

**A Hybrid Continuum and Discrete Element Method for  
Granular Media Modeling**

by

Maytee Chantharayukhonthorn

Submitted to the Department of Mechanical Engineering  
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

© Massachusetts Institute of Technology 2019. All rights reserved.

Author .....  
Department of Mechanical Engineering  
May 19, 2019

Certified by .....  
Kenneth Kamrin  
Associate Professor  
Thesis Supervisor

Accepted by .....  
Evelyn Wang  
Chairman, Department Committee on Graduate Theses



**A Hybrid Continuum and Discrete Element Method for Granular Media  
Modeling**

by

Maytee Chantharayukhonthorn

Submitted to the Department of Mechanical Engineering  
on May 19, 2019, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Mechanical Engineering

**Abstract**

In this thesis, I designed and implemented a compiler which performs optimizations that reduce the number of low-level floating point operations necessary for a specific task; this involves the optimization of chains of floating point operations as well as the implementation of a “fixed” point data type that allows some floating point operations to be simulated with integer arithmetic. The source language of the compiler is a subset of C, and the destination language is assembly language for a micro-floating point CPU. An instruction-level simulator of the CPU was written to allow testing of the code. A series of test pieces of codes were compiled, both with and without optimization, to determine how effective these optimizations were.

Thesis Supervisor: Kenneth Kamrin  
Title: Associate Professor



## Acknowledgments

This is the acknowledgements section. You should replace this with your own acknowledgements.



# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Granular Media Modeling . . . . .	14
1.1.1	Discrete Methods . . . . .	15
1.1.2	Continuum Models . . . . .	17
1.1.3	Related Hybridization Work . . . . .	19
<b>2</b>	<b>Discrete Element Method</b>	<b>21</b>
2.1	DEM Model . . . . .	22
2.2	DEM Algorithm . . . . .	23
<b>3</b>	<b>Continuum Model</b>	<b>27</b>
3.1	Hyperelastic-Plastic Model . . . . .	27
3.2	Hypoelastic-Plastic Model . . . . .	30
3.3	Material Point Method . . . . .	33
3.3.1	MPM Algorithm Overview . . . . .	34
3.3.2	MPM Formulation and Discretization . . . . .	35
3.3.3	Relevant Notes on MPM . . . . .	40
<b>4</b>	<b>Hybridization</b>	<b>43</b>
4.1	Hybridization Overview . . . . .	43
4.2	Reconciliation Zone Splitting . . . . .	45
4.3	Coupling Constraints . . . . .	47
4.3.1	Hybrid Coupling Discretization . . . . .	48
4.3.2	Nodal Coupling . . . . .	49
4.4	Domain Decomposition . . . . .	50

4.4.1	Oracle . . . . .	50
4.4.2	Homogenization and Enrichment . . . . .	52
4.4.3	Layered Hybridization . . . . .	54
4.5	Summary of Hybrid Algorithm . . . . .	54
<b>5</b>	<b>Enrichment and Homogenization Improvements</b>	<b>59</b>
5.1	Volume Change and Mass Conservation . . . . .	60
5.1.1	Mass Ledger . . . . .	62
5.1.2	Mass Ledger Implementation . . . . .	64
5.2	Packing Schemes . . . . .	65
5.2.1	Random Packing . . . . .	66
5.2.2	Grid Packing . . . . .	67
5.2.3	Circle Sweep . . . . .	69
5.2.4	Two Point Tangent . . . . .	70
5.2.5	Packing Scheme Results . . . . .	70
5.3	Hypoelastic Pressure Update . . . . .	70

# List of Figures

1-1	Flowing hourglass displaying three distinct phases, with entire hourglass (a) and closeup of bottom region (b) . . . . .	15
1-2	Aftermath of a landslide in La Conchita California. . . . .	17
2-1	Two disks in contact with relevant properties labeled for DEM. . . . .	22
3-1	Schematic of a single timestep in MPM. . . . .	35
4-1	Diagram of a hybrid simulation. . . . .	43
4-2	Blurred Density: (Left) The reference domain $\Omega$ of an object with density $\rho(\mathbf{x})$ . Mass density is colored in blue. (Right) A partition of unity of the density mediated by a weight function $w(\mathbf{x}, t)$ . . . . .	45

4-3 Initialization of a hybrid simulation: (A) We begin with a collection of DEM grains. (B) We next locate a level set corresponding to a given low density, here denoted as a black line. (C) Across the domain, we compute the distance to the density threshold, indicated by lines in lighter shades of red as the distance increases. (D) We select a user-tunable distance to the density level-set that serves as the center of the hybrid "reconciliation" zone. We denote this critical distance as a solid black line. (E) We extend the hybrid zone along the distance field by a given half-width in each direction, indicated by dotted lines. This hybrid reconciliation zone between the dotted lines defines a zone where the DEM system will be coupled to the continuum system. We homogenize the velocity and stress for use in step (G). (F) We delete all discrete grains that fall within the inner boundary of the reconciliation zone. (G) We run the "avoid-a-void" algorithm of Yue et al. [38] from the outer boundary in to populate the region with material points. The material point states are determined using the homogenized velocity and stress computed in step (E) . . . . .	51
5-1 Mechanisms of deviations away from exact mass conservation. . . . .	60
5-2 Evolution of a hybrid flow down a chute showing mass loss. . . . .	61
5-3 Mass fluxes in a hybrid system and the construction of a ledger. . . . .	63
5-4 Discrete particle entering a pure continuum regime. . . . .	64
5-5 Mass over time for random packing. . . . .	66
5-6 Non-optimal grain placement leading to inability to reach target packing fraction. . . . .	67
5-7 Chute flow example zoomed into hybrid zone. . . . .	68
5-8 Grid enrichment scheme. . . . .	68
5-9 Circle sweep scheme. . . . .	69

# List of Tables

5.1 Mass ledger contributions . . . . .	65
---	----



# Chapter 1

## Introduction

Modeling is a game of balance. Tractability and reasonable solution times fight against the physical realities of vast length and time scales. The assumptions one makes when formulating a model directly impact the solution methods that can be brought to bear to the problem at hand. There is almost always guaranteed trade-offs between the level of simplification of a model and the amount of time needed to solve that model.

Granular media present an interesting intermediary between the world of the discrete and continuum. While often times they are studied in contexts where continuum approximations are appropriate (i.e. geological), their behavior at much smaller length scales (a bucket of sand at the beach or flow through an hour glass) are still of great interest. However at those more everyday scales, the length scale of a single grain of sand is a relatively large proportion of the scale of the entire problem, and thus cannot be ignored. And even at the larger aforementioned geological scales, the initiation of an earthquake, for example, still relies on individual grains of sand slipping and deforming against each other.

The great challenge, then, is to formulate a model that can capture length scale effects but still have enough simplifying assumptions to make the problem solvable with efficient methods. This however seems to fly in the face of the modeling trade-off previously discussed; it is nearly impossible to have a single model that allows for fine-scale resolution and yet ignores those small length scales to become efficiently solvable. The solution proposed in this work addresses this issue by ignoring the "single" part of the "single model" clause, and instead hybridizes two distinct models to yield the two distinct attributes desired: resolution of length scale while maintaining efficient solvability. It is noted that the current work

focuses on cohesionless granular systems, and thus approximate dry granular systems with no sources of attraction, like liquid bridges or electrostatic charges.

The work is structured as follows. The remainder of chapter 1 puts the current work in the greater academic context and discusses prior work on the modeling of granular media. Hybridization models in and out of granular media contexts are also discussed.

Chapter 2 discusses the discrete element method used and some details of its algorithmic solution. While the level of detail presented may seem overly exhaustive, it provides important context for where the hybridization technique interfaces with the discrete element method.

Chapter 3 discusses the continuum models used. The method used to solve these models, the Material Point Method, is also discussed in detail.

Chapter 4 introduces the hybridization technique. Its goal, formulation, and solution are discussed.

Chapter 5 shows examples of the hybridization technique at work, with comparisons to the discrete and continuum model solutions, as well as to literature.

Finally chapter 6 concludes the thesis and discusses future work.

## 1.1 Granular Media Modeling

The ubiquity of granular media in everyday life cannot be understated. We walk on it on trails, drive over it on roads [33], ingest it in our pharmaceuticals, and eat it in our meals. Slightly less directly, granular media is second only to water for the type of material most commonly handled in industry [25]. Despite this ubiquity however, a comprehensive model that can capture the behavior of granular media remains elusive.

While much of the difficulty stems from the length and time scale problems mentioned before, granular media is also unique from many other materials in its ability to transition between different states. This is clearly evident in a flowing hourglass, as shown in Figure 1-1. At the bottom of the hourglass, settled sand acts as a solid, able to support compressive stress without flowing. At the top of the static pile is a region of grains flowing over the static region, acting like a liquid. In between the top and bottom of the hourglass the grains flow much like a dilute gas, with no cohesive structure and interacting via collisions.

Different models and solution techniques are able to capture granular behavior in a given



(a) data a



(b) data b

Figure 1-1: Flowing hourglass displaying three distinct phases, with entire hourglass (a) and closeup of bottom region (b)

state, though of course with trade-offs in accurately capturing behavior in other states. A summary of different methods are thusly discussed.

### 1.1.1 Discrete Methods

Perhaps the most straightforward way one could model a system of granular material is to model the grains themselves. Methods that model individual grains and the interactions between them fall under the umbrella of discrete methods. While this can be expensive for very large systems (greater than approximately 50,000 particles per core given current CPU capabilities), the one-to-one correspondence of a single simulated grain to a physical grain can produce accurate results. Discrete element methods can largely be broken down into two classes: penalty based methods and contact dynamics.

Penalty methods, as their names suggest, penalize the overlap of particles with some type of force that is a function of that overlap. The discrete element method (note that in some literature the term "discrete element method" is used to denote the larger class of what is here termed "discrete methods"), first formulated by Cundall and Strack, still enjoys much use due to its simplicity and accuracy [6]. Even within the confines of the simplicity of the proposed method though, great generalizability can be realized by having a free choice of the penalty function. The advantages of course come at some cost. For example, a major drawback is that, depending on the choice of penalty function, multiple material parameters may need to be fit to experiment. The material parameters themselves may then

put constraints on the computational solve time. To concretely demonstrate this point, many penalty models have some notion of an elastic parameter that must be tuned. However, most individual grains of sand are fairly stiff, with **FINDREFERENCE** bulk moduli on the order of  $49 GPa$  and densities on the order of  $2000 kg/m^3$ . These properties, combined with the small size of many grains ( 0.1 mm in diameter), result in a large wave speed through the material that travels a small distance, and this must be resolved within a given time step. Implicit methods of course exist to alleviate this issue somewhat, but those come with the usual drawbacks of additional computational overhead elsewhere.

Contact dynamics on the other hand treats grains as completely rigid and allow no overlaps. They are then formulated as optimization problems, and more specifically, mixed linear complementarity formulations, minimizing some potential with a no overlap constraint. While the question of material properties is then largely avoided in these methods, the introduction of friction and other properties is much less straightforward than in the formulation of discrete element methods. The lack of material properties is also a double-edged sword of sorts, as while infinitely stiff grains are often a better approximation of a system of grains than computational grains that are extremely soft, the reality is that grains do have a finite, though large, stiffness. Capturing that finite stiffness and its consequences, such as a finite wave speed and non-negligible grain deformation, can be crucial in some applications. These properties are in fact important for the hybrid scheme, and will be discussed later.

A key characteristic of both classes of discrete methods is that they can easily capture all phases of discrete matter. If compressed by exterior forces or boundaries, they act like a solid, able to support load through the creation of force chains, much like physical granular media. The removal of these forces and boundaries, and/or the introduction of shear forces, causes grains to flow over other grains in a liquid-like fashion. Pouring a system of discrete grains will see the grains separate, capturing a granular gas.

Another property of discrete element methods is that they are able to elucidate particle level properties and dynamics that are difficult to gather from experiment. Photoelastic disks can be used to investigate force chains, such as in the pioneering work of Behringer et al and continued by the likes of Daniels et al **FINDREFERENCE**. However these photoelastic disks are made of relatively soft polymers and are mostly used to investigate 2D arrangements of disks, and not 3D arrangements of spheres. On the other hand, discrete element methods calculate inter-particle forces out of necessity and can thus report quantitative data for these

contacts. Expanding from 2D to 3D is also a straightforward process, and every contact in a 3D system can be easily obtained. The dynamics of every single grain in a system modeled with discrete methods can also be tracked and studied, which can be done in experiment, but only with much difficulty and cost, i.e. methods such as X-ray tomography and CT scans [FINDREFERENCE](#). While computational expenses can limit the size of simulated systems, physical limitations of scanning equipment can limit the size of an experimental system that can be studied, greatly hampering one of the key advantages of experiments over simulations in granular media: scale.

Thus despite the drawbacks of discrete element methods, they are still popular and widely used in congruence with, and sometimes in the place of, physical experiment. The ability to accurately capture grain-scale level dynamics, and to obtain quantitative data for every grain and contact, means that they can effectively be treated as computational "ground truth" for simulated granular systems. When accuracy in a simulation is needed, discrete element methods can be used with confidence, at least compared with other methods.

### 1.1.2 Continuum Models



Figure 1-2: Aftermath of a landslide in La Conchita California.

As stated before, for large (what constitutes "large" varies from system to system, though 20 grain diameters per continuum element dimension is used as a rule of thumb [FINDREFERENCE](#)) granular systems, one can ignore the fact that there are individual grains of sand, treat the system as a continuous granular medium, and retain many physical properties of the system. A classic example of this can be seen in Figure 1-2, which shows the aftermath

of a landslide. A useful feature of the shown landslide is that a road can be seen that cuts through the hill, and acts as a deformation marker. The deformed shape is reminiscent of Poiseuille flow, suggesting that the moving bulk is well represented by a continuum.

Continuum theory applied to granular media in fact goes back more than 50 years, with the pioneering work of Coulomb who proposed a relation between shear stress, pressure, and a coefficient of friction in a granular continua, very similar in form to Coulomb friction [FINDREFERENCE](#). Since that initial proposal, granular continuum theory has been greatly expanded upon. The incorporation of additional complexity displayed in physical granular systems into the continuum theory have resulted in models that capture behavior such as critical-state, and anisotropy [27][7].

Much work on granular continuum theory has been conducted in the fields of civil engineering and soil mechanics, where understanding the behavior of granular systems under load is crucial [FINDREFERENCE](#). There is thus a large body of work on granular media in a solid phase, and continuum modeling of grains in this state is well understood. Granular gases too have been well investigated, with kinetic theory being effectively used to understand granular systems in this state.

The "liquid" flowing phase of granular materials has been much more difficult to model. It was only recently that a seminal study conducted by GDR MiDi suggested a possible model for flowing granular systems [11]. The rheological model posited, commonly referred to as the  $\mu(I)$  relation ( $\mu$  of  $I$ ), suggests a yield condition similar to that suggested by Coulomb nearly half a century ago, but with a friction coefficient dependent on a nondimensional inertial number,  $I$ , which describes the ratio of inertia to confining pressures in a granular system. Further work by Jop and De Cruz provided empirical relations between the friction coefficient  $\mu$  and  $I$  [15][FINDREFERENCE](#). Further extensions of this model have since been proposed, and work continues to this day on clarifying the high and low inertia number bounds of the  $\mu(I)$  relation.

All of the previously described models, while increasingly complex, still retain some simplicity in the sense that they are all local models. No length-scale is introduced, and thus no non-local effects are captured. This deficiency has been recently addressed by the work of Kamrin et al, who have proposed a non-local continuum model and thus introduce a notion of a length scale back into the continuum model [17]. At first glance this seems provides a possible solution to the beginning stated problem of capturing length-scale effects

while retaining the ability of efficient continuum equation solving methods. However, much additional work must be done to completely characterize these non-local models, and thus for now cannot be relied upon to have the fidelity of discrete methods which capture length-scale effects by their very nature.

### 1.1.3 Related Hybridization Work

Quasi-Continuum and Arlequin-type methods have been explored primarily for crystalline solids, to expedite otherwise lengthy atomistic simulations by hybridizing with a crystal plasticity continuum model in zones where atomistic refinement is not needed [35, 31, 29, 39, 8]. The idea of hybridizing discrete-particle and continuum approaches to simulate granular media is in its infancy, with only initial work done to show the validity of communicating mechanics between discrete grains and finite-element facets [37]. Recent work has explored when continuum and discrete treatments are simultaneously accurate [26, 16, 18], including an Arlequin-type method that couples statically-defined regions of a discrete element (DEM) simulation to the interior of a continuum FEM-based simulation to enrich stress fields around drill tips, for instance [36]. We build on and extend these ideas to target regimes in which enriched degrees of freedom are required at surfaces, and where the boundary between continuum and discrete regimes evolves dynamically.

In the granular physics and graphics literature, lower-level ideas have been tried where instead of implementing a general continuum model, the user imposes kinematic constraints to the particle motion in certain regions, often chosen based on experience with the problem at hand. The graphics literature has explored freezing rigid bodies that are sufficiently stationary [32]. Similar techniques have been proposed to accelerate the generation of granular packings for industrial applications [22]. In common granular setups such as rotating tumblers and growing sand piles, semi-empirical models can be used to guess zones of rigid material, and grains in these zones can be removed from the discrete update [21, 14, 40, 4]. Holladay et al.[13] carve out interior regions of granular materials moving at constant velocities and replace these groups of grains with meshes, but this method does not homogenize over rigidly rotating regions or over shear flows, and as the paper notes, can lead to volume loss. These ideas have been developed further in follow-up work [12, 23]. These methods make no claims as to the accuracy of the techniques for science and engineering applications, and have not yet demonstrated stable granular flows.



## Chapter 2

# Discrete Element Method

As described generally in Chapter 1, the discrete element method (DEM) models a system of grains by modeling each grain as a separate entity and calculates the dynamics of each grain by integrating what essentially amounts to  $\Sigma \mathbf{f} = m\mathbf{a}$  through time. In 2D, each grain is modeled as a disk with a radius  $r$ , parameterized by three degrees of freedom: two for the center of mass position of the disk (held by a position vector  $\mathbf{x}_d \subset \mathbb{R}^2$ ), and a third for the rotation of the disk relative to some rest state. The dynamics are captured by another three parameters: two for the center of mass velocity and a third for the angular velocity about the center of mass. In 3D this representation is generalized to a sphere, again with radius  $r$  and six degrees of freedom: three for the center of mass ( $\mathbf{x}_d \subset \mathbb{R}^3$ ) position and three for the angles that describe grain orientation. The dynamics analogously generalize to six parameters, with three for center of mass velocity and three for angular velocities. For a system of  $K$  particles, the degrees of freedom of all particles can be concatenated into a single degree of freedom list, the generalized coordinate vector  $\mathbf{q}_d$ . In 2D,  $\mathbf{q}_d \subset \mathbb{R}^{3K}$  and in 3D,  $\mathbf{q}_d \subset \mathbb{R}^{6K}$ . A generalized velocity vector,  $v_d$  can be similarly defined, with  $\mathbf{v}_d \subset \mathbb{R}^{3K}$  in 2D and  $\mathbf{v}_d \subset \mathbb{R}^{6K}$  in 3D. Momentum balance for the whole granular system can then be summarized with

$$M_d \mathbf{a}_d = \mathbf{f}_d(\mathbf{q}_d, \mathbf{v}_d, t)$$

where  $M_d \subset \mathbb{R}^{3K \times 3K}$  (2D) and  $M_d \subset \mathbb{R}^{6K \times 6K}$  (3D) is the mass matrix,  $\mathbf{a}_d \subset \mathbb{R}^{3K}$  (2D) and  $\mathbf{a}_d \subset \mathbb{R}^{6K}$  (3D) is the generalized acceleration vector, and  $\mathbf{f}_d$  is the force vector that encapsulates all internal and external forces of the system. The evolution of the system

configuration can then be described with

$$\dot{\mathbf{q}}_d = \hat{\mathbf{q}}_d(\mathbf{q}_d, \mathbf{v}_d, \mathbf{a}_d)$$

where  $\hat{\mathbf{q}}_d$  is a function that encapsulates configuration updates.

## 2.1 DEM Model

The construction of  $\mathbf{f}_d$ , and specifically the contact model that goes into  $\mathbf{f}_d$ , has been the source of much work. Popular contact models include Hertzian contact and linear spring-dashpot systems, the latter of which we use [FINDREFERENCE](#). While Hertzian contact in theory accounts for a nonlinear penalty force with respect to penetration depth due to geometric considerations not present in the simple Hookean spring model, the simplicity of the Hookean spring model along with its acceptable accuracy from literature motivate the latter's use in the current study [FINDREFERENCE](#).



Figure 2-1: Two disks in contact with relevant properties labeled for DEM.

In the current work, the contact force,  $\mathbf{f}_c$ , is a linear combination of a normal contact force  $\mathbf{f}_n$  and tangential contact force  $\mathbf{f}_t$ , such that simply  $\mathbf{f}_c = \mathbf{f}_n + \mu\mathbf{f}_t$ , where  $\mu$  is the coefficient of friction.  $\mathbf{f}_d$  more concretely is

$$\mathbf{f}_n = k_n d \mathbf{n} - \gamma_n \mathbf{v}_n$$

where  $k_n$  is the Hookean spring constant in the normal direction,  $d$  is the penetration depth

between the two disks,  $\mathbf{n}$  is the contact normal unit vector,  $\gamma_n$  is the normal damping coefficient, and  $\mathbf{v}_n$  is the normal component of the relative velocity between the two disks. Similarly, the tangential contact force is given by

$$\mathbf{f}_n \mathbf{t} = k_t \Delta s - \gamma_t \mathbf{v}_t$$

where  $k_t$  is the spring constant in the tangential direction,  $\gamma_t$  is the tangential damping coefficient, and  $\mathbf{v}_t$  is the tangential component of the relative velocity. Friction is captured in this model by requiring that

$$\mathbf{f}_t \leq \mu \mathbf{f}_n$$

which is accomplished by adjusting  $\Delta s$ . For a given enduring contact over time,  $\Delta s$  for that contact is the time integral of the tangential relative velocity during that contact.  $\Delta s$  is then rescaled so that it  $\mathbf{f}_t$  falls within the friction cone determined by  $\mu \mathbf{f}_n$ .

With the given spring-dashpot system, a coefficient of restitution (COR)  $e$  can be tuned as a function of the model parameters. Given a desired  $e$  and a normal spring coefficient  $k_n$ ,  $\gamma_n$  is determined as

$$\gamma_n = \sqrt{mk_n}(-2 \log e) / \sqrt{2(\pi^2 + \log e^2)}$$

where  $m$  is the mean mass of a grain [18]. Note that the use of a "mean" mass is due to the fact that in all of the simulations conducted in this study, a slight polydispersity in granular radii is used with a single density for all particles, resulting in a mass distribution. This is done to better match real shape distributions in a granular system, and to avoid crystallization that commonly arises in monodisperse systems. The choice of  $k_n$  and other material parameters is further explained for specific simulations later in the study, but in general is chosen to be as stiff as possible while still retaining a reasonable cost per time step, with a timestep usually on the order of  $10^{-6}$  seconds.

## 2.2 DEM Algorithm

The DEM code used in this study was built completely in house, though is similar in general algorithmic structure to many DEM codes that exist, such as LAMMPS or LIGHTS [FINDREFERENCE](#). Thus for transparency as well as necessity when later explaining the hybrid algorithm, the structure of the used DEM code is discussed.

---

**Algorithm 1** Overall\_DEM\_Algorithm

---

```
1: Broad_Phase_Collision_Detection
2: for  $i = 0 \dots num\_possible\_collisions$  do
3:   Narrow_Phase_Collision_Detection
4: end for
5: Collision_Update
6: for each_collision_type do
7:   Update_Properties
8:   Integrate_Δs
9: end for
10: Force_Update
11: for each_collision do
12:   Calculate_Penalty_Force
13:   Correct_Δs
14:   Add_Force_To_Contact_Grains
15: end for
16: Time_Integration
```

---

The *Broad\_Phase\_Collision\_Detection* creates an axis-aligned bounding box (AABB) around each grain, and checks the intersection of those AABBs with a background grid. Each grid cell then has a vector of AABBs that intersect it, with each combination of AABB in that vector constituting a possible collision. All possible collisions are then looped over for an actual collision detection (*Narrow\_Phase\_Collision\_Detection*) and any real collisions are added to a vector of actual collisions for a given type. Collision types include, for example, circle\_circle collisions for grains in contact, or circle\_plane collisions for grains in contact with a rigid plane. The list of all collisions for every collision type are then looped over, and properties such as penetration depth and  $\Delta s$  are updated. With this information, a penalty force is calculated at every contact according to the model presented in 2.1. With these forces, an explicit Forward Euler update is used to numerically integrate the velocity of the grains, which is then used to integrate the position of the grains.

Though dry, cohesionless grains are the focus of the current work, it is noted that the DEM framework allows for a simple extension for cohesive grains. A new collision type can be defined that allows for tracking of grain interactions at a distance. As will be explained later, some initial work has in fact been done on this, by tracking liquid bridges that provide a source of cohesion in the system, in order to extend the hybrid method for cohesive systems.

As a final note, the DEM code is an extension and modification of the SCISIM code developed by Smith for contact dynamics [FINDREFERENCE](#). In fact, as will be later

discussed, that contact dynamics code was first used as the discrete method of choice for the hybrid project. However, the explicit penalty method was determined to better suit the needs of hybridization.



# Chapter 3

## Continuum Model

### 3.1 Hyperelastic-Plastic Model

We model continuum granular regions as an elastoplastic material with a Drucker-Prager plastic yield criterion. The evolution of the system is governed by the conservation of momentum

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f}_{ext}, \quad (3.1)$$

and the conservation of mass

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0, \quad (3.2)$$

where  $\boldsymbol{\sigma}$  denotes the Cauchy stress tensor,  $\frac{D\cdot}{Dt}$  denotes the material derivative, and  $\mathbf{f}_{ext}$  denotes any external body forces (e.g. gravity). We use a multiplicative decomposition of the deformation gradient  $\mathbf{F} = \mathbf{F}^e \mathbf{F}^p$  where  $\mathbf{F}^e$  and  $\mathbf{F}^p$  denote the elastic and the plastic component of the deformation gradient, respectively. We denote the left Cauchy-Green strain by  $\mathbf{b} = \mathbf{F}\mathbf{F}^T$ .

Under small strains, the material behaves elastically. We employ a simple strain energy density to model the elastic behavior. In 2D the strain energy and Kirchhoff stress are given

by

$$W = \frac{\kappa}{2} \left[ \frac{1}{2}(J^2 - 1) - \ln J \right] + \frac{1}{2}\mu(Tr[\bar{\mathbf{b}}^e] - 2), \quad (3.3)$$

$$\tau = \frac{\kappa}{2} (J^2 - 1) \mathbf{I} + \mu dev[\bar{\mathbf{b}}^e], \quad (3.4)$$

where  $\bar{\mathbf{b}}^e = \det(\mathbf{b}^e)^{-1/2} \mathbf{b}^e$  is the volume preserving elastic left Cauchy-Green strain and  $J = \det(\mathbf{F})$ . In 3D, the strain energy density becomes  $W = \frac{1}{2}\kappa \left[ \frac{1}{2}(J^2 - 1) - \ln J \right] + \frac{1}{2}\mu(Tr[\bar{\mathbf{b}}^e] - 3)$  while  $\bar{\mathbf{b}}^e = \det(\mathbf{b}^e)^{-1/3} \mathbf{b}^e$ .  $\kappa$  and  $\mu$  are the bulk and shear moduli respectively of the material.

To allow the granular medium to separate, we consider the medium to be one-sided in the following sense: the granular medium can resist compression, but not extension. We model this effect with a free-flow mode, similar to Dunatunga et al. [10]. When we detect that the material is in extension, indicated by  $\det[\mathbf{b}^e] > 1$ , we project the strain to  $\det[\mathbf{b}^e] = 1$  by assigning  $\mathbf{b}^e \leftarrow \det(\mathbf{b}^e)^{-1/2} \mathbf{b}^e$  ( $\mathbf{b}^e \leftarrow \det(\mathbf{b}^e)^{-1/3} \mathbf{b}^e$  in 3D). The net result is zero pressure when the material is under extension.

We model plastic flow with the Drucker-Prager yield condition

$$\Phi = s - \alpha p \leq 0, \quad (3.5)$$

where  $s = \|dev[\tau]\|_F$  is the magnitude of the shear stress (recall that  $dev[\mathbf{X}] = \mathbf{X} - \frac{1}{2}Tr[\mathbf{X}]\mathbf{I}$ ),  $p = -\frac{1}{2}Tr[\tau]$  is the pressure ( $dev[\mathbf{X}] = \mathbf{X} - \frac{1}{3}Tr[\mathbf{X}]\mathbf{I}$  and  $p = -\frac{1}{3}Tr[\tau]$  in 3D), and  $\alpha$  controls the angle of repose. It often proves useful to express the yield in terms of strain, in which case we find that  $\Phi = \mu \|dev[\bar{\mathbf{b}}^e]\|_F + \alpha \frac{\kappa}{2} (J^2 - 1)$ . The Drucker-Prager yield criterion corresponds intuitively with the notion of Coulomb friction in the discrete setting; the shear stress (continuum analog of force in the tangential plane) is capped at the magnitude of the pressure (continuum analog of the normal force) scaled by a constant  $\alpha$  (continuum analog of  $\mu$ ). When  $\Phi \leq 0$ , the material behaves elastically. When  $\Phi > 0$ , however, the material sustains plastic shape change. Crucially, a Drucker-Prager treatment of yield allows us to tune the angle of repose of a pile to match that of a discrete element simulation.

We model the granular medium as perfectly plastic: all excess yield is immediately converted into plastic deformation. We discretize the elastoplastic update rule for the strain

$\dot{\mathbf{b}}^e = \nabla \mathbf{v} \mathbf{b}^e + \mathbf{b}^e \nabla \mathbf{v}^T + \mathcal{L}_v \mathbf{b}^e$  with the return mapping method [30]. The result is a predictor-corrector style update for the strain. We first update  $\mathbf{b}^e$  through  $\dot{\mathbf{b}}^e = \nabla \mathbf{v} \mathbf{b}^e + \mathbf{b}^e \nabla \mathbf{v}^T$ , ignoring any plastic flow. This predicted strain can now violate the yield condition. To remove yield excess from  $\mathbf{b}^{e,*}$ , where  $*$  denotes some violating state, we impose two constraints on the plastic flow to project the material to a violation-free state. First, the plastic granular flow should conserve volume, and the projected strain  $\text{det}[\mathbf{b}^{e,p}]$  should satisfy  $\text{det}[\mathbf{b}^{e,p}] = \text{det}[\mathbf{b}^{e,*}]$ . Second, to satisfy the yield condition, we seek  $\Phi(\tau(\mathbf{b}^{e,p})) = 0$ . We consider the flow to be in the direction of the shear and we decompose the projected strain as  $\mathbf{b}^{e,p} = \lambda_1 \mathbf{I} + \lambda_2 \text{dev}[\text{det}[\mathbf{b}^{e,*}]]$ .

Expanding the constraint  $\Phi(\tau(\mathbf{b}^{e,p})) = 0$ , we find:

$$\begin{aligned}\Phi(\tau(\mathbf{b}^{e,p})) &= \mu \|\text{dev}[\mathbf{b}^{e,p}]\|_F + \alpha \frac{\kappa}{2} (J^2 - 1) \\ &= \mu \lambda_2 \|\text{dev}[\mathbf{b}^{e,*}]\|_F + \alpha \frac{\kappa}{2} (J^2 - 1).\end{aligned}\tag{3.6}$$

Equating to 0 and solving for  $\lambda_2$ , we conclude that:

$$\lambda_2 = -\left(\alpha \frac{\kappa}{2} (J^2 - 1)\right) / (\mu \|\text{dev}[\mathbf{b}^{e,*}]\|_F).\tag{3.7}$$

Similarly, if we expand  $\text{det}[\mathbf{b}^{e,p}]$  we find

$$\text{det}[\mathbf{b}^{e,p}] = \text{det}[\lambda_1 \mathbf{I} + \lambda_2 \text{dev}[\mathbf{b}^{e,*}]].\tag{3.8}$$

In 2D, recalling that  $\text{det}[\mathbf{I} + \mathbf{A}] = 1 + \text{det}[\mathbf{A}] + \text{Tr}[\mathbf{A}]$ , we have  $\text{det}[\mathbf{I} + \text{dev}[\mathbf{A}]] = 1 + \text{det}[\text{dev}[\mathbf{A}]]$ . Then, we arrive at

$$\begin{aligned}\text{det}[\mathbf{b}^{e,p}] &= \lambda_1^2 \text{det}[\mathbf{I} + \frac{\lambda_2}{\lambda_1} \text{dev}[\mathbf{b}^{e,*}]] \\ &= \lambda_1^2 \left(1 + \frac{\lambda_2^2}{\lambda_1^2} \text{det}[\text{dev}[\mathbf{b}^{e,*}]]\right) \\ &= \lambda_1^2 + \lambda_2^2 \text{det}[\text{dev}[\mathbf{b}^{e,*}]].\end{aligned}\tag{3.9}$$

Equating to  $\text{det}[\mathbf{b}^{e,*}]$  and solving for  $\lambda_1$  we find:

$$\lambda_1 = \sqrt{\text{det}[\mathbf{b}^{e,*}] - \lambda_2^2 \text{det}[\text{dev}[\mathbf{b}^{e,*}]]}.\tag{3.10}$$

In 3D, recalling that  $\det[\mathbf{I} + \mathbf{A}] = 1 + \det[\mathbf{A}] + \text{Tr}[\mathbf{A}] + \frac{1}{2}\text{Tr}[\mathbf{A}]^2 - \frac{1}{2}\text{Tr}[\mathbf{A}^2]$ , we have  $\det[\mathbf{I} + \text{dev}[\mathbf{A}]] = 1 + \det[\text{dev}[\mathbf{A}]] - \frac{1}{2}\|\text{dev}[\mathbf{A}]\|_F^2$ . Then, we arrive at

$$\lambda_1^3 + \lambda_2^3 \det[\text{dev}[\mathbf{b}^{e,*}]] - \frac{\lambda_1 \lambda_2^2}{2} \|\text{dev}[\mathbf{b}^{e,*}]\|_F^2 - \det[\mathbf{b}^{e,*}] = 0. \quad (3.11)$$

We employ Cardano's method to analytically solve (3.11) and obtain  $\lambda_1$ . With  $\lambda_1$  and  $\lambda_2$  in hand, we are able to easily project  $\mathbf{b}^{e,*}$  and enable the plastic flow.

### 3.2 Hypoelastic-Plastic Model

In general, hypoelastic models differ from hyperelastic models in that the stress is not obtained from a gradient of a strain energy density function with respect to deformation. The specific hypoelastic granular continuum model used in this study was developed by Dunatunga and Kamrin [10]. To start one again begins with momentum balance and mass balance

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (3.12)$$

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad (3.13)$$

with all terms similarly defined as in the hyperelastic model. A useful quantity, the spatial velocity gradient  $\mathbf{L}$ , is defined as

$$\mathbf{L} = \nabla \mathbf{v} \quad (3.14)$$

$\mathbf{L}$  can be decomposed into a symmetric part (known as the strain rate tensor) and skew part (known as the spin tensor),  $\mathbf{D}$  and  $\mathbf{W}$  respectively, such that

$$\mathbf{L} = \mathbf{D} + \mathbf{W} \quad (3.15a)$$

$$\mathbf{D} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^T) \quad (3.15b)$$

$$\mathbf{W} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T) \quad (3.15c)$$

In contrast to the hyperelastic model, the used hypoelastic model takes an additive split of the strain and strain rate-like terms into an elastic and plastic part. For example,

$$\mathbf{L} = \mathbf{L}^e + \mathbf{L}^p$$

The elastic and plastic spatial velocity gradients can then be decomposed into spin and strain rate tensors

$$\mathbf{L}^e = \mathbf{D}^e + \mathbf{W}^e$$

$$\mathbf{L}^p = \mathbf{D}^p + \mathbf{W}^p$$

Due to the fact that a hypoelastic-plastic model is used and there is no tracking of the deformation gradient, an objective rate must be used to update the stress. While many exist, the Jaumann rate is used here as suggested by Dunatunga and Kamrin, and is defined as

$$\overset{\triangle}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} - \mathbf{W} \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \mathbf{W} \quad (3.16)$$

With the basic kinematic variables needed now defined, the next step is defining the constitutive model. As stated in the beginning of this section, this is a hypoelastic-plastic model, and so the elastic constitutive model, plastic yield condition, and plastic flow rule are needed to close the system. The material is assumed to be isotropic and linearly elastic, and the stress is assumed to only be a function of elastic strains. In general the stress rate can then be expressed as a function of the elastic strains contracted with a fourth-order elastic tensor  $\mathbb{C}$ , or  $\overset{\triangle}{\boldsymbol{\sigma}} = \mathbb{C} : \mathbf{D}^e$ . With the assumptions of isotropocity and first-order linear elasticity, the stress rate can then be more specifically defined as

$$\overset{\triangle}{\boldsymbol{\sigma}} = 2G\mathbf{D}^e + \lambda \text{tr}(\mathbf{D}^e)\mathbf{I} \quad (3.17)$$

where  $G$  is the shear modulus (or second Lamé constant) and  $\lambda$  is the first Lamé constant. However, there is an additional condition on the pressure, which is that

$$p = \begin{cases} 0, & \text{if } \rho \leq \rho_c \\ \frac{K_c}{\rho}(\rho - \rho_c), & \text{if } \rho \geq \rho_c \end{cases} \quad (3.18)$$

In other words, if the density of the granular material falls below a certain level, the continuum represents a region of grains that is very loosely packed and has no contacts, and thus cannot support stress. In the physical sense, the grains in this region have entered a gaseous regime (though with no pressure from collisions with the boundary).

The yield condition is very similar in form to a Drucker-Prager yield condition, i.e.

$$\bar{\tau} \leq \mu p \quad (3.19)$$

where  $\bar{\tau}$  is the equivalent shear stress and  $p$  is the pressure, defined by

$$\bar{\tau} = \sqrt{\frac{1}{2}(\boldsymbol{\sigma}_0 : \boldsymbol{\sigma}_0)} \quad (3.20)$$

$$p = -\frac{1}{3}\boldsymbol{\sigma} \quad (3.21)$$

A key difference between the previously explained hyperelastic model and the current hypoelastic model is that the hypoelastic model used by Dunatunga, and subsequently used here, is the introduction of the  $\mu(I)$  rheology proposed by Jop et al [15]. The  $\mu(I)$  rheology proposes a characteristic nondimensional number  $I$ , defined as

$$I = \dot{\gamma}^p \frac{\sqrt{d^2 \rho_s}}{\sqrt{p}} \quad (3.22)$$

which gives a measure of the inertia in a sheared granular system relative to the pressure of the system. An empirical fit between  $\mu$  and  $I$  is given as

$$\begin{cases} \mu = \mu(I) = \mu_s + \frac{\mu_2 - \mu_s}{I_0/I+1}, & \text{if } I > 0 \\ \mu \leq \mu_s & \text{if } I = 0 \end{cases} \quad (3.23)$$

where  $\mu_s$ ,  $\mu_2$  and  $I_0$  are material parameters. As suggested by 3.23,  $\mu_s$  is a static friction coefficient, or the value of friction in the limit that  $I$  approaches 0. As  $I$  approaches infinity  $\mu$  approaches  $\mu_2$ . Though the existence of an asymptotic  $\mu_2$  is still debated in literature, it serves as a good approximation for the levels of  $I$  reached in the simulations run in this study. Thus the plastic yield condition utilized here is more exactly stated as

$$\bar{\tau} \leq \mu(I)p \quad (3.24)$$

At plastic yielding, a flow rule must be defined to evolve the plastic strain. A commonly taken assumption that is also taken here is one of spin-less plastic flow, so that  $\mathbf{W}^p = \mathbf{0}$  and  $\mathbf{L}^p = \mathbf{D}^p$ . Plastic flow codirectionality with the stress deviator and isochoric plastic flow are

also taken as assumptions, leading to an plastic flow rate of

$$\mathbf{L}^p = \hat{\mathbf{D}}^p(\boldsymbol{\sigma}) = \frac{1}{\sqrt{2}} \dot{\gamma}^p(\boldsymbol{\sigma}) \frac{\boldsymbol{\sigma}_0}{\|\boldsymbol{\sigma}_0\|} \quad (3.25)$$

where  $\dot{\gamma}^p$  is the equivalent plastic shear strain rate.

As a final note, there is a desired behavior of a "no tension" rule, in that granular media can not support tensile stress states. While this is partly captured by the pressure dependence on the material density relative to a critical density expressed in 3.18, another check must be done. In the constitutive update to evolve the stress, if it is determined that the pressure of the material is negative (i.e. the material wants to contract in on itself because of volumetric tensile stresses), then the stress is set to 0. Exact implementation details of the stress update can be found in Dunatunga et al, with the relevant density and pressure checks of that update being most relevant for hybridization purposes.

### 3.3 Material Point Method

In order to discretize and solve the equations defined in the previous sections, an appropriate method must be chosen. Classically the finite element method has been the method of choice for problems involving solid mechanics. As stated before however, a singular granular system, i.e. flow in an hourglass, has that granular system existing in multiple states at once: a solid bottom pile, a flowing regime down the top of the pile and at the top flowing into the hourglass neck, and a gaseous regime as it exits the neck. Using a method like the finite element to track the deformation of the granular continuum would be nearly impossible, due to the large amounts of non-affine strain that accumulate in the system causing mesh inversions. Remeshing, or a method like the Arbitrary-Lagrangian-Eulerian method, could at first glance help resolve this. However the amount of remeshing that needs to occur would incur both a computational penalty for the remeshing algorithm, but also an accuracy penalty due to the need to constantly interpolate quantities.

On the other hand, methods used to solve equations in an Eulerian frame for fluid mechanics, like the finite volume method, may then seem appealing. Finite volume however brings with it its own drawbacks in the context of granular media. Finite volume methods have trouble modeling purely solid regimes **FINDREFERENCE**. They also do not inherently track free surfaces like Lagrangian finite element would. This free surface tracking is crucial

in the problems of interest in granular media study, as the evolution of the free surface, and the interactions of the free surface with surrounding matter, are what ultimately matter in, for example studying the effects of a landslide on anything downhill of the flow zone. Breakaway of granular material from an initial agglomeration of material and the ability to divide that agglomeration into smaller bodies of granular material are also behaviors that are exhibited that cannot be easily captured by finite volume.

The ideal method then is Lagrangian, can track free surfaces, can also handle the large non-affine strains introduced in the liquid and gaseous regimes of granular flow. A class of methods, called particle methods, aim to solve this niche of problem by tracking the evolution of the system through particles, instead of with a mesh. Many types of course exist, including the popular smoothed-particle hydrodynamics (SPH) diffusive element method, and the reproducing kernel method (RKPM) [FINDREFERENCE](#). All vary in their exact discretization of continuum quantities, representation of connectivity between points, and other details. The continuum discretization method used in this study, known as the Material Point Method (MPM), is a framework that both provides familiarity with methods like the finite element method while adding on the abilities desired.

MPM was developed in the mid 1990s by Sulsky et al and has enjoyed much use and development since [34][FINDREFERENCE](#). MPM is what is known as a mesh-free method, which as the name implies, denotes that there is no permanent mesh used to track deformation. This lack of a permanent mesh of course avoids the mesh deformation issue entirely. As a brief history aside, MPM is a derivative of the fluid-implicit-particle method (FLIP), which is itself a derivative of the particle-in-cell (PIC) method, where PIC was developed in the context of building a method to solve for fluid flow in a Lagrangian frame. Properties of both methods explicitly arise in MPM, which will be discussed later.

### 3.3.1 MPM Algorithm Overview

In MPM, a continuum body is first discretized via Lagrangian markers, known as MPM points. Quantities of interest, like mass, momentum, stress, and any internal variables, are held on these points. It should be noted that there is no explicit notion of connectivity stored on the points between pairs or groups of points, and so no nearest-neighbor search must be conducted, like in SPH or many other particle methods. A temporary (with an emphasis on the "temporary", as the introduction of a mesh may seem contradictory to



Figure 3-1: Schematic of a single timestep in MPM.

MPM being classified a mesh-free method) background grid is then introduced as a "computational scratch-pad" . The aforementioned quantities of interest are then projected onto the background grid with a chosen set of basis functions. As a note, while there is no strict requirement on the discretization of the background grid, often a simple Cartesian grid is chosen for convenience. With these quantities now having a nodal representation on the grid, a finite element-like update is conducted. The updated nodal quantities are then projected back onto the MPM points, so that the points are now in an updated state. The background grid is then destroyed, so that no accumulation of strain occurs. With new point quantities, the points are then advected, completing a timestep of MPM.

### 3.3.2 MPM Formulation and Discretization

As shown schematically in the previous section, at the beginning of a timestep  $n$ , each MPM point  $p$  has stored on it its position  $\mathbf{x}_p^n$ , velocity  $\mathbf{v}_p^n$ , mass  $m_p^n$ , velocity gradient  $\mathbf{L}_p^n$ , Cauchy stress  $\boldsymbol{\sigma}_p^n$ , volume  $V_p^n$ , and for the hyperelastic case,  $\mathbf{B}_p^{en}$  and  $J_p^n$ . The grid projection of any point quantity  $\phi_p$  onto a node  $i$  is done via the operation

$$\phi_i = \sum_p S_{ip} \phi_p \quad (3.26)$$

where  $S_{ip}$  is the value of the basis function  $S_i$  at location  $\mathbf{x}_p$ , or  $S_{ip} = S_i \mathbf{x}_p$ . Likewise the grid projection of the gradient of any point quantity  $\phi_p$  onto a node  $i$  is done via the

operation

$$\nabla \phi_i = \sum_p \nabla S_{ip} \phi_p \quad (3.27)$$

While one is free to choose from any number of function spaces for the basis functions, two types are used in this study. The first are classic linear "hat" functions, which in 1D are defined as

$$S_i(x) = \max[0, (1 - \frac{|x_i - x|}{h})] \quad (3.28)$$

where  $h$  is the element length. The gradient is then defined as

$$\nabla S_i(x) = \begin{cases} \frac{\text{sgn}(x_i - x)}{h}, & \text{if } |x_i - x| \leq h \\ 0, & \text{otherwise} \end{cases} \quad (3.29)$$

The second class of basis functions used are known as GIMP (Generalized Interpolation Material Point) basis functions. GIMP basis functions take into account a finite size for the points (instead of a delta function classically used), and integrate the bases across this point domain. This extended support for the GIMP basis functions result in smoother grid crossings and higher order approximations. First order GIMP basis functions (resulting in 2nd order field approximations) were used, with details being found in [1].

The product of these basis functions in additional directions in 2D and 3D then form the basis in those dimensions. Note that from now on, all basis function values are taken for the point locations at time  $n$ , and so for brevity the superscript  $n$  is not included for the basis functions  $S_{ip}$  and gradients  $\nabla S_{ip}$ .

To begin, the point masses and momenta are projected onto the nodes via the operations previously described.

$$m_i^n = \sum_p S_{ip} m_p^n, (\mathbf{mv}_i)^n = \sum_p S_{ip} (\mathbf{mv}_p)^n \quad (3.30)$$

For the hypoelastic model, the point volumes and stresses are then updated. The volume update is described by

$$V_p^{n+1} = V_p^n \exp(\Delta t r \mathbf{L}^n) \quad (3.31)$$

The stress update is then calculated with the constitutive law described in Section 3.2, and the exact numerical implementation can be found in Dunantunga and Kamrin [10]. In the

---

**Algorithm 2 Overall\_MPM\_Algorithm**


---

```

1: projectPointMasses
2: projectPointMomentum
3: if HypoelasticModel then
4:   updateVolume
5:   computeHypoelasticCauchyStress
6: else if HyperelasticModel then
7:   updateVolumetricStrain
8:   computeHyperelasticCauchyStress
9: end if
10: projectForces
11: updateNodalMomentum
12: resolveNodalPlaneCollision
13: updateNodalVelocityAndAcceleration
14: updatePointVelocityGradient
15: if HyperelasticModel then
16:   elasticPrediction
17:   plasticCorrection
18: end if
19: updatePointVelocities
20: updatePointPositions
21: resolvePointPlaneCollision
22: clearGridData

```

---

hyperelastic case, the volumetric strain  $J$  and stress are also updated here, as described in 3.1. The external forces  $\mathbf{b}_i^n$  and internal forces  $\mathbf{f}_i^n$  (internal forces being derived from the divergence of the just-calculated Cauchy stress) are then projected to the grid.

$$\mathbf{b}_i^n = \sum_p S_{ip} m_p^n \mathbf{b}_p^n, \quad \mathbf{f}_i^n = \sum_p -V_p \boldsymbol{\sigma}_p^n \cdot \nabla S_{ip} \quad (3.32)$$

Now, the nodes contain both the current momentum and current forces. The change in nodal momentum is then given by

$$(\dot{\mathbf{m}\mathbf{v}})_i^n = \sum_i \mathbf{F}_i^n = \mathbf{b}_i^n + \mathbf{f}_i^n \quad (3.33)$$

The time integration of the nodal momentums can then be done a number of ways, but here a simple forward euler is used, which produces

$$(\mathbf{m}\mathbf{v})_i^{n+1} = (\mathbf{m}\mathbf{v})_i^n + \Delta t(\mathbf{b}_i^n + \mathbf{f}_i^n) \quad (3.34)$$

Nodal interactions with boundaries are then taken into account. In the current code, two types of boundaries are supported: "sticky" rigid planes and "sliding" rigid planes. The algorithm for this interaction check loops through all grid nodes  $i$ , and checks to see if any mass has been projected to the node. If so, then the code loops through all defined planes, calculating the distance of the grid point to the given plane, to check for a nodal collision with the plane. If there is a collision found, then the relative velocity of the point to the plane is calculated. For a "sliding" boundary, the normal relative momentum is set to zero (allowing movement in the tangential velocity, and hence the "sliding"), while in the "sticky" boundary case the entire nodal momentum  $(\mathbf{m}\mathbf{v})_i^{n+1}$  is set to  $\mathbf{0}$ .

Next, the new nodal velocities and accelerations are calculated as

$$\mathbf{v}_i^{n+1} = (\mathbf{m}\mathbf{v})_i^{n+1} / m_i^n \quad (3.35)$$

$$\mathbf{a}_i^{n+1} = \frac{(\mathbf{m}\mathbf{v})_i^{n+1} - (\mathbf{m}\mathbf{v})_i^n}{\Delta t m_i^n} \quad (3.36)$$

The new nodal velocities are used to calculate the new velocity gradients  $\mathbf{L}_p^{n+1}$  on the points with

$$\mathbf{L}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} \otimes \nabla S_{ip} \quad (3.37)$$

In the hyperelastic model, the elastic prediction and plastic correction steps are then conducted, as explained in 3.1.

The next step, the update of the point velocity, is one that deserves extra attention. Two quantities are temporarily introduced, the PIC velocity  $\mathbf{v}_{pic}$  and the point acceleration  $\mathbf{a}_p$ , defined as

$$\mathbf{v}_{pic} = \sum_i S_{ip} \mathbf{v}_i^{n+1} \quad (3.38)$$

$$\mathbf{a}_p = \sum_i S_{ip} \mathbf{a}_i^{n+1} \quad (3.39)$$

From this it can be seen that there are two possible avenues to update the point velocity to  $\mathbf{v}_p^{n+1}$ . One, called the PIC update (so-called because this is the point velocity update that the PIC method used), directly uses the  $\mathbf{v}_{pic}$  velocity as the new point velocity, so  $\mathbf{v}_p^{n+1} = \mathbf{v}_{pic}$ . This means that the point velocities are directly interpolated from the background grid velocities, and thus the velocity field is constrained by the basis functions. The other, called the FLIP update (again so-called because FLIP uses this as its velocity update), instead

uses the nodal accelerations to construct a point acceleration. The FLIP velocity,  $\mathbf{v}_{flip}$  is then obtained by

$$\mathbf{v}_{flip} = \mathbf{v}_p^n + \Delta t \mathbf{a}_p \quad (3.40)$$

With this construction, the velocities live in a higher order vector space, allowing for higher order kinematic modes. The macroscopic result of these two updates is that, often, flows with PIC updates have a high degree of dissipation and do not conserve angular momentum. Physically realistic voritical flow and oscillations either do not appear or are quickly damped. This dissipative quality however means that PIC schemes are often stable. On the other hand, FLIP updated flows more often preserve those vortical effects and oscillations and better conserve angular momentum. This though means that instabilities can form and will not be damped out.

A strategy that is used to obtain some middle ground between the two strategies is to simply take a linear combination of the two updates. This is expressed functionally as

$$\mathbf{v}_p^{n+1} = (1 - \alpha) \mathbf{v}_{pic} + \alpha \mathbf{v}_{flip} \quad (3.41)$$

where  $\alpha$  is a parameter used to tune how much one wants a PIC vs a FLIP update. A large  $\alpha$  value is often used, as better angular momentum conservation is usually desired over better numerical stability, though a small portion of PIC velocity still helps with stability. In this study,  $\alpha$  was set to between 0.95 and 1.0 for all simulations.

The points are then advected via

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_i S_{ip} \mathbf{v}_i^{n+1} \quad (3.42)$$

Another collision check is conducted on the advected position of the points. This collision check is very similar to the nodal check, wherein all of the points are looped over, a check for collision against any rigid planes is done, and the normal relative velocity is set to 0 for sliding planes and the entire velocity is set to 0 for sticky planes. Finally, the nodal quantities on the grid are all set to 0, effectively resetting the grid state.

### 3.3.3 Relevant Notes on MPM

As alluded to with the "FEM-like solve", MPM can be interpreted as a finite element method with a single point quadrature integration rule, where the quadrature points are the MPM points. Both start from a weak form formulation and are discretized into points/elements, with interpolation conducted under some basis. However in MPM, there is no strict notion of connectivity. Instead, communication of points is done through the projection of the point quantities to the nodes.

In the original MPM formulation described by Sulsky, the extent of the points are described via  $\delta(x)$  functions, and so only project to the grid nodes of the element that the point is currently in. If two points project to the same grid node, then connectivity is in effect established between the points. This then implies that if two points start out "connected", or are at most in directly adjacent cells, and then become separated by a completely empty element, they lose connectivity, completely separating the two points. While this can be useful in some applications to capture fracture, such as in the modeling of snow breakage in graphics usage [FINDREFERENCE](#), it should be stated that this is in fact numerical error. While schemes that integrate a point across a finite extent, such as GIMP, ameliorate this problem somewhat, this is not a complete fix. Thus while MPM avoids having grid cell inversions and other issues inherent in a completely lagrangian mesh, a sort of numerical fracture becomes the byproduct.

Numerical fracture is not the only issue that can arise however. In a somewhat less catastrophic error, the fact that the MPM points are effectively quadrature points, and they are allowed to advect from cell to cell, means that integration errors can occur. For example, a simulation can begin with the same number of MPM points per cell for all cells representing the body, exactly integrating those elements. Over time however, the volumes that the MPM points represent deform and change in size, and the points themselves move. This can mean that if at a certain timestep, a cell contains much fewer points than the simulation started with, that portion of the body can be under-integrated. On the other hand, a cell may have a large concentration of points, with the volumes that the points represent actually overlapping; over-integration of that portion of the domain then results.

Techniques exist to address these points. To better track the deformation of the volume that the MPM point represents in order to help with over and under integration, the pre-

viously mentioned GIMP as well as the Convected Particle Domain Interpolation (CPDI) technique can be used. CPDI, as its name implies, convects and keeps track of the representation of the MPM point volume, so that the area over which one integrates changes with MPM deformation **FINDREFERENCE**.

Resampling can also help to avoid over or under integration, keeping the MPM point concentration similar to what the simulation started with. A resampling algorithm was developed by Yue et al in the context of using MPM for foam modeling, where very large deformations occur that can result in under integration [38]. This technique, called Avoid a Void, is used in the current study both for the MPM simulations as well as the basis for a component of hybridization later discussed.



# Chapter 4

## Hybridization

The previous two chapters described two very different modeling paradigms, both with their strengths and weaknesses. As mentioned in the introduction, the goal is to be able to utilize the strengths of both to offset their weaknesses in order to create a versatile method that can handle phenomena that span many magnitudes of length scales and all phases of granular matter. Namely, the goal is to utilize the accuracy of the discrete element method where it is needed, while in all other regions, using the continuum method to solve a simple continuum model with much fewer degrees of freedom than a full discrete simulation. The method to achieve this is subsequently explained.

### 4.1 Hybridization Overview



Figure 4-1: Diagram of a hybrid simulation.

Figure 4-1 gives a visual representation of a hybrid simulation of a collapsing pile of grains. On the exterior are discrete grains, modeled by DEM. On the interior is a region of continuum, utilizing MPM. A regime where both DEM and continuum overlap each other, what is deemed the "Reconciliation Zone" or "Hybrid Zone" (both are used interchangeably), then connects the two modeling regimes.

From this schematic there are a couple of things of note. A single MPM simulation is used to solve for the pure continuum region, as well as the continuum portion of the hybrid zone. The continuum region has only a single piece of information that delineates the pure continuum from the hybrid continuum: a weight field  $w_c(x)$  that has a value of 1 in the pure continuum and a value between 0 and 1 in the hybrid regime. Likewise, a single DEM simulation is used to advance the state of the discrete grains in the pure discrete region as well as the discrete portion of the hybrid zone. Again, a weight field  $w_d(x)$  delineates the pure discrete and hybrid regimes, with a value of 1 in the pure discrete region and a value between 0 and 1 in the hybrid region.

At a very high level the hybrid scheme can be described thusly. At the beginning of a hybrid timestep, the domain is decomposed into pure continuum, pure discrete, and hybrid regions. The determination of which region is represented with which model is conducted by what is termed the "oracle", which will be later explained. If the oracle determines that a region described by the continuum model needs greater accuracy, then a process called "enrichment" converts the continuum representation to a discrete representation. On the other hand, if the oracle determines that a discrete region no longer needs to be resolved to that level, then a process called "homogenization" converts that discrete representation into a continuum representation. A step of MPM for what regions are now continuum and a step of DEM for regions that are now discrete are then taken, with no communication between the two regimes other than the weightage split previously described, resulting in intermediary quantities of state. These steps will be referred to as "unconstrained" steps, as they are unconstrained of any coupling between the two distinct systems. A coupling step then acts to kinematically constrain the two partitioned systems in the hybrid zones. These updated quantities are communicated back to the DEM and MPM solvers, which results in the end of timestep quantities for the entire simulation.

The hybrid scheme can be split into three main components. The first is the nature of the split of the continuum and discrete representations in the hybrid zone. The second is the

coupling of those two split regions such that they end the step kinematically constrained. The third is the domain decomposition, and the processes of homogenization and enrichment.

## 4.2 Reconciliation Zone Splitting



Figure 4-2: Blurred Density: (Left) The reference domain  $\Omega$  of an object with density  $\rho(\mathbf{x})$ . Mass density is colored in blue. (Right) A partition of unity of the density mediated by a weight function  $w(\mathbf{x}, t)$ .

In order to understand the hybrid zone split, a more general system is introduced. Suppose that we view a single system as if it were two separate systems. Let the mass density  $\rho(\mathbf{x})$  be defined over the reference coordinates  $\mathbf{x} \in \Omega$  of a body (Fig. ??). We partition the density with a space-time dependent weight function  $w(\mathbf{x}, t) \in [0, 1]$ , ensuring that we recover the original density:

$$\rho(\mathbf{x}) = w(\mathbf{x}, t)\rho(\mathbf{x}) + (1 - w(\mathbf{x}, t))\rho(\mathbf{x}) . \quad (4.1)$$

If we treat the new partitions as separate systems with generalized coordinates  $(\mathbf{q}_1, \mathbf{v}_1)$  and  $(\mathbf{q}_2, \mathbf{v}_2)$ , we can recover the kinematic description of the original system by requiring that  $\mathbf{q}_1(\mathbf{x}, t) = \mathbf{q}_2(\mathbf{x}, t)$ .

With identical initial configurations  $\mathbf{q}_1(x, t_0) = \mathbf{q}_2(x, t_0)$ , we can equivalently enforce equal velocities via the constraint  $\mathbf{c}(\mathbf{x}, t) = \mathbf{v}_1(\mathbf{x}, t) - \mathbf{v}_2(\mathbf{x}, t) = \mathbf{0}$ . Consistent with Fig. ??, our ultimate goal is to treat the discrete particles and the continuum as the two subsystems with a *reconciliation* zone wherever  $0 < w < 1$ .

We derive our governing equations for the hybrid coupling from Hamilton's Variational

Principle [20]; for dissipative systems, the analogous derivation follows from the Lagrange d'Alembert Principle, but the end result for our purposes is the same. This is a didactic derivation, and we will later apply these coupling forces to systems with friction, which still satisfies force/momentum balance.

Suppose kinetic  $T$  and potential  $U$  energies are given by

$$\begin{aligned} T &= \frac{1}{2} \int_{\Omega} \rho w \mathbf{v}_1^T \mathbf{v}_1 dV + \frac{1}{2} \int_{\Omega} \rho(1-w) \mathbf{v}_2^T \mathbf{v}_2 dV, \\ U &= \int_{\Omega} \rho w e[\mathbf{q}_1] dV + \int_{\Omega} \rho(1-w) e[\mathbf{q}_2] dV, \end{aligned} \quad (4.2)$$

for potential energy per mass  $e$ . Here and henceforth, we omit explicit parameters  $(\mathbf{x}, t)$  when the dependence is clear. In addition, we couple the two systems via the augmented constraint  $C = \int_{\Omega} \boldsymbol{\lambda}^T (\mathbf{v}_1 - \mathbf{v}_2) dV$ , where  $\boldsymbol{\lambda}(\mathbf{x}, t)$  is a Lagrange multiplier field.

The Lagrangian  $L = T - U + C$  is then incorporated into the action functional  $\int_t L dt$ , and the calculus of variations yields the Euler-Lagrange equations

$$\begin{aligned} \int_{\Omega} w \rho \mathbf{a}_1 dV &= - \int_{\Omega} w \underbrace{\rho \frac{\delta e}{\delta \mathbf{q}_1}}_{\substack{\text{Force} \\ \text{Volume 1}}} dV - \underbrace{\int_{\Omega} \boldsymbol{\lambda} dV}_{\text{coupling force}}, \\ \int_{\Omega} (1-w) \rho \mathbf{a}_2 dV &= - \int_{\Omega} (1-w) \underbrace{\rho \frac{\delta e}{\delta \mathbf{q}_2}}_{\substack{\text{Force} \\ \text{Volume 2}}} dV + \underbrace{\int_{\Omega} \boldsymbol{\lambda} dV}_{\text{coupling force}}, \end{aligned} \quad (4.3)$$

which are subject to the coupling constraint  $\mathbf{v}_1 = \mathbf{v}_2$ . The coupling force, which acts equally and oppositely on the two systems to enforce the constraint, arises naturally from the calculus of variations, averting the formulation of ad-hoc communication models between the two systems.

If we sum the two equations and substitute in the coupling constraint, we recover the original equations of motion for the entire simulation domain. The weight function naturally defines a partition of unity for the masses and the energies, with smaller weight values corresponding to a system having less influence in a given region. Outside the reconciliation zone, (4.3) is simply the (usual) equations of motion for two independent systems.

The Lagrange multiplier term operates as an external (constraint) force to ensure that the velocities of each system are equal. Under operator splitting, the above equations can

be interpreted as first having two sets of (usual) decoupled equations of motion

$$\begin{aligned}\rho \mathbf{a}_1 &= -\rho \frac{\partial e}{\partial \mathbf{q}_1}, \\ \rho \mathbf{a}_2 &= -\rho \frac{\partial e}{\partial \mathbf{q}_2},\end{aligned}\tag{4.4}$$

that are scaled by the weights, with a subsequent correction from the equal and opposite constraint force to guarantee equal velocities.

We now replace these two abstract systems with a discrete particle system and a continuum system (Fig. ??). By ansatz, we require the stress in the continuum domain to be compatible with the homogenized frictional forces in the discrete domain. This homogenization is realizable through the so-called Christoffersen formula [5], which relates the continuum stress to the discrete frictional contact forces via:

$$\sigma_{ij} = \frac{1}{V} \sum_{\alpha \in contacts}^N \frac{1}{2} (\mathbf{f}_i^\alpha \mathbf{d}_j^\alpha + \mathbf{f}_j^\alpha \mathbf{d}_i^\alpha),\tag{4.5}$$

where  $\sigma_{ij}$  is the (i,j) component of the stress tensor,  $V$  is the volume about which one is homogenizing the stress,  $N$  is the number of contacts in that volume,  $\mathbf{f}_i^\alpha$  is the  $i^{th}$  component of the contact force vector at the  $\alpha^{th}$  contact, and  $\mathbf{d}_i^\alpha$  is the  $i^{th}$  component of the vector connecting the centroids of the two grains in contact.

### 4.3 Coupling Constraints

Having accounted for the velocity (and possibly position) update from the equations of motion, we can interpret each system in terms of forces-per-volume and then, within a finite volume element, correct the velocities and positions to enforce coupling via (4.3) subject to the constraint  $C$ . This allows us to interpret the constraint  $\mathbf{v}_1 = \mathbf{v}_2$  in an average or homogenized sense [2]: in the reconciliation zone, the velocity of every discrete particle is forced to agree with the interpolated velocity of the continuum. Let  $\lambda_k$  represent the constraint force on the  $k^{th}$  discrete particle. Given the reconciliation zone,  $\Omega_R$ , the  $p^{th}$

material point moves as

$$\begin{aligned} \frac{d}{dt} \mathbf{q}_p &= \mathbf{v}_p, \\ \frac{d}{dt} (w_p M_p \mathbf{v}_p) &= \underbrace{\frac{d}{dt} (w_p M_p \mathbf{v}_p^*)}_{\text{unconstrained step}} \sum_{k \in \Omega_R} - \underbrace{\sum_{k \in \Omega_R} \Gamma_{pk} \lambda_k}_{\text{constrained step}}, \end{aligned} \quad (4.6)$$

while the  $k$ th discrete particle moves as

$$\begin{aligned} \frac{d}{dt} \mathbf{q}_k &= \mathbf{v}_k, \\ \frac{d}{dt} ((1 - w_k) M_k \mathbf{v}_k) &= \underbrace{\frac{d}{dt} ((1 - w_k) M_k \mathbf{v}_k^*)}_{\text{unconstrained step}} + \underbrace{\lambda_k}_{\text{constrained step}} \frac{d}{dt}, \end{aligned} \quad (4.7)$$

where  $\mathbf{v}_p^*$  ( $v_k^*$ ) are the predictions from continuum (discrete) simulation before coupling forces are added, and  $\Gamma_{pk}$  are material-point to discrete-particle interpolation coefficients.

### 4.3.1 Hybrid Coupling Discretization

We now present the discretized, hybrid coupling algorithm that follows from the previous discussion. Note that while general, implicit equations of motion would require a non-linear Newton solve, with explicit equations of motion, the coupled solve can be reduced to a single linear solve with a predictor-corrector structure. See Appendix ?? for a full derivation. We further observe that MPM forces are defined on a background Eulerian grid, which provides a natural scratch-pad on which to compute the constraint forces. Concretely, a full hybrid time step begins by stepping the continuum and discrete systems in isolation, omitting the advection steps, to obtain the predicted momenta. In our implementation, we compute a predictor discrete step by taking a full step of the discrete system and discarding the position update. For the continuum system, we split the full MPM step into two stages, where the predictor first stage steps the system to just the instant that material point properties are rasterized to the grid. In the DEM force computation, we weight each discrete spring force by  $1 - w$ , evaluated at the corresponding contact point, and each body mass by  $1 - w$ , evaluated at the center of mass. Likewise, in the MPM force computation, we weight the stress and mass of each material point by  $w$ , evaluated at the point location. We next assemble and solve a linear system to constrain the velocities of the discrete and continuum

simulations in the reconciliation zone:

$$\begin{bmatrix} W_c M_c & 0 & \Gamma_c \\ 0 & W_d M_d & -\Gamma_d \\ \Gamma_c^T & -\Gamma_d^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_c^{n+1} \\ \mathbf{v}_d^{n+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} W_c (M_c \mathbf{v}_c^n + h \mathbf{f}_c^n) \\ W_d (M_d \mathbf{v}_d^n + h \mathbf{f}_d^n) \\ 0 \end{bmatrix}, \quad (4.8)$$

where  $W_c$  and  $W_d$  are diagonal matrices that contain the mass weights for the continuum and discrete systems,  $M_c$  and  $M_d$  are the mass matrices,  $\mathbf{v}_c$  and  $\mathbf{v}_d$  are the velocities, and  $\mathbf{f}_c$  and  $\mathbf{f}_d$  are explicit forces from the continuum and discrete systems.  $M_c \mathbf{v}_c^n + h \mathbf{f}_c^n$  and  $M_d \mathbf{v}_d^n + h \mathbf{f}_d^n$  are the predicted momenta of the continuum and discrete systems, respectively, and solving the linear system gives the corrected continuum and discrete velocities  $\mathbf{v}_c^{n+1}$  and  $\mathbf{v}_d^{n+1}$ .  $\Gamma_d$  and  $\Gamma_c$  are defined such that  $\Gamma_d^T \mathbf{v}_d - \Gamma_c^T \mathbf{v}_c$  produces the residual relative velocity of discrete bodies within the background velocity field defined by the material point grid. Under this definition,  $\Gamma_d$  reduces to the identity matrix, while each column of  $\Gamma_c$  contains the weights that recover each discrete body's center of mass from the MPM grid's basis functions. We compute  $\Gamma_c$  using the positions of the discrete bodies and material points at the beginning of the step. Note that we can restrict the size of this system to only the degrees of freedom in the reconciliation zone. After solving the linear system, we update the discrete velocities and advect the discrete bodies along the corrected velocities, concluding the discrete step, and we compute the latter stage of the material point step using the new, constrained velocities, concluding the continuum step. This concludes a full hybrid time step, the details of which are summarized in Alg. ??.

### 4.3.2 Nodal Coupling

While this method to compute coupling forces indeed works, a further speedup is possible by defining a second background grid that is co-located with the background MPM grid. The velocities of the discrete bodies can be projected to this second grid as if they were material points. The constraint matrices  $\Gamma_d$  and  $\Gamma_c$  now reduce to the identity, and the system in

(4.8) can be solved in closed form (4.12), giving the system of equations:

$$W_c M_c \mathbf{v}_c^{n+1} + \lambda = W_c M_c \mathbf{v}_c^*, \quad (4.9)$$

$$W_d M_d \mathbf{v}_d^{n+1} - \lambda = W_d M_d \mathbf{v}_d^*, \quad (4.10)$$

$$\mathbf{v}_c^{n+1} = \mathbf{v}_d^{n+1}. \quad (4.11)$$

Substituting  $\mathbf{v}_c^{n+1}$  for  $\mathbf{v}_d^{n+1}$  in Eq. (4.9), adding Eq. (4.9) and Eq. (4.10), and solving for  $\mathbf{v}_d^{n+1}$ , we find that:

$$\mathbf{v}_c^{n+1} = \mathbf{v}_d^{n+1} = (W_c M_c + W_d M_d)^{-1} (W_c M_c \mathbf{v}_c^* + W_d M_d \mathbf{v}_d^*). \quad (4.12)$$

For diagonal mass matrices, each degree of freedom can be solved for independently, and the formula reduces to an inelastic impact between two particles in one dimension.

Crucially, the solution at each grid node is independent of the other grid nodes, and takes the simple form of an inelastic collision between two particles with masses and velocities equal to those of the grid node (Alg. ??). This method to compute hybridization forces is simple, robust, and trivially parallelized. After solving this system, the discrete grid-based velocities are mapped back to the discrete bodies in the same manner as MPM points. We always employ this grid vs. grid hybridization technique, and never directly solve the linear system (4.8).

## 4.4 Domain Decomposition

### 4.4.1 Oracle

Critical to our hybridization method is an oracle that is able to flag regions of the simulation domain as safe for a continuum treatment. Regions are unfit for a continuum treatment when one of any number of conditions are satisfied. First, in regions of low pressure, grains are more likely to separate from the material bulk and undergo ballistic motion. Second, high strain rate gradients suggest that the granular flow varies too rapidly to be safely represented as a homogenized continuum [9, 16, 19]. Finally, in thin flows, grain-level dynamics can dominate, leading to finite size effects (e.g., clogging) not captured by local continuum models [3, 11, 24, 28]. We have found that the packing fraction serves as an effective signal



Figure 4-3: Initialization of a hybrid simulation: (A) We begin with a collection of DEM grains. (B) We next locate a level set corresponding to a given low density, here denoted as a black line. (C) Across the domain, we compute the distance to the density threshold, indicated by lines in lighter shades of red as the distance increases. (D) We select a user-tunable distance to the density level-set that serves as the center of the hybrid "reconciliation" zone. We denote this critical distance as a solid black line. (E) We extend the hybrid zone along the distance field by a given half-width in each direction, indicated by dotted lines. This hybrid reconciliation zone between the dotted lines defines a zone where the DEM system will be coupled to the continuum system. We homogenize the velocity and stress for use in step (G). (F) We delete all discrete grains that fall within the inner boundary of the reconciliation zone. (G) We run the "avoid-a-void" algorithm of Yue et al. [38] from the outer boundary in to populate the region with material points. The material point states are determined using the homogenized velocity and stress computed in step (E).

for these sources of fundamentally discrete behavior. The emphasis on packing fraction is further motivated by the fact that graphical fidelity should be retained on the exterior of granular systems of interest, as it is this low packing fraction exterior that is seen by an observer.

Our oracle begins by computing the packing fraction of the discrete particle system on a uniform background grid. Note that only discrete grains are considered when computing the packing fraction, as continuum and hybrid regions are, by ansatz, considered sufficiently dense. From this implicit representation, we extract an isocontour corresponding to a critical, threshold packing fraction  $\Phi_\rho$  (Fig. ?? (B)). We next compute the distance  $\Phi_d(\mathbf{x})$  to this threshold isocontour on a second, uniform grid (C), and define an isocontour corresponding to a user specified distance  $\phi_0$  as the centerline of the reconciliation zone (D). We label a zone as hybrid if the distance from the centerline is within a given half-width  $r_h$ , i.e.  $\phi_0 - r_h \leq \Phi_d(\mathbf{x}) \leq \phi_0 + r_h$ . Similary, we label a zone as continuum if  $\Phi_d(\mathbf{x}) > \phi_0 - r_h$  and a zone as discrete if  $\Phi_d(\mathbf{x}) < \phi_0 + r_h$  (Alg. 3).

#### 4.4.2 Homogenization and Enrichment

After updating the boundary between simulation domains, we are faced with four possible transition scenarios: a previously hybrid zone is now purely discrete, a previously hybrid zone is now purely continuum, a previously discrete zone is now hybrid, or a previously continuum zone is now hybrid. Note that after initialization, we do not permit direct transitions from continuum to discrete regions or vice versa. The transition away from a hybrid representation is quite simple. For a hybrid region transitioning to a purely discrete region, we simply delete all material points in the region. Similarly, for a hybrid region transitioning to a purely continuum region, we delete all discrete grains in the region. The transition to a hybrid representation is more involved, however. A region that previously contained only material points will require the insertion of discrete grains. Likewise, a previously discrete region will require the insertion of new material points. See Alg. 4.

The problem of adding samples to a dynamic simulation was addressed in the context of the material point method with the recently proposed avoid-a-void algorithm [38]. The avoid-a-void method applies Poisson disc sampling to maintain approximately constant material point distributions and to prevent the formation of non-physical voids within a simulated material. This technique is perfectly suited to our needs, where we need to insert

material points or discrete grains in regions of high material density recently labeled as hybrid.

For discrete grains, the new position is determined by the Poisson disc sampling procedure, while we draw the radius of new grains from the same normal distribution used to generate the initial grain radii. The initial velocity of new discrete grains is computed by averaging the velocity of surrounding discrete grains and material points within a radius of 6 (mean) grain diameters, with an exponential falloff (Alg. 6). This window width is slightly above the minimal size that recovers continuum-like quantities in a “granular volume element” [26]. These nearest neighbor queries are accelerated with a uniform background grid.

We first determine the positions of new material points with Poisson disc sampling. Next, we compute the velocity, (normalized) strain, and deformation gradient magnitude of new points with *homogenization*. We utilize a regular grid to rasterize the velocity of discrete grains; after rasterizing each discrete grain’s mass and momentum to the grid (using the same rasterization procedure as MPM), we divide the rasterized momentum by the rasterized mass to determine each grid node’s velocity. Likewise, we estimate the *homogenized* stress at grid nodes using a modification of the Christoffersen formula [5] that yields smooth stress fields via the MPM shape functions, as described in Alg. 7. After computing the homogenized velocities and stresses at nodes, we use the MPM shape functions to evaluate the velocity and stress at the newly generated material point positions. Finally, we convert the interpolated stress to strain: from the definition of the Kirchhoff stress, we can relate the Cauchy stress and the strain via  $\boldsymbol{\sigma} = \frac{\kappa}{2} \frac{(J^2 - 1)}{J} \mathbf{I} + \frac{\mu}{J} \text{dev}[\bar{\mathbf{b}}^e]$ . By taking the trace of both sides, we obtain  $\text{Tr}[\boldsymbol{\sigma}] = \frac{3\kappa}{2} \frac{(J^2 - 1)}{J}$  (in 3D) and solve for  $J$ . Next, by taking the deviator of both sides, we obtain  $\text{dev}[\bar{\mathbf{b}}^e] = \frac{J}{\mu} \text{dev}[\boldsymbol{\sigma}]$ . Finally,  $\bar{\mathbf{b}}^e$  is given in the form  $\bar{\mathbf{b}}^e = \text{dev}[\bar{\mathbf{b}}^e] + t\mathbf{I}$ , and we solve for  $t$  with  $\det[\bar{\mathbf{b}}^e] = 1$ .

We emphasize the importance of syncing the stress and strain during homogenization to preserve volume. Previously, we initialized new continuum elements to a stress-free state. This stress-free initialization caused the material to gain volume if the discrete grains were between a gaseous and liquid state while separating, however.

### 4.4.3 Layered Hybridization

We often found it beneficial to only use the discrete treatment for the topmost visible portion of a granular assembly. If the boundary treatment (at the side walls and floor) with the continuum elements is physically valid, and if the continuum elements are invisible from the outside, this *layered hybridization* approach yields additional speed improvements. We apply this approach to the bunny toss, excavator, bunny drill, and tire examples in Section ??.

---

#### **Algorithm 3 Identify\_Hybrid\_Zones**

---

- 1:  $\Phi_\rho \leftarrow \text{Discrete\_Packing\_Fraction\_Isocontours}(\mathbf{q}_d)$
  - 2:  $\Phi_d \leftarrow \text{Distance\_to\_Density\_Isocontours}(\Phi_\rho)$
  - 3:  $\text{continuum\_zone}(\mathbf{x}) \leftarrow \Phi_d(\mathbf{x}) > \phi_0 - r_h$
  - 4:  $\text{discrete\_zone}(\mathbf{x}) \leftarrow \Phi_d(\mathbf{x}) < \phi_0 + r_h$
  - 5:  $\text{hybrid\_zone}(\mathbf{x}) \leftarrow \phi_0 - r_h \leq \Phi_d(\mathbf{x}) \leq \phi_0 + r_h$
- 

---

#### **Algorithm 4 Update\_Hybrid\_Zones**

---

- 1:  $\text{Homogenize\_Velocity\_and\_Stress}(\text{hybrid\_zone})$
  - 2:  $\text{Discrete\_Avoid\_a\_Void}(\text{hybrid\_zone})$
  - 3:  $\text{Continuum\_Avoid\_a\_Void}(\text{hybrid\_zone})$
  - 4:  $\text{Delete\_Discrete\_Grains}(\text{continuum\_zone})$
  - 5:  $\text{Delete\_Continuum\_Particles}(\text{discrete\_zone})$
- 

---

#### **Algorithm 5 Update\_Hybrid\_State**

---

- 1:  $\text{Identify\_Hybrid\_Zones}$
  - 2:  $\text{Update\_Hybrid\_Zones}$
- 

## 4.5 Summary of Hybrid Algorithm

---

**Algorithm 6 Create\_Discrete\_Grain**

---

```
1:  $\mathbf{x} \leftarrow \text{Position\_from\_Avoid\_a\_Void}$ 
2:  $r \leftarrow N(r_{mean}, r_{sigma})$                                  $\triangleright$  Normally distributed radii
3:  $m \leftarrow \frac{4}{3}\pi r^3$ 
4:  $\mathbf{v} \leftarrow 0$ 
5:  $W \leftarrow 0$ 
6: for  $i = 0 \dots N_d$  do                                      $\triangleright$  Iterate over discrete grains
7:   if  $|\mathbf{x}_i - \mathbf{x}| < 12 r_{mean}$  then
8:      $w \leftarrow e^{|\mathbf{x}_i - \mathbf{x}|^2 / 2r_{mean}^2}$ 
9:      $\mathbf{v} \leftarrow \mathbf{v} + w \mathbf{v}_i$ 
10:     $W \leftarrow W + w$ 
11:   end if
12: end for
13: for  $i = 0 \dots N_c$  do                                      $\triangleright$  Iterate over material points
14:   if  $|\mathbf{x}_i - \mathbf{x}| < 12 r_{mean}$  then
15:      $w \leftarrow e^{|\mathbf{x}_i - \mathbf{x}|^2 / 2r_{mean}^2}$ 
16:      $\mathbf{v} \leftarrow \mathbf{v} + w \mathbf{v}_i$ 
17:      $W \leftarrow W + w$ 
18:   end if
19: end for
20:  $\mathbf{v} \leftarrow \mathbf{v}/W$ 
```

---

---

**Algorithm 7 Homogenize\_Velocity\_and\_Stress**

---

```
1: for each body  $\in$  Discrete_Grains do
2:   for node  $\in$  Stencil(body) do
3:      $w \leftarrow \text{Weight}(\text{body}, \text{node})$ 
4:     node.m +=  $w \cdot \text{body.m}$ 
5:     node.momentum +=  $w \cdot \text{body.m} \cdot \text{body.v}$ 
6:   end for
7: end for
8: for node  $\in$  Grid_Nodes do
9:   if node.m > 0 then
10:    homogenized_velocity  $\leftarrow$  node.momentum/node.m
11:   end if
12: end for
13: for each c  $\in$  collisions do
14:   for node  $\in$  Stencil(c) do
15:      $w \leftarrow \text{Weight}(c, \text{node})$ 
16:      $\mathbf{f}_c \leftarrow c.\text{collision\_force}$                                  $\triangleright$  Normal and friction forces
17:      $\mathbf{r}_c \leftarrow c.\text{arm\_vector}$ 
18:      $\boldsymbol{\sigma}_c \leftarrow \frac{1}{2} (\mathbf{f}_c \mathbf{r}_c^T + \mathbf{r}_c \mathbf{f}_c^T)$ 
19:     homogenized_stress +=  $w \cdot \boldsymbol{\sigma} / \text{cell\_volume}$ 
20:   end for
21: end for
```

---

---

**Algorithm 8 Discrete\_Avoid\_a\_Void**

---

```
1: for cell ∈ Hybrid_Zone do
2:   for i = 0 ... Max_Iters do
3:     Create_Discrete_Grain()
4:   end for
5: end for
```

---

**Algorithm 9 Create\_Continuum\_Particle]**

---

```
1: x ← Position_from_Avoid_a_Void
```

---

**Algorithm 10 Reassign\_Hybrid\_Continuum\_Properties**

---

```
1: for point ∈ Cell do
2:   point.m ← mpm_mass_per_cell/(#points ∈ Cell)
3:   point.vol ← cell_volume/(#points ∈ Cell)
4:   point.v ← homogenized_velocity(point.x)
5:   σ ← homogenized_stress(point.x)
6:   point.strain ← convert_stress_to_strain(σ)
7: end for
```

---

**Algorithm 11 Determine\_New\_Inner\_Continuum\_Properties**

---

```
1: for newly_sampled_point ∈ Cell do
2:   point.m ← gather_mass_of_extant_nghbr_pnts
3:   point.vol ← gather_vol_of_extant_nghbr_pnts
4:   point.v ← interp_vel_of_extant_nghbr_pnts
5:   point.strain ← interp_strn_of_extant_nghbr_pnts
6: end for
```

---

**Algorithm 12 Continuum\_Avoid\_a\_Void**

---

```
1: for cell ∈ Hybrid_Zone do
2:   for i = 0 ... Max_Iters do
3:     Create_Continuum_Particle()
4:   end for
5:   Reassign_Hybrid_Continuum_Properties
6: end for
7: for cell ∈ Continuum_Zone do
8:   for i = 0 ... Max_Iters do
9:     Create_Continuum_Particle()
10:    end for
11:    Determine_New_Inner_Continuum_Properties
12: end for
```

---

**Algorithm 13 Delete\_Discrete\_Grains**

---

```
1: for body ∈ Discrete_Grains do
2:   if body ∈ Continuum_Zone then
3:     delete(body)
4:   end if
5: end for
```

---

---

**Algorithm 14 Delete\_Condinuum\_Particles**

---

```
1: for point ∈ Material_Points do
2:   if point ∈ Discrete_Zone then
3:     delete(point)
4:   end if
5: end for
```

---



## Chapter 5

# Enrichment and Homogenization Improvements

The Results section showed that, for the problems run, the implementation of the hybrid method seems adequate to capture bulk behavior, like the end shape of a pile of collapsed grains. This however is interesting, as the previously described scheme includes a somewhat straightforward solution to ill-posed problems. The most ill-posed portion of the hybrid technique is the enrichment step, as a given continuum state can be represented by an infinite set of discrete element grain arrangements.

The solution applied to the enrichment problem, Avoid a Void, simply adds grains to all hybrid elements. There is no other information used to inform where Avoid a Void should place grains (as the Poisson Disk Sampling is random by nature), how many grains to add, or what kind of connectivity with other grains should be had. In the examples shown in the Results section, this is seemingly enough, as there are actually a relatively low number of conversions from discrete to continuum and continuum to discrete representations. For example, in the column collapse case, the core of the columns remain relatively steady, and conversion only happen around the exterior of the columns. Once the pile has collapsed, the pile remains steady and no other conversions occur. In the wheels driving over gravel example, there are also a lack of many conversions, as those conversions only occur for the elements directly under the wheel; once the wheel has driven past a set of elements, the granular bulk remains steady and the representations of the grains do not change.

The funnel flow example however indicates where the simple uninformed Avoid a Void

scheme for enrichment can begin to break down. As can be seen in Figure FINREFERENCE, the top half of the funnel starts with both the pure discrete and hybrid funnels at the same volume. However after all of grains have flowed from the top half of the funnel to the bottom half, it can be clearly seen that the hybrid pile has lost volume compared to the pure discrete pile. The transient behavior is also different, with the hybrid scheme flowing at a different rate than the discrete grains. A key difference between the funnel flow geometry and the other previously discussed examples is that in the funnel flow, there is a continuous change from continuum to discrete grains at the top, and discrete to continuum grains at the bottom. The result is that the deficiencies of the enrichment and homogenization schemes are exposed.

## 5.1 Volume Change and Mass Conservation



Figure 5-1: Mechanisms of deviations away from exact mass conservation.

Every hybrid update step, the Avoid a Void scheme attempts to pack in grains in the hybrid elements. Because it does so in an uninformed manner however, this can lead to mass and volume loss or gain. For example, If the discrete grains in a hybrid cell have a large enough packing fraction  $\Phi$  where the element remains hybrid, but below random close packing  $\Phi_{RCP}$ , the previously described enrichment scheme has a chance to pack in additional points. If that relatively low packing fraction is an accurate representation of the granular system at that element, then there may be mass introduced. If that hybrid element then is converted into a pure discrete element, permanent volume gain is then introduced into the system.

On the other hand, mass can also be permanently lost in the system. If for a given hybrid element, the discrete grains have a packing fraction  $\Phi < \Phi_{actual}$  based off of mass and volume that needs to be converted from a continuum representation, then the idea is that the Avoid a Void algorithm will eventually fill that missing space. However, Poisson Disk Sampling (PDS) has a limit and will not be able to reach  $\Phi_{RCP}$ . If  $\Phi_{actual} \approx \Phi_{RCP}$  or worse, if  $\Phi_{actual} \geq \Phi_{RCP}$  because the underlying discrete structure is crystalline, PDS will not be able to pack that space to the desired  $\Phi$ . In some geometries and flows, this may not be a problem, as, if the conversions happen at a slow rate compared to the hybrid update frequency, then the grains in an underpacked hybrid cell may rearrange to allow for PDK to pack in the enough grains. The column collapse and wheel examples for example, fall under this category. The funnel flow however has a continual flux of continuum into the hybrid zone which must be enriched. Mass loss occurs because the continuum to enrichment mass flux is larger than the source of discrete hybrid mass that the enrichment scheme can provide. The resulting lack of discrete particles then leads to volume loss.



Figure 5-2: Evolution of a hybrid flow down a chute showing mass loss.

As an extreme example, take the example of a flow down a chute with periodic boundary conditions. In this system, gravity is angled relative to the bottom boundary, driving continuous flow through the system. On the left side, discrete grains continuously enter a hybrid zone, where continuum points are in turn generated. As they move from left to right through the system, the discrete points are deleted as they enter the pure continuum zone and continuum points gain full weightage. The points then enter the hybrid zone on the right, where discrete grains are generated. Finally the continuum points exit into the

discrete zone where they are deleted, and the discrete grains gain full weightage. With the current scheme, mass and volume are continuously lost. Eventually the pile of flowing grains reduces to a point where there is insufficient height to support a continuum or hybrid scheme according to the oracle, resulting in a pure discrete system.

There are thus two main problems that must be addressed: mass conservation and volume change tracking. These two problems however must be resolved in the discrete representation and continuum representation in different ways. Mass conservation for the continuum representation (converting mass from discrete grains to continuum) is fairly simple, as the continuum nature allows for mass addition or subtraction in whatever increments desired. Mass in the continuum can thus be tracked exactly over time. Volume change though must be conducted in a manner consistent with that mass change, and this must be addressed.

For the discrete representation, mass conservation and volume change are completely coupled, as the density of the particles remains constant throughout the simulation. Mass conservation is more difficult to achieve in the discrete case, as mass conservation becomes a packing problem as previously described. Mass and volume in the discrete case also come in discrete units of a single grain at a time. This means that mass conservation cannot be exactly achieved in the discrete representation. The homogenization of a grain of radius  $r_a$  can be offset by the enrichment of a grain of radius  $r_b$ , but mass will only be exactly conserved if  $r_a = r_b$ . For a monodisperse system this will of course always be true, but all the simulations run in this study have some polydispersity, making that condition almost never true. Thus mass conservation can only be achieved in a time-averaged sense for the discrete particle representation.

### 5.1.1 Mass Ledger

In order to conserve mass and inform the enrichment scheme on how much mass must be converted from one representation to another, all of the relevant mass fluxes must be kept track of. Take as an example the simple hybrid system shown in Figure 5-3. Again, a simple 50/50 weight split in the hybrid system is shown both for simplicity and to reflect the weight function used in the current study. At the  $n^{th}$  timestep hybrid update, the marked DEM grain has radius  $r_d$ , density  $\rho_d$ , and weight  $w_d^n = 1$ , resulting in a mass  $m_d^n = m_d = w_d^n \rho_d \pi r_d^2$ . At the  $n^{n+1}$  hybrid update, the discrete grain  $d$  has moved into the neighboring hybrid element (Element 1). Now,  $w_d^{n+1} = 0.5$ , resulting in a mass  $m_d^{n+1} = 1/2 m_d^n$ . The mass



Figure 5-3: Mass fluxes in a hybrid system and the construction of a ledger.

difference  $\Delta m_d = 1/2m^d$  is mass that must be represented by continuum in order to maintain a partition of unity for the total mass that entered,  $m_d$ . There is thus a *deficit* of continuum mass in Element 1 that must be added, through addition of mass to the currently existing MPM points, addition of a new MPM point with that deficit, or some combination of both.

Moving attention to the marked MPM point, the MPM point  $p$  at the  $n^{th}$  hybrid update is in the hybrid zone and has weight  $w_p^n = 0.5$  and mass  $m_p^n = w_p^n m^p = 0.5m^p$ . At the next hybrid update its weight changes to 1, resulting in a mass of  $m_p^{n+1} = m^p$  and a resulting continuum mass *excess* in Element 2 of  $0.5m^p$ .

From these cases it can be seen there are multiple ways that discrete and continuum mass can accrue deficits or excesses depending on weight changes as they move between different zone types. These excesses or deficits are logged in a mass *ledger*, which informs the enrichment and homogenization schemes on how much mass of which type of representation is required. It should be noted that while there must always be at least one hybrid zone between a pure discrete and pure continuum zone, a grain in a pure discrete zone at hybrid update step  $n$ , it is still possible for it to advect to a pure continuum step  $n$  due to geometry and hybrid update frequency. This can be seen in Figure 5-4.

This case is not treated any differently to the previously described cases. While the discrete point is deleted from the system for being in a pure continuum zone, this can be interpreted as simply a weight change from  $w_d = 1$  to  $w_d = 0$ . This results in a continuum mass deficit of  $m_d$  in the element that the grain was deleted from. The same could of course occur for an MPM point entering a pure discrete region, resulting in a discrete mass deficit.



Figure 5-4: Discrete particle entering a pure continuum regime.

This analysis so far has assumed a fixed domain decomposition, where all element types remain fixed over time. This however does not need to be the case, as the framework of mass deficits or excesses as a result of weight changes can generalize to a system where DEM grains and MPM points advect through elements with different representations, as well as the representations themselves changing from timestep to timestep. For the mass ledger, all that needs to be known is the type of zone that the grain or point was in at the previous hybrid update, and what type of zone it is currently in. If the zone type was the same for the starting element at the previous update as the current element at the current update, then no change to the ledger needs to be made. If the zone types are different, then the mass difference due to the weight change is then marked in the ledger.

### 5.1.2 Mass Ledger Implementation

In order to keep track of mass deficits or excesses, two ledgers are introduced, which, in implementation, are simple vectors: the **dem\_deficit** and **continuum\_deficit**. Each vector stores the eponymous mass deficit on each element in the simulation. To establish convention, mass deficits are positive and excesses are negative. Assuming an MPM point starting with a mass  $m_p$  and a DEM grain with mass  $m_d$ , the ledger changes can be summarized as:

The mass ledgers for a hybrid update are calculated near the beginning of the hybrid update: it is done after the level set calculation and after the oracle has determined the zone types based on those calculations, but before the enrichment and homogenization steps. The ledgers are updated every hybrid update, so that there is a running list of mass deficits. This

Representation	Start Zone	End Zone	Ledger	Deficit
DEM Grain	Discrete	Discrete	N/A	N/A
		Continuum	<b>mpm_deficit</b>	$+m_d$
		Hybrid	<b>mpm_deficit</b>	$+0.5m_d$
	Hybrid	Discrete	<b>mpm_deficit</b>	$-0.5m_d$
		Continuum	<b>mpm_deficit</b>	$+0.5m_d$
		Hybrid	N/A	N/A
MPM Point	Continuum	Discrete	<b>dem_deficit</b>	$+m_p$
		Continuum	N/A	N/A
		Hybrid	<b>dem_deficit</b>	$+0.5m_p$
	Hybrid	Discrete	<b>dem_deficit</b>	$+0.5m_p$
		Continuum	<b>dem_deficit</b>	$-0.5m_p$
		Hybrid	N/A	N/A

Table 5.1: Mass ledger contributions.

thus enables a way to prevent mass gain and tackle mass loss. If an element has a negative deficit due to, for example, introducing a grain with a slightly larger radius than was needed, then the ledger can inform the enrichment scheme to not introduce any more DEM grains in that cell. On the other hand, if there is a positive deficit, then if at a given hybrid update the packing scheme is unable to introduce enough grains, then there is a chance in future updates that the grains have reached a configuration to allow for further grain introduction. Implementation wise, only elements that have a positive deficit are marked for enrichment.

## 5.2 Packing Schemes

Practically speaking, the mass ledger mostly addresses mass gain problems, as it stops enrichment in those elements. In elements that have a deficit, the enrichment scheme will continuously try to input grains and points, but this is not any different than the previously used scheme which tried to pack grains for all hybrid elements at every hybrid update. The problem remains that for certain geometries and flow profiles, the Avoid a Void's PDS scheme, which is random in nature, was unable to introduce DEM grains at a high enough packing fraction at a rate equal to or greater than the continuum mass flux into a hybrid cell. A new packing scheme is needed. The following section summarizes the different packing strategies that were attempted and discusses their characteristics.

In order to evaluate the effectiveness of new packing schemes, the inclined plane geometry shown in Figure 5-2 is used as a case study. The hyperelastic model was used to obtain

the result shown in Figure 5-2 for demonstration purposes; a switch is now made to the hypoelastic model, as it includes the  $\mu(I)$  relation that captures the correct velocity profile in an inclined chute flow, minimizing the role of the constitutive law in the mass loss issue.

### 5.2.1 Random Packing

Before any new packing methodologies are introduced however, it should be noted that the random scheme has a parameter *numTrials* which controls the number of attempts the code makes to introduce a new point. For all of the simulations shown in the Results section, this was set to 20, mostly as a compromise between speed and accuracy for the geometries that we. The first logical attempt at a solution to the packing problem is then to simply increase the number of attempts, trading the time lost in increasing the number of attempts for better packing.



Figure 5-5: Mass over time for random packing.

As Figure 5-5 shows, this is not enough. While increasing *numTrials* does help slow down the mass loss rate, the mass loss rate does reach a limit as a function of *numTrials*. At first pass this may seem counterintuitive, as in the limit of infinite trials a dense configuration should be sampled. However, this is not true do the fact that the random packing scheme is a greedy scheme. If the random scheme picks a point center with a random radius that has no collisions with any other point in the enriched element, then that point will be considered valid and added to the existing points. This limits the possible areas that new grains can now enter, as can be seen in Figure 5-6.



Figure 5-6: Non-optimal grain placement leading to inability to reach target packing fraction.

### 5.2.2 Grid Packing

Enrichment can occur often, depending on the geometry of the problem. Because of this, speed is still a requirement of any practical packing scheme. Thus, while less informed greedy schemes may not provide adequate packing, they do potentially show a much larger speed advantage over other schemes. Packing in general is a problem that shows up in many contexts, and solutions do exist that show the ability to attain very high packing fractions that still avoid crystalline packings [FINDREFERENCE](#). However these solutions are currently avoided, due to their computational expense. For example, some class of solutions solve a constrained minimization problem, maximizing packing fraction while constrained to limited or no overlaps. Introducing an optimization solve is potentially too slow for the current application. On the other hand there are a class of solutions that essentially inject grains at random in a system and substep a discrete element method to attain a viable configuration. Again, solving a sub discrete element problem with multiple timesteps needed to achieve an equilibrium state is too expensive.

Therefore, a more guided greedy method is desired. From Figure 5-2 it can be seen that as the grains flow from left to right, space intuitively opens up upstream of the hybrid elements. The next solution attempted utilized this fact, prioritizing grain injection attempts to the downstream-most available spaces in the element, to promote closer packing to the currently existing grains. To do this, the area of an enriched hybrid element was discretized into a grid. The nodes of the grid represent possible injection points. As a note, this presents a parameter that must be tuned: the discretization distance between nodes. Smaller discretizations can



Figure 5-7: Chute flow example zoomed into hybrid zone.

lead to less space between injected points, but also increases the number of attempts that must be made.



Figure 5-8: Grid enrichment scheme.

The scheme would then attempt to inject grains, starting from the downstream side. The downstream direction was determined by checking the surrounding elements of the element currently being enriched; if a continuum element neighbors the hybrid element, then the downstream element side is the side opposite of the shared element side. If a discrete element is adjacent to the enriched element, then that face is taken as the downstream side. If only hybrid elements surround the current element, then a direction is picked at random.

### 5.2.3 Circle Sweep

While the grid scheme was an improvement over the initial random packing, it proved to still be unable to provide a discrete mass source that offset the continuum mass flux into a hybrid cell. The goal of the scheme was to increase the chances of introducing points closer to the existing points; this was achieved, but not to the desired level. To address this deficiency, the next scheme, deemed here the "circle sweep" scheme, strictly forced at least one point of tangency for any injected grain to the other grains in the cell.



Figure 5-9: Circle sweep scheme.

In order to discuss the scheme, take as an example a pure continuum element that at the current hybrid update is now a hybrid cell, and must be enriched. To initialize the scheme, a grain is injected at a random location within the element and is given an index number  $i$  of 1. This index number is added to a queue of indices called the `seed_queue`. The angular space `FINDTERM` of  $2\pi$  around this grain is then discretized into  $n$  angular divisions. Grains injections are then attempted at all of these angular divisions at a distance equal to the sum of the radius of the seed grain  $r_i$  and the new grain  $r_{new}$ , assuring that any new grain has at least one point of tangency to other grains in the element. A new valid grain is given an index number of 1 plus the index number of the grain at the rear of the queue, which itself is then enqueued. Once injection attempts have been made at all of the possible angles around the seed grain, the index of the seed grain is dequeued. The next grain in the `seed_queue` then becomes the seed grain, and the process is continued. The scheme then stops once the queue is empty.

In hybrid elements with DEM grains already present, the circle sweep algorithm begins

by enqueueing all of those grains in the `seed_queue`, instead of injecting a grain at random to initialize the queue. The algorithm then continues as stated, working through the `seed_queue`. Again, this scheme introduces a tunable parameter  $n$ , which controls the number of injection attempts made around every seed grain.

#### 5.2.4 Two Point Tangent

The circle sweep scheme represents an improvement over the grid scheme, but again, for the chute flow geometry was not enough to offset the continual continuum mass flux into the hybrid elements. The final scheme attempted, called the "two-point-tangent" (TPT) scheme, as its name suggests, enforces two points of tangency for any new injected points.

In order to initialize the TPT scheme for a newly enriched element, a grain is injected at random and added to the same `seed_queue` data structure as the circle sweep scheme. The angular region around the grain is again discretized and grain injections are attempted at these angles. The divergence point between TPT and the circle sweep scheme occurs when a valid grain is found and added to the `seed_queue`. Instead of continuing injection attempts at the other remaining possible angles, the scheme stops, and uses these two grains as the starting point of the scheme. With these two tangent grains of radius  $r_a$  and  $r_b$ , a new grain with radius  $r_c$  can be injected that is tangent to both of the first two. Given the center points of the grains  $a$  and  $b$  and their radii, and the radius of the new grain  $r_c$ , an analytical formula can be derived for the two possible center points of the new grain  $c$  that maintains tangency with  $a$  and  $b$ . Injection attempts are then made at these two locations, and any valid attempts are added to the simulation and to the `seed_queue`.

#### 5.2.5 Packing Scheme Results

### 5.3 Hypoelastic Pressure Update

# Bibliography

- [1] Scott G. Bardenhagen and Edward M. Kober. The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–496, 2004.
- [2] Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. TRACKS: Toward directable thin shells. *SIGGRAPH (ACM Transactions on Graphics)*, 26(3):50, Jul 2007.
- [3] Wim A. Beverloo, H. A. Leniger, and J. Van de Velde. The flow of granular solids through orifices. *Chemical Engineering Science*, 15(3):260–269, 1961.
- [4] Jean-Philippe Bouchaud, Michael E. Cates, Jagadeeshan Ravi Prakash, and Sam F. Edwards. A model for the dynamics of sandpile surfaces. *Journal de Physique I*, 4(10):1383–1410, 1994.
- [5] Jes Christoffersen, Morteza Monte Mehrabadi, and Sia Nemat-Nasser. A micromechanical description of granular material behavior. *Journal of applied mechanics*, 48(2):339–344, 1981.
- [6] Peter A. Cundall and Otto D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47–65, 1979.
- [7] Yannis F. Dafalias, Achilles G. Papadimitriou, and Xiang S. Li. Sand plasticity model accounting for inherent fabric anisotropy. *Journal of Engineering Mechanics*, 130(11):1319–1333, 2004.
- [8] Hachmi Ben Dhia. Multiscale mechanical problems: the arlequin method. *Comptes Rendus de l'Academie des Sciences Series IIB Mechanics Physics Astronomy*, 12(326):899–904, 1998.
- [9] Joshua A Dijksman and Martin van Hecke. Granular flows in split-bottom geometries. *Soft Matter*, 6(13):2901–2907, 2010.
- [10] Sachith Dunatunga and Ken Kamrin. Continuum modelling and simulation of granular flows through their many phases. *Journal of Fluid Mechanics*, 779:483–513, 2015.
- [11] Groupement de Recherche Milieux Divisés (GDR MiDi). On dense granular flows. *The European Physical Journal E*, 14(4):341–365, 2004.
- [12] Seth R. Holladay. *Optimized Simulation of Granular Materials*. PhD thesis, Brigham Young University, 2013.

- [13] Seth R. Holladay and Parris Egbert. Solid-state culled discrete element granular systems. In Carlos Andujar and Enrico Puppo, editors, *Eurographics 2012 - Short Papers*. The Eurographics Association, 2012.
- [14] Shu-Wei Hsu and John Keyser. Piles of objects. *ACM Trans. Graph.*, 29(6):155:1–155:6, December 2010.
- [15] Pierre Jop, Yoël Forterre, and Olivier Pouliquen. A constitutive law for dense granular flows. *Nature*, 441(7094):727–730, 2006.
- [16] Ken Kamrin. Nonlinear elasto-plastic model for dense granular flow. *International Journal of Plasticity*, 26(2):167–188, 2010.
- [17] Ken Kamrin and Georg Koval. Nonlocal constitutive relation for steady granular flow. *Physical Review Letters*, 108(17):178301, 2012.
- [18] Ken Kamrin and Georg Koval. Effect of particle surface friction on nonlocal constitutive behavior of flowing granular media. *Computational Particle Mechanics*, 1(2):169–176, 2014.
- [19] Georg Koval, Jean-Noël Roux, Alain Corfdir, and François Chevoir. Annular shear of cohesionless granular materials: From the inertial to quasistatic regime. *Physical Review E*, 79(2):021306, 2009.
- [20] Cornelius Lanczos. *The variational principles of mechanics*. 2012.
- [21] Joseph J. McCarthy and Julio M. Ottino. Particle dynamics simulation: a hybrid technique applied to granular mixing. *Powder Technology*, 97(2):91 – 99, 1998.
- [22] Hiroshi Mio, Masatoshi Akashi, Atsuko Shimosaka, Yoshiyuki Shirakawa, Jusuke Hidaka, and Shinroku Matsuzaki. Speed-up of computing time for numerical analysis of particle charging process by using discrete element method. *Chemical Engineering Science*, 64(5):1019 – 1026, 2009.
- [23] Kevin W. Munns. Gaseous particulate interaction in a 3-phase granular simulation. Master’s thesis, Brigham Young University, 2015.
- [24] Olivier Pouliquen. Scaling laws in granular flows down rough inclined planes. *Physics of Fluids (1994-present)*, 11(3):542–548, 1999.
- [25] Patrick Richard, Mario Nicodemi, Renaud Delannay, Philippe Ribi  re, and Daniel Bideau. Slow relaxation and compaction of granular systems. *Nature Materials.*, 4:121–128, 2005.
- [26] Chris H. Rycroft, Ken Kamrin, and Martin Z. Bazant. Assessing continuum postulates in simulations of granular flow. *Journal of the Mechanics and Physics of Solids*, 57(5):828–839, 2009.
- [27] Andrew Schofield and Peter Wroth. *Critical State Soil Mechanics*. McGraw-Hill, London, 1968.
- [28] Hannah G. Sheldon and Douglas J. Durian. Granular discharge and clogging for tilted hoppers. *Granular Matter*, 12(6):579–585, 2010.

- [29] Tomotsugu Shimokawa, Toshiyasu Kinari, and S Shintaku. Interaction mechanism between edge dislocations and asymmetrical tilt grain boundaries investigated via quasicontinuum simulations. *Physical Review B*, 75(14):144108, 2007.
- [30] Juan C. Simo and Thomas J. R. Hughes. *Computational Inelasticity*. Springer, 1998.
- [31] Gregory S. Smith, Ellad B. Tadmor, Noam Bernstein, and Efthimios Kaxiras. Multiscale simulations of silicon nanoindentation. *Acta Materialia*, 49(19):4089–4101, 2001.
- [32] Russell Smith et al. Open dynamics engine. 2005.
- [33] Daniel E. Sullivan. Materials in use in u.s. interstate highways. USGS Report, October 2006.
- [34] Deborah Sulsky, Zhen Chen, and Howard L. Schreyer. A particle method for history-dependent materials. *Computer methods in applied mechanics and engineering*, 118(1):179–196, 1994.
- [35] Ellad B. Tadmor, Rob Phillips, and Michael Ortiz. Mixed atomistic and continuum models of deformation in solids. *Langmuir*, 12(19):4529–4534, 1996.
- [36] Christian Wellmann and Peter Wriggers. A two-scale model of granular materials. *Computer Methods in Applied Mechanics and Engineering*, 205-208(Supplement C):46 – 58, 2012.
- [37] Beichuan Yan, Richard A. Regueiro, and Stein Sture. Three-dimensional ellipsoidal discrete element modeling of granular materials and its coupling with finite element facets. *Engineering Computations*, 27(4):519–550, 2010.
- [38] Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. Continuum foam: A material point method for shear-dependent flows. *ACM Trans. Graph.*, 34(5):160:1–160:20, November 2015.
- [39] Zhennan Zhang and Xiurun Ge. A new quasi-continuum constitutive model for crack growth in an isotropic solid. *European Journal of Mechanics-A/Solids*, 24(2):243–252, 2005.
- [40] Bo Zhu and Xubo Yang. Animating sand as a surface flow. *Eurographics 2010, Short Papers*, 2010.