

# A fast contact detection algorithm for 3-D discrete element method

Erfan G. Nezami, Youssef M.A. Hashash<sup>\*</sup>, Dawei Zhao, Jamshid Ghaboussi

*2230 Newmark Civil Engineering Laboratory, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Avenue, Urbana, IL 61801, USA*

Received 8 December 2003; received in revised form 21 July 2004; accepted 11 August 2004

## Abstract

In the discrete element method, determining the contact points between interacting particles and the associated contact normals at each time step is a critically important and time consuming calculation. Common-plane (CP) algorithm is one of the more effective methods for contact detection when dealing with two-dimensional polygonal or three-dimensional polyhedral particles. A new approach, called fast common plane (FCP) method, is proposed to find the common plane between polygonal particles. FCP approach recognizes that a common plane has identifying characteristics, which dramatically reduce the search space for the common plane. In two-dimensions, the CP is found by checking only 5 possible candidate planes. In three-dimensions, the candidate planes fall within 4 types related to the geometry of the particles and their relative positions. Numerical experiments reveal that in three dimensions FCP algorithm can be up to 40 times faster than available search methods for finding the common-plane.

© 2004 Elsevier Ltd. All rights reserved.

## 1. Introduction

The idea of using discrete particles in numerical simulations was introduced in the work of Born and Huang [1] and Maradudin [2], where atoms are described via the concentrated mass and contact force of interacting atoms. Eisenstadt [3] models bonds between atoms by springs. Cundall [4] introduces the discrete element method (DEM) to simulate large deformations in jointed rock formations. Cundall and Strack [5] extend DEM to analyze assemblies of idealized granular particles composed of circular disks and spheres.

During the past two decades, DEM has proved to be a reliable tool to study the behavior of granular materials in both micro- and macro-scale. DEM simulations have been used in large scale geophysical applications such as landslides [6,7] and ice flows [8], as well as many

industrial and mining applications including dragline excavation [9], mixing in tumblers, [10,11], and silo filling [11].

At the micro-level, DEM is used to investigate distribution and evolution of micro parameters (such as contact normals and contact forces), and their relation to macro-parameters (such as stress and strain) [12,13]. The numerical results are in good agreement with micro-mechanical observations from experimental tests on natural sands [14,15], idealized photo-elastic sensitive rods [16,17], and spherical glasses [18].

The complexity of the behavior of granular materials and substantial computational time and effort required for DEM computations, however, had limited most of DEM simulations prior to 1990 to assemblies of circular disks or spheres. Examples are the DEM codes BALL [19] and TRUBAL [20]. Lin and NG [21] provide an extensive list of available circular or spherical DEM codes. In most soil-mechanics related applications, the assumption of spherical particles fails to capture essential aspects of mechanical behavior of the particulate material.

<sup>\*</sup> Corresponding author. Tel.: +1 217 333 6986; fax: +1 217 333 9464.

E-mail address: [hashash@uiuc.edu](mailto:hashash@uiuc.edu) (Y.M.A. Hashash).

As a result, during the last decade, many efforts have been made to incorporate non-spherical particles in DEM implementation. This includes ellipses [22,23], connected circular segments [24], super-quadratics [25], polygons [26], ellipsoids [21], polyhedrons [26,27].

## 2. Contact detection schemes in DEM

Regardless of rapid advances in computer hardware and parallel computation techniques, the huge computational time and effort required to calculate and update contact forces are still a major hindering factor in large scale DEM simulations. For complex particle geometries, such as three-dimensional polyhedrons, contact detection subroutines can easily take up to 80% of the total analysis time. Applicability of a DEM code is directly related to the efficiency of the employed contact detection scheme.

Contact detection in DEM is usually performed in two independent stages. The first stage, referred to as neighbor search, is merely a rough search that aims to provide a list of all possible particles in contact. Among available algorithms for neighbor searching, the most recent ones include No Binary Search (NBS) contact detection algorithm [28] and DESS algorithm [29]. A review of neighbor search methods is available in [30].

In the second stage, called geometric resolution, pairs of contacting particles obtained from the first stage are examined in more detail to find the contact points (or contact area if distributed contact forces are considered) and calculate the contact forces. Geometric resolution algorithms strongly depend on complexity of the geometric representation of particles. For example, if the boundaries of the particles are implicitly represented by a single function  $f(x,y,z) = 0$ , then a closed form solution is likely to be available (for example see [5] for contacts between disks and spheres [22], for two-dimensional ellipses, and [21] for three-dimensional ellipsoids). Efficiency of these contact detection schemes are mostly controlled by the simplicity of the resulting equations.

Where the boundary cannot be represented by a single function  $f(x,y,z) = 0$ , such as in polygons or polyhedrons, the contact detection can be quite cumbersome. Barbosa [26] introduces a simple algorithm for contact detection between polyhedrons that requires comparing all the vertices of one particle to all faces of the other one and vice versa. The algorithm has a high computational complexity of order  $O(N^2)$ , with  $N$  being the number of vertices. Williams and O'Connor [30] introduce Discrete Function Representation algorithm, DFR, which achieves a computational complexity of order  $O(N)$ . In DFR, the contact between particles is calculated by considering the interaction between the bounding boxes of particles. Krishnasamy and Jakiela [31] and later Feng and Owen [32] introduce energy-based meth-

ods for finding the contact forces, in which a potential energy function is defined for each contact as a function of the overlap area. Cundall [33] introduces the well-known class of “Common-Plane” (CP) methods: “A common plane is a plane that, in some sense, bisects the space between the two contacting particles”. If the two particles are in contact, then both will intersect the CP, and if they are not in contact, then neither intersects the CP. As a result of using CP, the expensive particle-to-particle contact detection problem reduces to a much faster plane-to-particle contact detection problem. Once the CP is established between two particles, the normal to the CP defines the direction of the contact normal, which in turn defines the direction of the normal contact force between the two particles. This is especially advantageous for vertex-to-vertex or edge-to-vertex contacts, where the definition of the contact normal is a non-trivial problem. The method has a complexity of order  $O(N)$  and has been successfully implemented in three-dimensional DEM code 3DEC [27].

The following sections introduce a new approach to obtain the CP for polygonal (2-D) and polyhedral (3-D) particles. Determination of contact points and contact forces are not discussed. Particles are assumed to be convex, rigid or deformable, while concave particles can be modeled as a combination of several convex particles attached to each other.

## 3. Definition of the common plane

The CP is identified by its unit vector normal,  $\mathbf{n}_{[h1]}$ , and any point  $V_0$  on it as shown in Fig. 1. For any point  $V$  in the space, the “distance”  $d^V$  of that point to any arbitrary plane in the space is defined as

$$d^V = \mathbf{n} \cdot (V_0 - V), \quad (1)$$

whereby,  $\mathbf{n}$  is the unit vector normal to the plane and  $V_0$  is any point on that plane. Both  $V$  and  $V_0$  are described

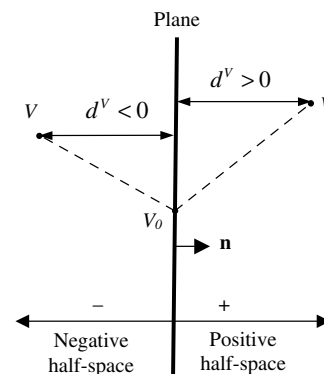


Fig. 1. Definition of distances and sign convention of a point to a plane.

in a global Cartesian coordinate system. Eq. (1) divides the space into positive and negative half-spaces, with points in positive half-space have positive distances and points in negative half-space have negative distances to the plane. For any polygonal or polyhedral particle **A** the “distance”  $d_A$  of the particle to any plane in the space is defined as

$$d_A = \begin{cases} \max(d_A^V) & \text{if } d_A^C < 0 \\ \min(d_A^V) & \text{if } d_A^C > 0 \end{cases}, \quad (2)$$

where  $d_{A[h2]}^V$  is the distance of a vertex  $V$  on the particle to the plane (Eq. (1)), and  $\min\{\cdot\}$  and  $\max\{\cdot\}$  represent minimum and maximum values, respectively, taken over all vertices of the particle.  $d_A^C$  is the distance of the centroid of the particle to the plane. If a face of the particle is parallel to the plane then more than vertex can define the distance  $d_A \cdot d_A^C = O_{[h3]}$  is of no practical interest as the CP will never pass through centroid of a particle. Subscripts and superscripts in all equations denote particles and vertices (points), respectively. The vertex (or each of the vertices) that define the distance in (2) is called the “closest vertex” of that particle to the plane.

For any two particles **A** and **B**, a CP is the plane which meets the following three conditions:

Condition 1.

Centroids of particles **A** and **B** are located on opposite sides of the CP. In this paper it is assumed that the centroid of particle **A** is located on the negative side and that of particle **B** in the positive side of CP, Fig. 2.

Condition 2.

The gap, defined as  $d_B - d_A$ , is a maximum.

Condition 3.

$$d_A = -d_B,$$

$d_A$  and  $d_B$  are the distances of particles **A** and **B**.

Condition 1 guarantees that the CP is between the particles. The gap  $d_B - d_A$  is only a function of direction  $\mathbf{n}$  of the CP and is independent of the location of the plane in space. Consequently, Condition 2 identifies the direction  $\mathbf{n}$  by maximizing the gap. Condition 3 specifies the location of the CP by setting  $d_A = -d_B$ . For separated particles the gap is always positive ( $d_A < 0$  and  $d_B > 0$ ), while for particles in contact the gap is always negative ( $d_A > 0$  and  $d_B < 0$ ).

Whenever  $d_B - d_A > \text{TOL}$ , where TOL is a small positive user-defined tolerance, then the particles are recognized as not in contact, no CP is developed. A “potential contact” is a contact for which  $0 < d_B - d_A < \text{TOL}$  the particles are likely to develop new contacts in the next few time steps. A real contact is a contact for which  $d_B - d_A < 0$ . Contact points and forces are found only for real contacts.

#### 4. Conventional algorithm for finding the CP

Cundall [33] suggests a two-stage procedure for finding the CP: the first stage specifies one point on the CP (referred to as the reference point, point  $M$  in Fig. 3). The second stage is an iterative process, in which a normal vector  $\mathbf{n}$ , corresponding to the maximum gap, is found by rotating the CP around the reference point. In two-dimensions, the CP is a line and the rotation is performed around the reference point  $M$ . In three-dimensions, two arbitrary orthogonal axes are chosen in the CP with their origin at the reference point. The CP is then perturbed around each of them in both negative and positive sense. If any perturbation produces a gap larger than that of the current CP, the new CP replaces the current one. In this case, the closest vertices and the reference point is updated based on the newly found CP [34]. If all the perturbations produce smaller

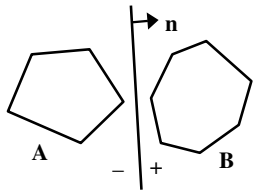
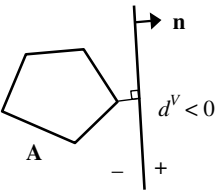
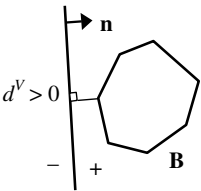
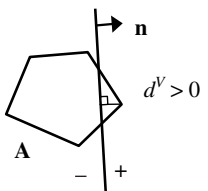
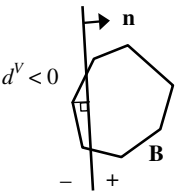
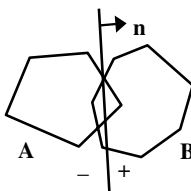
	Two particles and the CP	Distance of the particles to the CP	
Separated particles			
			
In-contact particles			

Fig. 2. The CP and its distance to the particles.

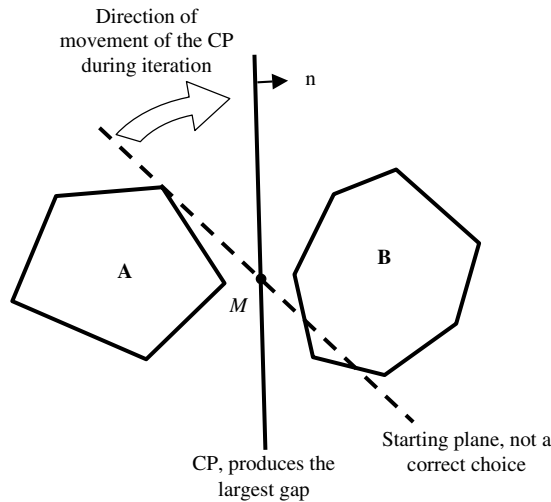


Fig. 3. Conventional iterative procedure for finding the CP.

gaps than that of the current CP, the next iteration starts with a smaller perturbation. The iteration process starts from an initial guess (either the CP from the previous time step or the perpendicular bisector of the line that connects the centroids of the particles), and continues until the direction of the CP is found with reasonable accuracy. At any stage of iteration, if the gap exceeds a positive tolerance TOL then the iterative process halts and the contact is deleted. A gap larger than TOL indicates that the particles are too far from each other to make a contact. The total number of iterations depends on the accuracy of the initial guess of the CP. In general, the algorithm requires a large number of iteration steps. The number of iteration steps is especially high for the first-time formation of the CP, where the initial guess and the actual CP are very different.

## 5. Fast identification of common plane candidates

When two particles are not in contact, the definition of the CP can be utilized to limit the number of candidate common planes and thus significantly reduce the computational cost of common plane selection.

### 5.1. CP identification in 2-D

**Statement:** In two-dimensions, the CP can be found by checking only 5 possible candidate planes.

The following provides a stepwise proof of the statement above leading to identification of the 5 possible candidate planes.

**Proof.** Let *A* and *B* be the closest vertices for two not-in-contact particles **A** and **B**, respectively.

(i) *The CP passes through the midpoint M of segment AB.*

Let  $\theta$  measure the angle between the CP and the perpendicular bisector (PB) of the segment *AB*, as shown in Fig. 4. Then

$$|d_A| = |MA| \cos \theta \quad \text{and} \quad |d_B| = |MA| \cos \theta_{[h4]}.$$

The Condition 3 (Section 3) of CP definition, ( $d_A = -d_B$ ), implies that  $|d_A| = |d_B|$  or

$$|MA| \cos \theta = |MB| \cos \theta \Rightarrow |MA| = |MB|.$$

$\therefore$  CP should pass through the midpoint *M* of segment *AB*.

(ii) *CP is completely located within the space S*

Space *S* is the area formed by rays *Mm*<sub>1</sub>, *Mm*<sub>2</sub>, *Mm*<sub>3</sub> and *Mm*<sub>4</sub>, drawn from the midpoint *M*, parallel to edges *AA*<sub>1</sub>, *BB*<sub>1</sub>, *AA*<sub>2</sub>, and *BB*<sub>2</sub>, respectively (the shaded area in Fig. 5) (A ray is the part of a straight line beginning at a given point and extending limitlessly in one direction).

Assume that line *L*, portion of which is located outside space *S*, is a candidate common plane (Fig. 5). Then, vertex *B*<sub>1</sub> is closer to this line than vertex *B*. This implies that *AB*<sub>1</sub> and not *AB* are the closest vertices. This contradicts *AB* being the closest vertices and the geometric arrangement of the particles. Therefore, line *L* cannot be a candidate common plane. Similarly all lines located partially or completely outside space *S* cannot be candidate common planes.

$\therefore$  The common plane is completely located within space *S*.

(iii) *The CP should produce the smallest angle with the PB of the line AB*

$$\text{From Fig. 4, } d_B - d_A = |AB| \cdot \cos \theta$$

$$\text{From Condition 2, } d_B - d_A \text{ is maximum}$$

$$\Rightarrow \cos \theta \text{ is maximum}$$

$$\therefore \text{ angle } \theta \text{ is minimum.}$$

(iv) *The CP is one of five candidate planes*

The CP is the line that

- is completely located in space *S* from proof (ii), and

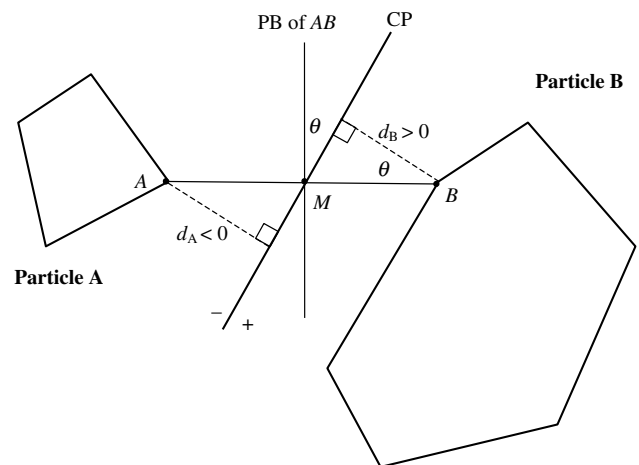


Fig. 4. PB of segment *AB* and the CP in 2-D; CP passes through the midpoint *M* of segment *AB*.

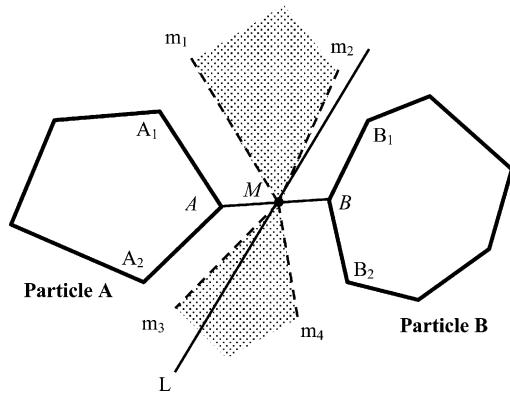


Fig. 5. CP is located inside space  $S$  defined in 2-D by rays  $Mm_1$ ,  $Mm_2$ ,  $Mm_3$  and  $Mm_4$ .

- makes the smallest possible angle with the PB of  $AB$  from proof (iii).

If the PB of the segment  $AB$ , is completely located in space  $S$  (Fig. 6(a)), then:

From proof (iii) the line that makes the smallest possible angle  $\theta$  with PB is the PB itself. The PB also satisfies proof (ii),

$\therefore$  The common plane is the PB (type a below).

If the PB is not located completely inside the space  $S$  (Fig. 6(b)), then:

- The PB is not the common plane.

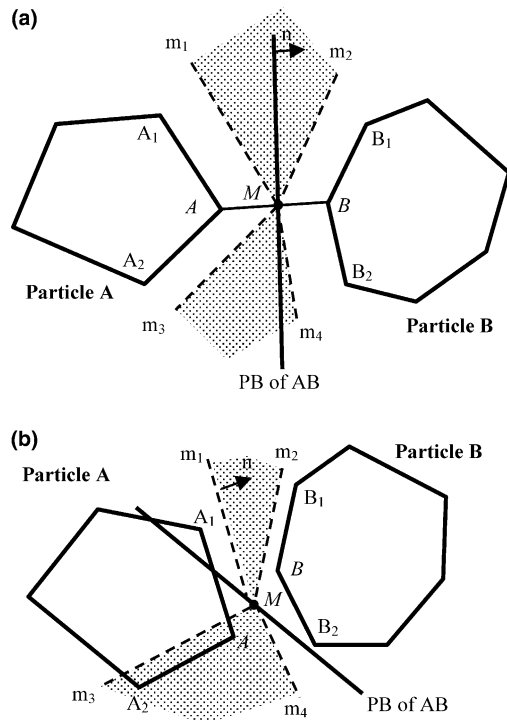


Fig. 6. (a) Perpendicular bisector of  $AB$  is outside Space  $S$ , so it is the CP; (b) perpendicular bisector of  $AB$  is outside Space  $S$ , the line  $MM_1$  is the CP.

- The common plane is the line with the smallest possible angle  $\theta$  to the PB (proof iii).

$\therefore$  The common plane is one of the boundary rays  $Mm_1$ ,  $Mm_2$ ,  $Mm_3$  or  $Mm_4$  ( $Mm_1$  in the figure, type b below). Any line inside space  $S$  other than on the boundaries will make a larger angle  $\theta$ .

Note that in Fig. 6(a) candidate common planes superimposed to rays  $Mm_1$  and  $Mm_2$  will partly be outside space  $S$  and can be eliminated as candidate planes as per proof (ii). However, from an algorithmic/practical numerical implementation point of view it is easier to maintain these as candidate planes and eliminate them using check Conditions (1) & (3) as defined in Section 3

$\therefore$  the CP is one of the following candidates:

Type a: The perpendicular bisector of segment  $AB$ .

Type b: The lines passing through the mid-point of segment  $AB$  and parallel to edges  $AA_1$ ,  $AA_2$  of particle A, or parallel to edges  $BB_1$  and  $BB_2$  of particle B.

The number of candidate planes is limited to five.  $\square$

## 5.2. CP identification in 3-D

**Statement:** In three-dimensions, the candidate planes fall within 4 types related to the geometry of the particles and their relative positions.

**Proof.** Let  $A$  and  $B$  be the closest vertices for two not-in-contact particles A and B, respectively.

(i) The CP passes through the midpoint  $M$  of segment  $AB$

Similar to the 2-D case, the CP should pass through the midpoint  $M$ . Note that in 3-D, PB and CP are both planes, rather than lines, and the angle  $\theta$  measures the dihedral angle between PB and CP (Fig. 7). The dihedral angle is the angle made by two perpendiculars to the intersection line of two planes, one in each plane.

$\therefore$  The CP passes through the midpoint  $M$  of segment  $AB$ .

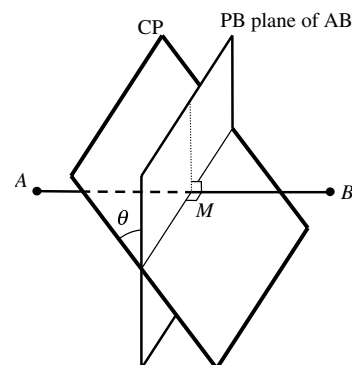


Fig. 7. PB plane of segment  $AB$ , CP, and dihedral angle  $\theta$  in 3-D.



(ii) *The CP completely located within the space S*

Rays  $Mm_1$  and  $Mm_2$  drawn parallel to edges  $AA_1$  and  $AA_2$ , respectively, define a semi-infinite quarter-plane  $m_1Mm_2$ , parallel to face  $A_1AA_2$  of particle A. (Fig. 8(a)). In the same way, for every face of particle A that shares vertex A, and for every face of particle B that shares vertex B, a quarter-plane can be constructed, passing through the midpoint M, parallel to that face (Fig. 8(b)). The space bounded between the quarter-planes associated with particle A from one side, and those associated with particle B from the other side, defines the space S in 3-D.

Similar to 2-D, any candidate common plane should be completely located inside the space S. Assume that plane P, portion of which is located outside space S, is a candidate common plane (Fig. 9). Then, vertex  $B_1$  is closer to this plane than vertex B, which contradicts AB being the closest vertices.

$\therefore$  The CP completely located within the space S.

(iii) *The CP should produce the smallest dihedral angle with the PB of segment AB*

$d_B - d_A = |AB| \cdot \cos\theta$  still holds in 3-D. From Condition 2,

$\therefore d_B - d_A$  is maximum when dihedral angle  $\theta$  is minimum.

(iv) *The CP is one of four different types*

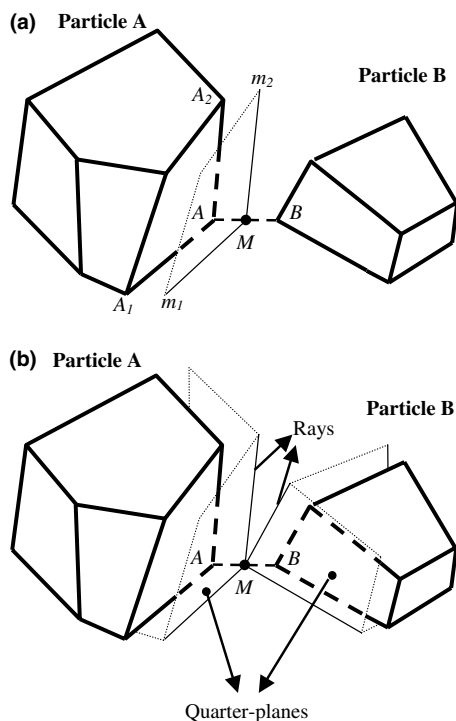


Fig. 8. (a) Rays  $Mm_1$  and  $Mm_2$  and quarter-plane  $m_1Mm_2$ ; (b) space S constructed by quarter-planes parallel to faces of particles A and B; (c) plane P cannot be a CP candidate as it is partially located outside the space S.

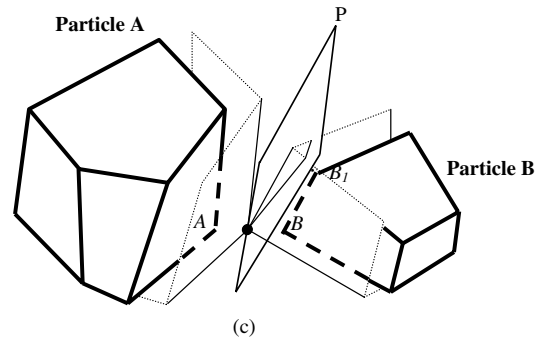


Fig. 9. Plane P cannot be a CP candidate as it is partially located outside the space S.

The CP is the plane that

- is completely located in space S from proof (ii), and
- makes the smallest possible dihedral angle with the PB plane of AB from proof (iii).

If the PB plane of segment AB, lies completely in space S, then:

From proof (iii) the plane that makes the smallest possible dihedral angle  $\theta$  with PB is the PB itself. The PB also satisfies proof (ii)

$\therefore$  The common plane is the PB (type a below).

If the PB plane is not completely located inside the space S, then:

- The PB is not the common plane.
- The common plane is the plane with the smallest possible angle  $\theta$  to the PB (proof iii).

$\therefore$  The common plane contains at least one ray from the boundary. Any plane which is completely inside the space S and does not contain any of the boundary rays, makes a larger dihedral angle  $\theta$  with PB and cannot be a candidate common plane. The number of rays included in CP can be used to further categorize it:

- The CP contains exactly two boundary rays. Then: If those two rays correspond to the same particle, then the CP contains the quarter-plane made by those rays. Therefore it is parallel to one of the faces of the particles (type b). If those two rays correspond to different particles, then the CP is parallel to corresponding edges from different particles (type c).
- The CP contains exactly one boundary ray. In this case, the CP is parallel to the corresponding particle edge (type d).
- The CP contains more than two boundary rays. Then any two of them can be utilized to identify the CP (the result is either type b or type c. For example, for the specific geometry shown in Fig. 10 whereby face  $B_1BB_2$  is parallel to the edge  $AA_1$ , the CP is parallel to three edges  $BB_1$ ,  $BB_2$  and  $AA_1$ . So it can be

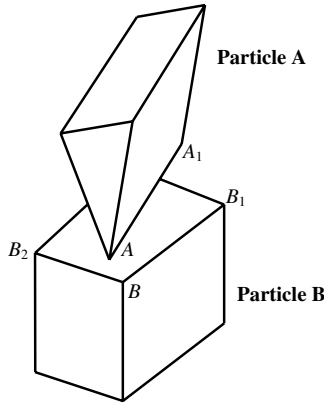


Fig. 10. Face  $B_1BB_2$  is parallel to edge  $AA_1$ . CP can be identified either as type b or type c.

either identified as type b (if  $BB_1$  and  $BB_2$  are chosen) or type c (if  $BB_1$  and  $AA_1$ , or  $BB_2$  and  $AA_1$  are chosen). In any case, the CP is the plane passing through the midpoint of  $AB$ , parallel to the face  $B_1BB_2$  and edge  $AA_1$ .

∴ the only possible candidates for the CP are as follows:

*Type a:* The PB plane of segment  $AB$ .

*Type b:* The plane passing through the midpoint of segment  $AB$  parallel to one of the faces of particles **A** or **B**. For particle **A**, only faces which include the vertex  $A$  are considered. For particle **B**, only faces which include the vertex  $B$  are considered.

*Type c:* The plane passing through the midpoint of segment  $AB$  parallel to one edge from particle **A** and one edge from particle **B**. For particle **A**, only those edges which share the vertex  $A$  are considered. For particle **B**, only those edges which share the vertex  $B$  are considered.

*Type d:* The plane passing through the midpoint of segment  $AB$  parallel to one edge from one of the particles. The plane can be fully defined by using the Conditions of the common plane described in Section 3.

Fig. 11 shows the CP containing only one ray ( $Mm_1$ ), parallel to the edge  $BB_1$  of block **B**. Condition 2 states

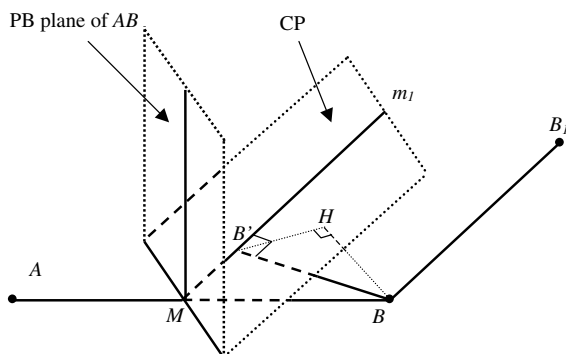


Fig. 11. Type d, CP is parallel to edge  $BB_1$ .

that  $d_B - d_A$  is a maximum, while Condition 3 states that  $d_A = -d_B$ . By substituting Condition 3 into Condition 2:

$$\Rightarrow 2d_B \text{ is a maximum,}$$

$$\Rightarrow d_B \text{ is a maximum.}$$

For this condition to be satisfied, the distance of particle **B**,  $d_B$ , to CP is defined by a line segment  $BB'$  perpendicular to ray  $Mm_1$ . By definition  $BB'$  defines the normal to CP and therefore the CP is fully defined.

This can be proved as follows: suppose that  $BB'$  is not the normal to CP, and  $BH$  is normal to CP ( $d_B = |BH|$ ).  $BHB'$  forms a right triangle whose hypotenuse is  $BB'$  and thus  $|BB'| \geq |BH|$ . It is therefore possible to find another plane passing through  $Mm_1$  with a larger distance  $d_B = |BB'|$  that maximizes  $d_B$ . That is the common plane.

The normal to CP can be found as follows:

$BB'$  (unit vector  $\mathbf{n}_{BB'}$ ) is perpendicular to ray  $Mm_1$  and therefore, edge  $BB_1$  (unit vector  $\mathbf{n}_{BB_1}$ ), Fig. 11

$$\mathbf{n}_{BB'} \perp \mathbf{n}_{BB_1}. \quad (3a)$$

Let  $\mathbf{n}_{AB}$  be the unit vector along segment  $AB$ . As  $BB'$ ,  $AB$  and  $BB_1$  are in the same plane,  $BB'$  is perpendicular to  $\mathbf{n}_{BB_1} \otimes \mathbf{n}_{AB}$

$$\mathbf{n}_{BB'} \perp (\mathbf{n}_{BB_1} \otimes \mathbf{n}_{AB}), \quad (3b)$$

where  $\otimes$  is the vector product. From Eqs. (3a) and (3b)

$$\mathbf{n}_{BB'} = \mathbf{n}_{BB_1} \otimes (\mathbf{n}_{BB_1} \otimes \mathbf{n}_{AB}), \quad (4)$$

$\mathbf{n}_{BB'}$ , is normal to the CP;  $\Rightarrow \mathbf{n} = \mathbf{n}_{BB'}$ .

Hence, from Eq. (4), normal to the CP is given by

$$\mathbf{n} = \pm \mathbf{n}_{BB_1} \otimes (\mathbf{n}_{BB_1} \otimes \mathbf{n}_{AB}), \quad (5)$$

$\pm$  is used to adjust the direction of the CP based on Condition 1.  $\square$

## 6. FCP algorithm

### 6.1. Particles not in contact

The FCP algorithm to find the CP consists of the following steps (Fig. 12):

*Step 1.* Initial guess: if there is a CP from previous DEM time step then use it as the initial guess for the CP in this time step. Otherwise, set the CP as the PB plane of the line connecting the centroids of the two particles.

*Step 2.* For this CP, find closest vertices  $AB$  in particles **A** and **B**. This can be performed by a quick search of the distances of all the vertices of particles to the CP, considering the sign convention for each particle. Among all segments such as  $AB$  that connect a closest vertex  $A$  of particle **A** to a closest vertex  $B$  of particle **B**, the one with the shortest length is chosen. If more

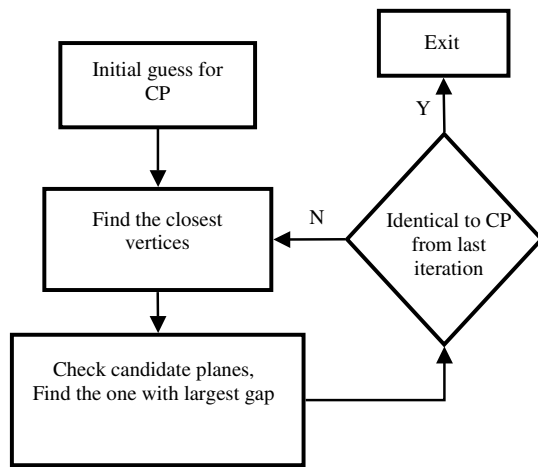


Fig. 12. FCP algorithm.

than one pair of closest vertices have the shortest length (i.e., those vertices are equidistant), then any of them can be chosen to proceed with the algorithm.

**Step 3.** For the two closest vertices *A* and *B* found in Step 2, check all candidate planes of Section 5 and find the one with the largest gap. If any candidate plane produces a gap larger than TOL, then halt the algorithm as the particles are too far from each other to make any, real or potential, contact. If the gap is less than zero then a real contact is detected and an additional step is required as described in Section 6.2.

**Step 4.** If the CP obtained in Step 3 is the same as the one in Step 2, then it is the correct common plane. Otherwise go to Step 2.

This is an iterative algorithm, with each iteration consisting of steps 2–4. The number of iterations required to find the CP, is usually very small and in most cases is less than 2 (see examples in Section 7.2). This is mainly because the iteration is done to locate the two closest vertices, rather than the CP itself.

### 6.2. Particles in contact

For particles in contact, Fig. 13(a), an additional step is performed before the algorithm of Section 6.1 is used, to temporarily separate the particles. This is accomplished by translating the two particles in a direction perpendicular to the CP from the previous time step (Fig. 13(b)). The translation distance TRAN for each particle should be as small as possible in order to make sure that the separated configuration of particles is not much different from the original configuration. However, it should be large enough to ensure the particles are not in contact. Use of a value of TRAN equal to the gap calculated for those particles in the previous DEM time step is recommended. The CP is then determined for this separated configuration. The CP of particles *A* and *B* in their original configuration (Fig. 13(c)) is

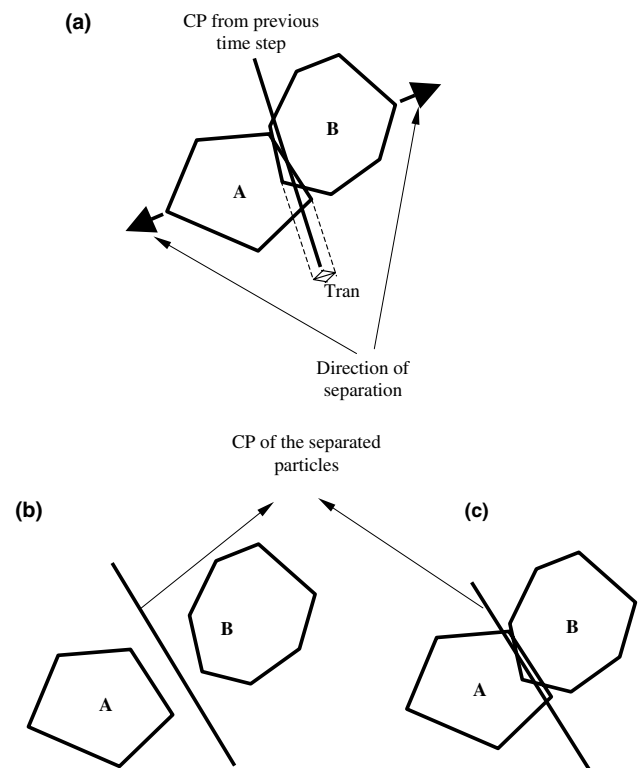


Fig. 13. Separation of particles: (a) configuration of particles in the current time step and the CP from previous time step; (b) the particles after Separation normal to the previous CP, and their (new) corresponding CP; (c) original configuration and the associated CP.

assumed to be the same as that of the separated configuration.

### 6.3. Approximation of CP

In most DEM applications, the location of the CP is not required with a high degree of accuracy. Usually, an approximation would be sufficient. Errors introduced by assumptions such as overlapping of particles when in contact, rather than actual deformation in particle shapes, and simplified constitutive models for contact force calculations are, in general, much larger than the error introduced by using an approximate CP. Although the relative position of two particles and consequently, the position of the CP, may change in every time step, the contact type as well as the two closest vertices to the CP, remain unchanged during a large number of successive time steps. This suggests that an approximated CP in one time step can be obtained from the CP in the previous time step, by assuming that the contact type and the closest vertices remain unchanged. For contact type b, it is assumed that the CP remains parallel to the same face as the previous time step. For contact types c or d, it is assumed that the CP remains parallel to the same edges as the previous time step.



For every particle, a variable PAD (particle accumulated displacement) keeps the accumulated displacement calculated based on the maximum vertex displacement at every time step

$$\text{PAD}_{t+\Delta t} = \text{PAD}_t + |\max(du)| \quad (6)$$

in which  $du$  is the displacement of any vertex of the particle and function  $\max(\cdot)$  is the maximum value, taken over all particle vertices at time step  $t + \Delta t$ . Whenever the value PAD associated with a particle becomes larger than  $0.5 \times \text{TOL}$ , a complete CP check (Sections 5 and 6) is performed for all the particles in the vicinity of that particle, and the value PAD is set to zero for those particles whose contacts are updated. As long as PAD for a particle is smaller than  $0.5 \times \text{TOL}$  the approximation method is employed.

FCP algorithm temporarily separates in-contact particles based on the position of the CP from the previous time step. It is essential to make sure that a CP does exist in the previous time step. A potential contact should be detected before the first occurrence of a real contact. On the other hand, if the gap between two particles is larger

than TOL then no CP is calculated for those particles. The choice of the threshold value,  $0.5 \times \text{TOL}$ , guarantees that complete contact detection will be performed when two particles have a relative displacement larger than or equal to TOL. This in turn warrants that a potential contact and a CP exist before the two particles get in contact.

## 7. Performance of the FCP algorithm

The performance of the FCP algorithm is demonstrated through a series of examples for particles in 2-D and 3-D. The computational time is compared with that using the conventional algorithm proposed in Cundall [33].

### 7.1. Contact detection in 2-D

Fig. 14 depicts 16 pairs of static particles in various configurations in 2-D. For each configuration, the CP is calculated using both FCP algorithm and the

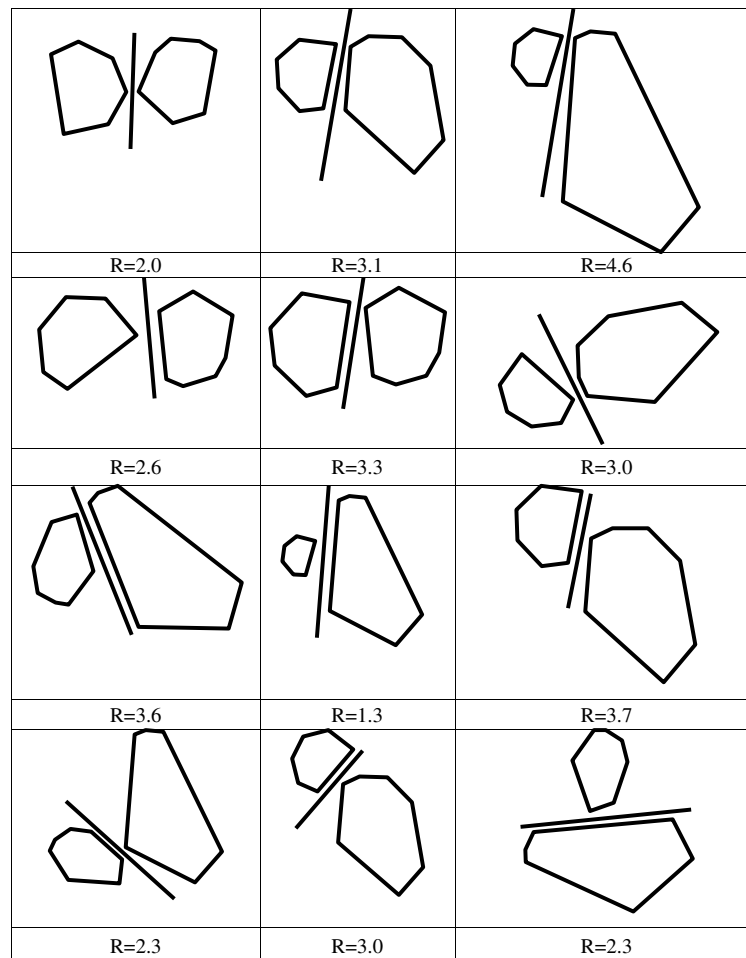


Fig. 14. Examples of CP in 2-D using FCP method: (a) CPs of type a; (b) CPs of type b.

conventional algorithm. The resulting CPs from both algorithms are identical. The speed up ratio  $R$ , defined as the ratio of the CPU run time for the conventional

algorithm to that for FCP algorithm, is shown for each configuration and varies from 1 to 5. FCP is faster than conventional algorithm when  $R > 1$ .

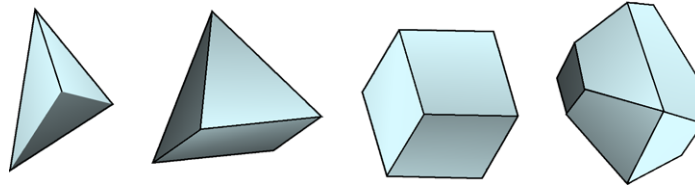


Fig. 15. Particles with 4 vertices (tetrahedron), 5 vertices (pyramid), 8 vertices (cube) and 14 vertices implanted in DBLOCK3D.

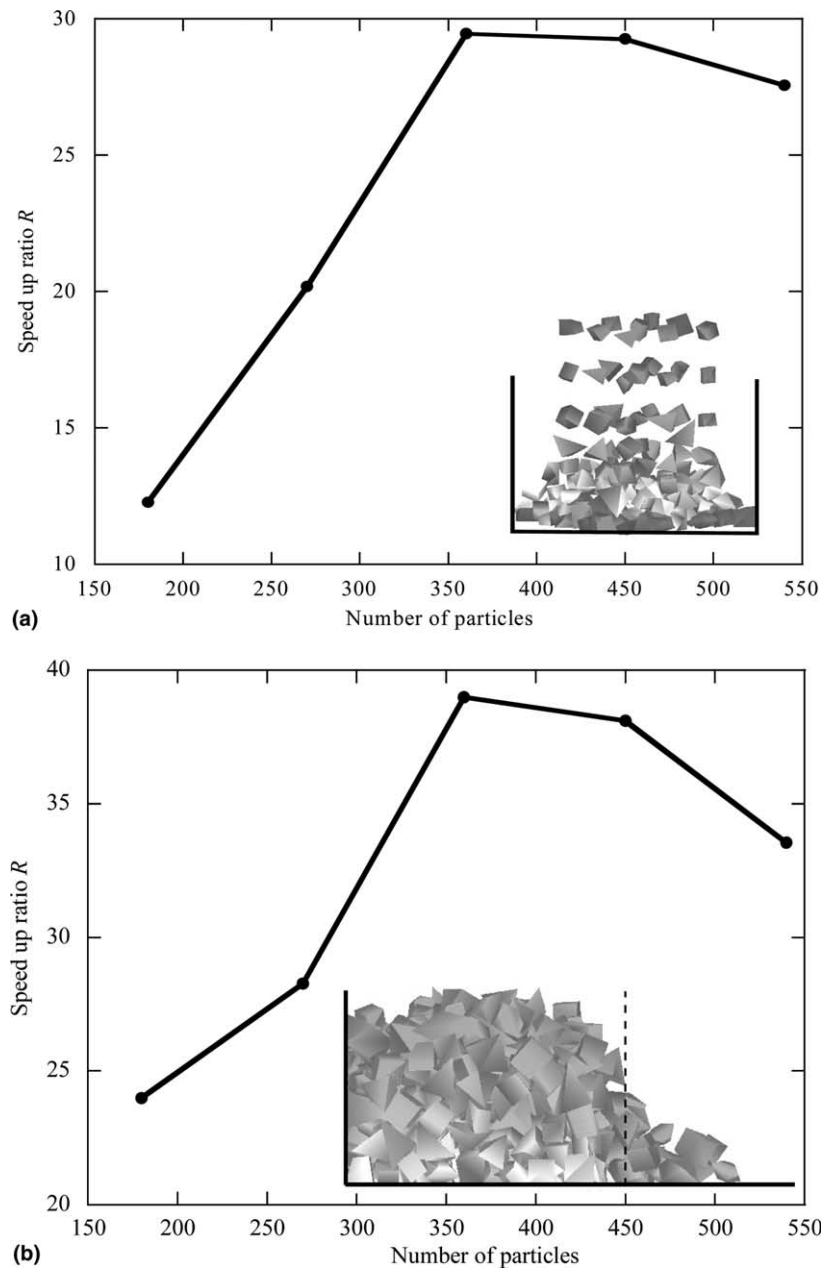


Fig. 16. Speed up ratio of the CPU run time as function of number of particles: (a) accumulation of free falling particles in a box; (b) box side wall removal.

## 7.2. Contact detection in 3-D

The FCP algorithm is implemented in a 3-D DEM code DBLOKS3D developed by the authors. The program incorporates granular assemblies consisting of polyhedral particles with any combination of particle sizes and geometries. Several examples are simulated to illustrate features of the FCP algorithm. For all examples, particles are generated according to a user-defined grain size distribution criterion with a minimum size of 2 cm and a maximum size of 4 cm.

The speed up ratio  $R$  in 3-D is computed for a series of examples with 180, 270, 360, 450 and 540 particles. Each example consists of two separate stages. In the first stage the particles are dropped into a  $30 \times 30$  cm box from a height of about 30 cm. In the second stage, the wall on the right hand side of the box is removed, allowing the particles to flow. The particle geometries are chosen evenly from those shown in Fig. 15. The speed up ratios for the first and the second stages of each example are then calculated from the contact detection algorithm CPU run time required for the first 0.5 s (20,000 time steps) of the simulations. The results, for stage one and stage two, are plotted as a function of number of particles in Fig. 16. The speed up ratio is dependent on the number of particles involved in the test as well as the nature of the test. The speed up ratio ranges from 12 to 38.

The relationship of particle geometry to CPU run time of the FCP algorithm is evaluated using assemblies of 162, 243, 324 and 405 particles for each of the geometries shown in Fig. 15. The particles are dropped on the ground from a height of about 30 cm. The simulations are continued for 1.25 s (85,000 time steps), until a stable configuration is achieved. For all simulations that include the same number of particles, the required CPU run time is normalized with respect to that of the corresponding simulation with 4-vertex particles and is plotted in Fig. 17. The required time for finding the CP does not monotonically increase with number of particle vertices. In contrast, there is always a reduction in the CPU run time when the number of vertices increases from 4 to 5. This suggests that for these examples, the complexity order of the FCP algorithm is smaller than  $O(N)$ . The order of complexity is difficult to determine as it is not only related to the geometry of particles but also to the portion of the total number of contacts which are detected through approximation algorithm of Section 6.3, and the problem being simulated.

Fig. 18 shows an example with 30,000 particles accumulated into a  $80 \text{ cm} \times 80 \text{ cm}$  box, up to a height of about 120 cm (Fig. 18(a)). Only the right and the back walls are shown in the figure. Particle geometries are chosen evenly from those shown in Fig. 15. The wall on the left and the front side of the box is then removed and the test is continued for 1.5 s (100,000 time steps) until a stable configuration is achieved for all particles.

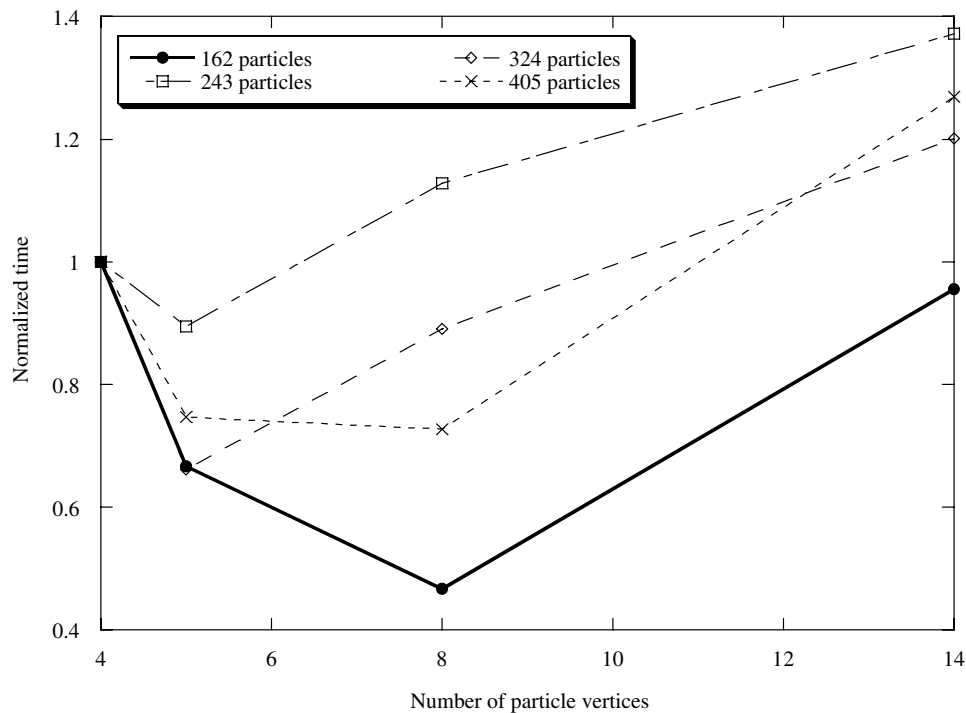


Fig. 17. CPU run time normalized by the corresponding CPU time for 4-vertex particle simulation required to calculate the CP, as a function of number of particles vertices.

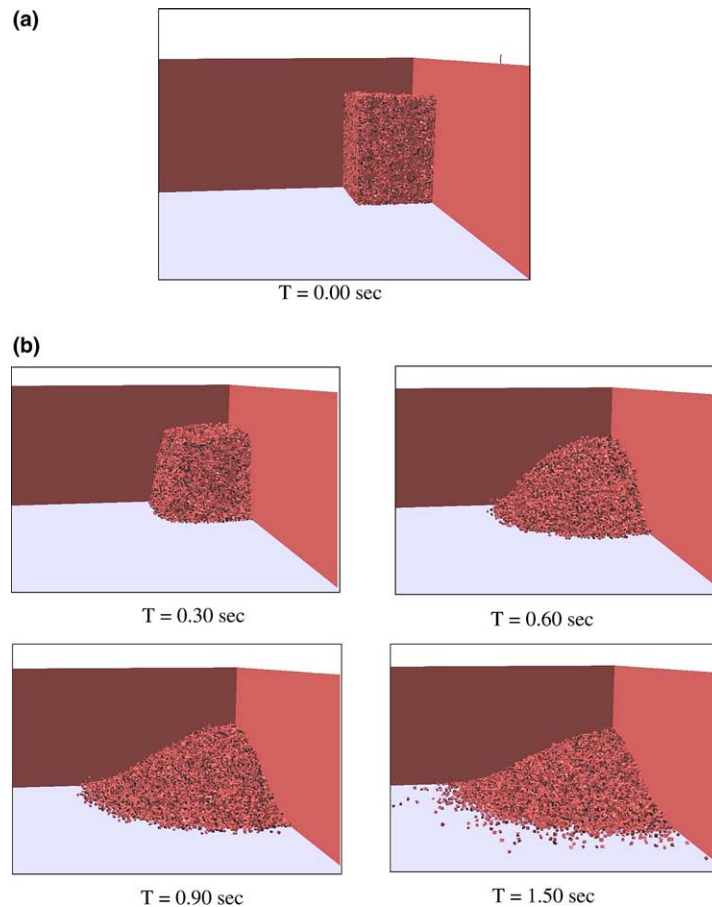


Fig. 18. Simulation with 30,000 particles: (a) particles inside a box, only two side walls are shown; (b) left and front side walls are removed.

Fig. 18(b) plots the view of the assembly at different times after removal of the wall. While the particles on the left and front experience high velocities, the particles on the right and the back are almost motionless. This allows extensive use of both approximation algorithm of Section 6.3 for the slow moving particle in lieu of the exact CP plane detection algorithm of Sections 6.1 and 6.2 used for the fast moving particle. The average number of iterations required to find the common plane using the FCP algorithm (Section 6) is 1.54, with a maximum number of iterations of 4. In general a small number of iterations is involved in FCP algorithm. A speed up ratio  $R = 31$  is computed.

## 8. Conclusions

An efficient algorithm is developed to find the common plane between two-dimensional polygons and three-dimensional polyhedrons. The algorithm takes advantage of properties of the CP to limit the search space for the plane. A quick updating algorithm is also introduced to approximate the new CP from the one in the previous time step. The method is then

compared with the available common-plane detection algorithm. It is observed that the proposed methodology is about 12–40 times faster than the conventional algorithm.

## Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CMS-0113745 and Caterpillar, Inc. 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 National Science Foundation or Caterpillar, Inc. This support is greatly acknowledged. The authors thank Ibrahim Mohammad for preparing the visualization code VisDEMSED.

## References

- [1] Born M, Huang K. Dynamic theory of crystal lattice. Oxford; 1954.
- [2] Maradudin AA. Screw dislocations and discrete elastic theory. J Phys Chem Solid 1958;9(1):1–20.

- [3] Eisenstadt MM. Introduction to mechanical properties of materials. New York: MacMillan; 1971.
- [4] Cundall PA. A computer model for simulating progressive, large scale movements in blocky rock systems. In: International symposium on rock mechanics, Nacy, France: ISRM; 1971.
- [5] Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. *Geotechnique* 1979;29(1):47–65.
- [6] Cleary PW, Campbell CS. Self-lubrication for long run-out landslides: examination by computer simulation. *J Geophys Res Solid Earth* 1993;98(B12):21911–24.
- [7] Campbell CS, Cleary PW, Hopkins MA. Large-scale landslide simulations: global deformation, velocities, and basal friction. *J Geophys Res Solid Earth* 1995;100(B5):8267–83.
- [8] Hopkins MA, Hibler WD, Flato GM. On the numerical simulation of the sea ice ridging process. *J Geophys Res Ocean* 1991;96(C3):4809–20.
- [9] Cleary PW. DEM simulation of industrial particle flows: case studies of dragline excavators, mixing in tumblers and centrifugal mills. *Powder Technol* 2000;209(1–3):83–104.
- [10] Mishra BK, Rajamani RK. Simulation of charge motion in ball mills. Part 1: experimental verifications. *Int J Miner Process* 1994;40(3–4):171–86.
- [11] Moakher M, Shinbrot T, Muzzio FJ. Experimentally validated computations of flow, mixing and segregation of non-cohesive grains in 3D tumbling blenders. *Powder Technol* 2000;109(1–):58–71.
- [12] Bardet JP. Numerical simulation of the incremental responses of idealized granular materials. *Int J Plast* 1994;10(8):879–908.
- [13] Bagi K. Stress and strain in granular assemblies. *Mech Mater* 1996;22:165–77.
- [14] Oda M, Koishikawa I, Higuchi T. Experimental study of anisotropic shear strength of sand by plane strain test. *Soil Foundation* 1978;18(1):25–38.
- [15] Oda M, Konishi J, Nemat-Nasser S. Some experimentally based fundamental results on the mechanical behavior of granular materials. *Geotechnique* 1980;30(4):479–95.
- [16] Subhash G, Nemat-Nasser S, Mehrabadi MM, Shodja HM. Experimental investigation of fabric-stress relations in granular materials. *Mech Mater* 1991;11:87–106.
- [17] Mehrabadi MM, Nemat-Nasser S, Shodja HM, Subhash G. Some basic theoretical and experimental results on micromechanics of granular flow. In: Sakate M, Jenkins JT, editors. *Micromechanics of granular materials*, Amsterdam, Netherland; 1988. p. 253–62.
- [18] Skinnier AE. A note on the influence of interparticle friction on the shearing strength of a random assembly of spherical particles. *Geotechnique* 1969;19:150–7.
- [19] Strack ODL, Cundall PA. The distinct element method as a tool for research in granular media, Part 1. Report to NSF, Department of Civil and Mineral Engineering, University of Minnesota; 1978.
- [20] Strack ODL, Cundall PA. Fundamental studies of fabric in granular materials. Report to NSF, University of Minnesota; 1984.
- [21] Lin X, Ng TT. A three-dimensional discrete element model using arrays of ellipsoids. *Geotechnique* 1997;47(2):319–29.
- [22] Ting J, Khwaja M, Meahum LR, Rowell HD. An ellipsoid-based discrete element model for granular materials. *Int J Numer Anal Meth Geomech* 1993;17:603–23.
- [23] Shodja HM, Nezami EG. A micromechanical study of rolling and sliding contacts in assemblies of oval granules. *Int J Numer Anal Meth Geomech* 2003;27(5):403–24.
- [24] Potapov AV, Campbell CS. A fast model for the simulation of non-round particles. *Granular Matter* 1998;1(1):9–14.
- [25] Williams JR, Pentland AP. Superquadrics and model dynamics for discrete elements in interactive design. *Eng Comput* 1992;9:115–27.
- [26] Barbosa RE. Discrete element models for granular materials and rock masses. PhD thesis, Urbana: Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign; 1990.
- [27] Itasca Consulting Group, I. 3DEC 3-Dimensional distinct element code. User's Manual, 2.0 2.0 2.0; 1998.
- [28] Munjiza A, Andrews KRF. NBS contact detection algorithm for bodies of similar size. *Int J Numer meth Eng* 1998;43(1):131–49.
- [29] Perkins E, Williams JR. A fast contact detection algorithm insensitive to object size. *Eng Comput* 2001;18(1–2):48–61.
- [30] Williams JR, O'Connor R. A linear complexity intersection algorithm for discrete element simulations of arbitrary geometries. *Int J CAE-Eng Comput, Special Ed Discrete Element Meth* 1995;12(2):185–201.
- [31] Krishnasamy J, Jakiela MJ. A method to resolve ambiguities in corner-corner interactions polygons in the context of motion simulation. *Eng Comput* 1995;12(2):135–44.
- [32] Feng YT, Owen DRJ. An energy based corner to contact algorithm. In: Cook BK, Jensen rP, Cook BK, Jensen RP, editors. *Discrete element methods, numerical modeling of discontinua*. Santa Fe, New Mexico, USA; 2002. p. 32–7.
- [33] Cundall PA. 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 Rock Mech Min Sci & Geomech Abstr* 1988;25(3):107–16.
- [34] Cundall PA. Personal communications with the authors; 2003.