# Using Surface Asperities for Efficient Random Particle Overlap Detection in the Generation of Randomly Oriented and Located Particle Arrangements

Christa E. Torrence[a*], Zachary Grasley[a], William B. Lawrimore[b], Edward J. Garboczi[c]

[a] *Texas A&M University, College Station, TX 77843 USA*

[b] *U.S. Army Engineering Research and Development Center, Vicksburg, MS 39180 USA*

[c] *National Institute of Standards and Technology, Boulder, CO 80305 USA*

**Abstract**

A novel method of detecting overlap between two irregularly shaped virtual particles whose surfaces are described by analytical functions is presented in this study. Utilizing the analytical equation that describes the surface of the particle, the first and second spatial derivatives are used to locate the major surface asperities via the surface curvature. During the particle placement process, detecting overlap between two particles is a crucial, but slow, step. In this work, particle asperities are checked for intersection with nearby particles first, leading to significant improvements in efficiency, compared to previous methods. While the asperity check does not detect all cases of overlap between two particles, it leverages the probability that the overlap is most likely to occur at a surface asperity. Microstructures are generated using four particle size gradations and four volume fractions to demonstrate the decrease in computational time associated with using the asperity overlap detection versus using a surface point cloud. This overlap detection method provides significant increases in efficiency, averaging a speedup of 10 when generating a 40 % volume fraction microstructure.

*Keywords:* Overlap detection, Irregular particle placement, Spherical harmonics, Particle asperities

## 1. Introduction

The precursor to many numerical simulations is a virtual recreation or development of a statistically representative surrogate of the material or structure to be studied. Particles and particulate composite materials present challenges when depicted virtually due to the complexity of accurately representing the morphology of the particulate components. Particle shape, size gradation, and spatial distribution have been shown to influence rheology [1–6], fracture behavior [7, 8], diffusivity [9–11], rate of reactions [12–14], and mechanical properties of composites [7, 15–17]. Thus an accurate representation of shape and distribution is of great interest.

Particle packing and spatial arrangement play important roles in many fields of research and industrial applications. Codes generating virtual representations of particles and composite materials have been increasingly used in simulating mechanical behavior [12, 13, 15], fluid dynamics [5, 18], and thermal conductivity [12, 14, 19–21]. The random packing and placement of regularly shaped particles such as spheres [20–25], cubes [26], cylinders [5, 27, 28], and ellipsoids [29–32] have been widely studied. For most practical virtual representations, portions of two particles cannot exist in the same space at the same time, so prevention of virtual particle-particle overlap is an important aspect of these codes. Particle-particle contact and collision has been extensively studied for the application of discrete element method (DEM) analysis, as DEM simulates the effects of contact-driven interparticle forces and external forces on the motion of collections of particles[33]. DEM random-shape particle representations typically utilize sphere clusters [34, 35] or superquadric functions to describe regular shapes such as ellipsoids, cubes, and cylinders [36, 37].

*Corresponding author
*Email address:* ctorrence@tamu.edu (Christa E. Torrence[a])

Detection of overlap between spheres is the most straightforward, warranting only a comparison of the distance between the centers to the sum of the radii. Furthermore, overlap between cubes can be determined by evaluating if any of the vertexes or line segments lie within another cube. In the case of cylinder-cylinder overlap detection, a variety of approaches are used - including projection on a line, separation axis tests, and three-dimensional parallel testing [38–40]. Contact between two ellipses can be determined by solving the quartic equation derived from combining their two equations. Minimization of the geometric potential of the two ellipses can be used to locate intersection points, as described in [41]. Similar contact and overlap detection algorithms for ellipsoids can be found in the literature [31, 42–45].

In contrast to the assemblage of particles with simple shapes, when placing or packing irregularly shaped particles, detecting overlap becomes more challenging. Some applications combine regular shapes to form more complex particle morphologies - such as tablets [46, 47], clusters of spheres [34–36], and sphere-cylinder combinations [38, 39, 46, 47]. Other methods of irregular particle shape representation include X-Ray Computed Tomography (XCT) scans [15], laser scans [48], random Voronoi polygons [49–51], and spherical harmonics [52, 53]. A review of published literature has shown the main approach to overlap detection of particles is calculating the minimum distance between the surfaces of two particles [53, 54]. The minimum distance approach provides accurate detection of overlap, but is computationally expensive. For random particle shapes represented as a cluster of component spheres of differing sizes, each sphere composing the particle in question is checked for overlap with the spheres composing the nearby particles. The accuracy of this method is dependent on the conformity of the sphere cluster to the original particle shape [55, 56]. Another method is to represent each particle's surface as a collection of planes, where the space enclosed by the intersecting planes represents the particle. The overlap test then checks if all the points on other particles lie on opposing sides of each of these planes from the particle in question [57]. Voxel based approaches treat particles as 3D images, composed of cubic voxels. Overlap detection is fast, however there is a tradeoff between the resolution of the particle's surface characteristics and reducing the number of voxels used [58]. A mathematically driven approach can be used to determine exactly where the two particle surfaces intersect, but for star-shaped particles is computationally expensive [59–61]. In this paper, the focus is on determining *if* two particles overlap, not precisely *where* they overlap.

The objective of this research is to develop an efficient, computationally inexpensive method of determining if two virtual, random-shape particles, each represented by a series of spherical harmonics functions, overlap. A novel asperity-based overlap detection algorithm is presented in this paper. This approach improves upon existing particle overlap detection algorithms during random particle placement by identifying, a priori, the asperities on the surface of a particle, which allows for more rapid overlap detection while preserving complex, particle surface geometries.

## 2. Particle Generation

Extensive work characterizing the shape of particles has been performed using X-ray Computed Tomography (XCT) [52]. Spherical harmonics is a mathematical technique that may be used to analyze and represent the shape of star-shaped particles derived from XCT data. A star-shaped particle is defined to be a shape where there exists at least one interior point (e.g. usually the center of mass works), such that any point on the surface of the particle can be connected to this special point with a line segment that lies entirely in the particle. In actuality, in any star-shaped particle, there is a convex subset of the particle containing all points that will suffice for this special point [62]. In this application, spherical harmonics are used to mathematically deform a sphere until it represents the original particle as scanned in the XCT data. The spherical harmonic function is

$$Y_n^m(\theta, \phi) = \sqrt{\frac{(2n+1)(n-m)!}{4\pi(n+m)!}} P_n^m(cos(\theta)) e^{im\phi}, \tag{1}$$

where the term $P_n^m$ represents the associated Legendre functions and $i$ is the square root of -1.

The radius at any set of angles in a spherical coordinate system is

$$r(\theta, \phi) = \sum_{n=1}^{n_{max}} \sum_{m=-n}^{n} a_{nm} Y_n^m(\theta, \phi), \tag{2}$$

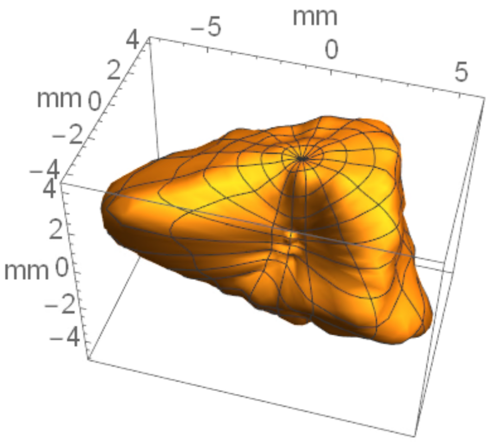


Figure 1: An aggregate particle reconstructed using spherical harmonics

where the azimuth angle, $\theta$, ranges from 0 to $2\pi$ and the polar angle, $\phi$, ranges from 0 to $\pi$. The parameter $a_{nm}$ is the spherical harmonic coefficient and is in principle unique for each $n$ and $|m|$ pair in the absence of any symmetry in the particle shape. Requiring $r(\theta, \phi)$ to be a real number constrains the $a_{nm}$ coefficient to be the conjugate conjugate of $a_{n(-m)}$. It is recommended that Equation (2) is carried out to an $n_{max}$ value of approximately 18 to 26. Additional summation terms will not give an appreciable level of detail and may even go beyond the level of detail captured in the XCT data [52]. A particle reconstructed using this method is shown in Figure 1.

### *2.1. Pre-Rotation of particles*

In order to generate randomly oriented and located particle arrangements, one must in principle, for every new particle, choose a random orientation and location in the unit cell. To match the random orientation, the spherical harmonic coefficients of the particle must be rotated. If the particle cannot fit in the chose location with the chosen orientation, then a new orientation must be generated. In order to reduce the computational time needed to generate a microstructure, a stockpile of pre-rotated particles was created. Each particle was used to make 10 pre-rotated copies of itself. This number is a parameter of the algorithm and can easily be increased as necessary. Pre-rotation of the aggregates was carried out by rotating the $a_{nm}$ coefficients by random angles around each the X-, Y-, and Z-axes. The $a_{nm}$ were rotated using the code provided in Appendix A.1. With the new coefficients, the pre-rotated particles were then re-constructed using Equation (2). An advantage of pre-rotation is that the computation must only be carried out once, and the rotated particles can be used to generate multiple microstructures or representations.

## 3. Asperity Detection

The first and second spatial derivatives of the analytical function describing the surface, given in Equation (2), are used to identify the local maxima in the particle's radii on the particle's surface, called asperities. These maxima can be detected by first determining the locations where the first derivatives with respect to $\phi$ and $\theta$ are equal to zero. For brevity, the parameter $f_{nm}$,

$$f_{nm} = \sqrt{\frac{(2n+1)(n-m)!}{4\pi(n+m)!}}, \tag{3}$$

is used to represent the factorials common to all the derivative equations [52]. The first derivative with respect to $\phi$,

$$r_{\phi} = \sum_{n=1}^{n_{max}} \sum_{m=-n}^{n} (im) a_{nm} Y_n^m(\theta, \phi), \tag{4}$$

and the first derivative with respect to $\theta$,

$$r_\theta = \sum_{n=1}^{n_{max}} \sum_{m=-n}^{n} \frac{-a_{nm} f_{nm}}{sin(\theta)}[(n+1)cos(\theta)P_n^m - (n-m+1)P_{n+1}^m]e^{im\phi}, \tag{5}$$

are equal to zero at locations of minima and maxima in the radius. Next, the second derivative with respect to $\phi$,

$$r_{\phi\phi} = \sum_{n=1}^{n_{max}} \sum_{m=-n}^{n} (-m^2) a_{nm} Y_n^m(\theta, \phi) \tag{6}$$

and the second derivative with respect to $\theta$,

$$\begin{aligned} r_{\theta\theta} = \sum_{n=1}^{n_{max}x} \sum_{m=-n}^{n} (\frac{a_{nm} f_{nm}}{sin^2(\theta)}[(n+1+(n+1)^2 cos^2(\theta))P_n^m \\ -2cos(\theta)(n-m+1)(n+2)P_{n+1}^m + \\ (n-m+1)(n-m+2)P_{n+2}^m]e^{im\phi}, \end{aligned} \tag{7}$$

are assessed [52]. The second angular derivatives are positive in minima locations and negative in maxima locations. Locations in the spherical coordinate system where the first derivatives are zero and the second derivatives are negative are identified as asperities and are stored for each particle. The asperities are unchanging on the particle surface, so their identification need only be performed once. The, coordinates of the asperities, relative to the center of mass of the particle, are properties of the individual particle and *not* the microstructure, and can thus be identified prior to any microstructure generation. The location data are stored and recalled as needed during the particle placement process. Pre-rotating the particles includes rotating the coordinates of the asperity locations for each particle copy.

In this work, the derivatives given in equations (4) - (7) are assessed numerically by calculating the value of the derivatives at incremental values of $\theta$ and $\phi$. Asperities are defined as locations where the first derivative is near zero and the second derivative is less than zero. This approach uses a tolerance of ±0.25 when analyzing the first derivative values due to the use of incremental numerical assessment of the derivatives, as the location of where the first derivatives are exactly zero may not fall exactly on the angle increment used. The number of increments is also a parameter of the algorithm. The increment value of $\frac{\pi}{120}$ have been found to give generally accurate results.

As the asperities of interest are the major maxima in the radii, not minor surface detail, a reduced number of summations can be used in the derivative equations because the lower $n$ values are responsible for shaping the largest features and contours of the particle. As $n$ increases, finer surface features can be resolved. In a subsequent section, the influence of the $n_{max}$ value used during asperity detection on the resulting efficiency improvement is assessed.

## 4. Particle Placement

To generate a virtual 3D microstructure, the particles are randomly placed into a domain with the smallest dimension at least 2.5 times the largest particle size, where particle size is determined by the width of the particle. The length (L), width (W), and thickness (T) of the particles are measured using ASTM D4791 [63]. Length is defined as the longest axis that is fully contained in the aggregate. The width is defined as the longest axis that is both perpendicular to the length axis and is also fully contained in the aggregate. Thickness is taken as a fully enclosed line segment that is perpendicular to each the length and width axes. This work employs cubic domains with periodic boundary conditions for demonstration. The code uses an imported experimental particle size gradation curve or a user-defined gradation to calculate the number of particles from each size range needed to achieve both the desired volume fraction and gradation. Periodic boundary conditions are achieved by enforcing opposite boundaries to be identical to each other. In order to realize this, any particle that intersects the boundary of the domain is cut at the boundary. The piece of the particle that is outside the domain is translated to the opposing side and properly rotated. An example is shown in Figure 2.

The particle placement process is described in Figure 3 and is largely based on the Anm model [59, 64]. First, all of the pre-rotated particles are sorted by size. Starting with the largest particle size range, a random particle in the size

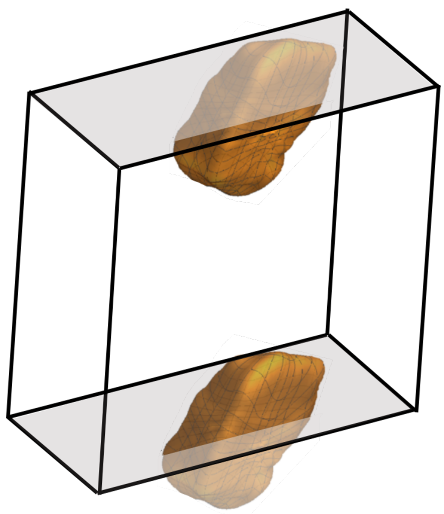

Figure 2: Example of periodic boundary conditions demonstrated with a particle that intersects the wall of the domain

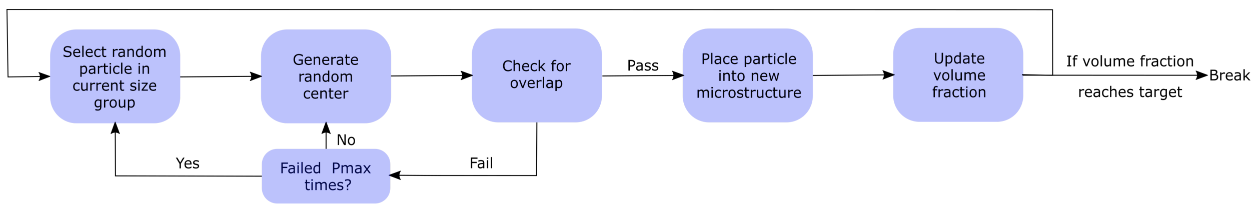


Figure 3: Particle packing algorithm

range is selected for placement. Random center coordinates are generated for the particle and it is temporarily placed into the domain. If the particle intersects any of the domain walls, it is sliced at the wall and the exterior piece is translated to the opposing side of the domain, as described previously in Figure 2. The asperity overlap check, which is described further in Section 5, is performed to determine if the particle is in contact with any of its neighbors. If the particle is found to overlap with another, it is randomly translated to a new, random location in the domain. This process is repeated up to *Pmax* times (typically *Pmax* is set at a value of 30) until the particle finds a suitable position in the domain. If *Pmax* tries are exceeded, the particle is discarded and the code moves on to the next particle in the same size range. Particles that are placed into the domain without overlapping with previously placed particles are then considered a permanent addition to the domain, and the code moves on to select and place the next particle. A completed microstructure is displayed in Figure 4.

## 5. Overlap Detection

Detection of overlap between two virtual particles in this work involves a three-step approach. The overlap check function is divided into three filters. The first, coarse filter compares the distance between the centroid of the particle in question and the centroid of a previously placed particle to the sum of the two particles' maximum radii. Next, the asperities on the surface of the two particles are assessed, which is the new methodology that comprises the novelty of this paper, followed by the third filter that applies a brute force check if necessary.

### *5.1. Radii Comparison*

Each particle has an imaginary bounding sphere with a radius defined to be equal to the maximum of all its radii and a minimum inscribed sphere with a radius defined to be equal to the minimum of all its radii. An enclosing sphere, as opposed to an enclosing box used in [59], was chosen due to compatibility with a spherical coordinate

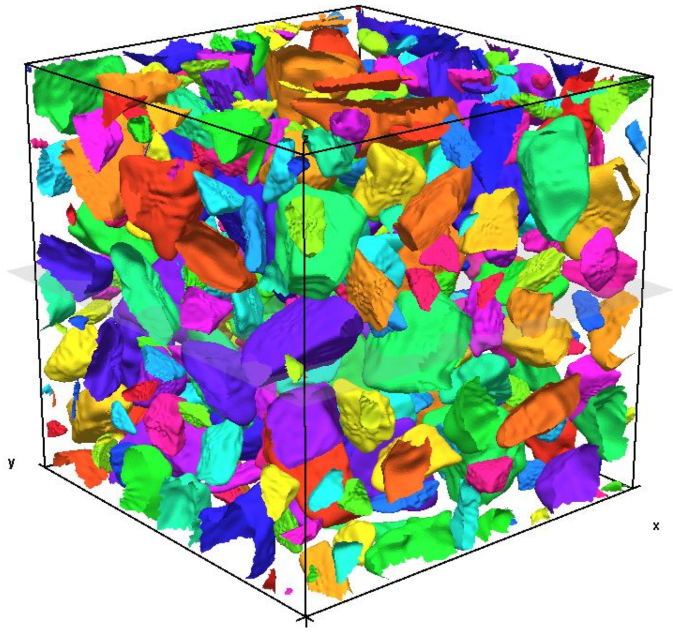


Figure 4: Completed microstructure, packed with 38 % volume fraction of particles. The colors indicate discrete particles

system. The bounding sphere can be imagined as the smallest sphere centered at the particle's center of mass that fully contains the particle. The inscribed sphere is the largest sphere that lies entirely in the interior of the particle's surface while also being centered at the particle's center of mass. These definitions differ from the usual geometric definition of inscribed and circumscribed spheres by requiring that the center of the two spheres be the same and equal to the particle's center of mass. If the distance between the centroids is greater than the sum of the maximum radii, the two particles cannot overlap. If the distance between the centroids is less than the sum of the maximum radii of the respective particles, the program then compares the centroid-centroid distance to the sum of the minimum radii. If the distance between the two centers is less than the sum of their minimum radii, the particles must overlap. This approach quickly determines if two particles do not overlap. However, the indications of overlap are mainly false positives, since it treats the particles as spheres. If the minimum radii assessment is passed, the algorithm advances to the asperity check.

### *5.2. Asperity Check*

The overlap between the two enclosing or bounding spheres forms a lens of potential intersection as seen in Figure 5. This lens represents the only 3D space where overlap between the two particles is possible. Only the asperities located in the lens of potential intersection are assessed in this filter.

Starting with the particle in question, the asperities on each particle, identified from the previously stored list of all the surface asperities for those two particles, are filtered to retain only those present in the lens of potential intersection. The locations are retrieved and the vector between an asperity on one particle and the center of the other particle is established. The vector's magnitude is compared to the radius of the other particle along the direction of the vector, as shown in Figure 6. If the radius is shorter than the vector's length, there is no overlap at that asperity and the code moves on to assessing the remaining asperities in the lens of potential intersection. If the radius is longer, overlap has been detected and the function exits. This radius-distance comparison was initially seen in [59], however it was used to sweep the entire region of potential intersection, whereas this approach targets specific, predetermined locations within the lens of potential intersection. If all the asperities from the first particle are checked and no overlap is found, the function begins checking the asperities on the second particle. If overlap is detected the function immediately stops and exits.

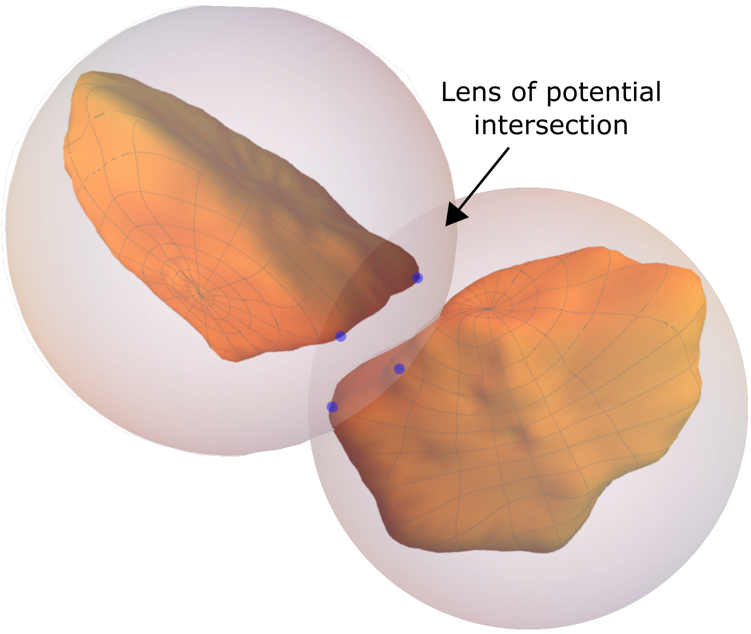


Figure 5: Lens of potential intersection formed by intersecting bounding spheres. Only asperities that lie in this lens are checked for overlap with the neighboring particle. In this example, the asperities that lie in this lens are identified with blue markers.

A similar approach is found in [53], where the $L_2$ norm of Equation (2), is used to determine the minimum distance between two particles. This approach differs from that in [53], where the radius equations for each pair of analyzed particles are assessed to find the minimum distance each time they are checked for overlap. In this work, the approach uses the radius equation's spatial derivatives to identify local radius maxima or asperities and stores the information for recall during future overlap checks.

*5.3. Brute Force Check*

If all asperities pass the previous assessment, so the particles appear to not overlap, the code applies the third filter - a brute force comparison of the surfaces. The brute force method assesses the point clouds of the two particles to determine if the points in the lens of intersection on one particle lies within the surface of the other particle. This approach is the most robust and will catch all instances of overlap, however it is computationally intensive. The brute force check operates with two filters. The first checks if any points on the surface of particle A lie within the inscribed sphere of particle B. If any points on particle A lie within this sphere, there is overlap between the two particles and the function exits. Otherwise, the function advances to the second filter, which filter relies on MATLAB's[1] built-in function "alphaShape", which creates a bounding region around a set of points. In this application, the alphaShape is a 3D surface that forms to the morphology defined by the surface points of the aggregate. As the number of points used to define the surface of each 3D particle in this work is high, 28,800 points, generated by assessing 120 and 240 increments of $\phi$ and $\theta$, respectively, the bounding region is very closely aligned with the morphology of the particle. Previous work by Garboczi (2017) found that 120 increments of $\phi$ and $\theta$ provided generally accurate results [65]. The subfunction, "inShape" assesses whether a specified point or set of points lie within the alphaShape. The most straightforward approach to using the alphaShape is to generate an alphaShape of the particle in question, herein referred to as particle A, and assess if the surface points of nearby particles lie within the alphaShape of particle A. While this catches *most* of the overlap between particle A and neighboring particles, one important exception exists - when particle A lies entirely inside another particle, referred to as particle B, such as the example shown in Figure 7.

[1]Certain commercial equipment, instruments, or materials are identified in this paper to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

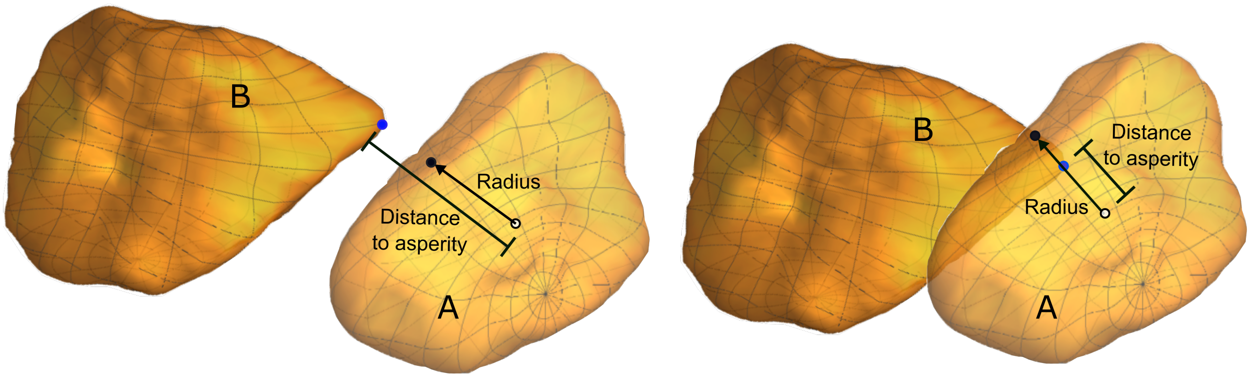


Figure 6: Assessing overlap by comparing the distance from the center of particle A (identified with a white marker) to the asperity on particle B (identified with a blue marker) with the radius of particle A along the same vector (identified with a black marker). On the left, the radius is shorter than the distance to the asperity, indicating there is not overlap with the asperity point, while on the right the radius is longer, leading to the detection of overlap.

This scenario is caught and prevented during the asperity check, because for any asperity on particle A, the radius of B will be larger than the distance to the asperity - resulting in detection of overlap. However the control copy of the code (i.e., the comparison code without the asperity check) needs an additional layer of consideration, because if particle A lies entirely within particle B, the points on the surface of particle B will not be detected as being 'in' the alphaShape of particle A. In order to detect this, the code checks if all points on particle A lie within the bounding sphere of particle B. If all of the points on particle A fall in the bounding sphere, the code generates an alphaShape of particle B and assesses if the points on A are located within it. In the case where part of particle A lies outside of the bounding sphere, there is no possibility of particle A being fully enclosed within particle B, thus the code uses the previously generated alphaShape of particle A to assess if any points on B lie within the alphaShape of A. Each neighboring particle is assessed in this manner in order to perform only one alphaShape assessment per pair of particles.

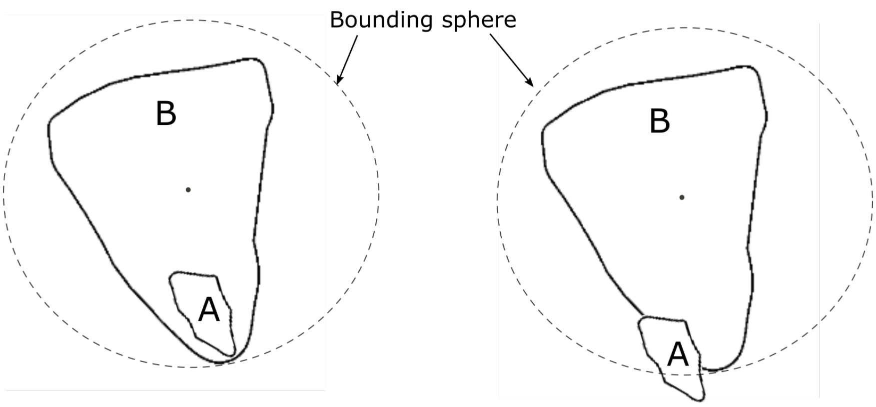


Figure 7: Two arrangements of particle A with respect to particle B - fully contained in the bounding sphere of particle B (left) and intersecting the bounding sphere of particle B (right)

## 6. Particle Shape Analysis

Four particle size distributions were studied to determine the impact of the asperity check on computation time in a variety of packing scenarios. The particles are divided into four size groups based on their width. These gradations are seen in Figure 8. Gradations A and B contain greater proportions of larger particles, whereas gradations C and D contain a finer mix of particles. Gradation curve C is based on the Fuller curve, adapted to consider only particles larger than 4.75 mm width[66].

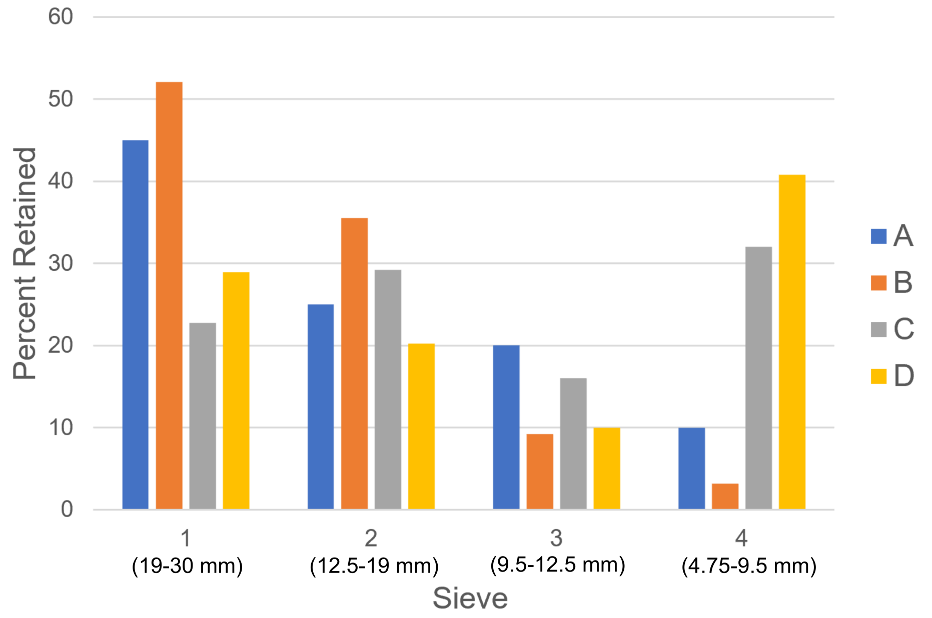


Figure 8: The four gradations studied - with two relatively coarse gradations (A and B), alongside two finer gradations (C and D)

To provide context for the particles used in this work, several shape parameters are assessed and presented here. As previously discussed, the length, width, and thickness of each particle is measured as defined in ASTM D4791 [63]. The ratio of the length and thickness of the particle, generally referred to as the aspect ratio, is assessed alongside the length to width ratio, and the ratio of the maximum and minimum radii. Additionally, the Wadell sphericity parameter, defined as

$$S_w = \frac{\text{Surface area of volume-equivalent sphere}}{\text{Surface area of particle}}, \tag{8}$$

is used to describe the roundness of a particle on a scale of 0 to 1, where a $S_w$ value of 1.0 is equivalent to a perfect sphere [67]. As seen in Figure 9, the shape parameters are relatively consistent between the four particle size gradations, indicating that the four gradations assessed in the next section can be compared simply without needing to account for differing particle shape ratios in the resulting microstructures. Note that the vertical bars shown in Figure 9 are not uncertainty limits, but show the standard deviation found by computing over the particles in question, reflecting the range of shapes.

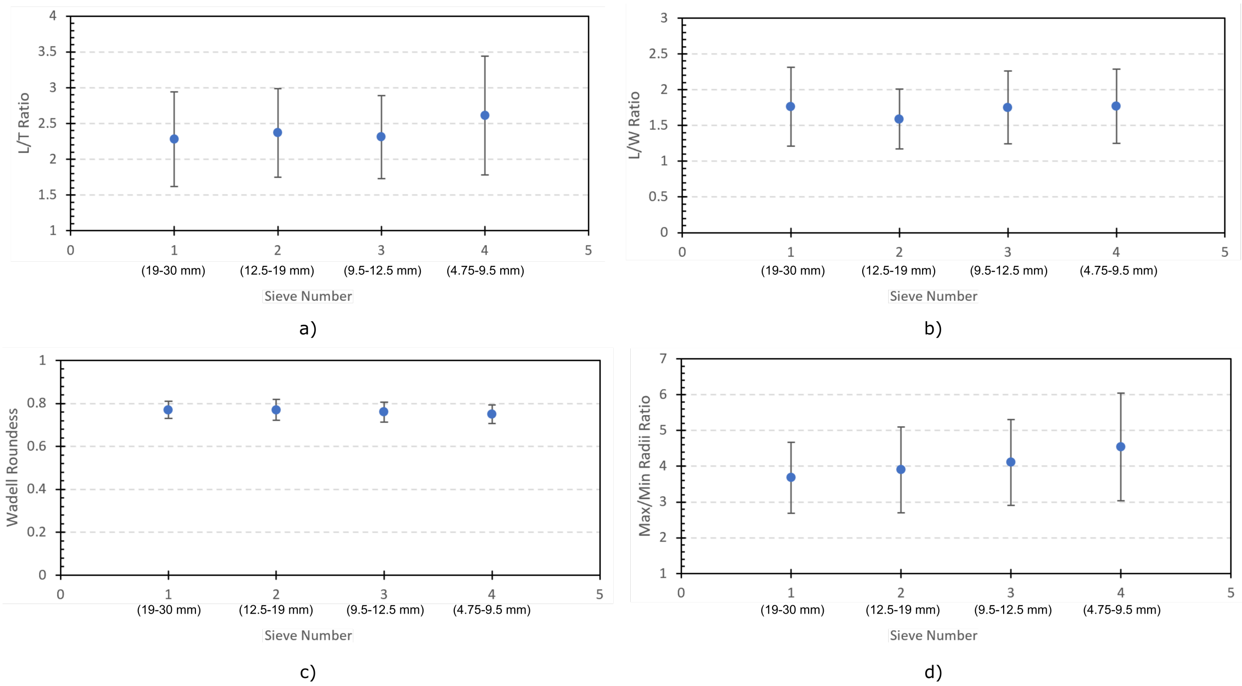


Figure 9: Shape parameters for each of the four size ranges sizes: a) length-thickness (L/T) ratio, b) length-width (L/W) ratio, C) Wadell sphericity, and d) maximum-minimum radius ratio. Error bars represent ± one standard deviation

## 7. Results & Discussion

All microstructure generation was carried out using Matlab R2018a on the Texas A&M University High Performance Computing Center's supercomputer Ada, a hybrid cluster from IBM with Intel Ivy Bridge processors and a Mellanox FDR-10 Infiniband interconnect. The plotting code was carried out serially, placing one particle at a time, on a single node with 2500 MB of allotted memory. For each gradation, three microstructures were generated at three particle volume fractions: 20 %, 30 %, and 40 %. The control code is a copy of the plotting code with the asperity check removed and an additional check in the brute force filter that assesses if a particle is fully enclosed by another.

### *7.1. Influence of $n_{max}$ Value*

As the asperities of interest are the major maxima in the radii, not the surface detail, a reduced number of summations can be used in the derivative equations. In Figure 10, an example of asperity detection using a variety of $n_{max}$ values is depicted.

To quantify the level of influence that the $n_{max}$ parameter had during asperity detection, the set of pre-rotated particles was run through the asperity detection code four times - using $n_{max}$ values of 6, 8, 10, and 12. The $n_{max}$ value of 4 was not considered due to the notable morphological difference between the particle generated with an $n_{max}$ value of 4 and the particles generated with higher $n_{max}$ values (see Figure 10).

Since the mean and median number of asperities detected by the varying $n_{max}$ values in Table 1 varies by up to 15 asperities per particle, a test to determine the optimal $n_{max}$ value to use during the asperity detection process was conducted. Three microstructures of 40 % volume fraction coarse particle were generated using each set of asperities. The average runtime of the microstructure generation for each $n_{max}$ value using gradation A is shown in Figure 11.

It was found that the $n_{max}$ values used during asperity detection did not cause a statistically significant difference in the overlap detection time, as each of the error bars overlapped. The lack of difference is likely due to the major asperities being visible and detected at an $n_{max}$ value as low as 4. The slightly lower computational times for the higher $n_{max}$ values indicate that the code does not need to cycle through several asperities within the lens of potential intersection prior to identifying an intersection if one exists. Furthermore, for very low values of $n_{max}$ used to identify

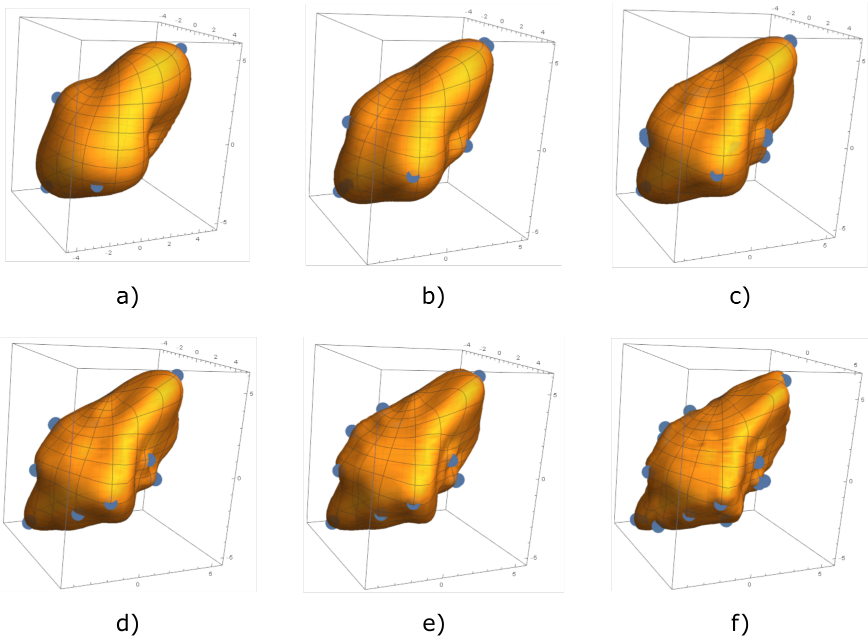


Figure 10: Asperities identified on a particle plotted and analyzed with an $n_{max}$ value of a) 4, b) 6, c) 8, d) 10, e) 12, f) 20. The largest asperities are identified on all, with increasingly subtle asperities detected as $n_{max}$ increases

Table 1: Mean and median number of asperities identified on each particle for varying $n_{max}$ values.

| $n_{max}$ Value | Mean | Median |
|---|---|---|
| 6 | 18.2 | 16 |
| 8 | 23.2 | 21 |
| 10 | 28.3 | 26 |
| 12 | 33.1 | 31 |

asperities, there may be some potential intersections that are missed by the asperity filter, leading to more detections in the (computationally more demanding) brute force filter. As seen in Figure 10, the morphology of the particle generated with an $n_{max}$ value of 4 is notably morphologically smoother and simpler than the particles with a higher $n_{max}$ value. In the remainder of this work, the asperities are detected using $n_{max} = 10$, as this value provides a balance of computational speed and morphological detail during the asperity detection process. The particles themselves were generated with an $n_{max}$ value ranging between 18 and 26.

### 7.2. *Efficiency Improvements*

Tables 2 thru 5 shows the runtime data differences between microstructure generation with and without the asperity check, giving the average runtime ± the standard deviation of the three runs. The edge length of the cubic microstructure was 75 mm and the largest particle size was 30 mm. Speedup is a parameter describing the improvement in code performance, according to

$$\text{Speedup} = \frac{\text{Average Runtime of Control Code}}{\text{Average Runtime of Code with Asperity Check}}. \tag{9}$$

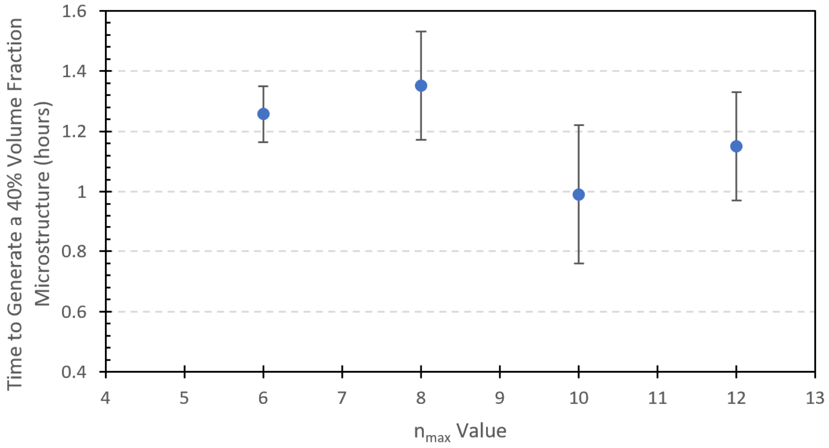


Figure 11: The time to generate a 40 % volume fraction microstructure, using gradation A, with asperities identified using an $n_{max}$ value of 6, 8, 10, and 12. Each data point represents the average runtime of generating three microstructures. Error bars represent ± one standard deviation

Table 2: Runtime and improvement results for each volume fraction using Gradation A

| Volume Fraction | Avg. Runtime ± Std. Dev. (hrs) | Avg. Runtime Control ± Std. Dev. (hrs) | Avg. Speedup |
|---|---|---|---|
| 20 % | 0.05 ± 0.0001 | 0.19 ± 0.009 | 3.7 |
| 30 % | 0.12 ± 0.0009 | 0.94 ± 0.16 | 7.6 |
| 40 % | 1.45 ± 0.55 | 12.24 ± 3.1 | 8.5 |

Table 3: Runtime and improvement results for each volume fraction using Gradation B

| Volume Fraction | Avg. Runtime ± Std. Dev. (hrs) | Avg. Runtime Control ± Std. Dev. (hrs) | Avg. Speedup |
|---|---|---|---|
| 20 % | 0.03 ± 0.001 | 0.11 ± 0.02 | 3.4 |
| 30 % | 0.16 ±0.02 | 1.08 ± 0.14 | 6.7 |
| 40 % | 2.92 ± 1.1 | 36.9± 8.5 | 12.6 |

Table 4: Runtime and improvement results for each volume fraction using Gradation C

| Volume Fraction | Avg. Runtime ± Std. Dev. (hrs) | Avg. Runtime Control ± Std. Dev. (hrs) | Avg. Speedup |
|---|---|---|---|
| 20 % | 0.11 ± 0.003 | 0.42 ±0.015 | 3.8 |
| 30 % | 0.54 ± 0.11 | 4.3 ± 0.89 | 7.9 |
| 40 % | 10.37 ± 1.8 | 91.3 ± 25.3 | 8.8 |

Table 5: Runtime and improvement results for each volume fraction using Gradation D

| Volume Fraction | Avg. Runtime ± Std. Dev. (hrs) | Avg. Runtime Control ± Std. Dev. (hrs) | Avg. Speedup |
|---|---|---|---|
| 20 % | 0.15 ± 0.004 | 0.52 ± 0.02 | 3.4 |
| 30 % | 0.68 ± 0.014 | 4.6 ± 0.77 | 6.7 |
| 40 % | 9.52 ± 0.2 | 98.9 ± 7.5 | 10.3 |

The speedup values for each volume fraction and gradation are shown together in Figure 12, where gradations A and B are the coarser gradations and C and D are the finer gradations. The average speedup demonstrated across the four gradations is 3.6, 7.2, and 10.0 for the 20 %, 30 %, and 40 % volume fractions, respectively. While the plotting code executes serially, the plotting time is, on average, reduced by over 70 % in the lowest volume fraction and over 85 % in the higher volume fractions by utilizing an asperity-based overlap detection algorithm. It can be seen that the average time to generate the microstructure increases with the fineness of the particle size gradation, which can be directly linked to the number of particles placed in order to reach the desired volume fraction. Furthermore, the relative speedup values for the higher volume fractions are greater than the lower volume fractions, which is the result of more particles plotted, and thus more overlap checks, to reach the desired volume fraction. Also, for lower volume fractions, it is increasingly likely that the bounding sphere overlap checks will suffice to show that a particle to be placed does not overlap any existing particle. As more particles are placed, at higher volume fractions, the aggregates on average will be closer together, meaning that the bounding sphere overlap checks will show potential overlap with a higher number of particles. Therefore, at higher volume fractions, speedup is increasingly dominated by the speed difference between the asperity check and the brute force check.

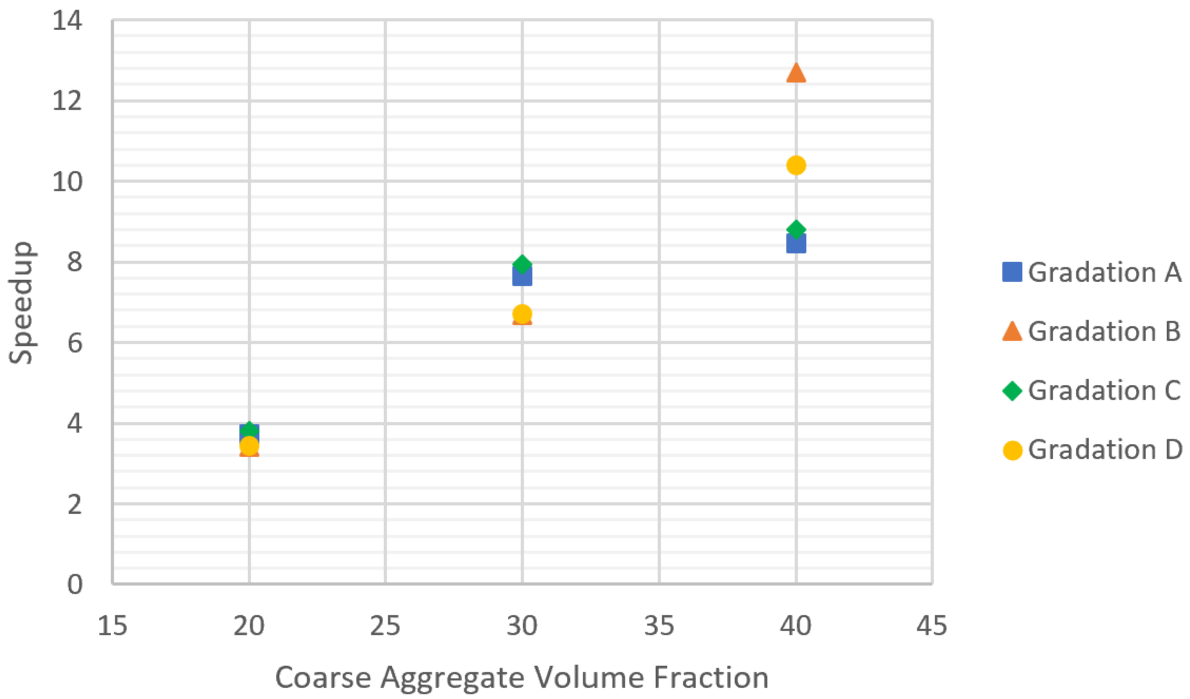


Figure 12: Combined speedup data of each volume fraction and particle size gradation

When reaching high percentages of runtime reduction, a small percentage difference can result in a large difference in speedup values. The relationship between the percentage reduction and speedup is

$$\text{Percentage difference} = 100 * \left(1 - \frac{1}{\text{Speedup}}\right), \tag{10}$$

meaning that high values of speedup don't yield large changes to the percentage due to the reciprocal relationship. As a result, this can lead to the appearance of a speedup difference between data sets when the percentage difference is actually small. For example, consider the average speedup values of gradations A and B for the 40 % microstructure generations. Gradation A has a speedup of 8.5, correlating to a percentage reduction of runtime of 88.2 %. Gradation B averages a speedup of 12.6, giving a percent reduction of runtime of 92.1 %. This difference of speedup values is 4.1, giving the appearance of a substantial difference, however the percentage reduction of runtime difference is 3.9

Table 6: Average percentage reduction in time to generate a microstructure for each volume fraction and particle size gradation combination

| Gradation | 20 % | 30 % | 40 % |
|---|---|---|---|
| A | 73.1 % | 86.9 % | 88.2 % |
| B | 70.1 % | 85.0 % | 92.1 % |
| C | 73.4 % | 87.4 % | 88.6 % |
| D | 70.9 % | 85.1 % | 90.4 % |

%. In order to most accurately represent the data, both the speedup values and percentage reduction of runtime values are presented in this paper. Figure 13 visualizes the similarity in the percentage reductions for each gradation and volume fraction. These data are the same as that of Figure 9, however it is presented in terms of percentage reduction in runtime to provide perspective on the gaps in the speedup values between the gradations.

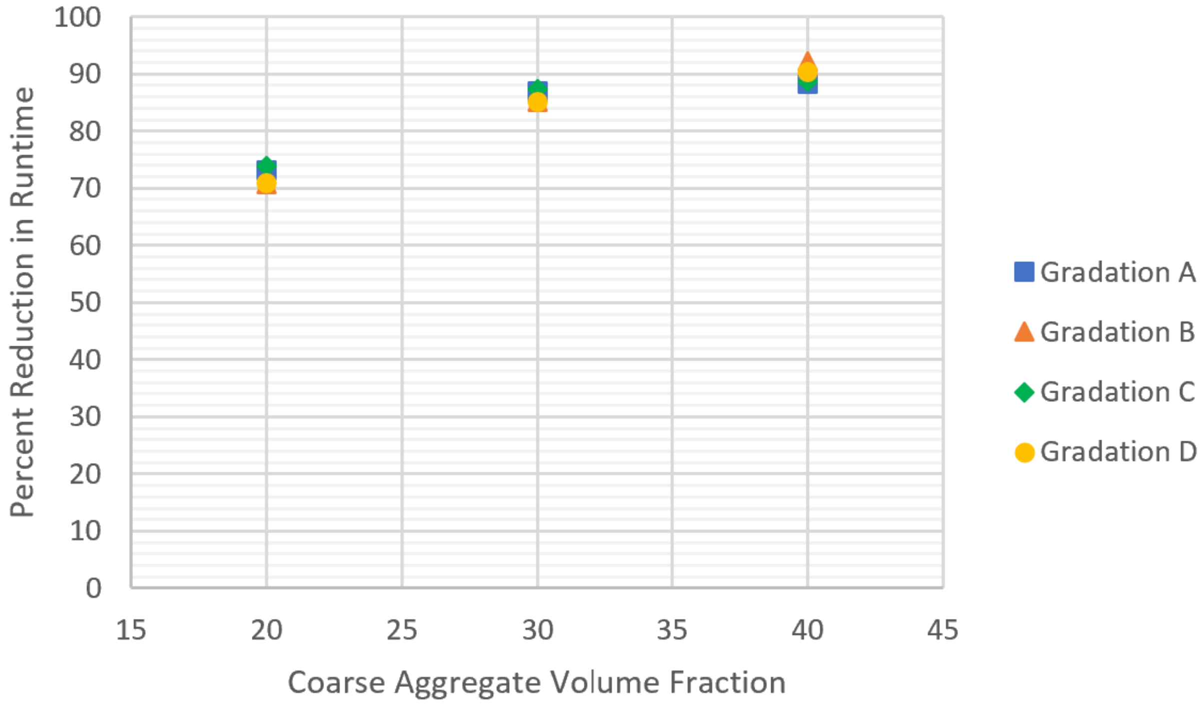


Figure 13: Average percentage reduction in runtime for each volume fraction and particle size gradation

*7.3. Proportion of Overlap Detection Methods*

While the improvements in efficiency due to the addition of the asperity check are clear, to better understand these improvements, counters for the proportion of overlap detected by each of the three filters were added to the code to track each instance of overlap detection. The data collected from these counters are given in Table 7, where the average proportion of overlap detected by each of the three overlap detection methods implemented in this code is presented at a volume fraction of 40 %. The data are taken as a simple average of overlap detection counters. The minimum proportion of overlap detected through the asperity-radii comparison is 86.4 %, confirming that this approach is highly effective. The proportion of overlap that makes its way to the brute force check is very low, at a maximum of 0.6 %. This data explains why the reductions in runtime given in Figure 13 are so substantial.

Table 7: Average proportion of overlap detected by each filter and average number of overlap instances detected for each particle placed in a 40 % volume fraction microstructure generated with gradations A-D

| Gradation | Minimum Radii | Asperity Check | Brute Force | Mean Number of Overlap Detections per Particle | Median Number of Overlap Detections per Particle |
|---|---|---|---|---|---|
| A | 13.1 % | 86.4 % | 0.5 % | 121 | 32 |
| B | 13.3 % | 86.5 % | 0.2 % | 440 | 48 |
| C | 10.7 % | 89.2 % | 0.1 % | 258 | 53 |
| D | 12.8 % | 86.6 % | 0.6 % | 376 | 49 |

Additionally, mean and median number of overlap instances detected for each particle are given in Table 7. Given the large difference in the mean and median values, to more thoroughly understand the data, the number of overlap detections for each particle is presented in Figure 14, where the number of overlap detections is considered to be the number of tries to place the particle as it is placed into the domain. A notable difference between the placement tries required for coarse gradations, A and B, and the finer gradations, C and D, can be seen. In Figure 14 the coarser gradations result in a smaller number of larger spaces in the microstructure, have a lower number of placement attempts for particles in size range #4, the smallest size range, than that of size range #2 and #3. The opposite is true of the finer gradations where it can be seen that the number of placement tries increases sharply for particles in size range #4. This is reasoned to be due to the high number of large particles placed in the domain in gradations A and B, making it easier to place the final, smallest aggregates. Due to the higher number of large particles in gradations A and B, there are overall fewer particles needed to reach the desired volume fraction in the coarse gradations than in the finer gradations. Furthermore, the finer gradations have a higher number of smaller particles from the smallest size range size that generate many smaller spaces in the microstructure, making the placement of the smallest particles more difficult because there are so many of them. It is the placement of these small particles that is the most time consuming due to the challenge of finding a vacant 3D space in the microstructure that is large enough and of suitable orientation such that the particle does not overlap with any neighbors.

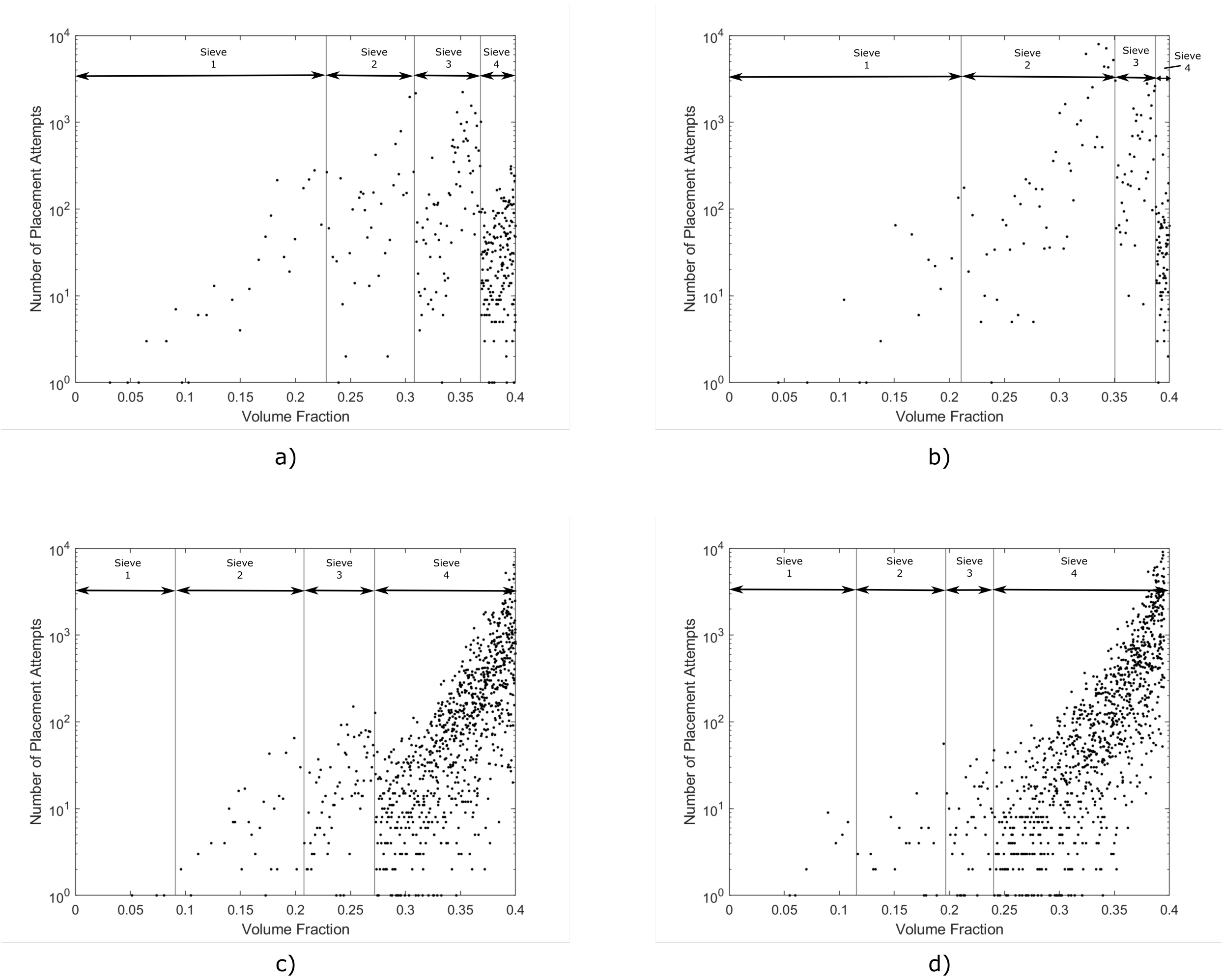


Figure 14: The number of overlap events detected for each particle placed in the domain for one microstructure of 40 % volume fraction for gradations A a), B b), C c), and D d).

## 8. Summary

As demonstrated, the asperity check substantially reduces the runtime to generate a microstructure of realistic volume fractions, averaging a runtime 10x faster than the control code for a 40 % particle volume fraction. The particles used in this study were reconstructions of gravel, however this approach is useful for any type of particle described by any analytical function with a real second derivative or with otherwise identifiable surface asperities. The code used in this work places particles serially; additional improvements in efficiency would be seen if at least some of the code could be executed in parallel. Additional future work could look at the impact of various shape parameters on the efficiency improvements to determine what kinds of particles are best suited to this method. The value of $n_{max}$ used during asperity detection was shown to not play an impactful role on the level of efficiency improvement but more work needed to more fully understand if there is a link between particle morphology and level of efficiency improvement. The following conclusions can be drawn from this work:

- Particles described by analytical functions can be assessed for the detection of surface asperities
- In the spherical harmonics reconstruction of the particles used in this work, the $n_{max}$ value used during asperity detection did not play a large role in the computational runtime during the generation of 40 % volume fraction microstructure.

- Using the asperity detection algorithm, the reduction in runtimes were more than 85 % for microstructures containing greater than 30 % volume fraction coarse particles.
- The speedup values increase with increasing volume fraction. The average speedup values were 3.6, 7.2, and 10.0 for the 20 %, 30 % and 40 % volume fractions, respectively, averaged over the four particle size gradations studied.
- The asperity check detected between an average of 85 % and 90 % of all particle-particle overlap in 40 % volume fraction microstructures of each gradation. Less than 1 % made it to the much less efficient brute force check

## 9. Acknowledgements

This research was supported by a Nuclear Energy University Partnership (NEUP) grant 16-10457 through US DOE, Office of Nuclear Energy, Light Water Reactor Sustainability (LWRS) Program. Additional support came from the US Army Research and Development Center (ERDC) grant. Portions of this research were conducted with high performance research computing resources provided by Texas A&M University (https://hprc.tamu.edu).

## References

[1] C. F. Ferraris, N. S. Martys, Relating fresh concrete viscosity measurements from different rheometers, Journal of research of the National Institute of Standards and Technology 108 (3) (2003) 229.

[2] J. Zheng, X. An, M. Huang, GPU-based parallel algorithm for particle contact detection and its application in self-compacting concrete flow simulations, Computers and Structures 112-113 (2012) 193–204. doi:10.1016/j.compstruc.2012.08.003.

[3] J. Hu, K. Wang, Effect of coarse aggregate characteristics on concrete rheology, Construction and Building Materials 25 (3) (2011) 1196–1204. doi:10.1016/j.conbuildmat.2010.09.035.

[4] M. R. Geiker, M. Brandl, L. N. Thrane, L. F. Nielsen, On the effect of coarse aggregate fraction and shape on the rheological properties of self-compacting concrete, Cement, Concrete and Aggregates 24 (1) (2002) 3–6. doi:10.1520/cca10484j.

[5] F. Dorai, C. Moura Teixeira, M. Rolland, E. Climent, M. Marcoux, A. Wachs, Fully resolved simulations of the flow through a packed bed of cylinders: Effect of size distribution, Chemical Engineering Science 129 (2015) 180–192. doi:10.1016/j.ces.2015.01.070.

[6] S. T. Erdoğan, N. S. Martys, C. F. Ferraris, D. W. Fowler, Influence of the shape and roughness of inclusions on the rheological properties of a cementitious suspension, Cement and Concrete Composites 30 (5) (2008) 393–402. doi:10.1016/j.cemconcomp.2008.01.003.

[7] H.-K. Man, J. G. M van Mier, Size effect on strength and fracture energy for numerical concrete with realistic aggregate shapes 154 (2008) 61–72. doi:10.1007/s10704-008-9270-y.

[8] G. Giaccio, R. Zerbino, Failure mechanism of concrete: combined effects of coarse aggregates and strength level, Advanced Cement Based Materialsdoi:10.1016/S1065-7355(97)00014-X.

[9] S. Dehghanpoor Abyaneh, H. S. Wong, N. R. Buenfeld, Modelling the diffusivity of mortar and concrete using a three-dimensional mesostructure with several aggregate shapes, Computational Materials Sciencedoi:10.1016/j.commatsci.2013.05.024.

[10] L. Liu, D. Shen, H. Chen, W. Xu, Aggregate shape effect on the diffusivity of mortar: A 3D numerical investigation by random packing models of ellipsoidal particles and of convex polyhedral particles, Computers and Structuresdoi:10.1016/j.compstruc.2014.07.022.

[11] J. Lin, H. Chen, Effect of particle morphologies on the percolation of particulate porous media: A study of superballs, Powder Technology 335 (2018) 388–400. doi:10.1016/j.powtec.2018.05.015.

[12] J. W. Bullard, E. J. Garboczi, A model investigation of the influence of particle shape on portland cement hydration, Cement and Concrete Research 36 (6) (2006) 1007–1015. doi:10.1016/j.cemconres.2006.01.003.

[13] D. P. Bentz, E. J. Garboczi, C. J. Haecker, O. M. Jensen, Effects of cement particle size distribution on performance properties of Portland cement-based materials, Cement and Concrete Research 29 (10) (1999) 1663–1671. doi:10.1016/S0008-8846(99)00163-5.

[14] B. Osbaeck, V. Johansen, Particle Size Distribution and Rate of Strength Development of Portland Cement, Journal of the American Ceramic Society 72 (2) (1989) 197–201. doi:10.1111/j.1151-2916.1989.tb06101.x.
URL http://doi.wiley.com/10.1111/j.1151-2916.1989.tb06101.x

[15] F. Bernachy-Barbe, B. Bary, Effect of aggregate shapes on local fields in 3D mesoscale simulations of the concrete creep behavior, Finite Elements in Analysis and Designdoi:10.1016/j.finel.2019.01.001.

[16] E. Piotrowska, Y. Malecot, Y. Ke, Experimental investigation of the effect of coarse aggregate shape and composition on concrete triaxial behavior, Mechanics of Materialsdoi:10.1016/j.mechmat.2014.08.002.

[17] A. C. Aydin, A. Arslan, R. Gül, Mesoscale simulation of cement based materials' time-dependent behavior, Computational Materials Sciencedoi:10.1016/j.commatsci.2007.02.012.

[18] L. W. Rong, Z. Y. Zhou, A. B. Yu, Lattice-Boltzmann simulation of fluid flow through packed beds of uniform ellipsoids, Powder Technology 285 (2015) 146–156. doi:10.1016/j.powtec.2015.06.047.

[19] J. Gan, Z. Zhou, A. Yu, Effect of particle shape and size on effective thermal conductivity of packed beds, Powder Technology 311 (2017) 157–166. doi:10.1016/j.powtec.2017.01.024.

[20] C. Argento, D. Bouvard, Modeling the effective thermal conductivity of random packing of spheres through densification, International Journal of Heat and Mass Transfer 39 (7) (1996) 1343–1350. doi:10.1016/0017-9310(95)00257-X.

[21] A. Jagota, C. Y. Hui, The effective thermal conductivity of a packing of spheres, Journal of Applied Mechanics, Transactions ASME 57 (3) (1990) 789–791. doi:10.1115/1.2897096.

[22] L. Y. Yi, K. J. Dong, R. P. Zou, A. B. Yu, Coordination number of the packing of ternary mixtures of spheres: Dem simulations versus measurements, Industrial and Engineering Chemistry Research 50 (14) (2011) 8773–8785. doi:10.1021/ie200765h.

[23] L. Y. Yi, K. J. Dong, R. P. Zou, A. B. Yu, Radical tessellation of the packing of spheres with a log-normal size distribution, Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 92 (3). doi:10.1103/PhysRevE.92.032201.

[24] K. Han, Y. T. Feng, D. R. Owen, Sphere packing with a geometric based compression algorithm, Powder Technology 155 (1) (2005) 33–41. doi:10.1016/j.powtec.2005.04.055.

[25] A. Z. Zinchenko, Algorithm for Random Close Packing of Spheres with Periodic Boundary Conditions, Tech. rep. (1994).

[26] F. Y. Fraige, P. A. Langston, G. Z. Chen, Distinct element modelling of cubic particle packing and flow, Powder Technology 186 (3) (2008) 224–240. doi:10.1016/j.powtec.2007.12.009.

[27] W. Zhang, K. E. Thompson, A. H. Reed, L. Beenken, Relationship between packing structure and porosity in fixed beds of equilateral cylindrical particles, Chemical Engineering Science 61 (24) (2006) 8060–8074. doi:10.1016/j.ces.2006.09.036.

[28] A. B. Y. R. P. Zou, Wall effect on the packing of cylindrical particles (1996). doi:10.1016/S0009-2509(96)80017-0.

[29] Z. Y. Zhou, R. P. Zou, D. Pinson, A. B. Yu, Dynamic simulation of the packing of ellipsoidal particles, Industrial and Engineering Chemistry Research 50 (16) (2011) 9787–9798. doi:10.1021/ie200862n.

[30] W. Man, A. Donev, F. H. Stillinger, M. T. Sullivan, W. B. Russel, D. Heeger, S. Inati, S. Torquato, P. M. Chaikin, Experiments on random packings of ellipsoids, Physical Review Letters 94 (19). doi:10.1103/PhysRevLett.94.198001.

[31] X. Lin, T. g, Contact detection algorithms for three-dimensional ellipsoids in discrete element modelling, International Journal for Numerical and Analytical Methods in Geomechanics 19 (9) (1995) 653–659. doi:10.1002/nag.1610190905.

[32] X. Lin, T. T. Ng, A three-dimensional discrete element model using arrays of ellipsoids, Geotechnique 47 (2) (1997) 319–329. doi:10.1680/geot.1997.47.2.319.

[33] P. A. Cundall, O. D. L. Strack, A discrete numerical model for granular assemblies, Géotechnique 29 (1) (1979) 47–65. doi:10.1680/geot.1979.29.1.47.
URL http://www.icevirtuallibrary.com/doi/10.1680/geot.1979.29.1.47

[34] Y. Fu, L. Wang, C. Zhou, 3D clustering DEM simulation and non-invasive experimental verification of shear localisation in irregular particle assemblies, International Journal of Pavement Engineering 11 (5) (2010) 355–365. doi:10.1080/10298436.2010.493585.
URL http://www.tandfonline.com/doi/abs/10.1080/10298436.2010.493585

[35] M. Lu, ·. G. R. Mcdowell, The importance of modelling ballast particle shape in the discrete element method 9 (2007) 69–80. doi:10.1007/s10035-006-0021-3.

[36] S. Abedi, A. A. Mirghasemi, Particle shape consideration in numerical simulation of assemblies of irregularly shaped particles, Particuology 9 (4) (2011) 387–397. doi:10.1016/j.partic.2010.11.005.

[37] K. Kildashti, K. Dong, B. Samali, An accurate geometric contact force model for super-quadric particles, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112774. doi:10.1016/j.cma.2019.112774.

[38] M. Kodam, R. Bharadwaj, J. Curtis, B. Hancock, C. Wassgren, Cylindrical object contact detection for use in discrete element method simulations. Part I - Contact detection algorithms, Chemical Engineering Science 65 (22) (2010) 5852–5862. doi:10.1016/j.ces.2010.08.006.

[39] M. Kodam, R. Bharadwaj, J. Curtis, B. Hancock, C. Wassgren, Cylindrical object contact detection for use in discrete element method simulations, Part II-Experimental validation, Chemical Engineering Science 65 (22) (2010) 5863–5871. doi:10.1016/j.ces.2010.08.007.

[40] J. S. Ketchel, P. M. Larochelle, DETC05/MECH-84699 COLLISION DETECTION OF CYLINDRICAL RIGID BODIES USING LINE GEOMETRY, Tech. rep.

[41] J. M. Ting, A robust algorithm for ellipse-based discrete element modelling of granular materials, Computers and Geotechnics 13 (3) (1992) 175–186. doi:10.1016/0266-352X(92)90003-C.

[42] A. D?iugys, B. Peters, A new approach to detect the contact of two-dimensional elliptical particles, International Journal for Numerical and Analytical Methods in Geomechanics 25 (15) (2001) 1487–1500. doi:10.1002/nag.180.
URL http://doi.wiley.com/10.1002/nag.180

[43] X. Jia, Y. K. Choi, B. Mourrain, W. Wang, An algebraic approach to continuous collision detection for ellipsoids, Computer Aided Geometric Design 28 (3) (2011) 164–176. doi:10.1016/j.cagd.2011.01.004.

[44] J. W. Perram, M. S. Wertheim, Statistical mechanics of hard ellipsoids. I. Overlap algorithm and the contact function, Journal of Computational Physics 58 (3) (1985) 409–416. doi:10.1016/0021-9991(85)90171-8.

[45] J. M. Ting, M. Khwaja, L. R. Meachum, J. D. Rowell, An ellipse-based discrete element model for granular materials, International Journal for Numerical and Analytical Methods in Geomechanics 17 (9) (1993) 603–623. doi:10.1002/nag.1610170902.
URL http://doi.wiley.com/10.1002/nag.1610170902

[46] Y. Song, R. Turton, F. Kayihan, Contact detection algorithms for DEM simulations of tablet-shaped particles, Powder Technology 161 (1) (2006) 32–40. doi:10.1016/j.powtec.2005.07.004.

[47] M. Kodam, J. Curtis, B. Hancock, C. Wassgren, Discrete element method modeling of bi-convex pharmaceutical tablets: Contact detection algorithms and validation, Chemical Engineering Science 69 (1) (2012) 587–601. doi:10.1016/j.ces.2011.11.011.

[48] Y. Zhou, H. Jin, B. Wang, Modeling and mechanical influence of meso-scale concrete considering actual aggregate shapes, Construction and Building Materialsdoi:10.1016/j.conbuildmat.2019.116785.

[49] K. Dong, C. Wang, A. Yu, Voronoi analysis of the packings of non-spherical particles, Chemical Engineering Science 153 (2016) 330–343. doi:10.1016/j.ces.2016.07.013.

[50] F. M. Schallerab, S. C. Kapferac, M. E. Evansa, M. J. Hoffmanna, T. Asted, M. Saadatfare, K. Meckea, G. W. Delaney, G. E. Schröder-Turka, Set Voronoi diagrams of 3D assemblies of aspherical particles, Philosophical Magazine 93 (31-33) (2013) 3993–4017. doi:10.1080/14786435.2013.834389.

[51] G. Mollon, J. Zhao, 3D generation of realistic granular samples based on random fields theory and Fourier shape descriptors, Computer Methods in Applied Mechanics and Engineering 279 (2014) 46–65. doi:10.1016/j.cma.2014.06.022.

[52] E. J. Garboczi, Three-dimensional mathematical analysis of particle shape using X-ray tomography and spherical harmonics: Application to aggregates used in concrete, Cement and Concrete Research 32 (10) (2002) 1621–1638. doi:10.1016/S0008-8846(02)00836-0.
URL http://www.sciencedirect.com/science/article/pii/S0008884602008360

[53] B. Zhou Ã, J. Wang, H. Wang, Three-dimensional sphericity, roundness and fractal dimension of sand particles, Geotechnique 68 (12) (2018) 1116–1123. doi:10.1680/jgeot.16.P.207.

[54] N. Chernov, Y. Stoyan, T. Romanova, Mathematical model and efficient algorithms for object packing problem, Computational Geometry: Theory and Applications 43 (5) (2010) 535–553. doi:10.1016/j.comgeo.2009.12.003.
URL http://dx.doi.org/10.1016/j.comgeo.2009.12.003

[55] A. Bertei, C.-C. C. Chueh, J. G. Pharoah, C. Nicolella, Modified collective rearrangement sphere-assembly algorithm for random packings of nonspherical particles: Towards engineering applications, Powder Technology 253 (Supplement C) (2014) 311–324. doi:10.1016/j.powtec.2013.11.034.
URL http://www.sciencedirect.com/science/article/pii/S0032591013007468http://dx.doi.org/10.1016/j.powtec.2013.11.034

[56] G. Liu, K. E. Thompson, Influence of computational domain boundaries on internal structure in low-porosity sphere packings, Powder Technology 113 (1) (2000) 185 – 196. doi:https://doi.org/10.1016/S0032-5910(00)00255-2.
URL http://www.sciencedirect.com/science/article/pii/S0032591000002552

[57] R. Zhou, Z. Song, Y. Lu, 3D mesoscale finite element modelling of concrete, Computers and Structures 192 (2017) 96–113. doi:10.1016/j.compstruc.2017.07.009.
URL http://dx.doi.org/10.1016/j.compstruc.2017.07.009

[58] T. Byholm, M. Toivakka, J. Westerholm, Effective packing of 3-dimensional voxel-based arbitrarily shaped particles, Powder Technology 196 (2) (2009) 139–146. doi:10.1016/j.powtec.2009.07.013.

[59] S. Thomas, Y. Lu, E. J. Garboczi, Improved Model for Three-Dimensional Virtual Concrete: Anm Model, Journal of Computing in Civil Engineering 30 (2) (2015) 04015027. doi:10.1061/(asce)cp.1943-5487.0000494.

[60] C. W. Boon, G. T. Houlsby, S. Utili, A new contact detection algorithm for three-dimensional non-spherical particles, Powder Technology 248 (2013) 94–102. doi:10.1016/j.powtec.2012.12.040.

[61] E. J. Garboczi, J. W. Bullard, Contact function, uniform-thickness shell volume, and convexity measure for 3D star-shaped random particles, Powder Technology 237 (2013) 191–201. doi:http://dx.doi.org/10.1016/j.powtec.2013.01.019.
URL http://www.sciencedirect.com/science/article/pii/S0032591013000363

[62] C. R. Smith, A characterization of star shaped sets, The American Mathematical Monthly 75 (10) (1968) 1150–1178. doi:10.1080/00029890.1968.11971141.

[63] ASTM D4791-19 - Standard Test Method for Flat Particles, Elongated Particles, or Flat and Elongated Particles in Coarse Aggregate, Standard, ASTM International (2019).

[64] Z. Qian, E. J. Garboczi, G. Ye, E. Schlangen, Anm: a geometrical model for the composite structure of mortar and concrete using real-shape particles, Materials and Structures/Materiaux et Constructions 49 (1-2) (2014) 149–158. doi:10.1617/s11527-014-0482-5.

[65] E. J. Garboczi, J. W. Bullard, 3D analytical mathematical models of random star-shape particles via a combination of X-ray computed microtomography and spherical harmonic analysis, Advanced Powder Technology 28 (2) (2017) 325–339. doi:10.1016/j.apt.2016.10.014.
URL http://dx.doi.org/10.1016/j.apt.2016.10.014

[66] S. Mindess, F. Young, D. Darwin, Concrete 2nd Edition, 2003.

[67] H. Wadell, Volume, shape, and roundness of rock particles, The Journal of Geology 40 (5) (1932) 443–451. arXiv:https://doi.org/10.1086/623964, doi:10.1086/623964.
URL https://doi.org/10.1086/623964