Original software publication

CLUMP: A Code Library to generate Universal Multi-sphere Particles

Vasileios Angelidakis a,*, Sadegh Nadimi a,*, Masahide Otsubo b, Stefano Utili a

a School of Engineering, Newcastle University, UK
b Institute of Industrial Science, The University of Tokyo, Japan

A R T I C L E I N F O

Article history:
Received 25 February 2021
Received in revised form 7 June 2021
Accepted 7 June 2021

Keywords:
Clumps of spheres
Clusters
Irregular particles
Discrete Element Method
Particle morphology

A B S T R A C T

Particle shape plays a key role in the mechanical and rheological behaviour of particulate and granular materials. The simulation of particulate assemblies typically entails the use of Molecular Dynamics, where spheres are the predominant particle shape, and the Discrete Element Method (DEM). Clumps and clusters of spheres have been used to simulate non-spherical particles, primarily due to the simplicity of contact detection among spheres and their ability to approximate practically any irregular geometry. Various approaches have been proposed in the literature to generate such clumps or clusters, while open-source numerical codes applying these are scanty. The CLUMP code, proposed in this paper, provides a unified framework, where a particle morphology can be approximated using different clump-generation approaches from the literature. This framework allows comparing the representations of the particle generated by the different approaches both quantitatively and qualitatively, providing the user with the tools to decide which approach is more appropriate for their application. Also, one novel generation technique is proposed. Outputs are provided in formats used by some of the most popular DEM codes. Moreover, the resulting clumps can be transformed into surface meshes, allowing for easy characterisation of their morphology. Finally, the effect of clump-generation techniques on the mechanical behaviour of granular assemblies is investigated via triaxial compression tests.

© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Motivation and significance

Particle morphology is believed to be the controlling factor in spreading of powder bed in additive manufacturing [1], wave velocity anisotropy [2], railway track sub-structure design [3] and pharmaceutical tableting [4], just to name a few. Modelling real particles as clumps of overlapping spheres or clusters of non-overlapping spheres is common practice in both academic and industrial studies, when particle morphology is of interest. Clustering spheres to represent a single particle offers versatility within a simulation, as the fine details of particle morphology such as local roundness, roughness or concavities can be introduced. Moreover, the computational cost is affordable due to the simplicity of performing contact detection among spheres, given the total number of spheres considered. An abundance of methods has been proposed in the literature to perform such
Fig. 1. The main modules of CLUMP. The surface of a rice grain and a sand grain are used to generate clumps of overlapping spheres. The surface of each clump can be extracted, allowing for a quantitative characterisation of their morphology.

![Diagram of CLUMP](image)

**Fig. 2.** Main objects of the code structure (a) clump (b) mesh.

2. Software description

The code comprises three main modules: (1) Functions to generate clumps, (2) Functions to export clumps and (3) Transformation of clumps to surface meshes, allowing for their morphological characterisation and comparison with the original particle. The first module is the main core of the software, while the second and third ones have an auxiliary role, functioning as utility scripts.

2.1. Software architecture

This section details the main structure of the code, which is also visualised in Fig. 1.

Module I: Generate clump. CLUMP supports three approaches to generate clumps, namely Favier et al. [5], the first multi-sphere approach proposed in the literature; Ferrellec and McDowell [11], a widely used approach; and a newly proposed approach based on the Euclidean distance transform of 3D images. Fig. 2 shows the main objects of the clump generation module.

Module II: Export clump. The generated clumps represented by the centroids and radii of multi-spheres can be exported in various formats (*.py, *.csv), compatible with some of the most mainstream commercial and open-source DEM codes, including EDEM, PFC3D, YADE, and LAMMPS.

Module III: Characterise clump. A surface extraction module is developed, transforming each clump into a surface mesh, which in turn can be imported in SHAPE [25], to perform shape characterisation.

CLUMP relies on several external functions available within the Matlab FEX community. In particular, the outsourced tasks include the handling of stereolithography files [26], mesh manipulations [27], and calculation of rigid body parameters [28].

2.2. Software functionalities

CLUMP supports the following clump-generation approaches:

2.2.1. Favier et al. [5]

This method only applies to symmetrical particles which are constructed of spheres whose centres are located on the particle axis of symmetry. Spheres may overlap and may vary in diameter along the length of the axis of symmetry. The surface of a particle is approximated by inscribing spheres such that the surface of each sphere is tangent to the surface of the particle at the point of contact. The position of each element sphere is fixed relative to the other elements within a particle.
2.2.2. Ferellec and McDowell [11]

The clump generation methodology of Ferellec and McDowell [11] selects random points of the mesh of the particle surface and generates tangent spheres of increasing radii, until they intersect a different vertex of the particle surface. This methodology was designed to eliminate artificial roughness of flat faces, through the consideration of large numbers of spheres per particle. Due to this, the approach of Ferellec and McDowell will not work as well if we want to generate 2 or 3 spheres per clump, as the selection of the initial points is random. This happens because the methodology lacks a deterministic criterion of which vertex to choose every time to generate spheres. In our implementation of this method, a seed parameter is introduced in the selection of vertices to achieve reproducible clumps. Though, this seed is not guaranteed to lead in any way to the best-possible clump, for the given number of spheres.

In more detail, the volume of the real particle is filled optimally with spheres of different sizes: from a point chosen randomly on the surface, a sphere is grown internally along the normal vector at that point to the maximum extent possible inside the boundary of the particle. In other words, the expansion of the sphere continues until it reaches another point on the surface of the particle. Then, the process is repeated for other points on the surface of the particle, which must be farther by a distance $d_{\text{min}}$ to any existing sphere. The number of spheres inside the clump is directly related to the number of points on the surface, by monitoring a percentage $p_{\text{max}}$, which if met, terminates the generation process. If the density of points on the surface of the particle is high enough, then the process can theoretically lead to a clump composed of thousands of spheres, although Ferellec and McDowell [11] set criteria that can reduce local crowding of spheres, using a minimal radius $r_{\text{min}}$, or the aforementioned $d_{\text{min}}$ and $p_{\text{max}}$ variables. If a large number of spheres per clump is computationally acceptable, this approach can lead to smaller artificial roughness of flat faces, compared to other methods, as Ferellec and McDowell report [11]. This approach was designed to be applied on surface meshes, but is also applicable directly on point clouds, since techniques do exist to estimate the normal vector of each member point of a cloud.

2.2.3. Euclidean distance transform

A new approach is presented in this paper to generate multi-sphere particles, based on the concept of the Euclidean distance transform of 3D images. Initially, the particle morphology is transformed from a surface mesh to a voxelated representation of a minimum dimension div, where the value of all voxels belonging to the particle is set equal to zero. Then, the inscribed sphere of the particle is found as the maximum value of the Euclidean distance transform of the voxelated image. The voxels corresponding to the inscribed sphere are then set equal to one and the Euclidean distance map is calculated for the new voxelated image. This process is repeated until a user-defined number of spheres is found ($N$) or until the user-defined minimum radius ($r_{\text{min}}$) has been reached, as the spheres are generated in decreasing sizes. This methodology can also generate overlapping spheres, if only a percentage of the voxels making each new sphere is set equal to one, instead of all of them, expressed by a variable overlap, which takes values within the range $[0, 1]$.

Fig. 3 shows the algorithms behind the Ferellec and McDowell [11] and Euclidean distance transform implementations. The
output of the above approaches is a set of spheres, represented by their centroids and radii, as needed in commercial and open-source codes. More information on these approaches is provided as supplementary material.

2.2.4. Particle surface extraction

One of the main objectives of CLUMP is to allow for a direct comparison among the morphologies of multi-sphere particles generated using different approaches. To achieve a quantitative comparison, particle shape-characterisation tools can be used, such as SHAPE [25]. To this end, the surface of each clump has to be extracted in the form of a surface mesh, allowing for compatibility between the clump generation and the shape characterisation codes.

Extracting the surface of a multi-sphere particle is not a trivial task. The problem can be tackled using many different approaches. In this implementation, the surface of each clump is extracted following the steps below:

1. Contact detection is performed between all possible pairs of the spheres making the clump, identifying the pairs with geometric overlap.
2. A point cloud is generated on the surface of each sphere.
3. For each pair of overlapping spheres, the points within the overlap region are deleted (Fig. 4).
4. A circle is found as the intersection of each pair of overlapping spheres and points are generated along each circle, which are added to the point cloud.
5. The point cloud is then tessellated using a surface reconstruction technique, making the surface of the clump.

It should be noted that generating points on the surface of each sphere introduces mesh-dependency in the calculation of local shape parameters, like roundness and roughness. Therefore a dense mesh should be chosen when these shape aspects are of interest. The local morphology can also be affected by the technique chosen to perform surface reconstruction, which in this case is an implementation of the Crust method [29] provided by Giaccari [30]. The points on the circles calculated as the overlap of each pair of spheres are added to the point cloud, aiming to minimise the numerical noise of the clump surface near the region of an overlap and provide a smooth transition from the surface of a sphere to its adjacent ones. Fig. 4 shows the surface of a clump made of two spheres, where the points inside the overlap region (removed from the point cloud) and on the intersection circle (added to the point cloud) can be observed.

2.3. Sample code snippets analysis

This section highlights two implementation details of CLUMP. First, Fig. 5(a) demonstrates how the newly introduced approach based on the Euclidean distance transform allows the generation of overlapping spheres, while Fig. 5(b) shows a typical loop of how a tangent sphere is grown in the methodology of Ferellec and McDowell [11].

```
% voxel related representation of particle
[dx,dy,dz]=meshgrid(1:size(vox,2),1:size(vox,1),1:size(vox,3));
for k=1:N % N: number of spheres
    editImage = bwdist(~vox); % Euclidean map
    radius = max(editImage(:)); % Inradius in voxel units
    [yCenter, xCenter, zCenter] = ind2sub(size(vox), find(editImage == radius));
    dists = sqrt(sum((dx-xCenter).^2+(dy-yCenter).^2+(dz-zCenter).^2));
    [~,i]=max(dists); % Index of the inscribed sphere
    sph=zeros(length(dy),length(dx),length(dz));
    sph=sqrt((dx-xCenter(i)).^2+(dy-yCenter(i)).^2+(dz-zCenter(i)).^2) > (1-overlap)*radius; % Sphere
    vox=and(vox,sph); % Append the new sphere in the particle
end
```

(a)

```
x=P(:,1); y=P(:,2); z=P(:,3);
for k=1:length(P) % P: vertices
    r=rmin;
    while reachedMaxRadius=false % while the sphere has not reached the surface
        while sphMin==tol
            xC=x+r*n(1); yC=y+r*n(2); zC=z+r*n(3);
            % n: normal vector of vertex k
            distance=sqrt((xC-x).^2+(yC-y).^2+(zC-z).^2);
            % Distances of all points to the center of the sphere
            sph=(distance/r).^2-1;
            % Potential function (negative for points inside the sphere)
            sphMin=min(sph); % Grow radius for next step
        end
        reachedMaxRadius=true;
    end
end
```

(b)
3. Illustrative examples

This example demonstrates the mechanical behaviour of assemblies of rice grains and soil grains under triaxial compression. The morphological characteristics of clumped spheres used in this illustrative example are presented in Table 1. Hereinafter, EU, FM and FA refer to grains represented by 25 spheres using the Euclidean transform, Ferellec and McDowell [11] and Favier et al. [5] methods, respectively. The samples consist of (1) rice grains using all three approaches, and (2) sand grains using the Euclidean transform and Ferellec and McDowell [11] approaches. The major axis length of the rice grain and silica sand was approximately 18 mm and 6 mm, respectively. The particle shear modulus and particle Poisson’s ratio were considered as 20 MPa and 0.1 for rice grains and 29.1 GPa and 0.23 for sand grains, respectively. The particle density was 1470 kg/m$^3$ for the rice
grains and 2650 kg/m$^3$ for the sand grains; these values were increased by 1000 times to save computational time.

The LAMMPS molecular dynamics code [31,32] is employed with a modified servocontrol, as used by Hanley et al. [33] and Morimoto et al. [34]. Each sample is composed of 5000 clumped grains, which are generated randomly without initial contacts with other particles, under no gravity conditions. The sample is subjected to an initial isotropic compression by moving the boundaries with a controlled velocity, to achieve an isotropic stress of 50 kPa. A simplified Hertz-Mindlin contact model with Coulomb friction is adopted. Dense, medium and loose samples are prepared using inter-particle friction coefficients ($\mu$) of 0.0001, $\approx$ 0.1 and 0.35, respectively, during the isotropic compression, giving relative density (Dr) values of 100%, 50% and 0%, following the approach of Morimoto et al. [34]. In the subsequent triaxial compression, the $\mu$ value was set to 0.35 for all the cases as a material constant. In the present simulations, no damping was used.

The cubical samples composed of rice grains in Fig. 6(a) were subjected to quasi-static axial compression in the Z-axis while keeping the lateral stresses in the X- and Y-axes at 50 kPa. The overall trend of stress–strain relationships is similar for each density level (Fig. 6(b)); however, the FA Dr=100% case underestimates the peak and residual stresses, while the FA cases overestimate the initial stiffness, i.e. the initial increase in deviator stress (Fig. 6(c)). A clear difference between FA and the other approaches is evident in the variation of mean coordination number (Fig. 6(d)). A representative sample composed of sand grains is illustrated in Fig. 7(a). Similar variations of stress–strain responses are observed between the EU and FM approaches (Fig. 7(b)), although measurable differences in the variation of void ratio are evident (Fig. 7(c)).

For the illustrative examples presented here, the EU and FM approaches gave comparable results, while the results of the FA approach differed. Referring to Table 1, this discrepancy can be attributed to the difference in the shape parameters of the generated multi-sphere particles. EU and FM provide similar shape parameters, indicating that quantifying shape parameters of the generated clumps is recommended as an effective measure to ensure the quality of modelling non-spherical grains. On the other hand, FA cannot represent particles with flatness, as the axial symmetric clumps it generates can only result in compact or elongated particle shapes, a fact that shows a distinct effect on the macro-mechanical behaviour of the simulated triaxial tests on rice. Interesting to note that apart from flatness, the rest of the studied morphological parameters between FA, and EU and FM are similar.

4. Impact

CLUMP enables engineers, researchers and scientists from a variety of disciplines and industries to examine the dependency of their application to particle shape effects, through the generation of irregular particle shapes. Recent advances in granular physics demonstrate a clear need for consideration of particle shape in numerical modelling. In engineering practice, the
clumped-sphere approach has been adopted in additive manufacturing, railway engineering, pharmaceuticals, etc. However, there is currently little evidence regarding how many spheres need to be generated. This is, mainly, due to the limited accessibility of open-source implementations of different clump-generation approaches. To this end, CLUMP offers two widely-used approaches to generate clumps and proposes a new one, aiming to provide an extendable library of available scripts. The implementations are computationally efficient, i.e. it takes few minutes to generate hundreds of spheres per grain.

CLUMP provides an option to extract the surfaces of the generated multi-sphere particles, allowing for the direct calculations of their shape descriptors, using available, open-source shape-characterisation codes. Comparing the morphological characteristics of the generated clumps to those of the original particle morphology, one can decide the number of spheres and method of application based on a quantitative and rigorous particle-generation approach.

5. Conclusions

We present CLUMP, a code library to approximate three-dimensional particle morphologies using different multi-sphere generation methods. The software allows exporting the results in formats compatible with several commercial and open-source DEM codes. CLUMP enables to assess the suitability of a clump-generation method for different applications and characterise the degree of manipulation of the original morphology in a quantitative manner.

The examples illustrated provide insight into the effect of the clump generation methods on the mechanical behaviour of two granular assemblies subjected to triaxial compression. We found that the adoption of particles with near zero flatness leads to the underestimation of the peak and residual stresses, while the adoption of larger contacting spheres leads to the overestimation of initial stiffness.

We hope code developers will find the framework provided here useful so that other existing clump-generating approaches and future ones will be implemented too.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

This work was supported by the UK Royal Society (reference number IEC/R3/183026), the Japan Society for the Promotion of Science (reference number JPSB120195701) and Engineering and Physical Sciences Research Council (EPSRC), UK grant number EP/ R001766/1 as a part of ‘Friction: The Tribology Enigma’ Programme Grant. The authors acknowledge Newcastle University for the doctoral scholarship of the first author.

### Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2021.100735.

### References

1. Nan W, Pasha M, Bonakdar T, Lopez A, Zafar U, Nadimi S, et al. Jamming during particle spreading in additive manufacturing. Powder Technol 2018;338:253–62.
2. Otsubo M, Liu J, Kawaguchi Y, Dutta TT, Kuwano R. Anisotropy of elastic wave velocity influenced by particle shape and fabric anisotropy under K0 condition. Comput Geotech 2020;128:103775.
3. Le Pen LM, Powrie W, Zervos A, Ahmed S, Aingaran S. Dependence of shape on particle size for a crushed rock railway ballast. Granul Matter 2013;15(6):849–61.
4. Abdel-Hamid S, Alshihabi F, Betz G. Investigating the effect of particle size and shape on high speed tabletting through radial die-wall pressure monitoring. Int J Pharm 2011;413(1–2):29–35.
5. Favier JF, Fard MH, Kremmer M, Raji AO. Shape representation of axi-symmetrical, non-spherical particles in discrete element simulation using multi-element model particles. Eng Comput: Int J Comput-Aided Eng Sci 2011;15(6):849–61.
6. Matsushima T, Saomoto H. Discrete element modeling for irregularly-shaped sand grains. In: Mestat, editor. Proceedings of 5th European conference on numerical methods in geotechnical engineering. Paris, France: Presses de l’Ecole Nationale des Ponts et Chaussées; 2002, p. 239–46.
7. Bradshaw G, O’Sullivan C. Adaptive medial-axis approximation for sphere-tree construction. ACM Trans Graph 2004;23(1):1–26.
[8] Price M, Muranui V, Morrison G. Sphere clump generation and trajectory comparison for real particles. In: 4th international conference on discrete element methods, Brisbane, Australia. 2007, http://www.diee.uct.ac.za/ mathew/publish/dem2007_sphereclump.pdf.

[9] Wang L, Park JY, Fu Y. Representation of real particles for DEM simulation using X-ray tomography. Constr Build Mater 2007;21(2):338–46.

[10] Garcia X, Latham J, Xiang J, Harrison J. A clustered overlapping sphere algorithm to represent real particles in discrete element modelling. Géotechnique 2009;59(9):779–84, https://doi.org/10.1680/geot.8.T.057.

[11] Ferellec JF, McDowell GR. A method to model realistic particle shape and inertia in DEM. Granul Matter 2010;12(5):459–67.

[12] Taghavi R. Automatic clump generation based on mid-surface. In: David Sainsbury, Roger Hart, Michele Nelson, editors. 2nd international FLAC/DEM symposium, Melbourne, Australia. 2011, p. 791–7, https://www.itascacg.com/events/continuum-and-distinct-element-numerical-modeling-in-geomechanics-2011.

[13] Gao R, Du X, Zeng Y, Li Y, Yan J. A new method to simulate irregular particles by discrete element method. J Rock Mech Geotech Eng 2012;4(3):276–81.

[14] Li CQ, Xu WJ, Meng QS. Multi-sphere approximation of real particles for DEM simulation based on a modified greedy heuristic algorithm. Powder Technol 2015;286:478–87.

[15] Zhong J, Hryciw RD. Roundness and sphericity of soil particles in assemblies by computational geometry. J Comput Civ Eng 2016;30(6):04016021.

[16] Haeri S. Optimisation of blade type spreaders for powder bed preparation in additive manufacturing using DEM simulations. Powder Technol 2017;321:94–104.

[17] Katagiri J. A novel way to determine number of spheres in clump-type particle–shape approximation in discrete-element modelling. Géotechnique 2019;69(7):620–6.

[18] Markauskas D, Kačianauskas R, Džiugys A, Navakas R. Investigation of adequacy of multi-sphere approximation of elliptical particles for DEM simulations. Granul Matter 2010;12(1):107–23.

[19] Tamadondar MR, Rasmussen A. The effect of carrier surface roughness on wall collision-induced detachment of micronized pharmaceutical particles. AIChE J 2020;66(1):e16771.

[20] Nan W, Ghadiri M, Wang Y. Analysis of powder rheometry of FT4: Effect of particle shape. Chem Eng Sci 2017;173:374–83.

[21] Hanley KJ, O'Sullivan C, Huang X. Particle-scale mechanics of sand crushing in compression and shearing using DEM. Soils Found 2015;55(5):1100–12.

[22] Suhr B, Six K. Simple particle shapes for DEM simulations of railway ballast: influence of shape descriptors on packing behaviour. Granul Matter 2020;22(2):1–17.

[23] Hanley KJ, O’Sullivan C, Huang X. Particle-scale mechanics of sand crushing in compression and shearing using DEM. Soils Found 2015;55(5):1100–12.

[24] Angelidakis V, Nadimi S, Utili S. Shape analyser for particle engineering (SHAPE): Seamless characterisation and simplification of particle morphology from imaging data. Comput Phys Comm 2021;265:107983, https://doi.org/10.1016/j.cpc.2021.107983.

[25] Micó. StlTools. MATLAB Central File Exchange; 2021, https://www.mathworks.com/matlabcentral/fileexchange/51200-stltools. Retrieved January 6, 2021.

[26] Fang Q. Iso2Mesh. GitHub; 2021, https://github.com/fangq/is02mesh. Retrieved January 6, 2021.

[27] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques; 1998, july. p. 415–21.

[28] Giaccari L. Surface reconstruction from scattered points cloud. MATLAB Central File Exchange; 2020, https://www.mathworks.com/matlabcentral/fileexchange/63730-surface-reconstruction-from-scattered-points-cloud. Retrieved July 26, 2020.

[29] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. J Comput Phys 1995;117(1):1–19.

[30] Nguyen TD, Plimpton SJ. Aspherical particle models for molecular dynamics simulation. Comput Phys Comm 2019;243:12–24.

[31] Hanley KJ, O’Sullivan C, Huang X. Particle-scale mechanics of sand crushing in compression and shearing using DEM. Soils Found 2015;55(5):1100–12.

[32] Hanley KJ, O’Sullivan C, Huang X. Particle-scale mechanics of sand crushing in compression and shearing using DEM. Soils Found 2015;55(5):1100–12.

[33] Wadell H. Volume, shape, and roundness of rock particles. J Geol 1932;40(5):443–51.

[34] Fogel J, Fonseca J. Quantification of the morphology of shelly carbonate sands using 3D images. Géotechnique 2018;68(3):249–61.