sigma_2)$ , MC and DP are strictly equivalent for  $\hat{\mu} = \sin \varphi$ .

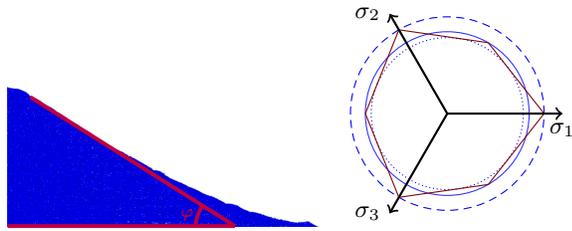
3D is more complex; MC generalizes as, for  $\sigma_1 \geq \sigma_2 \geq \sigma_3$ ,

$$\frac{1}{2}(\sigma_1 - \sigma_3) = \frac{1}{2} \sin \varphi (\sigma_1 + \sigma_3),$$

which defines a cone spanned by a non-regular hexagon in principal stress space (Figure 18). This hexagon degenerates to a triangle for  $\sin \varphi = 1$ , and tends towards a regular (yet vanishing) hexagon for  $\sin \varphi = 0$ . As  $J_2 = \frac{1}{6} \sum_{i \neq j} (\sigma_i - \sigma_j)^2$ , the DP yield surface defines a cone spanned by a circle in principal stress space. Its friction coefficient can therefore be defined with any of the 3 formulas,

- $\hat{\mu} = \frac{2\sqrt{3} \sin \varphi}{3 - \sin \varphi}$ , so that DP circumscribes MC
- $\hat{\mu} = \frac{\sin \varphi}{\sqrt{1 + \frac{1}{3} \sin^2 \varphi}}$ , so that DP inscribes MC
- $\hat{\mu} = \frac{2\sqrt{3} \sin \varphi}{3 + \sin \varphi}$ , so that DP interpolates MC at middle vertices.

Choice between these different values is application-dependent. For instance, risk-assessment simulations may want to use the inscribed surface, so that the predicted run-out length of an avalanche with DP will always overestimate the one using MC.



**Figure 18:** Left: definition of the friction angle  $\varphi$ . Right: 3D Mohr-Coulomb (red) with the three different Drucker-Prager yield surfaces (blue) in the plane of constant normal stress  $\sum \sigma_i = 3$ .

## References

- ACARY, V., CADOUX, F., LEMARÉCHAL, C., AND MALICK, J. 2011. A formulation of the linear discrete coulomb friction problem via convex optimization. *ZAMM - Journal of Applied Mathematics and Mechanics* 91, 2, 155–175.
- ALDUÁN, I., AND OTADUY, M. A. 2011. SPH granular flow with friction and cohesion. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 25–32.
- ALDUÁN, I., TENA, A., AND OTADUY, M. A. 2009. Simulation of high-resolution granular media. In *Proc. of Congreso Español de Informática Gráfica*, vol. 1.
- ANDREOTTI, B., FORTERRE, Y., AND POULIQUEN, O. 2011. *Granular media: between fluid and solid*. Cambridge University Press.
- BELL, N., YU, Y., AND MUCHA, P. J. 2005. Particle-based simulation of granular materials. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics.
- CHAMBON, G., BOUVAREL, R., LAIGLE, D., AND NAAIM, M. 2011. Numerical simulations of granular free-surface flows using smoothed particle hydrodynamics. *Journal of Non-Newtonian Fluid Mechanics* 166, 12-13 (July), 698–712.
- DAVIET, G., AND BERTAILS-DESCOUBES, F. 2016. Nonsmooth simulation of dense granular flows with pressure-dependent yield stress. *Journal of Non-Newtonian Fluid Mechanics* 234 (Aug.), 15–35.
- DAVIET, G., BERTAILS-DESCOUBES, F., AND BOISSIEUX, L. 2011. A hybrid iterative solver for robustly capturing Coulomb friction in hair dynamics. *ACM Transactions on Graphics* 30, 6 (Dec.), 1–12.
- DE GOES, F., WALLEZ, C., HUANG, J., PAVLOV, D., AND DESBRUN, M. 2015. Power particles: An incompressible fluid solver based on power diagrams. *ACM Transactions on Graphics* 34, 4 (July), 1–11.
- DUNATUNGA, S., AND KAMRIN, K. 2015. Continuum modelling and simulation of granular flows through their many phases. *Journal of Fluid Mechanics* 779 (Aug.), 483–513.
- FOLGAR, F., AND TUCKER, C. L. 1984. Orientation behavior of fibers in concentrated suspensions. *Journal of Reinforced Plastics and Composites* 3 (Apr.), 98–119.
- IHMSEN, M., WAHL, A., AND TESCHNER, M. 2013. A Lagrangian framework for simulating granular material with high detail. *Computers & Graphics* 37, 7 (Nov.), 800–808.
- IONESCU, I. R., MANGENEY, A., BOUCHUT, F., AND ROCHE, O. 2015. Viscoplastic modeling of granular column collapse with pressure-dependent rheology. *Journal of Non-Newtonian Fluid Mechanics* 219 (Mar.), 1–18.
- JEAN, M. 1999. The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering* 177, 3-4 (July), 235–257.
- JEFFERY, G. B. 1922. The motion of ellipsoidal particles immersed in a viscous fluid. *Proceedings of the Royal Society of London. Series A* 102 (Nov.), 161–179.
- JIANG, C., SCHROEDER, C., SELLE, A., TERAN, J., AND STOMAKHIN, A. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics* 34, 4 (July), 1–10.
- JOP, P., FORTERRE, Y., AND POULIQUEN, O. 2006. A constitutive law for dense granular flows. *Nature* 441, 7094 (June), 727–730.
- LAGRÉE, P.-Y., STARON, L., AND POPINET, S. 2011. The granular column collapse as a continuum: validity of a two-dimensional navier–stokes model with a  $\mu(i)$ -rheology. *Journal of Fluid Mechanics* 686 (Sept.), 378–408.
- LENAERTS, T., AND DUTRÉ, P. 2009. Mixing fluids and granular materials. In *Computer Graphics Forum*, vol. 28, 213–218.
- MAST, C. M., ARDUINO, P., MACKENZIE-HELNWEIN, P., AND MILLER, G. R. 2014. Simulating granular column collapse using the material point method. *Acta Geotechnica* 10, 1 (May), 101–116.
- MAZHAR, H., HEYN, T., NEGRUT, D., AND TASORA, A. 2015. Using Nesterov’s method to accelerate multibody dynamics with friction and contact. *ACM Transactions on Graphics* 34, 3 (May), 1–14.
- MOREAU, J. 1994. Some numerical methods in multibody dynamics: application to granular materials. *European Journal of Mechanics A* 13, 93–114.
- NARAIN, R., GOLAS, A., AND LIN, M. C. 2010. Free-flowing granular materials with two-way solid coupling. *ACM Transactions on Graphics* 29, 6 (Jan.), 1–10.
- SIDE EFFECTS SOFTWARE. 2015. *Houdini 15.0 Grains*. <http://www.sidefx.com/docs/houdini15.0/grains/about>.
- STARON, L., LAGRÉE, P.-Y., AND POPINET, S. 2012. The granular silo as a continuum plastic flow: The hour-glass vs the clepsydra. *Physics of Fluids* 24.
- STOMAKHIN, A., SCHROEDER, C., CHAI, L., TERAN, J., AND SELLE, A. 2013. A material point method for snow simulation. *ACM Transactions on Graphics* 32, 4 (Jan.), 1–10.
- TASORA, A. 2013. Efficient simulation of contacts, friction and constraints using a modified spectral projected gradient method. In *Poster Proceedings of WSCG 2013*, 69–72.
- WIECKOWSKI, Z., AND PAWLAK, M. 2015. Material point method in three-dimensional problems of granular flow. In *6th European Conference on Computational Fluid Dynamics*.
- WIECKOWSKI, Z., YOUN, S.-K., AND YEON, J.-H. 1999. A particle-in-cell solution to the silo discharging problem. *International Journal for Numerical Methods in Engineering* 45, 9 (July), 1203–1225.
- ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. *ACM Transactions on Graphics* 24, 3 (Jan.), 965–972.