

# Simulation of Realistic Particles with Bullet Physics Engine

Hantao He<sup>1</sup>, Junxing Zheng<sup>1,\*</sup>, Quan Sun<sup>1</sup>, and Zhaochao Li<sup>1</sup>

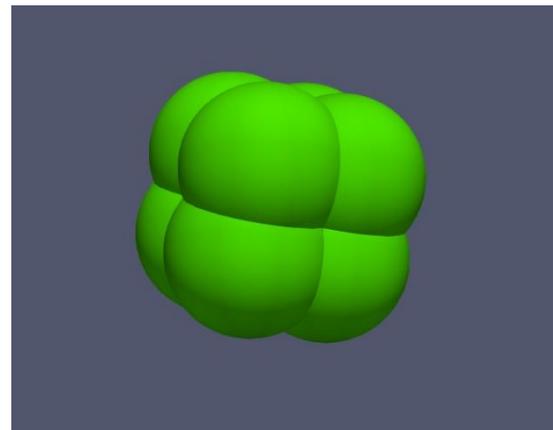
<sup>1</sup>Iowa State University, Department of Civil, Construction and Environmental Engineering, IA 50011, USA

**Abstract.** The traditional discrete element method (DEM) uses clumps to approximate realistic particles, which is computationally demanding when simulating many particles. In this paper, the Bullet physics engine is applied as an alternative to simulate realistic particles. Bullet was originally developed for computer games to simulate physical and mechanical processes that occur in the real world to produce realistic game experiences. Physics engines integrate a variety of techniques to simulate complex physical processes in games, such as rigid bodies (e.g., rocks, and soil particles), soft bodies (e.g., clothes), and their interactions. Therefore, physics engines have the capabilities to simulate realistic particles. This paper integrates three-dimensional laser scanner and Bullet to form a realistic particle simulation framework. The soil specimen collapse process is simulated to demonstrate the capability of the proposed framework to simulate realistic particles.

## 1 Introduction

The properties of granular materials have drawn great attention from various disciplines including earth science, geotechnical engineering, chemical engineering, powder metallurgy, pharmaceuticals, food processing and others. The discrete element method (DEM) has become the preeminent numerical tool for the study of granular material behavior. Typical DEM models use circular or spherical particles mainly due to the simplicity of contact detection and force calculations (Fig. 1). However, such oversimplified particulate models cannot provide adequately accurate insight to the mechanical behavior of granular assemblies. Methods to better simulate irregular particle shapes for use in DEMs have therefore been sought (Zheng and Hryciw [1]).

Simulation of realistic particles is an important research area in many disciplines. Researchers have explored the use of idealized particle shapes, such as ellipsoids [2], spherical cylinders [3], pentagons [4], rounded-cap elongated rectangles [5], and polyhedrons [6], in DEM simulations. These works have led to significant progress in the simulation of regular particles. However, these techniques cannot simulate irregular soil particles.



**Fig. 1.** Clumps of overlapping spheres applied by DEM to simulate objects.

Clumps of overlapping spheres (Fig. 1) are currently the most widely used method to model real soil particles (Zheng and Hryciw [1], McDowell and Harireche [7], Ferrellec and McDowell [8]). The contacts between overlapped spheres are ignored and the clumps behave as rigid bodies. The DEM computational load increases markedly with increasing numbers of spheres. Therefore, the clump method cannot simulate large numbers of particles.

Physics engines were originally developed for computer games to simulate physical and mechanical processes that occur in real life to produce a realistic game experiences. Physics engines integrate a variety of techniques to simulate complex physical processes in games, such as rigid bodies (e.g., rocks, and soil particles), soft bodies (e.g., clothes), fluids (e.g., water), and their

\* Corresponding author: [junxing@iastate.edu](mailto:junxing@iastate.edu)

interactions. Therefore, physics engines have the capabilities to simulate realistic particles.

Physics engines were first introduced to geotechnical engineering in 2014. Izadi and Bezujian [9] used the Bullet physics engine to simulate pluviation and vibration on three-dimensional (3D) randomly-shaped polyhedrons. Pthlos *et al.* [10] used a physics engine, Box2D, to simulate biaxial compression tests of two-dimensional (2D) randomly-shaped polygons. Izadi and Bezuijen [11] used Bullet to simulate 3D direct shear tests on uniform-sized steel spheres. These studies pioneered the applications of physics engines in granular soil simulations.

Izadi and Bezuijen [9, 11] have shown that the simulation speed of Bullet is 500–1000 times faster than traditional DEM code in simulations of 3D randomly-shaped polyhedrons. Bullet uses (1) both central processing units (CPUs) and graphics processing units (GPUs) of computers, (2) triangular face tessellations to represent realistic particles, and (3) a simplified contact model to accelerate computations. Despite of many physics engines, this research used Bullet as a simulation platform due to the high computational efficiency.

Prior studies [9, 11] focused on using Bullet to simulate 3D randomly-shaped polyhedrons. However, these particles cannot precisely represent realistic particles in physical granular soils. There is a lack of research addressing how to simulate realistic granular particles in Bullet. This paper will demonstrate the feasibility of using Bullet to simulate realistic particles by integrating 3D laser scanner and Bullet. 3D laser scanner is used to scan particle geometries, which are imported to Bullet to reconstruct virtual soil specimen. This study will also compare simulation strategies of Bullet and DEM.

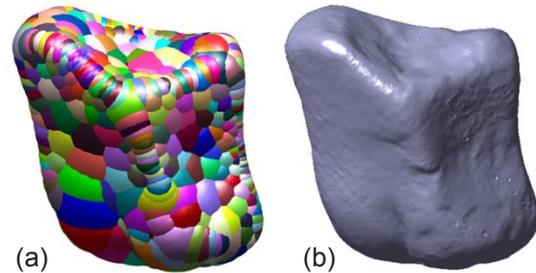
## 2 Bullet versus DEM

### 2.1 Particle Representations

In DEM, the 3D object geometries (e.g., soil particles) are usually approximated by sphere clusters, as shown in Fig. 2(a), for contact detection and force calculation. This method cannot precisely approximate particle geometries unless a very large number of spheres is used. The sphere cluster in Fig. 2(a) consists of 754 spheres. However, the artificial bumpy surface and some inaccurate representations of sharp corners are still visible. Having more spheres in a cluster gives more precise representations of particles, but higher computational loads are then needed.

On the other hand, in computer graphics and computer-aided design, 3D object geometries are usually represented by triangular face tessellations as shown in Fig. 2(b). For example, particles scanned by a 3D laser scanner are usually stored as triangular face tessellations in the STL format. More triangular faces yield more

precise representations of particles but require higher computational loads. Because Bullet can detect the contact and compute contact force between two triangular face tessellations, it can use triangular face tessellations in simulations, which precisely preserves particle shapes.

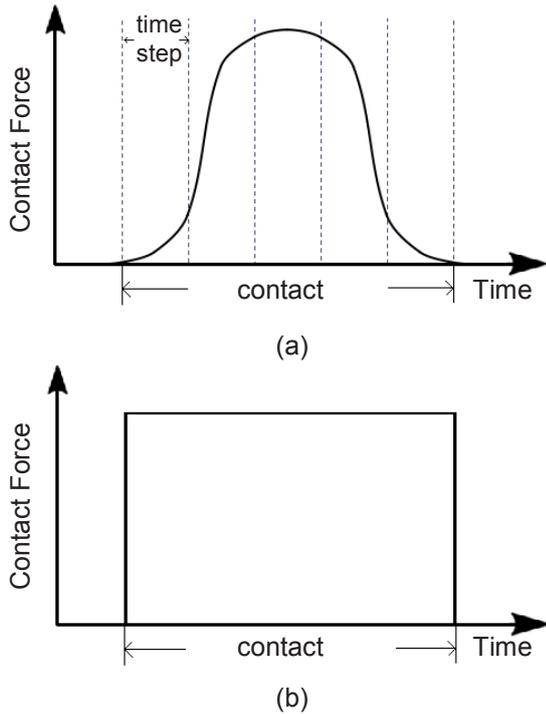


**Fig. 2.** Two representations of the realistic particle geometry in simulations: (a) The soil particle represented by a sphere cluster in DEM, and (b) The soil particle represented by a triangular face tessellation in Bullet.

### 2.2. Contact Model

Bullet and DEM use different contact models. DEM uses a soft contact approach, originally proposed by Cundall [12]. The soft contact model allows the contacting particles to overlap at contacts. The amount of overlapping determines the normal and frictional force following force-displacement laws as shown in Fig. 3(a). The key input contact parameters include normal stiffness, shear stiffness, and friction coefficient. The stiffnesses are usually set at very high values in order to yield small amounts of overlapping compared to particle size. Because of the high stiffnesses, the time step size in the computation has to be small to yield small elastic rebound at each step to ensure numerical stability. This significantly increases simulation time when simulating a large number of particles.

Bullet uses contact dynamics (CD), a hard contact model developed by Moreau [13-15] and Jean [16-18]. Here, the contacting particles are not allowed to overlap. After identifying two contacting particles, the velocities before collision are used to compute the velocities after collision. Then the Newton-Euler laws of motion are used to update the positions and orientations of the two particles and compute the contact force as shown in Fig. 3(b). The time step size used in the hard contact approach can be large compared to the soft contact approach without the risk of numerical instability. This reduces simulation times, especially when simulating a large number of particles. The key input parameters include the friction coefficient and the restitution coefficient.

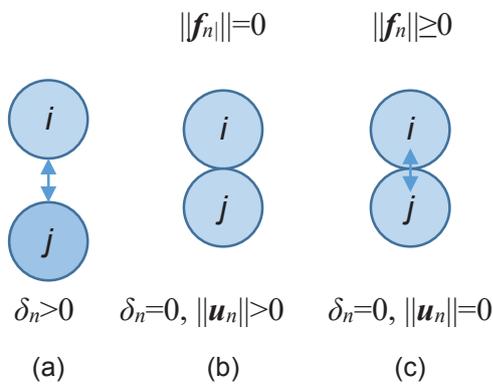


**Fig. 3.** Contact models in DEM and Bullet: (a) soft contact model in DEM, and (b) hard contact model in Bullet physics engine.

Two objects  $i$  and  $j$  are used to explain the hard contact model. The distance between objects  $i$  and  $j$  is  $\delta_n$ . Obviously, when  $\delta_n > 0$  (objects  $i$  and  $j$  are separated as shown in Fig. 4(a)), there is no contact force between objects  $i$  and  $j$ . When  $\delta_n = 0$  (objects  $i$  and  $j$  contact as shown in Fig. 4(b) and (c)), there is a normal force at the contact point, and the magnitude of the normal force  $f_n$  depends on the normal components of contact velocities (relative velocity of two objects)  $u_n$  as:

$$\begin{cases} \|f_n\| \geq 0 & \text{if } \|u_n\| = 0 \\ \|f_n\| = 0 & \text{otherwise} \end{cases} \quad (1)$$

Which is Signorini unilateral contact condition.



**Fig. 4.** Contact conditions between two objects when (a)  $\delta_n > 0$ , (b)  $\delta_n = 0$  and  $\|u_n\| > 0$ , (c)  $\delta_n = 0$  and  $\|u_n\| = 0$ .

The friction at the contact point,  $f_t$ , obeys the Coulomb's friction law, a complementary relation

between  $f_t$  and the tangential component of contact velocity  $u_t$  as:

$$\begin{cases} \|f_t\| \leq \mu \|f_n\| & \text{if } \|u_t\| = 0 \\ \|f_t\| = \mu \|f_n\| & \text{if } \|u_t\| > 0 \end{cases} \quad (2)$$

where  $\mu$  is the friction coefficient at the contact point.

In contact dynamics, the effect of contact is irrelevant to the process during the contact; the state before the collision determines the state after the collision. Thus, a contact velocity can be split into a left-hand velocity  $u^-$ , which is the velocity at a specific time  $t_0$ , and a right-hand velocity  $u^+$ , the velocity at the time of  $t_0 + \delta t$ . Then, contact velocities  $u_n$  and  $u_t$  depend on the left-hand and the right-hand contact velocities as:

$$u_n = \frac{u_n^+ + e_n u_n^-}{1 + e_n} \quad (3)$$

$$u_t = \frac{u_t^+ + e_t u_t^-}{1 + |e_t|} \quad (4)$$

where  $e_n$  and  $e_t$  are the normal and tangential parts of the restitution coefficients.

With the relations mentioned above, the motion for each particle in a system can be determined. In a two-dimensional physical system, the equations of motion of a particle depend on its mass  $m$ , moment of inertia  $I$ , mass center acceleration  $u$ , and angular acceleration  $\omega$  as:

$$m u = F + F_{ext} \quad (5)$$

$$I \omega = M + M_{ext} \quad (6)$$

where  $F$  and  $M$  are the vector sums of the forces and moments on all the contacts of the particle of interest respectively, and  $F_{ext}$  and  $M_{ext}$  represent the vector sums of external forces and moments applied on the particle of interest.

To determine the contact forces for a specific contact, the equations of motion for the contacts should be derived from the equations for the motion of particles. The conversions are written as:

$$u = G U \quad (7)$$

$$F = G^T f \quad (8)$$

where  $u$  is the contact velocities matrix,  $U$  is the particle velocities matrix,  $F$  is the particle forces matrix,  $f$  is the contact forces matrix, and  $G$  is a matrix containing the information about the geometry of the neighboring particles of the particle of interest. Thus, the contact forces on a specific contact  $f$  is determined as:

$$f = F_{neighboring} + F_{restitution} + F_{ext} \quad (9)$$

where  $F_{neighboring}$  is the resultant of the forces applied by the neighboring particles,  $F_{restitution}$  is the force due to the left-hand contact velocity, and  $F_{ext}$  is the resultant of the external forces applied on the contact of interest.

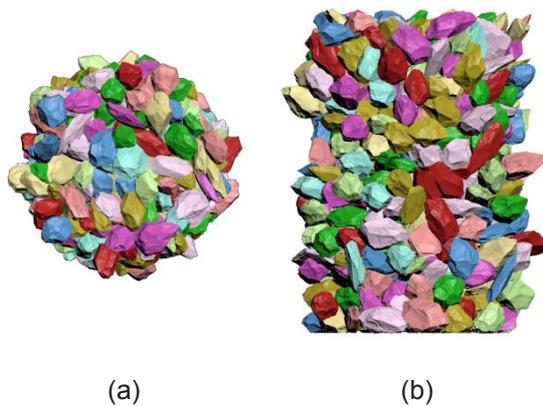
### 3 Particle Collapse Simulation

#### 3.1 Specimen Generation

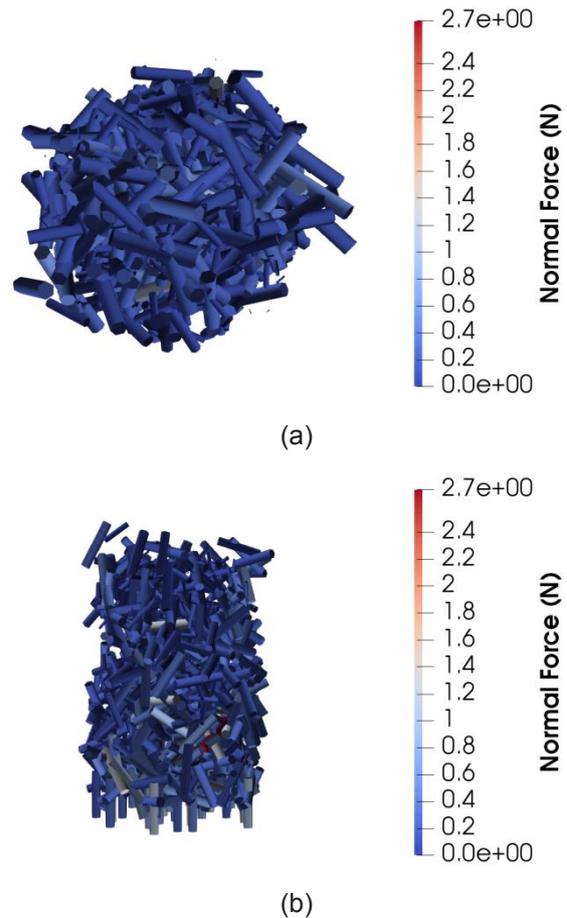
A total of 532 irregular uniform-sized particles having a size of around 30 mm are used in this study. These particles are scanned by a 3D laser scanner one by one, which are stored as “STL” files.

These “STL” files were imported into Bullet to build realistic particles. The simulation parameters include particle density = 2650 kg/m<sup>3</sup>, particle and wall friction coefficients = 0.5, the time step = 0.01 seconds, restitution coefficient = 0, and iterations per step = 70.

After importing particle geometries in Bullet, these virtual particles are funneled into a cylinder consisting of rigid wall. The particles settle down under gravity and generate a cylindrical soil specimen as shown in Fig. 5. The simulation is performed on the laptop with the following configurations: 8 core CPUs and 16 GB memory. The force chain of the generated cylindrical soil specimen is also calculated by Bullet and shown in Fig. 6.



**Fig. 5.** The specimen generated in Bullet: (a) top view and (b) the front view.

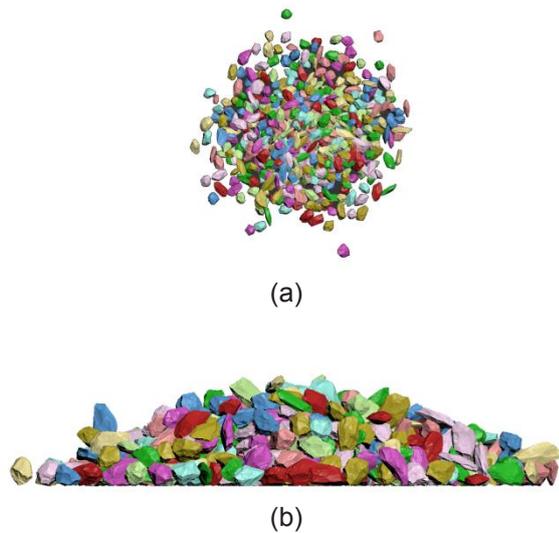


**Fig. 6.** The top view (a) and the front view (b) of the force chain of the generated cylindrical soil specimen before simulation.

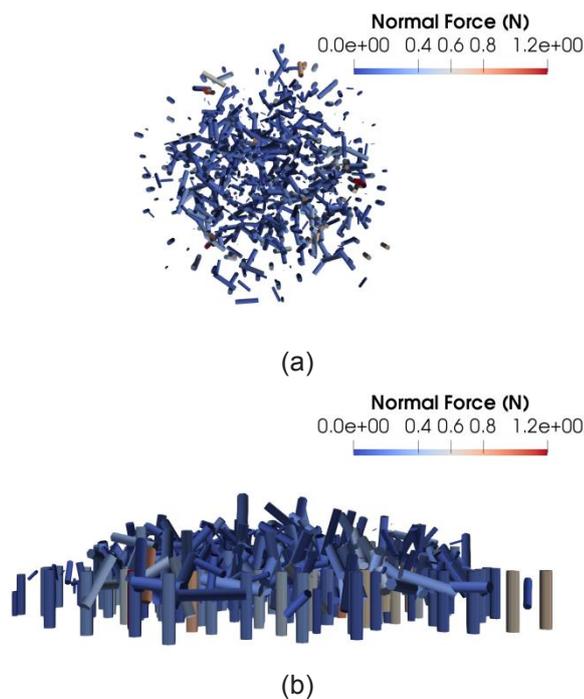
#### 3.2 Particle Collapse Simulation

After removing the cylinder, the particle column collapses under gravity. After all the particles are stable, the particles form an angle of repose of 20.56° (Fig. 7). After calculation, 1413 contact points as well as the normal contact forces on the points are found. The force chain is shown in Fig. 8.

The time of simulation is around 20 seconds. We have performed the same simulation in DEM, which took around 33 hours.



**Fig. 7.** Collapsed particles in Bullet: (a) top view and (b) the front view.



**Fig. 8.** The top view (a) and the front view (b) of the force chain of the formed heap.

## 4 Summary

In this paper, we have demonstrated the feasibility of simulating realistic particles using Bullet physics engine. The Bullet physics engine improves simulation speed and yields acceptable results. Physics engines can be promising alternative approach to DEM for realistic granular particle simulation, and have great potentiality for applications in the area of geotechnical engineering.

## References

1. J. Zheng, R.D. Hryciw, *Granul Matter*, **18**, 84 (2016)
2. T.-T. Ng, *Int J Numer Anal Methods Geomech*, **33**, 511-527 (2009)
3. L. Pournin, M. Weber, M. Tsukahara, J.A. Ferrez, M. Ramaioli, T.M. Liebling, *Granul Matter*, **7**, 119-126 (2005)
4. E. Azéma, F. Radjal, R. Peyroux, G. Saussine, *Phys Rev E*, **76**, 011301 (2007)
5. E. Azéma, F. Radjal, *Phys Rev E*, **81**, 051304 (2010)
6. E. Azéma, F. Radjal, G. Saussine, *Mech Mater*, **41**, 729-741 (2009)
7. G.R. McDowell, O. Harireche, *Géote*, **52**, 131-135 (2002)
8. J. Ferrellec, G. McDowell, *Géote*, **60**, 227-232 (2010)
9. E. Izadi, A. Bezuijen, *Proc. 3rd International Symposium on Geomechanics from Micro to Macro*, 1565-1570 (Taylor and Francis Group, London, 2014)
10. M. Pytlos, M. Gilbert, C.C. Smith, *Géote L*, **5**, 243-249 (2015)
11. E. Izadi, A. Bezuijen, *PLOS ONE*, **13**, e0195073 (2018)
12. P.A. Cundall, O.D.L. Strack, *Géote*, **29**, 47-65 (1979)
13. J.J. Moreau, *J Differ Eq*, **36**, 347-374 (1977)
14. J.J. Moreau, *Nonsmooth Mechanics and Applications*, 1-82 (Springer, Vienna, 1988)
15. J.J. Moreau, *Eur J Mech A. Solids*, **13**, 93-114 (1994)
16. M. Jean, E. Pratt, *Int J Eng Sci*, **23**, 497-513 (1985)
17. M. Jean, *Studies Appl Mech*, **42**, 463-486 (1995)
18. M. Jean, *Comput. Methods Appl Mech Eng*, **177**, 235-257 (1999)