XIV International Conference on Computational Plasticity. Fundamentals and Applications
COMPLAS XIV
E. Oñate, D.R.J. Owen, D. Peric & M. Chiumenti (Eds)

IMPROVED APPROXIMATION OF ARBITRARY SHAPES IN DEM SIMULATIONS WITH MULTI-SPHERES

FABIAN WESTBRINK* AND ANDREAS SCHWUNG†

* South Westphalia University of Applied Sciences
Lübecker Ring 2, 59494 Soest, Germany
e-mail: westbrink.fabian@fh-swf.de, web page: http://www.fh-swf.de/cms/fat

†South Westphalia University of Applied Sciences
Lübecker Ring 2, 59494 Soest, Germany
e-mail: schwung.andreas@fh-swf.de, web page: http://www.fh-swf.de/cms/fat

Key words: Discrete Element Method (DEM), Multi-sphere, Clustering, Fuzzy C-means, Clump

Abstract. DEM simulations are originally made for spherical particles only. But most of real particles are anything but not spherical. Due to this problem, the multi-sphere method was invented. It provides the possibility to clump several spheres together to create complex shape structures. The proposed algorithm offers a novel method to create multi-sphere clumps for the given arbitrary shapes. Especially the use of modern clustering algorithms, from the field of computational intelligence, achieve satisfactory results. The clustering is embedded into an optimisation algorithm which uses a pre-defined criterion. A mostly unaided algorithm with only a few input and hyperparameters is able to approximate arbitrary shapes.

1 INTRODUCTION

Modern Discrete Element Method (DEM) simulation tools provide the use of multi-spheres instead of only spherical particles to approximate the shape of real material more precisely. The use of multi-spheres is a method of clumping several spheres together. The total number of spheres, positions, radii and overlapping between spheres in a clump can be adjusted to obtain a broad range of arbitrary shapes. Some modern simulation tools offer inherent functions to create custom clumps or present a range of several different templates. Also, the use of super-quadrics instead of multi-spheres is another way to obtain non-spherical particles which are available in some DEM tools. To be not restricted by vendors specifications, e.g. LIGGGHTS [1] offers the possibility to include user-defined clumps. These clumps are saved in files which only consists of the local Cartesian coordinates and the radius of each sphere of the multi-sphere clump. A creation of these files by approximate desired shapes and transfer them into sphere constellations needs an appropriate algorithm. This algorithm requires several different steps to create proper

approximations. It begins with analysing the given shape, initial filling of these shape with random spheres, efficient clustering and observing the obtained result. Generally, there are three objectives which have to be fulfilled within the approximation. To save computation time in DEM simulations, a sufficient approximation of desired shapes by using a minimum number of spheres per clump is required. Secondly, the number of spheres must be large enough to achieve an adequate surface representation. And finally, the applicability of this algorithm for expected shapes and an acceptable algorithm cycle time is to be ensured. A user-friendly functioning along with a minimum number of possible adjustments and mostly automatic algorithm is desired. The complete algorithm is currently based on a MATLAB script, but generally translatable to other programming languages.

2 MULTI-SPHERE APPROXIMATION

Often real shapes are given as 3D surface representation by using laser scan technology or stereo photogrammetry. The stereolithography (STL) file format is the standard presentation form in this context and represents the shape as complete mesh consisting of a distinct number of triangles. Also, standard CAD tools are usually able to translate drawings into this file format. Due to possible complex shape structures, appropriate algorithms are required to define and place single spheres converging to a multi-sphere clump. In general, there a three approaches to fill 3D shapes. Phillips et al. explain that in a two-dimensional space, by using either mono-disperse or poly-disperse disks, an approximation of shapes is possible by packing, covering or filling [2]. These approaches can also be converted into the three-dimensional space. Packing does not allow an overlapping of spheres and protruding the surface of the shape, covering allows overlapping as well as protruding to cover the complete surface and filling only allows overlapping of spheres but no protruding of the surface edges. Due to the fact that overlapping is not possible, packing usually requires a much higher amount of spheres to reach the desired approximation instead of using overlapping spheres. Only covering and filling provide the performance for modern approximation algorithms.
The general approach to approximate shapes and generate spheres starts with placing distinct or randomly chosen spheres within the shape's volume. Afterwards, an optimisation problem defines different criteria which are used to cluster and reduce afterwards the total number of spheres. Some algorithms have to consider the inertia and centre of gravity during the creation of the resulting clumps. Deviations in these properties from the original body and clump can significantly change the clump behaviour in the DEM simulation itself. The DEM tool LIGGGHTS offers, therefore, the opportunity to set these properties independently from the clump shape. This reduces the effort enormously during the approximation. Most attention is thus on the exact surface and volume representation.

2.1 Related work

A current clump generation tool developed by Price et al. is made with randomised sphere positions to fill initially thousands of spheres into a shape. The spheres are placed

assuming that it needs exactly four random points of the shape, nodes of the mesh, to form a distinct sphere with the centre position and radius. This filling yields finally to the issue, that most of the spheres are deleted afterwards because their centre is often outside the shape. So the post processing starts first with deleting of unnecessary spheres and then starts clustering the spheres [3]. The algorithm by [4], uses only three different parameters, the minimum distance between surface nodes and spheres, the minimum radii of spheres, the percentage number of nodes which are covered and uses thus no need of post processed clustering at all. Clustering methods in recent works are often based on heuristic algorithms, [5, 6] present their clustering as flow chart algorithms which use individual conditions to cluster the spheres to appropriate clumps. One way of validating the results is the off-volume criterion. This criterion, which is also used by [7], compares the original shape volume with the approximated volume of the final clump representation.
The algorithms often need a relatively large number of spheres per clump for proper approximation results. It depends on the one hand on the optimisation indicators or the possible hyperparameters and on the other hand on the clustering methodologies how much spheres per clump are computed. Particularly the clustering methods offer space for optimisation. But also the prior steps, filling and observing of the given mesh, up to the quality analysis have the potential for improvements. A complete algorithm for arbitrary shapes, which uses only a minimum of spheres for the approximation is not developed yet.

2.2 Problem Statement

The presented algorithms offer different ways to approximate shapes into multi-sphere clumps for DEM simulations. Creation of multi-sphere clumps is always a compromise between accuracy of approximation and the total number of spheres per clump. With increasing approximation quality the number of spheres increases as well. The quality of an approximation can be assumed as a grade of similarity to the original shape. With an infinite number of spheres an identical approximation can be achieved. With fewer spheres, a good approximation is performed when the dimensions of the shape are completely filled and the roughness of the obtaining sphere constellation is relatively smooth. How much spheres are needed for a proper approximation always depends on the complexity of the desired shape and the allowed multi-sphere method, overlapping or non-overlapping method. To reduce the computation time of DEM simulations there is basically the idea of minimising the number of spheres in complete DEM simulation. Increasing number of spheres always extend the total computation time. Due to the simulation purposes, the desired approximation quality varies significantly. A general rule is that with a rising number of clumps within one simulation run in the DEM simulation, the quality of the approximation, respectively number of sphere per clumps is negligible reduced. But otherwise, when the degree of approximation is of particular importance, the shape approximation requires more spheres per clump, hence the total number of clumps in DEM is there relatively small.
Summarised there are four requirements defined which have to be fulfilled by the improved approximation algorithm:

- Minimum number of spheres per clump to achieve short DEM simulation time
- Approximation for arbitrary shapes with overlapping multi-spheres with sufficient approximation quality
- Minimised number of possible input parameters - highly autonomous approximation

3 FILLING

The improved algorithm presented in this paper is based on the filling approach. The spheres of the multi-sphere clump may protrude the shapes surface or other spheres. These overlaps to other spheres increase, compared to non-overlapping spheres, the accuracy and smoothness of the surface representation as well as decreases the total number of spheres by the same roughness.The initial filling is made generally with random spheres. There it is also ensured that all of these sphere's centres are within the surface of the shape. Compared to the algorithm of [3], it is not necessary to delete unfitting spheres. Each sphere with its centre and radius covers or is in contact with at least one node. When a prior defined number of nodes is covered with spheres, is the filling process ended. Depending on the quality of the mesh and the total number of nodes in it, the number of the filled spheres may very significantly. This node based filling ensures compared to a distinct number of initial spheres that the mesh is appropriately filled. A variation of the percentage nodes which are covered is adjustable to reach optimal filling with a sufficient number of spheres by using arbitrary shapes. A restriction of too close centres is not required because of the clustering afterwards and the different radii of each sphere. The radii of all initial spheres, where each radius is defined as the minimum distance to next available node of the shapes mesh, are computed. This distance is defined as the euclidean distance in the three-dimensional space, for each centre to its closest node, which is defined as:

$$d(n, c) = \sum_{i=1}^{3} \sqrt{(n_i - c_i)^2}. \tag{1}$$

To find the minimum distance and thus the closest node, each sphere compares the distances to all nodes iteratively. These smallest distances are now set as radii. Figure 1 shows a randomly set sphere inside a STL mesh, which radius is adapted to the distance D to the closest node. This sphere covers not only the contacted node but surrounding nodes as well. Depending on the shapes structure and the meshing quality it makes sense to increase the radius with a particular offset to ensure that most of the nodes are covered. When the desired number of nodes is covered with spheres, the filling process usually ends with up to several thousand spheres.

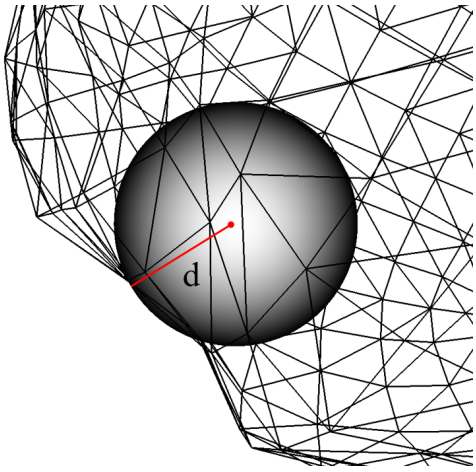


Figure 1: Define radius of initial sphere

The complete filling of the given shape by using this large number of spheres close to optimal. Figure 2a shows the original shape of a common piece of stone. This shape consists of close to 400 nodes, and thus moderate accuracy or quality. The few thousands of spheres in 2b are in the next step clustered and the total number reduced by using an optimisation algorithm and the off-volume criterion. To run the optimisation algorithm and thus the clustering properly, it is necessary to calculate or estimate the resulting volume of the clump in each optimisation step.

(a) Original shape

(b) Randomly filled clump

Figure 2: Initial filling

4 OPTIMISATION

Considering only two circles which are overlapping, calculating the entire volume of this resulting shape is trivially done by adding the volume of both circles and subtract once the intersection sector. This methodology can be also transferred into the three-dimensional space by using spheres instead of circles. But this analytical calculation is only possible with the intersection of maximum two spheres. When n spheres overlap together in one intersection, only a numerical volume estimation is possible . A recent methodology is the use of a Monte Carlo estimation by using random numbers. Another possibility is a discretized scan of the clump area into sufficiently small pieces with a

regular grid, which eliminates the uncertain factor of random numbers but needs more computation time. Depending on the dimensions of the shape, it is ensured that the grid is precise enough to compute the resulting volume sufficiently accurate.
One option to define the quality of an approximation is to observe the accuracy of the volume representation. The volume criterion, in general, does not give any information about the clump's roughness or surface quality. But when the resulting volume of multiple spheres reaches as close as possible the original volume, particularly when the dimensions of the spheres protrude the original mesh, the surface quality is also sufficient. The volume criterion is following defined as the ratio between the calculated off-volume of the resulting clump and the original volume of the input shape,

$$C_{vol}(n) = \frac{V_{off}(n)}{V_{Original}}, \tag{2}$$

where the off-volume changes with the total number of spheres n in the clump. This volume criterion or off-volume ratio is used as optimisation hyperparameter. The optimum is found when the ratio between off-volume and original volume is exactly one. It can be assumed that, as closer this ratio becomes to one, the quality of the approximation rises. The original volume, if unless not already available, is either read out with appropriate CAD software tools or computed with a convex hull of the mesh in the MATLAB script itself. Depending on the allowed protruding and on the complexity of the given shape, an off-volume ratio between $0.85 - 1$ yields to sufficient approximations.
The optimisation reduces the number of spheres by clustering them. As already said the volume criterion is used as hyperparameter for this optimisation. When the resulting clump grows up to the prior defined off-volume ratio, the clustering stops and outputs the final clump. Hence, this optimisation starts with a low number of clusters and increases them iteratively.

5 CLUSTERING

After the initial filling process, the shape approximation is made with several thousand spheres. Often the spheres centres are very close to each other, create dense areas, or the spheres overlap for the most part. In these cases, an exchange of these spheres against one single sphere, which represents a similar extension, is an advantage. These grouping of spheres, or data sets, is called clustering.
In the field of computational intelligence or machine learning, clustering belongs to the subfield of unsupervised learning. Instead of supervised learning, unsupervised learning considers only the input data without any consideration of the output information. Clustering always uses input data sets with n dimensions as vectors $x_1, ..., x_n$. In this application, the centre x_c, y_c, z_c and the radius r of each sphere are used as the input vector. The resulting clusters then also consist of a centroid with an appropriate radius. The most common and simplest clustering algorithm is the k-means clustering. As firstly explained by [8], this algorithm uses k numbers of clusters. To start the algorithm the total amount of k clusters and its centroids are priorly defined in the data set space. With every optimisation step, the input sets are assigned to nearest centroid by calculat-

ing the euclidean distance between them. After the assignment is done for all sets, the centroids position is updated according to the arithmetic mean of all assigned sets. This cost function is defined in the following equation:

$$J = \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2, \tag{3}$$

where J is the objective function which should be minimised with the sum of squares from each cluster k with its mean μ_k to every data point of S_i [8]. Due to this, it is clear that this clustering needs a prior defined number of different clusters. Because the clusters also define the number and positions of the spheres in the clump afterwards, a prediction of the number of spheres which are necessary for an arbitrary shape needs to be performed. But this prediction, especially for very complex structures, cannot be done easily. With different numbers of spheres, the expected volume and surface roughens changes. So, an overall optimisation, which increases the number of spheres iteratively and checks at each iteration the volume criterion is assigned to it. A major disadvantage of k-means clustering to obtain spheres properties for multi-sphere approximation is that the resulting clusters do not overlap at all. Each input data set is only assigned to one distinct cluster. And non-overlapping clump requires significantly more spheres to reach the same volume instead of overlapping clumps which is not feasible with k-means clustering. Therefore, for an optimal approximation, another clustering method is recommended.

5.1 Fuzzy clustering

Fuzzy clustering or fuzzy c-means clustering is an enhanced version of the conventional k-means clustering, developed first by [9]. This clustering algorithm extends the hard clustering k-means by a soft, fuzzy, attribute. Instead of a distinct assignment of each input data set to only one cluster, fuzzy c-means clustering allows an assignment to several clusters at the same time. How much percent one input set belongs to different clusters is defined as membership degree. Generally, the algorithm works similar to k-means, with defining a number of clusters, in this case, c, place them initially and optimise the mean of assigned sets, but with the enhancement of degree of membership and results following as

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} \mu_{ij}^{m} ||x_i - c_j||^2, \tag{4}$$

where for each cluster, the membership degree is defined as u_{ik} between 0 and 1, but the sum of all membership degrees is always 1 [9]. The partition matrix m can be set between 1 and ∞ and defines the degree of memberships. When m equals to 1, it represents hard clustering and the results are identical to k-means, if it is close to infinity the membership degree for all sets becomes equal. An adjustment of the m parameter defines the grade of membership or overlapping [9]. The grade of overlapping can be used afterwards to define how close the different centroids are placed and how much spheres per clump fit to the

volume criterion. It has been shown that values of m between 1.5 to 3.0 yield an optimal approximation. Compared to k-means clustering, fuzzy c-means clustering requires more computation time, especially when the partition matrix is also iteratively optimised.

6 APPROXIMATION PROCESS

The described improved approximation algorithm of arbitrary shapes with the overlapping multi-sphere method can be analysed in four steps. The first step analyses the desired shape, observes its nodes and defines its volume. The second step fills the shape's dimension with random spheres. These spheres are defined to cover almost one node of the shape's mesh and the radii are equal to the distance to the closest node. The filling stops when a distinct percentage of nodes is covered by spheres. Thirdly runs the optimisation with fuzzy clustering which reduces the total number of spheres rapidly. The fuzzy c-means clustering is implemented, due to the fact that overlapping of spheres can be represented by a partition matrix which fits perfectly for the creation of multi-spheres. By changing the value of the membership degree the grade of overlapping can be set. The partition matrix is changed iteratively to find a sufficient overlapping. When the desired volume criterion is reached, the optimisation stops automatically. The complete algorithm is shown as pseudo code in algorithm 1. There it is also shown that the resulting clump, with all necessary information, is stored afterwards in a file which fits the requirements of the LIGGGHTS DEM simulation tool.

Algorithm 1 Shape Approximation

Input Shape as STL file
Create Bounding Box
$V_{Original} = Volume(STL)$
while $NodesCovered <= NodesPercentage$ **do**
    $i = 0$
    $Sphere_{Centre}(i) = Random(BoundingBox)$ ▷ Centre in Bounding Box
    if $inhull = (Sphere_{Centre}(i), STL)$ **then** ▷ Centre inside STL mesh
        $Sphere_{Radius}(i) = EuclideanDistance(Sphere_{Centre}(i), STL)$
    $NodesCovered = Cover(Sphere_{Centre}(i), Sphere_{Radius}(i), STL)$
    i++
$clusters = 1$
while $Volume_{Criterion} >= V_{Off}/V_{Original}$ **do** ▷ Optimisation with volume criterion
    for $m = 1.5$ to $m = 3.0$ **do**
        $Cluster(Centre, Radius) = Fuzzy(Sphere_{Centre}, Sphere_{Radius}, m, clusters)$
        $V_off = Volume(Cluster)$
    $clusters++$
$WriteMultiSphere(Clusters)$ ▷ Create Multi-Sphere file

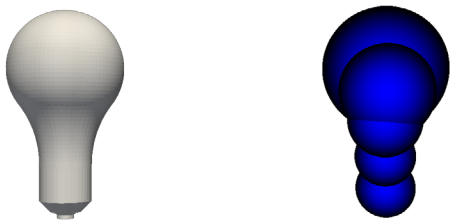
7 RESULTS

The approximation algorithm is generally made for every arbitrary shape, which is available as STL files. Shapes with other file formats are previously to export in binary or ASCII STL format. It starts with a random placement of spheres inside the mesh until the particular percentage of all node is covered with spheres. The given shapes as STL mesh can be arbitrary, but for a huge number of nodes in the mesh, the computational time significantly increases. Therefore, in some cases, the mesh has to be simplified and the number of nodes to be reduced.

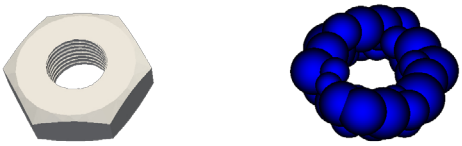
It is shown in figure 3 that different arbitrary shapes can be approximated appropriately by using a different number of overlapping spheres. Especially the approximations in figures 3b and 3c, with very complex meshes, are well approximated with only a few spheres. A prior adjustment of the approximation can be done by defining, on the one hand, the percentage of nodes which are covered with spheres in the initial phase and on the other hand the value of the off-volume criterion. Also, it is possible to let the spheres protrude out of the shape to decrease the surface roughness. The approximation in figure 3 is made with 90% covered nodes of the original mesh and an off-volume criterion of 90%. Be defining only these three parameters it is possible to get sufficient approximations for nearly each shape by considering the inherent inaccuracies of the multi-sphere method itself. The number of resulting sphere per clump is an appropriate range to simulate multi-spheres in DEM simulation shortly, also with a large number of clumps. Figure 4 shows how the number of spheres per clump varies with the different off-volume criteria.

A shape approximation of < 0.7 off-volume ratio often gives poor results and is useless for appropriate DEM simulations. The quality increases noticeably in the range between $0.7 - 0.9$ and the number of spheres is an adequate range. In the range of $0.95 - 1$, the number of spheres is significantly high and the approximation benefits are hardly present. Furthermore, the computation time of the approximation algorithm itself and later the DEM simulation becomes comparatively too high. After satisfactory results from the clustering algorithm, the spheres properties are stored in an input file for the DEM simulation. The clustering algorithm creates the Cartesian coordinates as the position of the spheres in the clump. Finally, the information about the centre coordinates with their radii is stored in a file which is readable in LIGGGHTS. As specified in section 2, information about the centre of mass or inertia tensor could be additional set in the LIGGGHTS input script. LIGGGHTS changes internally the density of each sphere in the clump to fit these mechanical properties.

To employ the algorithm for other DEM tools is not a tedious task. The multi-sphere file is modifiable for all necessary data types and thus usable for other DEM tools.
The computation time of the approximation depends on several criteria. Firstly, the quality of the shape, the complexity and the accuracy or resolution of the mesh influences the computation time. Also, the desired off-volume ration, which is mostly responsible for the number of spheres in the clump, is a factor for computation time. For arbitrary shapes, the default parameter values create sufficient results, but for well-known shapes,



(a) Light bulb; $n = 12 spheres$



(b) Nut; $n = 36 spheres$

(c) Stone; $n = 37 spheres$

Figure 3: Shape approximations

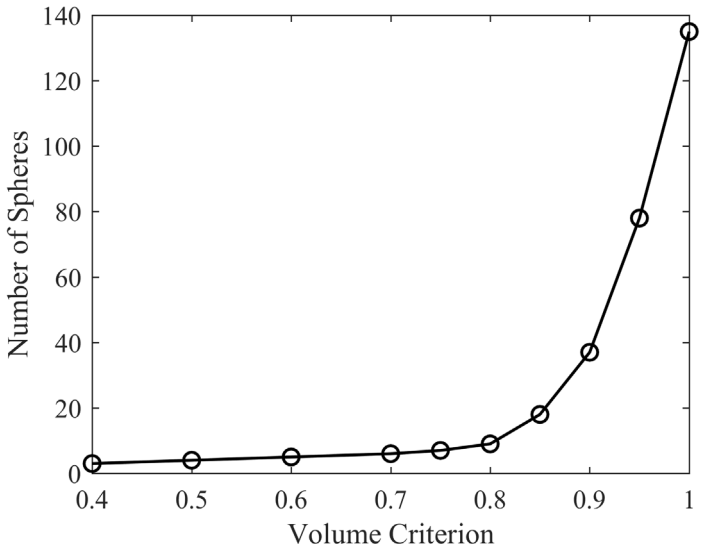


Figure 4: Changing number of spheres with volume criterion

the parameters can be improved to obtain faster computation time and better results.

8 CONCLUSIONS

The presented algorithm offers the ability to approximate arbitrary shapes with a low number of spheres. An optimisation with the volume criteria shows good results not only for the complete filling of the desired shape but also a smooth surface roughness. The upcoming need of arbitrary shapes in DEM simulation reasons rising effort of improved approximation algorithms. Especially the use of fuzzy c-means clustering performs inherent good results for overlapping multi-sphere representations.

Further improvements shall translate the MATLAB script into a LIGGGHTS in-build function. This would offer the possibility to call the desired shape as STL mesh within the LIGGGHTS input script and an approximation runs inside the DEM simulation. Also, other clustering algorithms like density bases clustering are under observation. Especially improvements in the roughness quality of the approximation and reducing of the computation time of the algorithm are foreseen.

REFERENCES

[1] C. Kloss, C. Goniva, A. Hager, S. Amberger, and S. Pirker, "Models, algorithms and validation for opensource dem and cfd-dem," *Progress in Computational Fluid Dynamics, An International Journal*, vol. 12, no. 2/3, p. 140, 2012.

[2] C. L. Phillips, J. A. Anderson, G. Huber, and S. C. Glotzer, "Optimal filling of shapes," *Physical review letters*, vol. 108, no. 19, p. 198304, 2012.

[3] M. Price, V. Murariu, and G. Morrison, "Sphere clump generation and trajectory comparison for real particles," *Proceedings of Discrete Element Modelling 2007*, 2007.

[4] J.-F. Ferellec and G. R. McDowell, "A method to model realistic particle shape and inertia in dem," *Granular Matter*, vol. 12, no. 5, pp. 459–467, 2010.

[5] X. Garcia, J.-P. Latham, J. Xiang, and J. P. Harrison, "A clustered overlapping sphere algorithm to represent real particles in discrete element modelling," *Géotechnique*, vol. 59, no. 9, pp. 779–784, 2009.

[6] C.-Q. Li, W.-J. Xu, and Q.-S. Meng, "Multi-sphere approximation of real particles for dem simulation based on a modified greedy heuristic algorithm," *Powder Technology*, vol. 286, pp. 478–487, 2015.

[7] S. Amberger, M. Friedl, C. Gonvia, S. Pirker, and C. Kloss, "Approximation of objects by spheres for multisphere simulations in dem," *ECCOMAS*, 2012.

[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. University of California Press, 1967, pp. 281–297.

[9] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, ser. Advanced Applications in Pattern Recognition. Boston, MA: Springer, 1981.