Powder Technology 253 (2014) 311–324

Contents lists available at ScienceDirect

Powder Technology

journal homepage: www.elsevier.com/locate/powtec

Modified collective rearrangement sphere-assembly algorithm for random packings of nonspherical particles: Towards engineering applications

A. Bertei [a,*], C.-C. Chueh [b], J.G. Pharoah [b,c], C. Nicolella [a]

[a] Department of Civil and Industrial Engineering, University of Pisa, Pisa, Largo Lucio Lazzarino 2 56126, Italy
[b] Queen's-RMC Fuel Cell Research Centre, 945 Princess St., 2nd Floor, Kingston, ON K7L 5L9, Canada
[c] Mechanical and Materials Engineering, Queen's University, Kingston, ON K7L 3N6, Canada

ARTICLE INFO

Article history:
Received 3 May 2013
Received in revised form 15 November 2013
Accepted 24 November 2013
Available online 8 December 2013

Keywords:
Packing algorithm
Random packing
Simulation
Nonspherical particles
Fuel cells
Agglomerates

ABSTRACT

A modified collective rearrangement algorithm, based on the sphere-assembly representation of the particle shape, is presented for simulating random packings of nonspherical particles and arbitrary shapes. Compared to other collective rearrangement methods, the modification introduced by this algorithm consists in i) avoiding the particle-to-container shrinking procedure by allowing particles to sequentially settle down and ii) in a stability constraint applied to all the particles in the packing. The coupling of these criteria ensures that all the particles are stable and contacting each other, allowing for an unambiguous detection of contacts, which is important in the evaluation of the effective properties desired in many engineering applications, such as percolation thresholds and effective conductivity. The effect of the internal parameters of the algorithm is investigated, showing that random close packings can be obtained. The algorithm is applied to simulate packings of rigid ellipsoids and cylinders with different aspect ratios, which are compared with simulation results provided by other packing algorithms, showing the consistency of our method. Simulations of inter-penetrating particles, mixtures of particles with different shapes and packings of agglomerates are shown, which confirm the applicability of the method to a broad range of packing problems of practical interest and, in particular, for fuel cell applications.

1. Introduction

Random packings of particles are widely considered in science and engineering applications: they have been suggested as models for liquid and glass structure [1,2] and they are used to represent granular materials [3], packed beds, and cermets [4] as well as in many other applications. In the last decades, several packing algorithms have been developed to represent the packing microstructure and for property evaluation. Many algorithms have been developed in particular for spherical particles: Monte-Carlo [5–7], drop-and-roll and sequential deposition algorithms [8–13], collective rearrangement [14–16], discrete element methods [17–19] and molecular dynamics [20,21] to cite the most common ones.

Recently, in order to have a better representation of particulate systems, and mainly due to the availability of increased computational resources, attention has shifted to simulating random packings of nonspherical particles. The first problem arising with nonspherical particles is the more complex shape than the spherical form. Different methods have been proposed to account for nonspherical shapes. In several methods the analytical equation of the particles is considered in the algorithm: packings of ellipsoids [22–25], spherocylinders [26], superballs [27,28], superellipsoids [29] and general convex particles [30] have been investigated. The use of the analytical equation to represent the particle has the clear advantage that the exact shape of the particles is considered. Although such a method is elegant and rigorous, the detection of overlaps is non-trivial and may lead to detection errors in some critical situations [31]. In addition, the algorithm is tailored for the specific shape under consideration and, clearly, particles with a shape that cannot be represented analytically cannot be simulated.

Another approach consists of tessellating the container and the particle shape with a grid, digitizing both the domain and the particles [32–35]. In this way, any particle shape can be approximated with a coherent collection of pixels (2D) or voxels (3D), and the collision and overlap detection are simply noting whether two objects occupy the same site in the grid. On the other hand, quantitative predictions of packing characteristics, such as packing density, are sensitive to the resolution used and increasing the resolution through a finer grid leads to too much higher memory requirements than other methods [32]. Furthermore, the movement of the particles is discretized, such that particle trajectories are affected by the resolution of the grid.

The third method is the so-called multi-sphere (or sphere-assembly) approach [36–40], in which particles are represented by an assembly of component spheres reproducing their shape. As the

* Corresponding author. Tel.: +39 050 2217865; fax: +39 050 2217866.
E-mail addresses: antonio.bertei@for.unipi.it (A. Bertei), chih-che.chueh@queensu.ca (C.-C. Chueh), pharoah@me.queensu.ca (J.G. Pharoah), c.nicolella@diccism.unipi.it (C. Nicolella).

0032-5910/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.powtec.2013.11.034

digitizing method, general particle shapes, analytical or otherwise, can be reproduced by varying the position and the size of the component spheres within the particle. The detection of particle overlaps is carried out by checking if two component spheres, belonging to different particles, overlap, which is much easier if compared with the first approach where the analytical particle equations are used. On the other hand, higher resolutions, obtained with a larger number of component spheres, slow down the algorithm and require more computer memory, though usually less than in the case of the digitizing method.

The packing procedure is the second important feature to consider and it can affect the resulting packing properties. Existing physical simulation models include the discrete element method (DEM) [29,40,39] and the molecular dynamics method (MD) [23,24,28]. In these algorithms the real interaction forces are taken into account, rigorously simulating the dynamics of the packing generation in time domain. While even jammed configurations can be obtained [22,41], these methods are usually very complex and, although specific technical solutions can be used to speed up the simulations, they are less computationally efficient than many purpose designed packing algorithms, at least for spheres [42]. Furthermore, for some engineering applications such a highly detailed physical representation is not necessary.

In Monte-Carlo methods [31], each particle is added to the domain one by one by selecting a random position and orientation and checking the overlaps with previously placed particles: if there are no overlaps, the current particle is accepted, otherwise a new position is tried and, after a predefined number of trials, the particle is rejected if an acceptable placement has not been found. Though this algorithm is straightforward, it is very time consuming and no rearrangement of particles is permitted (i.e., the orientation is completely random). This usually results in loose packings, and when rigid particles are simulated the majority if not all the particles are not in contact with one another.

Another packing algorithm, which has been widely adapted for non-spherical random packings [36,38], is the collective rearrangement method (CR). In this algorithm all the particles are randomly distributed and oriented in a domain which is smaller than the volume that all the particles may occupy. At the beginning particles experience large overlaps, which are individually removed by iteratively moving and rotating each particle under the action of a restoring force and a restoring moment generated in consideration of the overlaps. The rearrangement of particles is usually coupled with a process of particle to domain reduction, consisting of either reducing (i.e., scaling down) the particle size or increasing (i.e., scaling up) the domain volume. When simulating hard particles, the algorithm stops when all the overlaps have been removed. This technique is expected to be faster than DEM and MD, since the physics is only approximated in order to save computational time. While this method usually provides close packings, they are generally not strictly jammed. Moreover, there is no check about the contact information of particles during the rearrangement process: typically, in the final configuration particles are arranged in unstable positions or are isolated, feature which is pronounced by the particle-to-container shrinking procedure.

Finally, in principle each packing algorithm could be coupled with one of the three approaches described above to represent the particle shape, although some concerns have been recently arisen when the sphere-assembly approach is coupled with the discrete element method [39,40].

In this study, a collective rearrangement method coupled with the sphere-assembly approach is used, sharing some features in common with the Nolan and Kavanagh work [36]. However, in the present algorithm both the particles and the container maintain their dimensions, avoiding the particle shrinking procedure. A constraint has been introduced and applied to each particle in order to provide packings in which all the particles are stable. The coupling of these two characteristics ensures that all the particles contact each other in the final configuration, characteristics that cannot be generally guaranteed by conventional CR algorithms [21]. The unambiguous detection of contacts is a desired feature in many engineering applications because percolation thresholds and conduction properties of the packing are strictly related to the number of contacts [43–45,7]. In addition, the algorithm has been generalized to allow for multiple polydisperse phases and to a controlled degree of particle overlap in order to simulate deformable particles and sintered multi-phase packings.

The study focuses on the application to rigid and non-rigid ellipsoids, cylinders and agglomerates, though the algorithm is sufficiently general that any particle shape can in principle be used. The chosen sub-set of shapes is common in several engineering applications, such as in polymer electrolyte [46] and solid oxide [47–49] fuel cells which, in part, motivated the developments described herein. In these applications it is important that in the reconstructed microstructure the packing is representative of a stable configuration and the particles experience a desired degree of overlap, in order to ensure that charges can be transported and converted throughout the packing [48,50,47,51–53].

The paper is organized as follows: in Section 2 the algorithm is presented in detail; in Section 3, the algorithm is first explored by assessing the effects of its internal parameters, then simulation results for rigid spheres, ellipsoids and cylinders are compared to those obtained by other algorithms, and finally some results regarding the broader possibilities of the algorithm are shown.

2. Algorithm

2.1. General aspects

The algorithm was written in C++ programming language with the use of some functions provided in an open source finite element deal.II library [54]. The code was generalized to account for both 2D and 3D packings with one or more polydisperse or monodisperse phases of rigid or deformable particles: in this study only the three dimensional problem is described.

The algorithm begins with the definition of the domain size and the number, types and sizes of particles. The domain consists of a box of specified dimensions having a rigid floor and periodic boundary conditions in the horizontal directions. Based on the desired volume composition, the initial porosity and the particle size distribution of each phase, the number of particles for each phase is calculated. A very small and unrealistic initial porosity is used in order to fill the domain with more particles than there will be in the final configuration: in this way, packings that do not completely fill the chosen domain are excluded.

All the particles are labeled with a progressive index i, from 1 to the total number of particles N_{part}. Apart from its type and size, each particle i is characterized by its center coordinates $\underline{c}_i$ (i.e., the coordinates of its center of mass) and two axial directions, $\hat{\underline{x}}_i$ and $\hat{\underline{z}}_i$, which identify the orientation of the particle with respect to the global system of reference (see Fig. 1). Note that the third axis $\hat{\underline{y}}_i$ is not independent and can be easily calculated as $\hat{\underline{y}}_i = \hat{\underline{z}}_i \times \hat{\underline{x}}_i$.

Each particle is represented by an assembly of $n_c^{(i)}$ component spheres, which means that the radius ($r_k^{(i)}$) and relative position of each $k^{(i)}$ component sphere with respect to the local particle coordinate system ($\tilde{\underline{c}}_k^{(i)}$) are stored. Note that $\tilde{\underline{c}}_k^{(i)}$ represents the center of the $k^{(i)}$ component sphere in the local frame, which is centered on the particle center of mass and oriented along the particle axes. The center coordinate of the component sphere in the global frame is indicated with $\underline{c}_k^{(i)}$. There is a one-to-one correspondance between the center of the component sphere in the global (i.e., $\underline{c}_k^{(i)}$) and in the local (i.e., $\tilde{\underline{c}}_k^{(i)}$) coordinate systems as reported in Eq. (1).

$$\underline{c}_k^{(i)} = \underline{c}_i + \begin{vmatrix} \hat{\underline{x}}_i^T \\ \hat{\underline{y}}_i^T \\ \hat{\underline{z}}_i^T \end{vmatrix}^{-1} \tilde{\underline{c}}_k^{(i)} \quad \text{and} \quad \tilde{\underline{c}}_k^{(i)} = \begin{vmatrix} \left(\underline{c}_k^{(i)} - \underline{c}_i\right) \cdot \hat{\underline{x}}_i \\ \left(\underline{c}_k^{(i)} - \underline{c}_i\right) \cdot \hat{\underline{y}}_i \\ \left(\underline{c}_k^{(i)} - \underline{c}_i\right) \cdot \hat{\underline{z}}_i \end{vmatrix} \tag{1}$$

Eq. (1) shows that the component sphere center in the global frame $\underline{c}_k^{(i)}$ can be calculated from the local one $\tilde{\underline{c}}_k^{(i)}$ by applying a rotation matrix,