Powder Technology 120 (2001) 175–186

POWDER TECHNOLOGY

www.elsevier.com/locate/powtec

A packing algorithm for particles of arbitrary shapes

X. Jia, R.A. Williams [*]

Centre for Particle and Colloid Engineering, School of Process, Environmental and Materials Engineering, University of Leeds, Leeds LS2 9JT, UK

Received 1 June 2000; received in revised form 1 October 2000; accepted 16 January 2001

Abstract

Particle packing is a subject of both academic and industrial importance, and a number of packing algorithms have been proposed and widely used. However, most of these packing algorithms can only deal with spheres and a few regular shapes. If applied to arbitrarily shaped particles, they would have difficulties in at least one of three aspects. First, arbitrary shapes are notoriously difficult to model mathematically. Secondly, efficient algorithms for collision and overlap detection of arbitrary shapes are difficult to derive and more difficult to implement. Thirdly, the packing program would be very computationally expensive for routine and practical use. This paper describes a new, digital approach to particle packing, which can avoid many of the difficulties suffered by conventional methods. The key innovation is digitisation of both particle shapes and the packing space. Thus, a particle is now just a coherent collection of pixels or voxels, regardless of its shape, moving on a square lattice, onto which the packing space is mapped. Using the digital approach, it is easy to pack particles of any shapes and sizes into a container of any geometry, generally requiring no more than an ordinary PC. Although, as yet, the packing algorithm does not involve physical forces explicitly, it can simulate some physical phenomena such as size segregation. The ability to pack particles in their real shapes, rather than approximated as spheres, opens up many new industrial and academic opportunities, some of which are discussed. Examples of packing density predictions of particles subject to various effects, including vibration and rotation, are given. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Particle packing; Digitisation; Arbitrary shape

1. Introduction

Efficient packing of particles is a centuries-old problem. It has intrigued and attracted both theorists and experimentalists for as long, due to the prevalence of packing phenomena in nature, everyday life and many industrial processes [1,2]. By particle ‘packing’, we mean ‘putting together’ or ‘arranging’ particles in a confined space. By this definition, the means or process by which packing is actually achieved remains elusive. In fact, packing may be achieved by any means. Stacking oranges by hand at a grocery, filling up silos in bulk handling and storage plants and tableting in pharmaceutical and other industries are just a few examples of the means by which packing is achieved. Whatever the means, geometry constraints or steric hindrance usually determine the compactness of a packing structure, regardless of the means or packing process. A class of computer methods exists to generate packing structures by exploring the geometrical constraints. These methods are commonly referred to as ‘packing algorithms’. They are designed specifically to generate the structure, rather than simulate the process, of particle packing. There are some physical simulation models, such as distinct element method (DEM) or molecular dynamics (MD) methods, which can take the real interaction forces into account and simulate the dynamic process of particle packing and generate, as a result, the packing structure. These physical models, however, are generally not referred to as packing algorithms, since they are designed and used mainly for some other purposes. For packing of equal spheres at least, they are also less computationally efficient than many of the specially designed packing algorithms, although they may produce denser structure than some packing algorithms. The above definition of ‘packing algorithm’ is retained, for the sake of convenience, in the following discussion.

Most packing algorithms published to date are for spheres [2–8] or sphere composites [9–13]. Only a few are for non-spherical particles, and most of which are limited to analytical shapes such as ellipses, ellipsoids, spheroids

[*] Corresponding author. Tel.: +44-113-233-2789; fax: +44-113-2781. *E-mail address:* r.a.williams@leeds.ac.uk (R.A. Williams).

0032-5910/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0032-5910(01)00268-6

and cylinders [14–18]. Traditional packing algorithms may be classified into in three basic types.

- Ballistic algorithms—Particles follow well-defined or definable trajectories to find a resting place in the packing. A typical example was given by Vold [3] for random packing of spheres. Compared with other types, it is computationally efficient but difficult to implement, especially for containers of complex geometry. It is flexible enough to generate structures ranging from dense packing to fractal growth, simply by varying the so-called sticking probability.
- Random placement algorithms—Particles from a given size distribution is placed one by one in randomly selected positions in the packing space. If a newly introduced particle does not overlap with others, it is allowed to stay; otherwise, another position is tried. If, after a predefined number of trials, the particle still has not found a place to stay, it is discarded and another is selected and tried in the same manner. Implementation of the algorithm, even for containers of complex geometry, is easy. However, the method is very time consuming and, for non-spherical particles, rearrangement by way of rotation is often not possible. Moreover, since some particles may have to be discarded, the size distribution of the packed spheres will be somewhat different from the intended, original distribution. The packing density depends on the sequence of particle addition. For example, if larger spheres are added first, the final packing density tends to be higher than if the smaller spheres are put in first. However, because it is easy to implement, the method is widely in use.
- Growth algorithms—A particle packing can also be generated by ‘growing’ points, placed randomly or with some constraints with regard to inter-point distances, into the required shapes. One example is to let the points grow into spheres until they touch each other. Another example is to first tessellate the packing space using the points as seeds, then place the particles one in each tessellation cell. Without further rearrangements, the algorithm does not usually result in a dense packing. With rearrangements, for example, by allowing the particles to move around the their original positions and/or by allowing them to shrink or expand, denser packing can be generated, but there is no precise control over the size distribution in the final packing either. With rearrangements, the method can also be very time consuming.

The difficulty with these existing packing algorithms, when attempting to apply them to an arbitrary shape, can occur at three levels, namely mathematical, implementation and computational levels.

- Mathematical—Apart from a small number of regular shapes and their composites, the shapes of most particles encountered in nature and industry are difficult, if at all possible, to describe precisely in mathematical (or geometrical) terms. Nor can these descriptions be readily represented and manipulated precisely in the computer memory. Thus, if the traditional analytical approach (of first having a precise mathematical model of the objects) to particle simulation is followed, the currently available mathematical tools place a fundamental limitation to the kind of shapes that can be dealt with by computer simulations.
- Implementation—In a typical packing process, the most computationally costly part is usually collision and overlap detection. The speed of this depends critically on how the objects are represented and manipulated. Thus, how shapes are represented and manipulated is not simply a geometrical issue but also a computational one. Circles or spheres are the simplest to handle. Any other shapes, even the seemingly simple, regular shapes like lines, cylinders, polygons and polyhedrons, pose a considerable challenge to programmers implementing the algorithms. Moreover, a good algorithm for one particular shape may be bad for another. Round-off errors can also be a problem in particle simulations in which real numbers are used for dimensions and positions of the particles, as is usually the case. The way such errors are handled not only affect the computed packing properties such as the co-ordination number but also is known to make or break a packing process based on the ballistic algorithms. Thus, except for spheres (3D) or circles (2D), geometrical shapes are notoriously difficult to deal with, and software implementation is itself a major undertaking.
- Computational—In practice, most complex shapes are approximated as composite of simple shapes such as points and line segments (in 2D), and polygons and spheres (in 3D). A large number of these basic shapes may be required to represent one particle, and all the shapes are stored and computed in floating points. The computational cost can be enormous. Take an extreme example, 806,006 polygons were reportedly used to represent 16 chess pieces and packing these 16 pieces took about 4 days [19]. In another example, 500 CPU hours on an IBM RS600-560 workstation were reportedly used to generate a small number of sample structures containing up to 600 non-spherical particles [17]. Thus, even if the above two difficulties could be overcome, the resulting computer code would be too demanding on computer memory and CPU time to be widely and routinely used.

Dynamic simulation models, such as DEM and MD, are likely to face the same mathematical and implementation difficulties as the traditional packing algorithms, since they both use the same or similar approaches to shape representation and collision/overlap detection. They tend to be more computationally demanding than the dedicated packing algorithms, because they must also simulate correctly the dynamics, not just the end result, of the packing process. To be numerically stable, the time step used by these dynamic simulations is dictated by the smallest of the particles, and they frequently need millions of time steps to simulate just a few seconds of real time process.

This paper describes a new, digital approach to particle packing. The key is digitisation of both particle shapes and the packing space. Using the digital approach, many of the above difficulties can be avoided, making it much easier to pack particles of any shapes and sizes into a container of any geometry, requiring no more than an ordinary PC. The philosophy and advantages of the digital approach are described in Section 2. The approach is exemplified by a 2D implementation in Section 3. The packing algorithm should apply equally to 3D, but mapping arbitrary 3D shapes into computer's memory is not as straightforward as in 2D. 3D implementation will be dealt with in a future paper. The ability to pack particles in their real shapes, rather than approximated as spheres, generates many new industrial and academic applications, some of which are discussed briefly in Section 4.

2. Model description

In TV or computer images, everything is pixel based. Each single object in a scene or image, no matter how complex it is in shape, is represented by discrete picture elements or pixels. This 'pixelation' (2D) or 'voxelation' (3D) of real objects and the space is the basis of the new packing algorithm. Thus, as shown in Fig. 1, an arbitrary shape is now simply a coherent collection of pixels (in 2D) or voxels (in 3D). The resolution depends on how accurate the shape needs to be represented which, in turn, is often dictated by the requirements of a particular application. The shapes are allowed to move randomly, one lattice grid at time. At each time step, there is a number of possible directions for a particle to trial move: 8 in the case of 2D and 26 in the case of 3D, each direction has an equal probability to be selected. It is convenient to treat diagonal moves as composed of two orthogonal moves. For example, a move in the upper-left direction can be thought of as an upward move followed by a left move. In order to encourage particles to move downwards, the upward component of a move is usually only accepted with a so-called *rebounding probability*. In other words, at each time step, a random direction is selected for a trial move, but if a trial move has an upward component, the upward movement is only accepted with the rebounding probability. The result is a directional and diffusive motion for the particles, rather like a random walks based cluster–cluster aggregation and sedimentation model described elsewhere [20]. This diffusive movement allows the particles to effectively explore every available space. The movement on a square lattice simplifies the collision and overlap detection. The tendency to move downward simulates the effect of gravity. By allowing the shapes to move upward from time to time, the effect of a fast, small amplitude vibration can also be simulated.

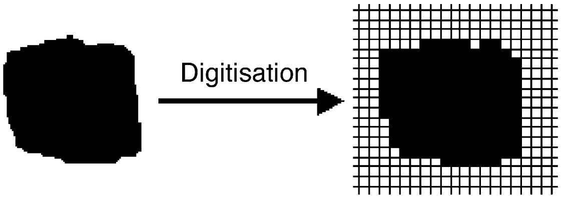


Fig. 1. Digitisation of a real object. The grid or resolution is usually the compromise between the accuracy required and computational cost involved.

For packing of arbitrary shapes, the digital approach has several advantages over the traditional methods.

- It is not limited to mathematically or computationally manageable shapes. An arbitrary shape can simply be scanned in and used. Both 2D and 3D scanners are commercially available today and can be used for this purpose.
- Since the objects reside and move in a grid, collision and overlap detection is now a simple matter of detecting whether two objects occupy the same site(s) at a given time, rather than having to compute and test intersections between objects, which is usually the most expensive part of such particle simulations.
- Given the resolution, the number of pixels used to represent a shape, hence the computational cost, depends on the area (in 2D) or volume (in 3D) of the shape, and does not necessarily increase with the complexity of the shape. In contrast, a conventional (e.g. polygonal) approach generally needs more elements to represent more complex shapes.
- The use of pixels or voxels means that the computations involved are mostly integer operations. For single processor computers, today's PCs have a very similar integer performance to the much more expensive RISC workstations. For example, the integer performance of a 450-MHz Pentium III processor is rated at SPECint95 18.7, and that of a 450-MHz UltraSPARC-II used in Sun's high-end workstations is 19.7, although the floating point performance of Intel based PCs still lag far behind (SPECfp95 13.7 for Pentium vs. 27.0 for SPARC). Therefore, the algorithm can be implemented and run on ordinary PCs.

As with any other algorithm, the digital approach has its own drawbacks. For example, it is just as, if not more, memory intensive as some traditional shape representations. Although it is possible to significantly reduce memory requirement, this will be at the cost of speed (see below for details). For simple shapes such as spheres, the digital algorithm may be slower than well-implemented traditional (ballistic) algorithms. Some techniques, for instance feature reduction [19] used by some conventional approaches to speed up program execution, cannot be used for the digital approach with the same success, because the number of pixels/voxels required depends more on the area/volume than the complexity of a shape. Quantitative prediction of packing characteristics such as packing density is sensitive to the resolution used, and the sensitivity is usually non-linear and shape-dependent. Despite these, the simplicity, ease of implementation and the speed of the digital algorithm make it an attractive alternative to the

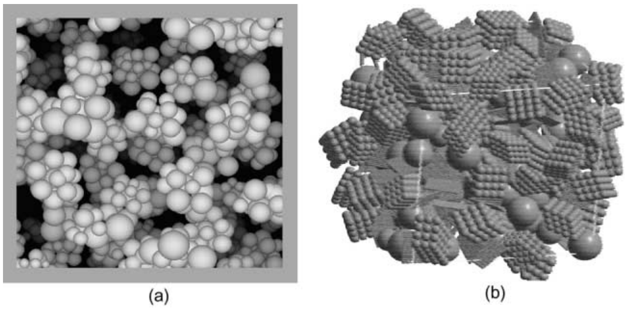


Fig. 2. Examples of complex structures approximated by packing of sphere assemblies. (a) Model structure of a sintered carbon membrane filter generated by modifying a loose packing of spheres: the original spheres are replaced by assemblies of smaller spheres to represent the blobs observable in the real membrane under SEM, the spheres are then allowed to expand slightly to create overlaps to simulate the effect of fusion caused by sintering. (b) Structure generated using a random placement algorithm, assemblies of small spheres are used here to represent plates and trapezoids. (b: Courtesy of Oxford Materials. The packing was created using its commercial software MacroPac.)

traditional packing algorithms when it comes to packing particles of arbitrary shapes.

At first sight, the digital approach is not too different from the sphere composite approach that has been in use for years: both approximate the real shapes using some basic building blocks. Fig. 2 shows two examples of the sphere composite approach. However, the sphere composite approach is still an analytical method and hence suffers from the same computational difficulties mentioned earlier. Besides, with the sphere composite approach, a shape is typically *constructed* by hand with the help of some shape building software. This is a tedious exercise and requires some model of the real shape to exist in advance, either explicitly or implicitly, much like building a house needs an architectural plan. With the digital approach, however, the real shape is typically *decomposed* by scanning into a collection of pixels/voxels. Thus, a mathematical model is not a prerequisite and the procedure is less tedious as much of it can be automated.

3. Examples of implementation of the simulation

The digital algorithm is exemplified here by a 2D implementation of the algorithm in the form of a PC program for Microsoft Windows. We refer to the program as DigiPac 2D. Details and some of the implementation issues are also described below. Sample results are presented and discussed to demonstrate the principles of the approach.

3.1. Digitisation

In 2D implementation, digitisation is achieved by a ‘pixel counting’ of drawn or scanned images.

The methodology involves constructing a program with a library of commonly encountered shapes whose dimensions (in pixels) are user-definable. These include arcs and chords, ellipses and circles, line segments, sine waves, polygons, polylines and curves, and text with user-definable fonts (see Fig. 3 for examples). These basic shapes are well supported, through provision of library functions, by C/C++ compilers such as Microsoft Visual C++ (VC) and Borland C++ Builder (CB). Thus, in 2D, generating any of these shapes is a simple matter of drawing the shape in the computer’s memory and collecting the pixels that belong to the drawn shape. Particles of the same shape and size are generated by copying, which is faster than the above drawing and collecting procedure. Particles of the same shape but different sizes can be generated by duplication and then scaling. Any size distribution can be used, although only the frequently used ones, such as linear, uniform, bimodal, normal and log normal distributions, are built-in. An arbitrary distribution can be read in as a histogram and used.

If the entire packing structure must be displayed ‘as is’ on a computer screen, then screen resolution imposes a practical limit on how large and hence how accurately the particles (shapes) can be represented. Otherwise, there is no limitation on size and accuracy—particles can be as large as required and the shapes can be represented to any desired accuracy. Most of the VC or CB graphics library functions used to draw the built-in shapes work on a canvas of $2^{15} \times 2^{15}$—more than enough for most purposes. Higher resolution could be obtained by self-implementing the drawing algorithms, many of which are well documented [21].

More complex shapes are imported as bitmaps. This allows engineering drawings and scanned images of real shapes to be used as templates to reproduce particles of the same shape. It is possible to retain the original colours or shades of the imported images when packing. This may be useful in artistic pattern design but less important for particle engineering applications. Therefore, all imported images, whether colour or mono, are converted to a binary representation—all coloured (i.e. not white) pixels are deemed to be part of the shape—before use. Different sizes of the same shape can be created by scaling, for which there are VC or CB library functions to use. In Fig. 4, the nail-shaped particles are generated from an imported bitmap.

Collecting 3D shapes using a 3D scanner is also a feasible prospect for the future. An automated procedure is currently being developed.

3.2. Particle motion

At each time step, a particle moves in a randomly selected direction, one grid at a time, on a square lattice. There are eight possible directions—four orthogonal and four diagonal—to choose from, all with equal probability.

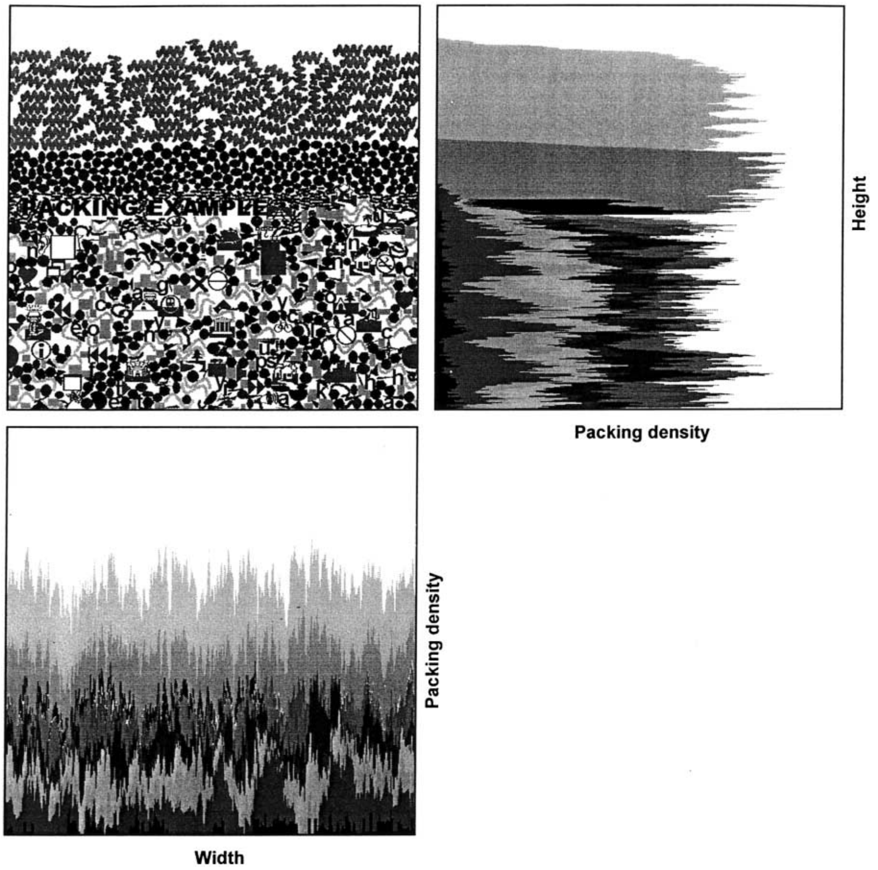


Fig. 3. Sample output of 2D packing of various built-in shapes including thick line segments, sine waves, rectangles, pentagons, triangles, ellipses, symbols and text. There are 1104 objects, packed in a 600 × 600 box with periodical boundary conditions on the left and right ‘walls’. The overall packing density profiles along the height and width of the simulation box, and contributions from the components (or shapes), are also shown.

A diagonal move is actually achieved by two orthogonal moves. For example, a move to the northwest (upper-left) is achieved by a move to the west (left) and from there to the north (up). To simulate gravity induced packing process, if the selected direction is ‘up’ or contains an upward component (for diagonal directions), the move or the upward component is only performed with a probability, referred to as rebounding probability. The overall effect created by this procedure is a diffusive and mostly downward particle motion. The diffusive part helps the particles to effectively explore every available space, while the downward part forces them to settle and form the packing.

3.3. Boundary conditions

Vertically, along the direction of gravity, there are two ways by which particles are introduced. These are referred

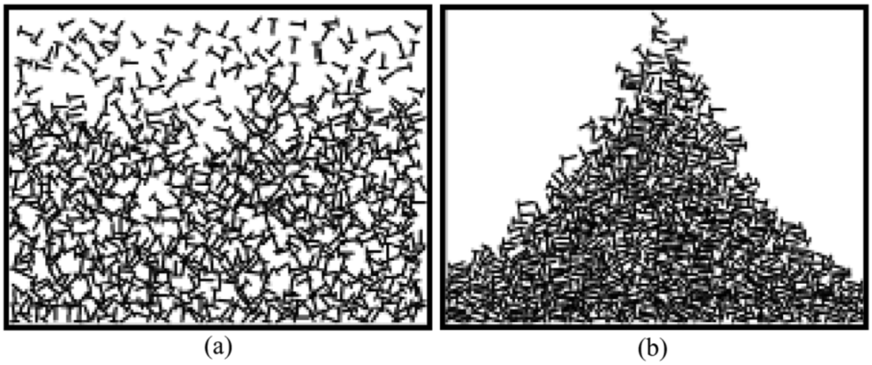


Fig. 4. Packing of randomly orientated, nail-shape objects, each containing 215 pixels. The shape is imported from a bitmap. In both cases, the simulation box is 800 × 600, bounded by solid walls. (a) Packing structure being generated by particles ‘pouring’ down at a fast rate of one particle per five time steps, no rebounding and no rotation. Typical packing density is 0.39. (b) Packing structure being generated by allowing particles to be added at a much slower rate, one particle per 50 time steps, and to rotate randomly and rebound with a probability of 0.5. Typical packing density is 0.60.

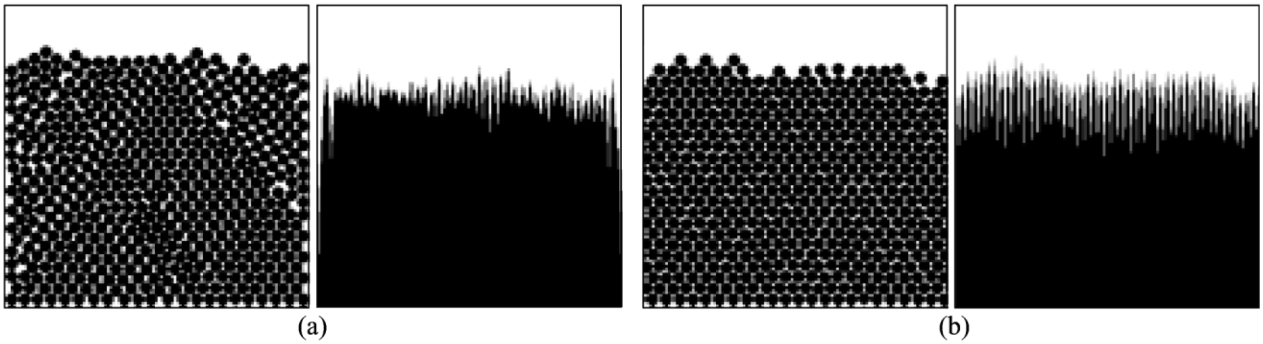


Fig. 5. The sphere packing structures and density profiles under different boundary conditions (a) solid walls and (b) periodical boundary conditions.

to in DigiPac 2D as the ‘lateral spread’ and ‘point source’. The former is also known as the rain model. In DigiPac 2D, if the ‘lateral spread’ option is selected, up to 10 particles are introduced simultaneous from random positions uniformly distributed across the width of the packing space (Fig. 4a). There may be times when the 10 particles are too big to be introduced at once. In such cases, as many particles as the width can ‘comfortably’ accommodate are added. If the ‘point source’ option is selected instead, particles are introduced one by one from a narrow orifice at the centre above the packing (Fig. 4b).

Laterally, the boundary conditions can be either periodical or solid walls. With periodical boundary conditions, if a particle moves out of one side and it comes in from the opposite side, at the same time. This avoids the wall effect and allows bulk properties to be predicted by a small sample. The width of the simulation box should be much larger than that of the largest particle (e.g. by a factor of 10). With solid walls, the packing is usually less dense near the wall than in the central part. This is the so-called wall effect, and is most evident for packing of spheres. As can be seen in Fig. 5, the wall effect can still be ‘felt’ several particle diameters into the packing.

The container (for solid walls) or the packing space (for periodical boundary conditions) dose not have to be rectangular in shape. Since particle movement and particle-wall collision or overlap detection are both pixel based, handling a container or packing space of a more complex geometry presents no additional complications. Fig. 6 shows an example of packing of circles and line segments into a container with two obstacles. A non-zero rebounding probability was used to simulate vertical vibration that helps the particles to fill the otherwise difficult to reach regions.

3.4. Control parameters and their effects

A range of parameters and options can be used to control the packing process. These include the rate of particle addition, rebounding probability, boundary conditions, whether to allow rotation, immobilisation and removal of certain particles. The way particles are introduced and whether to allow particles to rotate and/or rebound have a profound influence on the packing structure. For example, in Fig. 4a, the nails rain down the container and rotation is not allowed. The resulting structure is much porous than that in Fig. 4b, where the nails trickle down a narrow orifice one by one and both rotation and rebounding are allowed. The rate of particle addition also affects the packing structure, since it determines how long a particle has to explore the packing before being locked-in by new additions. Slow rate means particles have more time to explore the packing and hence more likely to find a tight fit in the packing. In Fig. 7, three packing structures, all with periodical boundary conditions, are compared. In Fig. 7a, particles are introduced from a point source at the rate of one particle per 50 time steps. In Fig. 7b and c, particles rain down across the width of the packing space at different rates: one particle per five time steps for Fig. 7b and one particle per 50 time steps for Fig. 7c. At a faster rate (Fig. 7b), a more random packing structure with many holes is generated; whereas at a slower rate (Fig. 7c), an ordered, much compact structured is generated. In Fig. 7a and b, the rate of addition is the same but particles are introduced differently. Both structures have holes, but one (Fig. 7a) is more ordered than the other (Fig. 7c).

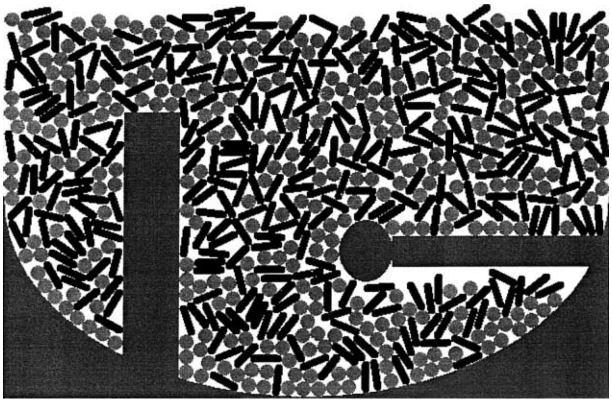


Fig. 6. An example of packing into a complex geometry.

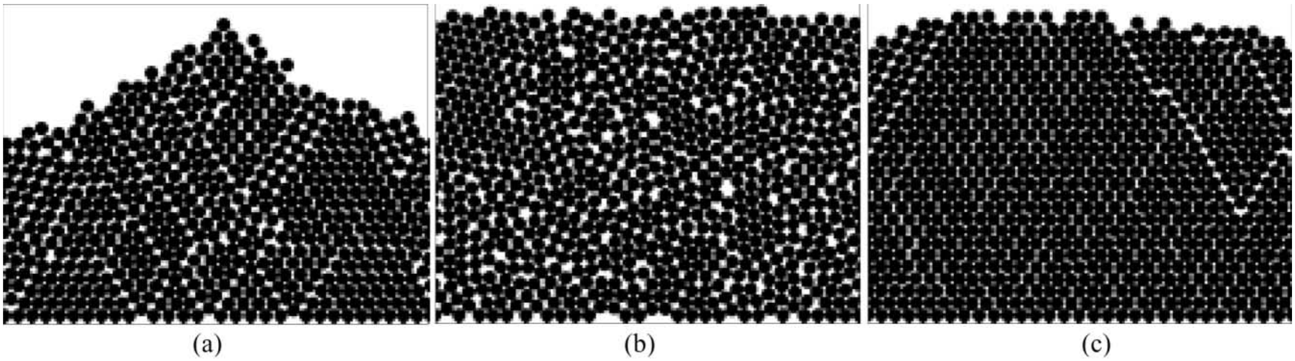


Fig. 7. Effects of particle addition rate and method. (a) Particles are introduced from a point source, at the rate of one particle per 50 time steps. Average packing density is 0.840. The structure is partially ordered, with some holes. (b) Rain model is used and particles are added at a fast rate of one particle per 5 time steps. Average packing density is 0.816. The structure is random, with many holes. (c) Rain model is used and particles are added at a slow rate of one particle per 50 time steps. Average packing density is 0.882. The structure is ordered, with no holes (although some cleavages are visible).

In practice, packing is often achieved in a controlled manner. The resulting structure is not truly and completely random, and it is not ordered either. Although the program essentially relies on biased random walks and generates pseudo random packing structures, certain measures can be used to help control the packing process. For example, rotation of each component can be turned ‘on’ or ‘off’ at any time. Any particle can be immobilised, or mobilised (after immobilisation), or permanently removed from the packing, at any time. For example, in Fig. 3, a few large particles were removed from below the text, which behaves like a rigid, non-rotational beam, to allow more contacts between the beam of text and other particles below it. Particles of different shapes can be mixed together before packing (e.g. lower part of Fig. 3), or they can be packed in layers in order of creation (e.g. upper part of Fig. 3).

Although not currently implemented, it is possible to generate ordered packing structures by, for example, only allowing particles to follow certain ballistic trajectories (which are of course also digitised). In between ordered and random packing, controlled or guided packing can also be realised by, for example, directing particles to fill certain holes, removing particles which are causing deadlocks, introducing particles in a particular order designed to maximise the packing density, etc. In 3D implementation, Virtual Reality techniques can be utilised to aid the control process.

3.5. Vibration and size segregation

Although the packing algorithm does not explicitly involve physical forces, some effects of physical interactions can still be simulated. Since particles are allowed to move side ways as well as up–down all the time—even after they form part of the packing—the effects of high frequency, small amplitude vibrations in both horizontal and vertical directions can be simulated. The results are a denser packing structure and, for particles of different sizes trickling down a point source, size segregation (Fig. 8). The vertical vibration is controllable, by the rebounding probability. A value of 0 means no vertical vibration, and a value of 1 means equal opportunity for the particles to move up and down, thus no chance to settle down and form a packing. Typically, a value between 0.2 and 0.5 is used. Using a rebounding probability of 0.5 and dropping particles from a point source at the rate of one particle per 50 time steps, size segregation of bimodal mixtures of circles can be observed for a size ratio as small as 1.1, and the critical size ratio does not appear to be affected by the proportions of the different sized circles in the mixtures.

3.6. Optimisations

We have considered several ways to speed up the packing simulation. For example, immobilisation of certain particles is not only a measure to achieved controlled packing but also helps to speed up the simulation, especially when applied in large numbers. The simulation can allow particles that have not been successfully moved over the last definable number of time steps to be fixed in place (i.e. taken out of consideration for subsequent trial moves). Another option allows particles whose index is below a definable value to be fixed. Since particle index corresponds to the order of packing, this option can effectively take a large part of the packing out of the subsequent computation.

Other measures of optimisation include using bit, instead of integer, representation of the pixels, and using outlines for solid shapes. Use of bit representation immediately gives a saving in computer memory by a factor of 32 (an integer is four-byte long and a byte contains eight bits). Although bit operations are slower than integer operations, for large-scale packing the saving in memory may still outweigh the comparatively slight increase in CPU time. Bit representation will not be suitable if the other informa-

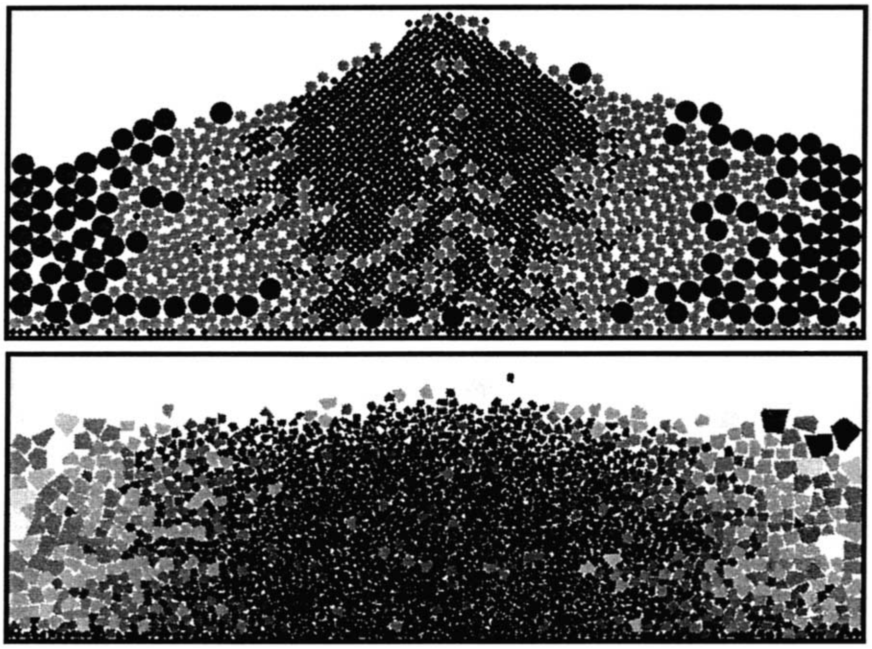


Fig. 8. Examples of size segregation due to simulated rapid, small amplitude vibration. Rotation is allowed for the polygonal shapes.

tion needs to be stored in-situ to simulate, for example, selective adhesion in packing.

In DigiPac 2D, all solid shapes are represented and stored in whole, as in Fig. 9a. In fact, non-boundary pixels are redundant as long as the shape is considered rigid during packing. The outline representation, as shown in Fig. 9b, is computationally much more efficient. An optimum 'wall' thickness of the object needs to be determined. If it is too thin, holes and gaps are likely to appear resulting in broken boundary after rotation transformation —if rotation is allowed. If it is too thick, the maximum possible efficiency cannot be achieved. This short-cut may not be appropriate in situations when frequent estimation of packing properties (e.g. density and permeability) is required, or when the shape is allowed to deform, during the packing simulation.

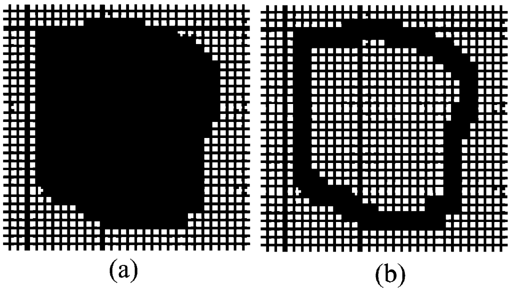


Fig. 9. Only the outline of a solid object is actually needed when moving it around and detecting overlaps it may have with other. This dramatically increases the speed of the packing algorithm. However, the boundary or hull needs to be re-filled when estimating properties such as packing density and mechanical strength.

3.7. Accuracy

Digitisation introduces initial errors, but such errors can generally be reduced by using finer resolutions. For example, in Fig. 10, a circle is represented by 76 and 316 pixels on a 10×10 and 20×20 grid, respectively. The area ratio between the circle and the square for the lower resolution case (Fig. 10a) is 0.76 ($= 76/100$), which represents a relative error of 3.2% from the true value of 0.7854 ($= \pi/4$). The ratio for the higher resolution case (Fig. 10b) is 0.79 ($= 316/400$), and error is now reduced to 0.6%.

During the packing process, translations do not incur round-off errors, and special measures are taken to minimise errors arising from rotations. For example, instead of using direct transformations, in DigiPac 2D rotation is by default accomplished by two successive shears along the X and Y axes [21]. The direct method is more straightforward

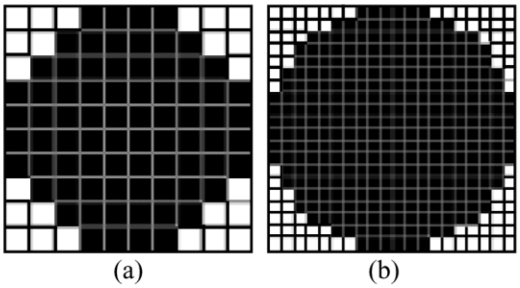


Fig. 10. Digital representations of a circle on a 10×10 and 20×20 grid, respectively. The higher the resolution, the lower the digitisation error.

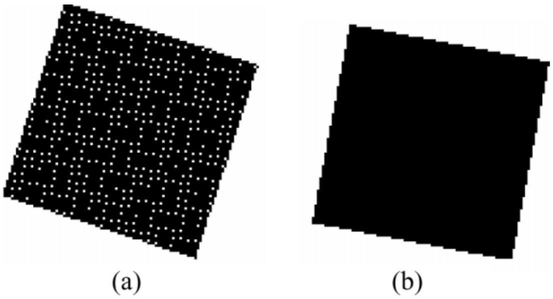


Fig. 11. A square block after being rotated using (a) direct transformation method and (b) shear method.

to implement and faster, but often results in patterned holes in the otherwise solid shapes, making them look like sieves (Fig. 11a). Therefore, unless specifically requested (e.g. for the sake of speed) by the user, this method is not used. The shear method is slightly more involved and noticeably slower, but it does not generate holes (Fig. 11b) and usually preserves the area (i.e. number of pixels). Even in the rare cases where the area is not strictly preserved, the errors are no more than a few pixels, which is small compared to the total number of pixels (usually in hundreds) used to represent a shape. Such errors do not accumulate because each rotation is from the original orientation, not from the previous one.

Table 1 compares the resampled packing density as a function of particle size for circles. The purpose of the resampling is to exclude contributions from, for example, particles still falling down and the unsettling top surface of the packing, in order to obtain a more representative mean packing density. By default, in taking the resampling from the density profile along H (cf. Fig. 3), bins with a value less than $2/3$ of the maximum density are neglected, i.e. only the most significant bins are used. All the simulations were performed in a simulation box of 800×600, with periodical boundary conditions; particles were added using the rain model at the rate of 10 particles every 500 time steps; the rebounding probability was 0.2. The 'relative error in area ratio' in Table 1 is against the true value of $\pi/4$, and is a measure of accuracy for single particle representation. The packing structure for particle size of 10 to 25 is dominated by large crystalline domains that run across almost the entire depth of the packing. The local packing density within these large crystals is between 0.90 and 0.92, very close to the theoretical value for hexagonal packing density 0.9069 ($= \pi/6\sqrt{3}$). For particle size of 5 and 30, much smaller crystalline domains exist.

4. Prediction of packing characteristics

Using this method, packing characteristics such as packing density and its distribution for various shapes can easily be obtained. Table 2 lists some examples. All simulations were performed on a 600×400 grid with periodical boundary conditions, 0.2 rebounding probability and the rate of particle addition of 10 particles per 500 time steps. The packing density given in Table 2 is calculated in the same manner as in Table 1, i.e. resampled using only the packing density values (bins) larger than $2/3$ of the maximum. The dimensions quoted in Table 2 are averages of the bounding boxes for the shapes. For multi-component mixtures, the ratios are between particle numbers (i.e. not volumes, areas, or weights). If rotation is allowed, the initial orientation of the particles no longer matters.

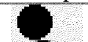
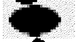
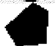
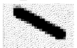
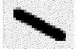
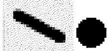
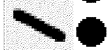
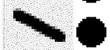
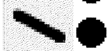
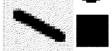
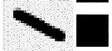
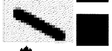
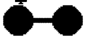
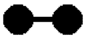
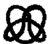
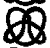
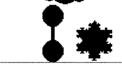
The predicted packing density for identical circles was 0.8746. This is between the values for random (0.82) and crystal (0.91) packing structures, since the generated structure contains crystal domains. Regarding Table 2, two further points should be noted. First, although rotation usually results in a more compact structure, this is not always the case. If rotation destroys an otherwise ordered structure, the packing may be less compact. For example, packing density in cases 15 and 17, both are random structures, is less than that in cases 14 and 16, both have large ordered domains. The second point to note is that in certain cases the resampling procedure can result in an overestimation of the true packing density. For example, in case 18 in Table 2, typical local density is 0.57, smaller than the resampled mean value of 0.61 reported in Table 2. The overestimation comes from the fact that the density profile, from which the resampling is performed, contains periodical fluctuations, due to the hollow structure of the knots, larger than the range over which the resampled average is taken, thus only peaks are included by the resampling.

Fig. 12 is a plot of the resampled packing density as a function of the aspect ratio for two types of 'fibres': one

Table 1
Packing densities as a function of particle size

	Particle diameter					
	5	10	15	20	25	30
Resampled packing density	0.8549	0.9282	0.8889	0.9082	0.8775	0.8506
Area ratio	0.8400	0.7600	0.7867	0.7900	0.7824	0.7622
Relative error in area ratio (%)	6.95	−3.23	0.16	0.59	−0.38	−2.95

Table 2
Typical packing densities of various shapes and compositions

Index	Shapes	Samples	Dimensions	Rotate	Orientation	Ratio	Density
1	Equal circles		32 × 32	No	-	-	0.8746
2	Equal ellipses		32 × 20	No	Parallel to X	-	0.8753
3	Irregular pentagons		32 × 32	No	Random	-	0.7500
4	Irregular pentagons		32 × 32	Yes	-	-	0.8078
5	Short fibres		32 × 8	No	Random	-	0.6035
6	Short fibres		32 × 8	Yes	-	-	0.7363
7	Short fibres + circles		32 × 8, 15 × 15	No	Random	1 : 1	0.7017
8	Short fibres + circles		32 × 8, 15 × 15	Fibres	-	1 : 1	0.7690
9	Short fibres + circles		32 × 8, 15 × 15	No	Random	1 : 2	0.7396
10	Short fibres + circles		32 × 8, 15 × 15	Fibres	-	1 : 2	0.7786
11	Short fibres + squares		32 × 8, 15 × 15	No	Random (fibres)	1 : 1	0.6984
12	Short fibres + squares		32 × 8, 15 × 15	Fibres	Parallel to X	1 : 1	0.7809
13	Short fibres + squares		32 × 8, 15 × 15	Both	-	1 : 1	0.7271
14	Stage 3 Koch’s snowflakes		30 × 35	No	-	-	0.7593
15	Stage 3 Koch’s snowflakes		30 × 35	Yes	-	-	0.7021
16	Dumbbells		64 × 32	No	Parallel to X	-	0.7391
17	Dumbbells		64 × 32	Yes	-	-	0.6847
18	Knots		60 × 54	No	-	-	0.6015
19	Knots		60 × 54	No	-	-	
20	Dumbbells + snowflakes		16 × 32, 15 × 18	Both	-	1 : 1	0.7233

with rounded ends and the other flat ends (i.e. more like rectangles). The packing conditions for all the simulations were kept the same, i.e. random orientation for all the fibres, no rotation, periodical boundary conditions in lateral directions, rebounding probability 0.2, rain model and the rate of particle addition 10 particles per 500 time steps.

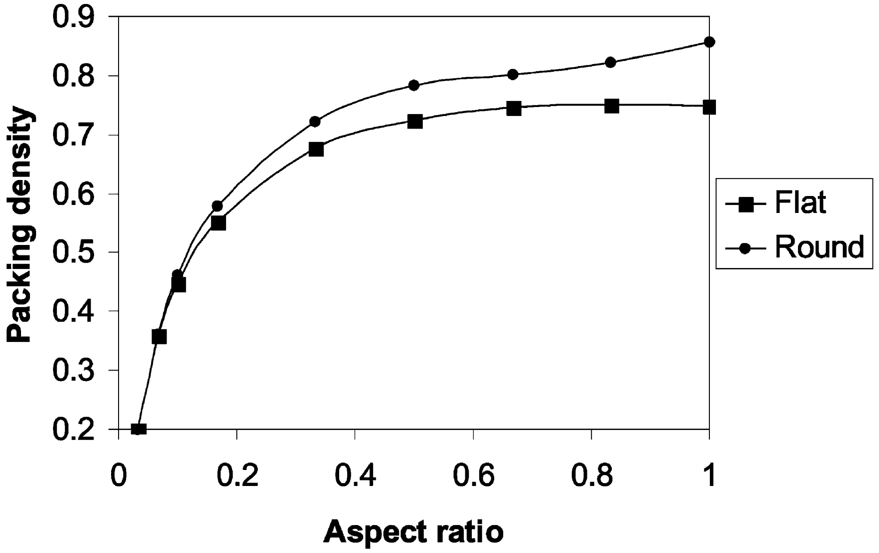


Fig. 12. Packing density as a function of aspect ratio for two types of fibrous particles—flat-ended and round-ended.

In all the simulations, the length of the fibres was fixed at 30, varying aspect ratios were achieved by varying the thickness of the fibres. The starting point for both types is the same, since for thickness of 1, it makes no difference whether the ends are round or flat. However, for thickness of 30, the round-ended fibres become circles, and flat-ended fibres become squares; and their packing densities differ the most—crystalline packing structure for the circles vs. the still random packing of squares.

5. Future work

We are now in a digital era, and it is befitting to have a digital algorithm for particle packing. The new algorithm not only provides a simple and effective solution to the age-old problem of generating particle packing but also opens up many new research opportunities. The approach described in this paper represents a significant development in the ability to predict packing properties of complex materials. Future work will consider further developments including the following.

- Extension to deformable shapes—This enables the packing program to simulate structural changes due to melting and sintering in, for example, powder metallurgy and rapid prototyping. It also provides a method for packing of soft materials, infiltration of one phase into another, etc.
- Implementation in 3D—The 2D implementation is good for demonstration purposes and useful in some but limited real situations. The full power of the digital approach will be unleashed through its 3D implementation. Extending the packing algorithm itself to 3D is straightforward. However, gathering 3D shapes and converting them to coherent collections of voxels rapidly and efficiently are far from trivial. In this respect, our effort will be focused on developing and implementing fast algorithms to automatically convert the output from a 3D scanner into a form directly useable by the packing algorithm. There are other, for example, stereographic [11] methods which may be used for the purpose.
- Validation—As with any simulation model, the digital packing algorithm needs validation. So far, the only quantitative validation is done for 2D random packing of equal spheres. DigiPac 2D can produce a packing density of 0.82—same as the generally accepted maximum value for this type of packing structures [4]. In addition, the packing contains the same crystalline domains typical of 2D ‘random’ packing of equal spheres. Other validations are being sought. For example, validations using tomographic measurements (e.g. data from X-ray microtomography and electrical capacitance tomography) will be carried out for 3D structures. The basic principles of tomographic techniques are given in [22] and recent example of using these techniques for structure characterisation can be found in Refs. [23,24].
- Applications—The usefulness of the packing algorithm is obvious in cases where the packing structure and packing density are the end goals. Use of the generated packing structure in other established numerical techniques, where a realistic model structure is required as input, will also be investigated. In particular, the feasibility of using the mesh grid of the void space in the packing for CFD style computation will be investigated. It is recognised that the uniform grid used in the packing algorithm may not be the most appropriate for CFD and is orders of magnitude smaller than the usual length scale used in CFD mesh, but it can be a starting point to form a more suitable mesh. It is possible to convert the digital shapes back to the traditional vector forms after the packing, so that some existing models based on the traditional shape description can make use of the packed shapes. Also to be investigated is the feasibility of using the digital version of the packing directly for estimating its mechanical strength—a discrete element approach.

Acknowledgements

The authors acknowledge financial support of the EPSRC under grant GR/N13357.

References

[1] W.A. Gray, The Packing of Solid Particles, Chapman & Hall, London, 1968.

[2] D.J. Cumberland, R.J. Crawford, The Packing of Particles, Elsevier, Amsterdam, 1987.

[3] M.J. Vold, The sediment volume in dilute suspension of spherical particles, J. Phys. Chem. 64 (1960) 1616–1671.

[4] W.M. Visscher, M. Bolsterli, Random packing of equal and unequal spheres in two and three dimensions, Nature 239 (1972) 504–507.

[5] M.J. Powell, Computer-simulated random packing of spheres, Powder Technol. 25 (1980) 45–52.

[6] W.S. Jodrey, E.M. Tory, Computer simulation of isotropic, homogeneous, dense random packing of equal spheres, Powder Technol. 30 (1981) 111–118.

[7] W. Scoppe, Computer simulation of random packings of hard spheres, Powder Technol. 62 (1990) 189–196.

[8] G.T. Nolan, P.E. Kavanagh, Computer simulation of random packing of hard spheres, Powder Technol. 72 (1992) 149–155.

[9] M.J. Vold, Sediment volume and structure in dispersions of anisotropic particles, J. Phys. Chem. 63 (1959) 1608–1612.

[10] K.E. Evans, M.D. Ferrar, The packing of thick fibres, J. Phys. D 22 (1989) 354–360.

[11] See http://www.oxmat.demon.co.uk/products/macropac.html for more information.

[12] M. Elimelech, J. Gregory, X. Jia, R.A. Williams, Particle Deposition and Aggregation: Measurement, Modelling and Simulation, Butterworth-Heinemann, Oxford, 1995, Chap. 14.

[13] G.T. Nolan, P.E. Kavanagh, Random packing of nonspherical particles, Powder Technol. 84 (1995) 199–205.

[14] J.M. Ting, M. Khwaja, L.R. Meachum, J.D. Rowell, An ellipse-based discrete element model for granular materials, Int. J. Numer. Anal. Methods Geomech. 17 (1993) 603–623.

[15] X.S. Lin, T.T. Ng, Contact detection algorithms for 3-dimensional

ellipsoids in discrete element modelling, Int. J. Numer. Anal. Methods Geomech. 19 (1995) 653–659.

[16] J.D. Sherwood, Packing of spheroids in three-dimensional space by random sequential addition, J. Phys. A 30 (1997) L839–L843.

[17] D. Coelho, J.F. Thovert, P.M. Adler, Geometrical and transport properties of random packings of spheres and aspherical particles, Phys. Rev. E 55 (1997) 1959–1978.

[18] R. Kohlus, M. Bottlinger, Computer simulation and characterisation of bulk solids built by irregularly shaped particles, in Proceedings of the 3rd World Congress on Particle Technology, Article No. 42.

[19] J.K. Dickinson, G.K. Knopf, Generating 3D packing arrangements for layered manufacturing, Rensselaer's International Conference on Agile, Intelligent, and Computer Intgrated Manufacturing, Troy, New York, Oct., 1998.

[20] X. Jia, D.J. Wedlock, R.A. Williams, Stochastic simulation of simultaneous colloidal aggregation and sedimentation, Miner. Eng. 13 (2000) 1349–1360.

[21] For example, the on-going Graphics Gems Series, published by Academic Press, from 1994 onwards.

[22] R.A. Williams, M.S. Beck, Process Tomography—Principles, Techniques and Applications, Butterworth-Heinemann, Oxford, 1995.

[23] D.M. Scott, O.W. Gutsche, ECT studies of bead fluidization in vertical mills, Proceedings of the 1st World Congress on Industrial Process Tomography, Buxton, Derbyshire, UK, 14–17 April, 1999, pp. 90–95.

[24] C.L. Lin, J.D. Miller, Network analysis of filter cake pore structure by high resolution X-ray microtomography, Chem. Eng. J. 77 (2000) 79–86.