

Research Article

Multisphere Representation of Convex Polyhedral Particles for DEM Simulation

Qiang Zhang ^{1,2,3}, Chaojun Jia ^{4,2}, Jun Yu ⁵, and Jiawen Zhou ³

¹China Institute of Water Resources and Hydropower Research, Beijing 100048, China

²State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources, Beijing 100048, China

³State Key Laboratory of Hydraulics and Mountain River Engineering, Sichuan University, Chengdu 610065, China

⁴School of Civil Engineering, Central South University, Changsha 410075, China

⁵School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China

Correspondence should be addressed to Chaojun Jia; jiachaojun@csu.edu.cn

Received 6 September 2020; Revised 12 January 2021; Accepted 22 January 2021; Published 2 February 2021

Academic Editor: Ma Jianjun

Copyright © 2021 Qiang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The representation of particles of complex shapes is one of the key challenges of numerical simulations based on the discrete element method (DEM). A novel algorithm has been developed by the authors to accurately represent 2D arbitrary particles for DEM modelling. In this paper, the algorithm is extended from 2D to 3D to model convex polyhedral particles based on multisphere methods, which includes three steps: the placement of spheres at the corners, along the edges, and on the facets in sequence. To give a good representation of a polyhedral particle, the spheres are placed tangent to the particle surface in each step. All spheres placed in the three steps are clumped together into a clump in DEM. In addition, the mass properties of the clump are determined based on the corresponding polyhedral particle to obtain accurate simulation results. Finally, an example is used to validate the robust and automatic performance of the algorithm in generating a sphere clump model for an assembly of polyhedral particles. A current FORTRAN version of the algorithm is available by contacting the authors.

1. Introduction

Discrete element method (DEM), initially proposed by CUNDALL and STRACK in 1979 [1], provides a powerful tool to perform numerical simulations of discrete particle systems. In most discrete element software, complex particles are simplified as discs (in 2D) or spheres (in 3D) for easy implementation of contact detection and time-saving. However, it is not accurate to represent a particle with only one disc or sphere, because one disc or sphere on its own cannot capture all the characteristics of the complex geometrical shape of most granular particles. Therefore, DEM modelling of granular materials in which the particles are represented by discs or spheres suffers from limited capability in capturing essential aspects of interparticle interactions and corresponding mechanical behaviors [2, 3].

To obtain more realistic behavior of particles, some particle shape approximations have been introduced into DEM towards realistic particle shape modelling [4]. Basically, there are two different methods: new primitive methods and multisphere methods. The new primitive methods mainly lie in the direct implementation of non-spherical primitives such as ellipsoids [5–7], polyhedra [2, 8, 9], and spherocylinders [10, 11] in DEM. It is a common drawback for new primitive methods that some complicated algorithms need to be developed for contact detection and determination of contact forces, leading to the deterioration of computational efficiency in the simulations. On the other hand, multisphere methods are based on that a nonspherical particle is represented by a clump of rigid spheres which may vary in size and even overlap each other [12]. A fairly comprehensive review of some available multisphere methods can be founded in scientific literatures

[13–18]. Currently, multisphere methods have been applied widely in DEM simulations and implemented into some DEM-packages such as PFC3D [19] and YADE [20] because contact detection and force calculation are still based on some simple algorithms valid for spheres and are, therefore, very efficient and robust.

Polyhedral particles are widely encountered in the field of granular materials such as railway ballast [21, 22], concrete aggregate [23], and soil-rock mixture (S-RM) [24]. In those numerical simulations, they are often simplified as convex and irregular polyhedra whose surfaces consist of several triangle facets. DEM offers a means of obtaining micromechanical insight into the behaviors of granular materials mentioned above. However, how to accurately represent polyhedral particles with clumps of spheres is still one of the key challenges for researchers in DEM simulations. Although the available multisphere methods have been proposed to represent such polyhedral particles with clumps of spheres, they cannot give a good approximation of the particle shape. In this work, a novel algorithm developed by ZHANG et al. [25] is extended from 2D to 3D to accurately represent convex and irregular polyhedral particles based on multisphere methods for DEM simulations.

2. Representation of Convex Polyhedral Particles

2.1. Algorithm. The algorithm developed by Zhang et al. [25] in 2D is briefly introduced firstly in this section. Figure 1 shows schematic operations of the algorithm for the representation of a star-like particle by a clump of discs. As shown in Figure 1, the algorithm includes three steps: placement of discs at the corners (Figure 1(a)), placement of discs along the edges (Figure 1(b)), and placement of discs in the domain (Figure 1(c)). All discs placed in the three steps can be combined as a clump in DEM simulation. To obtain a good representation of the particle, some discs are placed tangent to the interior side of the particle boundary in the first two steps, as shown in Figure 1(b). To be specific, some identical discs are placed at each corner, whose centers are located on the interior angle bisector of each corner in the first step, as shown in Figure 1(a); some identical discs tangent to each other are placed along the interior side of each edge in the second step, as shown in Figure 1(b). In the last step, a series of substeps are performed to place some discs in the domain step by step based on a local Delaunay tessellation, as shown in Figures 1(d)–1(f). In each substep, a set of the centers of some placed discs are selected to create a local Delaunay mesh based on Delaunay triangulation first. Then, the segments are extracted from the local Delaunay mesh. Afterward, two discs connected by each of the segments are called a front element, and some front elements are selected to place new disks, as shown in Figures 1(d)–1(f).

Similarly, the algorithm is extended from 2D to 3D for an accurate representation of convex polyhedral particles by a clump of spheres, as shown in Figure 2. In 3D, it requires that the surface of an irregular polyhedral particle is represented by a triangle mesh where a vertex (denoted by Vi)

corresponds to a corner (denoted by Ci) surrounded by the facets containing the vertex, as shown in Figure 2(a). The algorithm in 3D also includes three steps: placement of spheres at the corners (Figure 2(b)), placement of spheres along the edges (Figure 2(c)), and placement of spheres on the facets (Figure 2(i)). All spheres placed in the three steps are clumped together into a clump in DEM simulations. For the sake of clarity, a tetrahedral particle is used to demonstrate each step of the algorithm, as shown in Figure 2.

In the first step, some spheres of the identical radius (r'_s) are placed at each corner based on the condition that a sphere placed at a corner needs to be tangent to any three of the facets surrounding the corner, whose central position (x_s, y_s, z_s) can be determined by the following equations:

$$\begin{cases} n_{x1}(x_s - x_{p1}) + n_{y1}(y_s - y_{p2}) + n_{z1}(z_s - z_{p3}) = r'_s, \\ n_{x2}(x_s - x_{p2}) + n_{y1}(y_s - y_{p2}) + n_{z1}(z_s - z_{p3}) = r'_s, \\ n_{x2}(x_s - x_{p2}) + n_{y1}(y_s - y_{p2}) + n_{z1}(z_s - z_{p3}) = r'_s, \end{cases} \quad (1)$$

in which (n_{xi}, n_{yi}, n_{zi}) and (x_{pi}, y_{pi}, z_{pi}) are, respectively, the unit normal vector of a facet and the position of a point on the facet, and subscripts 1, 2, and 3 denote any three of the facets surrounding a corner. As shown in Figure 2(b), three spheres are placed at the corners C1, C2, and C3 according to the condition of placement of spheres at the corners.

In the second step, some spheres of identical radius r''_s are placed one by one along each edge, as shown in Figure 2(c). This process is similar to that of the placement of discs along the edges. Taking an edge V1V2, for example, the process of placement of spheres along this edge is given as follows: the first sphere (the blue one shown in Figure 2(c)) is placed according to the condition of being tangent to the two facets (V1V2V3 and V1V4V2) connected by the edge V1V2 and the sphere placed at the corner C1 in the first step (the red one shown in Figure 2(c)) and then the second one is placed next to the first placed one according to the similar condition (the yellow one shown in Figure 2(c)). The process of placement of spheres along the edge V1V2 is completed until the placed sphere overlaps with the sphere located at the corner C2 in the first step, and it is replaced by a new one (the green one shown in Figure 2(c)) which is tangent to the two facets, the last-placed sphere, and the sphere placed at the corner C2 in the first step. For the spheres placed along the edge V1V2 (the blue and yellow ones in Figure 2(c)), their central positions (x_s, y_s, z_s) can be obtained according to the following equations:

$$\begin{cases} (x_s - x_{s1})^2 + (y_s - y_{s1})^2 + (z_s - z_{s1})^2 = (r'' + r'')^2, \\ n_{x1}(x_s - x_{p1}) + n_{y1}(y_s - y_{p1}) + n_{z1}(z_s - z_{p1}) = r'', \\ n_{x2}(x_s - x_{p2}) + n_{y2}(y_s - y_{p2}) + n_{z2}(z_s - z_{p2}) = r'', \end{cases} \quad (2)$$

where (n_{xi}, n_{yi}, n_{zi}) and (x_{pi}, y_{pi}, z_{pi}) are, respectively, the unit normal vector of a facet and the position of a point on the facet, and subscripts 1 and 2 denote the two facets containing the same edge. Similarly, spheres are placed along

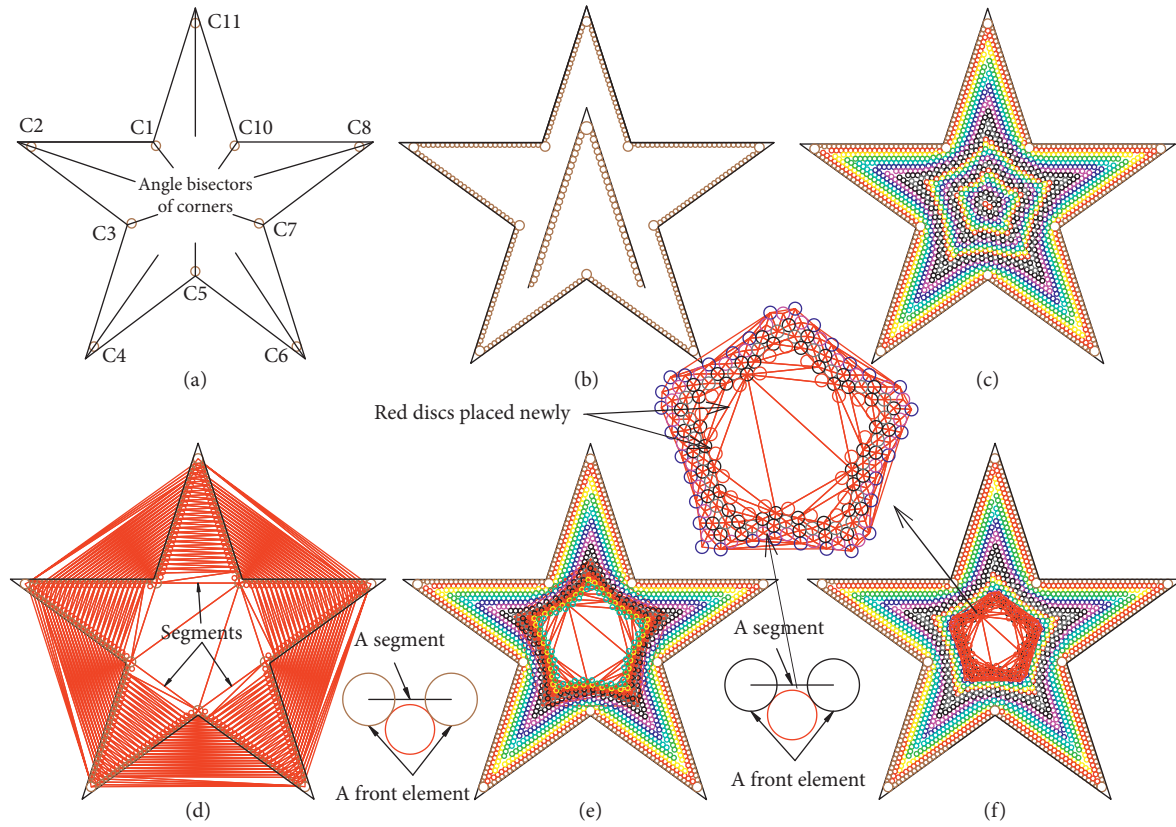


FIGURE 1: Schematic operations of the algorithm to represent a star-like particle: (a) placement of discs at corners; (b) placement of discs along edges; (c) placement of discs in the domain; (d), (e), and (f) the first, thirteenth, and nineteenth substeps in the third steps.

other edges such as V2V3 and V1V3 by repeating the above operations. As shown in Figure 2(c), twenty-four spheres are, respectively, placed along each of the edges (V1V2, V2V3, and V1V3) in this example.

As shown in Figures 2(d)–2(l), spheres of radius r_s between $[r_{\min}$ and $r_{\max}]$ are placed on each facet by a series of substeps based on a local Delaunay tessellation in the last step, where r_{\min} and r_{\max} are the minimum and maximum radii of the spheres to be placed. If r_{\min} is equal to r_{\max} , the radii of the placed spheres are identical. And the smaller the value of r_{\min} , the more spheres to be placed on each facet, which gives a more accurate representation of a polyhedral particle. This process is similar to that of placement of discs in the domain. Taking a facet V1V2V3, for example, the process of the placement of spheres on the facet is given as follows: a Delaunay mesh is created based on a set of points made up of the centers of the spheres including the ones placed at the corners V1, V2, and V3 in the first step and placed along the three edges of the facet in the second step, as shown in Figure 2(d); the segments are extracted from the Delaunay mesh, and similarly, two spheres connected by each of them are taken as a front element, as shown in Figure 2(d); some rational front elements are selected based on a constraint (it will be presented later), and they are used to place new spheres on the facet according to the condition of that a newly placed sphere is tangent to both spheres of the corresponding front element and the facet V1V2V3, as

shown in Figure 2(e). If the central positions and radii of two spheres of a front element are, respectively, represented by (x_{s1}, y_{s1}, z_{s1}) , (x_{s2}, y_{s2}, z_{s2}) , and (r_{s1}, r_{s2}) , the central position of a new sphere to be placed based on the front element (x_s, y_s, z_s) can be determined according to the following equations:

$$\begin{cases} (x_s - x_{s1})^2 + (y_s - y_{s1})^2 + (z_s - z_{s1})^2 = (r_s + r_{s1})^2, \\ (x_s - x_{s2})^2 + (y_s - y_{s2})^2 + (z_s - z_{s2})^2 = (r_s + r_{s2})^2, \\ n_{x1}(x_s - x_{p1}) + n_{y1}(y_s - y_{p1}) + n_{z1}(z_s - z_{p1}) = r_s, \end{cases} \quad (3)$$

where (n_{x1}, n_{y1}, n_{z1}) and (x_{p1}, y_{p1}, z_{p1}) are, respectively, a unit normal vector of a facet and the position of a point on the facet. A new sphere is placed on the facet if it does not overlap with the existing spheres placed on the facet and located inside the polyhedral particle. The first substep is finished when all rational segments are used to attempt to place new spheres on the facet, as shown in Figure 2(e). Then, the centers of the newly placed spheres are also added into the set of points, and the next substep will be performed by repeating the operations, as shown in Figures 2(i)–2(k). The process of placement of spheres on the facet V1V2V3 is completed until no new spheres can be placed on the facet in the current substep, as shown in Figure 2(l).

As shown in Figure 2(f), some unexpected segments extracted from a Delaunay mesh may exist. However, a new sphere cannot be successfully placed on the facet based on a front element corresponding to a long segment. Therefore, a constraint for a front element is suggested to save the calculating time. A front element is called a rational one and used to place a new sphere on the facet only if the length of its corresponding segment satisfies the following relation:

$$L_{ij} - r_i - r_j \leq 2r_{\max}, \quad (4)$$

where r_i and r_j are the radii of two spheres of a front element, respectively, and L_{ij} is the length of the segment corresponding to the front element.

Another constraint proposed by Liu et al. [26] is considered in the third step to save the calculating time, which focuses on ones which are selected from the existing spheres for the generation of a Delaunay mesh in the next substep. Similarly, a sequence number indicating the sequence of substeps is assigned to spheres placed on the facet in each substep. For example, the sequence number of all spheres placed on a facet in the n th substep is marked as n , and the spheres have the same sequence and are marked by the same color if they are placed in the same substep, as shown in Figures 2(h) and 2(i). Suppose that l_n represents the sequence number of the spheres placed in the current substep; then, the center of an existing sphere placed in previous substeps would be regarded as a member of a set of points for the next generation of a Delaunay mesh if the sequence number of the sphere l satisfies as follows:

$$l \geq l_n - h, \quad (5)$$

where h called an advancing-front thickness in Ref. [26] is an integer number greater than 0. If h is set to a larger number, it will require more calculating time to complete the processes of placements of spheres on a facet because all the placed spheres are selected for the generation of a Delaunay mesh in the next substep. On the other hand, if h is set to 1, only the spheres placed in the current substep are selected for the generation of a Delaunay mesh in the next substep. In this case, the algorithm may be not convergent, especially for complex polyhedral particles. Therefore, it is a good strategy that the spheres placed in the last several substeps are only selected for generation of a Delaunay mesh in the next substep, and the generated Delaunay mesh is called a local Delaunay mesh, as shown in Figures 2(g) and 2(h).

2.2. Parameters and Comparison. As presented in the last section, the spheres are placed based on different conditions in the three steps, and their central positions can be obtained easily by solving the corresponding equations such as equations (1)–(3) by the method presented in Ref. [27]. Determination of the radii of the spheres placed in each step

and the parameters of the algorithm are introduced as follows.

In the first two steps, r'_s and r''_s denote the radius of a sphere to be placed at a corner and along an edge, respectively. In general, there is no relation between r'_s and r''_s . For simplicity, a relation of $r'_s = r''_s$ is adopted here, which means that the radii of all spheres placed at the corners and along the edges are identical. In addition, a parameter N_s , denoting the number of spheres placed along the shortest edge, is proposed to determine r'_s and r''_s indirectly. For example, for a polyhedral particle, the length of its shortest edge (L_{\min}) are determined by comparing those of all its edges. Then, r'_s and r''_s are determined based on the relation of $r'_s = r''_s = L_{\min}/2N_s$ when N_s is given by users.

In the third step, r_s denotes the radius of a sphere to be placed on a facet, which ranges randomly and uniformly in an interval of $[r_{\min}, r_{\max}]$. Two parameters (f_r and r_{\min}/r_{\max}) are proposed to determine the interval of $[r_{\min}, r_{\max}]$. f_r is defined as a ratio of r_{\min} to r'_s or r''_s , and r_{\min}/r_{\max} is defined as a ratio of r_{\min} to r_{\max} . The relation between the radii of the spheres placed in the three steps are established by these two parameters, which is easy and convenient in use. In addition, the advancing-front thickness (h) is a key parameter for the efficiency and convergence of the algorithm, and it is suggested based on our experience that h is between 5 and 7.

As introduced previously, the algorithm only includes four input parameters including N_s , f_r , r_{\min}/r_{\max} , and h for clump generation of a polyhedral particle. And it is also very convenient for users to generate the clumps of different numbers of spheres by adjusting N_s . As shown in Figure 3, two clumps of different numbers of spheres for a polyhedral particle are generated by the algorithm when N_s is, respectively, 5 and 10. It can be clearly seen from Figure 3 that the clumps generated by the algorithm can give a good approximation to the particle shape. Table 1 lists all parameters of the algorithm for the polyhedral particle.

In order to show the validity of the algorithm for the representation of convex polyhedral particles, the comparison between the proposed algorithm and the algorithm in PFC 5.0 is conducted. By contrast, it can be seen from Figures 3 and 4 that the clump generated by the proposed algorithm gives a better approximation to the particle shape.

2.3. Mass Properties of a Clump. A polyhedral particle is represented by a clump consisting of multiple spheres as a rigid body in the discrete element modelling. The shape representation of the clump is only used for contact judgment in DEM simulations, while its motion is dominated by the equations of translational and rotational motions, which are determined by its mass properties including the total mass (m), center of mass (\mathbf{x}), and inertia tensor ($I_{3 \times 3}$). To obtain accurate simulation results, the mass properties of the

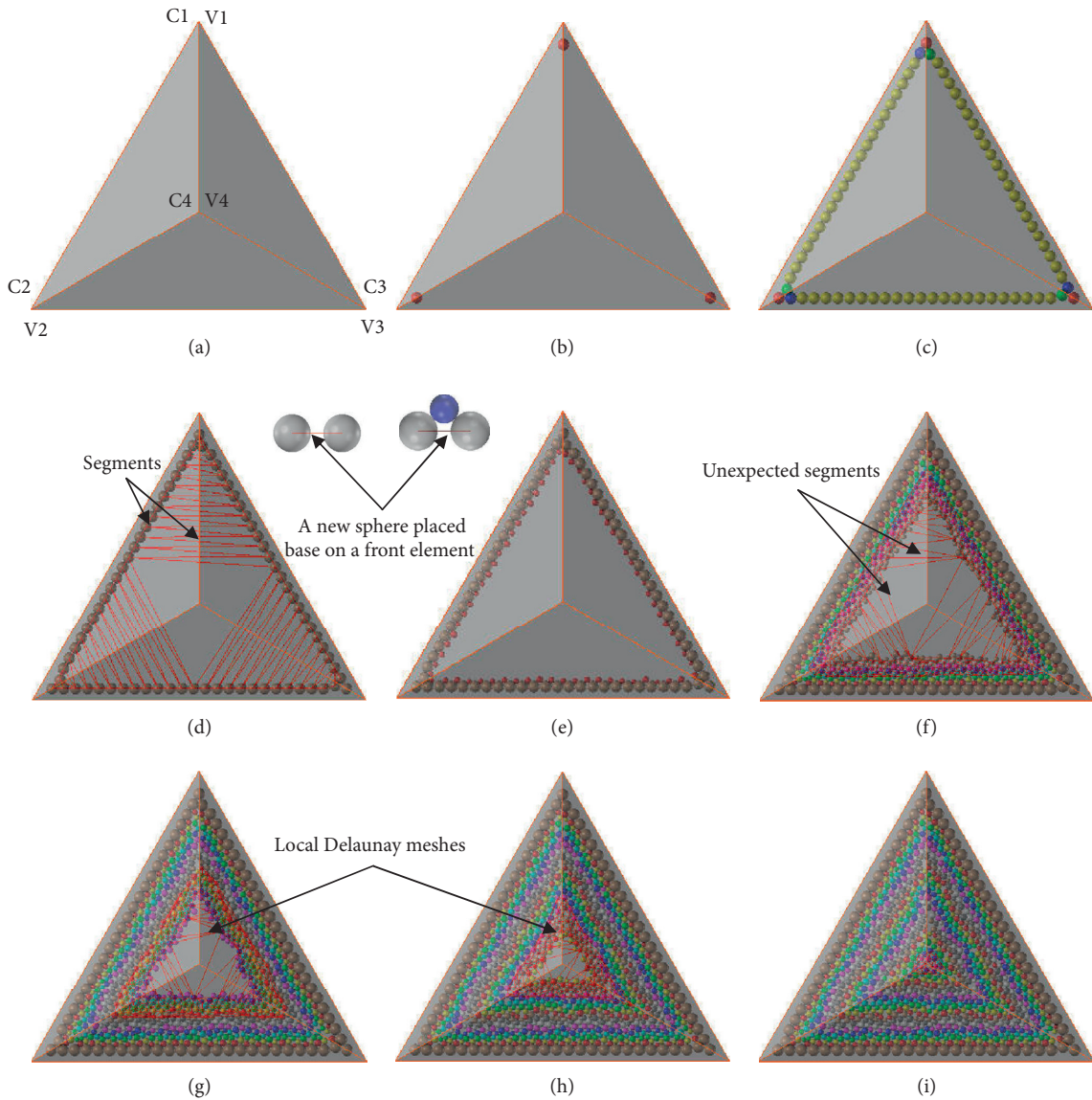


FIGURE 2: A flow chart of the algorithm for clump generation for a polyhedral particle:(a) the tetrahedral particle; (b) placement of spheres at corners; (c) placement of spheres along edges; (d), (e), (f), (g), (h), and (i) placement of spheres on a facet.

clump should give a precise approximation to that of the polyhedral particle. In this work, the mass properties of a clump are determined based on the corresponding polyhedral particle rather than the spheres, while the spheres are only used for shape representation. By separating the mass properties and shape representation, the clump can give an accurate representation of a convex polyhedral particle in terms of particle shape and mass properties.

For a polyhedral particle with uniform mass distribution, its basic mass properties including m , \mathbf{x} , and $I_{3 \times 3}$ can be determined easily by the method proposed in Refs. [28, 29] and are regarded as those of the clump to represent the polyhedral particle.

3. Examples

In the last section, an example is used to show the capability of the algorithm to generate a clump of multispheres for a polyhedral particle, as shown in Figure 3. In this section, another example is used to validate the robust and automatic performance of the algorithm in generating the clumps for an assembly of polyhedral particles. Figure 5 shows a flowchart of clump generation for an assembly of polyhedral particles. In this example, each convex polyhedral particle in the assembly is constructed based on the convex hull of several points randomly selected on the surface of a sphere, as shown in Figure 6(a). As shown in Figure 6(b), the assembly includes 366 polyhedral

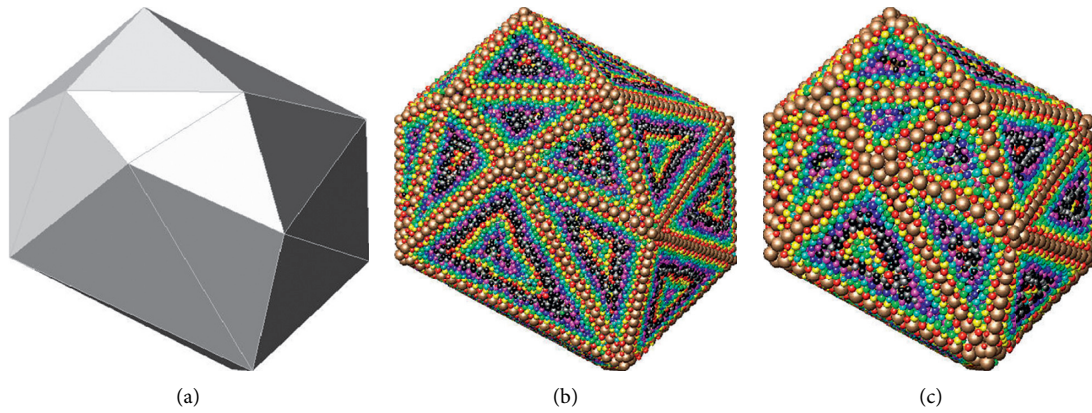


FIGURE 3: Representation of a polyhedral particle by a clump of spheres by the proposed algorithm: (a) the polyhedral particle; (b) Cluster-1; (c) Cluster-2.

TABLE 1: Parameters of the algorithm to represent a polyhedral particle by a clump of spheres.

Clumps	N_s	f_r	r_{max}/r_{min}	h
Clump-1	15	0.5	1.5	6
Clump-2	10	0.5	1.5	6

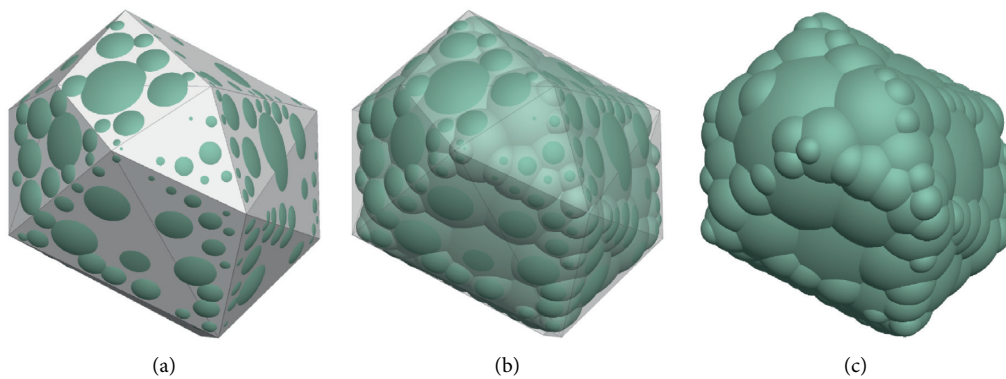


FIGURE 4: Representation of a polyhedral particle by a clump of spheres by the algorithm in PFC5.0: (a) and (b) the polyhedral particle and the clump; (c) the clump.

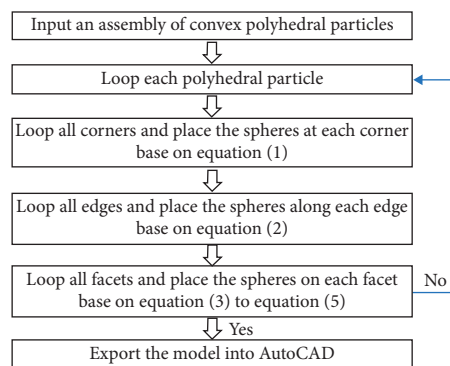


FIGURE 5: A flowchart of clump generation for an assembly of convex polyhedral particles by the proposed algorithm.

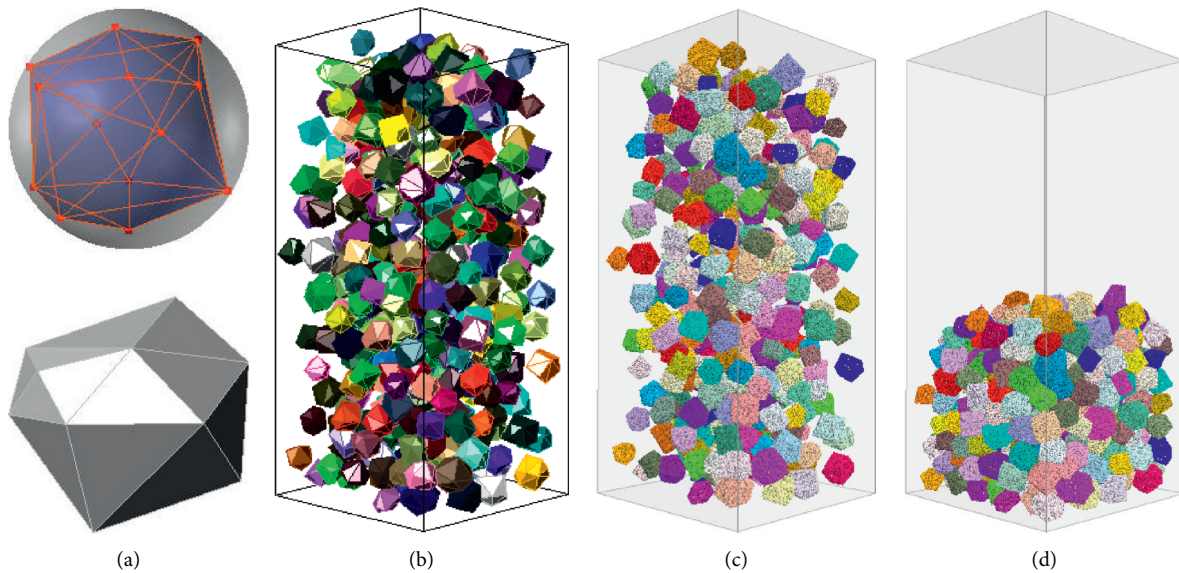


FIGURE 6: Generation of a sphere clump model for an assembly of polyhedral particles: (a) construction of a polyhedral particle based on a sphere; (b) an assembly of polyhedral particles constructed based on different spheres; (c) the corresponding sphere clump model imported in PFC3D; (d) the sphere clump model after settlement under gravity.

particles constructed based on different spheres. Based on the assembly of polyhedral particles, the clump for each polyhedral particle is generated by the algorithm based on the parameters listed in Table 1 for Clump-2. For DEM simulation later, an interface is also implemented into the algorithm to facilitate users to export the clumps into PFC3D [14]. Figure 6(c) shows the clumps imported in PFC3D, and Figure 6(d) shows the clumps after settlement under gravity in PFC3D.

4. Conclusions

- (1) A novel algorithm was extended from 2D to 3D to accurately represent convex polyhedral particles based on the multisphere method for DEM modelling in this work. In 3D, the algorithm includes three steps, and all spheres placed in the three steps are clumped together into a clump in DEM simulations. Clumps can give a better approximation to the particle shape because the spheres are placed tangent to the particle surface in each step. On the other hand, the mass properties of clumps are determined based on the corresponding polyhedral particles to obtain accurate simulation results.
- (2) The algorithm includes four input parameters, where the advancing-front thickness (h) is a key parameter for the efficiency and convergence of the algorithm. Based on our experience, it is suggested that h is between 5 and 7 for complex polyhedral particles.
- (3) A current FORTRAN version of the algorithm is available upon request for academic use, and an interface is provided to import the sphere clump model into PFC3D. It is quite easy and convenient for users to generate sphere clump models by the algorithm for DEM modelling.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work presented in this paper was financially supported by the Open Fund of State Key Laboratory of Hydraulics and Mountain River Engineering (Sichuan University), Grant no: SKHL1725; Open Research Fund of State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin (China Institute of Water Resources and Hydropower Research), Grant no: IWHR-SKL-KF202016; Natural Science Foundation of Hunan Province (CN), Grant no: 2020JJ5732; and National Key R&D Program of China, Grant no: 2017YFC1501100.

References

- [1] P. A. Cundall and O. D. L. Strack, "A discrete numerical model for granular assemblies," *Géotechnique*, vol. 29, no. 1, pp. 47–65, 1979.
- [2] S. J. Lee, *Developments in Large Scale Discrete Element Simulations with Polyhedral particles*, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 2014.
- [3] Z. Lai, Q. Chen, and L. Huang, "Fourier series-based discrete element method for computational mechanics of irregular-shaped particles," *Computer Methods in Applied Mechanics and Engineering*, vol. 362, Article ID 112873, 2020.
- [4] C. Hogue, "Shape representation and contact detection for discrete element simulations of arbitrary geometries," *Engineering Computations*, vol. 15, no. 3, pp. 374–390, 1998.

- [5] S. Johnson, J. R. Williams, and B. Cook, "Contact resolution algorithm for an ellipsoid approximation for discrete element modeling," *Engineering Computations*, vol. 21, no. 2/3/4, pp. 215–234, 2004.
- [6] X. Lin and T.-T. Ng, "A three-dimensional discrete element model using arrays of ellipsoids," *Géotechnique*, vol. 47, no. 2, pp. 319–329, 1997.
- [7] H. Ouadfel and L. Rothenburg, "An algorithm for detecting inter-ellipsoid contacts," *Computers and Geotechnics*, vol. 24, no. 4, pp. 245–263, 1999.
- [8] P. A. Cundall, "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," *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, vol. 25, no. 3, pp. 107–116, 1988.
- [9] R. Hart, P. A. Cundall, and J. Lemos, "Formulation of a three-dimensional distinct element model-Part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks," *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, vol. 25, no. 3, pp. 117–125, 1988.
- [10] C. R. A. Abreu, F. W. Tavares, and M. Castier, "Influence of particle shape on the packing and on the segregation of spherocylinders via Monte Carlo simulations," *Powder Technology*, vol. 134, no. 1-2, pp. 167–180, 2003.
- [11] P. A. Langston, M. A. Al-Awamleh, F. Y. Fraige, and B. N. Asmar, "Distinct element modelling of non-spherical frictionless particle flow," *Chemical Engineering Science*, vol. 59, no. 2, pp. 425–435, 2004.
- [12] J. F. Favier, M. H. Abbaspour-Fard, M. Kremmer, and A. O. Raji, "Shape representation of axi-symmetrical, non-spherical particles in discrete element simulation using multi-element model particles," *Engineering Computations*, vol. 16, no. 4, pp. 467–480, 1999.
- [13] J. Ferrellec and G. McDowell, "Modelling realistic shape and particle inertia in DEM," *Géotechnique*, vol. 60, no. 3, pp. 227–232, 2010.
- [14] X. Garcia, J.-P. Latham, J. Xiang, and J. P. Harrison, "A clustered overlapping sphere algorithm to represent real particles in discrete element modelling," *Géotechnique*, vol. 59, no. 9, pp. 779–784, 2009.
- [15] E. Lozano, D. Roehl, W. Celes, and M. Gattass, "An efficient algorithm to generate random sphere packs in arbitrary domains," *Computers & Mathematics with Applications*, vol. 71, no. 8, pp. 1586–1601, 2016.
- [16] I. Morales, R. Valera, C. Morfa, and M. M. de Farias, "Dense packing of general-shaped particles using a minimization technique," *Computational Particle Mechanics*, vol. 4, no. 2, pp. 165–179, 2017.
- [17] Y. Li and S. Ji, "A geometric algorithm based on the advancing front approach for sequential sphere packing," *Granular Matter*, vol. 20, no. 4, p. 59, 2018.
- [18] N. Das, *Modeling three-dimensional shape of sand grains using discrete element method*, PhD Thesis, Tampa, FL, USA.
- [19] Itasca Consulting Group Inc., *PFC3D (Particle Flow Code in 3 Dimensions)*, Itasca Consulting Group Inc., Minneapolis Minneapolis, MI, USA, 2005pp. 482–498, Version 3.1.
- [20] J. Kozicki and F. V. Donzé, "YADE-OPEN DEM: an open-source software using a discrete element method to simulate granular material," *Engineering Computations*, vol. 26, no. 7, pp. 786–805, 2009.
- [21] J. Elias, "Simulation of railway ballast using crushable polyhedral particles," *Powder Technology*, vol. 264, pp. 458–465, 2014.
- [22] S. J. Lee, Y. M. A. Hashash, and E. G. Nezami, "Simulation of triaxial compression tests with polyhedral discrete elements," *Computers and Geotechnics*, vol. 43, pp. 92–100, 2012.
- [23] J. Escoda, D. Jeulin, F. Willot, and C. Toulemonde, "Three-dimensional morphological modelling of concrete using multiscale Poisson polyhedra," *Journal of Microscopy*, vol. 258, no. 1, pp. 31–48, 2015.
- [24] W.-j. Xu, H.-y. Zhang, Y.-x. Jie, and Y.-z. Yu, "Generation of 3D random meso-structure of soil-rock mixture and its meso-structural mechanics based on numerical tests," *Journal of Central South University*, vol. 22, no. 2, pp. 619–630, 2015.
- [25] Q. Zhang, W.-Y. Wu, Q.-Y. Liu, W. Wang, and Q.-X. Meng, "A novel non-overlapping approach to accurately represent 2D arbitrary particles for DEM modelling," *Journal of Central South University*, accepted, 2015.
- [26] J. Liu, B. Yun, and C. Zhao, "An improved specimen generation method for DEM based on local Delaunay tessellation and distance function," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 36, no. 5, pp. 653–674, 2012.
- [27] J.-F. Jerier, V. Richefeu, D. Imbault, and F.-V. Donzé, "Packing spherical discrete elements for large scale simulations," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 25-28, pp. 1668–1676, 2010.
- [28] D. Eberly, *Polyhedral Mass Properties (Revisited)*, www.magic-software.com/Documentation/PolyhedratMassProperties.pdf, 2003.
- [29] B. Mirtich, "Fast and accurate computation of polyhedral mass properties," *Journal of Graphics Tools*, vol. 1, no. 2, pp. 31–50, 1996.