



# Three-dimensional discrete element simulation for granular materials

Three-dimensional simulation

Dawei Zhao, Erfan G. Nezami, Youssef M.A. Hashash and Jamshid Ghaboussi

*Department of Civil and Environmental Engineering,  
University of Illinois at Urbana-Champaign, Urbana, Illinois, USA*

749

## Abstract

**Purpose** – Develop a new three-dimensional discrete element code (BLOKS3D) for efficient simulation of polyhedral particles of any size. The paper describes efficient algorithms for the most important ingredients of a discrete element code.

**Design/methodology/approach** – New algorithms are presented for contact resolution and detection (including neighbor search and contact detection sections), contact point and force detection, and contact damping. In contact resolution and detection, a new neighbor search algorithm called TLS is described. Each contact is modeled with multiple contact points. A non-linear force-displacement relationship is suggested for contact force calculation and a dual-criterion is employed for contact damping. The performance of the algorithm is compared to those currently available in the literature.

**Findings** – The algorithms are proven to significantly improve the analysis speed. A series of examples are presented to demonstrate and evaluate the performance of the proposed algorithms and the overall discrete element method (DEM) code.

**Originality/value** – Long computational times required to simulate large numbers of particles have been a major hindering factor in extensive application of DEM in many engineering applications. This paper describes an effort to enhance the available algorithms and further the engineering application of DEM.

**Keywords** Finite element analysis, Simulation, Materials management, Flow

**Paper type** Research paper

## Nomenclature

$A_i, B_i$	= projection on the CP of vertices of particles A and B inside the contact zone	$\mathbf{F}$	= total force acting on a particle
$CP_A, CP_B$	= the boundaries of the contact zone	$\mathbf{F}_N, \mathbf{F}_S$	= normal and shear contact forces
$\mathbf{C}_M, \mathbf{C}_I$	= damping materices	$\mathbf{F}_{EN}, \mathbf{F}_{ES}$	= elastic components of normal and shear contact forces
$D_N$	= normal penetration distance of particles	$\mathbf{F}_{DN}, \mathbf{F}_{DS}$	= damping components of normal and shear contact forces
$d, d_1, d_2$	= damping parameters	$f$	= time factor



This work is sponsored by the National Science Foundation under Grant No. CMS-0113745. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The authors would also like to thank Mr Ibrahim Mohammad for the visualization program VisDEM3D.

$\mathbf{I_p}$	= moment of inertia tensor	$\mathbf{n}$	= contact normal vector
$I_i$	= ( $i = 1, 2, 3$ ) principal moment of inertia of a particle around the $i$ th principal axis	$\mathbf{U}$	= particle's translation vector
$K_N, K_{NN}, b$	= contact normal stiffness parameters	$\mathbf{U_N, U_S}$	= relative displacement of contact points in normal and shear direction at the location of contact
$K_S$	= shear stiffness of contact	$R$	= speed up ratio
$\mathbf{M}$	= mass matrix	$\alpha, \beta$	= convex hull of points $A_i$ and $B_i$
$M_i$	= ( $i = 1, 2, 3$ ) resultant moment vector along the $i$ th principal axis	$\alpha_d, \beta_d$	= viscous damping proportionality factor
$m$	= mass of particles	$\Delta t$	= time step length
$m_{\min}$	= mass of the smallest particle in the system	$\phi$	= inter-particle friction angle
$N$	= vertical normal force in direct shear test	$\boldsymbol{\theta}$	= particle's rotation vector
		$\rho$	= mass density of particles
		$\omega_{\max}$	= highest natural frequency of the discrete element system

## 1. Introduction

The study of granular material is of great interest in industrial and mining applications (Cleary, 2000), geotechnical engineering (Campbell *et al.*, 1995; Hopkins *et al.*, 1991), new machine or equipment design, productivity improvement, working safety enhancement, and procedure automation (Hemami, 1995; Singh, 1997). In all these applications the behavior of the granular material as well as the interaction between the soil and tool parts during digging, scooping and dumping (Hemami, 1995) is studied. For instance, in the design of wheel-loaders, designers need to understand the details of how the soil interacts with buckets, such as the impact force distribution and magnitude, and wear and tear positions. The nature of these interactions is determined by particle size, soil density, friction angle, cohesion, and other soil properties.

The simulation methods of granular materials have different characteristics from those based on continuum mechanics due to material discontinuity. Simulation of granular flow often involves modeling large motion of discrete particles and significant geometry changes of soil masses. Discrete element method (DEM) is one of the most effective tools for the study of large displacements of granular materials.

## 2. Background

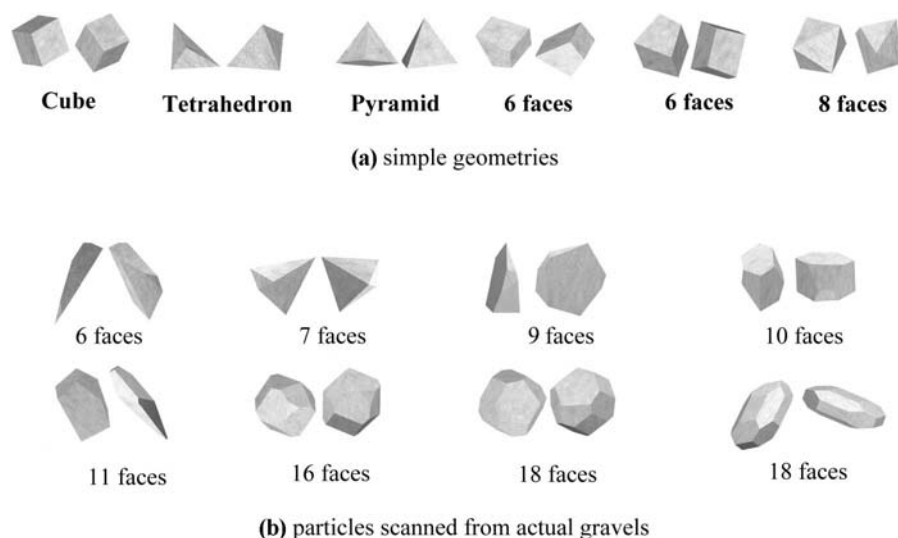
DEM was first introduced by Cundall(1971) for simulation of jointed rocks. Since, then, significant progress has been made in the development of DEM methodologies and applications. Some applications include simulation of landslides (Cleary and Campbell, 1993; Campbell *et al.*, 1995), ice flows (Hopkins *et al.*, 1991), pharmaceutical powder blending (Johnson *et al.*, 2005; Yang *et al.*, 2002), as well as industrial and road construction activities such as dragline excavation and mixing (Cleary, 2000), and silo filling (Moakher *et al.*, 2000).

Large-scale discrete element simulations demand massive computational resources due to large number of particles and time-consuming contact detection procedures. For computational convenience, the most commonly used particle geometries are circular discs or ellipses for 2D simulations and spheres or ellipsoids for 3D simulations. Some representative programs include BALL (Strack and Cundall, 1978; Cundall and Strack, 1979), TRUBAL (Strack and Cundall, 1984), ELLIPSE2 (Ng, 1992), ELLIPSE3D

(Lin and NG, 1997). Such particle geometries greatly simplify the contact detection procedure and reduce the CPU run time, but fail to capture essential aspects of mechanical behavior and geometric interaction of particulate and angular material (Matuttis *et al.*, 2000). Ghaboussi and Barbosa (Barbosa, 1990) developed 2D programs ParFlow, BLOCKS2D and DBLOCKS, which simulate rigid and deformable polygonal particles, respectively, and 3D program BLOCKS3D (Ghaboussi and Barbosa, 1990) to simulate polyhedral particles. In these codes, contact detection is performed by comparing the relative positions of every vertex of a particle to every face of adjacent particles and vice versa, to find the contact point and the penetration distance (Ghaboussi and Barbosa, 1990; Coutinho, 2001). Cundall (1988a, b) proposed a common plane method (CPM), and implemented it in the DEM program 3DEC. CPM defines a virtual common plane (CP) that bisects the space between two particles, whereby the particle-particle contact detection problem is simplified to an easy particle-plane contact detection problem. In general, CPM requires a large number of iterations to obtain the CP. Nezami *et al.* (2004) proposed an improved version of CPM called fast common plane method (FCP). In FCP, the number of iterations is significantly reduced by limiting the search space of the CP to a few candidates. In 2004, the shortest link method (SLM) is developed by Nezami *et al.* (2006).

### 3. Development of new algorithms and DEM code BLOKS3D

Several new algorithms and features are developed and implemented in a newly developed discrete element code BLOKS3D for the simulation of granular material flow and tool-soil interaction. The program is developed for simulation of rigid 3D polyhedral particles of any size. It uses a particle library in which geometric information of particle prototypes are pre-calculated and stored. The library can be edited, expanded, or reduced based on the simulation requirements. Figure 1 shows particles available in the current particle library, some of which are generated based on 3D scans from actual gravel particles (Tutumluur *et al.*, 2006)



**Figure 1.**  
Particle geometries  
available in particle  
library

Figure 2 shows a flow chart of the DEM code, and the new developments are highlighted as follows:

- Motion integration: New moment of inertia formulation for rigid particle motion updating.
- Particle contact resolution and detection:
  - Neighbor search algorithm: an efficient search for neighbors of each particle.
  - Contact detection: two new and efficient contact detection algorithms.
- Contact point(s) determination.
- Contact force(s) and damping.

These features are described in detail in the following sections.

3.1 Motion integration and rigid particle moment of inertia

The 3D rigid discrete elements have six degrees of freedom; three translational and three rotational. From a numerical point of view it is more efficient to update the translational degrees of freedom in a global coordinate system and to update the rotational degrees of freedom in the particle's principal coordinate system.

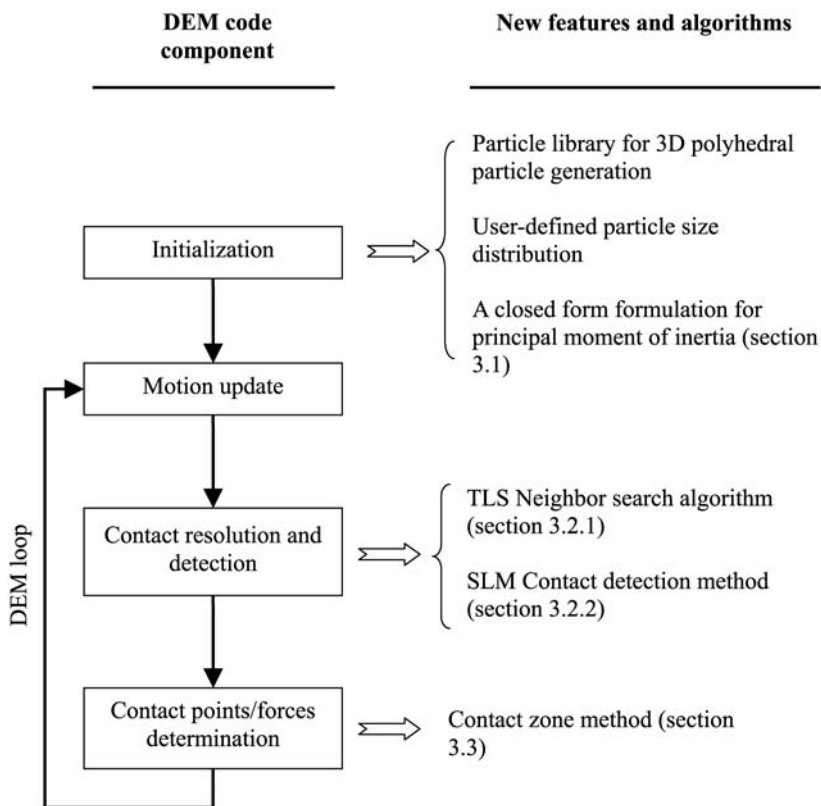


Figure 2.  
Flow chart of BLOKS3D  
discrete element code

In the principal coordinate system, the moment of inertia tensor is a diagonal matrix with diagonal terms being the principal moments of inertia. Euler's equations for the motion of a rigid particle can be expressed as (Cundall, 1988b):

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}_\mathbf{M}\dot{\mathbf{U}} &= \mathbf{F} \text{ for translation} \\ \mathbf{I}_\mathbf{P}\ddot{\boldsymbol{\theta}} + \mathbf{C}_\mathbf{I}\dot{\boldsymbol{\theta}} &= \bar{\mathbf{M}} \text{ for rotation} \end{aligned} \quad (1)$$

where:

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}$$

is the mass matrix:

$$\mathbf{I}_\mathbf{P} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

is the moment of inertia tensor in principal coordinate system;  $\mathbf{C}_\mathbf{M} = \alpha_\mathbf{d}\mathbf{M}$  and  $\mathbf{C}_\mathbf{I} = \alpha_\mathbf{d}\mathbf{I}_\mathbf{P}$  are damping matrices; and:

$$\bar{\mathbf{M}} = \begin{Bmatrix} M_1 + (I_2 - I_3)\dot{\theta}_2\dot{\theta}_3 \\ M_2 + (I_3 - I_1)\dot{\theta}_3\dot{\theta}_1 \\ M_3 + (I_1 - I_2)\dot{\theta}_1\dot{\theta}_2 \end{Bmatrix}.$$

By definition, the determination of the principal axes and the principal moments of inertia  $I_i$  of a particle involves integrations over the mass of the particle. For convex polyhedral particles, a closed form formulation (Appendix A) can be developed to calculate the moment of inertia tensor in any arbitrary coordinate system. Although the computations involved are not trivial, they can be done once for each particle prior to the simulation and the values are stored for later use during the analysis.

DEM uses the second central difference method to integrate the equations of motion in equation (1). As a result, the required time step length  $\Delta t$  should follow the stability criterion:

$$\Delta t \leq 2/\omega_{\max} \quad (2)$$

where:

$$\omega_{\max} = \sqrt{\frac{K_N}{m_{\min}}}$$

is the highest natural frequency of the discrete element system. Another constraint for  $\Delta t$  is due to accuracy and convergence considerations:  $\Delta t$  should be small enough to

guarantee that the penetrations that occur in a single time step remain small compared to the particle size even for the fastest moving particles. As a result equation (2) is usually modified:

$$\Delta t = f*2/\omega_{\max} \tag{3}$$

whereby time step factor  $0 < f \leq 1$  is a function of the maximum relative velocity of the particles. Numerical experiments (Barbosa, 1990) show that a time step factor between 0.05 and 0.4 (with suggested value of 0.1) is appropriate for the range of particle velocities observed in typical geotechnical simulations.

3.2 Particle contact resolution and detection

Contact resolution and detection is commonly performed in two consecutive phases (Perkins and Williams, 2001): neighbor search and contact detection (Figure 3). Neighbor search phase develops a neighbor list of all potential interacting particles within a neighborhood of the target particle. Neighbor search algorithms are usually independent of the particle shapes and are applicable to a wide range of particle geometries. An efficient neighbor search method should be able to minimize the size of neighbor lists by excluding as many particles as possible that are not in contact with the target particle.

Contact detection phase compares the geometry of the target particle and its neighbors in detail. Contrary to the neighbor search, contact detection schemes are developed for specific particle geometries and are not always applicable to other types of geometries. The contact detection run time increases with complexity of the particle geometry, while neighbor search time remains practically unchanged. As a result, for complex geometries such as polyhedrons, contact detection is computationally much more demanding than neighbor search.

3.2.1 Neighbor search. A two-level-search (TLS) scheme based on space decomposition and bounding spheres is developed and implemented in BLOKS3D. This scheme is an enhanced version of the neighbor search algorithm implemented in 3DEC (Cundall, 1988a). The whole space of interest is discretized into Eulerian cubic “boxes”. For illustration purpose, Figure 4 shows box discretization in 2D. Each particle has a “box list” consisting of all of boxes that overlap with that particle. Correspondingly, each box has a “particle list” that includes the particles that overlap

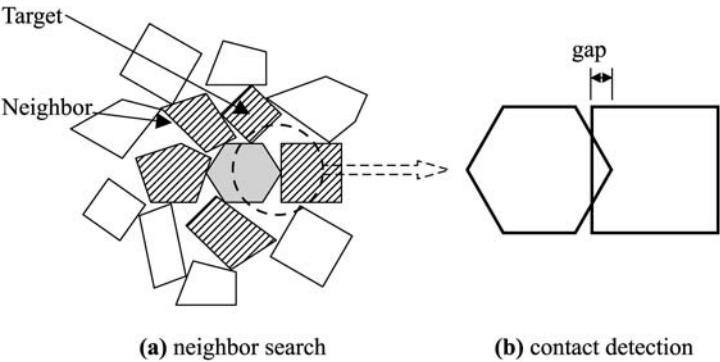
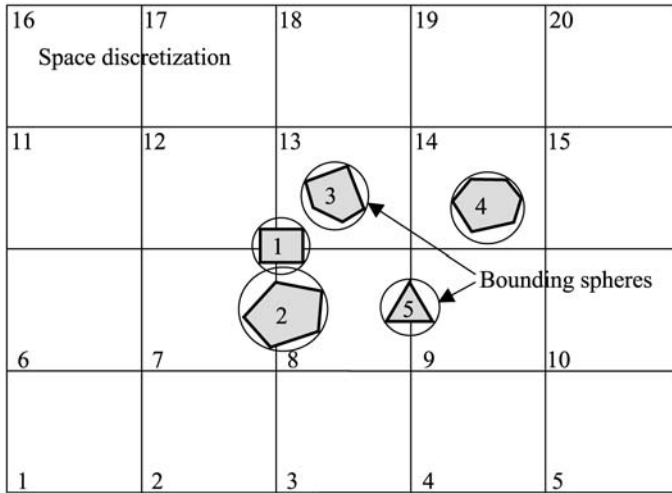


Figure 3.  
Contact resolution and  
detection



**Figure 4.**  
Two level neighbor search  
scheme: level 1: box  
list/particle list mapping;  
level 2: bounding sphere  
overlap check

with that box. For example, in Figure 4, particle 1's box list includes boxes 7, 8, 12, and 13; and box 8's particle list includes particles 1, 2 and 5. These lists are obtained by defining a cubic bounding volume around each particle and comparing it against the boxes. For simplicity, the cubic bounding volume is assumed to be parallel to axes of the global coordinate system. Box lists and particle lists have to be updated at every time step to account for the motion of particles.

The first level of TLS provides a preliminary list of neighbors, consisting of all particles that overlap the same box.

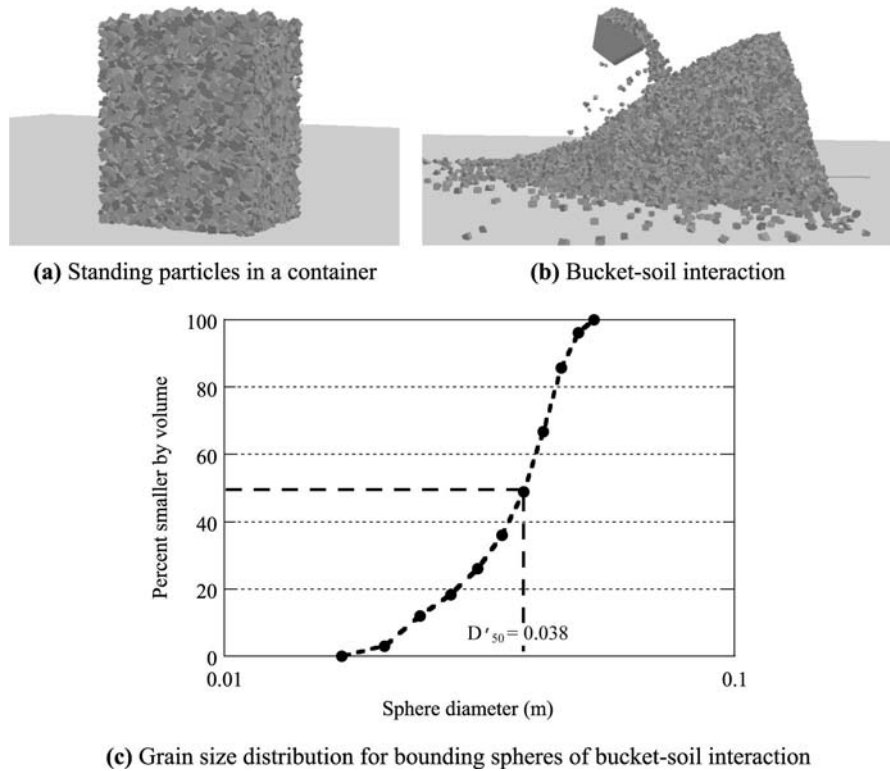
The second level search further reduces the number of neighbors by considering a "bounding sphere" for each particle. The center of sphere located at the centroid of the particle, and its radius equal to the largest distance from the centroid of the particle to its vertices. The bounding spheres of the neighbor pair obtained in the first level search are checked for overlap. If no overlap exists, these two particles are not in contact and are removed from the neighbor list.

The performance of the TLS algorithm is dependent on the particle shape and the ratio of the box size  $S$  to average bounding sphere size  $D_{50}$ . For a given mass of particles the average length of the particle lists decreases with the ratio  $S/D_{50}$ . The quantity  $D_{50}$  is the median diameter of bounding spheres by volume. This is similar to traditional definition of  $D_{50}$  in geotechnical engineering, but is applied to bounding spheres rather than the particles.

The relationship between TLS performance and  $S/D_{50}$  is investigated empirically through a series of analyses using two sets of particle configurations shown in Figure 5:

- (1) Uniform cubic particle assemblies in a rectangular box: Three different tests are performed with 9,000, 20,000 and 50,000 particles, respectively, (Figure 5(a)).
- (2) Loader bucket-medium interaction: the simulation represents an intermediate stage of a typical tool-medium interaction problem with 20,952 particles as shown in Figure 5(b). The particle shapes are randomly chosen from the particle library. The size distribution of bounding spheres is shown in Figure 5(c).





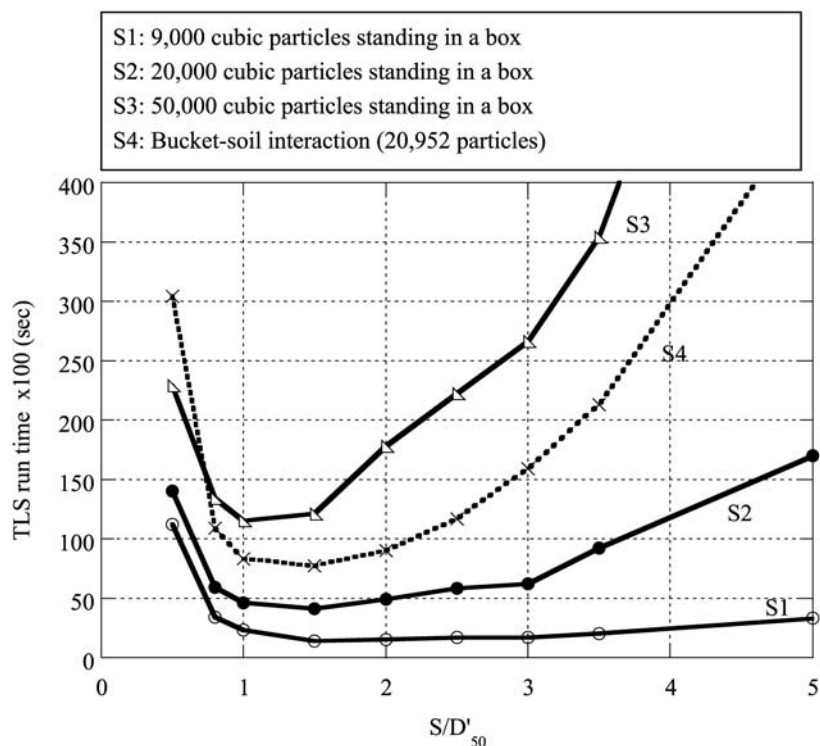
**Figure 5.**  
Case study scenarios for  
optimum box size  
determination for TLS  
neighbor search algorithm

For each particle configuration the TLS algorithm is used with various box sizes. Figure 6 shows the corresponding run times for the uniform particle assemblies and the bucket-soil interaction. The TLS algorithm run time first decreases with  $S/D'_{50}$  down to a minimum value and then increases. If the box size is too small (small  $S/D'_{50}$ ) there are simply many boxes to go through and one large particle may need to traverse many boxes to find out all its neighbors. The TLS run time increases as the box size decreases. However, for large box sizes (large  $S/D'_{50}$ ), each box contains a large number of particles and the TLS will need to perform unnecessary checks for pairs of particles too far from each other. Therefore, TLS run time increases again for large box sizes.

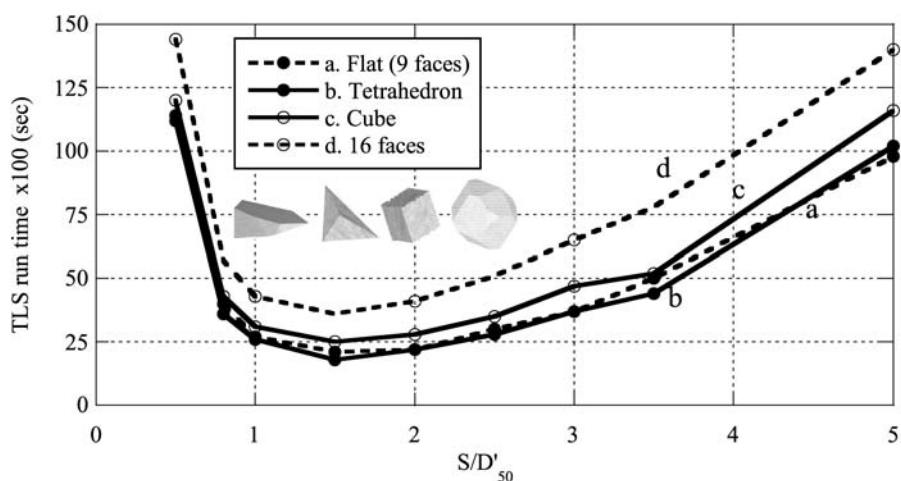
Figure 7 plots the effect of particle shape on the TLS run time for 50,000 uniform particles standing in a box. Four different particle types are examined including: 16-face polygons, cubes, tetrahedrons and flat plates. CPU run time increases as the particle shape approaches that of a spherical or rounded particle because on average particles with more rounded geometries overlap with a larger number of boxes, and thus have longer box lists.

Analyses in Figures 6 and 7 suggest that TLS optimal performance corresponds to approximately  $S/D'_{50} = 1.5$ . In BLOKS3D, implementation  $D'_{50}$  is calculated for each simulation and the optimum box size is obtained.





**Figure 6.** Box size dependency for TLS algorithm (simulation of particles standing in a box and bucket-soil interaction)



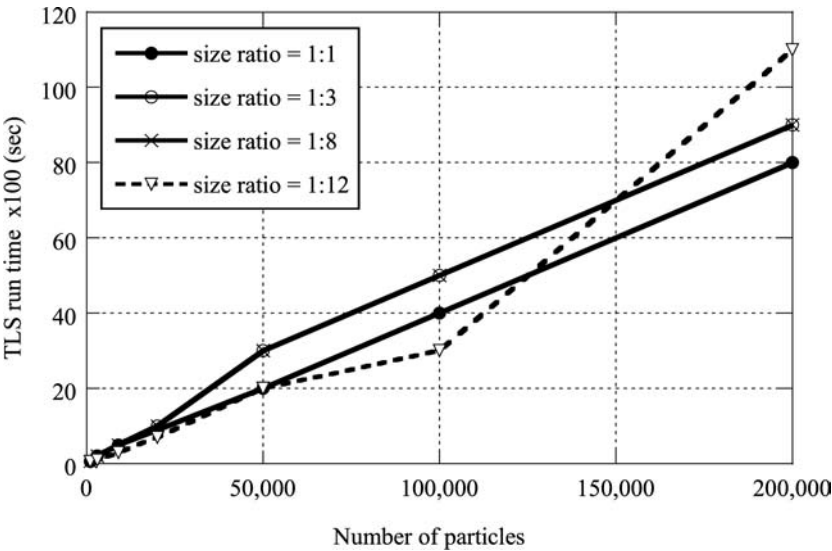
**Figure 7.** TLS performance: effect of particle shape (50,000 uniform particles standing in a box)

TLS algorithm belongs to the category of hashing algorithms with  $O(N)$  performance, where  $N$  is the total number of particles. It can be used with both concave and convex particles. The algorithm can be summarized as follows:

```
Loop over all particles (loop 1)
{
  Loop over the box list for current target particle (loop 2)
  {
    Loop over the particle list for current box (loop 3)
    {
      Check for bounding volume overlap between particles.
    }
  }
}
```

The operations performed in loops 2 and 3 are only proportional to the number of particles in the local area around the target particle, but not related to  $N$ . So the overall computation is proportional to  $N$  because loop 1 is over all particles.

The performance of the TLS is also studied as a function of particle size distribution. A measure of particle size distribution is the particle size ratio, defined as the ratio of smallest to largest diameters of particle bounding spheres. Figure 8 shows the influence of the particle size distribution on the performance of TLS algorithm for 1,000 to 200,000 cubic particles standing in a box. The particle size distribution varies from 1:1 to 1:12. The run times for TLS are linearly dependent on  $N$  and relatively independent of particle size ratio.

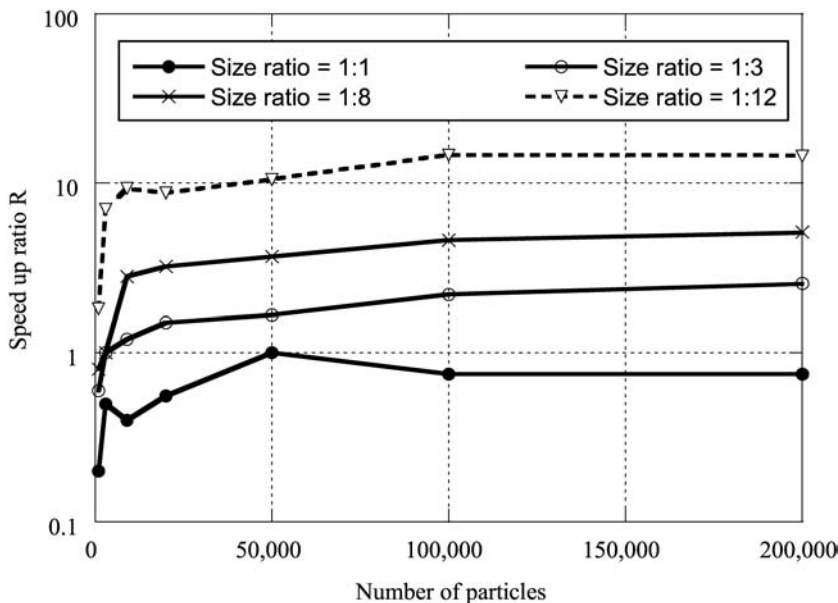


**Figure 8.**  
TLS performance: effect of  
size ratio (cubic particles  
standing in a box)

The performance of TLS algorithm is compared with two neighbor search algorithms, DESS (Perkins and Williams, 2001) and NBS (Munjiza and Andrews, 1998).

DESS is a spatial sorting algorithm that readily locates the particle neighbors after the particles are sorted in order in space. Its performance is insensitive to particle size variation. However, its computational cost depends on the type of sorting algorithm used. The run time is of the order of  $O(N^2)$  for insertion sort, and order of  $O(N \ln N)$  for heap sort. NBS is a spatial hashing algorithm. After the space is discretized into identical boxes, each particle is assigned to only one box. The neighbor search of each particle is only performed inside of the neighbor boxes. The run times of NBS is of the order of  $O(N)$ , which is advantageous for large-scale simulations. However, NBS may suffer significant performance degradation with large particle size variation that may result in a large amount of over-reporting of possible neighbors (Perkins and Williams, 2001).

Figure 9 shows the performance comparison between TLS algorithm and NBS algorithm for the standing particles simulation. The vertical axis is the speed up ratio  $R$ , defined as the ratio of run time of NBS algorithm to that of TLS algorithm. In order to reduce the over-reporting of neighbors in NBS algorithm for large particle size variation, the bounding sphere check is added to the original NBS algorithm so both algorithms are guaranteed to report the same set of particle neighbors. The run time of modified NBS algorithm used in the comparison also includes the computation of the bounding sphere check. The figure shows that modified NBS algorithm has better performance when particles are uniform. On the other hand, TLS algorithm outperforms modified NBS with increasing particle size variation. When particle size ratio is 1:12, the TLS algorithm is about ten times faster than modified NBS algorithm when more than 10,000 particles are used in the simulation.



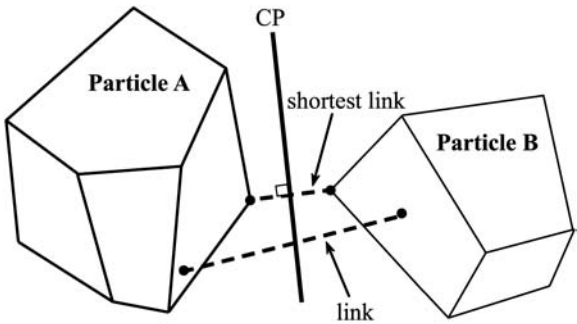
**Figure 9.**  
Speed up ratio  $R$  (CPU run  
time of NBS to that of  
TLS) for different size  
ratios (cubic particles  
standing in a box)

**3.2.2 Contact detection.** Once neighbor search is completed, contact detection is performed for each neighbor pair (Figure 3(b)). SLM contact detection algorithm proposed by (Nezami *et al.*, 2006) provides a new approach to find the CP and is faster than CP method (Cundall, 1988a, b) and fast common plane (FCP) method (Nezami *et al.*, 2004). The method is restricted to convex particles. Concave particles can be simulated as series of attached convex particles. In SLM, a “link” between particles A and B is defined as shown in Figure 10, and the two ends of the link are allowed to slide along particle surfaces to find the “shortest link”. The “perpendicular bisector plane” of the shortest link coincides with the CP between the particles. After the CP is located for a neighbor pair, a “gap” is defined to describe the relative distance between two particles. If two particles are in contact, both particles should intersect the CP, and the gap is equal to the sum of the penetration distance for each particle to the CP. SLM and FCP algorithms method are implemented in BLOKS3D.

### 3.3 Contact point and contact force determination

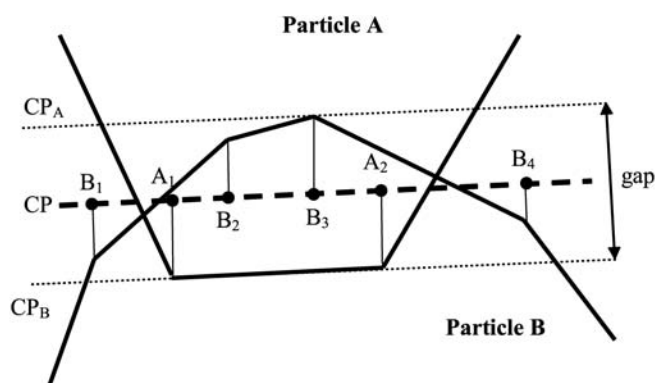
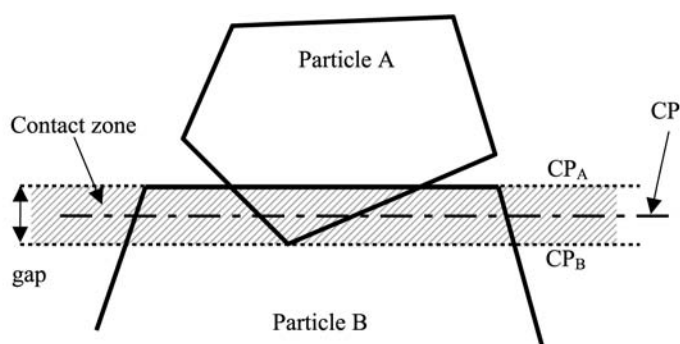
When two particles are in contact the CP intersects both of them. The intersections of the polyhedral particles with the CP form two convex polygons in the plane of the CP. These two polygons always intersect each other to form a convex overlapping area defining the contact area between the two particles. Ideally, the contact force can be represented as the resultant of a (normal and shear) stress distributed over the contact area. However, defining the intersection of the CP and particles, and computing the stress distribution requires lengthy calculations. As a result, the intersection of CP and each particle is approximated using a “contact zone” method explained in detail in the following sections. First a contact zone is defined at the location of the contact. The geometry of particles inside the contact zone is investigated in detail to obtain contact points and then the contact forces. A contact may include one or more contact points, and the contact forces are calculated at each contact point in normal and shear direction of the CP separately. As a result the stress distribution is replaced by a set of discrete (normal and shear) forces calculated at the locations of each vertex of the contact area based on the penetration distance of the vertex.

**3.3.1 Contact zone.** Let  $CP_A$  and  $CP_B$  denote planes parallel to and at a gap/2 distance from the CP (Figure 11). The narrow zone between the two planes  $CP_A$  and  $CP_B$  is defined as the “contact zone” between the two particles. Any possible overlap between the particles occurs inside the contact zone. Let  $A_i$  and  $B_i$  denote the respective projections on the CP of those vertices of particles A and B, which are located inside the contact zone (Figure 12(a)). Let  $\alpha$  and  $\beta$  represent the convex hull for points  $A_i$  and  $B_i$ ,

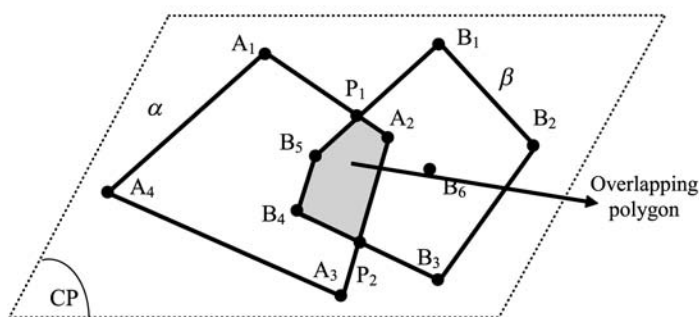


**Figure 10.**  
A link and shortest link in  
SLM

**Figure 11.**  
Definition of contact zone



**(a)** projection of vertices on the CP



**(b)** polygons  $\alpha$  and  $\beta$  are convex hulls for points  $A_i$  and  $B_i$  respectively

**Figure 12.**  
The contact zone method  
and definition of contact  
points

respectively, (a convex hull for a set of points is the smallest possible convex polygon containing all those points), Figure 12(b). These two polygons approximate the intersection of the CP and each particle. Note that in special cases, the polygons  $\alpha$  or  $\beta$  and consequently the contact area may consist of only one point or one line segment.

**3.3.2 Contact points and penetration distances.** The overlapping area between the two polygons  $\alpha$  or  $\beta$  (Figure 12(b)) shows the contact area between the two particles. A simple algorithm proposed by (Toussaint, 1985) is used to obtain the overlapping area between  $\alpha$  and  $\beta$ . The algorithm is of order  $O(m + n)$  where  $m$  and  $n$  are the number of vertices of the two polygons. In the current version implemented in BLOKS3D each vertex of the contact area (points  $A_2, B_4, B_5, P_1$  and  $P_2$  in Figure 12(b)) defines a contact point for which a corresponding contact force is calculated based on its associated penetration distance..

Every contact point has a corresponding point on one or both particles. For example, if the contact point is a vertex of polygons  $\alpha$  (such as point  $A_2$ ), then it is the projection of a corresponding vertex  $V$  of particle A on the CP; if the contact point is a vertex of polygon  $\beta$ , then it is the projection of a corresponding vertex of particle B on the CP; if the contact point is the intersection of one edge from  $\alpha$  and one edge from  $\beta$  (points  $P_1$  and  $P_2$ ), then it has corresponding points on both particles.

The penetration distance  $D_N$  of a contact point measures how much its corresponding point(s) has penetrated into the contact zone. If the contact corresponds to only one point, such as vertex  $V$  on particle A, then  $D_N$  is equal to the distance from  $V$  to the plane  $CP_A$  (Figure 12). If the contact point has corresponding points on both particles then  $D_N$  is the average value of the penetration distances of each corresponding point.

**3.3.3 Contact force and damping.** In BLOKS3D, a contact force is calculated at each contact point. In spring-dashpot contact models, normal and shear components of the contact force are divided into elastic and viscous damping parts:

$$\begin{aligned}\mathbf{F}_N &= \mathbf{F}_{EN} + \mathbf{F}_{DN} \\ \mathbf{F}_S &= \mathbf{F}_{ES} + \mathbf{F}_{DS}\end{aligned}\quad (4)$$

Note that in equation (4) and the rest of this section, variables in bold are vectors and variables in plain italic denote the corresponding magnitude of the vector. Elastic components account for the elastic deformation of the particle at locations of the contact. Damping components account for the energy loss due to inter-particle collisions. The total contact force is then obtained from  $\mathbf{F} = \mathbf{F}_N + \mathbf{F}_S$ .

The following equation is suggested to calculate the elastic component  $F_{EN}$  of the normal force:

$$F_{EN} = K_N D_N + K_{NN} D_N^b \quad (5)$$

Where parameters  $K_N$ ,  $K_{NN}$  and  $b$  are material constants. Setting  $K_{NN} = 0$  in equation (5) results in a typical linear force-displacement relationship. On the other hand,  $K_N = 0$  (with appropriate values for  $K_{NN}$  and  $b$ ) results in a Hertzian type of force-displacement relationship (Figure 13).

The elastic component of the shear force  $F_{ES}$  at time  $t$  is computed incrementally as:

$$\mathbf{F}_{ES}^t = (\mathbf{F}_{ES}^{t-\Delta t} - (\mathbf{F}_{ES}^{t-\Delta t} \cdot \mathbf{n}^t) \mathbf{n}^t) + K_S \Delta \mathbf{U}_S^t \quad (6)$$

Where superscripts denote time. A Coulomb-type friction law is enforced to limit the magnitude of  $F_{ES}$  in equation (6) to  $F_{N} \tan(\phi)$ . Any contact for which  $F_{ES}$  has reached its maximum value is defined as a “sliding contact”. Otherwise it is “non-sliding contact”.

In spring-dashpot models, normal and shear damping forces are computed as:

$$F_{DN} = \beta_d K_{LN} \dot{U}_N \quad (7a)$$

$$\mathbf{F}_{DS} = \beta_d K_S \dot{\mathbf{U}}_S \quad (7b)$$

According to equations (7a) and (7b), for high values of relative velocity, damping forces will become unreasonably large. For instance, consider a cubic particle sliding along a slope surface (Figure 14(a)). If the slope angle is larger than the friction angle between the particle and the slope, the velocity of the particle should keep on increasing with time. By using equation (7b) the magnitude of  $\mathbf{F}_{DS}$  will increase with increasing velocity, up to the point where the total shear force  $\mathbf{F}_S + \mathbf{F}_{DS}$  will balance the weight component along the slope. Beyond this point the particle will continue to move with a constant velocity. In order to avoid such unrealistic results the following cap is placed on the magnitude  $F_{DS}$ :

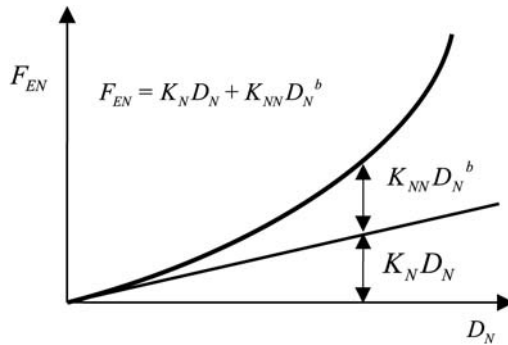
$$F_{DS} \leq d * F_{ES} \quad (8)$$

where  $d$  is a user-defined constant. Theoretically, the damping should have very little or no effect on the velocity of a sliding particle but should eliminate undesirable vibrations during a simulation. As a result, a “dual damping criterion” is suggested for variable  $d$ :

$$d = \begin{cases} d_1 & \text{sliding contacts} \\ d_2 & \text{non-sliding contacts} \end{cases} \quad (9)$$

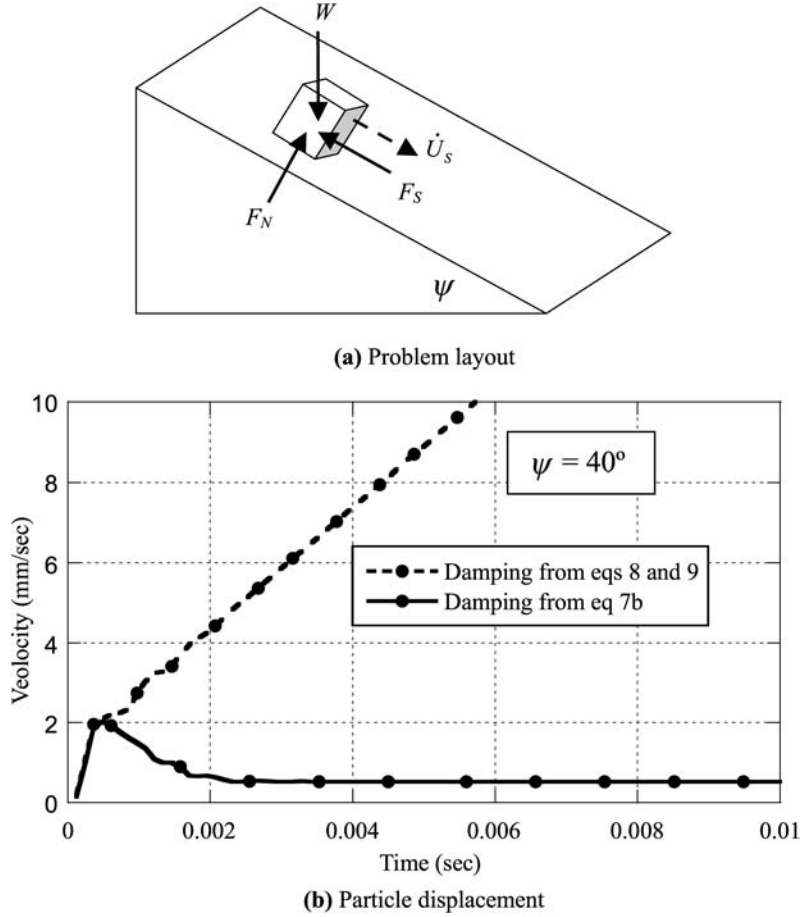
$d_1$  and  $d_2$  are user defined variables, whereby  $d_1$  is much smaller than  $d_2$ , such that the shear damping contact force decreases after sliding is initiated.

The sliding test of Figure 14(a) is simulated with damping model proposed in equations (8) and (9). The slope angle is  $40^\circ$  and particle/slope friction angle is  $35^\circ$ .  $d_1$  and  $d_2$  are chosen to be 0.01 and 0.9, respectively. The particle is released to move along the surface at time  $t = 0$  and its velocity is recorded afterwards. Figure 14(b) shows the velocity as a function of time. For comparison the result of the same test is shown using equation (7a). The typical dashpot model results in a constant velocity of the particle during sliding, while the velocity with dual damping criterion matches the theoretical value.



**Figure 13.**  
Relationship between  
normal contact force  $F_{EN}$   
and penetration distance  
 $D_N$  of the contact point





**Figure 14.**  
Sliding of a cubic particle

For similar reasons, the magnitude of normal damping force  $F_{DN}$  is limited by:

$$F_{DN} \leq d * F_{EN} \quad (10)$$

which is not affected by initiation of sliding.

The contact zone method and the multiple contact point scheme are proposed for large-scale simulations of granular material behavior, whereby the computational cost is a critical consideration. The definition of the contact zone and the computation of the contact forces using penetration distance is necessitated by the assumption that the particles are rigid. Using the contact zone method, in some situations, may result in local discontinuity in contact forces during the relative motion of a pair of particles, and in locally erroneous forces at a given time step. However, the potential errors are corrected in subsequent time steps due to global equilibrium considerations. The current approximation is considered acceptable and provides reasonable estimate of mass particle behavior.

#### 4. Simulation of direct shear test

The performance of BLOKS3D is verified by simulating a series of direct shear tests with different void ratios. Each test is performed using 800 particles as shown in Figure 15 with particle geometries shown in Figure 16. The average particle size is 0.005 m. Other parameters used for the simulation are listed below:

$$\phi = 35^\circ$$

$$K_N = 130 \text{ kN/m}$$

$$K_{NN} = 1.40 \times 10^6 \text{ kN/m}$$

$$b = 3$$

$$K_S = 102 \text{ kN/m}$$

$$d_1 = 0.01$$

$$d_2 = 0.90$$

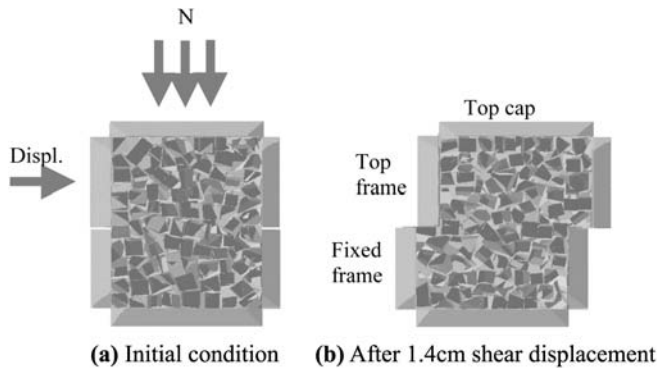
$$\alpha_d = 0$$

$$\beta_d = 0.02$$

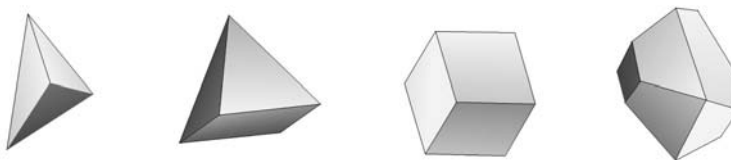
$$\Delta t = 0.0001216 \text{ s}$$

##### 4.1 Sample preparation and simulation process

All particles are generated in a space above the containers. Then the particles are allowed to fall into the frame while temporarily removing the top cap. This results in a loose assembly of particles with a high void ratio and uneven top surface. A lower void ratio is achieved by temporarily increasing the gravity constant  $g$  and at the same time reducing the friction angle  $\phi$  between particles. The resultant void ratio depends on how much  $g$  is increased and  $\phi$  is decreased. The higher the  $g$  or the lower the  $\phi$ , the



**Figure 15.**  
Direct shear test cross section



**Figure 16.**  
Particles with four vertices (tetrahedron), five vertices (pyramid), eight vertices (cube) and 14 vertices used in direct shear test simulation

denser is the resultant sample. Once the desired void ratio is obtained,  $g$  and  $\phi$  are set back to their original values and the particles above a certain elevation are removed to produce a flat surface before the top cap is placed. The cap can freely move in the vertical direction while its horizontal displacement is constraint by the top frame. Rotation of the frame, including the cap, is not allowed. A vertical force  $N = 1000\text{ N}$  is applied on the top cap and then the top frame and the top cap are moved together with a horizontal velocity of  $0.001\text{ m/sec}$ . Each test is continued up to a total shear displacement of  $0.014\text{ m}$ .

4.2 Simulation results

Three samples with void ratios of  $e = 0.47, 0.66$ , and  $0.75$  are sheared. Figure 17 plots the shear force and vertical displacement during shearing. The  $N \tan(\phi)$  line marks the shear force associated with only particle-particle friction. For  $e = 0.47$ , shear force in Figure 17(a) increases to a peak after which the force drops to a residual value. For samples with a higher void ratio, the shear force increases monotonically. For  $e = 0.47$ , the material experiences small contraction at the start of shearing, Figure 17(b), followed by significant dilation. The samples with a higher void ratio contract monotonically during shearing.

5. Conclusions

This paper presents the development of several algorithms for discrete element modeling of polyhedral particles. The algorithms include: motion integration scheme, contact resolution and detection algorithm, contact point and force determination and contact damping. These developments are proposed to increase computational efficiency for large-scale granular material simulation. The algorithms are implemented in the DEM program BLOKS3D. Several simulations are presented to demonstrate the performance of the proposed algorithms and the overall performance of BLOKS3D.

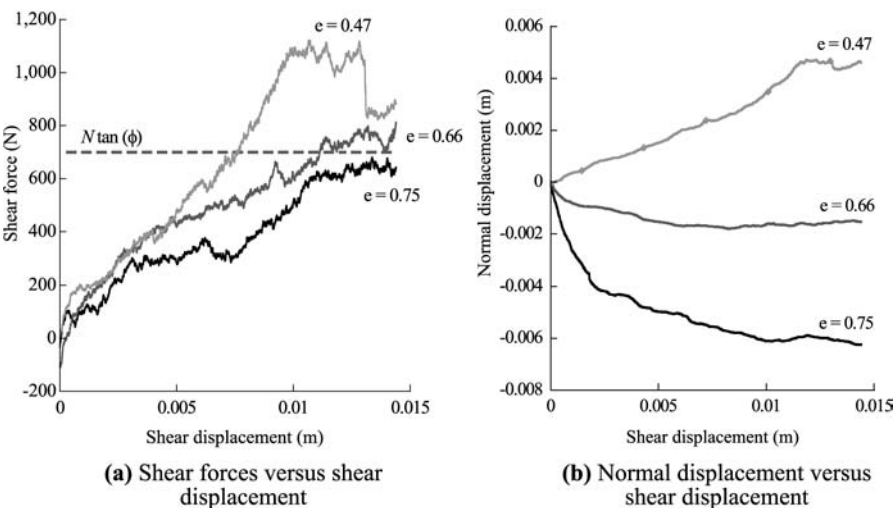


Figure 17.  
Direct shear test  
simulation with BLOKS3D

---

**References**

- Barbosa, R.E. (1990), "Discrete element models for granular materials and rock masses", PhD thesis.
- Campbell, C.S., Cleary, P.W. and Hopkins, M.A. (1995), "Large-scale landslide simulations: global deformation, velocities, and basal friction", *Journal of Geophysical Research – Solid Earth*, Vol. 100, B5, pp. 8267-83.
- Cleary, P.W. (2000), "DEM simulation of industrial particle flows: case studies of dragline excavators, mixing in tumblers and centrifugal mills", *Powder Technology*, Vol. 209 Nos 1-3, pp. 83-104.
- Cleary, P.W. and Campbell, C.S. (1993), "Self-lubrication for long run-out landslides: examination by computer simulation", *Journal of Geophysical Research – Solid Earth*, Vol. 98, B12, pp. 21911-24.
- Coutinho, M.G. (2001), *Dynamic Simulations of Multibody Systems*, Springer-Verlag, New York, NY.
- Cundall, P.A. (1971), "A computer model for simulating progressive large-scale movements in block rock mechanics", *Proc. Symp. Int. Soc. Rock Mech. Nancy*, p. 2.
- Cundall, P.A. (1988a), "Formulation of a three-dimensional distinct element model – Part I: a scheme to detect and represent contacts in a system composed of many polyhedral blocks", *Int. J. of Rock Mech., Min. Sci. & Geomech. Abstr.*, Vol. 25 No. 3, pp. 107-16.
- Cundall, P.A. (1988b), "Formulation of a three-dimensional distinct element model-Part II: mechanical calculations for motion and interaction of a system composed of many polyhedral blocks", *Int. J. of Rock Mech., Min. Sci. & Geomech. Abstr.*, Vol. 25 No. 3, pp. 117-25.
- Cundall, P.A. and Strack, O.D.L. (1979), "A discrete numerical model for granular assemblies", *Geotechnique*, Vol. 29 No. 1, pp. 47-65.
- Ghaboussi, J. and Barbosa, R. (1990), "Three-dimensional discrete element method for granular materials", *International Journal for Numerical and Analytical Methods in Geomechanics*, Vol. 14, pp. 451-72.
- Hemami, A. (1995), "Fundamental analysis of automatic excavation", *Journal of Aerospace Engineering (ASCE)*, Vol. 8 No. 4, pp. 175-9.
- Hopkins, M.A., Hibler, W.D. and Flato, G.M. (1991), "On the numerical simulation of the sea ice ridging process", *Journal of Geophysical Research-Oceans*, Vol. 96, C3, pp. 4809-20.
- Johnson, S., Williams, J.R. and Cook, B. (2005), "Experimental and numerical investigation of pharmaceutical powder blending", paper presented at 8th US National Conference on Computational Mechanics, Austin, TX.
- Lin, X. and NG, T.T. (1997), "A three-dimensional discrete element model using arrays of ellipsoids", *Geotechnique*, Vol. 47 No. 2, pp. 319-29.
- Matuttis, H.G., Luding, S. and Herrmann, H.J. (2000), "Discrete element simulations of dense packings and heaps made of spherical and non-spherical particles", *Powder Technology*, No. 109, pp. 278-92.
- Messner, A.M. and Taylor, G.Q. (1980), "Solid polyhedron measures", *ACM Transactions on Mathematical software*, Vol. 6 No. 1, pp. 121-30.
- Moakher, M., Shinbrot, T. and Muzzio, F.J. (2000), "Experimentally validated computations of flow, mixing and segregation of non-cohesive grains in 3D tumbling blenders", *Powder Technology*, Vol. 109 Nos 1/3, pp. 58-71.
- Munjiza, A. and Andrews, K.R.F. (1998), "NBS contact detection algorithm for bodies of similar size", *Int. J. of Num. meth. Engng.*, Vol. 43 No. 1, pp. 131-49.
- Nezami, E., Hashash, Y.M.A., Zhao, D. and Ghaboussi, J. (2004), "A fast contact detection algorithm for 3D discrete element method", *Computers and Geotechnics*, Vol. 31, pp. 575-87.

- Nezami, E.G., Hashash, Y.M.A., Zhao, D. and Ghaboussi, J. (2006), "Shortest link method for contact detection in discrete element method", *International Journal of Numerical and Analytical Methods in Geomechanics*, Vol. 30 No. 8, pp. 783-801.
- Ng, T.-T. (1992), "Numerical simulations of granular soils using elliptical particles", paper presented at Microstructural Characterization in Constitutive Modeling of Metals and Granular Media, ASME Summer Mechanics and Materials Conference, Tempe, AR, pp. 95-118.
- Perkins, E. and Williams, J.R. (2001), "A fast contact detection algorithm insensitive to object size", *Engineering Computations*, Vol. 18 Nos 1/2, pp. 48-61.
- Singh, S. (1997), "State of the art in automation of earthmoving", *Journal of Aerospace Engineering (ASCE)*, Vol. 10 No. 4, pp. 179-88.
- Strack, O.D.L. and Cundall, P.A. (1978), "The distinct element method as a tool for research in granular media", Part I, Report to NSF, Department of Civil and Mineral Engineering, University of Minnesota, Minneapolis, MN.
- Strack, O.D.L. and Cundall, P.A. (1984), "Fundamental studies of fabric in granular materials", report to NSF, University of Minnesota, Minneapolis, MN.
- Toussaint, G.T. (1985), "A simple linear algorithm for intersecting convex polygons", *The Visual Computer*, Vol. 1, pp. 118-23.
- Tutumluer, E., Huang, H., Hashash, Y.M.A. and Ghaboussi, J. (2006), "Aggregate shape effects on ballast tamping and railroad track lateral stability", paper presented at American Railway Engineering and Maintenance of Way Association (AREMA) Annual Conference, Louisville, KY, September 17-20.
- Yang, X.S., Lewis, R.W., Gethin, D.T., Ransing, R.S. and Rowe, R.C. (2002), "Discrete-finite element modeling of pharmaceutical powder computation: a two-contact detection algorithm for non-spherical particles", paper presented at Third International Conference on Discrete Element Methods: Numerical Modeling of Discontinua, ASCE/Geo Institute, Santa Fe, NM, pp. 74-8.

#### Appendix. Determination of the principal moment of inertia

In any coordinate system  $xyz$ , the moment of inertia tensor  $\mathbf{I}$  can be expressed as:

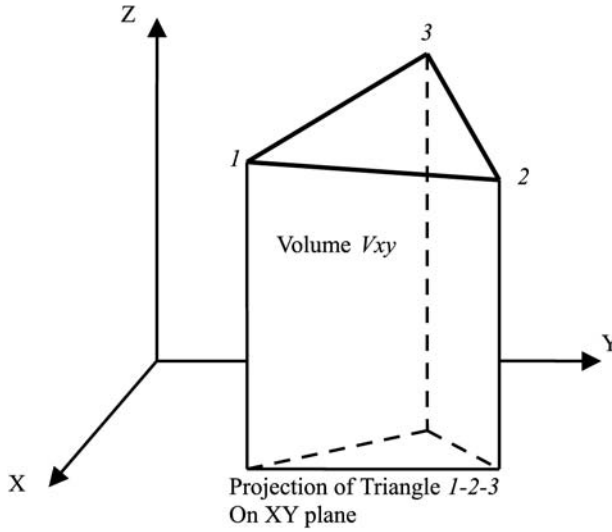
$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \quad (\text{A1})$$

where

$$\begin{aligned} I_{xx} &= \int_V (y^2 + z^2) dm, & I_{yy} &= \int_V (z^2 + x^2) dm, & I_{zz} &= \int_V (x^2 + y^2) dm, \\ I_{xy} &= - \int_V xy dm, & I_{yz} &= - \int_V yz dm, & I_{zx} &= - \int_V xz dm \end{aligned} \quad (\text{A2})$$

The integrations are taken over the whole mass of the particle.

For polyhedral particles, a closed form solution based on triangulation of faces can be derived for integral values of equation A2 as follows. Each face of the particle is triangulated where each triangle (such as  $i-j-k$  in Figure A1) and its projections on coordinate planes form three polyhedral volumes (only one polyhedral volume is shown in Figure A1). Contribution of each polyhedral volume into the integral values of equation A2 is calculated separately. The results are then



**Figure A1.**  
Calculation of moment of  
inertia

accumulated over all triangles of all faces to get total values. (Messner and Taylor, 1980) have used a numerical integration for each polyhedral volume using four-point Gaussian quadrature over the triangles. In this paper, a closed form integration is derived using MATLAB Symbolic Toolbox as follows:

Assume that:

$$X_2 = \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix}, \quad X_3 = \begin{Bmatrix} x_3 \\ y_3 \\ z_3 \end{Bmatrix} \quad \text{and} \quad X_1 = \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}$$

denote the coordinates for the three vertices 1-2-3. Furthermore, assume that 1-2-3 are arranged in such an order that the direction given by the right hand rule always points outward from the particle. Define:

$$\begin{aligned} \text{Narea}_z &= \frac{\sum_k \varepsilon_{ijk} x_i y_j}{2}, \quad \text{Hns}_z = \frac{\left( \sum_i z_i \right) \left( \sum_i z_i^2 \right) + \Pi z_i}{30}, \quad n_1 = |\varepsilon_{ijk}| x_i y_j z_k, \\ n_2 &= \sum_i x_i y_i z_i, \quad n_3 = \sum_i \sum_j \sum_k x_i y_j z_k - n_1 - n_2, \quad \text{Hnt} = \frac{n_1}{60} + \frac{n_2}{10} + \frac{n_3}{30} \end{aligned} \quad (\text{A3})$$

$\varepsilon_{ijk}$  is the permutation function over  $ijk$ .  $\Pi$  implies multiplication. All indices range from 1 to 3. Einstein summation convention for repeated indices is used.  $\text{Narea}_x$ ,  $\text{Narea}_y$ ,  $\text{Hns}_x$ ,  $\text{Hns}_y$  can be obtained by cyclic permutation  $x \Rightarrow y \Rightarrow z \Rightarrow x$  in the first two equations. If the particle's centroid is chosen as the origin of the coordinate system then it can be shown that:

$$\begin{aligned} \int_{V_{xy}} z^2 dm &= \rho^* \text{Narea}_z^* \text{Hns}_z, \quad \int_{V_{xy}} xy dm = \rho^* \text{Narea}_z^* \text{Hnt} \\ \int_{V_{yz}} x^2 dm &= \rho^* \text{Narea}_x^* \text{Hns}_x, \quad \int_{V_{yz}} yz dm = \rho^* \text{Narea}_x^* \text{Hnt} \\ \int_{V_{zx}} y^2 dm &= \rho^* \text{Narea}_y^* \text{Hns}_y, \quad \int_{V_{zx}} xz dm = \rho^* \text{Narea}_y^* \text{Hnt} \end{aligned} \quad (\text{A4})$$

and

$$\begin{aligned}
I_{xx} &= \sum_p \sum_q \left( \int_{V_{zx}} y^2 dm + \int_{V_{xy}} z^2 dm \right), & I_{yz} &= - \sum_p \sum_q \int_{V_{yz}} yz dm \\
I_{yy} &= \sum_p \sum_q \left( \int_{V_{xy}} z^2 dm + \int_{V_{yz}} x^2 dm \right), & I_{zx} &= - \sum_p \sum_q \int_{V_{zx}} zx dm \\
I_{zz} &= \sum_p \sum_q \left( \int_{V_{yz}} x^2 dm + \int_{V_{zx}} y^2 dm \right), & I_{xy} &= - \sum_p \sum_q \int_{V_{xy}} xy dm
\end{aligned} \tag{A5}$$

In equation A5, the summation  $p$  is taken over all the faces of the particle, and summation  $q$  is taken over all the triangles of each face.

After the moment of inertial tensor  $\mathbf{I}$  is obtained, the eigenvalue problem  $\mathbf{I}\Phi = \lambda\Phi$  should be solved. Eigenvalues  $\lambda$  define the principal moments of inertia and eigenvectors  $\Phi$  define the directions of principal axes.

#### Corresponding author

Youssef M.A. Hashash can be contacted at: [hashash@uiuc.edu](mailto:hashash@uiuc.edu)