

Free-Flowing Granular Materials with Two-Way Solid Coupling

Rahul Narain

Abhinav Golas

Ming C. Lin

University of North Carolina at Chapel Hill*

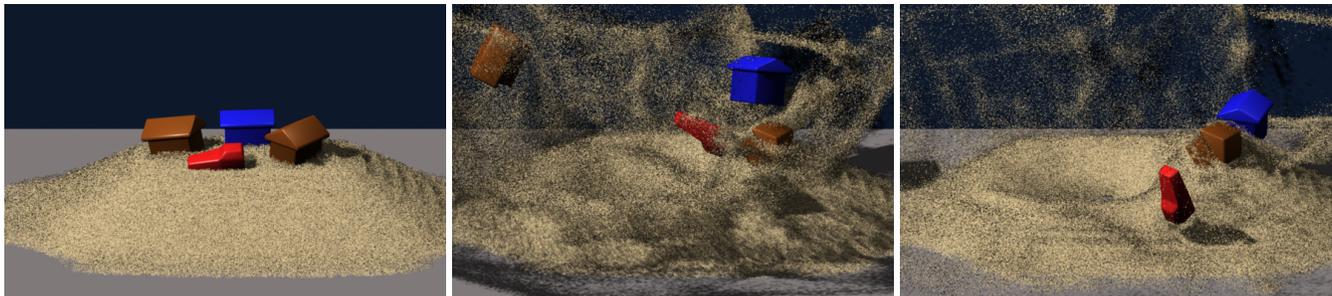


Figure 1: An explosion goes off inside a sand pile, sending freely splashing sand and rigid bodies flying in the air (running at less than 20 seconds per frame on a single-processor PC). In such a scenario, sand needs to be modeled as a cohesionless granular material.

Abstract

We present a novel continuum-based model that enables efficient simulation of granular materials. Our approach fully solves the internal pressure and frictional stresses in a granular material, thereby allows visually noticeable behaviors of granular materials to be reproduced, including freely dispersing splashes without cohesion, and a global coupling between friction and pressure. The full treatment of internal forces in the material also enables two-way interaction with solid bodies. Our method achieves these results at only a very small fraction of computational costs of the comparable particle-based models for granular flows.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: sand, granular materials, simulation, animation

1 Introduction

Granular materials such as sand, powders, cereal grains, and gravel are commonplace in the physical world around us. Being composed of very large numbers of mesoscopic grains, they show unique physical behavior that is unlike other materials such as fluids and deformable bodies, which have been well studied in computer graphics. In particular, granular materials disperse freely in free fall, flow plastically under forcing, and yet settle in stable piles.

*E-mail: {narain,golas,lin}@cs.unc.edu

The physical behavior of such materials arises from the interplay of contact and frictional forces between thousands to millions of tiny grains. Simulating the motion of each such grain is computationally prohibitive for large-scale scenarios or fine-grained materials like sand. We instead take a continuum approach, treating a granular material as a continuous fluid. Continuum-based approaches have many advantages, including efficient numerical methods due to a regular computational domain, and the ability for the user to choose the resolution of the simulation as a trade-off between performance and quality.

In order to capture the unique behaviors that such materials exhibit, we depart significantly from traditional fluid simulation techniques. Firstly, to allow the material to disperse freely when agitated but maintain its volume when at rest, we replace the existing fluid-based model’s assumption of incompressibility with a unilateral variational constraint. Secondly, unlike fluid viscosity, friction in granular materials can counteract gravity to maintain stable piles in equilibrium. This requires solving for the internal stresses in a global fashion. We present an efficient method for this numerical problem, permitting appropriate frictional behavior and solid body interaction.

1.1 Related work

Granular materials have received much attention in the fields of geomechanics, engineering, and computational physics. Being “a distinctive form of matter that exhibits behavior rather different from that of ordinary solids, liquids, or gases” [Behringer et al. 1999], granular materials show surprisingly complex behavior that is still being studied in the physics and engineering communities. Due to the central role of friction and the dissipative nature of inelastic contact interactions, granular flow resists analysis by traditional approaches of statistical mechanics, and describing it by a continuum model which abstracts over individual grain interactions continues to be a topic of active research.

For visual applications, however, the use of a simplified continuum model that only treats the more common phases of granular flow can still give very convincing results. This was demonstrated by Zhu and Bridson [2005], who modeled sand as an incompressible fluid by adding frictional forces to a traditional fluid simulator. Their approach relies on identifying rigidly moving regions of ma-

terial to obtain stable piles. Lenaerts and Dutré [2009] adapted this method to the setting of smoothed particle hydrodynamics (SPH), and also enabled simulating sand-water interaction in this setting. The incompressibility assumption, however, can lead to undesirable cohesive behavior, and it is not known if the rigidification approach for friction supports two-way coupling with solid bodies. In soil mechanics, continuum-based approaches have also been employed to study large-scale phenomena such as avalanches and landslides [Aranson and Tsimring 2001; Quecedo et al. 2004; Josserand et al. 2004].

Another natural method for animating granular materials is to directly simulate the interactions between individual grains. This includes the class of discrete element methods used in engineering applications [Bićanić 2004]. In graphics, Luciani et al. [1995] developed a particle system model for granular materials using damped spring forces. Bell et al. [2005] used a molecular dynamics method while modeling grains as rigid compounds of spheres, exhibiting compelling behavior with two-way solid body interaction. However, discrete methods incur a heavy computational expense to model fine-grained materials due to the sheer number of particles that must be resolved. This issue was addressed in part by Alduán et al. [2009], who proposed a post-processing step that interpolates fine grains over a coarser discrete simulation.

We briefly survey some other approaches that have been taken for modeling granular materials in computer graphics. Some of the earliest work used particle systems with heuristic inter-particle forces to give fluid-like or sand-like behavior [Miller and Pearce 1989; Luciani et al. 1995]. Sand has also been modeled using height fields [Li and Moshell 1993; Chanclou et al. 1996], and several extensions have been proposed for modeling footprints and tracks [Sumner et al. 1999], using multi-valued height fields for some 3D effects [Onoue and Nishita 2003], and combining with a flowing surface layer composed of particles [Zhu and Yang 2010]. Cellular automata can be used for interactive sand manipulation [Pla-Castells et al. 2006].

Our method builds on the continuum approach, and is more general, allowing non-cohesive behavior, globally coupled frictional handling, and efficient two-way interaction with solids. These characteristics allow qualitatively realistic, compelling results similar to fine-scale discrete methods at a far lower computation time and significantly reduced memory requirements.

1.2 Overview

In our method, we assume that the material’s grain size is so small that the precise motion of individual grains is unimportant. Therefore, we treat the granular material instead as a continuous fluid flowing under the action of external forces and internal stresses. The internal stresses represent the contact and frictional forces between individual grains in the material. Computing the motion of the material under these forces requires two stages: first, to determine the internal stresses given the current state of the material, and second, to integrate the motion of the material under the influence of these forces.

While stress computation is most efficiently computed on a regular Eulerian grid, integrating the motion of a granular material proves to be best suited to a Lagrangian setting. We therefore take a hybrid approach, informed by previous work such as the FLIP method [Zhu and Bridson 2005], where an Eulerian representation is used to compute internal forces, while advection is performed using particles. Note that these simulation particles do not represent individual grains, but rather moving “clumps” of matter which act as samples of the material.

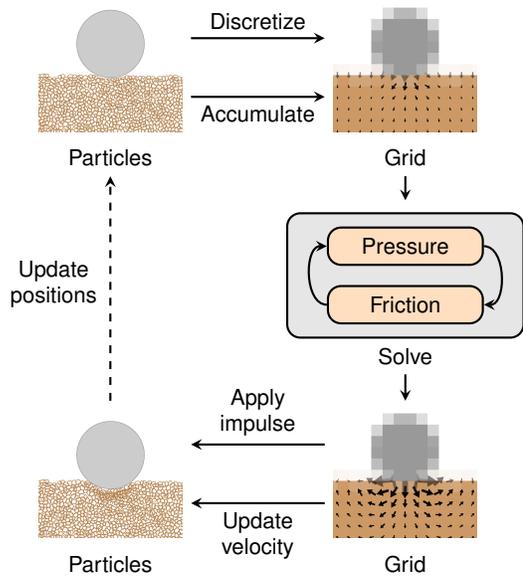


Figure 2: The main simulation loop of our algorithm. See Section 1.2 for details.

We briefly describe the overall simulation loop, illustrated in Fig. 2. At the beginning of a time step, the state of the material is projected to the Eulerian setting by accumulating particle values onto the grid. Using the Eulerian representation, the material’s internal stresses and interaction forces with other bodies are computed by the continuum model, taking into account the coupling between pressure and friction (see Fig. 3). The computed stresses are used by the Lagrangian representation to update the particle velocities. The motion of each particle is then integrated over the time step, giving the updated state of the material.

The remainder of the paper is organized as follows. The continuum model is described in Section 2, and the particle-based advection scheme in Section 3. Further implementation details are given in Section 4. In Section 5, we discuss the results of our method on several example scenarios.

2 Granular material as a continuum

Our approach has much in common with traditional fluid simulation. However, the behavior of stresses in a granular material differs qualitatively from pressure and viscosity in a fluid, and novel techniques are required to treat them faithfully in a continuum model.

One characteristic property of granular materials is that they can disperse freely. Dry grains apply no attractive forces on each other, so in macroscopic terms the material displays little cohesion. This means that unlike a liquid, a granular material in motion may not have a clearly defined surface at all! Consider a sand pile transitioning to a cloud of grains under impact: where does the pile surface end and the cloud begin? Another significant difference is the effect of friction. It can transmit forces across large distances, and unlike fluid viscosity, it continues to act even at rest, resulting in stable piles and other quasi-rigid behaviors.

Indeed, many of the problems encountered in this work mirror those of rigid body dynamics transferred to a continuum setting, since granular materials are essentially aggregates of numerous rigid bodies in contact. In order to capture their complex behavior, we synthesize ideas from both the fields of computational fluid dynamics and rigid-body simulation together to derive our new formulation.

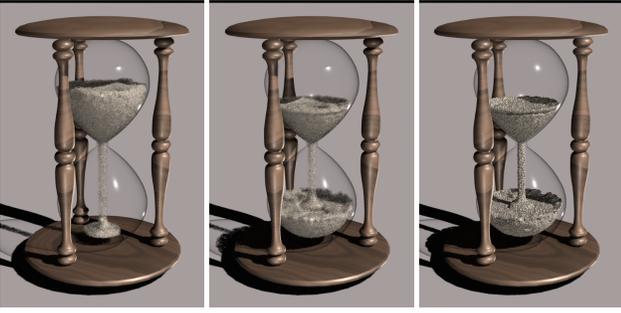


Figure 3: Sand trickles down an hourglass. The rate of flow is constant due to the coupling between pressure and friction. On the right, the simulation particles used for advection are visualized.

2.1 Fundamentals

From a continuum viewpoint, we represent the physical state of a granular material through its mass density ρ and flow velocity \mathbf{v} . The system is acted upon by external forces \mathbf{f}_{ext} and internal stress $\boldsymbol{\sigma}$. The stress $\boldsymbol{\sigma}$ is a symmetric tensor field which can be decomposed into an isotropic mean stress (a “pressure”) p and a traceless deviatoric component \mathbf{s} , which represents frictional stresses.

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mathbf{s}. \quad (1)$$

The equations of motion of the material may be derived from the conservation laws for mass and momentum. If we denote the Lagrangian time derivative by $D/Dt = \partial/\partial t + (\mathbf{v} \cdot \nabla)$, the time evolution of the system is given by the transport of mass,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (2)$$

and the effect of forces,

$$\rho \frac{D\mathbf{v}}{Dt} = \mathbf{f}_{\text{ext}} - \nabla p + \nabla \cdot \mathbf{s}. \quad (3)$$

The stress response of frictional materials can be described in terms of plastic yielding: the material resists deformation when the stress is within a certain *yield criterion*, and begins to flow plastically when the yield criterion is violated. Following previous work [Zhu and Bridson 2005], we employ the Drucker-Prager yield criterion for its computational simplicity. When the cohesion is negligible, as is the case for most dry granular materials, the yield criterion can be expressed by the inequality

$$\|\mathbf{s}\|_F \leq \sqrt{3}\alpha p \quad (4)$$

where $\|\mathbf{s}\|_F = \sqrt{\sum s_{ij}^2}$ is the Frobenius norm of \mathbf{s} , and α is the frictional coefficient. This is related to the angle of repose θ by $\alpha = \sqrt{2/3} \sin \theta$.

We use an impulse-based time-stepping scheme throughout our simulation, where forces are considered to act instantaneously at the beginning of each time step. Let the density and velocity field of the material at the beginning of time step n be ρ^n and \mathbf{v}^n respectively. For a time step of length Δt , we may define the intermediate velocity field $\tilde{\mathbf{v}}$ for given stress components p and \mathbf{s} as

$$\tilde{\mathbf{v}} = \mathbf{v}^n + \frac{\Delta t}{\rho^n} (\mathbf{f}_{\text{ext}} - \nabla p + \nabla \cdot \mathbf{s}). \quad (5)$$

This velocity field may be used to update the particle velocities and positions in the advection step.

2.2 Pressure

We make two simplifying assumptions to the dynamics of granular materials. Firstly, we assume that the material cannot be compressed beyond a fixed *critical density* ρ_{max} which permits free flow. This can be thought of as the density of a stable pile of material at rest. (In reality, this density is slightly higher than critical, as grains at rest must separate a little to move past each other, but this effect is visually imperceptible.) Secondly, when $\rho < \rho_{\text{max}}$, grains are not packed together and only interact via intermittent collisions, and we neglect the effect of these interactions.

Under these assumptions, the granular material has a maximum density ρ_{max} , and the pressure p acts to prevent any further compression of the material. However, in contrast to traditional incompressible fluids, where both positive and negative flow divergence are always nullified by the pressure, an absence of cohesion in granular material implies that a material undergoing diverging flow experiences no internal forces.

This behavior can be expressed as an inequality on the density of the material, or more generally on its “volume fraction” ϕ ,

$$\phi = \frac{\rho}{\rho_{\text{max}}} \leq 1. \quad (6)$$

A similar constraint was recently applied to simulation of dense crowds of pedestrians on a 2D plane by Narain et al. [2009], which they called the unilateral incompressibility constraint. We adapt the same numerical method for computing the corresponding pressure p to satisfy the constraint in 3D space. For completeness, we briefly describe this method below.

The volume fraction at the end of a time step Δt can be estimated by discretizing (2) with $\mathbf{v} = \tilde{\mathbf{v}}$. This gives

$$\phi^{n+1} = \phi^n - \Delta t \nabla \cdot (\phi^n \tilde{\mathbf{v}}) \quad (7)$$

$$= \phi^{n+1}|_{p=0} + \frac{\Delta t^2}{\rho_{\text{max}}} \nabla^2 p, \quad (8)$$

where $\phi^{n+1}|_{p=0}$ is the predicted volume fraction when pressure is zero.

The pressure p is chosen so that the constraint (6) is satisfied for ϕ^{n+1} . Assuming inelasticity, the pressure must be such that it only just maintains the constraint, but does not cause any additional “bounce”. This implies a complementarity

$$p(1 - \phi^{n+1}) = 0 \quad (9)$$

That is, when p is active (nonzero), ϕ^{n+1} must equal 1, not fall below it. Finally, the absence of cohesion implies that $p \geq 0$, which can also be seen from the yield condition (4) itself.

These conditions define a linear complementarity problem

$$\begin{aligned} \mathbf{A}_1 \mathbf{p} + \mathbf{b}_1 &\geq 0, \\ \mathbf{p} &\geq 0, \\ \mathbf{p}^T (\mathbf{A}_1 \mathbf{p} + \mathbf{b}_1) &= 0, \end{aligned}$$

where

$$\mathbf{A}_1 = \frac{\Delta t^2}{\rho_{\text{max}}} \mathbf{D}_1^T \mathbf{D}_1, \quad (10)$$

$$\mathbf{b}_1 = 1 - \phi^{n+1}|_{p=0}, \quad (11)$$

and \mathbf{p} is a vector containing the pressure values at all 3D grid cells. \mathbf{D}_1 denotes the finite difference matrix mapping a scalar field p to



Figure 4: Three stable piles with friction coefficients $\alpha = 0.3, 0.5,$ and $0.7,$ respectively. Lower friction leads to flatter piles.

the vector field ∇p . Boundary conditions are treated using the standard technique of ghost cells, on which we set $\phi^{n+1}|_{p=0} = 1$ to prevent any flow across domain boundaries.

Because A_1 is positive semidefinite, these are simply the KKT conditions for minimizing a quadratic function

$$F = \frac{1}{2} \mathbf{p}^T A_1 \mathbf{p} + \mathbf{b}_1^T \mathbf{p}, \quad (12)$$

for \mathbf{p} with all non-negative components. Therefore, the pressure p is equivalently defined by the minimization problem

$$\min F(\mathbf{p}) : \mathbf{p} \geq 0. \quad (13)$$

2.2.1 Density correction

Numerical error in advection can cause the density to violate (6) slightly at the beginning of a time step, which would cause the solver to apply spurious corrective pressure at those cells and lead to oscillations. To avoid this issue, we first redistribute the density so the initial state is valid.

Imagine freezing the material in time and applying an instantaneous displacement $\Delta \mathbf{x}$ such that each bit of material at \mathbf{x} is moved to $\mathbf{x} + \Delta \mathbf{x}$. We choose $\Delta \mathbf{x}$ such that the new volume fraction

$$\phi^n \leftarrow \phi^n - \nabla \cdot (\phi^n \Delta \mathbf{x}) \quad (14)$$

satisfies (6). Letting $\Delta \mathbf{x} = -\nabla y / \rho^n$ for some scalar field y , we can solve this using the pressure solve itself, substituting $\Delta \mathbf{x}$ for $\tilde{\mathbf{v}}$ and 1 for Δt .

At the beginning of the time step, we solve this and update ϕ^n with (14) and ρ^n correspondingly. We also store $\Delta \mathbf{x}$ for use during advection.

2.3 Friction

The frictional stress \mathbf{s} is a symmetric, trace-free, rank-2 tensor,

$$\mathbf{s} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{xy} & s_{yy} & s_{yz} \\ s_{xz} & s_{yz} & s_{zz} \end{bmatrix},$$

$$s_{xx} + s_{yy} + s_{zz} = 0,$$

subject to the yield condition (4). Because the yield constraint is convex, it can be linearized by replacing the constraint surface with a set of hyperplanes tangent to it. Taking the hyperplanes orthogonal to each component of \mathbf{s} yields bound constraints

$$-s_{\max} \leq s_{ij} \leq s_{\max} \quad (15)$$

for $i, j \in \{x, y, z\}$, and with $s_{\max} = \alpha p$. If desired, more hyperplanes can be included to improve the isotropy of the frictional response. The effect of different values of α is shown in Figure 4.

For plastic flow, the principle of maximum plastic dissipation [Simo and Hughes 1998] states that among all possible stresses satisfying

the yield criterion, the actual stress is that which maximizes the rate of dissipation of kinetic energy. We compute the frictional stress by directly applying the maximum dissipation principle over discrete time steps. This automatically captures the interplay between pressure and friction within the material both in motion and at rest, eliminates the need for an additional rigidity condition for stable piles, and naturally generalizes to interaction with solid bodies.

To maximize dissipation, we compute the frictional stress by requiring that it minimize the kinetic energy of the system. However, from a numerical perspective, directly using the total kinetic energy results in a poorly conditioned system due to the division by ρ^n in the definition of $\tilde{\mathbf{v}}$. This can be ameliorated by using an additional weighting $w = \rho^n / \rho_{\max}$ on the energy:

$$E = \frac{1}{2} \int w \rho^n \|\tilde{\mathbf{v}}\|^2 dV \quad (16)$$

This modification makes the problem far more efficient, and because p and hence \mathbf{s} are only ever nonzero when ρ^n is close to ρ_{\max} , w is 1 over almost the entire support of \mathbf{s} so only a small amount of error is introduced.

The energy can be expressed as a quadratic functional

$$E = \frac{1}{2\rho_{\max}} \int \|\rho^n \tilde{\mathbf{v}}|_{s=0} + \Delta t \nabla \cdot \mathbf{s}\|^2 dV$$

$$= E|_{s=0} + \frac{1}{2} \mathbf{s}^T A_2 \mathbf{s} + \mathbf{b}_2^T \mathbf{s}. \quad (17)$$

where

$$A_2 = \frac{\Delta t^2}{\rho_{\max}} D_2^T D_2, \quad (18)$$

$$\mathbf{b}_2 = \frac{\Delta t}{\rho_{\max}} D_2^T \rho^n \tilde{\mathbf{v}}|_{s=0}. \quad (19)$$

Here \mathbf{s} is treated as the vector composed of the concatenation of the components of frictional stress at all grid points, and D_2 is the matrix mapping a tensor field \mathbf{s} to the vector field $\nabla \cdot \mathbf{s}$. Minimizing E subject to the constraints (15) determines the frictional stress:

$$\min E(\mathbf{s}) : -s_{\max} \leq s_{ij} \leq s_{\max}. \quad (20)$$

Boundary conditions are treated as follows. Physically, the normal force at the boundary due to friction must vanish. This corresponds to the diagonal components s_{ii} on the ghost cells, so these are fixed at zero. The tangential force corresponds to the off-diagonal components on edges along the boundary; the associated finite difference stencil does not refer to cells outside the domain at all, so no boundary conditions need to be specified here.

2.4 Interaction with solid bodies

By posing both the pressure and friction solves as minimization problems, two-way solid-fluid interaction (as shown in Fig. 5) can be very naturally handled along the lines of variational fluid-solid coupling [Batty et al. 2007].

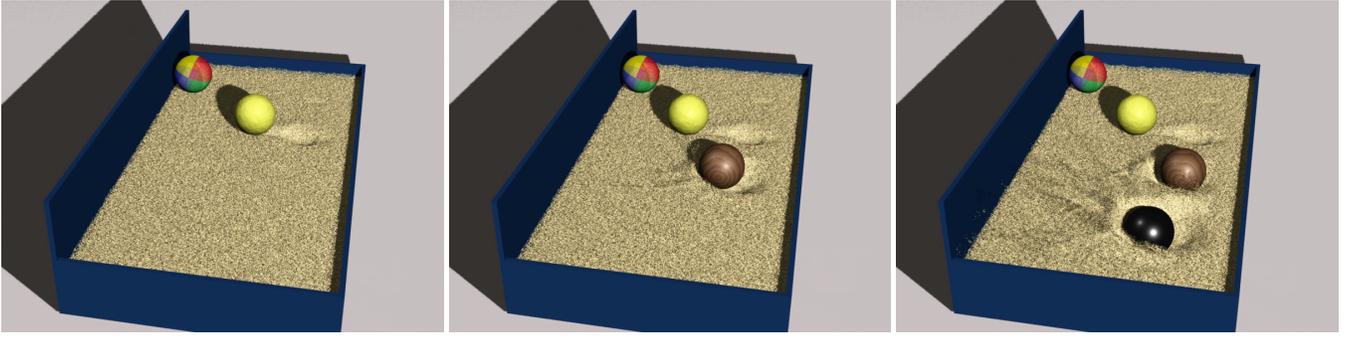


Figure 5: Spheres of masses $m = 0.3, 1, 3,$ and 10 units strike a sand surface. The heaviest spheres are denser than the sand itself, yet internal friction allows the material to support their weight.

For a solid body interacting with a fluid or granular material, we can define a linear operator \mathbf{J} which integrates stresses on its surface to give generalized forces. For example, for rigid bodies, we can represent generalized forces as 6-vectors with 3 components of force followed by 3 components of moment. The \mathbf{J} operator for scalars then is given by

$$\mathbf{J}_1 p = \int \left[\begin{array}{c} \nabla p \\ \nabla p \times \mathbf{x} \end{array} \right] \phi_s dV, \quad (21)$$

while that for tensorial stresses is

$$\mathbf{J}_2 \mathbf{s} = \int \left[\begin{array}{c} \nabla \cdot \mathbf{s} \\ (\nabla \cdot \mathbf{s}) \times \mathbf{x} \end{array} \right] \phi_s dV, \quad (22)$$

in terms of the fraction of volume ϕ_s covered by the solid body. Under given material stresses p and \mathbf{s} and external forces \mathbf{F}_{ext} , the net generalized force on the body is simply

$$\mathbf{F} = \mathbf{F}_{\text{ext}} - \mathbf{J}_1 p + \mathbf{J}_2 \mathbf{s}. \quad (23)$$

Furthermore, the adjoint \mathbf{J}_1^T yields a scalar field $\mathbf{J}_1^T \mathbf{V}$ which gives the distribution of normal velocity times differential area over the surface of the body; intuitively, this describes how the space occupied by the body changes due to its velocity.

Consider a body with inertia matrix \mathbf{M} and initial generalized velocity \mathbf{V}^n . After applying the impulse $\Delta t \mathbf{F}$, it moves with the updated velocity $\tilde{\mathbf{V}} = \mathbf{V}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}$.

For the pressure coupling, we replace the volume fraction constraint (6) with

$$\phi + \phi_s \leq 1. \quad (24)$$

Enforcing this constraint requires estimating the volume fraction covered by the body at the next time step. Using the adjoint property of \mathbf{J}_1 , this is given by

$$\begin{aligned} \phi_s^{n+1} &= \phi_s^n + \Delta t \mathbf{J}_1^T \tilde{\mathbf{V}} \\ &= \phi_s^{n+1}|_{p=0} - \Delta t^2 \mathbf{J}_1^T \mathbf{M}^{-1} \mathbf{J}_1 p, \end{aligned} \quad (25)$$

where $\phi_s^{n+1}|_{p=0}$ is the predicted volume fraction without pressure coupling. Thus we add $\Delta t^2 \mathbf{J}_1^T \mathbf{M}^{-1} \mathbf{J}_1$ to the matrix \mathbf{A}_1 , and $-\phi_s^{n+1}|_{p=0}$ to the linear term \mathbf{b}_1 in (10) and (11) respectively.

In the friction solve, we simply need to add the kinetic energy of the rigid body to E . This is given by

$$\begin{aligned} E_s &= \tilde{\mathbf{V}}^T \mathbf{M} \tilde{\mathbf{V}} \\ &= E_s|_{s=0} + \frac{1}{2} \Delta t^2 \mathbf{s}^T \mathbf{J}_2^T \mathbf{M}^{-1} \mathbf{J}_2 \mathbf{s} + \Delta t \mathbf{s}^T \mathbf{J}_2^T \tilde{\mathbf{V}}|_{s=0}. \end{aligned} \quad (26)$$

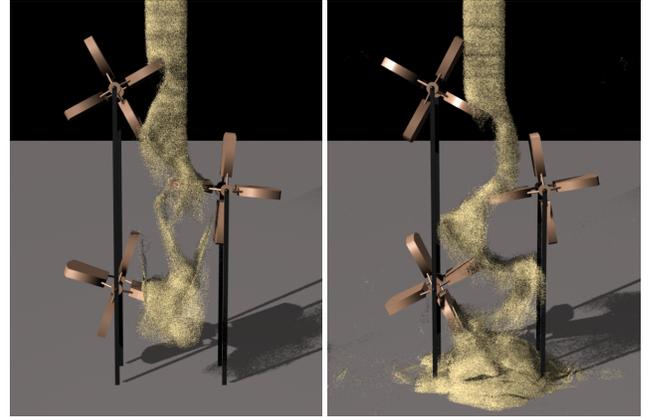


Figure 6: Sand falls on a series of paddle wheels, setting them in motion.

Thus, we add $\Delta t^2 \mathbf{J}_2^T \mathbf{M}^{-1} \mathbf{J}_2$ to the matrix \mathbf{A}_2 and $\Delta t \mathbf{J}_2^T \tilde{\mathbf{V}}|_{s=0}$ to the linear term \mathbf{b}_2 in (18) and (19).

It is also necessary to modify the velocity updates of the granular material to account for the reduced volume occupied by it. Assuming that the body forces are distributed volumetrically in a partially occupied cell, the velocity update rule (5) becomes

$$\tilde{\mathbf{v}} = \mathbf{v}^n + \frac{\Delta t}{\rho^n} (1 - \phi_s) (\mathbf{f}_{\text{ext}} - \nabla p + \nabla \cdot \mathbf{s}). \quad (27)$$

Once the pressure and frictional stresses are computed, the rigid body can be updated by applying an impulse $\Delta t \mathbf{F}$ using (23) and advancing it through one time step.

2.5 Putting it together

Given the current state of the system in terms of density ρ and velocity \mathbf{v} , the pressure p and frictional stress \mathbf{s} are determined by minimization of two coupled quadratic programs $F(p)$ and $E(\mathbf{s})$, subject to corresponding linear inequality constraints. The problem of solving these coupled minimizations mirrors that addressed by Kaufman et al. [2008], who treated the contact and frictional forces between rigid bodies in the same way. This solution procedure, using staggered projections on each minimization in turn, extends to the continuous case we consider.

The staggered projection method works by fixing the value of one variable, say \mathbf{s} , and finding p through (13) using the current value of \mathbf{s} . Then, the resulting p is fixed and \mathbf{s} is updated through (20). This pair of minimizations forms one iteration, which is repeated.

This procedure of solving (13) and (20) can be shown to be a non-expansive mapping, of which the coupled solution is a fixed point; we refer the reader to the supplementary document for the proof. In the discretized setting, both problems become quadratic programs which can be solved efficiently, as we describe in Section 4. Because of the non-expansive property of staggered projection iterations, the iteration process is guaranteed not to diverge, and in practice we have always observed it to converge extremely quickly.

After the internal stresses p and s have been computed, it remains to integrate (2) and (3) to advance the physical state of the material to the next time step. This is described in the following section.

3 Particle-based advection

Unlike traditional fluids considered in computer graphics, the flow of a granular material may not be purely incompressible due to the absence of cohesion. As a consequence, it is difficult to apply techniques based on semi-Lagrangian advection while ensuring the conservation of mass and momentum. Furthermore, a granular material that exhibits a coherent surface in a pile at rest may transition into a sparse cloud of grains in very dynamic events such as splashes or free fall, rendering surface tracking methods used for animating liquids inapplicable. We therefore use a Lagrangian approach for advecting the material, which can easily handle these properties. Our method can be considered as an extension of the fluid-implicit-particle (FLIP) method [Brackbill and Ruppel 1986; Zhu and Bridson 2005].

In the Lagrangian setting, the granular material is represented as a set of simulation particles. Each simulation particle represents not a single grain but a macroscopic sample of the material—a moving “clump” of matter with mass m_i centered at a point \mathbf{x}_i and moving with average velocity \mathbf{v}_i . At the beginning of a time step, the continuum values of ρ^n and \mathbf{v}^n are defined by accumulating the values of particles near each grid cell. Each particle is treated as a point mass, and its contribution is divided among its neighboring 8 grid nodes using standard trilinear weights. The internal stresses are then computed through the continuum model to determine the intermediate velocity $\tilde{\mathbf{v}}$. Finally, advection is performed by updating the particles using this velocity field.

In the advection step, we update each particle’s position based on the grid velocity. First the displacement $\Delta\mathbf{x}$ for density correction (Section 2.2.1) is applied, then particle positions are advanced using the velocity field. Particle velocities are updated by adding the change in grid values from the previous time step, following the FLIP method.

3.1 Particle shapes and split/merge operations

Unlike fluids like water which are practically incompressible, granular materials can exhibit visibly diverging flow, such as a dispersing mass of sand thrown into the air. This presents a difficulty for traditional particle-based advection techniques designed for incompressible flow: particles spread farther and farther from each other, and the simulated fluid eventually separates into clumps corresponding to individual particles instead of spreading uniformly. While this is satisfying for liquids, or for wet sand which exhibits cohesion, grains in a dispersing cloud of *dry* sand are often smoothly distributed.

To ensure that the material remains faithfully sampled in such cases, it is necessary to somehow track the spreading of simulation particles, and insert additional particles where necessary. We do this simply by attaching to each particle a shape centered around the particle position, initially spherical, which is stretched

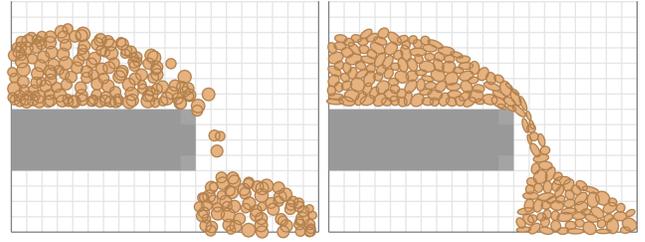


Figure 7: If particle distortion is not tracked (left), particles cannot remain well distributed in the material during diverging flow. By tracking the shape of particles under the flow (right), we can split and merge them appropriately to maintain a good distribution.

and squeezed by the flow. When the particle becomes too large or too small, it is split into two or merged with an adjacent particle, thus automatically maintaining a good distribution of particles (Fig. 7). Previous work in incompressible SPH-based fluid simulation has used splitting and merging as a level of detail approach to accelerate simulation [Adams et al. 2007], but here it is a necessity for representing a dispersing mass of not incompressible material.

In general, the shape of a particle is an ellipsoid, which we represent as the region $(\mathbf{x} - \mathbf{x}_i)^T \mathbf{A}_i^{-1} (\mathbf{x} - \mathbf{x}_i) \leq 1$ for a symmetric positive definite matrix \mathbf{A}_i . The semi-axes of the ellipsoid are given by the eigenvectors and square roots of eigenvalues of \mathbf{A}_i . To first order, as this ellipsoid is advected through the velocity field \mathbf{v} , its time evolution is given by

$$\frac{d\mathbf{A}_i}{dt} = \mathbf{A}_i \mathbf{J}_{\mathbf{v}}^T(\mathbf{x}_i) + \mathbf{J}_{\mathbf{v}}(\mathbf{x}_i) \mathbf{A}_i \quad (28)$$

where $\mathbf{J}_{\mathbf{v}}(\mathbf{x}_i)$ is the Jacobian of the velocity field at position \mathbf{x}_i .

A user-specified parameter r controls the size of particles. We define a particle as *valid* if the lengths of its semi-axes lie within the range $[\frac{1}{\sqrt{2}}r, \sqrt{2}r]$. We have found that setting r to one-fourth the grid spacing is generally effective. The bounds on the axis lengths prevent particles from becoming too big, too small, or too skinny, so that they can be treated as points when interpolating grid values. Split and merge operations are applied to particles that become invalid. These operations are defined so that they conserve the mass, momentum, and center of mass of the system.

A split operation divides a particle into two identical particles along the longest axis. Each child particle has half the axis length along this direction, while the other two axes remain the same.

A merge operation replaces two nearby particles with one larger particle centered at their center of mass, carrying their total mass and momentum. Upon merging particles i and j , the shape of the new particle, say k , is given by

$$m_k \mathbf{A}_k = m_i (\mathbf{A}_i + \Delta\mathbf{x}_i \Delta\mathbf{x}_i^T) + m_j (\mathbf{A}_j + \Delta\mathbf{x}_j \Delta\mathbf{x}_j^T), \quad (29)$$

where $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_k$, and similarly $\Delta\mathbf{x}_j$, are the displacements of the old particles from the new center of mass. We chose this rule so that immediately merging the children of a split returns the original particle.

We perform split and merge operations only when the resulting particle(s) are valid. In practice, this yields a consistent set of particles without oscillations.

4 Implementation details

An overview of all the steps of our method is shown in Figure 8. Below we describe some details of our implementation.

For each time step:

1. Accumulate density ρ^n and velocity \mathbf{v}^n onto grid.
2. Perform density correction on ρ^n .
3. Repeat until convergence or maximum iterations:
 - (a) Compute friction \mathbf{s} by minimizing (20) for fixed p .
 - (b) Compute pressure p by minimizing (13) for fixed \mathbf{s} .
4. Find intermediate velocity $\tilde{\mathbf{v}}$ through (5).
5. Update solid bodies with an impulse $\Delta t \mathbf{F}$ (23).
6. Update particles:
 - (a) Update velocities using FLIP.
 - (b) Move particles through the velocity field $\tilde{\mathbf{v}}$.
 - (c) Update shapes using (28).
 - (d) Split and merge particles.

Figure 8: The main steps of our method.

For the continuum model, we use a regular Cartesian grid on which physical quantities are stored in a “staggered” fashion following Goktekin et al [2004]. Scalars ρ and p and diagonal components of \mathbf{s} are stored at cell centers, vector components of \mathbf{v} at cell faces, and off-diagonal components of \mathbf{s} at cell edges. Spatial derivatives are computed through finite differences. For stability, we choose time steps so that no particle moves more than half of a grid cell in a single time step. As the impulse-based integration scheme means that forces only act on particles at the beginning of a time step, a forward Euler step suffices to perform particle advection.

In the discretized setting, (13) and (20) are quadratic programs (QPs) with sparse, symmetric, positive semidefinite matrices. These problems can be solved efficiently using the recent algorithm of Dóstał and Schöberl [2005], which we extended to incorporate an MIC(0) preconditioner [Bridson and Müller-Fischer 2007] and support two-sided bound constraints. However, a naïve approach causes the friction solve to converge slowly, because of the coupling between different components of \mathbf{s} . By instead minimizing with respect to one component of \mathbf{s} at a time (which amounts to optimizing over orthogonal subspaces in turn), an approximate frictional solution can be found which converges much more rapidly. Because the projections are nevertheless repeated in an outer loop, the correct solution remains the fixed point of the procedure. Please see the supplementary document for more details.

Convergence is also greatly accelerated by warm-starting the solver using the pressure and friction values computed at the previous time step as initialization. We found that in practice, performing just a few iterations of staggered projections sufficed to give stable and convincing results; only 2 iterations were used in all our results.

4.1 Multiple interacting materials

Our method can easily be extended to handle multiple granular materials with different properties interacting in a single scene (Fig. 9).

To account for materials of varying densities and friction coefficients, we associate each simulation particle i with its own values of $\rho_{\max,i}$ and α_i . For the pressure constraint, the volume fraction ϕ must be redefined because ρ_{\max} will vary for materials of different densities. Instead of accumulating m_i of each particle on the grid to obtain ρ and computing $\phi = \rho/\rho_{\max}$, we obtain ϕ directly by accumulating the minimum volume of each particle $V_{\min,i} = m_i/\rho_{\max,i}$. The pressure solve is then given by

$$\mathbf{A}_1 = \Delta t^2 \mathbf{D}_1^T \frac{\phi^n}{\rho^n} \mathbf{D}_1, \quad (30)$$

$$\mathbf{b}_1 = 1 - \phi^{n+1}|_{p=0}, \quad (31)$$

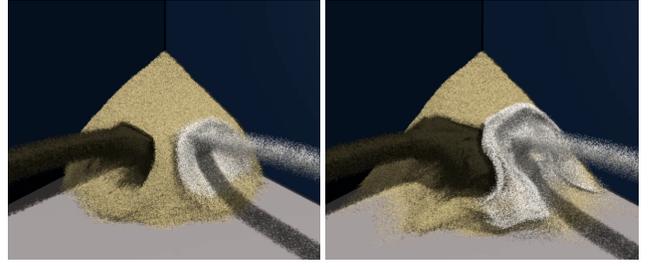


Figure 9: An example of multiple granular materials interacting in a scene. From left to right: high density and friction (black), medium density and friction (sandy), low density and friction (white).

instead of (10) and (11). ϕ^n/ρ^n here is treated as a diagonal matrix.

In the friction solve, the weighting w now simply equals ϕ . This leads similarly to replacing (18) and (19) with

$$\mathbf{A}_2 = \Delta t^2 \mathbf{D}_2^T \frac{\phi^n}{\rho^n} \mathbf{D}_2, \quad (32)$$

$$\mathbf{b}_2 = \Delta t \mathbf{D}_2^T \phi^n \tilde{\mathbf{v}}|_{\mathbf{s}=0}. \quad (33)$$

Since the coupling matrices of Section 2.4 are computed independently of the material, they remain exactly the same. In the particle-based advection part, we only merge two particles if they have identical material properties. This ensures that each simulation particle is associated with only one distinct material, and facilitates rendering. If it is desired to simulate materials with properties that vary continuously over space, this condition can be relaxed.

5 Results

We have applied our method to several scenarios, showing many characteristic behaviors of granular materials including stable pile formation, freely dispersing clouds of grains, and two-way rigid body interaction.

Absence of cohesion: In Figure 1, several rigid bodies are placed on top of a sand pile, and an explosion goes off inside the pile, sending the sand and the bodies into the air. We modeled the explosion as an instantaneous outward impulse applied to particles in a small sphere in the interior of the pile. In an incompressible fluid, this divergence would be immediately nullified by the pressure projection, but our cohesionless pressure solve allows the sand to disperse in a realistic manner.

Pressure/friction interaction: Figure 3 shows sand falling in an hourglass. Friction plays a central role here, as it maintains a constant rate of flow of sand through the neck of the hourglass, unlike a traditional liquid whose rate of flow would depend on the height of the liquid above it.

Solid coupling: Figure 5 demonstrates two-way coupling between rigid bodies and granular material. Note that an accurate coupling of frictional stress is necessary for the material to be able to support the weight of bodies much denser than itself. Friction also causes the lighter spheres roll without slipping on the sand surface. Another example with rotating paddle wheels is shown in Figure 6.

Multiple materials: Figure 9 shows different kinds of granular materials colliding and interacting in a single scene. The differing densities and friction of the materials give rise to different interactions.

Comparison with previous work: In Figure 11, we compare the behavior of our simulator with the results of Zhu and Bridson

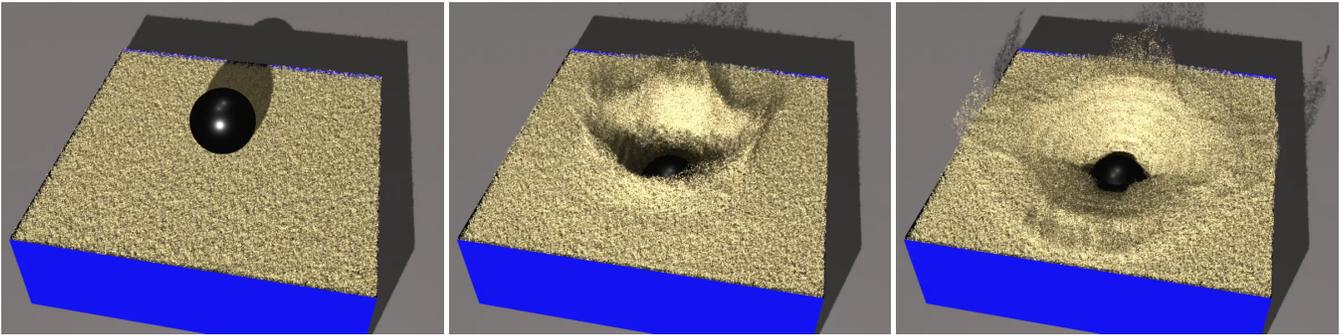


Figure 10: A 1-inch metal sphere hits sand at high velocity, creating a splash and a large crater.

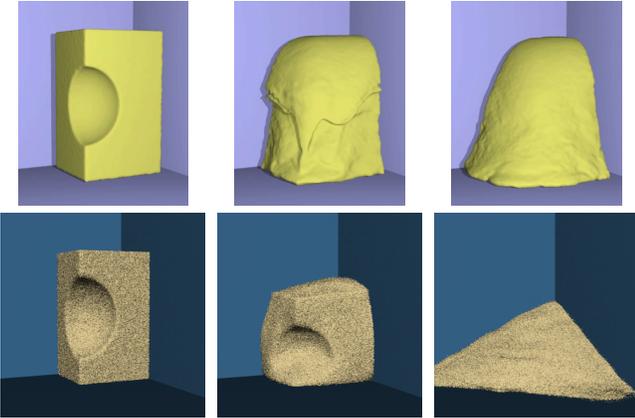


Figure 11: A column of granular material is simulated using Zhu and Bridson’s method [2005] (top) and our method (bottom).

[2005]. Due to their incompressibility assumption, their approach works well for modeling cohesive materials like wet sand, while ours behaves more like a dry granular material with zero cohesion.

Real-world comparison: Figure 10 shows an impact scenario with a fast-moving metal sphere. We modeled this on a real experiment from the Discovery Channel, viewable at <http://dsc.discovery.com/videos/time-warp-deep-impact.html>. Our result is qualitatively consistent with the real-world behavior of sand in this scenario. Some anisotropy visible in the splash is due to the linearization of the friction constraint (15). This can be avoided by adding more constraint hyperplanes, at the cost of higher simulation time.

Rendering granular materials from a continuum-based simulation poses its own challenges, as fine grains for rendering must be sampled from the simulation in a temporally coherent manner. Previous methods either simulated a cohesive material which maintained a well-defined surface for rendering [Zhu and Bridson 2005], or attached grains rigidly to simulation particles [Lenaerts and Dutré 2009] leading to visible clumps. Neither of these is applicable to our technique, so we have developed a heuristic approach that we found to perform well for many situations.

We render sand as a cloud of points, producing a granular appearance. To each simulation particle, we associate a number of render points sampled within the particle’s ellipsoid, that are passively advected with the flow. Points are reassigned to new simulation particles upon split and merge events, and resampled if they fall outside their parent ellipsoid. We also reduce the number of points being rendered by detecting connected regions of high density and avoiding sampling points inside them. For illumination, the point normal is taken to be the gradient of the density plus a per-particle random

Example	Grid size	Sim/render particles	Time/frame
Explosion (Fig. 1)	$75 \times 50 \times 75$	242k / 4.7M	19.5 s
Hourglass (3)	$25 \times 50 \times 25$	44k / 2.4M	6.5 s
Impacts (5)	$50 \times 40 \times 100$	425k / 4.1M	18.3 s
Paddles (6)	$50 \times 100 \times 50$	199k / 2.7M	32.5 s
Materials (9)	$60 \times 40 \times 60$	415k / 4.4M	13.9 s
Comparison (11)	$75 \times 50 \times 75$	506k / 4.0M	11.6 s
Splash (10)	$50 \times 40 \times 50$	403k / 5.2M	6.1 s

Table 1: Performance measurements for all of our examples.

jitter. All scenes were rendered using Pixar’s RenderMan®.

The performance of our single-threaded implementation was measured on a 3.33 GHz Intel Core i7 machine with 5.8 GB of RAM. The detailed performance numbers are shown in Table 1. On average, the time per frame was spent as follows: 17% pressure, 35% friction, 33% particle update, and the remainder in other steps.

All of our simulations took between 6 and 33 seconds per frame on average, which is comparable to the performance of [Zhu and Bridson 2005] on similar PCs. The simulation time per frame is proportional to both the number of occupied grid cells and the speed of motion in the scene due to adaptive timestepping. In comparison to [Bell et al. 2005], their hourglass scenario with 110k particles and rigid body impact (“splash”) with 187k particles both take roughly 200 seconds per frame. Accounting for the scaling factor in the number of particles, our method running on the similar scenes and hardware is about one order of magnitude faster.

6 Conclusion

We have presented an efficient method for simulating free-flowing, dynamic granular materials using a continuum model. Our method supports a very general, diverse set of granular behaviors, including dispersing flow and two-way interaction with solid objects. We achieved this through a novel mathematical treatment of granular flow that robustly handles the complex interplay of internal stresses in the granular material. In addition, we developed a generalization of the particle-in-cell method that maintains a good distribution of particles in general compressible flows.

6.1 Limitations and future work

Our method models purely cohesionless behavior, which is an assumption satisfied by most dry granular materials. However, certain materials such as wet sand and soil show a finite amount of cohesion, which prevents the material from dispersing under small forces. Existing continuum models [Zhu and Bridson 2005; Lenaerts and Dutré 2009] which assume incompressibility can model such materials to an extent, but do not handle cases when

cohesive forces are overcome and the material breaks apart. An approach that can faithfully model granular materials with varying amounts of cohesion remains an open challenge.

Currently, we do not model the detailed effects of inter-grain interaction. These include subgrid-scale variations in motion, and interactions between grains of widely differing sizes. The addition of a model for such subgrid-scale interactions would allow for the simulation of even more complex scenes such as avalanches involving objects of many different shapes and sizes. Such an approach could also be useful in engineering applications.

To derive a tractable model for granular material dynamics, we have adopted a critical state assumption and neglected inter-particle interactions in the collisional regime of lower density. These assumptions allow many scenarios of interest to graphics applications to be simulated efficiently, but preclude modeling the more counter-intuitive behaviors such as formation of convection layers and surface waves upon shaking, and the Brazil nut and reverse Brazil nut effects. These surprising phenomena are beyond the scope of our current work, and remain as puzzling, challenging effects to model for the physics and mathematics communities.

Finally, the issue of sampling and rendering millions of grains from a continuum representation of granular material is an interesting research problem in itself, and can further enhance the visual appearance of our key contribution on simulation. Further independent investigation of this problem is valuable.

Acknowledgments: This work is supported in part by the Army Research Office, National Science Foundation, and RDECOM. The first author was supported by an Intel PhD fellowship.

References

- ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, ACM, 48.
- ALDUÁN, I., TENA, A., AND OTADUY, M. A. 2009. Simulation of high-resolution granular media. In *Proc. of Congreso Español de Informática Gráfica*.
- ARANSON, I. S., AND TSIMRING, L. S. 2001. Continuum description of avalanches in granular media. *Phys. Rev. E* 64, 2 (Jul), 020301.
- BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.* 26, 3, 100.
- BEHRINGER, R., JAEGER, H., AND NAGEL, S. 1999. Introduction to the focus issue on granular materials. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 9, 3, 509–510.
- BELL, N., YU, Y., AND MUCHA, P. J. 2005. Particle-based simulation of granular materials. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer animation*, ACM, 77–86.
- BIĆANIĆ, N. 2004. Discrete element methods. In *Encyclopedia of Computational Mechanics*, E. Stein, R. de Borst, and T. J. Hughes, Eds., vol. 1. Wiley, ch. 11.
- BRACKBILL, J. U., AND RUPPEL, H. M. 1986. Flip: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J. Comput. Phys.* 65, 2, 314–343.
- BRIDSON, R., AND MÜLLER-FISCHER, M. 2007. Fluid simulation. In *ACM SIGGRAPH 2007 courses*, 1–81.
- CHANCLOU, B., LUCIANI, A., AND HABIBI, A. 1996. Physical models of loose soils dynamically marked by a moving object. In *Computer Animation '96 Proceedings*, 27–35.
- DOSTÁL, Z., AND SCHÖBERL, J. 2005. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Comput. Optim. Appl.* 30, 1, 23–43.
- GOKTEKIN, T. G., BARGTEIL, A. W., AND O'BRIEN, J. F. 2004. A method for animating viscoelastic fluids. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)* 23, 3, 463–468.
- JOSSERAND, C., LAGRÉE, P.-Y., AND LHUILLIER, D. 2004. Stationary shear flows of dense granular materials: a tentative continuum modelling. *The European Physical Journal E: Soft Matter and Biological Physics* 14 (06), 127–135.
- KAUFMAN, D. M., SUEDA, S., JAMES, D. L., AND PAI, D. K. 2008. Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics (SIGGRAPH Asia 2008)* 27, 5, 164:1–164:11.
- LENAERTS, T., AND DUTRÉ, P. 2009. Mixing fluids and granular materials. *Computer Graphics Forum* 28, 213–218(6).
- LI, X., AND MOSHELL, J. M. 1993. Modeling soil: realtime dynamic models for soil slippage and manipulation. In *SIGGRAPH '93*, ACM, 361–368.
- LUCIANI, A., HABIBI, A., AND MANZOTTI, E. 1995. A multi-scale physical model of granular materials. In *Proceedings of Graphics Interface*.
- MILLER, G., AND PEARCE, A. 1989. Globular dynamics: A connected particle system for animating viscous fluids. *Computers and Graphics* 13, 3, 305–309.
- NARAIN, R., GOLAS, A., CURTIS, S., AND LIN, M. C. 2009. Aggregate dynamics for dense crowd simulation. *ACM Trans. Graph.* 28, 5, 1–8.
- ONOUE, K., AND NISHITA, T. 2003. Virtual sandbox. In *Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, 252.
- PLA-CASTELLS, M., GARCÍA-FERNÁNDEZ, I., AND MARTÍNEZ, R. J. 2006. Interactive terrain simulation and force distribution models in sand piles. In *Cellular Automata*. 392–401.
- QUECEDO, M., PASTOR, M., HERREROS, M. I., AND MERODO, J. A. F. 2004. Numerical modelling of the propagation of fast landslides using the finite element method. *International Journal for Numerical Methods in Engineering* 59, 6, 755–794.
- SIMO, J., AND HUGHES, T. 1998. *Computational Inelasticity*. Springer.
- SUMNER, R. W., O'BRIEN, J. F., AND HODGINS, J. K. 1999. Animating sand, mud, and snow. *Computer Graphics Forum* 18, 1, 17–26.
- ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. In *ACM SIGGRAPH 2005 Papers*, ACM, 965–972.
- ZHU, B., AND YANG, X. 2010. Animating sand as a surface flow. In *Eurographics 2010, Short Papers*.