CSCE 645 Research Project Proposal:
“Interesting Interactions with Granular Material”

Gabriel Stella

1 Proposal

The study of granular material has a long history. From gravel to medicinal powders, it is ubiquitous on the Earth and in human industry. Recently, computational simulations of granular matter have become more and more important. However, this poses unique challenges due to the particular behavior of this form of matter. The density of individual grains – thousands in just a spoonful of sand – makes complete simulation of even medium-scale scenes intractable. The particular behavior of granular matter also complicates things.

While matter is generally either thought of as solid, and approximated using rigid bodies, or as liquid, and simulated using grid-based motion systems, granular matter is not well-modelled by either of these: it is made up of solid granules, occasionally with interstitial fluids (e.g., in mud), but the sheer quantity of these granules causes fluid-like phenomena including flow, splashing, etc. Beyond this, behaviors unique to granular matter such as the formation of piles with a specific *angle of repose* are caused by microscopic inter-granule forces, which fluid solvers can only approximate by adding spring-like cohesion forces between neighboring particles to prevent slipping. Thus, there is a necessary tradeoff between efficiency and accuracy: simulating more complex interactions, or a larger or higher-resolution system, drastically increases computation time.

Indeed, though progress has come far in this domain, even the most obvious granular matter – sand – can still only be simulated in limited quantities and varieties. Sand comes in many forms, with granule sizes varying by orders of magnitude and varying shapes that can cause drastic changes in macroscopic behavior. Additionally, different types of sand have complex interactions with water, some forming mud and others becoming a material capable of supporting arches and other shapes. Current methods require extensive computation time to simulate even the most basic properties of sand at any useful scale; support for these additional, important phenomena is sadly lacking in the literature.

I have two general goals: first, incorporating additional properties of granular matter, such as the wetting of sand to produce new and interesting behaviors; second, designing structures that will allow for more efficient simulation of large quantities of sand without losing its essential properties. To keep these goals tractable in the span of a semester, I will begin by working with 2D particle-based simulations of sand. This will enable simpler visualization, more intuitive interaction, and simulation of larger-scale systems. Some more specific checkpoints are as follows, being conservative to account for setbacks:

1. By time of first update (Oct 28): implement a real-time 2D sand simulation environment using a rigid-body polysphere system, with support for some form of user interaction with the simulation

2. By time of second update (Nov 18): add additional support for a fluid layer, either water (which sticks to grains) or airflow (which pushes and pulls grains)

3. By time of final update (Dec 7): experiment with methods to compress the representation, e.g. grouping inactive grains and skipping computation where possible; investigate how well water can be integrated with this compression

2 Literature Review

Extensive research has been done on simulation of materials with varying properties. This includes rigid or ductile bodies, occasionally with fracturing [1]; fluid simulations for flame, water, etc [2–4]; and particle methods [5]. Each technique has some intrinsic properties that makes it better suited to certain materials (i.e., those that share the same properties that the simulation forces upon them) [6]. For example, particle methods easily calculate lossless advection; however, it is complex to calculate pressure and enforce incompressibility with this representation, and small timesteps are required to achieve stability [7].

One of the most important types of material in the world is *granular matter*. Due to its ubiquity, accurate and efficient simulation is of importance in many contexts: it can help predict natural disasters such as avalanches and earthquakes, design better manufacturing processes, and produce better animated graphics [4, 8]. Unfortunately, the complex phenomena associated with granular matter are difficult to simulate using existing methods. Microscopic inter-particle forces, which are essential in determining the macroscopic behavior of the material, are not well-approximated by grid-based methods, which scale well but lose fine details. On the other hand, it is extremely expensive to simulate *each particle* in a group of any interesting size. Hybrid methods help bridge the gap, but introduce their own new problems, such as unintentional energy dissipation.

2.1 2.5D Methods

Early methods focused on 2D or partially-3D models due to the limited computational power available at the time. One of the most popular approaches is to model sandy terrain using a heightmap. [9] first introduces a 2D heightmap (one horizontal axis and one height axis), where each vertical slice contains a trapezoidal column of sand. Calculations of soil forces are used to derive critical angles at which the top of a trapezoid will begin sliding into an adjacent slice. The model is then extended to a pseudo-3D system, where each cell is a triangular prism. The three sides of each cell are then treated as individual 2D simulations, and interpolation is used to get the height of each vertex (where 6 prisms meet).

In [10], a grid-based heightmap method is described, whose evolution is determined by a continuum treatment of the problem. To increase efficiency, they separate each cell into two populations: immobile grains and rolling grains. Threshold rules determine when immobile grains should begin moving (be dislodged) and when rolling grains should halt (become stuck). They also analyze the tendency of sand piles to cascade, finding that stationary sand piles can be *metastable* at angles steeper than the angle of repose (the steepest angle that flowing sand will maintain) but that eventually, at a *spinodal angle*, any infinitesimal perturbation will cause an exponential cascade (i.e., an avalanche) that resets the pile to stability. Between the angle of repose and the spinodal angle, perturbations cause only finite cascades rather than affecting the entire pile.[1] The methods in this paper are well-known in the literature, generally referred to as the BCRE method.

A similar heightmap method was applied in [2] to generate trails from objects moving over granular terrain, including varying types of sand and mud. It adds a pseudo-particle method, where a *cohesion* parameter allows the user to configure how particles stick to the underside of objects that rest on the terrain. Then, when the object lifts up, the particles slowly fall off under the influence of gravity and the mass is returned to the heightmap. A similar method is described in [8] which allows for parts of the heightmap to become disconnected, so that e.g. sand can be filled into a bucket and lifted out of their "virtual sandbox". A rendering technique is also provided to smooth the edges of the heightmap where it contacts objects.

In [11] a similar, though slightly more advanced, method is described. While an efficient heightmap is used to represent low-detail terrain away from user interaction, sand near dynamic objects is processed using

[1] This fact is well-known to anyone who spent their summers digging holes at the beach, and it's exciting that such a simple model can generate this behavior.

a particle system. This hybrid approach allows for larger masses of sandy terrain to be simulated while also giving better resolution and realism where necessary.

A more unique model is described in [12]. Here, they model deformable terrain as a *plastic carpet* – a 2D sheet of particles connected into a mesh, where the connections can deform as excessive force is applied. While their model is simple and low-resolution, it is effective at simulating the warping of loose terrain as heavy objects pass over (e.g., a vehicle's tires).

While heightmaps are able to simulate many interesting properties of granular materials, such as deformation under pressure, cascades, and even complete separation of groups of particles, there are many behaviors that are not suited for this approach. Granular matter experiences complex phenomena, such as near-constant flow rate due to bounded pressure in vertical containment; stress chains due to random shape and alignment of particles; splashing upon high-energy impact; etc. To properly simulate these processes, we need fully 3D methods. In the next section, 3D *Lagrangian particle-based* methods are discussed.

2.2 Particle Methods

Following the terms of classical field theories, rigid grid-based methods are called *Eulerian* while particle-based methods are called *Lagrangian*. Though Eulerian methods have found great success in fluid simulation, they are not perfectly suited to all of the phenomena (especially fine-grained details) observed in granular material. Particle-based methods, on the other hand, reflect with great accuracy the reality of granular materials. These are also variously called *finite element methods* or *discrete element methods*, since they model each element (i.e., grain) in the system. The primary downside of this approach is the computational cost of simulating the necessary number of particles – thousands are contained in just a spoonful of sand. As such, research has focused both on increasing accuracy and on finding ways to make simulations more efficient.

[13] gives an excellent description of two primary approaches to particle simulation. *Event driven* (ED) simulations model particles as rigid spheres with perfectly elastic conditions. However, processing a simulation of this type requires three conditions to ensure correctness and stability: binary collisions (if a particle is in a collision, it collides with only a single other particle at a time), short contact durations, and long intervals between contact events. Clearly, these three are all broken in even the most simple example of granular behavior, formation of a pile.

The *molecular dynamics* (MD) approach, on the other hand, models particles as soft spheres (or polyhedra), allowing a small amount of overlap during collisions. This is much more suitable to situations in which contacts are common and long-term, and [13] reports that MD reproduces interesting behaviors such as "heap formation, patterns in vibrating layers, force chains, and particle size segregation".

Sphere collision forces are separated into normal and tangent forces and handled separately. Normal forces determine how hard particles are pushes apart after collisions; rather than forcing perfect elasticity, they configure the system parameters so that high-energy impacts dissipate more energy than low-energy impacts (it is worth noting that their model can be tuned to produce a variety of energy-elasticity curves). Tangent forces are used to generate *shear friction* between grains, which is one of the most important factors determining the large-scale behavior of granular systems, including the particular angle of repose of a material.

They note that friction forces, while necessary, are not sufficient to produce this behavior: friction can only slow particles, not bring them to rest. While there are several approaches in previous literature, such as erasing tangent velocities below a certain threshold or attaching springs to keep contacting particles together, [13] takes the approach of using non-spherical granules that can interlock and stop movement entirely. They model their granules as a rigid collection of spheres; doing so simplifies translation of forces from the individual sphere (where collisions are calculated) to the entire grain. In this same way, entire rigid body objects can be represented as an enormous collection of statically-connected spheres; they describe a method to convert triangular meshes to this representation. To accelerate collision detection, the space is broken down into voxels and references to each particle are placed into the voxel it resides in and its neighboring voxels (to accelerate lookup at the cost of more expensive updates each time a particle crosses a voxel boundary).

Using this method, they are able to demonstrate interesting applications such as an hourglass with near-constant flow rate due to granular friction; an avalanche of particles pouring over rigid-body buildings; a ball being thrown into a pile of sand and causing a splash; and a heap of 1000 rings being dropped onto a series of poles, each ring using their particle-based rigid body representation.

A hybrid low-high resolution particle system based on this work is introduced in [14]. Here, two populations of particles coexist: a small population of "guide particles" and a larger population of normal particles. By computing simulation steps that do not require fine spatial resolution only over the guide particles and then interpolating the results over the entire population, they are able to achieve in one case an $11\times$ speedup over the original method with comparable qualitative results.

Another method for increased efficiency is to group together particles until granular behavior is necessary. The automatic subdivision approach described in [1] allows solid chunks of material to adaptively separate based on proximity to other objects or applied force, all the way to granular size, naturally becoming a particle-based simulation method based on the Bullet physics engine. This approach could also be used to coalesce grains that are inactive, saving on memory and computation.

While particle simulations are capable of recreating many of the behaviors of real granular materials, they are expensive to run, and typically progress much slower than real time. The work in [15] aims to bring sand simulation to interactive rates by moving computation onto powerful, parallel modern GPUs. In the tradition of [13], they model sand grains as a rigid group of four spherical sand particles (arranged on the vertices of a regular tetrahedron). In addition to producing interesting physical behaviors, this allows for natural rendering effects such as granule self-shadowing to help capture the particular look of particulate matter. On an NVIDIA 8800 GPU, they were able to achieve interactive rates with over 200k tetrahedral granules. They found that performance scaled linearly with granule count, perhaps due to the massive parallelization of the GPU, so the capabilities of this simulation method on modern hardware (almost a decade later) are likely impressive.

While extensive work has been done on particle simulation methods, there are still several open problems. First, handling cohesion forces (e.g., in wet sand) and, more generally, coupling between granular materials and fluids in the same simulation is not yet solved [13]. Second, while much of the literature has adopted a model of grain shape that treats each granule as a rigid collection of spheres, real sand particles come in a much wider variety of shapes and sizes, some of which are ellipsoidal while others are not round at all; each type produces widely varying macroscopic behaviors. It's possible that a more general description of granules could also enable better coupling with rigid bodies; the current particle-surface approximation method has several problems, including excessive roughness.

While particle methods excellently capture the dynamics of granular systems, they are not the only approach. Eulerian grid-based simulations, similar to fluid dynamics solvers, are capable of reproducing many of the properties of sand. However, hybrid *particle-in-cell* methods are even better suited to the task, dividing the work between both representations based on which is better suited. These especially help with simulation of large-scale fluid-like behavior, such as avalanches, while not losing as much of the fine detail as pure grid-based methods do.

2.3 Hybrid Methods

Hybrid models attempt to combine the advantages of both particle- and grid-based methods. These are sometimes called *particle-in-cell* (PIC) methods, though this term can also be used specifically to refer to the traditional PIC methods. These were first introduced many decades ago and originally found use in plasma simulation. The traditional PIC approach is to compute some parts of an update step using Lagrangian particles, use this to generate a grid, compute other parts on the grid, and use this to update the particles. This way, calculations that are better suited to a certain representation can be used more simply and effectively. However, this *transfer* causes significant energy dissipation in each step; when applied to granular matter, this leads to clumpy, wet-looking sand [16]. Many variations have been introduced to overcome this and other issues.

The *Fluid-Impicit-Particle* (FLIP) method adds a computation path that can skip the representation transfer, reducing energy loss. However, this drastically impacts the simulation's stability, resulting in wildly chaotic particle behavior.

In [7], a composite FLIP-PIC approach is described. These approaches are adapted to incompressible flow and applied to sand animation. Sand is modeled as a solid with *plastic yielding*, meaning that once an input force reaches a certain threshold (based on inter-grain friction), the entire mass permanently deforms. Technically, when sand transitions from stationary to flowing (i.e., during deformation), the volume slightly expands to allow grains to move across each other; for simplicity, they assume this effect is negligible. They also ignore possible elastic deformation of the grains themselves, since these are typically stiff particles.

Their most noticeable simplification was a change to the pressure terms that results in unbounded pressure increases as the height of a column of particles increases. Though this occurs naturally in fluids such as water and air, one of the most important properties in real granular materials is that when formed into a column, there is a fixed upper limit on the pressure within the pile. Thus, systems that depend on this behavior (e.g., the constant flow in an hourglass) cannot be simulated with this method.

In addition, they tuned cohesion and adhesion terms so that stable sand piles can form while still not being too "sticky"; without this, the particles would perpetually slip across the floor and container walls and settle like water. However, this method is actually *too* stable: it would not be able to properly simulate materials with their own cohesion, such as soil. Finally, they propose a new method for surface reconstruction by constructing a signed distance field based on a weighted average of nearby particles. Their new rendering method is compared to "blobbies", and is found in some cases to better fit the shape of a set of particles rather than just their density distribution.

The *material point method* (MPM) adapted FLIP to focus on solid bodies rather than fluids [17]. This method was applied in [6] to simulate snow and its many features; in fact, this paper describes the work done to simulate snow for the movie *Frozen*. They discuss the different types of snow, and how they affect simulation: fluffy vs. packed snow; dry vs dense snow; etc. They then demonstrate interesting results, such as: a snowball rolling down a hill, accumulating snow and destroying a snowman; snowballs exploding upon impact; a snow castle being destroyed by a flying ball; characters and vehicles interacting with snow; and several qualitative demonstrations of the physical effects of different material parameters (producing different types of snow).

Finally, the *affine particle-in-cell method* (APIC) is a recent and powerful new model for the simulation of granular materials. It aims to fix the problems present in PIC and FLIP. In PIC, energy is constantly dissipated by the simulation itself, leading to grains sticking in large clumps. In FLIP, the method that prevents energy dissipation instead introduces chaos; the particle behavior becomes unrealistically noisy, e.g. with particles shooting out of their proper path and becoming excessively mixed. In [16], the authors show that giving each particle a locally affine 3x3 matrix description of flow "reduces dissipation, preserves angular momentum and prevents instabilities". It does so by preventing dissipation during the particle-grid-particle transfer steps *without* the unstable transfer bypass that FLIP uses. In fact, their method is applicable to both fluids (by using typical PIC/FLIP fluid dynamics equations) *and* granular matter (by using the MPM [6]).

References

[1] M. Clothier, "3d voronoi subdivision for simulating destructible and granular materials," Ph.D. dissertation, Oregon State University, 2017.

[2] R. W. Sumner, J. F. O'Brien, and J. K. Hodgins, "Animating sand, mud, and snow," *Computer Graphics Forum*, vol. 18, 1999.

[3] H. J. Herrmann and S. Luding, "Modeling granular media on the computer," *Continuum Mechanics and Thermodynamics*, vol. 10, pp. 189–231, 1998.

[4] C. R. K. Windows-Yule, D. R. Tunuguntla, and D. J. Parker, "Numerical modelling of granular flows: a reality check," *Computational Particle Mechanics*, vol. 3, pp. 311–332, 2016.

[5] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, "Unified particle physics for real-time applications," *ACM Trans. Graph.*, vol. 33, no. 4, Jul. 2014. [Online]. Available: https://doi.org/10.1145/2601097.2601152

[6] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, "A material point method for snow simulation," *ACM Trans. Graph.*, vol. 32, no. 4, Jul. 2013. [Online]. Available: https://doi.org/10.1145/2461912.2461948

[7] Y. Zhu and R. Bridson, "Animating sand as a fluid," *ACM Trans. Graph.*, vol. 24, no. 3, p. 965–972, Jul. 2005. [Online]. Available: https://doi.org/10.1145/1073204.1073298

[8] K. Onoue and T. Nishita, "Virtual sandbox," in *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, ser. PG '03. USA: IEEE Computer Society, 2003, p. 252.

[9] X. Li and J. M. Moshell, "Modeling soil: realtime dynamic models for soil slippage and manipulation," *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993.

[10] J.-P. Bouchaud, M. E. Cates, J. R. Prakash, and S. F. Edwards, "Hysteresis and metastability in a continuum sandpile model," *Phys. Rev. Lett.*, vol. 74, pp. 1982–1985, Mar 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.74.1982

[11] C. Leach, "A gpu-based level of detail system for the real-time simulation and rendering of large-scale granular terrain," Master's thesis, University of Cape Town, 2014.

[12] B. Chanclou, A. Luciani, and A. Habibi, "Physical models of loose soils dynamically marked by a moving object," in *Proceedings Computer Animation '96*, 1996, pp. 27–35.

[13] N. Bell, Y. Yu, and P. Mucha, "Particle-based simulation of granular material," 01 2005, pp. 77–86.

[14] I. Alduan, n. Tena, and M. Otaduy, "Simulation of high-resolution granular media," 09 2009.

[15] J.-P. Longmore, P. Marais, and M. M. Kuttel, "Towards realistic and interactive sand simulation: A gpu-based framework," *Powder Technology*, vol. 235, pp. 983–1000, 2013.

[16] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The affine particle-in-cell method," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015. [Online]. Available: https://doi.org/10.1145/2766996

[17] D. Sulsky, S.-J. Zhou, and H. L. Schreyer, "Application of a particle-in-cell method to solid mechanics," *Computer Physics Communications*, vol. 87, no. 1, pp. 236–252, 1995, particle Simulation Methods. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010465594001707