



An improved contact detection algorithm for bonded particles based on multi-level grid and bounding box in DEM simulation

Peng Huang*, Yifei Ding, Qiuhua Miao, Gao Sang, Minping Jia

School of Mechanical Engineering, Southeast University, Nanjing 211189, China

ARTICLE INFO

Article history:

Received 17 September 2019

Received in revised form 4 June 2020

Accepted 5 July 2020

Available online 11 July 2020

Keywords:

Contact detection algorithm

Discrete element method

Bonded particle model

Grid subdivision

ABSTRACT

Non-spherical particles have shown increasing importance in the simulation of discrete element methods. Bonded particle model (BPM) is often used to realize the inclusion of non-spherical particles into the calculation using spherical calculation methods. However, the traditional grid subdivision method shows low efficiency and high computational complexity in the face of particle clusters containing a large number of elemental spheres. In this paper, an improved contact detection algorithm for bonded particles based on multi-level grid and bounding box method is proposed. The algorithm treats unbroken or broken new clusters as a unit in the process of contact detection. The contact detection problem including clusters is split into two steps: a pre-detection between the bounding spheres in the primary grid and an accurate contact detection of the elemental spheres in the secondary grid. The hopper experiment and the comminution simulation show that the algorithm has high accuracy and the calculation results are in line with the actual situation. The numerical simulation results of multiple sets of comparisons show that the improved algorithm has higher computing speed than the traditional grid subdivision algorithm, especially when the particle distribution in the calculation domain is relatively loose, the number of particle clusters is large or the composition of the cluster is more complicated.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Most of the particles found in nature, such as gravel, stones, seeds, etc., are non-spherical. Particles of different shapes have different properties. Particle shape has an important influence on the properties of a single particle or an entire particle system. Non-spherical particles tend to exhibit different mechanical and kinematic properties than spherical particles [1].

Discrete element method is a numerical method that can describe the mechanical behavior of particle systems. Since being proposed by Cundall et al. [2,3] in 1971, the discrete element method has been widely used in the simulation analysis and research of particle systems, showing its good simulation accuracy and computational performance. The basic idea of the discrete element method is to treat each particle as an element with a given parameter such as mass, moment of inertia, velocity, position, and so on. Within a given time step, the interaction between elements is calculated according to Newton's laws of mechanics. Then, velocities and positions of all the elements are updated through multiple iterations [4–6].

When iterative calculations of multi-body systems over time are required, contact detection problems between objects occur. The contact detection of the geometric relationship between objects is a key issue

in the discrete element method, because it accounts for a large part of the calculation, and the highest may account for 80% of the total calculation time [7,8]. Contact detection is also encountered in vision systems, CAD, computer animation, virtual reality, contact finite element analysis, component packaging, and robot path planning. In the discrete element method, the contact detection algorithm usually includes a spatial sorting phase and a detection resolution phase. The spatial sorting phase avoids the comparison between all objects by eliminating objects that are not in contact with the target object, thereby reducing the amount of calculation of subsequent contact resolution. If the spatial sorting phase is skipped and the contact between any two objects is directly checked, the time complexity of the algorithm will reach $O(n^2)$ (where n is the number of computed objects). The contact resolution phase accurately determines the details of the contact between two checked objects.

In the spatial sorting phase, the geometry of the object can be abstracted to a higher degree, such as using a bounding sphere or a bounding box. Li proposed a collision detection algorithm based on mixed bounding box using k -dops and bounding sphere [9]. Walizer implied a bounding box algorithm, and test simulations were performed to assess the performance of the bounding box method. Based on the gathered data, the bounding box algorithm was found to be effective at reducing overall simulation runtimes when compared to a basic cell-search algorithm [10]. Ding proposed a collision detection algorithm based on combination of sphere and OBB (oriented bounding box).

* Corresponding author.

E-mail address: huangpeng@seu.edu.cn (P. Huang).

The bounding box is divided into upper and lower layers, the upper layer uses sphere to eliminate many disjoint objects, and the lower layer uses OBB to detect the collision between objects. The algorithm can save time and improve the efficiency of collision detection compared with the traditional single bounding box collision detection [11].

The cost of spatial ordering is positively related to the number of objects n . For systems that cannot make assumptions about the spatial consistency of an object, sorting requires multiple operations. Common spatial sorting methods include: grid subdivision [7], adaptive grid [12], octree method, body based cells, spatial heap sort and other methods. By determining the relative order of the spatial arrangement of the particles, these methods eliminate particles that are too far away from the target particles (the particles are impossible to be in contact with the target object because they are too far from it). Only the position relationships between the target particles and adjacent particles are compared, thereby greatly reducing the computational cost.

The method of contact resolution phase is often related to the particle representation. For example, the most famous contact detection algorithm for convex polyhedrons is the Common Plane (CP). Convex polyhedral particles are often defined by corners, edges, and faces as shown in Fig. 1(a)(b). The positions of the corners are determined by a series of centroid vectors, and each plane corresponds to a unit outward normal vector. The position of each polygon/polyhedron in space consists of a series of vectors pointing to the center of gravity of the particle. The common plane bisects the space between a pair of contact polyhedrons. If the two polyhedrons are in contact with each other, they will intersect the CP and vice versa. The difficulty of this algorithm is to quickly determine the CP surface. Nezami proposed a fast coplanar (FCP) algorithm to find coplanarities between two polytopes [13]. On this basis, Nezami also proposed the Shortest Link Method (SLM) to obtain the shortest link and CP plane more efficiently [14]. Non-spherical particles expressed by continuous functions, such as implicit functions or super-quadratic planes, often determine the contact by solving the expression functions of the two particles in parallel. But this solution is often very difficult, especially in three-dimensional situations [15]. Discrete function representation (DFR) is defined to represent the shape of a particle through a set of bounded nodes with specific associated information. In DFR, the utilize of specific criteria to discretize the surface of a particle into a nodal grid (such as uniform discretization or adaptive discretization) provides us with an ordered set of points that can be quickly searched. The purpose of this representation is to reduce the computational cost of contact detection and to solve the more difficult contact resolution problems. As shown in Fig. 1(c), a circle or a sphere can be represented by a number of discrete points. For the contact resolution between objects defined by n points, the calculation has the maximum theoretical complexity of $O(n)$ [16].

All the above particle representation methods are single particle approaches. These methods do not have consistent applicability when representing different particle shapes, and the corresponding contact detection algorithms are not widely applicable to different particle types. Due to the widespread use of spherical particles in the past and very mature calculation methods, as well as the low consumption of spherical particle expression and detection algorithms, researchers have attempted to use a combination of multiple spherical particles for the simulation of non-spherical particles. Composite particle approaches are put forward in this way. The two most common composite particle models are the Multi-sphere Model (MSM) and the Bonded Particle Model (BPM) [17]. In MSM, particles of any shape can be approximated by a number of overlapping circles (in 2D) or spheres (in 3D). Although the elemental spheres constituting the particles overlap, there is no interaction repulsion between them. These circles or spheres for constitution are considered to be fixed to each other to constitute a rigid body. In Fig. 1(d), the multi-sphere model is used to represent a cylinder, and the spherical elements overlap each other to improve the approximation to the cylinder. With the MSM model, rigid particles of any shape can be represented by centroid coordinates and radii of

several circles or spheres. Moreover, the problem of contact detection between aspherical particles is converted into contact detection between a plurality of spheres, making it possible to take the huge advantage of low calculation consumption of the sphere in contact detection phase. Multi-spheres models have been used effectively in simulating sand and gravel deposits, biomaterials, pharmaceutical tablets, industrial pellets, etc.; especially in simulating the mixing process of particles of different shapes [18,19]. You used MSM to describe ellipsoid particles to study the accumulation and flow characteristics of ellipsoid particles, and designed three experiments to compare the multi-sphere model with the ellipsoid model [20]. He developed a numerical model based on DEM to study the compaction behavior of non-spherical particles. Spheroidal and tetrahedral particles of different aspect ratios were approximated by a multi-sphere approach in which overlapping spheres were glued together to represent the particle shapes [21].

Unlike the MSM model used to represent rigid bodies, BPM is often used to represent particles that have elastoplastic properties or may be broken. BPM also includes a number of elemental spherical particles, except that they do not overlap each other, and the repulsion due to overlap is not negligible. These elemental spheres are considered to be independent of each other in contact detection and force and displacement calculations. Inside a cluster there are breakable bonds between the elemental spheres which provide the force to gather the elemental spheres. Under certain external forces, these bonds will break; enough bond breakage will cause a non-spherical particle to break into multiple non-spherical particles. This makes the model often used for simulations involving deformable particles and particle breakage [22–24]. The most commonly used BPM is based on Hertz-Mindlin model. The bond between two elemental particles (or primary particles) is regarded as an approximate cylindrical connection, as shown in Fig. 2. The forces and moments are calculated as shown in Eqs. (1)–(4)

$$F_b^n = -k_b^n \cdot A_b \cdot v^n \cdot \Delta t \quad (1)$$

$$F_b^s = -k_b^s \cdot A_b \cdot v^t \cdot \Delta t \quad (2)$$

$$M_b^n = -k_b^s \cdot J \cdot \omega^n \cdot \Delta t \quad (3)$$

$$M_b^s = -k_b^n \cdot I \cdot \omega^t \cdot \Delta t \quad (4)$$

where

$$A_b = \pi \cdot R_b^2 \quad (5)$$

$$J = \frac{1}{2} \pi \cdot R_b^4 \quad (6)$$

$$I = \frac{1}{4} \pi \cdot R_b^4 \quad (7)$$

where k_b^s and k_b^n are tangential spring coefficient and normal spring coefficient respectively; v^n , v^t , ω^n and ω^t are normal velocity, tangential velocity, normal angular velocity and tangential angular velocity respectively; I , J and A_b are the moment of inertia, polar moment of inertia and cross-sectional area respectively; R_b is the radius of the bond, which depends on the radius of the smallest element particle; F_b^n , F_b^s , M_b^n and M_b^s are normal force, tangential force, normal torque and tangential torque respectively [25,26]. In Fig. 1(e), a plurality of spheres is used to form a cylinder which can be deformed and broken by an external force.

In general, the problem of contact detection between composite particles is considered to be problem of contact among the elemental spheres. Therefore, the contact detection method between the spherical particles can still be applied to the non-spherical composite particles. However, the basic elements (balls, circles, etc.) used in this composite approach tend to be large in number to more approximately represent non-spherical particles. In some simulations involving composite

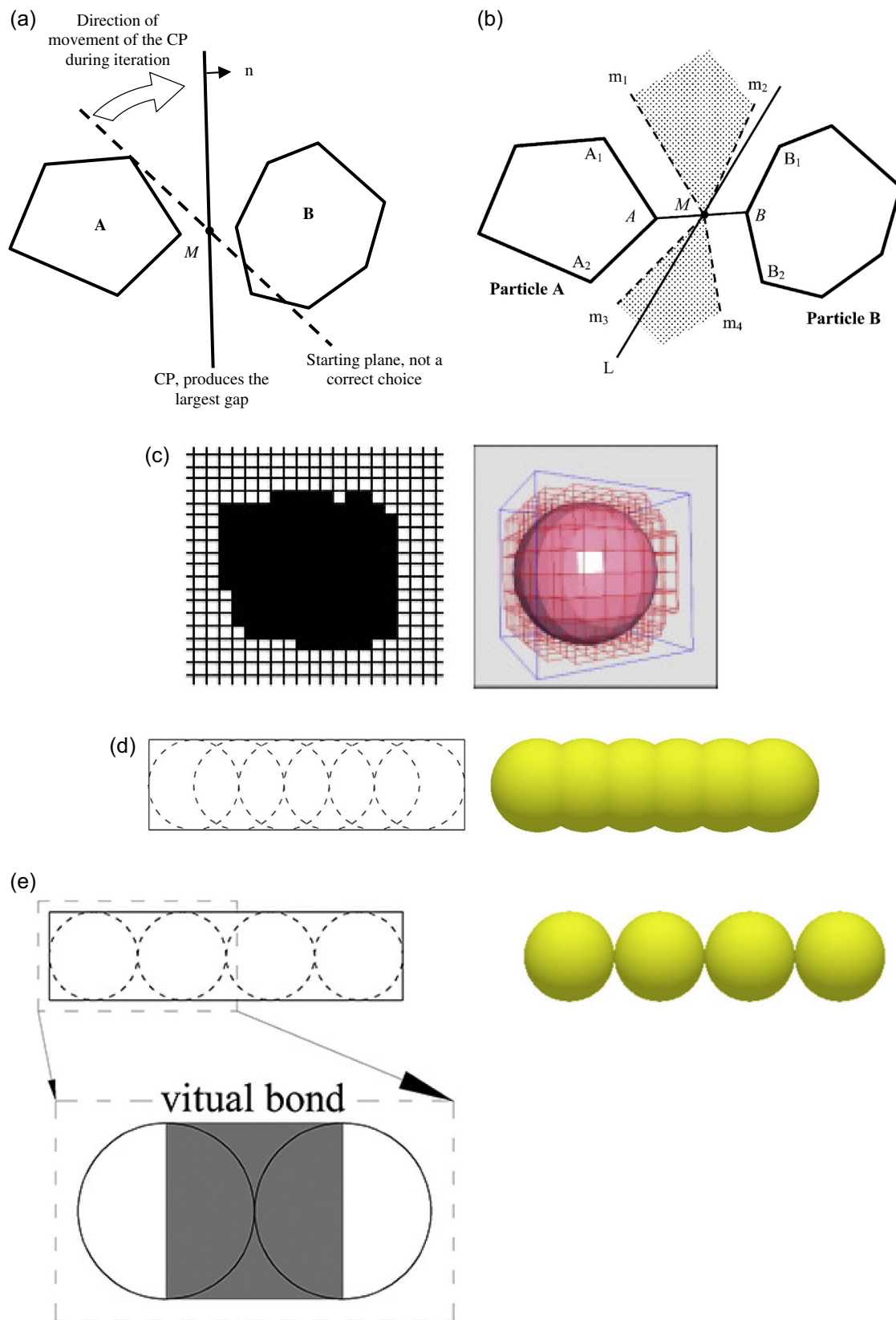


Fig. 1. Common plane methods (a) Traditional common plane method [13] (b) Fast common plane method [14] (c) The circle and sphere represented by DFR method [16] (d) Multi-sphere model (e) Bonded particle model.

particles, a cluster may consist of thousands of spheres, which greatly increases the number of spheres in the simulation domain and the amount of computation of the program [27,28]. This is mainly due to the fact that although each cluster is considered to be a non-spherical particle in the physical sense, it is still independent elements in the calculation process. As a result, the huge increase in the number of elements has led to even more terrible growth in computing expenses.

The traditional grid subdivision method has a very good effect when used in the spatial sorting phase. This method has no requirements for the shape of the particles and is suitable for simulation of different particle representations. However, when this traditional method is used for simulations involving particle bonding models, it is not able to sufficiently reduce the computational time consumption due to the large number of elements and the complex spatial distribution of particles (spheres in the same cluster tend to be very tight while the clusters may be very loose). He proposed a multi-level grid algorithm for general contact detection problems - MGCD. The performance of this algorithm is not sensitive to object size distribution, which is different from other methods [29]. Fang proposed a multi-grid contact detection algorithm for multi-sphere models, which has higher efficiency and less memory overhead than single grid algorithm [30]. However, the improved grid subdivision approach to particle bonding models has rarely been studied.

This paper proposes a contact detection algorithm for bonded particles based on multi-level grid and bounding box method. In this algorithm, a cluster consisting of several balls is considered as a whole. An unbroken cluster and a new smaller cluster fragmented from an original cluster are considered to be an independent non-spherical unit. Through the method of multi-level grid, the primary grid subdivision is performed in the entire simulation domain and the secondary grid is divided within two clusters. The contact detection problem is split into two problems: the contact pre-detection among the clusters and the precise contact resolution between the element spheres in two clusters. Cumbersome contact resolution between all elemental particles belonging to different clusters is no longer required. In order to simplify the expression of clusters and the problem of contact resolution among clusters in the primary grid, the bounding box method is also introduced, and the non-spherical clusters are wrapped with a bounding sphere with virtual boundaries.

In the following, the determination of the cell size and the implementation steps of grid subdivision are first introduced. Then, the implementation process of the multi-level grid method is introduced, including the determination of the equivalence centroid and the equivalent radius of the cluster, the coordinate transformation from the global coordinate system to the local coordinate system and so on. Moreover, an experiment of hopper drop was used to verify the accuracy of the contact detection algorithm. Finally, the advantages of using the multi-level grid method over the traditional grid method in terms of computational speed are compared through several numerical simulations.

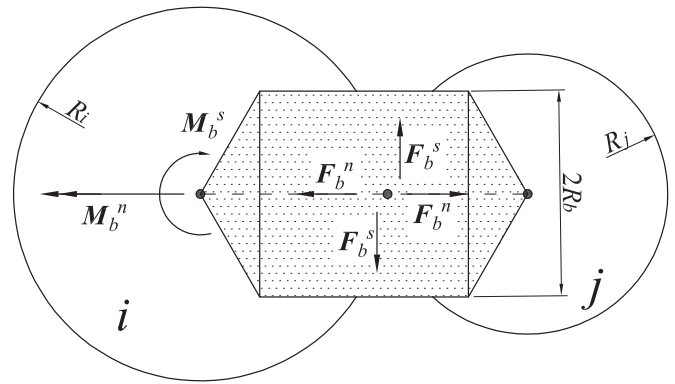


Fig. 2. Bonded particle model based on Hertz-Mindlin.

2. Grid subdivision

2.1. Implementation steps of grid subdivision

In the contact detection of spherical particles, grid subdivision can be used to search for candidate particles in the vicinity of the target particle, so it is used in the spatial sorting stage. Sorting algorithm based on grid subdivision is an effective search algorithm for spherical particles in DEM simulation. By assigning all the particles in the computational domain to a unique cell in the grid, the relative position and relative distance between the particles are determined. This approach provides a simple and effective strategy for rapidly shrinking the set of potential contact particles.

The grid method divides the entire computational domain into a number of cells, and the dimensions of these squares (in two-dimensional) or cubes (in three-dimensional) are l . Each particle generated in the computational domain will be assigned to a particular cell based on its location. During the contact resolution process, it is only necessary to accurately compare the target particles with the particles located in the same cell, and then with the particles in adjacent cells. In this way, all particles in contact with the target particles can be found. Particles in other cells are excluded because they are too far from the target particle to be in contact with it. This effective screening greatly reduces the number of contact resolution calculations. After calculating the overlapping volume (area) and contact force of the particles, it is possible to determine where they are and which cell they belong to in the next iteration. During the next iteration, contact detection and force and displacement calculations are repeated again based on the new position and grid. When the grid subdivision is used in the spatial sorting phase of discrete element method contact detection, the process of simulation iteration is shown in Algorithm 1.

Algorithm 1. Discrete element method with grid subdivision.

```

Generate all the spherical particles in the domain;
Determine the size of the cells and divide the grid in the computational domain;
Assign each particle to a cell based on its coordinates;
while did not reach the stop time
    for each cell  $i$ 
        for each particle  $j$  in the cell  $i$ 
            compare the contact of particle  $j$  with particles in the same cell and adjacent cells;
            calculate the force on the particle  $j$  based on the overlapping volume;
            determine the position of particle  $j$  at the next time step;
            determine the cell to which the particle  $j$  belong in the next time step;
        end for
    end for
end while

```


Since the search in the entire space is avoided, the time complexity of this algorithm is $O(n^2/(L/l)^2)$ in two dimensions and $O(n^2/(L/l)^3)$ in three dimensions. Where n is the number of all elements in the calculation domain; L is the length of the calculation domain; l is the length of each cell. It seems that as long as the l is smaller, the algorithm will take less time [30]. However, the smaller l also brings a larger memory footprint. The space complexity of the algorithm is $O((L/l)^2)$ in two dimensions and $O((L/l)^3)$ in three dimensions. It is not suitable to reduce the mesh size unlimittedly, so the determination of the appropriate cell size becomes a very important issue for the algorithm.

2.2. Rules for determining cell size

There are two common rules for determining cell size, which are respectively based on the maximum and minimum diameter of the sphere in the computational domain.

For the maximum diameter rule, the size of the cell should not be less than the maximum diameter of the sphere in the calculation domain. That is,

$$l \geq d_{\max} \quad (8)$$

where l is the length of the cell and d_{\max} is the diameter of the largest sphere in the domain.

As shown in Fig. 3(a), the target particle needs to be compared with spherical particles in 8 (in two-dimensional) or 26 (in three-dimensional) adjacent cells in addition to the potential contact particles in the same grid. When

$$|O_i O_j| < r_i + r_j \quad (9)$$

is satisfied, there is an overlap between the two spheres. Where O_i and O_j are the spherical coordinates of the two spheres, respectively; r_i and r_j are the radii of the two spheres, respectively.

For the minimum diameter rule, the cell size should be no more than $\sqrt{2}/2$ times of the diameter of the smallest sphere in the calculation domain. That is,

$$l \leq \frac{\sqrt{2}}{2} \cdot d_{\min} \quad (10)$$

where l is the length of the cell and d_{\min} is the diameter of the smallest sphere in the domain.

As shown in Fig. 3(b), the candidate particles that need to be compared with the target particle are not only in the directly adjacent mesh, but in all cells from $I_{q,l}$ to $I_{q,m}$, where $q = x, y$. $I_{q,l}$ and $I_{q,m}$ are calculated by the following formula:

$$\begin{aligned} I_{q,l} &= \text{Int}((U_{q,i} - (r_i + r_{\max}) - U_{\min}^q)/l) + 1 \\ I_{q,m} &= \text{Int}((U_{q,i} + (r_i + r_{\max}) - U_{\min}^q)/l) + 1 \end{aligned} \quad (11)$$

where $U_{q,i}$ is the spherical center coordinate of the target particle i ; r_i is the radius of the target particle i ; r_{\max} is the radius of the largest sphere in the calculation domain; U_{\min}^q is the minimum coordinate of the calculated domain boundary; l is the cell size determined according to eq. (10) [30,31].

Both of these two rules are common methods for determining grid size during meshing, and they have a significant effect on determining the appropriate cell size to speed up the efficiency of contact detection in the discrete element method.

3. Contact detection algorithm based on bounding box and multi-level grid

3.1. Bounding sphere for bonded particles

Particle clusters generated in the DEM simulation domain can be used to represent non-spherical particles of various shapes. Clusters are generated according to a set of coordinate parameters to fit the geometric shape of the crushable particles, and then the bonded particle model is used to generate bonds within the clusters to simulate the mechanical properties of the crushable particles. This process occurs in the particle generation phase before the simulation begins. For example, in Fig. 4, a plurality of spheres of inconsistent size are used to form a cluster of particles which are used to approximate a tablet-shaped non-spherical particle by applying a bonding force therebetween.

If the conventional contact detection calculation method is used, the contact problem between the clusters is converted into a contact problem between the balls constituting the cluster. This obviously avoids complicated contact detection calculations between non-spherical particles, but the computational cost of such contact detection will also be very large when the number of elemental spheres contained in the cluster is relatively large. Contact detection between two clusters of n spheres each has a time complexity of $O(n^2)$. In the entire computational domain, the time complexity of contact detection will reach $O\left(\left(\sum_{i=1}^N n_i\right)^2 / (L/l)^2\right)$ (in two dimension) or $O\left(\left(\sum_{i=1}^N n_i\right)^2 / (L/l)^3\right)$ (in three dimension), where N is the total number of particle clusters, n_i is the number of spheres contained in cluster i , L is the length of the computational domain, l is the size of each cell.

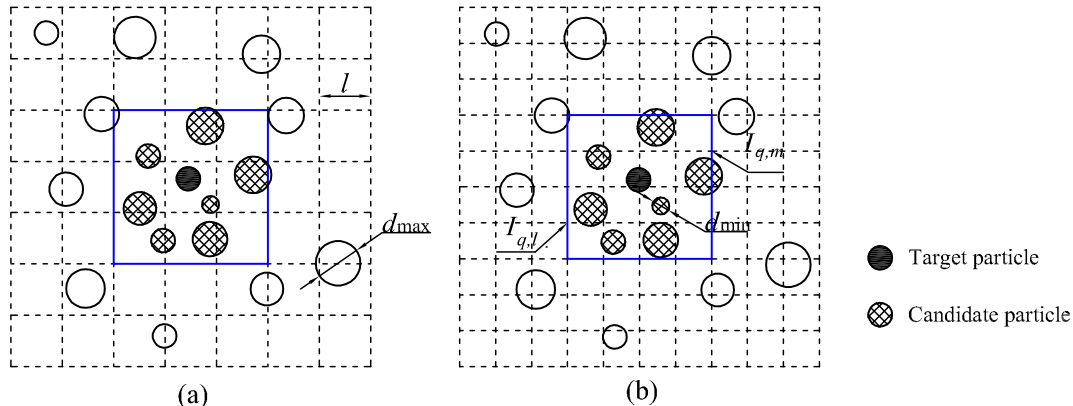


Fig. 3. Two rules for determining cell size (a) Maximum radius rule (b) and minimum radius rule (right).

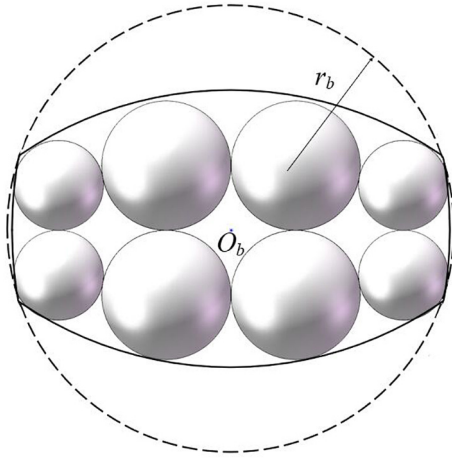


Fig. 4. A tablet-shaped cluster using bonded particle model and its bounding sphere.

At this point, a step-by-step approach to the contact detection problem may reduce computational overhead. Each cluster will be treated as a whole in the multi-level grid contact detection algorithm, but this will inevitably introduce the contact detection problem of non-spherical particles. In order to simplify the problem of contact detection between non-spherical particles, a bounding box method will be used for the particle cluster. This method is generally used for non-spherical single particles, so it is also obviously feasible to use it to wrap a cluster which is used for approximating non-spherical particles.

The spherical bounding box is chosen for its simple shape. The bounding sphere should be the smallest sized sphere that can completely wrap the cluster. Zhao believes that the center of the bounding sphere of the cluster using multi-sphere model can be defined as the centroid of the multi-sphere cluster, and the radius of bounding sphere is equal to the maximum distance from the centroid to the boundary [30,32]. This calculation method is also applicable to particle clusters based on bonded particle model.

The spherical center coordinates of the bounding sphere are the centroid coordinates of the cluster. For any cluster i containing n_i element spheres, its centroid coordinates $O_b^i = [x_b^i, y_b^i, z_b^i]$ can be calculated from

$$x_b^i = \frac{\sum_{j=1}^{n_i} V_j^i x_j^i}{\sum_{j=1}^{n_i} V_j^i}, y_b^i = \frac{\sum_{j=1}^{n_i} V_j^i y_j^i}{\sum_{j=1}^{n_i} V_j^i}, z_b^i = \frac{\sum_{j=1}^{n_i} V_j^i z_j^i}{\sum_{j=1}^{n_i} V_j^i} \quad (12)$$

where $[x_j^i, y_j^i, z_j^i]$ is the center coordinate of the element sphere j in the cluster i ; V_j^i is the volume of the element sphere j in the cluster i ; the range of i is from 1 to N ; and the range of j is from 1 to n_i .

The radius of the bounding sphere is the distance from the center of the bounding sphere to the farthest boundary of elemental spheres. That is

$$r_b^i = \max_{1 \leq j \leq n_i} (|O_b^i O_j^i| + r_j^i) \quad (13)$$

where O_b^i is the centroid coordinate of the bounding sphere i surrounding the cluster i ; O_j^i is the spherical center coordinate of the element sphere j in bounding sphere i ; r_j^i is the radius of the element sphere j in bounding sphere i ; the range of i is from 1 to N ; and the range of j is from 1 to n_i .

Accordingly, by wrapping each non-spherical cluster with a bounding sphere, contact detection among the non-spherical clusters is converted to contact detection among the bounding spheres.

3.2. Multi-level grid contact detection algorithm

3.2.1. Primary grid and contact detection among the bounding spheres

In the following algorithm introduction, the tablet-shaped cluster in Fig. 4 and single-sphere particles are included together in the three-dimensional simulation domain. Each cluster is wrapped with a bounding sphere as described above. The coordinates O_b^i and radius r_b^i of the bounding sphere are calculated by Eqs. (12) & (13). For convenience of representation, the latter schematics are only in two dimensions.

The bounding spheres and the single spheres are considered to be the smallest unit in the computational domain, that is, the independent unit during the contact detection process. In this case, it is equivalent to that the computational domain contains only a few spheres of different sizes.

Grid subdivision can be a good way to sort these spheres in space. The primary grid divides the entire computational domain, as shown in Fig. 5. The cell size of the primary grid needs to be determined first, and the maximum diameter rule mentioned above will be used here. The cell size L satisfies

$$L \geq 2R_{\max} \quad (14)$$

$$R_{\max} = \max \left\{ \max_i (R_b^i), \max_k (r^k) \right\}$$

where R_b^i is the radius of bounding sphere i ; r^k is the radius of the single sphere k . That is, the side length of the cell should be no smaller than the maximum diameter of the bounding sphere and the single sphere.

In the computational domain, the three-dimensional coordinate system $Oxyz$ is adopted. It is divided into N_x, N_y and N_z cells respectively in three dimensions of x, y and z , which satisfy:

$$N_x = \text{floor} \left(\frac{x_L}{2R_{\max}} \right) \quad (15)$$

$$N_y = \text{floor} \left(\frac{y_L}{2R_{\max}} \right)$$

$$N_z = \text{floor} \left(\frac{z_L}{2R_{\max}} \right)$$

where $\text{floor}(\cdot)$ means rounding down; x_L, y_L , and z_L are the lengths of the computational domain in the x, y , and z directions, respectively.

It should be noted that N_x, N_y and N_z should be at least 1. Thus, each bounding sphere and single sphere particles in the computational domain will be grouped into a cell, as shown in Fig. 5. Each target object (including single and bounding spheres) only needs to detect contact with objects located in its same cell and adjacent 26 cells. Taking the bounding sphere a as an example, there are only four single-sphere particles and bounding spheres in the 9

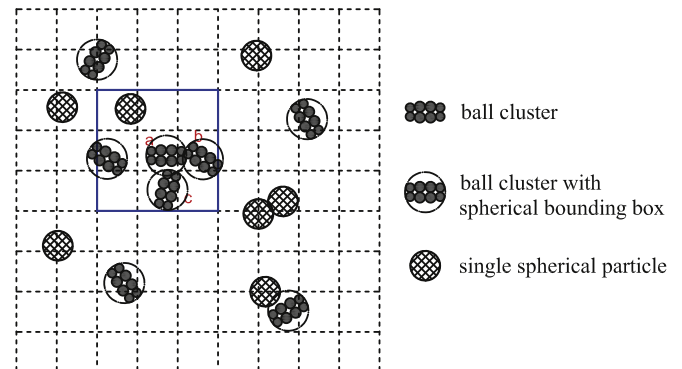


Fig. 5. Primary grid subdivision.

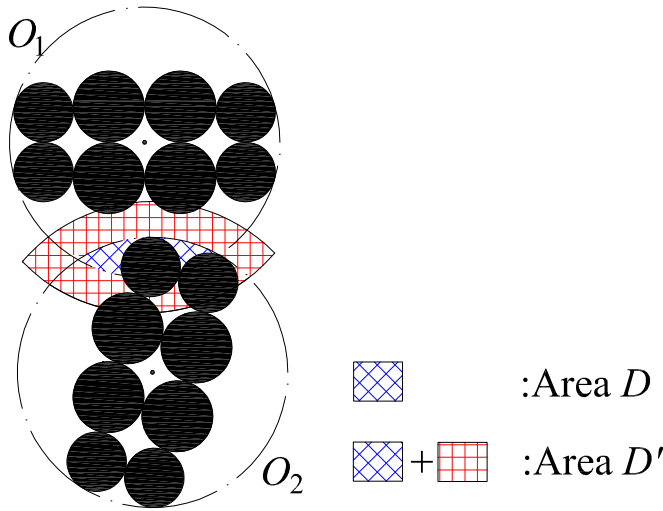


Fig. 6. Two overlapping bounding spheres.

cells around it, among them, only *b* and *c* overlap with the bounding sphere *a*.

Depending on the contact object, there are three types: single sphere - single sphere, single sphere - bounding sphere, bounding sphere - bounding sphere. These three cases are discussed separately below.

1) single sphere - single sphere

The contact detection between two single spheres is very simple, just compare the sum of the radii of the two spheres and the distance between the centers of the two spheres. If the Eq. (9) above is satisfied, it can be determined that the two spheres actually contact. The two spheres in contact will be counted in the actual contact objects set.

2) single sphere - bounding sphere

First, determine if the ball is in contact with the bounding ball. Suppose there is a bounding sphere *i* and a single sphere *k*. If

$$|O_b^i O^k| < R_b^i + r^k \quad (16)$$

is satisfied, the single sphere *i* is in contact with the bounding sphere *k* and they should be included in potential contact objects set.

However, it has not been determined whether the single sphere *k* is actually in contact with the cluster *i* in bounding sphere *i*. If $\exists 1 \leq j \leq n_i$, n_i is the number of elemental spheres in cluster *i*, satisfies

$$|\vec{O_k O_j^i}| < r^k + r_j^i \quad (17)$$

it can be determined that the single sphere *k* is actually in contact with the cluster *i* and they can be counted in the actual contact objects set. Where O_k is the coordinate of the single sphere *k*, O_j^i is the coordinate of the element sphere *j* in the cluster *i*, r^k is the radius of the single sphere *k*, r_j^i is the radius of the element sphere *j* in the cluster *i*.

In the two cases of single sphere - single sphere and single sphere - bounding sphere, it is not necessary to divide the secondary grid to get the result of contact detection, and the corresponding contact detection process can be finished.

3) bounding sphere - bounding sphere

The situation between the bounding sphere and the bounding sphere is the most complicated. First, it should be determined if the

two bounding spheres are in contact. For bounding sphere O_1 and bounding sphere O_2 , judgment of their contact is similar to Eq. (9): it is only necessary to compare the sum of the radii of the two bounding spheres and the distance between the centers of the two bounding spheres.

However, the bounding spheres that contact each other cannot be directly counted into the potential contact objects set, which will make the set too large. Due to the large blank areas in bounding spheres, many “false” contacts often occur and be counted in the potential contact objects set, such as the contact pair *a* & *c* in Fig. 5. Although the two clusters corresponding to *a* & *c* did not contact each other, their bounding spheres apparently overlap. If all elemental particles in cluster *a* & *c* are compared directly, it may be time consuming and meaningless.

Before determining whether the two clusters are to be counted in the set of potential contact objects, one more step is to determine whether there are elemental spheres belonging to each other in bounding sphere O_1 and bounding sphere O_2 (not the center of the elemental spheres, but any part of them).

The overlapping area of the two bounding spheres O_1 & O_2 is denoted as *D*, which is the blue shaded area in Fig. 6. We expand the region outward by length r_{\max} , $r_{\max} = \max_{p=1}^{n_1+n_2} (r_p)$, where r_p is the radius of the elemental spheres *p* in the two clusters, n_1 and n_2 are the number of elemental spheres in bounding sphere O_1 and bounding sphere O_2 , respectively. This gives the area *D'* - the red shaded areas and blue shaded areas in Fig. 6.

When the center coordinate of an elemental sphere is located in the region *D'*, it is equivalent to the condition that the elemental sphere is located in the overlap region *D*.

If and only if the region *D'* contains both elemental particles belonging to the two clusters, these two clusters are counted in the potential contact objects set and move on to the next step; otherwise, there is no actual contact between the two clusters.

3.2.2. Conversion of global coordinates to local coordinates

Secondary grid needs to be divided in the object pairs of the type bounding sphere - bounding sphere in potential contact object set. For convenience, the coordinates of the element spheres in the two bounding spheres need to be converted from global coordinates to local coordinates.

A local coordinate system $O'x'y'z'$ is established to determine the local coordinates of the elements in the cluster. The local coordinate system is shown in Fig. 7.

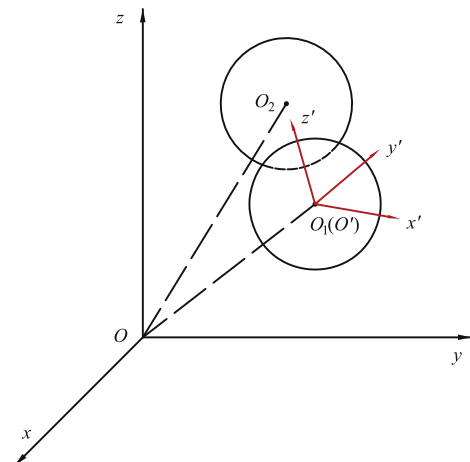


Fig. 7. Global coordinate system and local coordinate system.

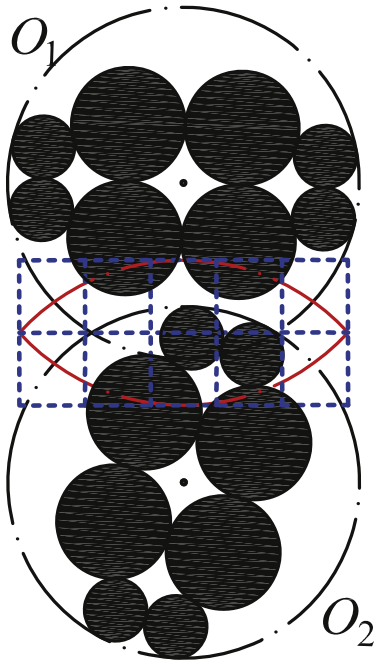


Fig. 8. Secondary grid subdivision.

The center O_2 of the sphere surrounding is chosen as the origin O' of the local coordinate system, and the line $\vec{O_1O_2}$ pointed from the center of O_1 to the center of O_2 is chosen as the z' axis direction. Take the perpendicular line of plane OO_1O_2 at O_1 as the x' axis. Then, y' axis can be determined through the cross product of the vectors.

The vector of x' , y' and z' is unitized to obtain column vector β_1, β_2 and β_3 . β_1, β_2 and β_3 constitute a standard orthonormal basis of $O'x'y'z'$. The standard orthogonal basis of $Oxyz$ is $\alpha_1 = (1, 0, 0)^T$, $\alpha_2 = (0, 1, 0)^T$, $\alpha_3 = (0, 0, 1)^T$. Let $A = (\alpha_1, \alpha_2, \alpha_3)$, $B = (\beta_1, \beta_2, \beta_3)$, then the transition matrix from standard orthonormal $\alpha_1, \alpha_2, \alpha_3$ to standard orthogonal $\beta_1, \beta_2, \beta_3$ is:

$$P = A^{-1}B \quad (18)$$

If the coordinate of a point in the coordinate system $Oxyz$ is $X = (x_1, x_2, x_3)$, its coordinate in the local coordinate system $O'x'y'z'$ is $Y = (y_1, y_2, y_3)$. Then,

$$Y = P^{-1}(X - \vec{OO'}) \quad (19)$$

According to the Eq. (19), the local coordinates of all the element spheres in the region D' are obtained.

3.2.3. Secondary grid and contact detection among elemental spheres

Divide the secondary grid in the area D' (the fusiform area surrounded by red lines), as shown in Fig. 8. In order to facilitate the division, area actually divided is the smallest cuboid region D'' that is capable of wrapping the region D' . The boundary of the region D'' is parallel to x' , y' and z' . In the later calculation, however, since it is only necessary to compare the elemental

spheres in the area D' , the divided area is still equivalent to only the area D' .

In the case of secondary grid subdivision, it is still necessary to determine the size of the cell. Here the minimum diameter rule introduced before is chosen. So, the cell size l satisfies:

$$l \leq \sqrt{2}r_{\min} \quad (20)$$

where r_{\min} is the radius of the smallest elemental sphere in the region D' .

For any target object (elemental sphere), the candidate element spheres that need to be compared with it are not only in the directly adjacent cells, but in the cells from $I_{q,l}$ to $I_{q,m}$, where $q = x, y$. $I_{q,l}$ and $I_{q,m}$ are calculated by:

$$\begin{aligned} I_{q,l} &= \text{Int}((U_{q,i} - (r_i + r_{\max}) - U_{\min}^q)/l) + 1 \\ I_{q,m} &= \text{Int}((U_{q,i} + (r_i + r_{\max}) - U_{\min}^q)/l) + 1 \end{aligned} \quad (21)$$

where $U_{q,i}$ is the coordinate of the target sphere i ; r_i is the radius of the target sphere i ; r_{\max} is the radius of the largest sphere in the region D'' ; U_{\min}^q is the minimum coordinate of the boundary of region D'' ; l is the size of secondary cell determined according to Eq. (20).

The coordinates involved in the secondary grid division above are all in the local coordinate system $O'x'y'z'$.

At this point, each element sphere in the region D' will be assigned to a cell. For each elemental ball, it is compared with elemental balls in the same cell and adjacent cells from $I_{q,l}$ to $I_{q,m}$. If $\exists p \subseteq O_1, q \subseteq O_2$ satisfy $|O_p \vec{O}_q| < r_p + r_q$, the two clusters that the element spheres belonging to are in contact. Where O_p, O_q are the coordinates of the element spheres p, q , respectively; r_p, r_q are the radii of the element spheres p, q , respectively.

The contact of element spheres belonging to two different clusters means that there is actual contact between the two clusters. Then, the two clusters will be included in the actual contact objects set.

3.3. Summary of the process of contact detection algorithms

In the contact detection algorithm based on the bounding box and the multi-level grid described above, the cluster formed by the bonded particle model is included in the simulation domain together with the single sphere particles. The use of both bounding boxes and multi-level grids is primarily directed to contact detection between clusters and clusters.

For contact detection among clusters, the time complexity of the algorithm in the primary grid phase is $O(N^2)$ and the time complexity of each of the two potentially contacted clusters in the secondary grid phase is $O(n^2)$, where N is the number of clusters in the simulation domain and n is the number of element spheres in the overlap region D' of the two clusters. In this multi-level grid algorithm, the time complexity of all processes is $O(N^2/(L_C/L)^2 + k \cdot n^2/(L_L/L)^2)$ (in two dimensions) or $O(N^2/(L_C/L)^3 + k \cdot n^2/(L_L/L)^3)$ (in three dimensions), where k is the number of cluster pairs that need contact resolution in secondary grid. Compared with the traditional method which uses a single-level grid and use the elemental spheres as the basic unit, the time complexity of this improved algorithm is significantly reduced. Of course, the actual effect may be related to the value of k , which varies depending on the simulation situation.

The process of this improved contact detection algorithm used among clusters is shown in Algorithm 2.

Algorithm 2. Improved contact detection algorithm used among clusters.

```

Determine the centroid coordinates and bounding sphere size of all the clusters in simulation domain;
Calculate the cell size of the primary grid;
while did not reach the stop time
    for each cell  $i$  in primary grid
        for each cluster  $p$  in the cell  $i$ 
            compare bounding sphere of  $p$  with bounding spheres in the same cell and adjacent cells;
            if  $\exists$  cluster  $q$ , bounding sphere of  $p$  contact with bounding sphere of  $q$ 
                determine the overlapping area of  $p$  &  $q$ ;
                if  $\exists$  elemental spheres belonging to the two clusters in overlapping area
                    dividing the secondary grid in the overlapping area;
                    for each cell  $j$  in secondary grid
                        for each elemental sphere  $m$  in the cell  $j$ 
                            compare  $m$  with spheres in the same cell or adjacent cells;
                            if  $\exists$  elemental sphere  $n$ ,  $m$  contact with  $n$ 
                                cluster  $p$  contact with cluster  $q$ ;
                            else cluster  $p$  doesn't contact with cluster  $q$ ;
                            end if
                        end for
                    end for
                else cluster  $p$  doesn't contact with cluster  $q$ ;
                end if
            else no potential contact bounding spheres in primary grid
            end if
        end for
    end for
end while

```

4. Algorithm accuracy analysis**4.1. Basic formulations of DEM**

The simulation program used in this article is a homemade DEM code which is written in C++ on the Microsoft Visual Studio. The contact detection algorithm used in this code is the multi-level grid algorithm introduced above and the contact model used is the Hertz-Mindlin model. This contact model is based on Hertzian contact theory and Mindlin-Deresiewicz work [33–37]. The normal force F_n and tangential force F_t are given by Eqs. (22)–(23).

$$F_n = -K_n \delta_n + C_n \vec{v}_n^{\text{rel}} \quad (22)$$

$$F_t = \min \left\{ \mu_s F_n, K_t \delta_t + C_t \vec{v}_t^{\text{rel}} \right\} \quad (23)$$

where

$$K_n = \frac{4}{3} E^* \sqrt{R^* \delta_n} \quad (24)$$

$$C_n = 2 \sqrt{\frac{5}{6}} \frac{\ln e}{\sqrt{\ln^2 e + \pi^2}} \sqrt{2 E^* \sqrt{R^* \delta_n} \cdot m^* v_n^{\text{rel}}} \quad (25)$$

$$K_t = 8 G^* \sqrt{R^* \delta_t} \quad (26)$$

$$C_t = 2 \sqrt{\frac{5}{6}} \frac{\ln e}{\sqrt{\ln^2 e + \pi^2}} \sqrt{K_t m^* v_t^{\text{rel}}} \quad (27)$$

where δ_n and δ_t are respectively normal and tangential overlaps; \vec{v}_n^{rel} and \vec{v}_t^{rel} are the normal and tangential components of relative velocity, respectively; K_n and K_t are the normal and tangential elastic coefficients, respectively; C_n and C_t are the normal and tangential damping

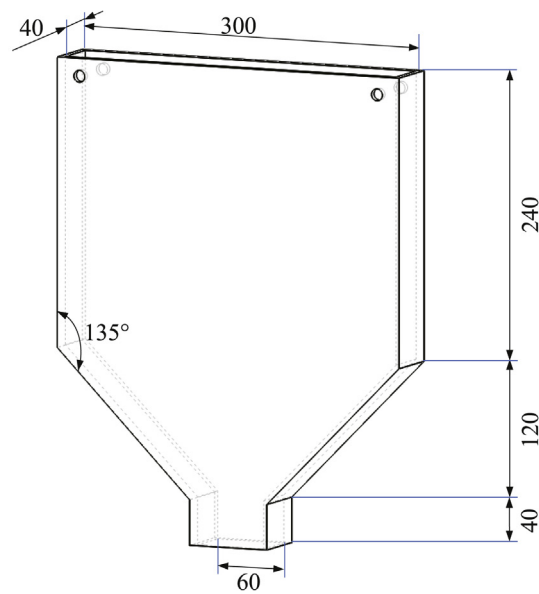


Fig. 9. Hopper for the experiment.

Table 1
Parameters of particles.

Particles	Radius (mm)	Materials	Density (kg/m ³)	Young's modulus (GPa)	Poisson's ratio
Small particle/Single sphere particle	3	PMMA	2500	3.2	0.32
Large particles/Clusters	5				

coefficients, respectively. μ_s is the coefficient of static friction; R^* , E^* , G^* and m^* are equivalent radius, Young's modulus, shear modulus and mass, respectively.

The discrete element method uses a spring damping system to describe interactions between particles, so the relationship between motion and force of the simulation object can be expressed by the differential Eqs. (28)–(30),

$$m^* \frac{d^2 u_n}{dt^2} + C_n \frac{du_n}{dt} + K_n u_n = F_n \quad (28)$$

$$m^* \frac{d^2 u_s}{dt^2} + C_s \frac{du_s}{dt} + K_s u_s = F_s \quad (29)$$

$$I^* \frac{d^2 \theta}{dt^2} + \left(C_s \frac{du_s}{dt} + K_s u_s \right) s = M \quad (30)$$

where I^* is the equivalent moment of inertia of the particle; s is rotation radius; u_n and u_s are the normal and tangential components of relative movement, respectively; θ is the rotation angle of particle; M is external torques on particle.

The critical time step of simulation, namely Rayleigh time-step, can be obtained from Eq. (31).

Table 2
Parameters of material interactions.

Materials	Rolling friction coefficient	Sliding friction coefficient	Restitution coefficient
PMMA-PMMA	0.3	0.1	0.5

$$dt_r = \frac{\pi r_{g \min} \sqrt{\frac{\rho}{G}}}{0.1631v + 0.8766} \quad (31)$$

where G is the shear modulus; ρ is the material density; v is the Poisson's ratio; $r_{g \min}$ is the radius of the smallest particle in simulation domain (which is equal to the radius of the element spheres in this simulation) [38]. Generally, 20%–30% of the critical time step is taken as the simulation time step to ensure the stability of simulation results.

The geometries in the following simulations are described as some continuous functions because of their simple shapes. For example, a

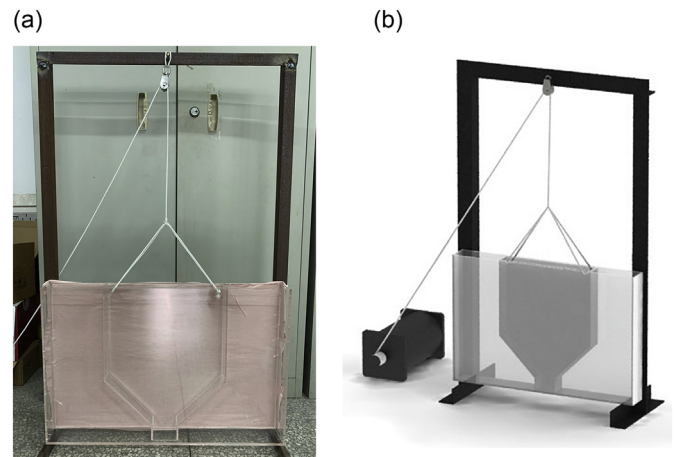


Fig. 10. The experimental devices (a) Photo of experimental devices (b) Schematic diagram of the experiment.

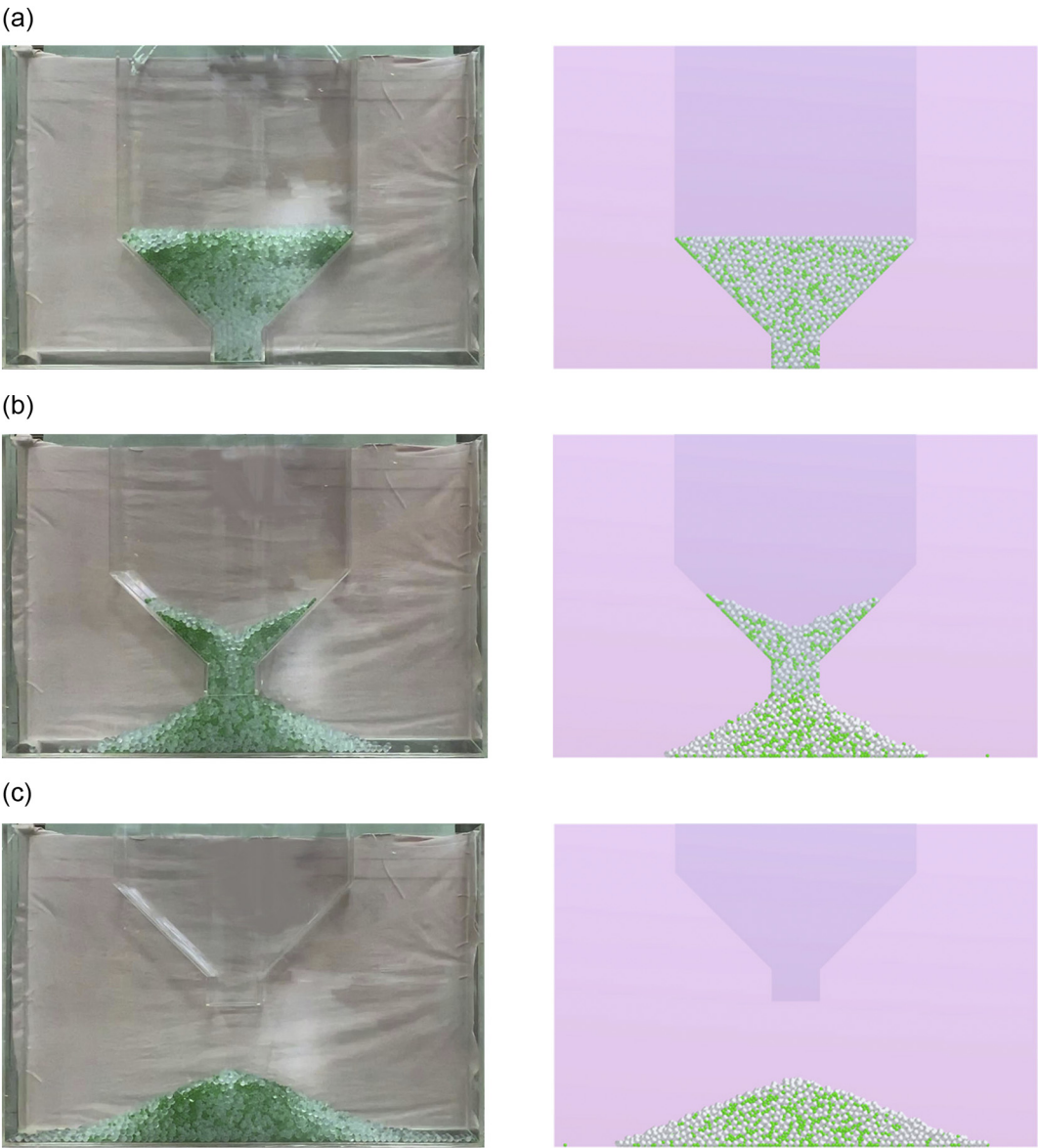


Fig. 11. Experiment and simulation of hopper lifting (a) $t = 0\text{ s}$ (b) $t = 2\text{ s}$ (c) $t = 4\text{ s}$.

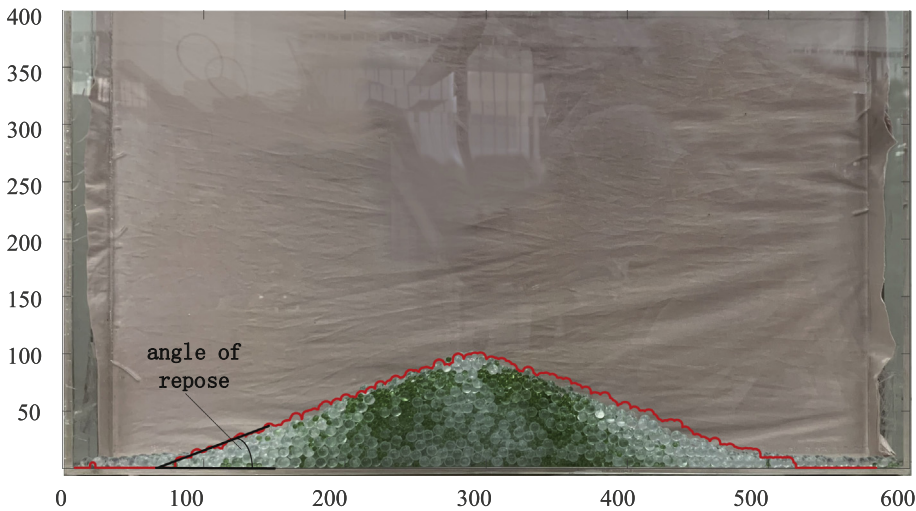


Fig. 12. The upper edge curve of simulation and experiment.

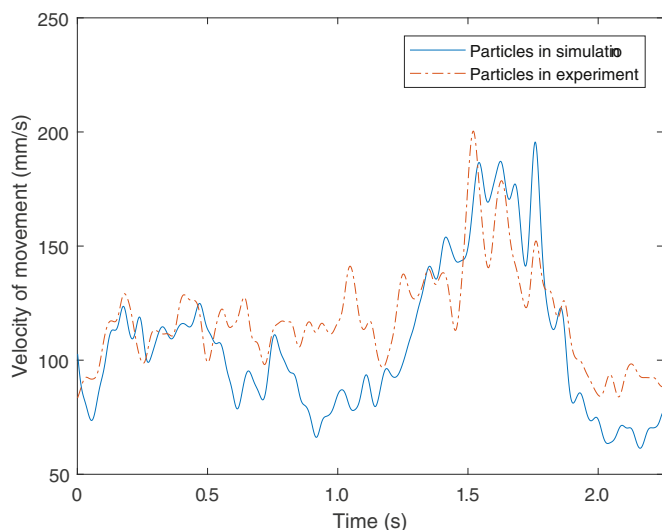


Fig. 13. The velocity of the tracked particles in the simulation.

cuboid or a polyhedron is represented as a combination of several plane equations, a cylinder drum with a smooth inner wall is represented as a cylinder equation, and a drum with lifting bars is represented as a combination of a cylinder equation and several plane equations. When dealing with the problem of contact detection between geometry and particles, the position relation between each particle and geometry is calculated by means of equation solution.

4.2. Hopper experiment

4.2.1. Experimental design and experimental device

The accuracy of the detection algorithm was verified by a physical experiment with a lifting hopper. The experimental equipment consists of a hopper, a rectangular box, a steel frame, a pulley and drive components. The geometry of the hopper is shown in Fig. 9. The drive components include a stepper motor and motor drives. In addition, a high-speed camera is required for collecting experimental images.

In the simulation, particle clusters are often used to simulate breakable particles. Therefore, in the hopper experiment, small particles and large particles correspond to single spherical particles and particle clusters in the simulation respectively. This hopper experiment is used to test the contact detection algorithm without involving the crushing process, so the objects in the hopper experiment are only intact particles. The parameters of the particles are shown in Tables 1 and 2. The materials of the granules and the hopper and the rectangular box are all PMMA (transparent plexiglass), and the material parameters are obtained by parameter calibration [6].

In this experiment, the particles were placed in the hopper, which is lifted at a constant speed to observe the falling and accumulation of particles. The experimental equipment is shown in Fig. 10. The hopper is placed in a PMMA box and suspended under the iron frame by ropes and pulleys.

4.2.2. Experimental and simulation results

Before the experiment started, the hopper touched the bottom and the rope was straightened. A total of about 3000 particles are placed

in the hopper, and the ratio of the large and small particles is 1:1. The upper boundary of the particles is parallel to the horizontal line. After the loading of the particles is completed, the stepping motor pulls the rope to raise the hopper at a constant speed of $v = 0.05$ m/s. The experimental image was recorded using a high-speed camera.

Establish the same simulation environment as the experiment. The material of the particles and the geometry is also PMMA, whose parameters are shown in Tables 1 and 2. First, uniformly distributed single-sphere particles and particle clusters are generated in the hopper. There are 1500 single spherical particles with a radius of 3 mm and 1500 clusters with an equivalent radius of 5 mm. Each particle cluster consists of 10 elemental spheres with a radius of 2 mm to approximate a sphere, and the bonded particle model is used to apply bonds among the elemental spheres. The size of the single-sphere particles is consistent with the small particles in the experiment, and the equivalent size of the particle clusters is consistent with the large particles in the experiment. After the particles in the hopper are stabilized, the hopper is uniformly raised at a speed of $v = 0.05$ m/s. The time at which the movement begins is $t = 0$ s.

The images of each stage in the experiment and simulation process are shown in Fig. 11. Single sphere particles are green and particle clusters are white translucent in the figure. It can be seen from Fig. 11 that the simulation results are similar to the experimental results at various stages of the motion.

4.2.3. Analysis of results

4.2.3.1. Angle of repose. It can be seen from the experimental and simulation results that the particles form a stable accumulation after $t = 4$ s. Calculate the upper edge of the particle pile in the simulation results and plot the upper edge curve onto the experimental photograph, as shown in Fig. 12. It can be seen that the upper edges of the particle piles in the simulation and experiment are very similar. The measurement yields an angle of repose of approximately 21° .

4.2.3.2. Kinematic parameters. To analyze the motion of the particles in the experiment, the motion analysis software Molysis was used to track the motion trajectories of the two particles colored in red and yellow. The coordinates of the two particles in each frame of the recording are determined. Then, the velocity of the particles can be calculated from the coordinates of the particles being tracked in each frame of the image in the experimental record. The velocity of the tracked particles in the simulation is also calculated and plotted in Fig. 13 along with that in the experiment.

As can be seen in Fig. 11 and Fig. 13, the tracked particles are not at a high speed after starting the motion because they are initially located at the edge of the upper boundary. After the hopper began to lift for more than 1 s, the speed of the particles A and B suddenly increased, and the flow of particles begins to slide down the inclined inner wall of the hopper with a high speed. Then the velocity of the particles decreases rapidly and tends to be very slowly after 2 s. At this time, the particles A and B have fallen onto the pile of particles and rolled slightly down the edge of the pile for a distance, eventually becoming stationary.

Therefore, not only the motion of the particles including their direction of motion, trajectory and velocity of motion, but also the angle of repose and stacking pattern (the upper edge of the pile) in the experiments and simulations are very similar. This shows that the improved contact detection algorithm has high precision and the simulation results are in line with reality. It is possible to accurately reflect the motion

Table 3
Parameters of the comminution simulation.

Drum inner diameter (mm)	Drum length (mm)	Number of clusters	Number of balls	Number of lifting bars	Lifting bar height	Lifting bar width	Lifting bar type
600	100	100	500	12	14	18	trapezoidal

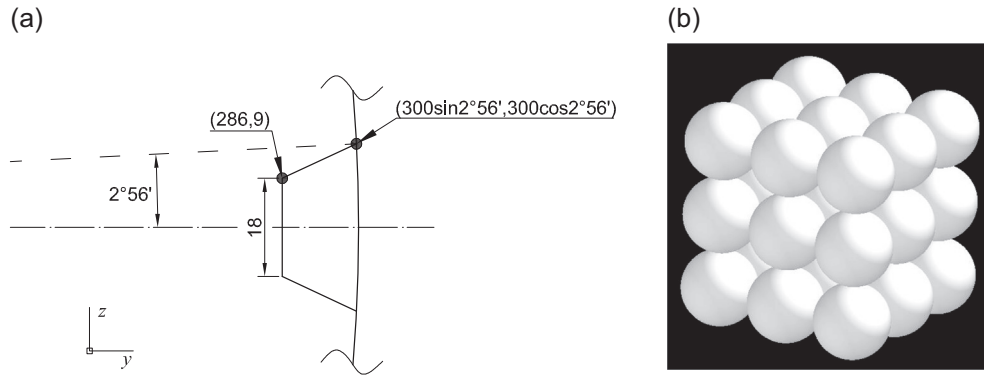


Fig. 14. (a) Lift bars of the drum (b) Cubic cluster.

state of a real particle system in the case of containing particle clusters using bonded particle model.

4.3. Comminution simulation

4.3.1. Simulation parameter configuration

The simulation accuracy of the algorithm including crushing process is verified by a ball mill simulation. In the simulation process, the continuous breaking and decomposition of bonded particles means that the feed is grinded in the ball mill. Parameters of drum and contents inside are shown in Table 3, where the shape of lifting bars is shown in Fig. 14 (a). The graph is plotted on the cross section yOz , where the coordinates are (y, z) . In order to simplify the spatial computation of particles and geometric bodies, the boundary of the cylinder with lifting bars can be expressed by a series of equations. In our work, these boundaries are represented as an elliptic equation and 36 plane equations, with each lift bar corresponding to three plane equations. Assuming that the cylinder is symmetric about the plane yOz and its axis is collinear with the x -axis, the boundary of the cylinder can be expressed as Eq. (32).

$$y^2 + z^2 = 300, \quad x \in [-50, 50] \quad (32)$$

And where there is a lift bar, its boundary is represented by equations of the planes. Take one of the lifting bars on the horizontal plane as an example, the equations of 3 planes can be expressed as Eqs. (33)–(35)

$$y = 286, \quad z \in [-7, 7], \quad x \in [-50, 50] \quad (33)$$

$$\frac{y-286}{300 \sin 2^\circ 56' - 286} = \frac{z-9}{300 \sin 2^\circ 56' - 9}, \quad y \in [286, 300 \sin 2^\circ 56'], \quad x \in [-50, 50] \quad (34)$$

$$\frac{y-286}{300 \sin 2^\circ 56' - 286} = \frac{z+9}{-300 \sin 2^\circ 56' + 9}, \quad y \in [286, 300 \sin 2^\circ 56'], \quad x \in [-50, 50] \quad (35)$$

Table 4
Material parameters of steel and rock.

Material	Poisson's ratio	Density (kg/m ³)	Young's modulus (GPa)
Steel	0.25	7800	200
Rock	0.21	4800	230

For the equations of other 11 lift bars, we only need to rotate these equations around the x -axis by corresponding angles. Due to the use of periodic boundary, there is no boundary at both ends of the drum.

The materials of the drum and balls inside are steel, and the materials of the clusters are rock. Material parameters and material interaction coefficients are shown in Tables 4 and 5. The steel balls and rock clusters in the ball mill move as the drum turns. Different from the spherical clusters mentioned above, the clusters used in this simulation have an approximate cube shape, as shown in the Fig. 14(b). The edge length of each cube cluster is about 20 mm and is composed of 27 spherical element particles with a radius of 3.30 mm. With the drum rotating at a constant speed of 40 rpm, the clusters inside are continuously broken under the impact of steel balls. The simulation lasts for 8 s.

4.3.2. Analysis of simulation results

The crushing simulation of the ball mill is shown in the Fig. 15. It can be seen that with the progress of simulation process, the cluster in the ball mill is continuously broken. Cluster breakage is directly related to the broken of bond between particles. There will be a large number of bonding bonds in the particle cluster, and the number of bonding bonds will be the largest at the time of initial formation. Under the action of external forces, the relative displacement of elements in the cluster leads to the deformation of the particle cluster, which corresponds to the deformation process of materials under stress. Next, the action of the external force causes some bond to break because the force exceeds its critical stress, which corresponds to the formation of a small crack in the material. The work done by external forces will make the fracture of the bond more and the deformation of the cluster more intensified, which corresponds to the crack propagation process on the material. Finally, a sufficient number of bond breaks will lead to cluster separation and the formation of new sub-cluster, which corresponds to the crushing link of the material.

The clusters in Fig. 15 are colored white, while the steel balls are colored according to their velocities. It can be seen from the figure that the movement speed of the steel ball has the following performance: the ball has the lowest speed after the landing, the speed remains constant during the lifting process, and the speed keeps rising during the dropping process. The velocity distribution is consistent with the general law of ball mill in reality.

Table 5
Interaction parameters of steel and rock.

Materials	Coefficient of restitution	Static friction coefficient	Rolling friction coefficient
Steel-Steel	0.6	0.74	0.05
Steel-Rock	0.5	0.55	0.05
Rock-Rock	0.35	0.44	0.05

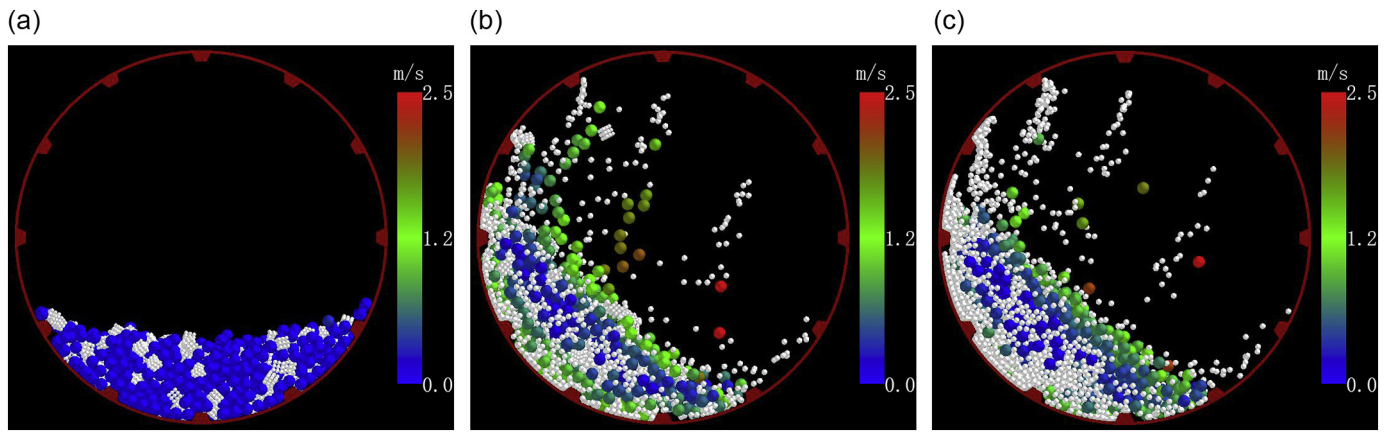


Fig. 15. Ball mill comminution simulation (a) $t = 0$ s (b) $t = 5$ s (c) $t = 10$ s.

The broken rate of the bonds can be obtained by calculating the ratio of the number of broken bonds to the total number of generated bonds. The changes in the broken rate of bonds during the simulation are shown in Fig. 16. It can be found that the bonds in the drum break continuously with the advance of time. The increase of the number of broken bonds also means continuously breakage of the feed in the drum. When the time reaches 8 s, most of the bonds in the drum have broken.

In the comminution simulation of ball mill, the improved contact detection algorithm obtained the correct results. The particle motion state and cluster fragmentation process in the simulation domain are all in line with the actual situation. Thus, the accuracy of the contact detection algorithm used in DEM crushing simulation can be seen verified. In addition, spherical steel balls and non-spherical clusters are included in this simulation, and the mixing of these media with different forms does not cause errors.

5. Algorithm efficiency analysis

5.1. Establishment of numerical simulation

The homemade DEM program is still used in this session, and there is also a homemade program using traditional single-level grid subdivision algorithm as a comparison.

Four particle distributions and six cluster structures were used in the numerical simulation test. The four kinds of particle distributions are shown in Table 6 and Fig. 17. It can be seen that the total numbers of objects (single spheres or clusters) in the four particle distributions are all 11,000. There are two kinds of particle ratios and two levels of density. The particles in distributions 1 and 3 are in a state of stationary accumulation, with small particle spacing and relatively tight distribution, while the particles in distributions 2 and 4 are loosely distributed because they are thrown up with the motion of the drum.

Six kinds of cluster structures that fitting a sphere were used in the test, as shown in Table 7 and Fig. 18. It can be seen that from cluster A to cluster F, the number of elemental particles constituting the cluster gradually increases while the diameter of those decreases accordingly; the shape of the cluster is also closer to the shape of the geometry to be fitted.

The above four kinds of particle distributions and six kinds of bond structures constitute a total of 24 cases. All of these simulations are just one-step tests under the corresponding conditions. Some parameters of these simulations are shown in Table 8. Material parameters and material interaction coefficients are shown in Tables 4 and 5. The computer used in the simulation equips an CPU of Intel (R) Xeon (R) E5–2603 v3 @1.60GHz and 64GB RAM. The contact detection algorithm based on multi-level grid proposed in this paper and the traditional grid subdivision method are used for contact detection

respectively, in order to compare the calculation speeds of the two algorithms. The timing in the program uses the Windows API function GetTickCount(), which has a single run error of approximately ± 25 ms. In order to reduce the error of the timing function, the test for each case runs continuously for 100 times to take the mean. The time units in this section are all milliseconds.

The time statistics of the single-step operation of the two algorithms are shown in Tables 9 and Table 10. Grid subdivision time refers to the time used for grid subdivision, which is the time of the spatial sorting phase mentioned in Section 1. The contact resolution time refers to the time of comparing the contact between the objects, corresponding to the contact resolution phase mentioned in Section 1.

In order to more clearly observe the calculation time of the two algorithms in the case of different particle distributions and clusters of different complexity, Figs. 19, 20 and 21 are drawn. Fig. 19 shows the calculation time of clusters A–F using two algorithms under four particle distributions, mainly comparing the effect of the complexity of the clusters on the calculation speed of the two algorithms. Fig. 20 shows the calculation time of cluster F under different particle distributions to compare the effect of particle distribution on the calculation speed of the two algorithms. Fig. 21 shows the total calculation time of the two algorithms in different cases.

For the convenience of comparison, the difference between the two methods is calculated. As shown in Table 11, the values in the table are the time of the single-level grid minus the time of the multi-level grid. A positive time value indicates that the multi-level grid algorithm takes less time than the traditional single-level grid algorithm in this case; and vice versa.

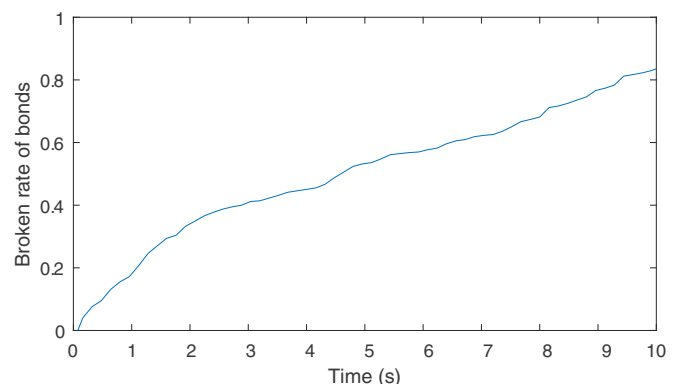


Fig. 16. Broken rate of the bonds in comminution simulation.

Table 6

Four kinds of distribution of particles.

	Number of single spheres	Radius of single spheres(mm)	Number of clusters	Equivalent radius of clusters(mm)	Density
Distribution 1	10,000	10.00	1000	20.00	tight
Distribution 2	10,000	10.00	1000	20.00	loose
Distribution 3	5500	10.00	5500	20.00	tight
Distribution 4	5500	10.00	5500	20.00	loose

5.2. Algorithm efficiency analysis

Comparing the time of the two algorithms, it is not difficult to find that the time of multi-level grid algorithm is significantly shorter than that of the single-level grid algorithm in the grid subdivision phase; the comparison of the contact resolution time and the total time is relatively complicated. After comparative analysis, the following conclusions can be drawn:

- 1) The sensitivity of the multi-level grid algorithm to the complexity of the cluster is significantly lower than that of the traditional grid algorithm.

The multi-level grid algorithm reduces the number of computational units in the computational domain by considering the cluster as a whole. It can be seen in Fig. 19 that the time usage of grid subdivision using multi-level grid algorithm is obviously shorter than that using the single-level grid method. With the increase of cluster complexity, the time usage of grid subdivision using multi-level grid algorithm is

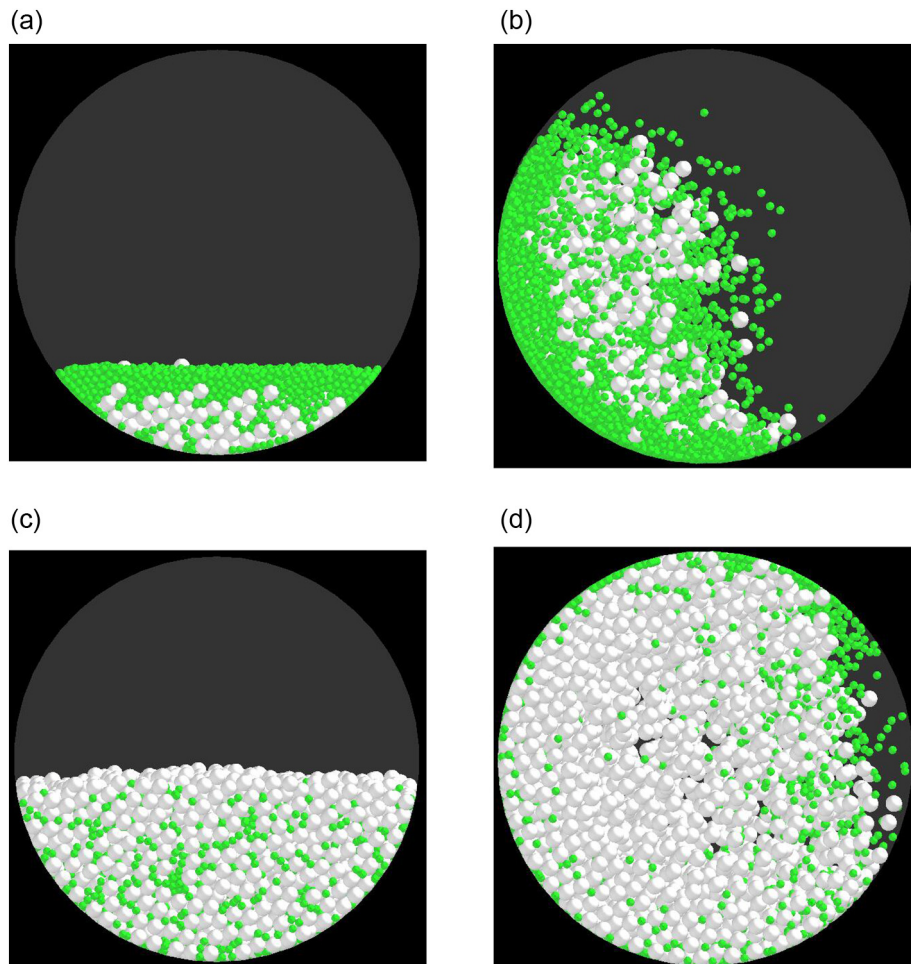
Table 7

Six kinds of cluster structures.

	Number of elemental spheres	Radius of elemental spheres (mm)
Cluster A	6	8.28
Cluster B	10	7.50
Cluster C	25	5.80
Cluster D	50	4.70
Cluster E	75	4.10
Cluster F	100	3.73

almost constant while the time usage of grid subdivision using single grid algorithm is gradually increased, and the difference of time usage becomes more and more obvious.

During the contact resolution stage, when the structure of the cluster is simple and contains a small number of element particles, the time usage of the single-level grid algorithm is slightly shorter than the multi-level grid method. With the increase of the complexity of the


Fig. 17. Four kinds of distribution of particles (a) Distribution 1 (b) Distribution 2 (c) Distribution 3 (d) Distribution 4.

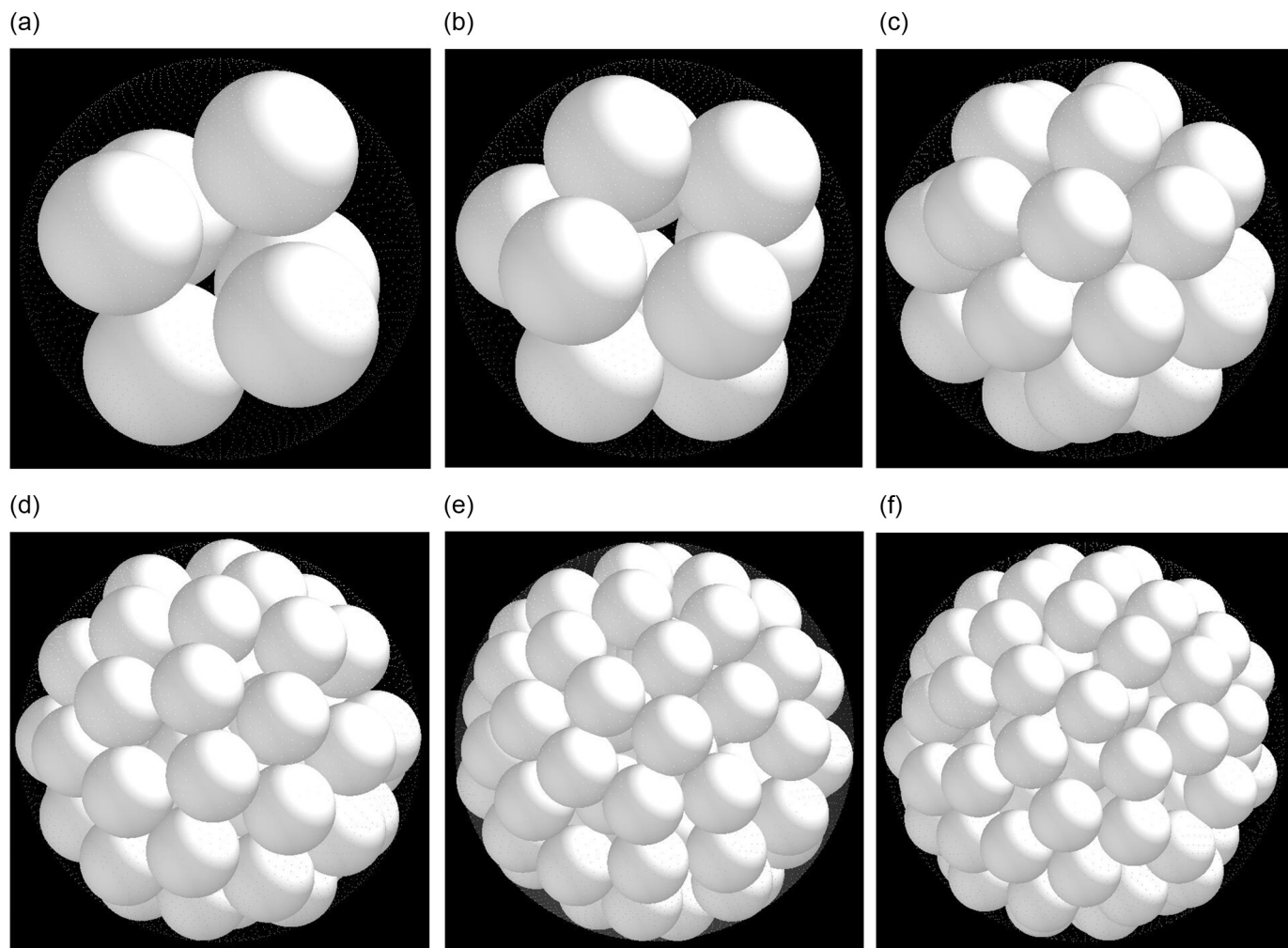


Fig. 18. Six kinds of clusters (a) Cluster A (b) Cluster B (c) Cluster C (d) Cluster D (e) Cluster E (f) Cluster F.

Table 8
Parameters of the 24 cases of simulations.

Drum diameter (mm)	Drum length (mm)	Drum material	Ball material	Cluster material
1000	1000	Steel	Steel	Rock

cluster structure, the time usage of the multi-level grid algorithm increases slowly while the time usage of the single grid algorithm shows

a steep increase trend. The time usage of the single-level grid algorithm far exceeds the multi-level grid algorithm with the increase of elemental particles, and even there are ten or even dozens of time gaps in the case of cluster F.

Comparing the total time usage in one step, the result is similar to the contact resolution phase because the contact resolution occupies the vast majority of the single step time usage, especially when the structure of the cluster is more complicated.

It can be summed that the sensitivity of time usage of the multi-level grid method to the complexity of the cluster is significantly lower than

Table 9
The one-step time for multi-level grid contact detection algorithm (ms).

		Cluster A	Cluster B	Cluster C	Cluster D	Cluster E	Cluster F
Distribution 1	Grid subdivision	128.9	128.9	128.95	128.9	128.1	128.9
	Contact resolution	613.3	636.7	672.65	725.8	782.05	8.05
	Total time in one step	742.2	765.6	801.6	854.7	910.15	942.95
Distribution 2	Grid subdivision	138.3	139.05	138.3	138.3	138.25	138.3
	Contact resolution	331.25	337.5	342.95	353.1	367.2	366.4
	Total time in one step	469.55	476.55	481.25	491.4	505.45	504.7
Distribution 3	Grid subdivision	146.1	145.35	145.3	146.9	146.85	147.65
	Contact resolution	356.25	431.25	561.75	797.65	1057.85	1188.3
	Total time in one step	502.35	576.6	707.05	944.55	1204.7	1335.95
Distribution 4	Grid subdivision	154.7	155.5	155.5	153.9	154.65	153.9
	Contact resolution	182	228.1	278.9	385.15	503.15	536.75
	Total time in one step	336.7	383.6	434.4	539.05	657.8	690.65

Table 10

The one-step time for single-level grid contact detection algorithm (ms).

		Cluster A	Cluster B	Cluster C	Cluster D	Cluster E	Cluster F
Distribution 1	Grid subdivision	286.7	328.95	454.7	637.5	807.8	973.45
	Contact resolution	114.1	165.6	509.35	1599.25	3342.95	5731.25
	Total time in one step	400.8	494.55	964.05	2236.75	4150.75	6704.7
Distribution 2	Grid subdivision	284.35	327.35	466.4	664.05	844.55	1013.3
	Contact resolution	74.25	105.45	299.25	899.2	1846.1	3130.45
	Total time in one step	358.6	432.8	765.65	1563.25	2690.65	4143.75
Distribution 3	Grid subdivision	551.55	782.8	1472.7	2457.8	3383.6	4282
	Contact resolution	221.85	497.65	2546.05	9503.15	20,901.5	36,709.4
	Total time in one step	773.4	1280.45	4018.75	11,960.95	24,285.1	40,991.4
Distribution 4	Grid subdivision	543.75	785.95	1526.55	2572.65	3541.4	4464.05
	Contact resolution	164.05	357.8	1787.5	6637.5	14,578.1	25,544.5
	Total time in one step	707.8	1143.75	3314.05	9210.15	18,119.5	30,008.55

that of the single-level grid method and does not increase sharply with the increase of the number of elemental spheres in cluster.

2) The multi-level grid algorithm is significantly less sensitive to the proportion of cluster than the single-level grid algorithm.

As can be seen from Table 6 and Fig. 17, the number of single spheres and clusters in distributions 1, 2 is 10:1, and the number of single spheres and clusters in distributions 3 and 4 is 1:1. The proportion of clusters in distributions 3 and 4 is higher than that in the distributions 1 and 2.

Compare the differences of calculation speed at different particle ratios. As can be seen from Fig. 20(a), the increase in the proportion of the clusters does not greatly affect the time usage of grid subdivision,

contact resolution and the total time while using the multi-level grid algorithm. As can be seen in Fig. 20(b), the increase in the proportion of clusters greatly increases the time usage of grid subdivision and contact resolution while using the single-level grid algorithm. This is because the increase in the proportion of clusters means an increase in the number of elemental spheres and an increase in the number of computational units in the computational domain for a single-level grid algorithm. As can be seen in conjunction with Fig. 19, this increase in computational complexity is particularly evident when the cluster structure is complex.

Therefore, it is not difficult to conclude that the multi-level grid algorithm is significantly less sensitive to the proportion of cluster than the single-level grid algorithm.

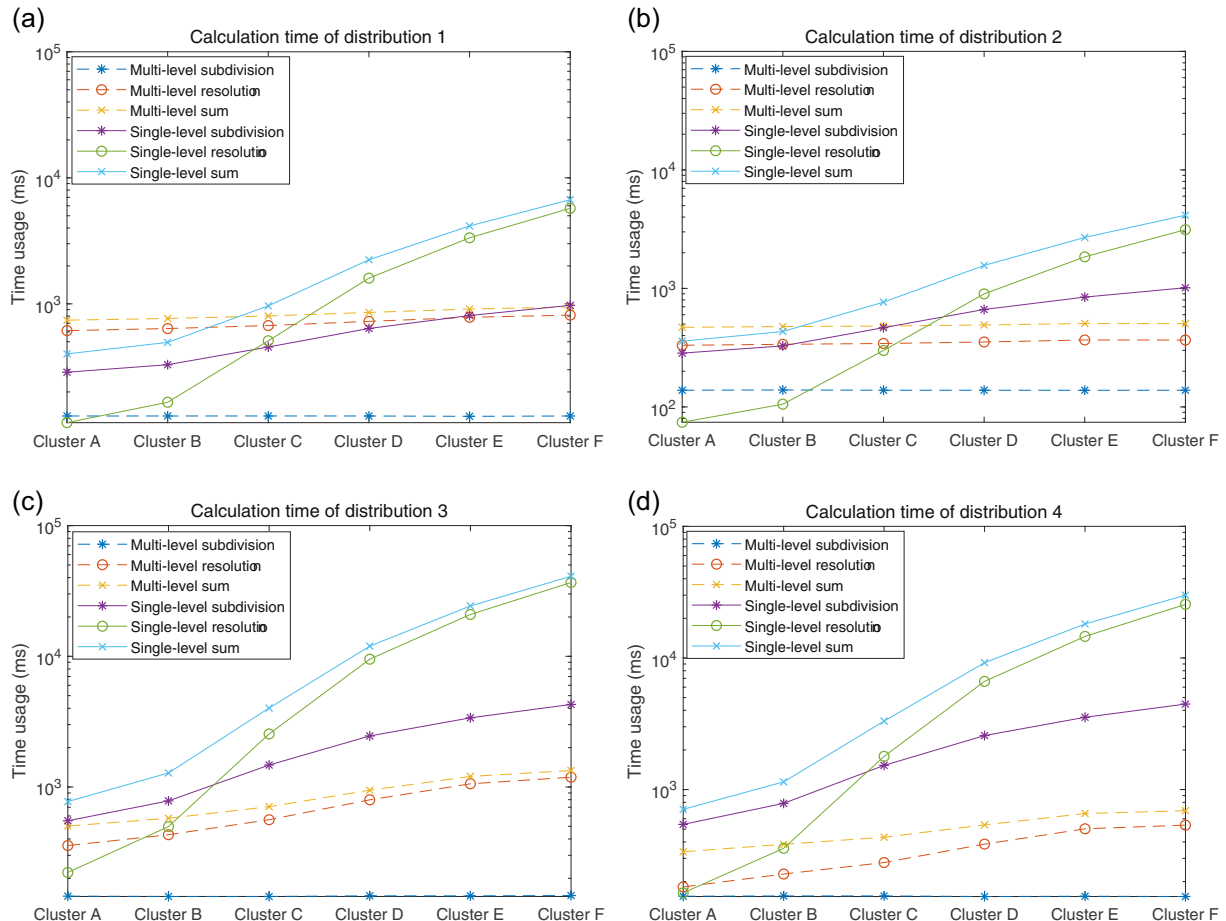


Fig. 19. Calculation time of the two algorithms in the case of different particle distributions and clusters (a) Distribution 1 (b) Distribution 2 (c) Distribution 3 (d) Distribution 4.

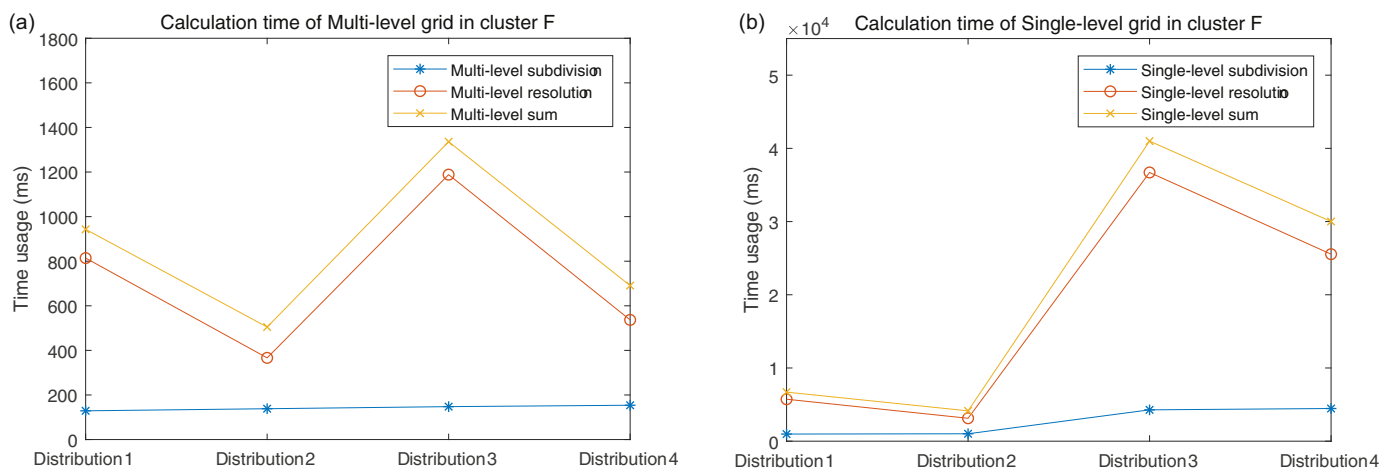


Fig. 20. Calculation time of the two algorithms in the case of clusters F (a) Multi-level grid subdivision (b) Single-level grid subdivision.

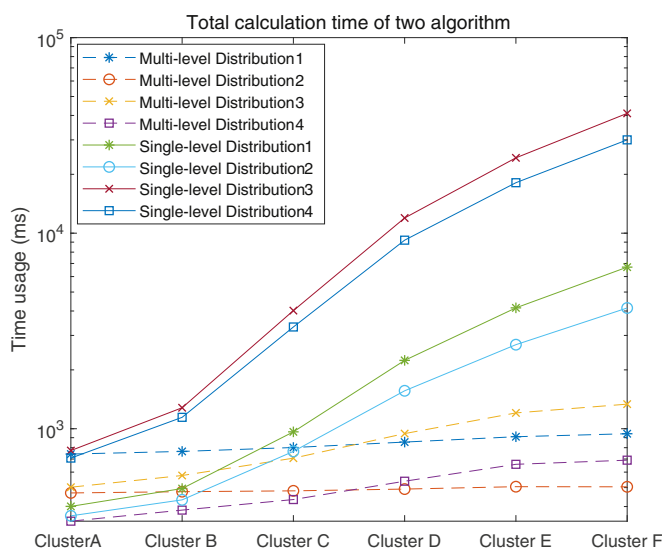


Fig. 21. The total calculation time of the two algorithms in different cases.

3) The multi-level grid algorithm is significantly more sensitive to the density of particle distribution than the single-level grid algorithm.

Among the four kinds of particle distributions, the particles in the distributions 1 and 3 are closely packed and the distribution is tight,

while the particles in 2 and 4 are thrown up in the drum and the distribution is relatively loose.

As can be seen from Fig. 20(b), for the conventional single-stage grid algorithm, the time usage when the particle distribution is loose is slightly shorter than the time usage when the particle distribution is tight. Although the density of the particle distribution also has an influence on the calculation time, the effect is not large. Even in the case of cluster F, this time difference does not exceed 25% of the total time.

As can be seen from Fig. 20(a), it is not difficult to find that when the particle distribution is loose, the detection time is much lower than when the particle distribution is tight for the multi-level grid algorithm. From the results in Table 11, this time gap can be several times. This phenomenon is related to the characteristics of the bounding box method. Since the clusters are surrounded by bounding spheres as virtual surface in the primary grid, the bounding spheres often contain some blank areas. Therefore, the tight distribution of the particles inevitably leads to the overlapping of the bounding spheres of the clusters that are not in contact with each other, just like the clusters a & c in Section 3.2.1.

Therefore, the multi-level grid algorithm is significantly more sensitive to the density of particle distribution than the single-level grid algorithm. When the particle distribution is loose, the calculation time is significantly shorter than when the particle distribution is tight.

4) The more elemental spheres in the cluster, the more clusters in the domain, the more obvious the computational speed advantage of the multi-level grid algorithm.

Observe the difference in time usage between the two methods in Table 11. The positive value indicates that the multi-level grid algorithm

Table 11

The time of the single-level grid algorithm minus that of the multi-level grid algorithm (ms).

		Cluster A	Cluster B	Cluster C	Cluster D	Cluster E	Cluster F
Distribution 1	Grid subdivision	157.8	200.05	325.75	508.6	679.7	844.55
	Contact resolution	−499.2	−471.1	−163.3	873.45	2560.9	4917.2
	Total time in one step	−341.4	−271.05	162.45	1382.05	3240.6	5761.75
Distribution 2	Grid subdivision	146.05	188.3	328.1	525.75	706.3	875
	Contact resolution	−257	−232.05	−43.7	546.1	1478.9	2764.05
	Total time in one step	−110.95	−43.75	284.4	1071.85	2185.2	3639.05
Distribution 3	Grid subdivision	405.45	637.45	1327.4	2310.9	3236.75	4134.35
	Contact resolution	−134.4	66.4	1984.3	8705.5	19,843.65	35,521.1
	Total time in one step	271.05	703.85	3311.7	11,016.4	23,080.4	39,655.45
Distribution 4	Grid subdivision	389.05	630.45	1371.05	2418.75	3386.75	4310.15
	Contact resolution	−17.95	129.7	1508.6	6252.35	14,074.95	25,007.75
	Total time in one step	371.1	760.15	2879.65	8671.1	17,461.7	29,317.9

under this condition takes less time than the single-level grid algorithm; vice versa.

The value of each phase in Table 11 is increasing from the upper left corner to the lower right corner. Combining the previous conclusions 1), 2), and 3), it can be concluded that the multi-level grid algorithm has no advantage when the number of clusters is extremely small and the cluster structure is extremely simple, even takes a little longer time than single-level algorithm. However, as the number of clusters increases and the cluster structure becomes more complex, the multi-level grid algorithm presents an increasingly obvious advantage in speed. In the case of the particle distribution 4, the cluster F, it even shows a speed of about 43 times that of the single-level grid algorithm. This trend can also be seen from Fig. 21.

Therefore, it is not difficult to draw a conclusion that the computing speed advantage of the multi-level grid algorithm becomes more obvious when the computing domain contains more complex clusters.

6. Conclusion

In this paper, an improved contact detection algorithm for bonded particles based on multi-level grid and bounding box method is proposed. In the process of contact detection, the algorithm treats the unbroken or broken new cluster as a basic unit and use multi-level grid. The primary grid divides the entire computing domain to search for potential contact objects pairs, whose cell size is determined according to the maximum diameter rule; bounding spheres are used to wrap the clusters in this level of grid. The secondary grid divides the common area of the two potential contact bounding spheres to determine whether there is substantial contact between the elemental spheres of the two candidate clusters, whose cell size is determined according to the minimum diameter rule.

Theoretical analysis shows that in this improved algorithm, the time complexity of all processes is $O(N^2/(L_C/L)^2 + k \cdot n^2/(L_L/L)^2)$ (in two dimensions) or $O(N^2/(L_C/L)^3 + k \cdot n^2/(L_L/L)^3)$ (in three dimensions), where k is the number of cluster pairs that need contact resolution in secondary grid. Compared with the time complexity of the traditional grid subdivision algorithm which uses a single-level grid and treats elemental spheres as basic unit, the time complexity of the improved algorithm is significantly reduced.

By comparing the results of the hopper experiment with the improved algorithm simulation results, the particle accumulation and the motion of the individual particles in the experiment and simulation were analyzed. The improved algorithm was proved to have high accuracy and the simulation results were consistent with the experiment. In addition, the reliability of the algorithm is also verified by the crushing simulation of non-spherical materials in the ball mill.

The calculation speed of this improved algorithm and the traditional single-level grid algorithm is compared by numerical simulation consisting of single balls and clusters in the drum. The particle distribution in the simulation domain and the complexity of the particle cluster were changed in several simulations, and the following conclusions were obtained:

- 1) The sensitivity of the multi-level grid algorithm to the complexity of the cluster is significantly lower than that of the traditional grid algorithm.
- 2) The multi-level grid algorithm is significantly less sensitive to the proportion of cluster than the single-level grid algorithm.
- 3) The multi-level grid algorithm is significantly more sensitive to the density of particle distribution than the single-level grid algorithm.
- 4) Multi-level grid algorithm has great advantages over single-layer grid algorithm in computing speed, especially when the computing domain contains more complex clusters.

In general, the improved algorithm effectively improves the computational speed of the simulation including bonded particles on the basis of ensuring the accuracy of the simulation results. However, the

algorithm still has some shortcomings, such as using only one kind of spherical bounding box, which does not necessarily fit well with clusters of different shapes. The judgment logic of the algorithm is also cumbersome, which may be detrimental to the parallelization of the algorithm and its application in GPU computing.

Declaration of Competing Interest

None.

Acknowledgement

The authors gratefully acknowledge the financial support of the National Natural Science Foundation of China (No. 51775109) and Natural Science Foundation of Jiangsu Province (BK20181274).

References

- [1] F. Jin, H. Xin, C. Zhang, Q. Sun, Probability-based contact algorithm for non-spherical particles in DEM, *Powder Technol.* 212 (2011) 134–144, <https://doi.org/10.1016/j.powtec.2011.05.003>.
- [2] J. Orford, Alcoholism and marriage. The argument against specialism, *J. Stud. Alcohol* (1975) 1537–1563, <https://doi.org/10.15288/jsa.1975.36.1537>.
- [3] P.A. Cundall, O.D.L. Strack, A discrete numerical model for granular assemblies, *Geotechnique* 29 (1979) 47–65, <https://doi.org/10.1680/geot.1979.29.1.47>.
- [4] X. Deng, J. Scicolone, X. Han, R.N. Davé, Discrete element method simulation of a conical screen mill: a continuous dry coating device, *Chem. Eng. Sci.* 125 (2015) 58–74, <https://doi.org/10.1016/j.ces.2014.08.051>.
- [5] S.S. Razavi-Tousi, J.A. Szpunar, Effect of ball size on steady state of aluminum powder and efficiency of impacts during milling, *Powder Technol.* 284 (2015) 149–158, <https://doi.org/10.1016/j.powtec.2015.06.035>.
- [6] P. Huang, Y. Ding, L. Wu, S. Fu, M. Jia, A novel approach of evaluating crushing energy in ball mills using regional total energy, *Powder Technol.* 355 (2019) 289–299, <https://doi.org/10.1016/j.powtec.2019.07.050>.
- [7] J.R. Williams, R. O'Connor, A linear complexity intersection algorithm for discrete element simulation of arbitrary geometries, *Eng. Comput.* 12 (1995) 185–201, <https://doi.org/10.1108/02644409510799550>.
- [8] B. Mirtich, *Impulse-Based Dynamic Simulation of Rigid Body Systems*, 1996.
- [9] H. Li, D. Zhou, Y. Wu, Collision detection algorithm based on mixed bounding box, *J. Comput. Appl.* 30 (2011) 3304–3306, <https://doi.org/10.3724/sp.j.1087.2010.03304>.
- [10] L.E. Walizer, J.F. Peters, A bounding box search algorithm for DEM simulation, *Comput. Phys. Commun.* 182 (2011) 281–288, <https://doi.org/10.1016/j.cpc.2010.09.008>.
- [11] X.J. Ding, Research on collision detection algorithm based on combined bounding box, *Adv. Mater. Res.* 912–914 (2014) 1353–1356, <https://doi.org/10.4028/www.scientific.net/AMR.912-914.1353>.
- [12] R. Löhner, Automatic unstructured grid generators, *Finite Elem. Anal. Des.* 25 (1997) 111–134, [https://doi.org/10.1016/S0168-874X\(96\)00038-8](https://doi.org/10.1016/S0168-874X(96)00038-8).
- [13] E.G. Nezami, Y.M.A. Hashash, D. Zhao, J. Ghaboussi, A fast contact detection algorithm for 3-D discrete element method, *Comput. Geotech.* 31 (2004) 575–587, <https://doi.org/10.1016/j.compgeo.2004.08.002>.
- [14] E.G. Nezami, Y.M.A. Hashash, D. Zhao, J. Ghaboussi, Shortest link method for contact detection in discrete element method, *Int. J. Numer. Anal. Methods Geomech.* 30 (2006) 783–801, <https://doi.org/10.1002/nag.500>.
- [15] G. Lu, J.R. Third, C.R. Müller, Critical assessment of two approaches for evaluating contacts between super-quadratic shaped particles in DEM simulations, *Chem. Eng. Sci.* 78 (2012) 226–235, <https://doi.org/10.1016/j.ces.2012.05.041>.
- [16] J.R. Williams, R. O'Connor, Discrete element simulation and the contact problem, *Arch. Comput. Methods Eng.* 6 (1999) 279–304, <https://doi.org/10.1007/BF02818917>.
- [17] D.O. Potyondy, P.A. Cundall, A bonded-particle model for rock, *Int. J. Rock Mech. Min. Sci.* 41 (2004) 1329–1364, <https://doi.org/10.1016/j.ijrmms.2004.09.011>.
- [18] B. Ren, W. Zhong, B. Jin, Y. Shao, Z. Yuan, Numerical simulation on the mixing behavior of corn-shaped particles in a spouted bed, *Powder Technol.* 234 (2013) 58–66, <https://doi.org/10.1016/j.powtec.2012.09.024>.
- [19] W. Zhong, A. Yu, X. Liu, Z. Tong, H. Zhang, DEM/CFD-DEM modelling of non-spherical particulate systems: theoretical developments and applications, *Powder Technol.* 302 (2016) 108–152, <https://doi.org/10.1016/j.powtec.2016.07.010>.
- [20] Y. You, Y. Zhao, Discrete element modelling of ellipsoidal particles using super-ellipsoids and multi-spheres: a comparative study, *Powder Technol.* 331 (2018) 179–191, <https://doi.org/10.1016/j.powtec.2018.03.017>.
- [21] Y. He, T.J. Evans, Y.S. Shen, A.B. Yu, R.Y. Yang, Discrete modelling of the compaction of non-spherical particles using a multi-sphere approach, *Miner. Eng.* 117 (2018) 108–116, <https://doi.org/10.1016/j.mineng.2017.12.013>.
- [22] J. Quist, C.M. Evertsson, Cone crusher modelling and simulation using DEM, *Miner. Eng.* 85 (2016) 92–105, <https://doi.org/10.1016/j.mineng.2015.11.004>.
- [23] L. Gao, Q. Luo, Y. Xu, G. Qing Jing, H. Ke Jiang, Discrete element method of improved performance of railway ballast bed using elastic sleeper, *J. Cent. South Univ.* 22 (2015) 3223–3231, <https://doi.org/10.1007/s11771-015-2860-8>.

- [24] Y. Guo, C. Wassgren, B. Hancock, W. Ketterhagen, J. Curtis, Computational study of granular shear flows of dry flexible fibres using the discrete element method, *J. Fluid Mech.* 775 (2015) 24–52, <https://doi.org/10.1017/jfm.2015.289>.
- [25] N. Jiménez-Herrera, G.K.P. Barrios, L.M. Tavares, Comparison of breakage models in DEM in simulating impact on particle beds, *Adv. Powder Technol.* 29 (2018) 692–706, <https://doi.org/10.1016/j.apt.2017.12.006>.
- [26] DEM Solutions Ltd, *EDEM 2.6 Theory Reference Guide*, 2014 35.
- [27] J.F. Ferrellec, G.R. McDowell, A method to model realistic particle shape and inertia in DEM, *Granul. Matter* 12 (2010) 459–467, <https://doi.org/10.1007/s10035-010-0205-8>.
- [28] J.F. Ferrellec, G.R. McDowell, A simple method to create complex particle shapes for DEM, *Geomech. Geoen.* 3 (2008) 211–216, <https://doi.org/10.1080/17486020802253992>.
- [29] K. He, S. Dong, Z. Zhou, Multigrid contact detection method, *Phys. Rev. E - Stat. Non-linear, Soft Matter Phys.* 75 (2007) <https://doi.org/10.1103/PhysRevE.75.036710>.
- [30] Z.Q. Fang, G.M. Hu, J. Du, Z. Fan, J. Liu, A contact detection algorithm for multi-sphere particles by means of two-level-grid-searching in DEM simulations, *Int. J. Numer. Methods Eng.* 102 (2015) 1869–1893, <https://doi.org/10.1002/nme.4875>.
- [31] H. Mio, A. Shimosaka, Y. Shirakawa, J. Hidaka, Optimum cell size for contact detection in the algorithm of the discrete element method, *J. Chem. Eng. Jpn.* 38 (2005) 969–975, <https://doi.org/10.1252/jcej.38.969>.
- [32] D. Zhao, E.G. Nezami, Y.M.A. Hashash, J. Ghaboussi, Three-dimensional discrete element simulation for granular materials, *Eng. Comput. (Swansea, Wales)* 23 (2006) 749–770, <https://doi.org/10.1108/02644400610689884>.
- [33] H. Hertz, Ueber die Berührung fester elastischer Körper, *J. Fur Die Reine Angew. Math.* 1882 (1882) 156–171, <https://doi.org/10.1515/crll.1882.92.156>.
- [34] R.D. Mindlin, Compliance of elastic bodies in contact, *Collect. Pap. Raymond D. Mindlin Vol. I* 16 (1989) 197–206, https://doi.org/10.1007/978-1-4613-8865-4_24.
- [35] R.D. Mindlin, H. Deresiewicz, Elastic spheres in contact under varying oblique forces, *Collect. Pap. Raymond D. Mindlin Vol. I* 20 (1989) 269–286, https://doi.org/10.1007/978-1-4613-8865-4_35.
- [36] T. Rabczuk, S.P.A. Bordas, G. Zi, Computational methods for fracture, *Math. Probl. Eng.* 2014 (2014) <https://doi.org/10.1155/2014/593041>.
- [37] B. Blais, D. Vidal, F. Bertrand, G.S. Patience, J. Chaouki, Experimental methods in chemical engineering: discrete element method—DEM, *Can. J. Chem. Eng.* 97 (2019) 1964–1973, <https://doi.org/10.1002/cjce.23501>.
- [38] M. Lupo, D. Sofia, D. Barletta, M. Poletto, Calibration of DEM simulation of cohesive particles, *Chem. Eng. Trans.* 74 (2019) 379–384, <https://doi.org/10.3303/CET1974064>.