文章编号：1000-8608(2011)02-0236-08

Three-dimensional particles generation algorithm

LIU Jun*, GENG Qing-dong

(State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: Particles generation is the pre-step of discrete element simulation of practical problems. One advancing front face (AFF) algorithm for three-dimensional sphere/polyhedron particles packing is advanced. It can be used to constructively generate a random initial packing of spheres / polyhedrons with different radii within a 3D domain. Firstly, select three radii randomly and generate three spheres which contact with each other. The centroids of the initial three spheres form two advancing front faces. Then, select an AFF as the active face, randomly select a radius and generate a new sphere which contacts with the three old spheres forming the current active AFF. Update AFF list, select a new active AFF and generate a new sphere until all needed spheres have been generated or the domain has been filled in. If polyhedron particles are required, vertices are generated randomly on the sphere surface and these points are connected to form a convex hull. Go through all spheres and polyhedron particles are formed and filled in the 3D domain. The generated packing is not a globally optimal arrangement, but a locally highest density is achieved from the algorithmic point of view. The performance of the algorithm is illustrated by several examples. The major benefit of this algorithm is the significant reduction of CPU time required for the preparation of an initial discrete object configuration in discrete element method (DEM)/discontinuous deformation analysis (DDA) simulations. It is demonstrated that it takes only 86 s for the proposed algorithm to generate 50 thousand particles on a PC with one 2.8 GHz processor.

Key words: particles generation; advancing front face; discrete element method (DEM); discontinuous deformation analysis (DDA); granular material

0 Introduction

The discrete element method (DEM) / discontinuous deformation analysis (DDA), coupled with the finite element technique, has become a powerful numerical approach, particularly for engineering problems involving discrete/discontinuous phenomena [1-3]. Typical examples include process simulation (e. g. granular assembly, powder compaction and particle dynamics) and fracture damage modeling (e. g. cohesive frictional materials, rock

Received by: 2009-04-15; **Revised by**: 2011-01-20.

Supported by: Program of Innovation Group of Liaoning Province (Grant No. 2009T017); "973" National High Technology Research and Development Program of China Project (Grant No. 2010CB731502).

Corresponding author: LIU Jun* (1972-), Male, Doc., Prof., E-mail:junliu@dlut.edu.cn.

blasting, mining applications and high speed projectile impact). Using simplified geometric entities, such as circular disks/spheres, to model discrete objects has been demonstrated to provide an acceptable approximation to many complex physical phenomena [4]. The pre-step in the discrete element simulation of practical problems often requires the generation of discrete objects packed in a form which can represent various realistic situations. Particle packing/arrangement has been one of the most attractive research fields in recent years.

In the case of equal size objects and regular containers, the initial packing of spheres may be implemented by regular arrangement. In many practical situations, however, different size spheres have to be used in the simulation. Preparation of an initial distribution of a large number (sometimes, it is up to millions) of such spheres in a realistic (random) manner is not trivial.

Currently, many approaches have been used in general practice. Several of these approaches are briefly reviewed in the following [5-11].

The first approach is random generation. The particles are sequentially generated by determining their central locations randomly. When the newly generated particle conflicts or overlaps with the existing ones, they are then rejected or removed to a suitable position. However, with the increase of generated particles number, the times of rejections or movement also increase significantly. So the disadvantage of this approach is the time consumption of the whole procedure. Another disadvantage is that the predefined particle size distribution may not be achieved due to the rejection of many particles.

The second approach adopts a two-stage procedure. In the first stage, all spheres are assumed to have a maximum radius with their center being placed in a regular style. While for many particles, the actual radii are smaller. There are usually relatively large gaps between the spheres. In the second stage, the spheres are compressed to reduce some gaps by pushing the boundaries toward the spheres in one or several directions.

In the third approach, a hopper is used to mix different sizes of spheres together. The spheres are initially packed regularly in layers at the top of the hopper, and the spheres with larger radii are placed above. Under the action of gravity, the spheres fall through the chute and settle down with a random mixture of different sizes of spheres in the container beneath the hopper.

As to the first approach, many spheres are not getting in touch with some of the others. This reduces the density of the sample. Both the compression of the boundary and the movement of spheres under gravity force in the latter two approaches mentioned above are normally simulated by the same numerical procedure employed in the DEM / DDA. Real forces are involved in the packing procedure for these approaches. However, the CPU time required to undertake this pre-processing phase can be time-consuming.

Han, *et al*. [12] also suggested a compression algorithm of randomly filling a domain with spheres of different sizes. The algorithm is a two-step procedure. Firstly, spheres of different radii are randomly distributed or regularly positioned to create an initial packing within a geometric domain, and the initial packing is compressed to a much denser condition. Secondly, the remaining space in the first step is refilled. If successful, it compresses the spheres by using the technique in the first step and repeats the procedure until the domain is full. Cui, *et al*. [13] proposed to generate small balls of a given number firstly, and then made the balls to grow larger and larger by increasing their radii. The process will stop when the domain is fully filled. At last, with the compression of the boundary and the action of the gravity, the balls will fall, roll and come to a stable and dense condition. This particle generation algorithm is widely used in the particle flow code (PFC).

Feng, *et al.* [4] proposed an advancing front-based algorithm to constructively generate a random initial packing for disks with different radii within a 2D domain. Lit. [14] extended the methodology to other shaped discrete objects including elliptic particles and convex polygons, and as well to 3D spheres with different sizes. But the details of the algorithm in 3D particles generation aren′t presented in his paper, and there is no detailed solutions presented for the situation, in which some spheres may penetrate through the facets without necessarily overlapping existing spheres.

In this paper, a random filling approach, termed advancing front face (AFF) algorithm, is presented in a 3D domain based on Feng′s ideology in detail to reduce the CPU time required for the preparation of an initial discrete object configuration in DEM / DDA simulations.

1 Spheres generation

Firstly, select three random radii and generate three spheres which contact with each other. The centroids of the initial three spheres form two AFFs. Then select an AFF as the current active face, select a random radius and generate a new sphere which contacts with the three spheres forming the current active AFF. Update AFF list, select a new active AFF and generate a new sphere until all needed spheres have been generated or the domain has been filled in. If polyhedron particles are required, vertices are generated randomly on the sphere surface and these points are connected to form a convex hull. Go through all spheres and polyhedron particles are formed and filled in the domain.

1.1 The first three spheres and initial front faces

The first three spheres, denoted as B_1, B_2 and B_3, can be generated to have the densest packing by being contact with each other in the center of the domain as shown in Fig. 1. Firstly, select randomly a radius and locate the first sphere (B_1) at the center of the domain. Then, randomly select a radius and put the second sphere (B_2) to contact with the first one. Lastly, randomly select a radius as the radius of the third one and put it to contact with the existing two spheres. A plane ($B_1B_2B_3$) is formed by connecting their centers together, and two faces ($B_1B_2B_3$ and $B_3B_2B_1$) of the plane are obtained. The two faces are defined to have directions, with $B_1 \rightarrow B_2 \rightarrow B_3$ or $B_3 \rightarrow B_2 \rightarrow B_1$ as positive. And the oriented faces $B_3B_2B_1$ and $B_1B_2B_3$ are termed the initial advancing front faces and recorded in front faces list. The region surrounded by faces is considered to be or have been occupied by spheres. The defined directions of the faces ensure that any new sphere should be placed on the outward normal side when traveling along the front faces.

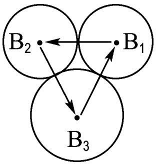


Fig. 1 The first three spheres

With the initial front faces established, new spheres can be generated to fill the space by incrementally advancing the front face until the original domain has been nearly full-filled or the needed spheres have been generated. Details of the generation of a new sphere and the updating of the current active front faces are described below.

1.2 New sphere generation and front faces update

Starting from the two initial faces, one face, e. g. $B_1B_2B_3$ is chosen as the current active face (CAF). At first, randomly select a radius and generate sphere B_4, which is in contact with B_1, B_2 and B_3 and lies in the outward normal direction of the face $B_1B_2B_3$, as shown in Fig. 2. After B_4 is generated, the current active face $B_1B_2B_3$ is deleted from the front faces list and three new front faces $B_1B_2B_4$, $B_2B_3B_4$ and $B_3B_1B_4$ are added to the front faces list. Now there are four front faces available in the list. Then, the last front face (e. g. $B_3B_1B_4$, which is the last one in the front faces list) is chosen as the current face to generate new sphere, and then, a new sphere B_5 is generated in a similar manner. The current active face is deleted from the list and another three new front faces are then added to the list. The number of front faces in the list is increased to six.

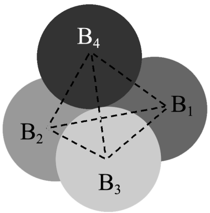


Fig. 2　Generation of the fourth sphere

The above procedure will be performed until the whole domain is filled in or all the particles are used up. In general cases, new generated sphere may partly locate out of the domain or overlap with the existing spheres in the domain. Several measures are therefore taken to deal with these problems. Now the algorithm of the spheres generation is proposed as below.

(a) Generate the first three spheres and form the front face list.

(b) Select a front face as the current active face and select randomly a radius as the radius of the new generated sphere. Determine the central coordinates of the new sphere B_i, which is in contact with the three spheres (B_a, B_b and B_c) on the current active face. This is a simple geometric problem. As shown in Eq. (1), (x, y, z, r) are the center coordinates and radius of the new ball and (x_i, y_i, z_i, r_i) (i=1,2,3) are the center coordinates and radius of the three balls on the active front face. Three unknown variables of the central coordinate of the new sphere can be solved through Eq. (1). Two sets of solutions are obtained. One is located in the inside of the face and the other one is located in its outside. The latter one needed as the inside has been filled by the existing balls. If the center point is located in the outside of the face, the determinant in Eq. (2) should be greater than zero. Illustration is shown in Fig. 3.

$$
\begin{aligned}
(x-x_1)^2+(y-y_1)^2+(z-z_1)^2 &= (r+r_1)^2\\
(x-x_2)^2+(y-y_2)^2+(z-z_2)^2 &= (r+r_2)^2\\
(x-x_3)^2+(y-y_3)^2+(z-z_3)^2 &= (r+r_3)^2
\end{aligned}
\tag{1}
$$

$$
\begin{vmatrix}
1 & x_1 & y_1 & z_1\\
1 & x_2 & y_2 & z_2\\
1 & x_3 & y_3 & z_3\\
1 & x_4 & y_4 & z_4
\end{vmatrix} > 0
\tag{2}
$$

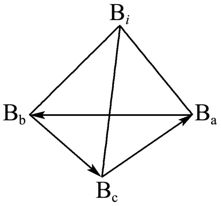


Fig. 3　Generation of the ith sphere

(c) Check if the new sphere, B_i, is located in the outside of the domain or overlaps with the existing spheres. If any of these two conditions appears for the new sphere with the selected radius, it is then changed to a relatively smaller radius. This process continues until a suitable sphere is found, which means there is no overlapping and outside locating. If all the given radii are tried and failed in the above process, which means, based on the current active face, even the smallest sphere can not be filled into the domain. The space here is then treated as too small to be filled, and the current active face is then deleted from the face list. This phenomenon happens when the current active face reaches the boundary.

(d) If all the front faces are used or all the needed particles are generated, the filling process is finished.

The spheres generation procedure in a container introduced above can be described by the flow chart in Fig. 4.

The following features can be derived from the above algorithm: A locally optimal packing is achieved when adding a new sphere since it is in contact with the three spheres associated with the current active face. As intermediate active faces are only temporarily presented and removed after new spheres are successfully generated, the available current faces always form a closed polyhedron (may be concave), and leave no gap between spheres on the current active face. The region inside the front faces represents the sphere-filled domain and any new sphere is placed outside the region.

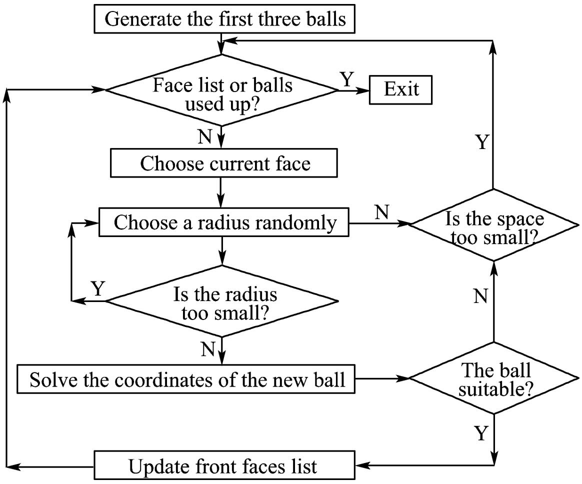


Fig. 4 Flow chart of spheres generation

2 Further issues

2.1 Face visit sequence

Theoretically, the faces on the front can be visited in any order but a logical way of choosing an active face will benefit. Here the new generated front face, which has the largest face list number, is chosen. First, it will be easy for programming. Second, it will benefit the domain filling on the behalf of filling density.

2.2 Further tightening

As the current algorithm can achieve only a local optimal packing, the spheres may be further packed by means of gravity compaction and boundary compression. This further tightening may also be able to eliminate local instability of some spheres.

2.3 Extension to other domain shapes

The algorithm is discussed on the basis of a cuboids domain to be filled. It can be easily extended to other domain shapes, such as sphere, cylinder or even more complex geometric shapes. What it needs to do is just to adjust the boundary conditions.

2.4 Disadvantage

The disadvantage of this algorithm is that relatively large gaps may appear near the boundary. In order to avoid this disadvantage, small radius group should be supplied to fill in this space.

3 Generation of polyhedron in sphere particle

After the generation of spheres, if polyhedrons are needed, a polyhedron with all vertices on the sphere surface can be generated in each sphere. In order to assure that the volume of the polyhedron in a sphere is not too small in contrast to the sphere, several distance conditions are defined. The procedure of the polyhedron′s generation is introduced as follows:

(a) The vertex number (N) of a polyhedron is determined randomly, which should not be smaller than four so as to form a polyhedron.

(b) The first vertex of the polyhedron is randomly located on the sphere surface.

(c) The second vertex is located randomly but it should be not too close to the first one.

(d) The third vertex is located randomly but it should be not too close to the former two points and not too close to the line connecting the former two vertices.

(e) The fourth vertex is determined randomly, and is kept far enough from the former three vertices and the plane formed by the former three vertices. If within many loops (e.g. 500 iterations), the fourth vertex can not be obtained, then the location of the fourth vertex is directly determined with the largest distance to the plane of the former three vertices.

(f) The following vertices are randomly generated one by one, which should be with a distance limit to the formers.

(g) All the vertices on the sphere surface form a convex hull that is the polyhedron.

Steps (a) to (g) are repeated until all spheres have been changed to polyhedrons.

The chart of the polyhedron's generation algorithm is shown in Fig. 5.

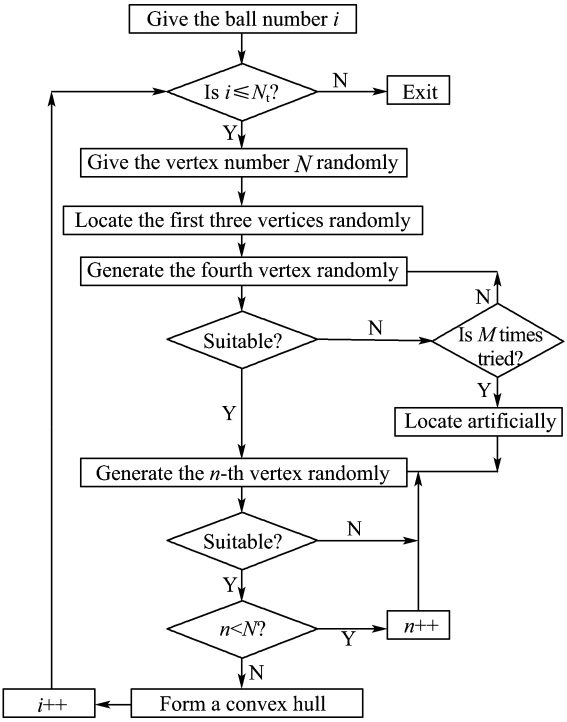
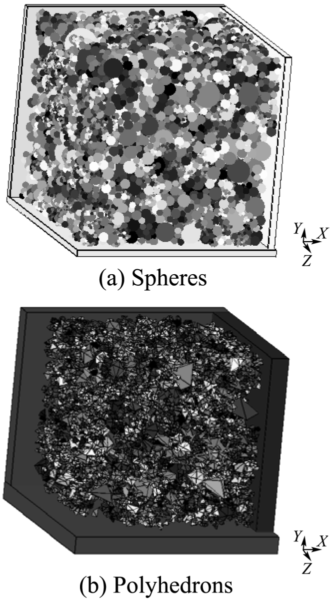

Give the ball number *i*
Is *i*≤*N*t?
N
Exit
Y
Give the vertex number *N* randomly
Locate the first three vertices randomly
Generate the fourth vertex randomly
N
Suitable?
N
Is *M* times tried?
Y
Y
Locate artificially
Generate the *n*-th vertex randomly
Suitable?
N
Y
n<*N*?
Y
n++
N
i++
Form a convex hull

Fig. 5　Flow chart of a polyhedron generation

4　Examples

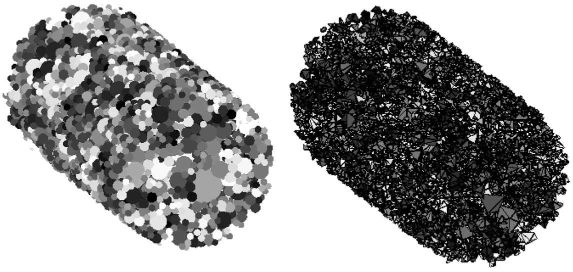
There are four examples to illustrate the validity of the algorithm. The first container is a cube with the side length of 10 cm. This sample has four fractional groups, and their diameters are 2.0 cm, 1.0 cm, 0.6 cm and 0.4 cm. The results of spheres generation and polyhedrons generation are shown in Fig. 6. In the second example, the container is a cylinder, whose height and diameter are 20 cm and 10 cm respectively. Also four fractional groups are used as above mentioned. The results are shown in Fig. 7. In the third example, it is a cuboid container. The side length of the bottom square is 10 cm, and the height of the container is 20 cm. In order to have a denser simulation, ten fractional groups are considered. And the minimum and the maximum diameter are 0.5 cm and 5.0 cm respectively. The results of generated spheres and polyhedrons are shown in Fig. 8. The fourth example is shown in Fig. 9. The container is more complicated than the previous three examples. Only one diameter is set to generate spheres.



(a) Spheres

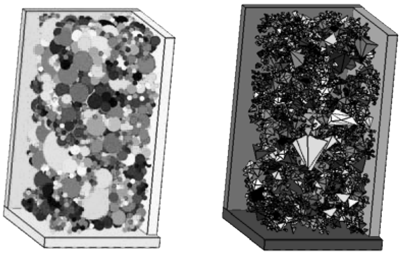
(b) Polyhedrons

Fig. 6　Particles generation in a cube



(a) Spheres　(b) Polyhedrons

Fig. 7　Particles generation in a cylinder



(a) Spheres　(b) Polyhedrons

Fig. 8　Particles generation in a cuboid

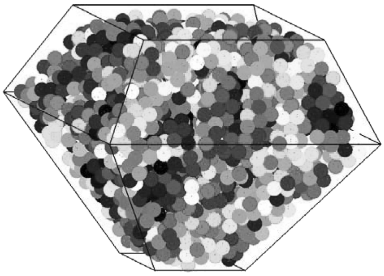


Fig. 9 Particles generation in a complicated container

This paper also gives a comparison between the practical gradation and the simulated gradation. The container is a cube with the side length of 30 cm. The simulation result of the particles is shown in Fig. 10, and the sieving data of the material is listed in Tab. 1. From the Tab. 1, two coefficients of the sample can be calculated, and they are the coefficient of uniformity C_u and the coefficient of curvature C_c. $C_u=5.29$, which is larger than 5, and $C_c=1.32$, which is between 1.0 and 3.0. So the sample is uniformly and well graded. The gradation comparison between the practical and simulated samples is plotted out in Fig. 11. It is shown that the two curves meet well.

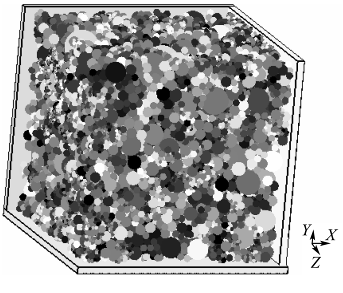


Fig. 10 Particles generation simulation result

Tab. 1 Result comparison

D / cm	Percent finer f	
	Practical	Simulated
⩽0.1	0.002 4	0.003 8
⩽0.2	0.031 2	0.057 2
⩽0.5	0.192 7	0.243 8
⩽1.0	0.316 5	0.438 1
⩽1.5	0.398 4	0.570 8
⩽2.0	0.614 3	0.723 6
⩽3.0	0.846 8	0.801 3
⩽4.0	0.917 9	0.896 0
⩽5.0	0.979 0	0.974 6
⩽7.0	1.000 0	1.000 0

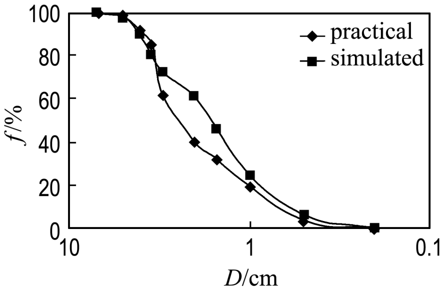


Fig. 11 Comparisons between practical and simulated samples

5 Conclusion

In this work, a new algorithm, called advancing front face algorithm, for spheres and polyhedrons generation is advanced. Although the generated packing is not a global optimal arrangement, it has achieved a locally highest density from the algorithmic point of view and should be sufficient to represent a practical situation, as illustrated in the examples presented.

The performance of the algorithm and the final spheres distribution may be further improved by taking several additional issues into consideration in implementation as discussed. Additional compression and compaction can be applied to further tightening the packing if a higher density is required in practice. Actually in many cases the initial stage of DEM / DDA computation can act as a means of further tightening.

The major benefit of this algorithm is the significant reduction of the CPU time required for the preparation of an initial discrete object configuration in DEM / DDA simulations. It is demonstrated that it takes only 86 s for the advanced algorithm to generate 50 thousand spheres on a PC with one 2.8 GHz processor, while using a hopper to mix spheres may take a few days to achieve the same goal.

References:

[1] CUNDALL P A, STRACK O D L. A discrete numerical model for granular assemblies [J].

Geotechnique, 1979, **29**(1):47-65

[2] SHI G H. Discontinuous deformation analysis: a new numerical model for the statics and dynamics of block systems [D]. Berkeley: Department of Civil Engineering, University of California at Berkeley, 1988

[3] LIU Jun. Three-dimensional discontinuous deformation analysis coupled with finite element method [D]. Dalian: Dalian University of Technology, 2001

[4] FENG Y T, HAN K, OWEN D R J. Filling domains with disks: an advancing front approach [J]. **International Journal for Numerical Methods in Engineering**, 2003, **56**:699-713

[5] VISSCHER W M, BOLSTERLI M. Random packing of equal and unequal spheres in two and three dimensions [J]. **Nature**, 1973, **239**:504-507

[6] ANISHCHIK S V, MEDVEDEV N N. Three-dimensional apollonian packing as a model for dense granular systems [J]. **Physical Review Letters**, 1995, **75**:4314-4317

[7] DU Cheng-bin, SUN Li-guo. Numerical simulation of concrete aggregates with arbitrary shapes and its application [J]. **Shuili Xuebao**, 2006, **37**(6):662-673 (in Chinese)

[8] CHAN S K, NG K M. Geometrical characteristics of the pore space in a random packing of equal spheres [J]. **Powder Technology**, 1988, **54**:147-155

[9] COELHO D, THOVERT J F, ADLER P M. Geometrical and transport properties of random packings of spheres and aspherical particles [J]. **Physical Review**, 1997, **E55**:1959-1978

[10] SOBOLEV K, AMIRJANOV A. The development of a simulation model of the dense packing of large particulate assemblies [J]. **Powder Technology**, 2004, **141**:155-160

[11] STOYAN D. Models of random systems of non-intersecting spheres [C] // **Proceedings of Prague Stochastics'98**. Prague: JCMF, 1998:543-547

[12] HAN K, FENG Y T, OWEN D R J. Sphere packing with a geometric based compression algorithm [J]. **Powder Technology**, 2005, **155**:33-41

[13] CUI L, O' SULLIVAN C. Analysis of a triangulation based approach for specimen generation for discrete element simulations [J]. **Granular Matter**, 2003, **5**(3):135-145

[14] FENG Y T, HAN K, OWEN D R J. An advancing front packing of polygons, ellipses and spheres [C] // **3rd International Conference on Discrete Element Methods**. New Mexico: ASCE, 2002:93-98

三维颗粒生成算法

刘　君*，　耿庆东

（大连理工大学 海岸和近海工程国家重点实验室，辽宁 大连　116024）

摘要： 颗粒生成是进行散粒体细观数值模拟的前提.为快速生成指定级配的颗粒并减少自由堆积过程，针对三维球形及多面体颗粒生成，改进了颗粒生成的前进面算法.其基本思想是：随机选取 3 种粒径，生成 3 个相互接触的球，3 个球的球心构成初始的 2 个前进面；随机选取前进面和一种粒径，在前进面的外侧生成新的球，使新球与构成前进面的 3 个球相切；更新前进面，进而生成新的球，如此往复直到生成全部颗粒或填满容器.在球的表面随机生成点，这些点构成一个凸锥-多面体，进而生成多面体颗粒.编制了前进面法颗粒生成程序，5 个算例证明了该算法的高效性.采用此法生成的颗粒不一定是全局最优的排列，但可以保证局部最密分布.该算法的最大优点是可以大大减少 CPU 运算时间.在主频为 2.8 GHz 的 PC 机上生成 5 万个颗粒只需 86 s.

关键词： 颗粒生成；前进面；离散元法(DEM)；非连续变形分析(DDA)；散粒体

中图分类号： TU44　　**文献标志码：** A

收稿日期： 2009-04-15；　**修回日期：** 2011-01-20.

资助项目： 辽宁省创新团队科研基金资助项目(2009T017)；"九七三"国家高技术研究发展计划资助项目(2010CB731502).

作者简介： 刘　君*(1972-)，男，博士，教授，E-mail：junliu@dlut.edu.cn.