Sphere Packing and Applications to Granular Structure Modeling

Azeddine Benabbou[1], Houman Borouchaki[2], Patrick Laug[3], and Jian Lu[4]

[1] INRIA, BP 105, 75153 Le Chesnay cedex, France
azeddine.benabbou@inria.fr
[2] University of Technology of Troyes, 12 rue Marie Curie BP2060, 10010 Troyes Cedex, France
houman.borouchaki@utt.fr
[3] INRIA, BP 105, 75153 Le Chesnay cedex, France
patrick.laug@inria.fr
[4] Hong Kong Polytechnic University, Kowloon, Hong Kong
jian.lu@inet.polyu.edu.hk

Summary. This paper presents a new sphere packing algorithm for generating granular structures in either two or three dimensions. Such a structure is often modeled by a parallelepiped containing spherical balls in three dimensions or by a rectangle filled with disks in two dimensions. The grains (spherical balls or disks) are separated by interfaces with specific thicknesses, called grain boundaries, and their size correspond to a size distribution experimentally obtained. The geometrical modeling of such a structure, which consists in determining the repartition of the set of disjoint grains according to the above specifications, can then be considered as the classic sphere packing problem. The proposed method is a constructive algorithm based on an advancing-front approach, which is well known in a different context, namely mesh generation. Since the use of the advancing-front approach leads to empty areas near front collisions, a point relocation algorithm, using weighted Delaunay triangulation, is then introduced to balance the local density on the whole structure. Moreover, we propose a method to transform spherical balls (disks) into polyhedral (polygonal) cells similar to the real grain shape. Numerical 2D and 3D examples are provided to illustrate the capability and the efficiency of our approach. The algorithms and techniques presented here can find applications to generate aggregates in all fields concerned by the granular structures such as metallurgy, ceramics, soil science, cements, biomechanics, etc.

Keywords: sphere packing, advancing-front approach, granular, nanostructures, Laguerre diagram, quality meshes.

1 Introduction

The geometrical modeling of granular structures represents a key step in the simulations and studies of their behavior (mechanical, electrical, thermal, ...). This modeling is often based on experimental data provided by microscopic analysis in the form of a parallelepipedic (rectangular) sample and the distribution of the grain sizes within the considered structure. The physical results of simulations

on these structures, by the molecular dynamics models or the finite element method or other methods of resolution, strongly depend on the geometric models provided. Hence, the geometrical modeling of these structures is a crucial task requiring special attention. Usually, a sample of a granular structure is modeled by a parallelepiped (rectangle) containing grains modeled by spherical balls (disks). Beyond the large simplifications in the geometrical calculations, this modeling provides an acceptable model in many cases where the computing with the real grain geometries (complex shapes) is very difficult to achieve [1]. Using this approximation, the geometrical modeling of granular structures is then very similar to the classic sphere (circle) packing which consists in filling a domain with spheres (circles) whose radii follow a size distribution. In our concern, we are particularly interested in the modeling and the meshing of nanomaterials considered as granular structures. This is due to their considerable scientific and industrial potentialities. Indeed, it is experimentally proved that the physical properties of nanomaterials are very different, and often better, than those of ordinary materials. This quality improvement of materials is due to changes of their structures on small scales, in other words to their nanostructuring. The structure of an ordinary material is constituted by a multitude of small polyhedral volumes called grains whose size usually ranges from 2 to 20 micrometers, separated by grain boundaries with specific thicknesses. Compared to the configuration of an ordinary material, the grain size in a nanomaterial is 1000 times smaller and the proportion of grain boundaries is more important. These two features, the grain size and the proportion of grain boundaries, are mostly responsible for changes in the material behavior when its structure changes from the micrometer to the nanometer scale [2, 3].

In our concern, the nanostructures studied are issued from the SMAT (Surface Mechanical Attrition Treatment) process [2], considered as granular structures.

2 Sphere Packing and Granular Structure Modeling

2.1 Sphere Packing

The term of sphere (circle) packing is generally used to evoke the mathematical study of the arrangements of non-overlapping spheres (circles) which fill a given domain. A typical sphere packing problem is to determine the repartition of the particles (spheres or circles) which maximizes the density in the domain. Indeed, the density is the proportion of space filled by the particles having this repartition. This density depends on the particle size distribution, the way these particles are packed and the volume of the domain to fill. In practice, to determine the efficiency of a filling algorithm to generate structures with high densities, we consider it in the particular case where the particles have an identical size and where the domain has a sufficiently large volume. Indeed, in this particular case, the density of the generated structure can be compared with $\frac{\pi}{\sqrt{18}} \simeq 0.74048$, the density of the cubic close packing arrangement in the three-dimensional Euclidean space (Kepler conjecture), and $\frac{\pi}{\sqrt{12}} \simeq 0.9069$, the density

of the hexagonal packing arrangement in the two-dimensional Euclidean space. Note that the maximal density value in 3D concerns regular arrangements only. For irregular arrangements in 3D (which is usually the case in the sphere packing algorithms), the highest density (in the particular case where the particles have the same size) is generally about 0.64, which corresponds to the *jammed* configuration.

Several approaches have been proposed to fill a domain with particles in two and three dimensions. These methods can generally be classified into two major families: dynamic techniques and constructive techniques. The dynamic methods are based on the motion and/or the resizing of the particles. In these techniques, two approaches are essentially used. A purely geometric approach, where calculations are based only on the particles positions and sizes [4, 5, 6, 7], and a second approach in which physical properties are introduced allowing each particle to find an equilibrium position depending on its interactions with the other particles [8, 9, 10, 11]. Usually, the dynamic methods are very costly in terms of computing time, because the position and/or the size of each particle are/is modified during the whole filling process. The second family of methods is the constructive techniques, in which calculations are purely geometric: The position and often the size of each particle are kept throughout the filling process. In this kind of techniques, we can distinguish the methods where the filling in a given step depends on the situation of the system in the previous steps (like the advancing-front approach) [12, 1, 13, 14, 15], and the other pseudo-constructive methods where the repartition of the particles is randomly generated [16, 17, 18] or based on a pre-built triangulation [19] for example. In many contexts, the constructive techniques are considered as more advantageous than the dynamic ones.

2.2 Granular Structure Case

A granular domain, in particular a nanostructure, can be described by a parallelepiped (rectangle) containing spherical balls (disks) whose sizes are defined by a given distribution. The spherical balls (disks) represent the grains and the spaces between them the grain boundaries. The geometrical modeling of these structures can be considered as an application of the sphere (circle) packing when spheres (circles) are modeling grains. In addition to the conformity with grain size distribution, in the nanostructure case, the grains must be separated by grain boundaries with specific thicknesses and the generated aggregate must be irregular (no periodicity).

In terms of algorithmic aspects, further constraints must be considered. In fact, the algorithm for generating models of granular structures must also be fast and robust, especially in the three dimensions case.

Modeling grains by spherical balls (disks) obviously creates empty areas in the structure, *a fortiori* if the filling method is an advancing-front type. Since these empty areas, as well as their non homogeneous repartition, do not exist in the real structure, fatal errors may occur during the physical simulations. Thus, in most cases, an optimization algorithm, to equally distribute these empty areas on the

whole structure, must be applied after the filling process. The resulting structure may be transformed, using the weighted Delaunay triangulation techniques, into an aggregate of grains in the form of Laguerre cells closer to reality.

3 Geometrical Modeling Scheme

3.1 Filling Algorithm (Sphere Packing Algorithm)

The algorithm we propose to generate granular structures, which is a new sphere packing algorithm, is a constructive method based on an advancing-front approach. The term "constructive" means that the construction at a given step depends and uses the result of the previous steps, and that particles keep their size and position during the filling process. The advancing-front approach used in our algorithm is a new variant of a classical approach in the context of meshing. Indeed, the idea of the advancing-front method is to mesh a domain by an iterative inwards spreading of a front initialized on its boundary until its complete recovery [20, 21]. The front is a set of elements (segments in 2D or triangles in 3D) connecting the vertices of the mesh. The advancing-front approach consists in using the front elements (a previous construction) to create and insert the new points in the mesh. At each step, an element of the front is selected to propose the position of the new point according to a given criterion. If the new point is valid, then the selected front element is deactivated and new front elements are created, connecting this new point to the points of the deactivated front element and possibly to other points of the mesh. A new point is said valid (and then inserted in the mesh) if it satisfies the specified criterion and if the possible created front elements do not intersect the current front elements. In the event that one or more points of the current mesh better fit the specified criterion using the selected front element (compared to the new point), then the best of these points is chosen and connected to the points of the selected front element. The convergence of the advancing-front method strongly depends on how the elements are selected from the front, the identification of the optimal points and the validation of the created front elements. The advancing-front method remains highly empirical, unlike Delaunay-type methods for example [22, 23], which are based on mathematical formulations. Thus, the advancing-front methods are often intuitive and their convergence, especially in the 3D case, cannot be guaranteed by a rigorous formalism. The use of the advancing-front approach, in our context, requires to introduce some new concepts in order to ensure its convergence.

In the analogy between this meshing advancing-front approach and our sphere packing algorithm, a node of the mesh becomes a particle in our geometrical modeling. The sizes of these particles are defined using a given grain size distribution. In our case, a front element is a virtual triangle (resp. segment) connecting the centers of 3 spherical balls (resp. 2 disks). A spherical ball (resp. disk) is the association of a center position and a radius defined according to the specified size distribution. The location of each spherical ball (resp. disk) is determined, using an element of the front, to maximize the local density with the particles (spherical balls or disks) constituting this front element. This position is valid if

this particle does not overlap any of the existent particles and if it is completely contained in the domain. A similar sphere packing algorithm have been proposed by many authors including Yamakawa et al. [24], the proposed method defines the metric using a sphere (bubble) packing possibly overlapping approach. Our algorithm can be summarized by the following scheme:

1. Create an initial front
2. Do while the front still contains active elements
   a) Generate a random radius according to the grain size distribution
   b) Determine the level l of the front (= the first active element level),
   c) Do the following for the front of level l
      i. Select an active element if its level l_e is less than or equal to l
      ii. Place the new particle P_{new} using the selected front element e_s
      iii. Control the validity of P_{new}
      iv. If P_{new} is valid then
         - Deactivate e_s
         - Connect P_{new} to e_s particles [1] and go to (a)
      v. If the position of P_{new} is not valid and all the active elements with a level $\leq l$ have been tested to place P_{new} without success then
         - Deactivate the first active element e_f in the front
         - Determine the existent particle maximizing the local density with e_f particles
         - Add possible new elements to the front [2] and go to (b)
   d) End do
3. End while

The first step of our algorithm is to create an initial front by connecting the particles which are placed on the interior side of the domain boundary. Then the filling is made by an inwards spreading of the front. In figure 1, an illustration of a filling example in 2D is shown. Figure 2 shows a filling example using our algorithm in 3D by illustrating the increasing particle level towards the domain center (each color corresponds to a level in figure 2).

The convergence of the new variant of the advancing-front method is assumed by the front level concept. Indeed, the level of a front is defined by the level of its first active element. A front element is said active if it has not been deleted (deactivated) from the front, either because it has been used to insert a new particle or because it has been the first active element of a front in the case where no element of this front could insert the new particle. The level of an element is defined by the sum of the levels of its particles (spherical balls or disks). The particle level ranges from 0 (the level of the initial front particles) to a level n corresponding to the level of the last front particles. The level of

[1] This connection will create new front elements (two in 2D and three in 3D) with a level $\geq l_e$.

[2] For each set s containing $d-1$ particles of e_f, (d is the space dimension), do: if the determined particle is not connected to the particles of s, then connect it if the level of the resulting new front element is greater than l in the 3D case (this is necessary for the convergence of the method). In the 2D case, the new front element is created even if its level is less than or equal to l.

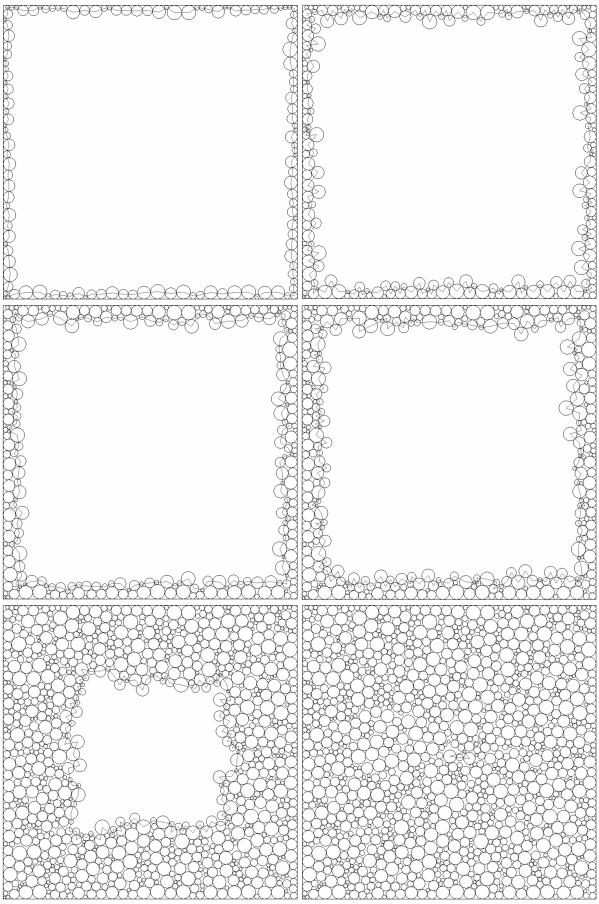

Fig. 1. A 2D illustration of a filling example using our algorithm

a new particle is a function of the levels of the particles of the front element used to insert this new particle. Indeed, the level l_{new} of the new particle P_{new} is calculated by:

$$l_{new} = \min_i(l_i) + max(1, \max_i(l_i) - \min_i(l_i)) \tag{1}$$

where l_i are the levels of the particles of the front element used to insert the new particle. For the convergence of the method, the following two conditions must be satisfied: (a) a new particle level is equal or greater than to the minimum of the levels of the particles of the front element used to insert this new particle, (b) the filling using the elements of a front with a level l is initiated only when the fronts with levels less than l are completely saturated. The detection and analysis of conflicts (intersections) between the particles constitutes a crucial task which generally takes the half of the computing time of the whole filling

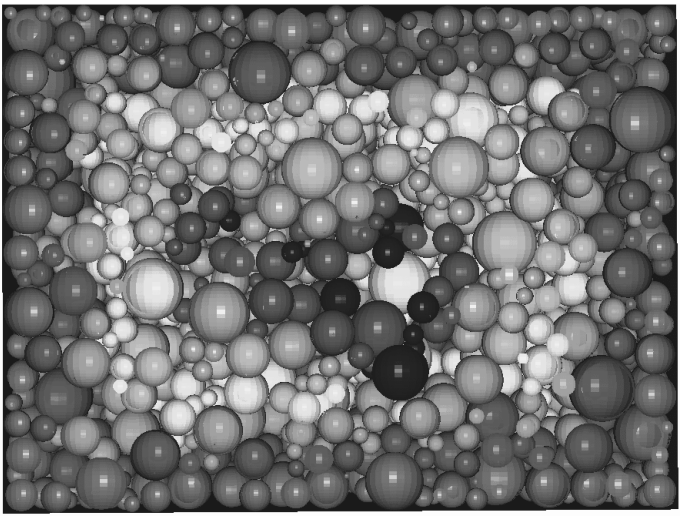

Fig. 2. Illustration of a filling example in 3D

process. To make this task faster, this detection is localized using a grid. In addition to the speed of the method, the use of this grid makes our algorithm linear. For more details of this algorithm see [1, 26, 27].

3.2 The Algorithm Efficiency

In this section, we give some criteria to characterize the sphere packing algorithm. In addition to the computing time and the density of the generated structure, another criterion is introduced. Indeed, this criterion characterizes the elements of the front by giving their quality. The quality q_e of a front element (P_1, P_2, P_3) (respectively (P_1, P_2) in the 2D case) is given by:

$$q_e = \frac{\delta_{12} + \delta_{23} + \delta_{31}}{3} \quad \text{(respectively, } q_e = \delta_{12}) \tag{2}$$

with $\delta_{ij} = d(P_i, P_j) - R_i - Rj$, where $d(P_i, P_j)$ denotes the Euclidean distance between the centers of the particles P_i and P_j with radii R_i and R_j respectively. The efficiency Q of the algorithm is presented by an histogram giving the frequency corresponding to each quality interval. Indeed, we define the minimum quality q_{min}, the maximum quality q_{max} and a quality step defined by $q_s = \frac{q_{max} - q_{min}}{10}$ for example. The frequency f_i corresponding to the ith quality interval $[q_{min} + (i-1)q_s, q_{min} + iq_s[$ is $\frac{n_i}{n_e}$, where n_i is the number of front elements whose quality is belonging to the quality interval i, and n_e is the total number of the front elements. It is said that a filling has a good quality if the histogram, representing Q, is a Gaussian centered in q_{min}.

3.3 Application

In this section, we consider a nanostructure of pure iron [25]. The domain to fill is a parallelepiped with dimensions $240 \times 300 \times 240$ nm (a square with dimensions 240×300 nm in the 2D case), the grain size distribution has radii ranging from 2 to 8 nm as shown in table 1, and the grain boundary thickness is a function of the grain radius given by $Gb(R) = 2/R$. In the filling process, the radius of each grain will be calculated as $R + Gb(R)$, where R is a radius randomly chosen in the grain size distribution (for more details see [1, 26, 27]).

Table 1. The grain size distribution of the pure iron nanostructured

radii (nm)	frequency (%)
[2 - 2.5[	15.8
[2.5 - 3[	21
[3 - 3.5[	20.2
[3.5 - 4[	16.7
[4.5 - 5[	6.8
[5 - 5.5[	4
[5.5 - 6[	3
[6 - 6.5[	0.8
[6.5 - 7[	0.7
[7 - 7.5[	0.6
[7.5 - 8[	0.4

In these conditions, using a machine with a 1.4 GHz processor, our algorithm generates a structure containing 33,738 spherical balls (respectively 1,330 disks in 2D) with a density of 0.494 (0.80) and a computing time less than 3 s (0.02 s). We can note that the filling is quite fast and that the density (in the 2D case) is quite satisfactory. The obtained density in the 3D case cannot be compared to a theoretical value because the distribution contains different grain sizes. To characterize our algorithm concerning the densities that can be obtained in the 3D case, we apply it in the particular case of a fixed radius (4 nm), keeping the same domain size and the same grain boundary expression. Indeed, the idea is to compare the density of the generated structure with the value 0.64 corresponding to the highest possible density in a irregular assembly of constant size particles. In this particular case, the generated structure contains 39,061 spherical balls with a density of about 0.606. This density is much better than the value 0.494 previously obtained with the specified grain size distribution.

The generated structures in 2D and 3D are illustrated by figure 3. We can observe that there are many empty areas in these structures (more visible in the 2D case). As mentioned above, this is due to the modeling of the grains by spherical balls (disks) and to the advancing-front method intrinsic characteristic of generating empty areas when two fronts are meeting. In addition to these factors, the constraints of generating random irregular structures (no periodicity) and the conformity with the grain size distribution, in our case, accentuate these

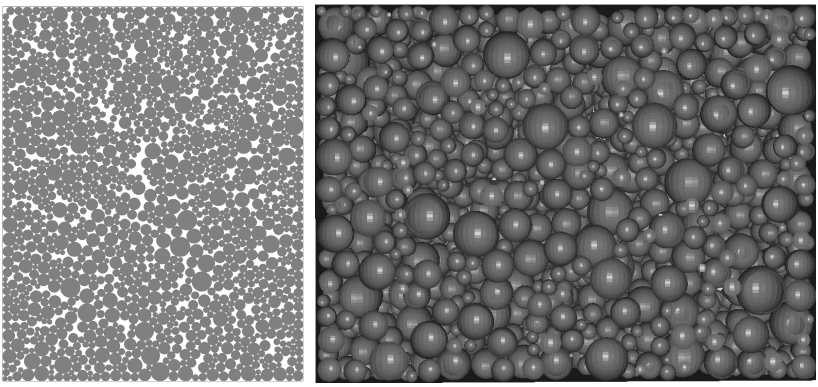

Fig. 3. Illustrations in 2D and 3D of the pure iron nanostructured model

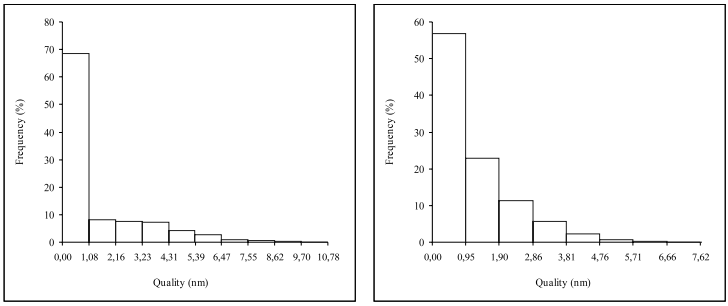

Fig. 4. The quality histogram for the cases 2D (left) and 3D (right)

empty areas. Since these empty areas, introduced by the filling algorithm, are residual, *i.e.* they do not physically exist in the real structure, then they must be equally distributed on the whole structure. This is the main purpose of the next section.

The filling quality histograms (of both 2D and 3D cases) for the pure iron example are shown in figure 4. We can observe that the filling quality in the 2D case is a Gaussian more pronounced (almost a Dirac) than in the 3D case. Indeed, the filling quality in the 3D case is calculated, on each triangle (a front element), by the average edges (of this triangle) quality, which does not show the gap between the qualities of these edges as in the 2D case.

4 Optimization of the Generated Structure and Cell Construction

The advancing-front method usually leads to an heterogeneity of the local density in the structure (some regions are denser than others, see figure 3). This feature

can have a direct impact on the results of the physical simulations, particulary in the nanomaterials case. Indeed, in a nanostructure, an empty area is considered as an additional grain boundary. However, the proportion and the repartition of grain boundaries in such a structure are the major features of its behavior (obstacles to dislocations propagation for example [3]). Hence, a point relocation algorithm is necessary to equally distribute these additional empty areas on the whole structure, and then to reduce as much as possible their impact on the physical simulations.

The main idea of the point relocation algorithm that we propose is the following principle: "*in order to equally distribute the empty areas on the whole structure, one just has to equally distribute them locally around each particle*". Thus, we need to know the local configuration around each point (particle center), *i.e.* the positions and radii of its nearest particles. For this purpose, we use the weighted Delaunay triangulation (dual of Laguerre diagram) built on the point cloud constituted by the particle centers (where the weights are the radii of these particles) and the points of the domain boundary discretization (considered as particles with radius zero). The domain boundary discretization must be as fine as possible so that all the Laguerre diagram cells (dual of this triangulation) are completely contained in the domain.

4.1 Domain Boundary Discretization

Several approaches can be used to obtain the discretization of the domain boundary. As explained before, this discretization must allow all the Laguerre diagram vertices to be completely contained in the domain, but it must also be as least costly as possible (minimum number of points). To obtain this discretization, we are proceeding in an iterative scheme. At a first time, the weighted Delaunay triangulation, using the set of points constituted by the particle centers and the 8 points of the parallelepiped (4 points of the rectangle) representing the domain, is built. Then, for each tetrahedron (triangle) whose vertices are including at least one of the particle centers, we check if the center of the weighted "circumscribed" sphere (disk) is outside (or on) the domain boundary. If it is the case (outside or on the domain boundary), the centers of these particles are orthogonally projected onto the domain boundary side where the center of this "circumscribed" sphere (disk) is outside (or on it). These projected points are then added to the point cloud and a new weighted Delaunay triangulation is built. These second and third steps iterate until all the Laguerre diagram vertices (the tetrahedra (triangles) "circumscribed" spheres (disks) centers) are completely contained in the domain (including if they are on the domain boundary). In the 3D case, from two points associated with the same particle center (projection on two perpendicular planes), a third point (corresponding to the orthogonal projection of these two points on the segment, intersection of these two planes) is added to the point cloud. In figure 5, an example of the weighted Delaunay triangulation and its dual (Laguerre diagram) of a point cloud in 2D is given.

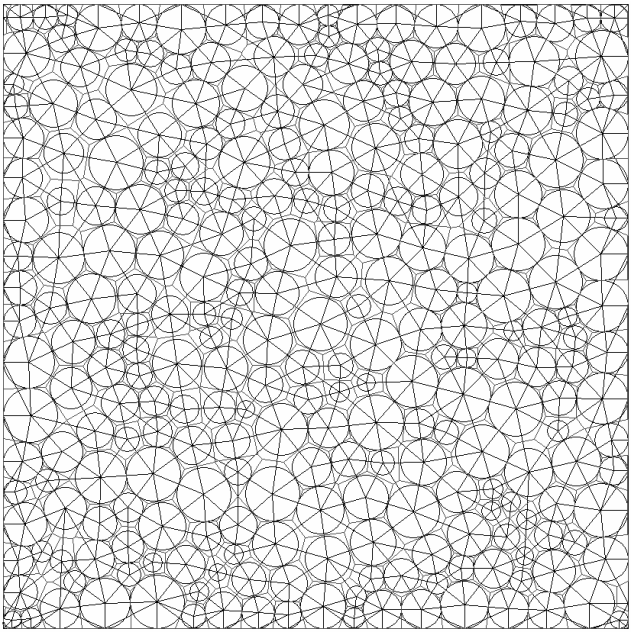

Fig. 5. The weighted Delaunay triangulation and its dual Laguerre diagram (red color) of a point cloud in 2D

4.2 The Point Relocation Algorithm

The algorithm we propose here to balance the density on the whole structure is an algorithm of point relocation under constraints. Indeed, the optimization produced by this algorithm must not lead to intersections between the particles, and thereby preserves the validity of the generated structure. The idea of this algorithm is to use the weighted Delaunay triangulation, as described above, to move locally each particle by considering its nearest neighbors (its topological ball). Indeed, each particle center is repositioned using the front defined by all the sides of its ball boundary. Let F_i (respectively S_i) be the ith face or triangle (segment) of the front defined by the sides of the point P ball boundary, we can assume that there is a unique position P_i of P calculated using F_i (S_i). This point relocation method is to move, step by step, the point P towards the barycenter P^* of the points P_i. At each step, the validity of the point P (no intersection with the other particles, and the particle centered on P is completely contained in the domain) is checked. The general scheme of our point relocation algorithm can be described by the few following lines:

- Do while there are relocated points
  - For each point P (particle center)
    - Define the ball of P using the weighted Delaunay triangulation
    - Calculate the positions P_i associated with the faces F_i (segments S_i) of the front defined by the sides of P ball boundary

- Calculate the barycenter P^* of the points P_i
- Move the point P, step by step, towards P^* (do $\alpha = 0.3$ to 0.05 by -0.05)
  - $P_\alpha = (1-\alpha)P + \alpha P^*$
  - Control the validity of P_α
  - If the point P_α is valid then the position of the point P is replaced by P_α, exit from the loop on α
- End of the loop on α

– End of the loop on the points P
– Reconstruct the weighted Delaunay triangulation

• End while

One may speculate about the use of this new point relocation algorithm, whereas other algorithms exist, especially the smoothing algorithm, usually used to smooth the edge lengthes in a mesh. Indeed, we have tested two variants of the smoothing algorithm in our context. The first variant is the classical smoothing scheme, which leads to intersections between the particles (and then do not preserve the validity of the generated structure). In the second variant, where the constraint of no overlapping between the particles is introduced, the optimization is very weak and the generated structure keeps a big heterogeneity in the local density.

4.3 Characterization of the Point Relocation Algorithm

In this section, we introduce a criterion to characterize the effectiveness of the point relocation algorithm. This criterion supposes that the weighted Delaunay triangulation WDT, based on the point cloud constituted by the generated particle centers and the points of the domain boundary discretization, is given. Indeed, with each point P of the WDT (except the centers of the border particles and the points of the domain boundary discretization) is associated a function $F(P)$ defined by:

$$F(P) = \max_j(\delta_j) - \min_j(\delta_j) \tag{3}$$

with $\delta_j = d(P, P_j) - w - w_j$, where $d(P, P_j)$ is the Euclidean distance between point P and point P_j (one of the points of the topological ball of point P given by the WDT), w if the weight of point P defined by $w = R + gb(R)$, where R is the radius of the particle centered on P and $gb(R)$ is the grain boundary around this particle, and w_j is the weight (in the same manner as w) of point P_j. Note that in our case, as shown in figure 6, we have necessarily $\delta_j \geq 0$.

By minimizing the function $F(P)$, the maximum and the minimum of δ_j tend to an average value δ_{moy} and then the optimization is realized. To present the results of the application of this criterion, we draw the graph of the function F between F_{min} (the minimum of $F(P)$) and F_{max} (the maximum of $F(P)$) by a frequency histogram in 10 intervals:

$$[F_{min} + i\frac{F_{max} - F_{min}}{10}, F_{min} + (i+1)\frac{F_{max} - F_{min}}{10}[, \quad \text{for i=0,...,9}$$

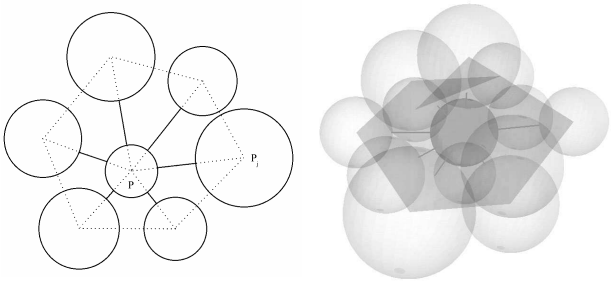

Fig. 6. The quantities δ_j are presented by solid lines for the 2D case (left) and by green lines for the case 3D (right)

4.4 Construction of Cells

The generated particles with spherical (circular) shape can be transformed, using the dual of the weighted Delaunay triangulation obtained after the point relocation procedure, into cells more similar to the grain shape observed in reality. This transformation procedure is optional, depending on the grain shape required by the user. To construct the cells representing the grains, we consider the dual of the weighted Delaunay triangulation (Laguerre diagram) obtained after the structure have been optimized (concerning the repartition of empty areas). As shown in figure 5, the cells entirely contain the particles, which is a principal feature of the duality between the weighted Delaunay triangulation and the Laguerre diagram. In our case, the weight of each particle, in the weighted Delaunay triangulation, is $w = R + gb(R)$, meaning that the grain boundaries are included in the weights of the particles. To separate the grain boundaries, these cells must be isotropically shrunk. Thus, from the vertices o_i of a cell and point P center of its correspondent particle, we define the vertices $\tilde{o}_i$ of the shrunk cell by:

$$\tilde{o}_i = o_i + (1 - \frac{R}{w})\overrightarrow{o_i P} \tag{4}$$

5 Numerical Applications

In this section, we consider two numerical examples in both 2D and 3D cases. The first example is a model where particles have the same radius, whereas in the second one, the particles have different radii. The machine used in the two examples has a Pentium M processor with 1.4 GHz.

5.1 Example 1

The 2D model consists in a domain with dimensions 100×100 nm and particles of radius 0.9 nm with a constant grain boundary of 0.1 nm. In the 3D model, the particles radius and the grain boundary are the same as in the 2D model and

Table 2. Modeling results in 2D and 3D cases

Results	2D model	3D model
Nb particles	2,789	17,787
Density	0.88	0.60
Time(s)	0.03	1

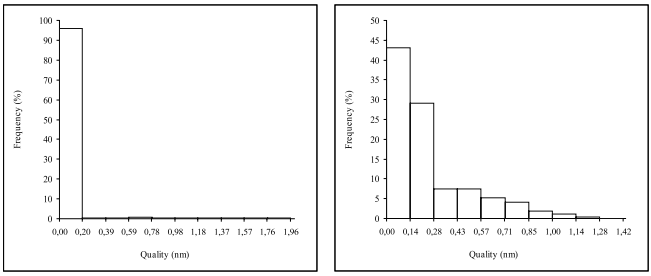

Fig. 7. The quality histograms for the cases 2D (left) and 3D (right)

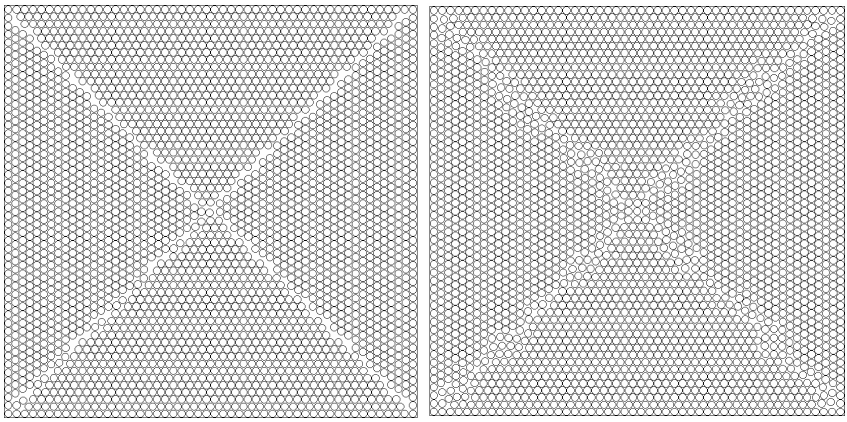

Fig. 8. The generated structure in the 2D model, before (left) and after (right) the application of the optimization algorithm

the domain is with dimensions $50 \times 50 \times 50$ nm. The results of the geometrical modeling are given in table 2.

We can observe that the filling is quite fast and gives satisfactory densities. The filling qualities, in both 2D and 3D are illustrated in figure 7. In figure 8 we illustrate the generated structure in the 2D model before and after the application of the point relocation algorithm. The histogram of function F, giving the repartition of the empty areas on the structure, before and after the application of the optimization algorithm is given in figure 9. We can observe that the two

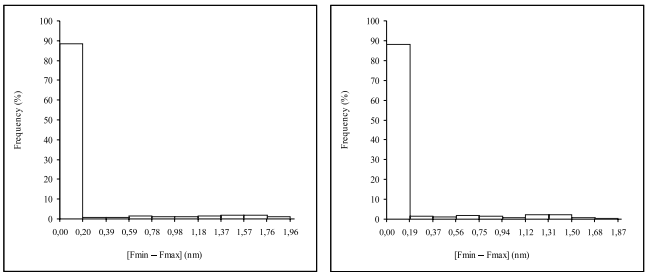

Fig. 9. The function F, in the 2D model, before (left) and after (right) the application of the optimization algorithm

histograms are quite similar: this is because, before the application of this algorithm, the function $F(P)$ was optimal (near 0) for most of points P (as we can see it in figure 8), and because for some points, the constraint of no intersection between the particles leads to a locked state.

5.2 Example 2

The 2D model consists in a domain with dimensions 200×200 nm and a uniform grain size distribution with radii in the interval $[0.8\,\text{nm}, 2.8\,\text{nm}]$ and a constant grain boundary of 0.2 nm. In the 3D model, the grain size distribution and the grain boundary are the same as in the 2D model and the domain has dimensions $50 \times 50 \times 50$ nm. The results of the modeling are given in table 3.

Table 3. Modeling results in 2D and 3D cases

Results	2D model	3D model
Nb particles	2324	1482
Density	0.81	0.51
Time(s)	0.02	0.05

In Figure 10, the structures with cells and particles in both 2D and 3D are presented. Note that each cell entirely contains a particle, and thus the volume (area) of the cell is always greater than that of its correspondent particle. Indeed, in the filling process, the thickness of grain boundaries is included in the radii of the particles, and despite of this, empty areas (intrinsic to the advancing-front method) still exist in the structure. These empty areas, which have no physical sense as nonexistent in the real structure (model obtained experimentally), are equally distributed on the whole structure leading to two kinds of models: either the model with spherical (circular) grains where the grain sizes are maintained and the thickness of grain boundaries is slightly increased, or the model with cells where the grain sizes are slightly increased and the thickness of grain boundaries is kept.

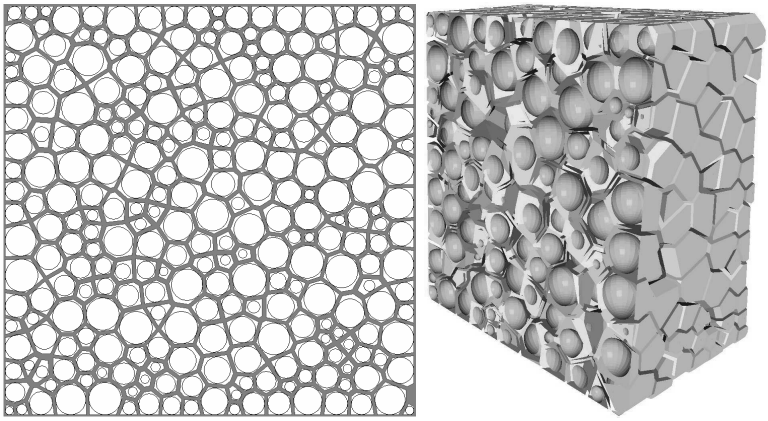

Fig. 10. Illustrations in 2D and 3D of the structures with cells and particles

6 Conclusions and Outlook

In this paper we have considered the problem of granular structure geometrical modeling under constraints in both two and three dimensions. We have considered this modeling as the classic sphere (circle) packing problem, and introduced a new method based on an advancing-front approach generally used in the meshing context.

Applying this new method in many cases, especially for nanostructures, we have shown that it can generate structures with a large number of grains (over 30,000) in a short computing time (about 2 s) with a satisfactory density and an ensured convergence.

The advancing-front approach used often generates heterogeneities in the repartition of empty areas in the structure. Thus, we have also proposed an optimization algorithm for equally distributing these areas on the whole structure. The proposed method is based on techniques of point relocation under constraints.

Finally, we have introduced a simple method to transform the structure with spherical (circular) grains into an assembly of cells closer to the grain shape observed in reality. This method is based on the weighted Delaunay triangulation, built on the point cloud constituted by particle centers and the points of the domain boundary discretization, by considering its dual Laguerre diagram.

To ensure the convergence in the 3D case, some faces (front elements) have been neglected in the filling process. An improvement in the algorithm on this point can be investigated, which may increase the density. Also, this improvement can be achieved by considering a densification procedure by a compromise between the density and the conformity with the grain size distribution.

In the point relocation algorithm, the optimization is based on the minimization of a functional calculated, for each particle, using the length of the empty

areas (introduced by the advancing-front method) along some directions. Another approach is to consider this empty space in all directions by calculating the functional using the volume (the area) of this space around each particle. This method is more difficult to implement and more costly in terms of computing time, but it may give better results in the repartition of the empty areas on the structure.

This geometrical modeling of granular structures (nanostructures in particular) is a pre-step of their simulation by the finite element method. Another key task, which constitutes one of our principal activities, is the generation of quality meshes of these structures (in 2D and 3D) required by this simulation.

References

1. Benabbou, A., Borouchaki, H., Laug, P., Lu, J.: International Journal for Numerical Methods in Engineering (to appear, 2008)
2. Lu, K., Lu, J.: Materials Science and Engineering A 375, 38–45 (2004)
3. Kumar, K.S., Van Swygenhoven, H., Suresh, S.: Acta Materialia 51(19), 5743–5774 (2003)
4. Jodrey, W.S., Tory, E.M.: Phys Rev A 32, 4 (1985)
5. Lubachevsky, B.D., Stillinger, F.H.: Journal of Statistical Physics 60(5-6), 561–583 (1990)
6. Lin, X., Ng, T.: Geotechnique 47(2), 319–329 (1997)
7. Jiang, M.J., Konrad, J.M., Leroueil, S.: Computers and Geotechnics 30(7), 579–597 (2003)
8. Stillinger, F.H., Dimarzio, E.A., Kornegay, R.L.: J. Chem. Phys 40, 1564 (1964)
9. Visscher, W.M., Bolsterli, M.: Nature 239 (October 1972)
10. Bagi, K.: Mechanics of Materials 16(1-2), 101–110 (1993)
11. Han, K., Feng, Y.T., Owen, D.R.J.: Powder Technology 155, 33–41 (2005)
12. Bennett, C.H.: J Appl Phys 43, 6 (1972)
13. Feng, Y.T., Han, K., Owen, D.R.J.: International Journal for Numerical Methods in Engineering 56(5), 699–713 (2002)
14. Löhner, R., Onate, E.: International Journal for Numerical Methods in Engineering 61, 1977–1991 (2004)
15. Bagi, K.: Granular Matter 7, 31–43 (2005)
16. Stoyan, D.: Models of random systems of non-intersecting spheres. In: Prague Stochastics, JCMF 1998, pp. 543–547 (1998)
17. Häggström, O., Meester, R.: Random Struct. Algor. 9, 295–315 (1996)
18. Evans, J.W.: Rev Mod Phys 65, 1281–1304 (1993)
19. Cui, L., O'Sullivan, C.: Granular Matter 5(3), 135–145 (2003)
20. George, J.A.: Computer implementation of the finite element method, PhD thesis, Stanford university (1971)
21. Lo, S.H.: International Journal for Numerical Methods in Engineering 21, 1403–1426 (1985)
22. Delaunay, B.: Otdelenie Matematicheskikh i Estestvennykh Nauk 7, 793–800 (1934)
23. George, P.L., Borouchaki, H.: Delaunay Triangulation and Meshing, Hermes, Paris (1998)
24. Yamakawa, S., Shimada, K.: International Journal for Numerical Methods in Engineering 57(13), 1923–1942 (2003)

25. Tao, N.R., Wang, Z.B., Tong, W.P., Sui, M.L., Lu, J., Lu, K.: Acta Materialia 50(18), 4603–4616 (2002)
26. Benabbou, A., Borouchaki, H., Laug, P., Lu, J.: Rapport de recherche Inria (6414) (2008), http://hal.inria.fr/inria-00203375/fr/
27. Laug, P., Borouchaki, H., Benabbou, A., Lu, J.: C. R. Mecanique 336, 506–511 (2008)