A Study of a DEM-Based Granular Dynamics Solver (DRAFT 4)

T. Yee
M.Sc Candidate

C. Christara
University of Toronto

J. Lait
SideFX Software

July 14, 2009

Abstract

Granular dynamics is the dynamics of a large set of small particles (grains). Convincing simulation of natural granular phenomena (e.g. dust, sand or powders) is a challenging mathematical and computational problem. Our observation is that the more realistically the collection of grains approaches its static state, the more natural the simulation appears. This study focuses on the simulation of sets of grains as the set approaches its static state. The method begins with a discrete element (also referred to as molecular dynamics) model of the inter-particle contacts within the granular collection. Inertia terms (friction/dampening) are added to the basic contact model to facilitate static piling. An examination of the different contact models on the formation of the final static state and a discussion of the numerical consequences of each model is presented. The discrete element approach demonstrates to be a straightforward and natural way to model many granular behaviors. Its versatility makes it possible to use it to build a general-purpose granular solver.

1 Introduction

Granular material is a special class of matter that responds to external forces in unconventional ways compared to traditional solids, liquids and gases. The dynamics of a collection of grains exhibit complex behaviors such that there does not yet exist a consistent theory that can encompass all phases of its motion. Yet granular systems are ubiquitous in the natural world and efforts in the literature to model granular behavior are typically only suitable for the granular system studied at hand. It is easy to recognize the value of a general granular dynamics solver that can simulate the response of any granular assembly under a number of external conditions. This work examines the limitations and merits of using a discrete element method as a general granular dynamics solver. The greatest challenge in developing such a solver is ensuring computational efficiency.

1.1 The General Problem

Granular material such as sand, dust or powders exhibit behaviors that is incompletely described by existing solid, liquid or gas models of matter [1]. Small particles (also referred to as grains) can pack like solids and take freestanding shape, flow like liquids and move in a cohesive unit or expand like gases and evolve with complex dynamics. Under appropriate conditions, granular material can behave like any one of these states of matter and can spontaneously transition between these states (e.g. jamming of powders in a hopper, avalanching on the surface of a stable sandpile [4]). A general-purpose granular dynamics solver needs to accommodate this range of motion and facilitate the transition from one dynamic state to another.

Developing appropriate models of granular matter is an active area of research. The majority of models in the literature are constructed to describe a specific granular assembly. Existing models may be categorized as either a grain-scale method, where each grain-grain collision is identified and the bulk behavior is implied; or a continuum-scale method, where the bulk behavior of the collection is described and any grain-level detail is generalized. Within each category, there is also variation in the mathematical description and solution method of the model. Between categories, the motion indicated by one set of models may not coincide with the predicted bulk behavior of the other set of models. A suitable model for a solver should be able to simulate both fine-grain and bulk features of the system; this is the reason grain-scale models are the focus of this work. A way to determine the quality of various grain-scale models is to evaluate how well the model can achieve desired bulk properties.

Grain-scale models pose many computational challenges. Grain-scale models typically begin by calculating the forces between all pairs of grains in contact. Newton's equations of motion are then integrated for each grain. This leads to a large system of differential equations to solve and the method quickly becomes computationally infeasible for real-world systems with a large number of grains. In addition, the computational complexity also increases as the grain geometry becomes more complex. A practical granular dynamics solver needs to be robust to various grain geometries and scalable to a large number of grains. Successful implementations of grain-scale models so far are limited to systems consisting of a few thousand particles.

1.2 Problem Statement and Scope

This work studies the feasibility of using discrete element methods (DEMs) [1] to build a general-purpose granular dynamics solver. In principle, DEMs are suitable to model composite solids, liquids and gases. However, existing instances of discrete element models do not necessarily encompass the many phases of granular behavior. Also, in many cases, DEMs are computationally too expensive to be a practical solver. To evaluate the proposed solver, we focus on simulating the dynamics of a collection of grains as it approaches a static state. We want to observe a set of grains transitioning from a kinetic regime to a static regime, possibility interacting with obstacles and/or boundaries in between, and settling into a static state that we identify by observing evidence of stable granular piles. The study begins with a basic discrete element model of granular dynamics [2], [17]. The model is then extended to include inertia terms so that the collection can form granular heaps. The solver is finally evaluated by its ability to form realistic static heaps under contrasting formation conditions. The computational cost is directly related to the number of inter-particle and particle-object calculations needed to simulate the response. The computational expense of this model under each scenario is examined.

1.3 Background

Modern simulations of natural granular phenomena began in the 80's, and their main application was to study industrial processes that suffered unexplained inefficiencies. Physically-based models were developed and numerically approximated for each specific application. Due to the different situations that these models arise from, three distinct approaches to grain-scale models of granular systems are developed: discrete element methods (DEM), rigid body dynamics (RBD) and cellular automata (CA).

Discrete element methods typically use is a soft-shape method where small overlapping of grains due to collisions is allowed and these overlaps are used to estimate the resulting impulse forces that govern the motion. DEMs are well suited for problems where the contact forces can be determined from small mutual deformations. This makes the method immediately appropriate for granular systems of soft to moderately hard particles that exhibit fast dynamics. It's main disadvantage is that large impulse forces in the equations of motion can cause the typical explicit integration scheme to use small time steps.

Rigid body dynamics is a constraint-based approach where the contact forces of each collision must be resolved by defining the so-called Linear Complementarity Problem (LCP) and searching for a set of forces that satisfy the no-penetration constraints. RBD methods are well-suited for granular systems consisting of very hard materials where there is little mutual deformation and the system dynamics is considered slow. In contrast to DEMs, the RBD approach does not require an explicit definition of the interacting forces but solves for the forces from a set objective functions and constraints. However, a direct consequence of not modeling the contact forces is that specific material properties are not included. Also, the contact forces computed by RBD are, in general, only one set out of infinitely many sets of forces that satisfy the equations and constraints; thus extra work may be needed to assure the system behaves in a physically-realistic way.

[1] Also referred to as molecular dynamics methods (MD).

The computation of each step in RBD is more complicated than DEMs but the overall computational cost can be similar to that of DEMs because larger stepsizes may be taken.

Cellular automata is a rule-based model where the domain of the simulation is discretized into a finite grid and simple rules are defined at each grid point such that they dictate the motion of a particle as it passes the point. This model applies the idea of self-organized criticality (SOC) where complex systems may emerge from simple local interactions defined by these automaton rules. The appeal of SOC is the fact that the behavior of the system is insensitive to any control parameters; changing fine details of the model does not affect the system from reaching its critical behavior. This method has been successfully used to model heap growth, avalanches, dune and ripple patterns in granular systems. The CA approach is capable of explaining granular phenomena. However, it is unclear how to define the automaton rules such that the method can be used as a general-purpose solver.

In modern computer graphics, particle-based models, such as smoothed particle hydrodynamics (SPH), have become methods of choice to model the motion of deformable substances, liquids, and cloth. SPH is popular because it is possible to compute a system's motion in real-time for a large number of situations. However, these particle-based methods are not immediately appropriate for granular materials because they do not model real material properties and inertial forces that are present in systems of grains. Instead, is is more suitable to choose a grain-scale model of granular dynamics and evolve the model to include a wider range of granular motion.

There have been other attempts to adapt fluid solvers to simulate granular systems [5]. This approach addresses the computational limitation of grain-scale models. However, on close examination of the simulation results, the inability to simulate scattering or shearing of subsets of grains from the collection makes the dynamics seem suitable for very fine powders but not for coarse grain collections like sand or gravel. Retrofitting a fluid solver to be a granular solver may only produce limited granular behavior.

2 A Granular Dynamics Solver

A general-purpose granular dynamics solver is designed to be a black box solver that can simulate the dynamics of an arbitrary set of grains given the geometry of the grains and material properties. Ideally, the end-user can control the grain behavior by adjusting the exposed parameters. The ease of which the end-user can operate the solver is directly related to the complexity of the contact force models.

Within the solver, the solver needs to decide on how to represent the grain geometry, how to use the geometry and material properties to model grain-grain and grain-object interaction and how to approximate the motion from the model efficiently. The grain geometry, material properties and computation technique all have an affect on the formation of the final static state. The techniques used at each stage of the solver determines the robustness of this approach.

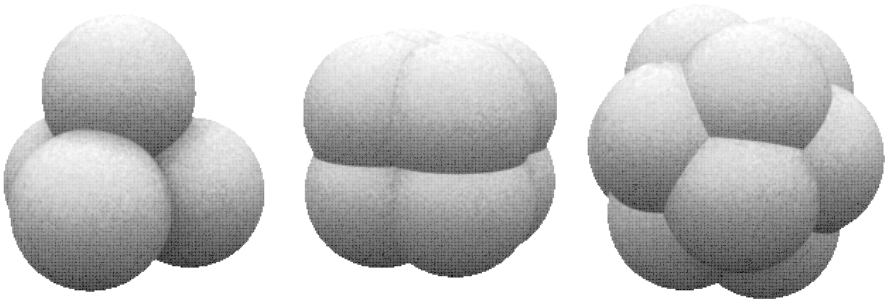

Figure 1: Example of grains composed of rigidly connected spheres.

2.1 DEM-based Granular Dynamics Solver

The general workflow for this solver is:

1. Parameterize the grain.
2. Define the contact forces between grains.
3. Define the contact forces between a grain and external objects.
4. Form the equations of motion.
5. Numerically solve the equations of motion.

The grain geometry is typically either approximated with triangles, spheres, or another finite element discretization of the surface is used. Triangulation is favorable since any grain shape may be triangulated to an arbitrary level of detail. However, this geometry leads to complicated equations in the model that adds to the complexity of the computation. Approximating grain shapes with composite spheres is less precise but the equations in the model are simpler to define and easier to compute. Like triangulation, finite element discretization of the grain surface incurs a significant computational cost. This level of detail is only necessary when object deformation is an important component of the simulation.

This solver is developed to use grains approximated by a set of spheres. The resulting contact force at a grain-grain interaction is easily estimated by the amount of overlap of the spheres. A similar force may be estimated at grain-object collisions. Forming the equations of motion and solving this system of equations efficiently at each time step determines the cost and therefore the feasibility of this solver.

2.2 Grain Geometry: Composite Grain

Each grain in a granular system is approximated by a set of rigidly connected spheres (Figure 1). The composite sphere surface approximates the grain's surface and can easily be used to approximate any grain shape. The grain shape itself can be used to model static friction [6]. A grain shape with significant concavity is a geometric mechanism that promotes interlocking of particles and thus a model of static friction. This work chooses to use simple convex grain shapes for its simulations to keep the mass and moments of inertia simple. Variations on the composite grain use springs to connect the spheres. This extra level of complexity is not considered crucial for the solver.

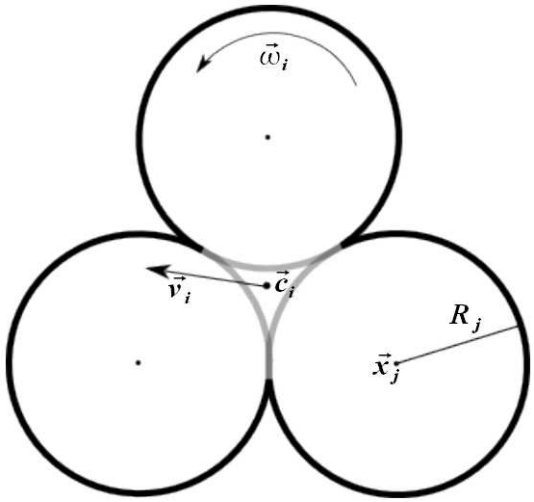

Figure 2: Grain Parameters.

An explanation of the notation used in Figure 2 is given as follows. The index of a grain is denoted by i. For each grain i, the centre-of-mass parameters are defined to be:

$$\begin{aligned}
\vec{c}_i &= (c_{i,1}, c_{i,2}, c_{i,3}) && \text{position of grain } i. \\
\vec{v}_i &= (v_{i,1}, v_{i,2}, v_{i,3}) && \text{velocity of grain } i. \\
\vec{o}_i &= (o_{i,1}, o_{i,2}, o_{i,3}) && \text{orientation of grain } i. \\
\vec{\omega}_i &= (\omega_{i,1}, \omega_{i,2}, \omega_{i,3}) && \text{angular velocity of grain } i.
\end{aligned}$$

The index of a sphere that comprises a grain is denoted by j. For each sphere j, the following coordinates are also defined:

$$\begin{aligned}
\vec{x}_j &= (x_{j,1}, x_{j,2}, x_{j,3}) && \text{position of sphere } j \text{ in some grain.} \\
R_j &&& \text{radius of sphere } j \text{ in some grain.}
\end{aligned}$$

Since the spheres in the composite grains are rigidly connected, the velocity and angular velocity of sphere j are the same as those of grain i:

$$\begin{aligned}
\vec{v}_{ij} &= (v_{ij,1}, v_{ij,2}, v_{ij,3}) = \vec{v}_i && \text{velocity of sphere } j \text{ in grain } i. \\
\vec{\omega}_{ij} &= (\omega_{ij,1}, \omega_{ij,2}, \omega_{ij,3}) = \vec{\omega}_i && \text{angular velocity of sphere } j \text{ in grain } i.
\end{aligned}$$

In subsequent sections, the grain index i is dropped from the notation for the velocity and angular velocity of sphere j. For brevity of notation, let $\vec{v}_j \equiv \vec{v}_{ij}$ and $\vec{\omega}_j \equiv \vec{\omega}_{ij}$ refer to the velocity and angular velocity, respectively, of a sphere j that belongs to some grain i; similarly, $\vec{v}_k$ and $\vec{\omega}_k$ refer to the velocity and angular velocity respectively of a sphere k on another grain i'. When the identity of the grain is important, the index i is explicitly included. If references to specific grains or spheres are ambiguous, the notation $\vec{v}_{i=2,j=3}$ or $\vec{v}_{k=5}$ will be used.

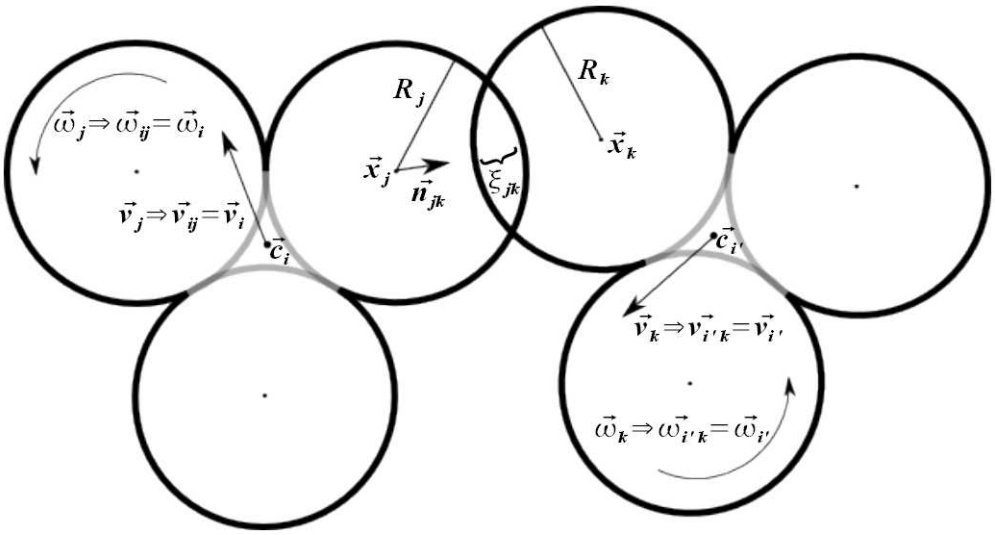

Figure 3: Grain-Grain Contact Quantities.

Grains used in this work are approximated by identical spheres. This assumption is non-essential and may be relaxed to model more realistic grain shapes. However, the potential of using composite spheres to model sharp-edged grains is limited. This solver is best used for granular motion where very sharp-angled grains are not fundamental to the dynamics.

2.3 Contact Forces

In soft-sphere discrete element methods, the (repulsive) forces, $\vec{F}_i$, and torques, $\vec{\tau}_i$, acting on each grain are given by sums of pairwise interaction of the grain with all other grains and objects, that is,

$$\begin{aligned} \vec{F}_i &= \sum_{j,k} \vec{F}_{jk} + \sum_{j,q} \vec{F}_{jq} \\ \vec{\tau}_i &= \sum_{j,k} \vec{\tau}_{jk} + \sum_{j,q} \vec{\tau}_{jq} \end{aligned}$$

where j is the index of the j^{th} sphere of grain i, k is the index of the k^{th} sphere of the other grain i' and q is the index of the q^{th} object in contact with grain i.

Soft-sphere simulations does not explicitly enforce the no-penetration constraint among its grains but defines forces to apply this condition. These forces are derived from measurable quantities during mechanical contact.

2.3.1 Contact Quantities

In a granular assembly, grain collision may be quantified by defining values that describe the mutual compression/deformation of spheres j and k during the collision of two grains i and i':

$$
\begin{aligned}
\xi_{jk} &= \max(0, R_j + R_k - |\vec{x_j} - \vec{x_k}|) \\
\dot{\xi_{jk}} &= (\vec{v_j} - \vec{v_k}) \cdot \vec{n}_{jk} \\
\vec{n}_{jk} &= \frac{\vec{x_k} - \vec{x_j}}{|\vec{x_k} - \vec{x_j}|} \\
\vec{z}_{jk} &= \vec{x_j} + (R_j - \frac{\xi_{jk}}{2})\vec{n}_{jk} \\
\vec{V}_{jk} &= \vec{v_k} - \vec{v_j} + \vec{\omega}_k \times (\vec{z}_{jk} - \vec{c}_{i'}) - \vec{\omega}_j \times (\vec{z}_{jk} - \vec{c}_i) \\
\vec{W}_{jk} &= \vec{\omega}_k \times (\vec{z}_{jk} - \vec{c}_{i'}) - \vec{\omega}_j \times (\vec{z}_{jk} - \vec{c}_i).
\end{aligned}
\tag{1}
$$

In the above, ξ_{jk} is the inter-penetration of grains that is a measure of the mutual compression/deformation of spheres j and k, $\dot{\xi_{jk}}$ is the relative velocity of the compression, $\vec{n}_{jk}$ is the direction of the compression, $\vec{z}_{jk}$ is a contact point used to represent the contact area, $\vec{V}_{jk}$ and $\vec{W}_{jk}$ are the instantaneous relative velocity and angular velocity, respectively, of the contact point respectively.

The forces, $\vec{F}_{jk}$, and torques, $\vec{\tau}_{jk}$, acting at a collision between spheres j and k of two different grains are derived from the above contact quantities. Their explicit dependencies may be written as:

$$
\begin{aligned}
\vec{F}_{jk} &= \vec{F}_{jk}(\vec{x}_j, \vec{x}_k, \vec{v}_j, \vec{v}_k, \vec{o}_j, \vec{o}_k, \vec{\omega}_j, \vec{\omega}_k) \\
\vec{\tau}_{jk} &= \vec{\tau}_{jk}(\vec{x}_j, \vec{x}_k, \vec{v}_j, \vec{v}_k, \vec{o}_j, \vec{o}_k, \vec{\omega}_j, \vec{\omega}_k)
\end{aligned}
$$

The notation $\vec{F}_{jk}$ and $\vec{\tau}_{jk}$ is used as a convenient shorthand. The discussion of forces and torques acting at a collision point is more easily described in terms of their normal and tangential components:

$$
\begin{aligned}
\vec{F}_{jk} &= \begin{cases} \vec{F}^n_{jk} + \vec{F}^t_{jk} & \text{if } \xi_{jk} > 0 \\ 0 & \text{otherwise} \end{cases} \\
\vec{\tau}_{jk} &= \begin{cases} \vec{R_j} \times \vec{F}^t_{jk} & \text{if } \xi_{jk} > 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Expressions for the normal and tangential forces, $\vec{F}^n_{jk}$ and $\vec{F}^t_{jk}$, and subsequently the torques are still an active area of research ([18] [19] [20] [21] [22] [23]). In particular, different models of the tangential forces give rise to different granular dynamics. Several accepted models are studied and presented. All models work best when the inter-penetration of grains, ξ_{jk}, is small. The solver should monitor that this condition is always valid.

2.3.2 Contact Forces: Basic Model

The most basic model of forces and torques acting at a collision is

$$
\begin{aligned}
\vec{F}^n_{jk} &= (-k_\alpha \xi^{\dot{\alpha}}_{jk} \xi_{jk} - k_\beta \xi^{\beta}_{jk}) \vec{n}_{jk} \\
\vec{F}^t_{jk} &= -\min(\mu_s |\vec{F}^n_{jk}|, k_t |\vec{V}_{jk}|) \frac{\vec{V}_{jk}}{|\vec{V}_{jk}|} \\
\vec{\tau}_{jk} &= (\vec{z}_{jk} - \vec{c}_i) \times \vec{F}^t_{jk}
\end{aligned}
\tag{2}
$$

where $\alpha, \beta, k_\beta, k_\alpha, k_t, \mu_s$ are model and material-dependent constants. The normal force is a widely accepted model of the interaction of viscoelastic spheres derived by N. V. Brilliantov et al. [8]. The tangential force is a simple ansatz of a frictional force that opposes the tangential velocity limited by Coulomb friction condition. By definition, torques are derived from the forces. An instance of this model is in work by Bell. et al. [2].

This model of inter-grain collisions is straightforward to implement and has low computational cost. For simulations where rapid dynamics is the focus, this model is generally adequate. However, the tangential repulsive force is not sufficient to form stable piles. For small relative velocities ($\vec{V}_{jk}$), the tangential force vanishes so there is no longer an opposing force to further decelerate the particle. The result is a collection of grains that slowly collapses under its own weight. See Section 3.1 for an example.

2.3.3 Contact Forces: Simple Static Friction Model

A simple extension of the basic model includes a term to describe surface friction acting in the plane tangential to the contact normal and a rolling resistance to torque forces. Thus the contact forces become

$$
\begin{aligned}
\vec{F}^n_{jk} &= (-k_\alpha \xi^{\dot{\alpha}}_{jk} \xi_{jk} - k_\beta \xi^{\beta}_{jk}) \vec{n}_{jk} \\
\vec{F}^t_{jk} &= -\mathrm{sign}(\delta_{jk}) \min(\mu_s |\vec{F}^n_{jk}|, -k_t \delta_{jk}) \frac{\vec{V}_{jk}}{|\vec{V}_{jk}|} \\
\vec{\tau}_{jk} &= (\vec{z}_{jk} - \vec{c}_i) \times \vec{F}^t_{jk} - \mu_r |\vec{F}^n_{jk}| \frac{\vec{W}_{jk}}{|\vec{W}_{jk}|}
\end{aligned}
\tag{3}
$$

where

$$
\delta_{jk} = \int_{t_0}^{t_n} |\vec{V}_{jk}(t)| dt \tag{4}
$$

is the contact displacement for a contact that persists from time t_0 to t_n, and μ_r is the rolling friction coefficient. Contact displacement is basically a cumulative scalar measure of the distortion of the contacting surfaces due to rolling and sliding limited by Coulomb friction.

T. Elperin et al. [7] used this model to simulate stable granular piles for spherical grains. However, this model is only able to form piles if the grains in the system are packed sufficiently slowly. The model does not have enough mechanisms to dissipate the kinetic energy in any system rapidly enough to achieve piling

from arbitrary formation conditions. In Elperin et al's original work grain velocities were artificially set to zero at a fixed number of intervals at the beginning of the simulation to decrease the effects of grains with large inertia. Although the formation conditions of this model are limited, the endurance of contacts, δ_{jk}, is shown to be an inherent component of any static friction model.

2.3.4 Contact Forces: Static Friction Model with Dampening

An adaptation of the simple static friction model adds dampening terms to handle grains with large inertia. This placates the need to manually interfere with pile construction and is effective at simulating static piles for many formation conditions. In this model, the contact forces are

$$\begin{aligned} \vec{F}^n_{jk} &= (-k_\alpha \dot{\xi^\alpha_{jk}} \xi_{jk} - k_\beta \xi^\beta_{jk}) \vec{n}_{jk} \\ \vec{F}^t_{jk} &= -\mu_s |\vec{F}^n_{jk}| (1 - (\frac{1 - \min(|\vec{\delta}_{jk}|, \delta_{\max})}{\delta_{\max}})^\gamma) \frac{\vec{\delta}_{jk}}{|\vec{\delta}_{jk}|} + d_t \vec{V}^t_{jk} \\ \vec{\tau}_{jk} &= (\vec{z}_{jk} - \vec{c}_i) \times \vec{F}^t_{jk} - \mu_r |\vec{F}^n_{jk}| \frac{\vec{W}_{jk}}{|\vec{W}_{jk}|} \end{aligned} \tag{5}$$

where $\gamma, d_t, \delta_{\max}$ are a material-dependent constants.

The tangential friction, $\vec{F}^t_{jk}$, is Coulomb friction re-written to use $\vec{\delta}_{jk}$, the vector of accumulated tangential displacement between spheres j and k, given by

$$\vec{\delta}_{jk} = \int_{t_0}^{t_n} \vec{V}_{jk}(t) dt.$$

When $|\vec{\delta}_{jk}| << \delta_{\max}$, the tangential force is in an elastic response region. When $|\vec{\delta}_{jk}| \rightarrow \delta_{\max}$, the repulsive force maximizes to $\mu_s |\vec{F}^n_{jk}|$.

Investigators ([9][10]) have used instances of this model to study granular piling. Simulations show successful heap formation by a series of avalanches in the boundary layer of the pile. The dampening terms assist the granular system to reach a static state but its role in pile formation is secondary to contact displacement. Even with this model of static friction, the stability of the granular pile is still sensitive to its formation conditions. Numerically, the final heap is not absolutely still; the kinetic energy is never completely dissipated and residual kinetic energy manifests as local vibrations within the pile. A study of the sensitivities of a granular pile to friction is given [16].

2.3.5 Contact Forces: Grain-Object Friction Model

Particles that collide with external objects also experience repulsive forces. To adapt the technique used for spheres, the object surface is assumed to be smooth such that the portion of the object surface in contact with the grain may be approximated as an infinitely large sphere. The basic model may then be used to estimate

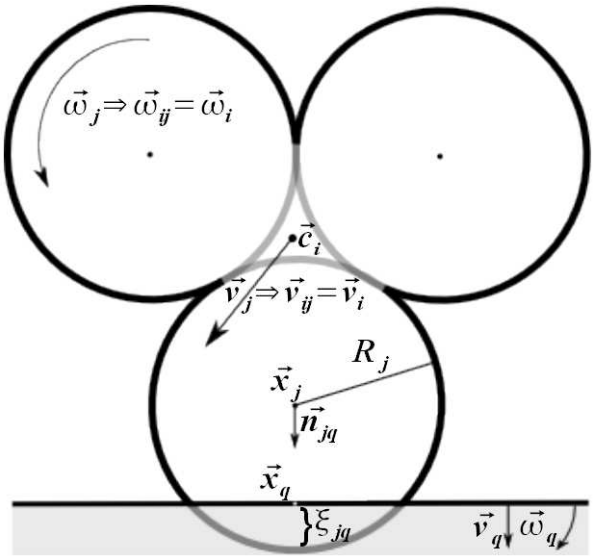

Figure 4: Grain-Object Contact Quantities.

the resulting forces such that the contact quantities of sphere j of particle i with object q, may be described by

$$\begin{aligned}
\xi_{jq} &= \max(0, R_j - |\vec{x_j} - \vec{x_q}|) \\
\dot{\xi_{jq}} &= (\vec{v_j} - \vec{v_q}) \cdot \vec{n}_{jq} \\
\vec{n}_{jq} &= \frac{\vec{x_q} - \vec{x_j}}{|\vec{x_q} - \vec{x_j}|} \\
\vec{z}_{jq} &= \vec{x_j} + (R_j - \frac{\xi_{jq}}{2})\vec{n}_{jq} \\
\vec{V}_{jq} &= \vec{v_q} - \vec{v_j} + \vec{\omega}_q \times (\vec{z}_{jq} - \vec{c}_{i'}) - \vec{\omega}_j \times (\vec{z}_{jq} - \vec{c}_i) \\
\vec{W}_{jq} &= \vec{\omega}_q \times (\vec{z}_{jq} - \vec{c}_{i'}) - \vec{\omega}_j \times (\vec{z}_{jq} - \vec{c}_i).
\end{aligned}$$

where the contact area is approximated by the point, $\vec{x}_q$, which is a point on object q that is closest to $\vec{x}_j$, the centre of the contacting sphere j.

Adapting from the basic model, the repulsive force at a particle-object collision may instantaneously be modeled as

$$\begin{aligned}
\vec{F}^n_{jq} &= (-k_a \dot{\xi^{\alpha}_{jq}} \xi_{jq} - k_b \xi^{\beta}_{jq}) \vec{n}_{jq} \\
\vec{F}^t_{jq} &= -\min(\mu_q |\vec{F}^n_{jq}|, k_q |\vec{V}_{jq}|) \frac{\vec{V}_{jq}}{|\vec{V}_{jq}|} \\
\vec{\tau}_{jq} &= (\vec{z}_{jq} - \vec{c}_i) \times \vec{F}^t_{jq} - \mu_\tau |\vec{F}^n_{jq}| \vec{\omega}_j
\end{aligned} \tag{6}$$

where $k_a, k_b, \mu_q, k_q, \mu_\tau$ are material constants.

However, recall that this model does not model static friction. For particle-object collisions, static friction is of particular importance. Of immediate relevance is the substrate that a granular pile is formed on. A static friction model for particle-substrate collisions needs to be able to bring the particles to rest in a short time interval and remain at rest such that the particles can form a rough base for subsequent particles to rest on. It is necessary for the solver to be able to construct a stable foundation of particles at rest in order to observe piling.

Previous simulations modeled objects as surfaces covered with spheres on the scale of the grains [2][7][10]. Consequently, the substrate is a rough base made of discrete particles the same size as the grains in the system. For a general-purpose solver, this prerequisite undesirably adds an extra layer of complexity to the geometry of objects. Also, for complex simulations where there may be heterogeneous grains and dynamics, it is unclear how to form this surface in a controlled way to affect the static state. Instead, this solver explicitly enforces the static friction (stiction) condition in order to model the large impulses needed to bring (and keep) the grains quickly to rest.

In mechanics, there does not exist an explicit expression for static friction. If a body is sliding, the dynamic frictional force is known to be $|\vec{F}_{friction}| = \mu_d|\vec{F}^n|$; however, if a body is experiencing stiction, the static frictional force is unknown except to satisfy the observed condition $|\vec{F}_{friction}| \leq \mu_s|\vec{F}^n|$. That is, when a body is in "stiction", the frictional force is such that no relative acceleration is observed.

Conceptually, this means that if a particle is able to resist any applied forces, static friction occurs such that it opposes the tangential component of the applied forces as well as incurs an additional force that acts to bring the grain to rest. If $\vec{t}_{jq}$ is the unit vector in the tangential direction of motion between the grain and object and $\vec{V}_{jq}$ is the relative velocity of the grain and object, then the stictional force, $\vec{F}^s_{jq}$, may be expressed as

$$\vec{F}^s_{jq} = -\sum_{k=k_0}^{K} (\vec{F}^t_{jk} \cdot \vec{t}_{jq})\vec{t}_{jq} - (\vec{F}^t_{jq} \cdot \vec{t}_{jq})\vec{t}_{jq} - k_s \frac{(\vec{V}_{jq} \cdot \vec{t}_{jq})\vec{t}_{jq}}{t_q} \tag{7}$$

where K is the number of spheres of another grain in contact with sphere j, k_s is a constant and t_q is the impulse duration. So, during stiction, the frictional force negates any tangential forces acting on the grain as well as contributes a force, $-k_s\frac{(\vec{V}_{jq}\cdot\vec{t}_{jq})\vec{t}_{jq}}{t_q}$ that acts to bring the grain to rest. An analogous set of conditions may be defined to describe the torques during stiction, that is,

$$\vec{\tau}^s_{jq} = -\sum_{k=k_0}^{K} (\vec{\tau}^t_{jk} \cdot \vec{n}_{jq})\vec{n}_{jq} - (\vec{\tau}^t_{jq} \cdot \vec{n}_{jq})\vec{n}_{jq} - k_\tau \frac{\vec{W}_{jq}}{t_q}$$

where $\vec{W}_{jq}$ is defined in (1). This solver implements this stiction condition in order to model the slip-stick motion of grain-object collisions.

2.4 Equations of Motion

Newton's equations of motion govern the dynamics of the grains in the system. The equations are defined for the center-of-mass coordinates of each grain. The forces and torques evolve as the granular configuration changes. The resulting equations of motion for one grain is system of second-order ordinary differential equations (ODEs) that is written as a system of coupled first-order ODEs.

$$\begin{aligned}
\dot{\vec{c}}_i &= \vec{v}_i \\
\dot{\vec{v}}_i &= \vec{F}_i = \vec{g} + \frac{1}{m_i} \sum_{j,k,q} (\vec{F}^n_{jk} + \vec{F}^t_{jk} + \vec{F}^n_{jq} + \vec{F}^t_{jq} + \vec{F}^s_{jq}) \\
\dot{\vec{o}}_i &= \vec{\omega}_i \\
\dot{\vec{\omega}}_i &= \vec{\tau}_i = \frac{1}{I_i} \sum_{j,k,q} (\vec{\tau}_{jk} + \vec{\tau}_{jq} + \vec{\tau}^s_{jq})
\end{aligned}$$

where m_i is mass and I_i is the tensorial moment of inertia of grain i. For the simple grain shapes used in this work, the moment of inertia is approximated by the moment of an approximate sphere which simplifies I_i to a scalar value. The forces, $\vec{F}$, and torques, $\vec{\tau}$, are defined in Section 2.3. The notation $\sum_{j,k,q}$ means the sum is over all other particles and objects that are in contact with the spheres that compose grain i.

The entire ODE system to be solved for a system of N grains is

$$\dot{\vec{y}} = \vec{f}(t, y)$$

where $\vec{y} = (\vec{c}_1, ..., \vec{c}_N, \vec{v}_1, ..., \vec{v}_N, \vec{o}_1, ..., \vec{o}_N, \vec{\omega}_1, ..., \vec{\omega}_N)^T$. Explicitly, the ODE system is:

$$\begin{bmatrix}
\dot{\vec{c}}_1 \\ \vdots \\ \dot{\vec{c}}_N \\ \dot{\vec{v}}_1 \\ \vdots \\ \dot{\vec{v}}_N \\ \dot{\vec{o}}_1 \\ \vdots \\ \dot{\vec{o}}_N \\ \dot{\vec{\omega}}_1 \\ \vdots \\ \dot{\vec{\omega}}_N
\end{bmatrix} =
\begin{bmatrix}
\vec{v}_1 \\ \vdots \\ \vec{v}_N \\
\vec{g} + \frac{1}{m_1} \sum_{j,k,q} (\vec{F}^n_{jk} + \vec{F}^t_{jk} + \vec{F}^n_{jq} + \vec{F}^t_{jq} + \vec{F}^s_{jq}) \\ \vdots \\
\vec{g} + \frac{1}{m_N} \sum_{j,k,q} (\vec{F}^n_{jk} + \vec{F}^t_{jk} + \vec{F}^n_{jq} + \vec{F}^t_{jq} + \vec{F}^s_{jq}) \\
\vec{\omega}_1 \\ \vdots \\ \vec{\omega}_N \\
\frac{1}{I_1} \sum_{j,k,q} (\vec{\tau}_{jk} + \vec{\tau}_{jq} + \vec{\tau}^s_{jq}) \\ \vdots \\
\frac{1}{I_N} \sum_{j,k,q} (\vec{\tau}_{jk} + \vec{\tau}_{jq} + \vec{\tau}^s_{jq})
\end{bmatrix}$$

This system of coupled ODEs is typically solved by explicit numerical schemes. Implicit methods may be used to solve the system but explicit schemes are historically preferred due to their simplicity [14]. The equations, however, may be stiff; any explicit method chosen may require small timesteps to ensure stability.

2.5 Numerical Solution

Other researchers [11] conclude that the Runge-Kutta methods provide a good trade-off between efficiency and accuracy for discrete element simulations. The adaptive Runge-Kutta-Fehlberg method (RKF45) is often the method of choice to integrate the equations of motion. This fourth-order method is considered computationally expensive per step, but it is still preferred because its accuracy allows larger time steps to be taken when compared to lower-ordered methods in the same family. Furthermore, the integration of the equations of motion is not the dominant cost in the total cost of approximating a given granular system; the dominant cost is in evaluating the contact forces that comprise the equations, therefore, we do not expect a major change in efficiency by using lower order methods. Our solver uses the RKF45 method by default. For comparison, in the analysis section, we present some results from the application of a lower order RK formula.

2.5.1 The Algorithm

The RKF45 formula computes the positions, velocities, orientations and angular velocities of every grain at each time step in the simulation. Each RKF45 time step requires six derivative evaluations (also referred to as stages). Each evaluation requires the determination of the forces and torques acting at a point in time. After each accepted time step, the contact displacement (if used) must be updated.

Algorithm 1 describes the procedure for solving for all grain parameters at each timestep in the simulation. A variation of the standard method is used to form the forces and torques in each evaluation of $f(t, \vec{y})$. The forces are computed in succession (lines $7 - 11$): instead of computing all forces at $(t, \vec{y})$, each force is computed in sequence, where after each force is determined, the grain parameters, $\vec{y}$, are updated to reflect the affects of the force before the next force is computed. This way, subsequent forces react to previous forces and this acts to dampening oscillations in the system due to the non-linear repulsive forces. This means the order that the forces are determined may affect approximation. For simulations in this work, only grain-grain, grain-object and gravitational forces are considered. The order in which these forces are evaluated in this solver do not significantly affect the results.

The contact displacement vector, $\vec{\delta}_t$, has elements defined to be the non-zero contact displacement quantities, $\vec{\delta}_{jk}$, that are active at time t. Contact displacement is an accumulated quantity that needs to be maintained after a successful integration step. At each step, the contact displacements are assumed to remain active during the entire timestep; the contact displacement vector is not updated during an RKF45 stage. This means the initialization of new contacts may be underestimated by less than one timestep. The effect of this assumption on the overall dynamics of the system is minimal.

$t = 0$; $h = h_0$; $\vec{y}_t = \vec{y}_0$; // Initialization

while $t \leq T$ **do**

  // Solve 6 stages of RKF45

  foreach $s = 1 : 6$ **do**

    // Set-up stage s. γ, α are RKF45 stage constants

    $t^s = t + \gamma^s h$

    $\vec{y}_t^s = \vec{y}_t + \sum_{\hat{s}=1}^{s-1} \vec{\alpha}^{s,\hat{s}} \vec{k}^{\hat{s}}$

    // Estimate $f(t^s, \vec{y}_t^s)$. Apply forces, $\vec{F}$, incrementally.

    $$\vec{y}_t^{s'} = \begin{cases} \vec{c}_i^s + h\vec{v}_i^s & \text{for } i = 1..N \\ \vec{v}_i^s & \text{for } i = 1..N \\ \vec{o}_i^s + h\vec{\omega}_i^s & \text{for } i = 1..N \\ \vec{\omega}_i^s & \text{for } i = 1..N \end{cases}$$

    foreach $\vec{F} = (m\vec{g}, \sum \vec{F}_{jk} + \vec{F}_{jq}, ...)$ **do**

      // Compute the force and torque.

      $\vec{F}(t^s, \vec{y}_t^{s'})$

      $\vec{\tau}(t^s, \vec{y}_t^{s'})$

      // Apply the force and torque to $\vec{y}_t^{s'}$.

      $$\vec{y}_t^{s'} = \vec{y}_t^{s'} + \begin{cases} \frac{h^2}{2m_i}\vec{F}_i & \text{for } i = 1..N \\ \frac{h}{m_i}\vec{F}_i & \text{for } i = 1..N \\ \frac{h^2}{I_i}\vec{\tau}_i & \text{for } i = 1..N \\ \frac{h}{I_i}\vec{\tau}_i & \text{for } i = 1..N \end{cases}$$

    end

    // Calculate s-stage of RKF45

    $$\vec{k}^s = hf(t^s, \vec{y}_t^s) = \begin{cases} h\vec{v}_i^s & \text{for } i = 1..N \\ \vec{v}_i^{s'} - \vec{v}_i^s & \text{for } i = 1..N \\ h\vec{\omega}_i^s & \text{for } i = 1..N \\ \vec{\omega}_i^{s'} - \vec{\omega}_i^s & \text{for } i = 1..N \end{cases}$$

  end

  // Integrate Step. β's are RKF45 step constants

  $\vec{y}_{t+1,RK5} = \vec{y}_t + \sum_{\hat{s}=1}^{6} \beta_5^{\hat{s}} \vec{k}^{\hat{s}}$

  $\vec{y}_{t+1,RK4} = \vec{y}_t + \sum_{\hat{s}=1}^{5} \beta_4^{\hat{s}} \vec{k}^{\hat{s}}$

  $h_{opt} = \left(\frac{\epsilon h}{2 \max(||\vec{c}_{t+1,RK5} - \vec{c}_{t+1,RK4}||)} \right)^{\frac{1}{4}} h$; // Optimal time step.

  if $h_{opt} >= 0.75h$ **then**

    // Accept this step.

    $t = t + h$;

    $\vec{y}_{t+1} = \vec{y}_{t+1,RK4}$

    $\vec{\delta}_{t+1} = \vec{\delta}_t + h\vec{v}$; // Update all contact displacements.

  end

  $h = \min(\max(h_{opt}, h_{min}), h_{max})$

end

Algorithm 1: General RKF45 Procedure for Granular Solver.

Algorithm 2 describes the procedure for computing the forces and torques among grains and between grains and objects at each stage of the RKF45 method (lines 8-9 in Algorithm 1).

Input: $t, \vec{y}$
Output: $\vec{F}(t,\vec{y}), \vec{\tau}(t,\vec{y})$
// Update hash grid
for $i = 1 : N$ **do**
  hashG $\leftarrow$ insertHashGrid($\vec{c}_i$)
end
// For each grain i
for $i = 1 : N$ **do**
  $\vec{F}_i = 0$
  $\vec{\tau}_i = 0$
  // for each sphere j in grain i
  for $j = 1 : J$ **do**
    collideG $\leftarrow$ hashG(i, j) ; // Find inter-grain collisions
    foreach k *in collideG* **do**
      // Compute forces and torques
      $\vec{F}_i = \vec{F}_i + \vec{F}^n_{jk} + \vec{F}^t_{jk}$
      $\vec{\tau}_i = \vec{\tau}_i + \vec{\tau}_{jk}$
    end
    collideObjs $\leftarrow$ getObjs(i, j) ; // Find grain-object collisions
    $\vec{N} = \vec{0}$; // Average collision normal
    foreach q *in collideObjs* **do**
      // Compute forces, torques and normal
      $\vec{F}_i = \vec{F}_i + \vec{F}^n_{jq} + \vec{F}^t_{jq}$
      $\vec{\tau}_i = \vec{\tau}_i + \vec{\tau}_{jq}$
      $\vec{N} = \vec{N} + \vec{N}_q$
    end
  end
  $\vec{N} = \vec{N}/||\vec{N}||$
  $\vec{F}^n_i = \vec{F}_i \cdot \vec{N}$
  $\vec{F}^t_i = \vec{F}_i - \vec{F}^n_i$
  if $\vec{F}^t_i < \mu_q \vec{F}^n_i$ **then**
    // Apply stiction conditions
    $\vec{F}_i = \vec{F}^n_i \vec{N} - k_s \frac{\vec{v}_i - (\vec{v}_i \cdot \vec{N})\vec{N}}{h}$
    $\vec{\tau}_i = \vec{\tau}_i - (\vec{\tau}_i \cdot \vec{N})\vec{N} - k_\tau \frac{\vec{\omega}_i}{h}$
  end
end

Algorithm 2: Procedure to compute forces and torques among grains and between grains and objects.

2.5.2 Computational Cost

Components of the solver that contribute to its cost:

1. RKF45 integration of the equations of motion.
2. Detection of grain collisions.
3. Detection of grain-object collisions.
4. Force calculation during grain collisions.
5. Force calculation during grain-object collisions.
6. Maintaining contact displacement quantities.

The integration of the equations of motion is an important factor in determining the overall simulation time. The dominant cost of the solver at each timestep is the computation of the instantaneous forces and torques active on the grains. Efficient calculation and maintenance of granular quantities, in particular contact detection and contact displacement, affects the overall efficiency of the solver. The stiction condition expressed in Section 2.3.5 is in practice implemented such that it adds no significant computational expense.

RKF45 Integrator

The computational time of a simulation grows linearly with the number of timesteps required. The explicit RKF45 method is best suited for non-stiff problems. For very stiff problems, the method must use very small timesteps to ensure stability of the integration and the overall simulation can become inefficient. Multi-body problems that experience large impulse forces are known to be stiff; however, many granular systems are not so stiff such that an explicit method is not able to solve the dynamics adequately. In soft-sphere discrete element models, the magnitude of impulse forces is directly related to the mutual deformation, ξ_{ij}. Experimentally, if the mutual deformation is small, the magnitudes of impulses are not likely to cause stiffness in the ODE system and simulation times are reasonable. The magnitude of the deformation in a system may be controlled by adjusting the elastic repulsion, k_α, in the normal forces, $\vec{F}_n$.

Collision Detection

Efficient contact detection affects the scalability of the solver. In line 2 of Algorithm 2, a spatial hash grid is created based on the positions of the spheres on all grains to provide a fast look-up of contacting spheres. Spatial hashing refers to a hashing technique where spatial positions are mapped (via a hash function) to grid coordinates and the grain location is stored in a hash table. To identify collisions, only the grains in nearby grid voxels adjacent to the current grain need to be computed. Instancing the hash grid is an $O(NJ)$ computation where N is the number of grains and J is the number of spheres on each grain. The hash grid is formed at each stage before the contact forces are determined [2]; this prevents the force computation from becoming an $O(N^2)$ algorithm in the average case.

[2]To improve computation time, the hash structure should be kept and updated as the grains move between voxels. For slow moving collections, the contacts change infrequently and the cost to update the hash structure would be negligible.

Force Calculation

The forces and torques active at any contact point are customarily determined from instantaneous particle parameters (i.e. $\vec{x}, \vec{v}, \vec{o}, \vec{\omega}$) except when the force model uses the contact displacement quantity, $\vec{\delta}_t$. For force models that only use the instantaneous parameters such as the basic model (Section 2.3.2), the computational and storage costs are insignificant. Force models that make use of the contact displacement quantity require additional storage and access to its values.

Contact Displacement

Maintaining the contact displacement vector, $\vec{\delta}_t$, requires an interaction over all spheres that comprise the grains in the system. For each grain, the identity of the contacting grains, the interacting spheres and the cumulated displacement vector is placed in a list. During an update, this list of contacting grains is traversed and maintained. For composite-sphere grains, the average number of contacts, C, is typically a small number, $C << N$. No special search structure is used to maintain the contact displacement list and the computation on line 20 of Algorithm 1 may be considered an $O(N)$ operation.

Stiction Condition

In practice, the stiction condition is applied by definition. If a particle is experiencing stiction, then the particle resists applied tangential forces as well as experiences an impulse that works to decelerate the tangential velocity. If $\vec{F}$ is the sum of all forces on the grain, $\vec{V}$ is the relative velocity of the grain with respect to the object and $\vec{n}$ is the normal direction of the object, then applying stiction means $\vec{F}$ is redefined to be

$$\vec{F} = \vec{F}^n + \vec{F}^t$$

such that

$$\begin{aligned} \vec{F}^n &= (\vec{F} \cdot \vec{n})\vec{n} \\ \vec{F}^t &= -k_s \frac{\vec{V} - (\vec{V} \cdot \vec{n})\vec{n}}{t_q} \end{aligned}$$

where k_s is a constant and t_q is a small time interval. This implementation of stiction avoids the need to sum over all contacts as in Equation 7.

Similarly for torques, by definition, stiction implies the particle resists applied torques in the normal direction and experiences a resistive rolling friction that decelerates any instantaneous angular velocity. The expression on line 26 of Algorithm 2 describes this condition.

2.6 Simulation Parameters

The model and material parameters necessary for grain-grain collisions are identified in Sections 2.3.2, 2.3.3 and 2.3.4. Their descriptions and typical values used in our simulations are prescribed by:

Parameter	Relation/Sample Values	Default Value	Description
ν	$0, 0.1, ...$	0.3	Poisson ratio of the material
E_j, E_k	$...10^7, 10^8, ...$	10^8	Young's modulus
E_{jk}	$\frac{1}{E_{jk}} = \frac{1}{E_j} + \frac{1}{E_k}$		effective Young's modulus
R_{jk}	$\frac{1}{R_{jk}} = \frac{1}{R_j} + \frac{1}{R_k}$		effective radius
α	$..., \frac{3}{2}, ...$	$\frac{3}{2}$	elastic repulsion response
k_α	$\frac{4}{3}\frac{E_{jk}}{2(1-\nu^2)}\sqrt{R_{jk}}$		elastic repulsion strength
β	$..., \frac{1}{2}, ...$	$\frac{1}{2}$	viscous dampening response
k_β	$..., 50, 100, 200, ...$	50	viscous dampening strength
μ_s	$0, ..., 1$	0	coefficient of static friction
k_t	$0, ..., 1$	0	static friction strength
μ_r	$0, ..., 0.001, ...$	0	coefficient of rolling friction
γ	$..., \frac{1}{2}, ...$	$\frac{1}{2}$	static friction response
d_t	$k_t\frac{\gamma\mu_s\lvert\vec{F}^n_{jk}\rvert}{\delta_{\max}}(1-(\frac{1-\min(\lvert\vec{\delta}_{jk}\rvert,\delta_{\max})}{\delta_{\max}})^{\gamma-1})$		tangential dampening strength
$\delta_{\max}$	$\mu_s\frac{2-\nu}{2(1-\nu)}\xi_{jk}$		maximum static friction

Similarily, the description and typical values of the model and material parameters necessary for grain-object collisions identified in Section 2.3.5 are:

Parameter	Relation/Sample Values	Default Value	Description
E	$...10^7, 10^8, ...$	$2 \cdot 10^8$	Young's modulus
a	$..., \frac{3}{2}, ...$	$\frac{3}{2}$	elastic repulsion response
k_a	$\frac{4}{3}\frac{E}{(1-\nu^2)}\sqrt{R_{jq}}$		elastic repulsion strength
b	$..., \frac{1}{2}, ...$	$\frac{1}{2}$	viscous dampening response
k_b	$..., 50, 100, 200, ...$	50	viscous dampening strength
μ_q	$0, ..., 1$	1	coefficient of static friction
k_q	$0, ..., 1$	1	static friction strength
μ_τ	$0, ..., 0.001, ...$	0.0005	rolling friction strength
k_s	$0, 1, 2, ...$	2.1	stiction strength
k_τ	$0, ..., 0.001, ...$	0.1	rolling stiction strength
t_q	$0, 0.001...$	0.001	static friction duration

Unless otherwise specified, the model and material parameter values used to simulate the results presented in Sections 3 and 4 are assigned the default values as specified in the above tables.

The difficulty in granular simulations, in particular simulations of granular statics, is that the formation of the final steady-state may be achieved in an indefinite number of ways. The resulting dynamics is highly dependent on the materials past history, and as a consequence, simulations are sensitive to its input parameters.

Arbitrary parameter values may lead to spurious results. The complex relationship between the simulation parameters and the dynamics may incur additional numerical difficulties on the solver.

Researchers in the field have a practice of calibrating their computations to achieve the desired results. DEM simulations commonly use a material calibration procedure where material parameters such as $E, \nu, \mu_s, ...$ are tuned to match model predictions with experimental observations. In static equilibrium computations, algorithmic calibration, where time intervals, dampening values and loading rates are adjusted, is typically performed to achieve static-states under general conditions. In some situations, simulation parameters may be assigned values without physical or mathematical justification but compelled only by a preferred response. In recent work, Tu et al. [12] introduces a procedure to set suitable parameters values for steady-state granular simulations. Essentially, the parameters are set by trial and error until some criteria is satisfied for a small subset of the entire particulate system. Though effective, the procedure only applies to steady-state simulations and does not extend to a wide range of system dynamics. It is not yet clear how to define a similar framework for our granular solver.

3 Simulations

Granular material may be distinguished by their ability to pack, pile, jam, fracture, fragment and avalanche. These complex behaviors are nonlinear in nature and the necessary conditions that lead to the emergence of these phenomena are ill-defined. Nonetheless, a granular solver may be evaluated by its ability to simulate these characteristic properties. A suitable solver is ideally able to demonstrate this set of behaviors in a controllable fashion.

3.1 Comparison of Force Models

Grain piles formed by grains flowing through a hopper is a basic example of granular behavior. This situation is used to evaluate the merits of the three force models considered in Sections 2.3.2, 2.3.3 and 2.3.4. Different from previous studies, this system is not enclosed with bounding walls. Apart from a floor, the grains do not interact with any other object that may interrupt grain motion and induce piling. The floor is also not a geometrically rough base comprised of discrete particles but is treated as a smooth object that interacts with the granular system through contact forces (Section 2.3.5). This environment necessitates the need for an effective static friction model by reducing the influence of geometric friction and physical barriers on pile formation.

Figure 5 compares the ability of the three grain-grain force models to form piles. All three models begin with identical initial conditions, the same parameters values (where applicable) and use the same model for grain-object collisions. The first (or bottom) layer of grains in a grain pile is formed by the grain-object contact model. Subsequent grain behavior on top of this layer is mostly due to the grain-grain contact model. To judge the ability of each grain-grain force model to form piles, the first layer of grains should not be considered.

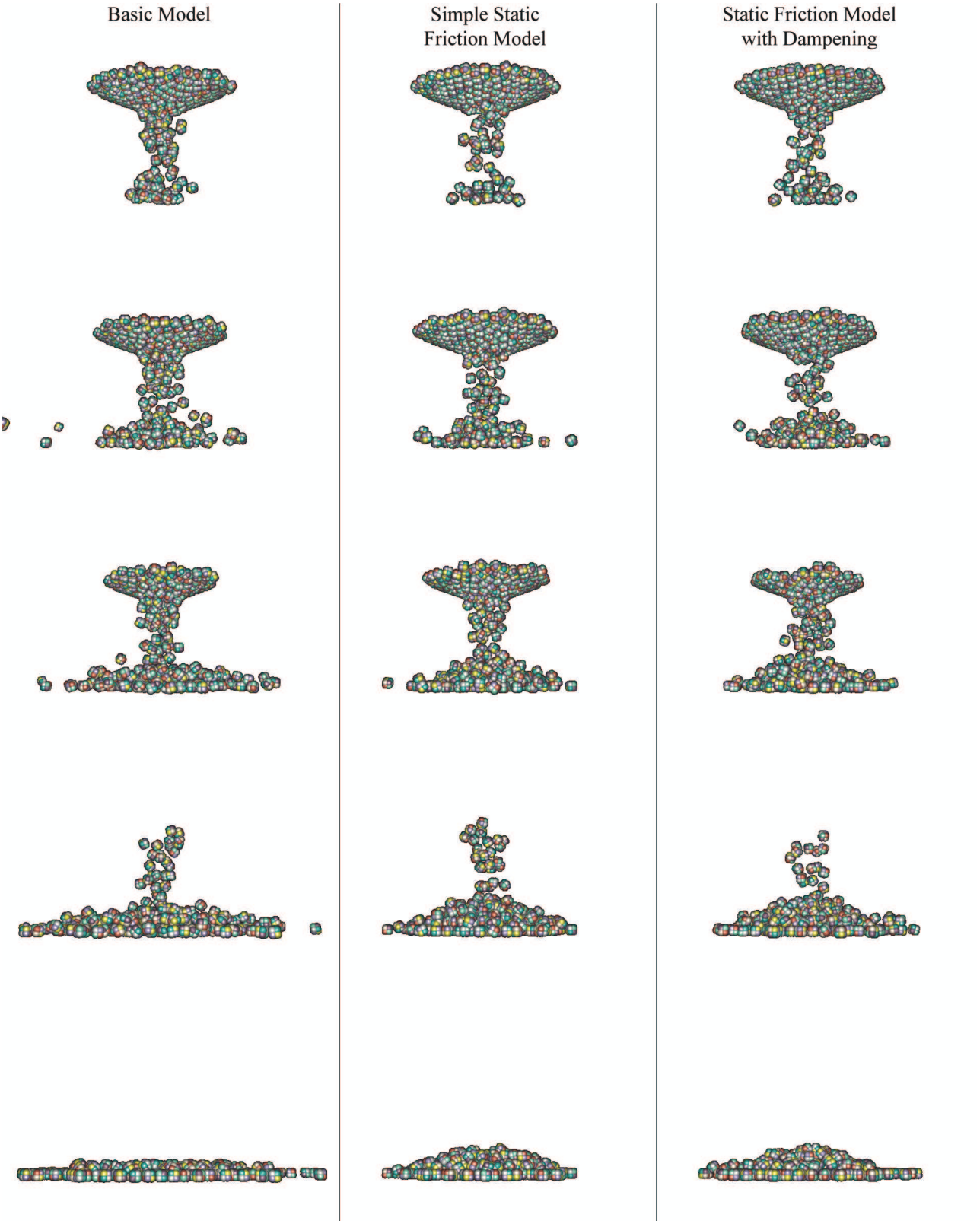

Figure 5: Five snapshots at similar stages of hopper flow with 397 grains under three different force models: Section 2.3.2: Basic Model, Section 2.3.3: Simple Static Friction Model and Section 2.3.4: Static Friction Model with Dampening. The hopper is not drawn to expose the grain dynamics.

All three simulations in Figure 5 use grain-grain friction parameters $\mu_s = 0.5$ and $\mu_r = 0.00025$. The basic force model, on the farthest left, shows little evidence of granular piling. This is expected since this model does not explicitly model the static friction necessary for piling. In fact, any pile formed using this basic model will always collapse under its own weight. When the grains are still in motion, the behavior of the grains looks reasonable. However, as the system begins to settle into a static state, the grain pile collapse occurs in a smooth continuous manner. This makes the slow-moving behavior of the grains look liquid-like. This basic model is not suitable for situations where the discontinuous slip-stick motion of the grains in the material needs to be more apparent.

In the second model in Figure 5, in the middle, there is evidence of stable static piles. As the pile forms, grains that come in contact with the surface of the pile roll down the slope of the heap. Grains on the interior of the pile experience much less motion. This demonstrates that static friction, modeled by the cumulated contact displacement, $\delta_{j,k}$ (Equation 4), is necessary for piling to occur. However, although this model can simulate piling, the model is rather sensitive to changes in initial conditions. It is also difficult to know how to set the parameters and initial conditions to achieve a desired result. This simple static friction model is not robust enough to be a general-purpose granular solver.

The final model considered in Figure 5, on the right, also shows evidence of stable static piles. The discontinuous slip-stick motion is even more pronounced in this model than in the second model. Detailed expressions for the friction and dampening parameters make this model easier to manipulate and control the final result. This static friction model with dampening is the force model selected for our solver. The simulations in the rest of this work use this contact force model.[3]

3.2 Hopper Flow: Piling and Jamming

Grains in a hopper is an interesting situation because varying the frictional parameters not only can alter the steady-state pile dimensions but may also introduce jamming of the grains in the hopper. Varying the inter-grain friction parameters, μ_s and μ_r, varies the rate that the pile forms as well as the final pile dimensions. Figure 6 demonstates this behavior for a hopper initially filled with 1575 cube-like grains.

Increasing the grain-grain and grain-object friction parameters, μ_s, μ_r, μ_q and μ_τ, can cause jamming of the granular system. For the granular system in Figure 6, if $\mu_s = \mu_q = 1.0$ and $\mu_r = \mu_\tau = 0.0005$ where q and τ refer to the hopper parameters, then the system jams for over 50 seconds of a 2 minute simulation. Jamming under the influence of static friction at grain-object boundaries agrees with current proposed theories [13].

3.3 Granular Collapse

Under typical conditions, a granular collection naturally packs and piles as the system tends to a quasi-stable state. When conditions occur that disrupt the quasi-stable system, the granular collection ostensibly fractures and avalanches. In such systems, there is a clear boundary between actively moving grains and stable static

[3] See Appendix A for another comparison of simulations from the first and third force model.

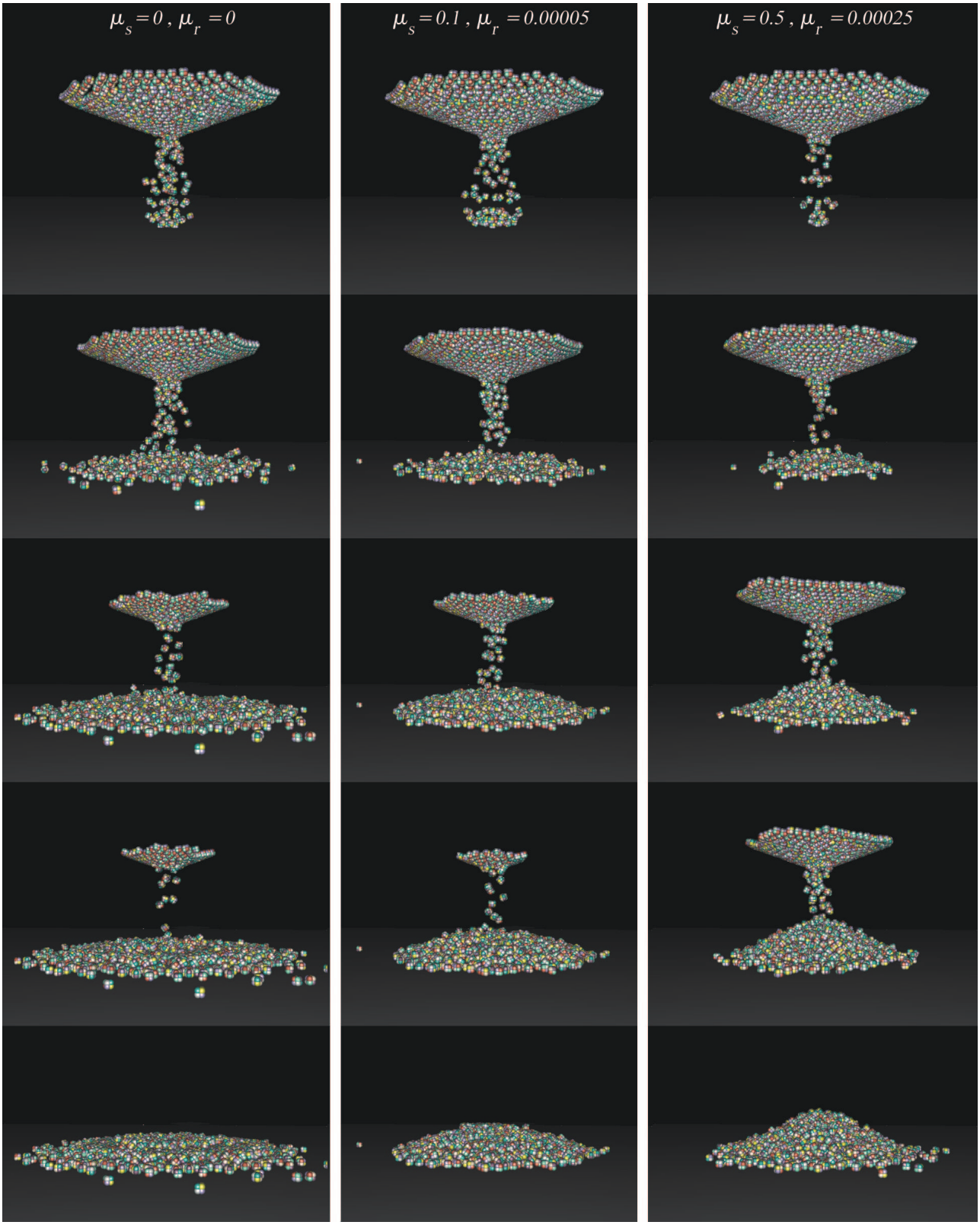

Figure 6: Five snapshots of hopper flow with 1575 grains for three pairs of values of (μ_s, μ_r): $(0.0, 0.0), (0.1, 0.00005)$ and $(0.5, 0.00025)$. The parameters μ_q and μ_τ are identical for all three examples. The hopper is not drawn to expose the grain dynamics.

grains. A simple demonstration shows that this set of behaviors is natural to this solver.

In Figure 7, a set of grains is allowed to pack in a space with a resistant floor and five frictionless bounding walls. A quasi-stable state is reached by frame 300 and remains stable for the next 300 frames (top row). At frame 600, the slanted wall is removed and surface avalanching occurs (second row). The grains underlying the active surface remain stable. Stable grains near the active boundary form a roughened surface to resist continual avalanching (third row). By frame 1000, avalanching ceases and two new stable grain collections are formed. At frame 1300, the right wall is gradually moved to the right, away from the pile (fourth row). The large granular system begins to collapse. A clear fracture surface is observed where grains remain static on one side and grains have significant motion on the other side. As the surface of the heap changes, avalanching of surface grains occurs (fifth row). The evolution of the dynamics of this system illustrates many characteristics unique to grains.

3.4 Dynamic Range

Grains exhibit a range of motion that may be categorized as solid-like, liquid-like or gas-like. The model and material parameters in this solver only serve to define the character of the grains at the particulate scale; the motion of the grains is contingent on its initial values and environmental conditions. The transition of grains from solid to liquid regimes (or solid to gas, liquid to gas or vice-versa) is abrupt in nature and it is this abruptness that is fundamental to realistic simulations.

A progression of grain motion from slow to swift to static is an example of how the environment can foment sudden changes in a collection's dynamics. In Figure 8, a set of grains is placed in a rectangular container with low coefficient of friction and only three sides (top row). Subject to gravity, this set of grains begins to settle which causes some grains to gradually fall off the unenclosed side of the container (second row). At frame 200, the container abruptly rotates and large impact forces are imparted onto the grains (third row). The granular collection responds and displays a liquid-like motion where a wave-like splashing action is observed. As the grains collide with the resistant floor, the grains quickly become inert (fourth row). This rapid dissipation of kinetic energy allows inert grains to begin to form static structures that can support solid-like configuration of grains. The final arrangement of the grains on the floor show a stable non-uniform terrain that is consistent with the simulation (fifth row). The range of dynamics attainable by a set of grains is separate from the particulate model of the grains. The initial conditions and the environment determine the type of granular motion. The independence of the solver from any assumption of the motion permits general-purpose use of this solver.

4 Analysis

Grain piles formed from grains flowing through a hopper is the setup used to verify the suitability and robustness of this solver. Piling from hopper flow is studied because it provides a simple way to control the conditions that affect pile formation. Analysis of the timesteps taken in a simulation indicates the appropriateness of the numerical scheme and the overall applicability of the solver.

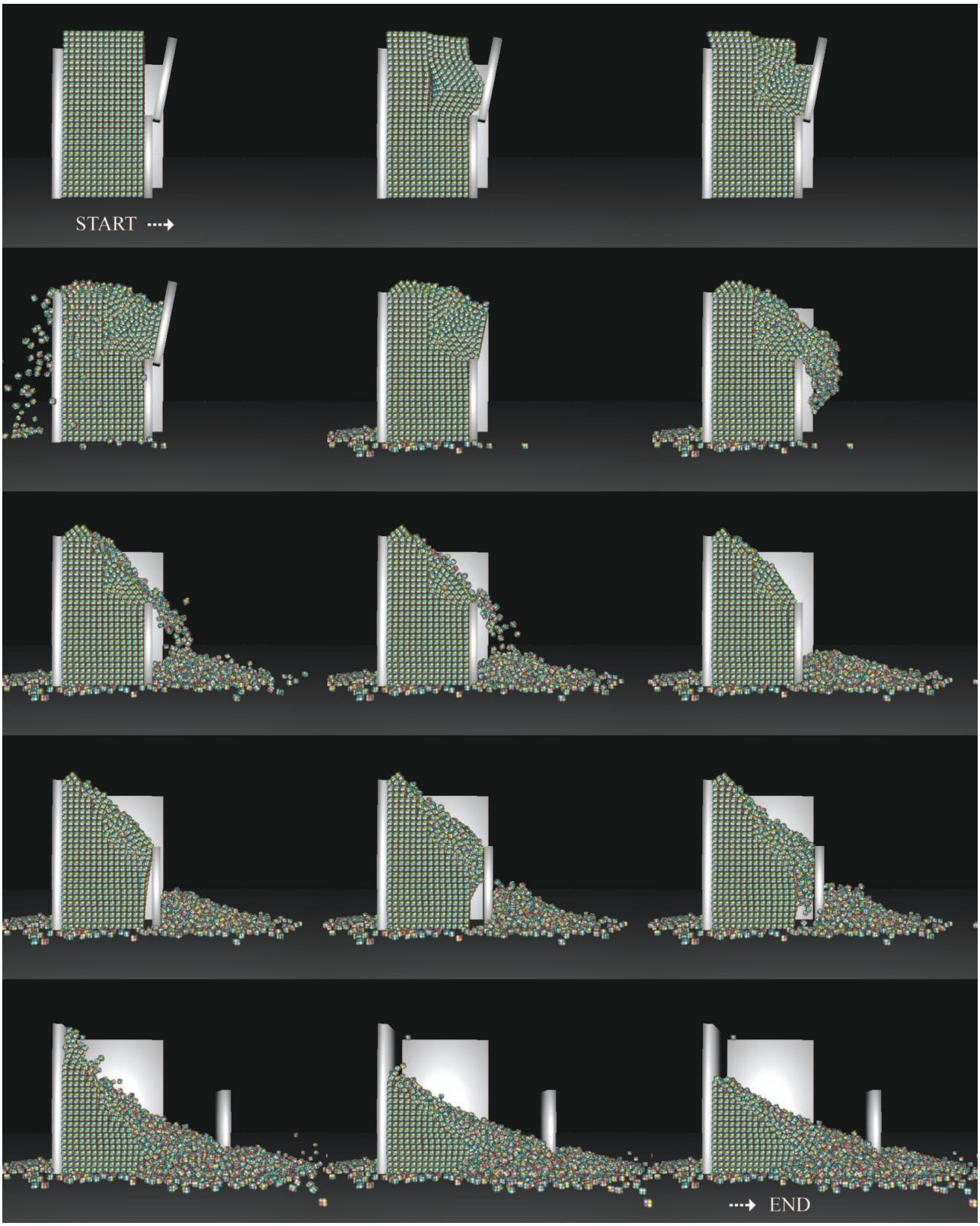

Figure 7: Snapshots of packing and fracturing of 5157 grains during granular collapse, for $\mu_s = 0.5, \mu_r = 0.00025$. The wall facing the viewer is not drawn to expose the grain dynamics.

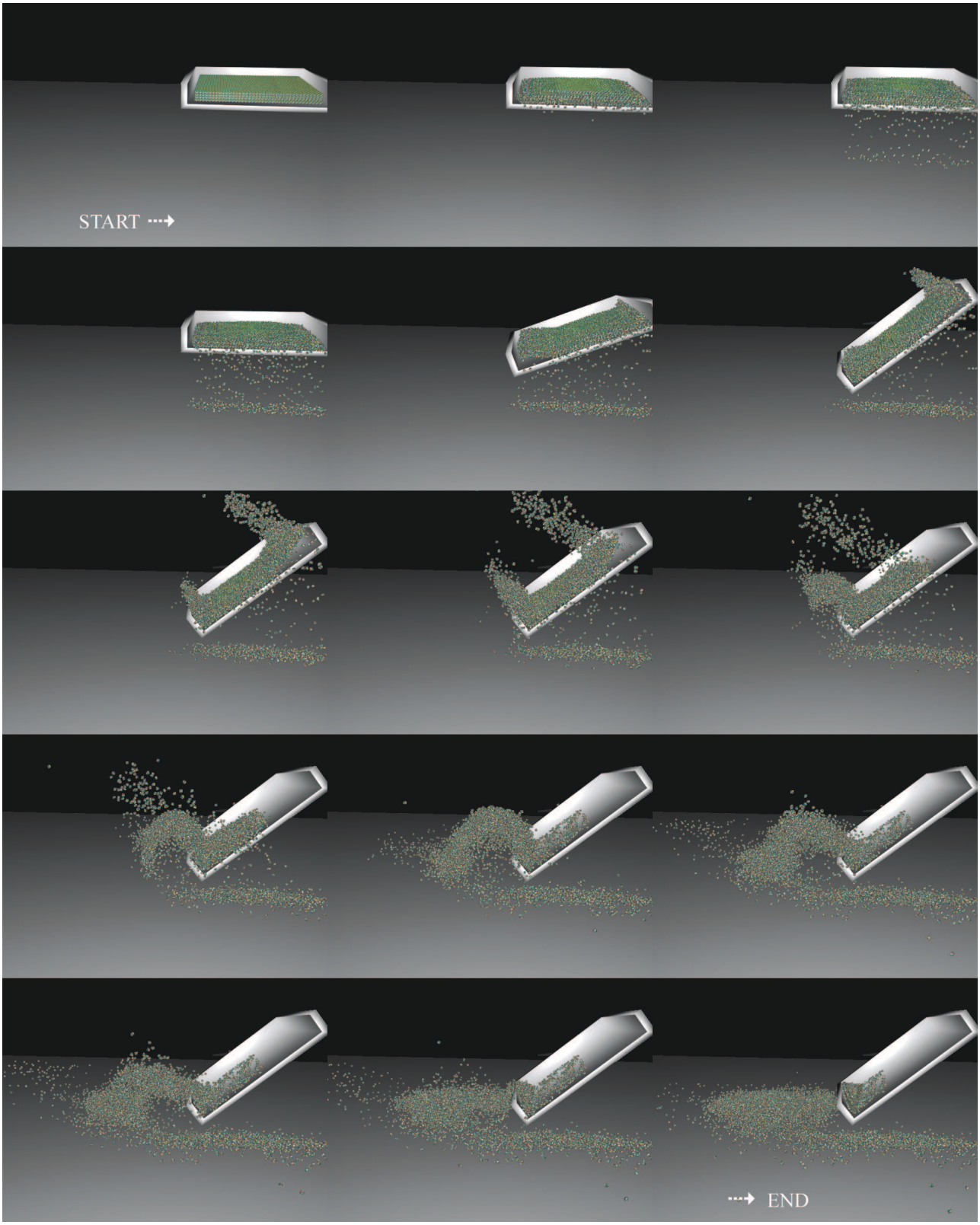

Figure 8: Snapshots of a range of granular dynamics with 6889 grains, for $\mu_s = 0.5$ and $\mu_r = 0.00025$.

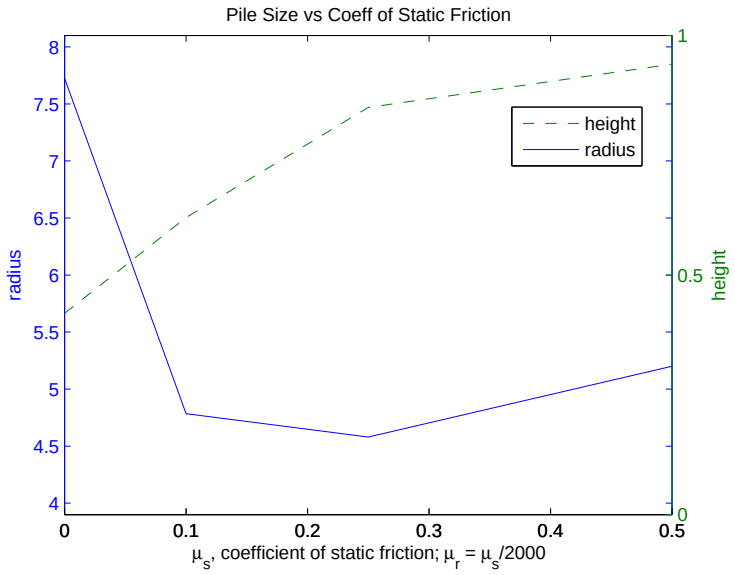

Figure 9: Pile size vs Friction Parameters

4.1 Friction Parameters and Pile Formation

Varying the friction parameters in a simulation controls the pile height. There are two sets of friction parameters to consider: the inter-grain friction parameters, μ_s and μ_r, and the grain-object friction parameters, μ_q, k_q, μ_τ, k_s, k_τ and t_q. The grain-object friction parameters, μ_q, k_q, and μ_τ, only serve as an estimate of the instantaneous force from grain-object collisions; the total force from all collisions is then evaluated to determine if stiction conditions apply. The stiction parameters, k_s, k_τ and t_q, determine how quickly a grain is brought to rest and remains static, such that piling may begin forming on top of it. The grain-object friction parameters are identical for all simulations in this paper.

Figure 9 shows how the inter-grain friction parameters affect piling for a simple hopper simulation of 1575 cube-like grains. For small values of μ_s and μ_r, the pile is short and wide; for large values of μ_s and μ_r, the pile is tall and thin. Values of μ_s from 0 to 0.25 affect the pile size considerably; values of μ_s from 0.25 to 0.5 affect the pile size less significantly. For this simulation, small values of μ_s indicate that the inter-grain friction forces are the dominant mechanism in pile formation. For larger values of μ_s, less sensitivity to μ_s suggests the pile size is dominated by the geometric configuration of the constituent grains.

The effects of changing μ_s and μ_r on the stiffness of the equations of motion may be observed by looking at the timesteps of the numerical solution. Table 1 summarizes some timesteps statistics taken for various μ_s and μ_r. For the frictionless case, the numerical approximation requires marginally smaller timesteps than simulations with friction. This seems to suggest that the presense of inter-grain friction does not significantly affect the stiffness of the system of differential equations. This is further explored in Section 4.2.

		No. of time steps		Time stepsize		
μ_s	μ_r	Accepted	Rejected	Max	Min	Avg.
0	0	9611	763	0.041667	0.002083	0.011367
0.1	0.0001	9291	1310	0.041667	0.003472	0.013449
0.25	0.000125	8287	762	0.041667	0.003086	0.013198
0.5	0.00025	9083	488	0.041667	0.004630	0.012771

Table 1: Timestep statistics for several values of μ_s and μ_r, $TOL = 0.01$

4.2 Friction Parameters and Stiffness

The stiffness of the equations of motion can often be explicitly quantified by the eigenvalue of the Jacobian with the largest negative real part. Since the ODE system has $3 \cdot 4 \cdot N$ number of equations, the Jacobian is only practically formed for a small number of grains. For this analysis, 48 cube-like grains in a pyramid configuration are suspended a small distance from a surface and allowed to fall and collide with the surface. Grain-object friction is present to bring the granular system to rest. The inter-particle friction is varied. A local Jacobian may be estimated by finite differences in a small time interval. A set of these Jacobians sampled at a number of discrete times gives an indication of the progression of the eigenvalues of the Jacobian of the system over time.

Table 2 lists estimates of the eigenvalue of the Jacobian with the maximum negative real part for various grain-grain friction parameters at 20 discrete times. These values may be used to understand the general trend of a system's stiffness as the granular system evolves. As these values are estimates, and as the dynamics of the system are different for different pairs of (μ_s, μ_r), the eigenvalue estimate for a pair of (μ_s, μ_r) at some time t_n does not necessarily have any relation to the eigenvalue estimate of another pair of (μ_s, μ_r) at the same time.

Eigenvalues with a large negative real part and a small imaginary part indicate a stiff ODE system. On occasion, such as timesteps $t_7, (\mu_s, \mu_r) = (0.25, 0.000125)$ and $t_3, (\mu_s, \mu_r) = (0.5, 0.00025)$, the eigenvalues are larger; however, in general, the eigenvalues in Table 2 are not exceptionally large. This suggests that, for this example, the ODE system is, in general, not stiff and the timesteps taken by the integration formula are not restricted by the stability of the numerical method. Although the many-body problem is potentially stiff, for the dynamics studied in this example, the problem is not so stiff such that the stability of the explicit method limits the time stepsizes in the numerical integration.

Table 3 lists estimates of the eigenvalue of the Jacobian with the maximum absolute real part at the same 20 discrete times as in Table 2. Large positive eigenvalues in this table indicate the system is predominately unstable. Large negative eigenvalues, again, indicate stiffness. This table may be used to understand how the largest eigenvalues evolve as the simulation progresses.

(μ_s, μ_r)	$(0, 0)$	$(0.25, 0.000125)$	$(0.5, 0.00025)$	$(0.75, 0.000325)$	$(1.0, 0.0005)$
t_1	-1776.7	-2991.8	-2781.2	$-1053.2 + 929.3i$	-2532.4
t_2	-501.7	-548.9	-952.6	-988.3	-285.9
t_3	-8397.8	$-140.3 + 138.5i$	-78298.3	$-362.7 + 1161.5i$	-169.5
t_4	-311.2	-3421.8	$-1170.1 + 1417.4i$	-417.2	-1048.7
t_5	-226.4	-186.2	-1386.9	-389.4	-9445.1
t_6	-962.4	$-1102.7 + 2724.8i$	-8467	-2420.5	-110.4
t_7	-172	-16714.1	-519.6	-2442.2	-425.4
t_8	-64	-100.9	-441.2	-1440.4	-7619.1
t_9	-3202.5	-226.9	-1026.2	-9171.2	-148.5
t_{10}	-851.5	-300.5	-1431.6	-49.1	-369.6
t_{11}	-421.5	-160.8	-2341.9	-369.4	-232.4
t_{12}	-326.1	-1566.1	-260.3	-158.5	-297.2
t_{13}	-2212.3	-1281	-184	-1534.2	-2706.3
t_{14}	-500.8	-516.3	-9311.4	-2207.1	-552.2
t_{15}	-2147	-12035.3	-587.7	-405.6	$-768.3 + 153.8i$
t_{16}	-10384	-1888.8	-4363.5	-163.3	-3211.6
t_{17}	-5.5	-242.8	-164.2	-132.4	-280.8
t_{18}	-268	-2192.9	-154.5	-6.5	$-478.6 + 270.8i$
t_{19}	-2632.2	-225.7	-683.8	-298.4	-655
t_{20}	-10	-4932.3	-709.1	-163.2	-215.1

Table 2: Estimates of the eigenvalue of the Jacobian with largest negative real part at 20 discrete times during the simulation. $TOL = 0.01$

(μ_s, μ_r)	$(0, 0)$	$(0.25, 0.000125)$	$(0.5, 0.00025)$	$(0.75, 0.000325)$	$(1.0, 0.0005)$
t_1	9758.9	13633.9	14887.6	10523.3	13282.2
t_2	2392.6	3708.5	4832.5	7701.4	1531
t_3	-8397.8	1221.6	-78298.3	2057	3142.1
t_4	3057	-3421.8	12422.2	3277.3	6314.5
t_5	3277.1	330	-1386.9	631.6	-9445.1
t_6	-962.4	6411	-8467	-2420.5	-110.4
t_7	-172	-16714.1	-519.6	-2442.2	-425.4
t_8	467.2	885.1	-441.2	-1440.4	-7619.1
t_9	-3202.5	625.5	-1026.2	-9171.2	-148.5
t_{10}	-851.5	1579.2	-1431.6	302.8	-369.6
t_{11}	6202.4	808.5	-2341.9	-369.4	-232.4
t_{12}	1493.8	-1566.1	1436.7	-158.5	947.5
t_{13}	-2212.3	-1281	338.5	-1534.2	-2706.3
t_{14}	-500.8	-516.3	-9311.4	15995.3	-552.2
t_{15}	-2147	-12035.3	-587.7	-405.6	6151.6
t_{16}	-10384	11271.1	-4363.5	564.3	-3211.6
t_{17}	412.6	-242.8	371.2	-132.4	1971.8
t_{18}	7504.4	12357.7	2939.2	28.4	$-478.6 + 270.8i$
t_{19}	-2632.2	-225.7	-683.8	-298.4	1421.9
t_{20}	94.8	6711.4	-709.1	1362.5	1241.8

Table 3: Estimates of the eigenvalue of the Jacobian with largest absolute real part at the same 20 discrete times during the simulation as in Table 2. $TOL = 0.01$

	No. of time steps		Time stepsize		
TOL	Accepted	Rejected	Max	Min	Avg.
0.1	1509	15	0.041667	0.005208	0.016540
0.05	1458	22	0.041667	0.006944	0.017118
0.01	1709	77	0.041667	0.004630	0.014604
0.005	2000	140	0.041667	0.002083	0.012479
0.001	3412	298	0.041667	0.001437	0.007315

Table 4: Timestep statistics for RKF45 using $\mu_s = 0.5$ and $\mu_r = 0.00025$.

In Table 3, large positive eigenvalues at t_1 indicates that each simulation begins very unstable. This time point corresponds to the initial collision of the grains with the surface, and becomes more stable as the effects of friction settle the grains. For simulations with small values of (μ_s, μ_r), the system fluctuates between stable and unstable states which coincides with recurrent slip-stick motion, common for grains with these parameters. At some time points, such as $t_3, (\mu_s, \mu_r) = (0.5, 0.00025)$ the eigenvalue is larger and negative. This indicates that the ODE system may occasionally be stiff during the simulation. Given these trends, the explicit RKF45 formula is adequately applied to this problem.

4.3 Step Size, Timesteps and Tolerance

The step size tolerance, ϵ, in the numerical scheme controls the step size in the adapted timesteps. For RKF45, the tolerance is essentially a limit on the maximum difference between the fourth and fifth order estimate of the grain positions, $\vec{c}_i$. Lowering the tolerance causes the solver to take smaller stepsizes and consequently increases the total number of timesteps taken. The number of rejected timesteps also increases with a more restrictive tolerance. Sensitivity of the time stepsizes to different tolerances is another indication that the integration method is not operating at the boundary of its stability region. As suggested in Section 4.2, the time stepsizes taken do not seem to be restricted by the stability requirements of the RKF45 method.

Figure 10 plots the stepsizes taken at each timestep for a range of tolerances. A simple hopper flow simulation is used with 397 particles where $\mu_s = 0.5$ and $\mu_r = 0.00025$. For each tolerance the final pile size is observed to be the same.

Table 4 summarizes some statistics of the timesteps. Indeed, as the tolerances become more restrictive, the number of accepted and rejected timesteps increases while the smallest timestep taken decreases, as expected. This can lead to lengthy simulation times and tolerances lower than 0.001 are not considered for RKF45.

In Algorithm 1, the suggested adapted timestep, h_{opt}, is actually constrained such that there is an integer number of timesteps within one time frame. Any timestep that is less than or equal to h_{opt} is a valid timestep to take, as far as satisfying the tolerance is concerned. For simulations in this paper, one time frame is $1/24 = 0.041667s$. Any timestep taken divides evenly into 0.041667.

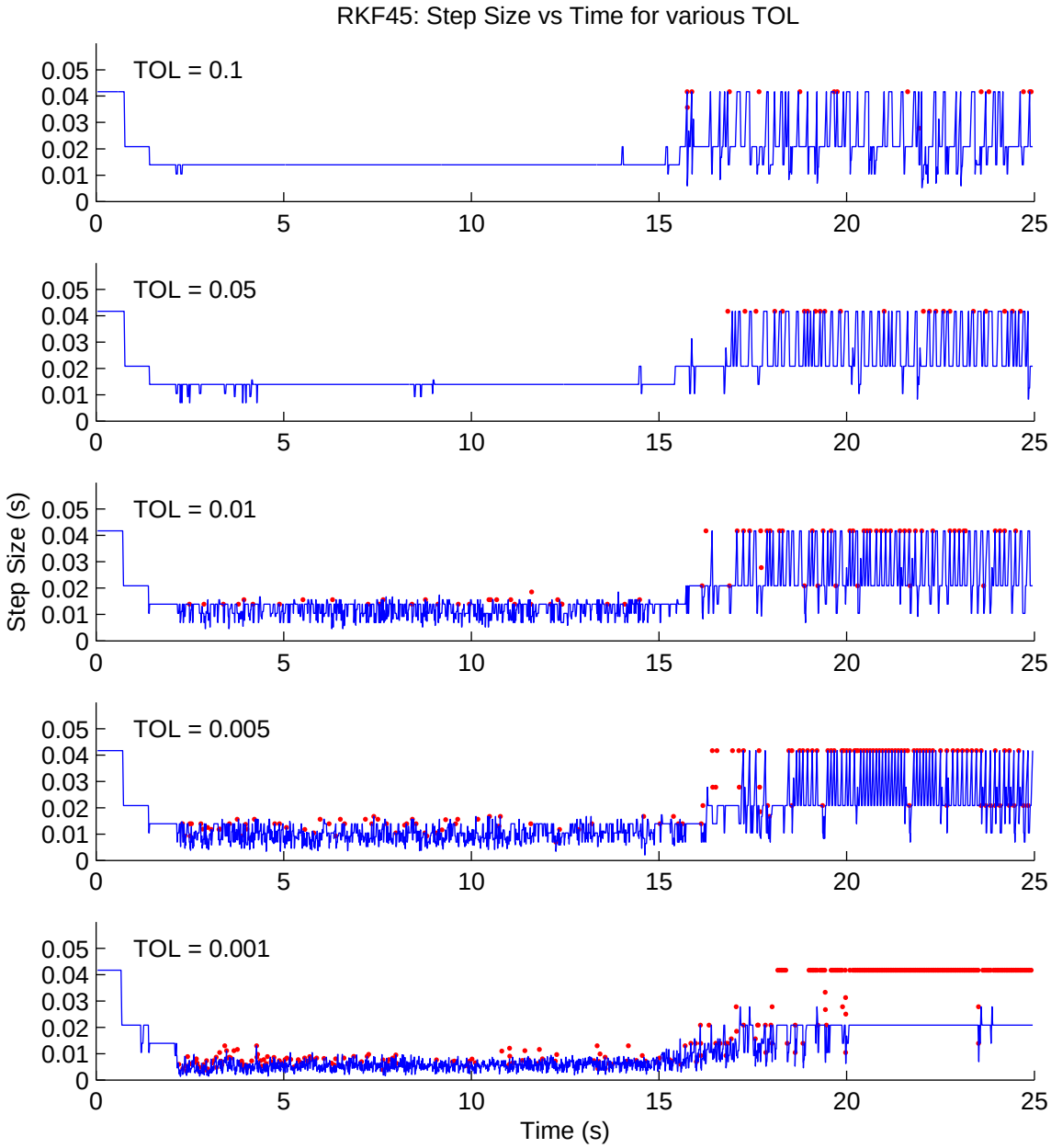

Figure 10: RKF45 stepsizes versus time for various tolerances. Accepted (rejected) steps are indicated in blue (red).

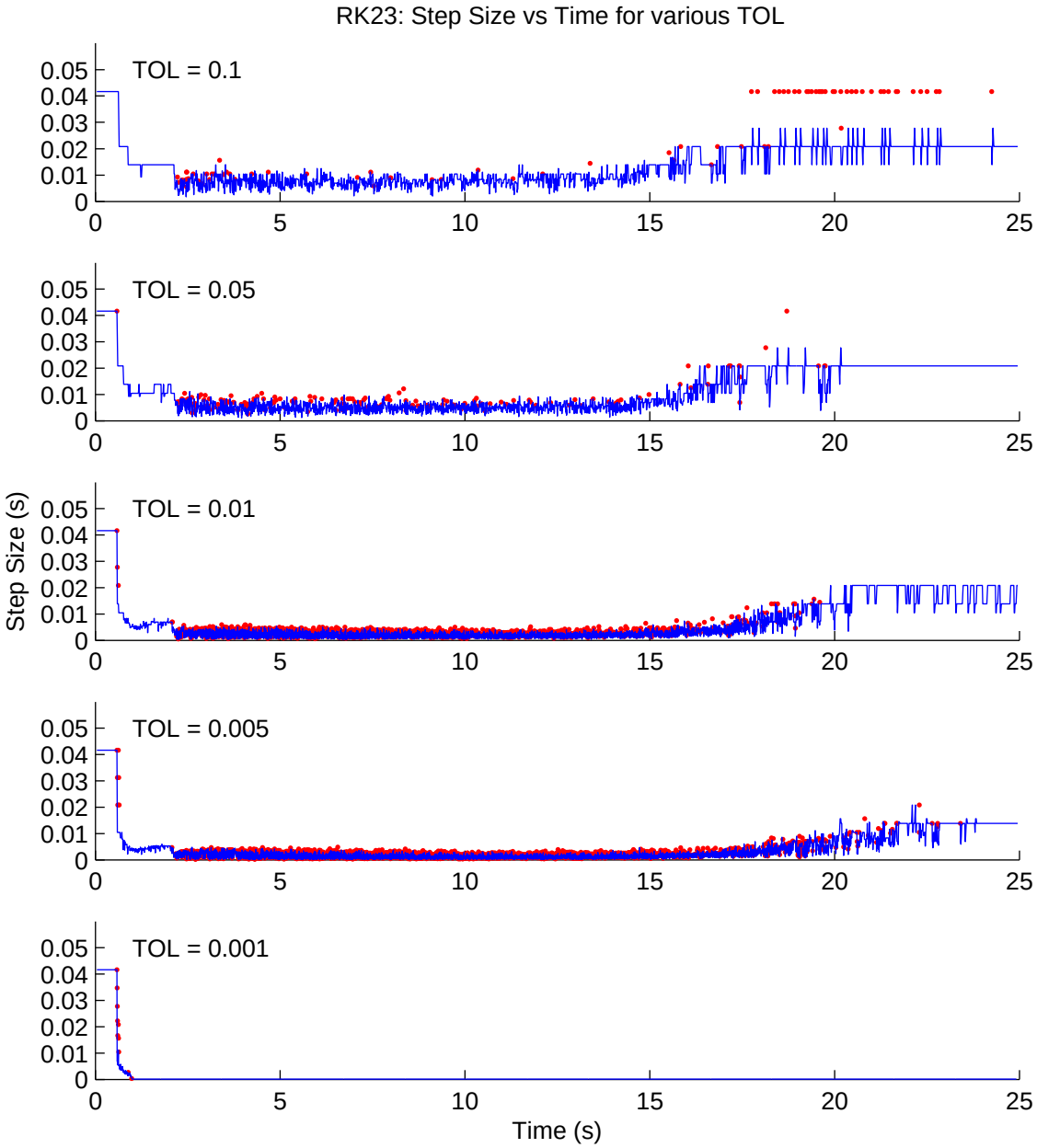

Figure 11: RK23 (Bogacki-Shampine) stepsizes versus time for various tolerances. Accepted (rejected) steps are indicated in blue (red).

	No. of time steps		Time stepsize		
TOL	Accepted	Rejected	Max	Min	Avg.
0.1	2445	81	0.041667	0.001736	0.010208
0.05	3537	164	0.041667	0.001042	0.007056
0.01	9493	1660	0.041667	0.000284	0.002629
0.005	15706	3008	0.041667	0.000141	0.001589
0.001	169908	10	0.041667	0.000141	0.000147

Table 5: Timestep statistics for RK23 using $\mu_s = 0.5$ and $\mu_r = 0.00025$.

In Figure 11, the same analysis is performed for the lower order Runge-Kutta (Bogacki-Shampine) 23 (RK23) method. Table 5 summarizes some timestep statistics. As expected, the lower the tolerances, the more timesteps are taken as well as the step sizes are smaller. There is also a significant increase in the number of accepted and rejected timesteps compared to RKF45. For a tolerance of 0.001, the step size of most of the accepted timesteps is the minimum step size of 0.000141. This suggests the actual step size required to satisfy a tolerance of 0.001 is smaller than 0.000141 and the adaptive component of the numerical scheme is no longer applicable. This leads to lengthy simulations times and tolerances of 0.001 or lower. For this hopper simulation, using RK23 should not be considered without lowering the minimum step size.

The higher-order RKF45 method takes fewer number of timesteps, larger step sizes and more tolerances compared to the RK23 method. This indicates that granular simulations that exhibit piling benefit from the increased accuracy of the higher-order method. Since the numerical scheme is not the dominant computational cost in terms of flops, RKF45 with adaptive timestepping is preferred to the RK23 method, even for the larger tolerances.

4.4 Collection Size and Computational Time

The actual performance of this solver is problem dependent, since the dynamics of the system are also problem dependent. Nonetheless, as a simple demonstration of the $O(N)$ complexity of the solver, the computational time for a hopper flow simulation with piling for an increasing number of cubic-like grains comprised of eight spheres is presented in Table 6.

5 Conclusion and Future Work

The appeal of using discrete element methods for granular simulations is that the approach is straightforward and natural. Intuitively, the macroscopic behavior of a collection of grains originates from the micromechanics of the grains at the particulate scale. Successful application of this solver to model packing, piling, jamming, fracturing, fragmenting and avalanching demonstrates that, with suitable force models, the approach is effective and versatile.

Number of Grains	Number of Frames	Avg. Time/Frame (s)
536	300	2.4
1051	400	5.28
2071	600	13.2
4096	1500	28.8
8036	2500	58.2

Table 6: Computational time versus collection size using $\mu_s = 0.5$ and $\mu_r = 0.00025$.

For dry granular materials, the slip-stick motion of individual grains is the principal mechanism behind behaviors like packing, piling and fractures. Force models that have history-dependent terms are essential for proper modeling of the static friction necessary for the grains to result in avalanching or piling. The force model in Section 2.3.4 is the best suited model considered for this solver.

The performance of this solver is problem-dependent. When designing simulations, attention to the effects of the model, material and object parameters on the stiffness of the equations of motion is critical for reasonable computational times. Future work to ease the design of simulations and improve the computational expense is expected to promote this solver further.

There are three natural extensions to this work. In the current framework, it is trivial to introduce secondary forces to the equations of motion to describe alternative granular behavior. Cohesive forces or fluid-solid interaction terms may be simply added to the current model to simulate wet grains. The short-range influence of the forces in this chosen DEM method makes it easy to model wet and dry materials as well as a progression of one to the other. This is a valuable function of any general granular solver.

In conjunction with alternative force models, a robust numerical method is fundamental to be able to seamlessly handle a diversity of granular dynamics. Potential stiffness in the system of ODEs will hinder widespread application of this solver. Although the dynamics studied in this work are not particularly stiff, implicit or semi-implicit methods may mitigate the sensitivity of the numerical approximation on problem-specific stiffness. The additional complexity of any chosen implicit/semi-implicit method should not outweigh the gain in the solver's robustness.

For arbitrary granular systems of large N, this $O(N)$ solver is still computationally impractical. Steps to accelerate the computations are underway. Techniques include DEM-FEM multi-scale methods, coarse-graining and other hybrid approaches [15]. General techniques, such as parallel computation, only serve to improve the simulations times even more.

References

[1] Mohammad Saadatfar: *Computer Simulation of Granular Materials*. Computing in Science and Engineering, IEEE (11.1), p66-74. Jan/Feb 2009

[2] Bell N., Yu Y., Mucha P.J.: *Particle-Based Simulation of Granular Materials*. Proc. of ACM SIGGRAPH/Eurographics symposium on Computer animation, p77-86. 2005

[3] Poschel T., Schwager T.: *Computational Granular Dynamics*. Springer-Verlag, p13-241. 2005.

[4] Mehta A.: *Granular Physics*. Cambridge University Press, p1-24,132-146,233-297. 2007.

[5] Zhu Yongning, Bridson Robert: *Animating Sand as a Fluid*. Int'l Conf. on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2005 Papers, p965-972. 2005.

[6] Poschel Thorsten, Buchholtz Volkhard: *Static Friction Phenomena in Granula Materials: Coulomb Law verses Particle Geometry*. Phys. Rev. Lett. (71.24), p3963-3966. 1993.

[7] Elperin T., Goshtein E.: *Comparison of different models for tangential forces using the particle dynamics method*. Physica A (242), p332-340. 1997.

[8] Brilliantov N. V., Spahn F., Hertzsch J., Poschel T.: *Model for collisions in granular gases*. Phys. Rev. E (53.5), p5382-5392. May 1996.

[9] Zhou Y.C., Wright B.D., Yang R.Y., Xu, B.H., Yu A.B.: *Rolling Friction in the dynamic simulation of sandpile formation*. Physica A (269) p536-553. 1999.

[10] Baxter J., Tuzun U., Burnell J., Heyes D.M.: *Granular simulation soft two-dimensional heap formation*. Phys. Rev. E (55.3), p3246-3554. Mar 1997.

[11] Kruggel-Emden H., Sturm M., Wirtz S., Scherer V.: *Selection of an appropriate time integration scheme for the discrete element method (DEM)*. Comput. Chem. Eng (10.23), p2263-2279. 2008.

[12] Tu Xuxin, Andrade Jose E.: *Criteria for static equilibrium in particulate mechanics computations*. Int'l J. for Numerical Methods in Engineering (75.13), p1581-1606. 2008.

[13] Duran J.: *Sands, Powders, and Grains*. Springer, p3-140. 2000.

[14] O'Sullivan C., Bray JD. *Selecting a suitable time step for discrete element simulations that use tne central difference time integration scheme*. Engineering Computations (21), p278-303. 2004.

[15] Munzija A. *The Combined Finite-Discrete Element Method*. Wiley. 2004.

[16] Goldenberg C., Goldhirsch I. *Effects of Friction and Disorder on the Quasi-Static Response of Granular Solids to a Localized Force*. Phys. Rev. E (77.4), p041303-1 - 041303-19. 2007.

[17] Hinrichsen H., Wolf D. E. *The Physics of Granular Media*. Wiley, p.299-322. 2004.

[18] Kruggel-Emden H., Wirtz S., Scherer V.: *A study on tangential force laws applicable to the discrete element method (DEM) for materials with viscoelastic or plastic behavior*. Comput. Chem. Eng (63.6), p1523-1541. March 2008.

[19] Di Maio F. P., Di Renzo A. *Analytical solution for the problem of friction-elastic collisions of sphereical particles using the linear model*. Chem. Eng. Science (59.16), p3461-3475. 2004.

[20] Di Maio F. P., Di Renzo A. *Comparison of contact-force models for the simulatoin of collisions in DEM-based granular flow codes*. Chem. Eng. Science (59.3), p525-541. 2004.

[21] Brilliantov N. V., Albers N., Spahn F., Poschel T.: *Collision dynamics of granular particles with adhesion*. Phys. Rev. E (76.5), p051302-1 - 051302-12. 2007.

[22] Khan K. M., Bushnell G.: *Comment on "'Rolling friction in the dynamic simulation of sandpile formation"'*. Physica A (352). p522-524. 2005.

[23] Brilliantov N. V., Poschel T.: *Rolling friction of a viscous sphere on a hard plane*. EuroPhys. Letters (42.5), p511-516. 1998.

[24] Kruggel-Emden H., Rickelt, S., Wirtz S., Scherer V.: *A study on the validity of the multi-sphere Discrete Element Method*. Powder Technol. (188.2), p153-165. Dec 2008.

A Comparison of Grain-Grain Force Models 1 and 3: Supplemental Example

In this example, we start with a set of cube-like grains suspended a small distance from the ground and allowed to collide with the ground under gravity to eventually settle into a pile. A solid object is then pushed thru the stable pile to divide the pile into two smaller piles. Under force model 1, Section 2.3.2, Figure 12, we observe a separation of the original pile. However, viewing the last frame from different angles, there is no significant piling in the two smaller piles that result. Under force model 2, Section 2.3.4, Figure 13, we observe a separation of the original pile and piling in the resulting two smaller piles. This property enables us to simulate collections of particles in situations inappropriate with force model 1. Figure 14 is an example of tracing shapes through a pile of sand.

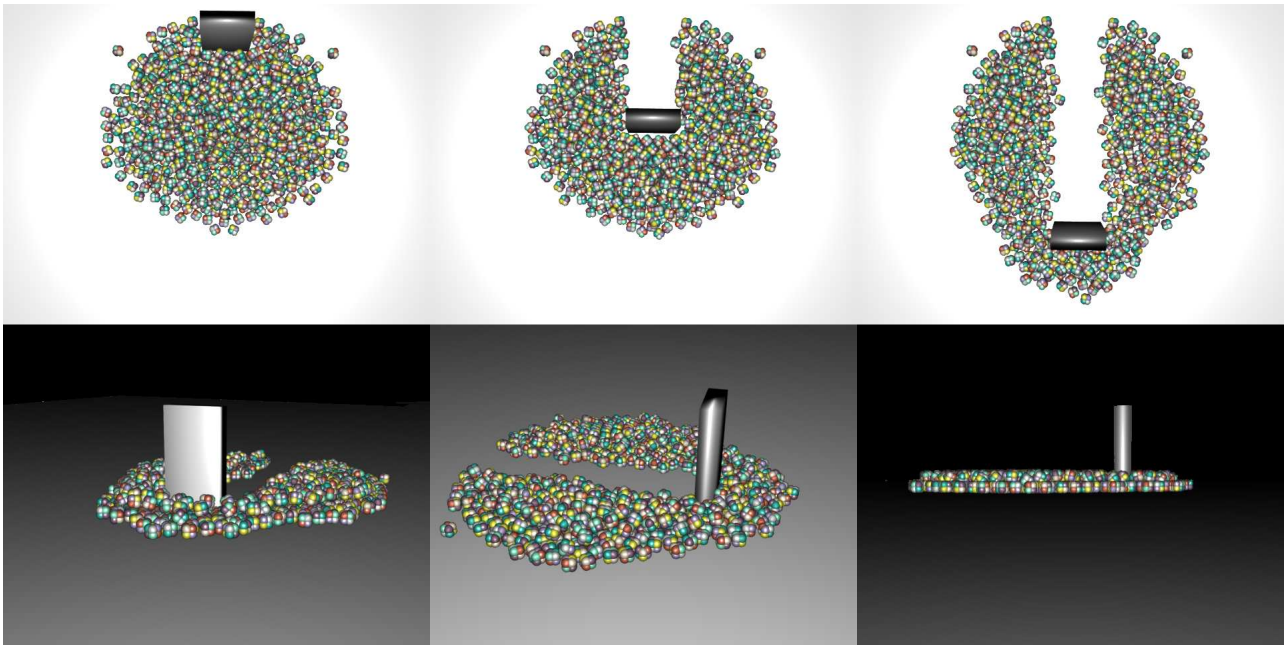

Figure 12: Snapshots of an object pushed through a pile of cube-like grains under force model 1, Section 2.3.2: Basic Model. $(\mu_s, \mu_r) = (1.0, 0.0005)$. First row: snapshots of three time frames during the simulation. Bottom row: different views of the last frame.

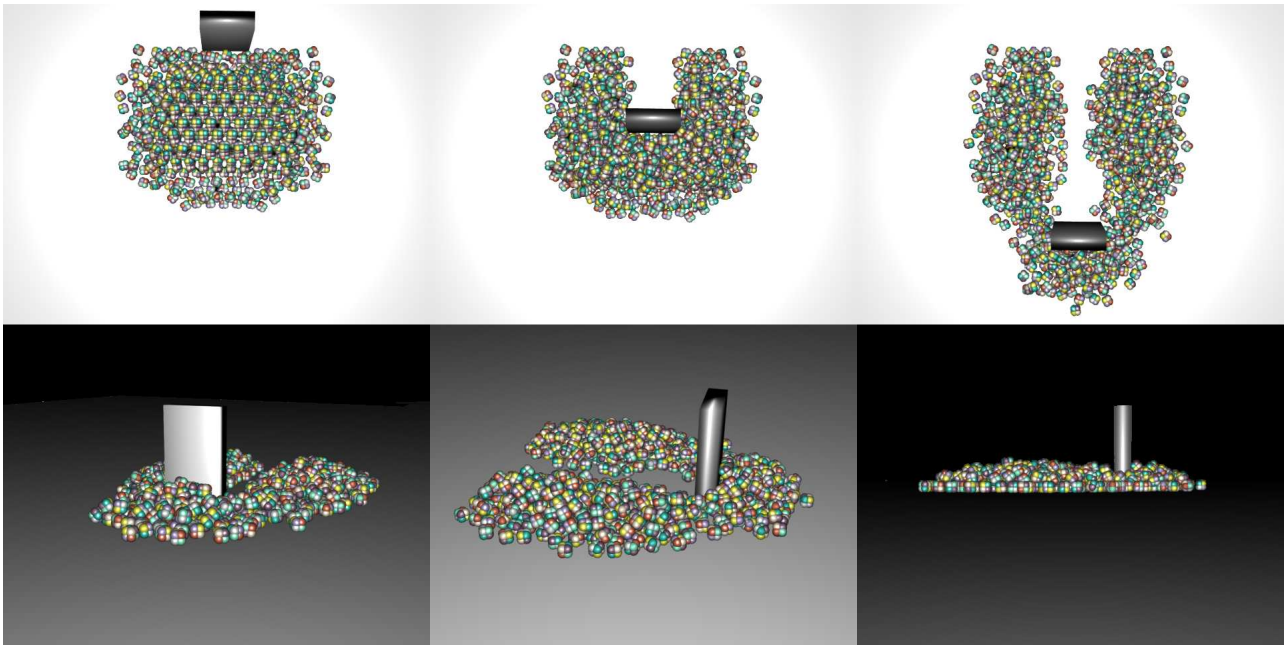

Figure 13: Snapshots of an object pushed through a pile of cube-like grains under force models 3, Section 2.3.4: Static Friction Model with Dampening. $(\mu_s, \mu_r) = (1.0, 0.0005)$. First row: snapshots of three time frames during the simulation. Bottom row: different views of the last frame.

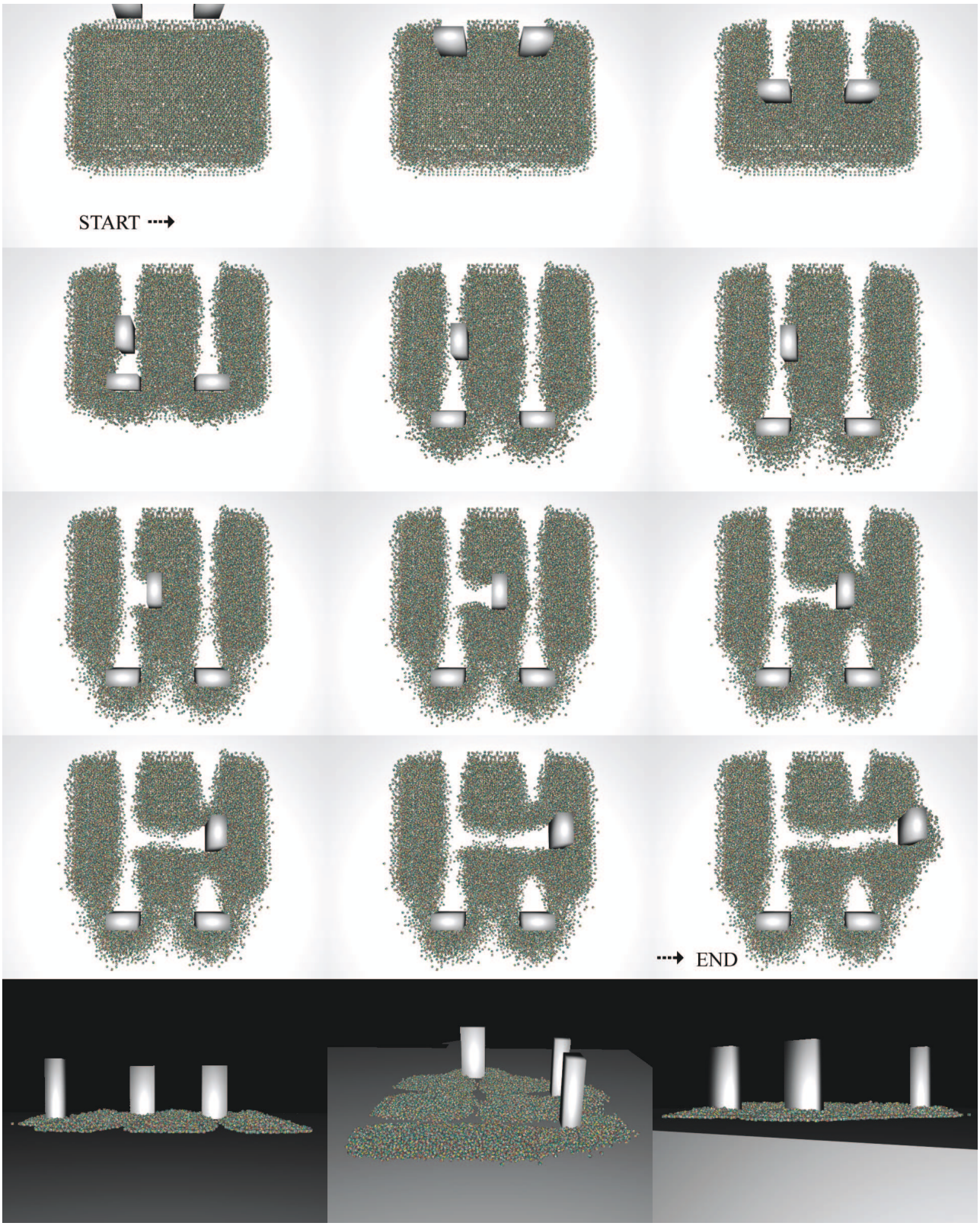

Figure 14: Simulation of tracing shapes through a pile of sand (10411 grains). $(\mu_s, \mu_r) = (0.5, 0.00025)$. First four rows: snapshots during the simulation. Bottom row: different views of the last frame.