An Effective Energy-Conserving Contact Modelling Strategy for Spherical Harmonic Particles Represented by Surface Triangular Meshes with Automatic Simplification

Y. T. Feng*
Zienkiewicz Centre for Computational Engineering, Swansea University, UK

Abstract

An effective discrete element modelling strategy for triangular mesh represented spherical harmonic particles is proposed. It features: 1) using a golden spiral lattice on the unite sphere to generate an initial triangular mesh with any number of vertices/triangles for a star-shaped surface; 2) applying an edge contraction mesh simplification algorithm to reduce the mesh size to any desired level; and 3) adopting an energy-conserving linear normal contact model to compute the contact geometric and force features of contacting particles. In particular, the edge contraction algorithm is applicable to any triangular mesh. It is algorithmically very simple and highly effective, and can be easily incorporated into existing discrete element frameworks. Numerical experiments are conducted to demonstrate that the simplified mesh by the edge contraction can not only have a very low geometric approximation error but also achieve expected mechanical responses. Thus this mesh simplification approach can serve as an ideal pre-processing tool to optimise a large input triangular mesh in order to significantly reduce the computational cost associated with discrete element simulations without compromising the modelling accuracy.

KEYWORDS: Spherical harmonic, Mesh simplification, Discrete element method, Energy conserving contact model

1 Introduction

Real particles encountered in applications are irregular and mostly non-convex shaped. Using the discrete element method (DEM) [1] to model arbitrarily shaped particles, and non-convex in particular, has been a major challenging issue in two aspects: how to represent real particle shapes or their surfaces; and how to model contacts between these particles.

A 3D surface can be expressed either in an analytical, discrete or digital form. An analytical form can be implicit or explicit (i.e. parametric). A surface Ω is implicit if it can be expressed by the zero-set of a scalar-valued function $f : \mathbb{R}^3 \to \mathbb{R}$, i.e. $\Omega = \{\mathbf{x} \in \mathbb{R}^3 \,|\, f(\mathbf{x}) = 0\}$, where $\mathbf{x} = (x, y, z)$ are the coordinates of a point on the surface. In an explicit form the coordinate components (x, y, z) are expressed as three individual functions in terms of two parameters in a 2D parametric space. However, even an implicit form may need to be parametrised to facilitate the subsequent contact modelling between these particles.

*e-mail: y.feng@swansea.ac.uk

Very limited implicit functions can be used to represent a non-convex shape in the discrete element method, as comprehensively reviewed in [2]. Super-quadrics [3, 4, 5] have been used in DEM to represent non-spherical particles, and by choosing certain ranges of shape constants in the super-quadric expression, some non-convex shapes can be defined. However, the resulting shapes may be too special to be taken as a good representation of real particles. Similarly, NURBS (or B-spline) [6] and iso-geometric functions [7], both being parametric functions, can also be used to represent complex solids or surfaces [8, 9] in DEM, but these methods use control points to indirectly manipulate the resulting surfaces. This is not sufficiently flexible and thus limits their ability to conveniently construct arbitrary non-convex shapes.

A more powerful analytical approach is to utilise spherical harmonics to represent more realistic particles [10, 11, 12]. Spherical harmonics form a complete set of orthonormal functions on the unit sphere, and possess a coarse to fine representation hierarchy whereby using more high frequency harmonic functions, more detailed surface features can be captured. These favourable properties make spherical harmonics a natural choice as the basis functions to represent a 3D surface. Note that the spherical harmonic representation is also expressed in a 2D parametric form. Standard spherical harmonic expansions [11, 12] can only represent star-shaped surfaces, but can be extended to construct more general shapes [13]. As demonstrated in [14], spherical harmonics can be adopted to compactly represent solids or volumetric surfaces acquired by digital technologies. In general, a large number of spherical harmonic functions may often be needed to adequately describe a complex surface.

3D shapes can also be represented in discrete or digital forms. The most universal approach to acquire realistic particle shapes is to employ various 3D shape digitalisation technologies, such as laser and computerised tomography (CT) scanning [14, 15]. The acquired 3D model is processed and represented either by a discrete pixel/voxel cloud, a polygonal/triangular surface mesh or a smoothed analytical form, such as the spherical harmonic expansion. In most cases, a triangular mesh provides the most generic and versatile representation to accommodate any arbitrary surface. However, to capture finer details of a real surface, a large number of vertices/triangles are needed.

Assuming that non-convex particles have been represented either in spherical harmonics or in a triangular mesh, the next stage in the discrete element modelling of such particles is contact detection. This involves: 1) determining if two particles are in contact; 2) for particles in contact, computing the contact geometric features, and 3) evaluating contact forces between the contact pairs.

For two non-convex shaped particles, multiple contacts can occur. In this case the contact cannot be effectively established by simultaneously solving two spherical harmonic equations that represent the particles, so some discretised forms have to be adopted. In [11], a sophisticated contact search procedure is proposed in two stages: 1) check if the bounding boxes of the two particles have overlap; 2) use the overlap box of the bounding boxes to conduct a detailed local search to establish if the two particles are in real contact. This procedure relies on an equal angle discretisation of the 2D parametric space for each particle and the total computational cost is related to the total number of the discrete points involved. In [12], the second stage of the procedure is refined. The surface triangulation of each particle is firstly constructed from the equal angle discretisation grid, from which a tetrahedral discretisation of the particle is established by connecting each surface triangle to the centre of the particle. By utilising the separating axis theorem [16] along a particular direction, a potential contact list of the tetrahedra from the two particles is generated. From the list, the actual contact status of the two particles can be found by checking the contact between two tetrahedra

on the list. However, no details are provided in [11, 12] on how the contact geometry and the contact force are computed, which are essential for non-spherical particles and should be addressed.

In a series of our recent work [17, 18], a general energy-conserving contact model for arbitrarily shaped particles has been re-established based on the previous work [19, 20]. A specialised contact-volume-based contact model with new computational procedures has also been proposed in [18] to more effectively model contacts between two particles represented by surface triangular meshes. A large number of numerical experiments [18] have been conducted to validate the robustness and applicability of the contact model.

This energy-considering contact modelling framework is also applicable to spherical harmonic particles in discrete forms. However, there are two additional issues to be adequately addressed. The first issue is *how to develop a more effective approach for the surface triangulation of a spherical harmonic shape* than the equal angle discretisation of the 2D parametric space used in [11, 12]. The numerical experiment conducted in [12] suggests that $5000 \sim 6000$ triangles may be needed to represent a spherical harmonic surface with less than 0.5% total volume error. Using such a fine mesh will inevitably be computationally intensive for the contact modelling as the computational complexity of the contact model for a pair of triangular meshes is proportional to the number of triangles involved. This is also a common issue for any real particle that is digitally acquired resulting in a fine triangular mesh representation.

On the other hand, if too fewer vertices/triangles are used to represent a particle surface in order to reduce the computational cost, some surface geometric features may be lost which can also have an adverse effect on the mechanical solution accuracy. Therefore, a balance needs to be made between the level of the geometric approximation and the computational costs. To the author's best knowledge, no previous work has been reported to address this issue. This leads to the second issue: *how to reduce a large triangular mesh down to a size where a reasonable modelling accuracy can still be attained.*

This work will address the above two issues and the paper is organised as follows. Spherical harmonics and its expansion to represent star-shaped surfaces will be briefly reviewed in Section 2. Three schemes used to generate surface triangulation will be discussed in detail. In particular, the golden spiral lattice on the unite sphere $\mathbb{S}^2$ will be recommended as it not only provides a better vertex distribution, but also offers a full control of the number of vertices/triangles to generate. Section 3 presents a simple but effective mesh simplification algorithm based on a well-established edge contraction operation. This approach iteratively removes one vertex and two triangles each time under some local optimal conditions and can achieve any desired level of vertex/triangle reduction. It is highlighted that this approach is applicable not only to surface triangulated star-shaped particles, but also to any triangular mesh for any particle shapes. Thus it provides an ideal pre-processing tool to optimally simplify a fine triangular mesh to a required level. The performance of the algorithm will be assessed in terms of geometric approximation errors. Section 4 outlines the energy-conserving linear contact model [18] that effectively handles the contact between any two triangular meshes. Two examples will be provided in Section 5 to quantitatively assess the geometric approximation errors and the mechanical response of the simplified meshes. Conclusions will be drawn in Section 6.

2 Spherical Harmonic Representation of 3D Shapes

2.1 Spherical Harmonics and Shape Representation

Spherical harmonics are functions defined on the unit sphere $\mathbb{S}^2 = \{\mathbf{x} \,|\, \mathbf{x}^T\mathbf{x} = 1, \mathbf{x} \in \mathbb{R}^3\}$. The spherical harmonic $Y_n^m(\theta, \phi)$ of degree n and order m is expressed as:

$$Y_n^m(\theta, \phi) = c_{n,m} P_n^m(\cos\theta)\, e^{im\phi} \quad (n \geq 0; -n \leq m \leq n) \tag{1}$$

where $i = \sqrt{-1}$; the pair (θ, ϕ) are the coordinates in the parametric domain $[0, \pi] \times [0, 2\pi]$; $c_{n,m}$ is a normalisation constant

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

and $P_n^m(x)$ is the associated Legendre polynomial [21]. Spherical harmonics form a complete set of orthonormal functions. Thus any spherical or radial function $r(\theta, \phi)$ can be expanded as a unique linear combination of these spherical harmonics:

$$r(\theta, \phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} a_{n,m} Y_n^m(\cos\theta) \tag{3}$$

where the complex coefficients $a_{n,m}$ are determined by :

$$a_{n,m} = \int_0^{\pi} \int_0^{2\pi} r(\theta, \phi) \overline{Y}_n^m(\theta, \phi) d\theta\, d\phi \tag{4}$$

in which $\overline{Y}_n^m$ is the complex conjugate of Y_n^m. Since r is a real valued function, the coefficient $a_{n,m}$ has the following property

$$a_{n,-m} = (-1)^m \overline{a}_{n,m}, \quad (m > 0) \tag{5}$$

where $\overline{a}_{n,m}$ is the complex conjugate of $a_{n,m}$. Thus only the coefficients $a_{n,m}$ where $m \geq 0$ need to be evaluated or stored.

The spherical harmonic expansion (3) is equivalent to a Fourier transform for functions defined on the sphere $\mathbb{S}^2$, which transfers the radial function r into its frequency spectrum. In addition to its orthonormality and completeness, spherical harmonics also have a property of a coarse-to-fine representation hierarchy, which make them a natural choice of basis functions to represent star-shaped surfaces.

A shape Ω is said to be star-shaped if there exists a point $\mathbf{c} \in \mathbb{R}^3$ such that every (line) ray emitting from $\mathbf{c}$ in any direction intersects the surface of Ω at exactly one point; which implies that $\mathbf{c}$ must be an interior point of the shape: $\mathbf{c} \in \Omega$ and the shape has genus zero. Thus there is a one-to-one bijective mapping between each point on the surface and one point in the spherical or parametric coordinates. Taking $\mathbf{c}$ as the centre of a spherical coordinate system, the radial function r, denoting the distance from $\mathbf{c}$ to every point on the surface of Ω, is well defined and thus can be represented by spherical harmonics. The spherical harmonics expansion technique can also be extended to more general shapes [13]. In this work, the discussion will be limited to star-shaped particles.

In practice, the expansion (3) is always truncated to include only limited terms

$$r(\theta, \phi) = \sum_{n=0}^{N} \sum_{m=-n}^{n} a_{n,m} Y_n^m(\cos\theta) \tag{6}$$

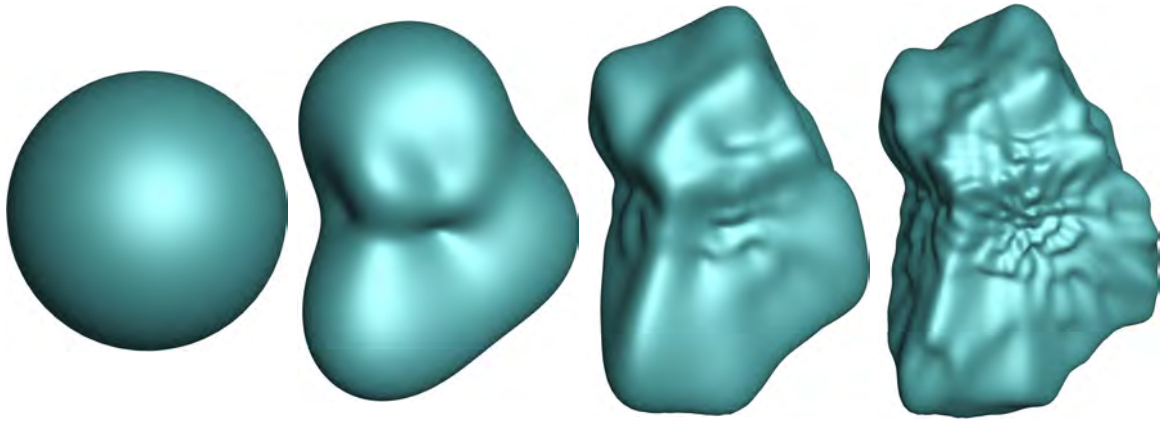


Figure 1: A star-shaped surface approximated by spherical harmonics with an increasing maximum degree $N = 0, 5, 15$ and 30 (from left to right), showing a coarse-to-fine representation hierarchy

where the maximum degree N of the spherical harmonics is determined by the required approximate accuracy and the computational cost involved. This truncation ignores higher frequency components than N, and thus some finer details may be lost. Figure 1 shows how a 3D surface is approximated by different maximum degrees $N = 0, 5, 15, 30$, where fine features of the surface gradually emerge from a sphere when N increases, or vice versa.

For a given N, the total number of coefficients $a_{n,m}$ is $(N+1)^2$. For $N = 30$ (which is used for all the spherical harmonic shapes presented in the current work), there are 961 coefficients. However, due to the property (5), only $a_{n,m}$ with $m \geq 0$ are needed. So the required coefficients can be reduced to $(N+1)(N+2)/2$. The property (5) can also be used to improve the numerical efficiency as will be discussed below.

2.2 Shape Reconstruction

Given a set of coefficients $a_{n,m}$ for the spherical harmonic representation of a surface, the surface can be reconstructed by reversely applying the expansion (6). In practice, an approximate representation of the surface is required, which involves two procedures: point sampling and topology representation. Point sampling obtains sufficient points on the surface by generating their spherical or parametric coordinates (θ, ϕ), while the topology representation provides a connectivity of these sampled surface points. For the purpose of the discrete element modelling to be discussed in Section 4, a triangular mesh from the sampled points will be used to approximately represent a 3D spherical harmonic surface.

2.2.1 Point sampling

For a given pair of coordinates (θ, ϕ) in the parametric domain, the expansion (6) is used to compute the radial coordinate $r(\theta, \phi)$. Then from the resulting spherical coordinates (r, θ, ϕ), the Cartesian coordinates of the point $\mathbf{x} = (x, y, z)$ on the surface can be attained through the transformation

$$\mathbf{x}(\theta, \phi) = r(\theta, \phi)\mathbf{x}_0(\theta, \phi), \quad \text{with } \mathbf{x}_0(\theta, \phi) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \tag{7}$$

Since $\mathbf{x}_0$ is the projection of the parametric coordinates (θ, ϕ) on $\mathbb{S}^2$, the shape reconstruction can be virtually done in two steps: 1) compute the radial distance $r(\theta, \phi)$; and 2) project the parametric point (θ, ϕ) to $\mathbf{x}_0$ on $\mathbb{S}^2$ and scale $\mathbf{x}_0$ by r to obtain the point $\mathbf{x}$.

The key step of the point sampling is to compute the radial function by (6). Clearly, for a large N and many points to be sampled, the computational cost involved in the evaluation can be substantial. It is therefore important to seek a more efficient scheme than directly applying (6).

As $a_{n,m}$ and $e^{im\phi}$ are complex numbers, it is desirable to perform all the operations in (6) in real numbers. Define a (real) spherical associated Legendre polynomial $R_n^m(\theta)$ as

$$R_n^m(\theta) = c_{n,m} P_n^m(\cos\theta) \tag{8}$$

Because the associated Legendre polynomial $P_n^m(x)$ has a property

$$P_n^{-m}(\cos\theta) = (-1)^m P_n^m(\cos\theta) \tag{9}$$

and by using the definition of $c_{n,m}$ in (2), it is easy to establish that

$$R_n^{-m}(\theta) = (-1)^m R_n^m(\theta) \tag{10}$$

Then for $m > 0$, let

$$r_n^m(\theta,\phi) = a_{n,-m} R_n^{-m}(\theta) e^{-im\phi} + a_{n,m} R_n^m(\theta) e^{im\phi} \tag{11}$$

Utilising (5), (10) and $e^{-im\phi} = \overline{e^{im\phi}}$ (the complex conjugate of $e^{im\phi}$), we have

$$r_n^m(\theta,\phi) = R_n^m(\theta)\left(a_{n,m}\, e^{im\phi} + \bar{a}_{n,m}\, \overline{e^{im\phi}}\right) \tag{12}$$

which can be further simplified as

$$r_n^m(\theta,\phi) = 2R_n^m(\theta)\left[a_{n,m}^R \cos m\phi - a_{n,m}^I \sin m\phi\right] \tag{13}$$

where $a_{n,m}^R$ and $a_{n,m}^I$ are the real and imaginary parts of $a_{n,m} = a_{n,m}^R + i a_{n,m}^I$.

Finally (6) can be re-written as

$$r(\theta,\phi) = \sum_{n=0}^{N}\left\{R_n^0(\theta) a_{n,0}^I + 2\sum_{m=1}^{n} R_n^m(\theta)\left[a_{n,m}^R \cos m\phi - a_{n,m}^I \sin m\phi\right]\right\} \tag{14}$$

In this new expression, all the operations involved are real numbers. In addition, only N pairs of $\{\cos(m\phi), \sin(m\phi)\}, m = 1, ..., N$, function evaluations are needed, and can be pre-computed by using the relation $e^{i(m+1)\phi} = e^{i\phi} e^{im\phi}$. The remaining part is the evaluation of each spherical associated Legendre function R_n^m which is already supported by several computer programming languages, or by using (10) where the associated Legendre function P_n^m is supported by almost all modern programming languages.

The new expression (14) can be employed not only to obtain points on the surface, but also to check if a given point, either in the parametric space or in the global (x, y, z) space, is inside or outside of the surface, which is the most important and costly operation in contact detection for spherical harmonic particles [11, 12].

2.2.2 Triangular mesh representation

Several approaches can be adopted to obtain a triangular mesh from the sampled points. A common practice is to firstly construct a triangular mesh in the rectangular (θ, ϕ) domain. Then by making use of the periodicity in ϕ, the same triangular connectivity is used as the mesh for the mapped points on the surface Ω. As parametric coordinates are equivalent to

Table 1: Numbers of vertices (v) and triangles (f) of icosahedral subdivision meshes at different levels

Level	0	1	2	3	4	5
(v, f)	(12, 20)	(42, 80)	(162, 320)	(642, 1280)	(2562, 5120)	(10242, 20480)

spherical coordinates on the unit sphere $\mathbb{S}^2$, an alternative approach attempts to generate a triangular mesh on $\mathbb{S}^2$. Again, the mapped triangular mesh on the surface Ω will be a valid triangulation of the surface. Three triangulation schemes within the above two approaches are introduced as follows.

1). The equal angle grid in $[0, 2\pi] \times [0, \pi]$. Divide this rectangular parametric domain into equal divisions along both θ and ϕ directions to form a regular triangular mesh. This is the most common approach that has been used in the previous work related to discrete element modelling of star-shaped particles [11, 12].

Despite its simplicity, the scheme has an obvious drawback. Figure 2(a) shows such an equal angle triangular mesh from a 26×26 grid in the parametric domain when mapped on the unit sphere $\mathbb{S}^2$. Although the mesh is evenly distributed in the parametric domain, the mapped vertices on the unit sphere are far from an even distribution as the mapping is not distance-preserved. Severe distortion occurs near the two poles where too many vertices are located, resulting in a less efficient mesh representation.

The basic idea to overcome this uneven point distribution issue is to directly use the triangulation of points which are evenly distributed on $\mathbb{S}^2$. Research on distributing points evenly on a sphere has a long history in spherical geometry, but except for a few special cases based on platonic solids, it is generally impossible to precisely distribute equally-spaced points on a sphere, and therefore sub-optimal solutions have to be sought.

The second scheme to be presented is an *optimal* solution based on the recursive refinement of the initial triangulation provided by one platonic solid, icosahedron, and thus can achieve a perfectly even point distribution on $\mathbb{S}^2$. The third scheme is a *sub-optimal* solution which uses the Fibonacci or golden spiral lattice on the parametric domain and the convex hull of these lattice vertices mapped on $\mathbb{S}^2$ as the triangular mesh for spherical harmonic surface reconstruction.

2). The icosahedral subdivision on the unit sphere $\mathbb{S}^2$. The surface triangles of the icosahedron generates a perfect triangular mesh on $\mathbb{S}^2$. To introduce more vertices but maintain the perfectly even distribution property, each triangle is split into four equal sub-triangles by introducing a vertex at the middle point of each edge, then projecting these newly introduced vertices radially onto $\mathbb{S}^2$ to attain a refined triangular mesh. The same procedure can be repeated recursively, resulting in more refined but still evenly distributed triangular meshes until a desired mesh is obtained.

Table 1 lists the numbers of vertices (v) and triangles (f) of the icosahedral triangular meshes at different subdivision levels. The original icosahedral mesh is denoted as level 0. Note that based on Euler's formula, v and f satisfy the relation of $v = f/2 + 2$.

Although a perfectly even point distribution is always created, the main problem with this scheme is that the numbers of vertices/triangles at a particular level are fixed and quadrupled by each subdivision, resulting in increasing number gaps between different levels. Thus the size of the mesh generated cannot be arbitrarily chosen to meet the needs of practical problems.

3). The Fibonacci or golden spiral lattice

The problem with the first equal angle division scheme is that multiple points have the same θ or ϕ value, while the second recursive subdivision scheme cannot generate a mesh with a required number of vertices/triangles. This third scheme attempts to address these two issues but with a slightly less perfect point distribution on $\mathbb{S}^2$. However, this imperfection is not an issue as will be further discussed in the next section.

This scheme uses the Fibonacci lattice in the parametric domain where the parametric coordinates of the lattice nodes are determined by [22, 23]

$$(\theta, \phi)_i = \left(\arccos\left(1 - \frac{2i}{F_2}\right), \frac{2\pi F_1}{F_2} i\right), \quad (0 \le i < F_2) \tag{15}$$

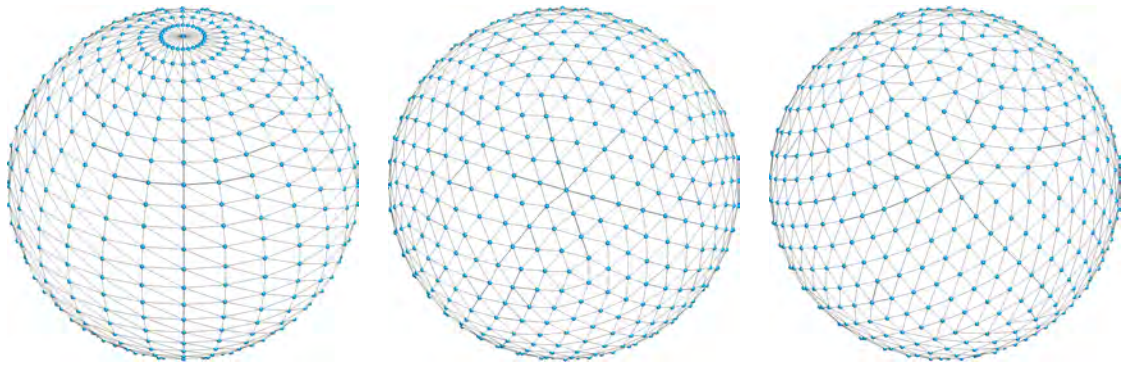
in which F_1 and F_2 (with $F_1 < F_2$) are two consecutive Fibonacci numbers, and F_2 is also the total number of the lattice nodes. One notable difference from the first scheme is that the lattice has the equal height partition along the z-direction, instead of equal angle in θ. Although this lattice achieves a near even point distribution on $\mathbb{S}^2$, it suffers from the same problem as the second scheme: the number of vertices (and triangles) is fixed as one of the Fibonacci numbers.

By recognising the important role that the ratio F_2/F_1 plays in the scheme and the fact that the ratio tends to the golden ratio when F_1 and F_2 increase, a generalised scheme is used instead, in which the lattice coordinates are determined by

$$(\theta, \phi)_i = \left(\arccos\left(1 - \frac{2(i+\epsilon)}{M - 1 + 2\epsilon}\right), \frac{2\pi}{\lambda} i\right), \quad (0 \le i < M) \tag{16}$$

where $\lambda = (1 + \sqrt{5})/2 \approx 1.618$ is the golden ratio; $\epsilon = 0.36$ is an empirically determined parameter to improve the point distribution around the two pole areas [24]; and M is the number of lattice nodes that can be freely set. The resulting lattice is termed the *golden spiral*.

The triangular mesh of the mapped points on $\mathbb{S}^2$ from the Fibonacci or any golden spiral lattice is generated from the convex hull of these points. Compared to the Fibonacci lattice, the golden spiral lattice can have any number of vertices/triangles and therefore is a preferred choice.



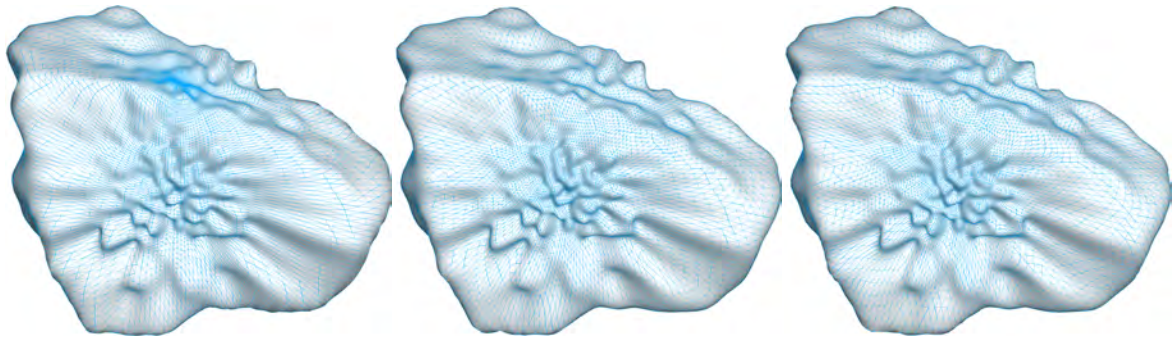
(a) (b) (c)

Figure 2: Triangular meshes on the unit sphere $\mathbb{S}^2$ generated by: (a) a 26×26 equal-angle grid in the parametric (θ, ϕ) domain; (b) the level 3 icosahedral subdivision; and (c) the golden spiral lattice. The number of vertices in (b) and (c) is 642.

Figure 2(b) and (c) respectively show two triangular meshes on $\mathbb{S}^2$, one from the level 3 icosahedral subdivision and the other from the golden spiral lattice, with the same number of vertices $M = 642$. Clearly, a much better point distribution than the first scheme is achieved by both schemes, and between them there is no visually significant difference in the point distribution quality, although the element connectivities are different. However, due to the flexibility in the selection of the number of vertices/triangles, the golden spiral lattice is used to represent a triangulated spherical harmonic shape in this work.

Even though a desirable triangular mesh scheme for $\mathbb{S}^2$ has been found, our goal is to use the mesh to reconstruct a triangulated approximate to a spherical harmonic shape. The performance of the golden spiral lattice and the other two schemes are tested for representing a spherical harmonic shape. This shape has the coefficients $a_{n,m}$ up to the maximum degree $N = 30$. The three mesh representations are displayed in Figures 3(a), (b) and (c), respectively.

As expected, vertices in the equal angle mesh in (a) are clustered around the areas corresponding to the mappings of the two poles of $\mathbb{S}^2$. However, vertices are also closely located in the central area in all the three meshes, even the underlying parametric meshes in (b) and (c) are evenly distributed on $\mathbb{S}^2$. The reason that such vertex clusters occur is that the mapping (6) from $\mathbb{S}^2$ to the actual surface is not distance-preserving, but multiplied by the radial distance $r(\theta, \phi)$. As a result, vertices in the region with smaller r, i.e. closer to the origin of the local coordinate system, will appear with shorter vertex distances. Consequently, a better solution should be sought if the goal is to achieve a more even vertex distribution for any spherical harmonic surface triangulation.



(a) (b) (c)

Figure 3: Three triangular meshes of a star-shaped surface, generated by (a) the equal-angle grid; (b) the icosahedral subdivision; and (c) the golden spiral lattice. The number of vertices in each mesh is around 10240.

3 Triangular Mesh Simplification and Optimisation

As illustrated in Figure 3, using a fairly evenly-distributed mesh on the unit sphere cannot guarantee a high quality representation of a complex 3D shape described by spherical harmonics. The uneven distribution of vertices on the surface reduces the representation accuracy with a fixed number of vertices/triangles. In other words, a triangular mesh with an uneven vertex distribution is not ideal. A desirable approach is to utilise more effective meshing technology to achieve an optimal triangulation for any surface. Hence, curvature-based adaptive meshing would be an ideal option, whereby more points are generated in the areas of a high curvature.

Curvature-based adaptive meshing approaches [25] have found applications in many fields, including solid mesh generation, computer graphics and numerical simulations (finite element modelling in particular). Although different meshing schemes have been proposed and curvature-based meshing is widely available in commercial CAD/CAE software, this is not the preferred meshing approach for spherical harmonic represented surfaces.

In the curvature-based meshing approach, the curvature of a surface point needs to be computed, which is an expensive operation for spherical harmonic represented surfaces - much more costly than a computationally intensive point sampling operation as shown in (6). Also a maximum allowed geometric discretisation error often needs to be specified. However, because the relationship between this error and the size of the resulting mesh is unknown in *a priori*, there is no easy way to control the size of the generated triangular mesh. In addition, no simple meshing algorithm is available.

As mentioned in the introduction, other surface triangulated 3D shapes used in the DEM may be digitally acquired by laser or CT scans [14, 15], and the resulting mesh size may be too large to be effectively modelled. Thus it is also desirable to apply mesh simplification to these meshes. In this case no original underlying surface information is available except for the mesh itself. Thus the curvature-based meshing adaptivity may not be applicable unless some discrete curvature estimation based mesh is used. This would makes the method more tedious.

Hence, an ideal mesh simplification approach in the context of the discrete element modelling should possess the following features: 1) applicable to any surface triangular mesh; 2) using the triangular mesh as the sole input, and no additional information about the underlying surface is required; 3) algorithmically simple to implement in existing discrete element modelling systems to act as a mesh pre-processor; and 4) highly effective and robust.

Fortunately, there exists such an approach that meets all these rather stringent requirements: this is a well established edge contraction (or collapse) based mesh simplification approach, proposed in [26] and extensively used in many applications. In the following subsections, the main aspects of this approach, including quadric error metric, edge contraction, mesh consistency check and iterative edge contraction algorithm, will be briefly presented. The details and further extension of the approach can be found in [26, 27, 28].

3.1 Quadric Error Metric

Consider a plane described by the equation

$$\mathbf{n}^T\mathbf{x} + d = 0 \tag{17}$$

where $\mathbf{n}$ is the unit normal, and d is a constant. Then the squared distance of any point $\mathbf{v}$ to the plane equals to

$$D^2 = (\mathbf{n}^T\mathbf{v} + d)^2 = (\mathbf{n}^T\mathbf{v} + d)^T(\mathbf{n}^T\mathbf{v} + d) = \mathbf{v}^T(\mathbf{n}\mathbf{n}^T)\mathbf{v} + 2d\mathbf{n}^T\mathbf{v} + d^2 \tag{18}$$

which is a quadratic function of $\mathbf{v}$. Define the *quadric* of the plane, Q, to be

$$Q = (\mathbf{A}, \mathbf{b}, c) = (\mathbf{n}\mathbf{n}^T, d\mathbf{n}, d^2) \tag{19}$$

Note that $\mathbf{A} = \mathbf{n}\mathbf{n}^T$ is symmetric, and therefore it requires 10 coefficients to store Q. Then D^2 is defined as the *quadric error* of $\mathbf{v}$ associated with Q

$$Q(\mathbf{v}) = \mathbf{v}^T\mathbf{A}\mathbf{v} + 2\mathbf{b}^T\mathbf{v} + c \tag{20}$$

For a triangle with the outer unit normal $\mathbf{n}$, its support plane is defined as the one on which the triangle resides. Thus, the quadric Q of the triangle is defined to be the same as (19). Every vertex $\mathbf{v}$ in a triangular mesh has a set of adjacent triangles, denoted as $\mathcal{F}$.

The quadric of vertex $\mathbf{v}$ is defined as the weighted summation of the quadrics from all its adjacent triangles

$$Q_v = \sum_{i \epsilon \mathcal{F}} w_i Q_i \tag{21}$$

where $Q_i = (\mathbf{A}_i, \mathbf{b}_i, c_i)$ is the quadric of the i-th adjacent triangle in $\mathcal{F}$; and w_i is the associated weighting. Possible choices of the weighting include: 1) $w_i = 1$; 2) $w_i = a_i$ where a_i is the area of the i-th triangle; and 3) $w_i = a_i^2$. The first two options are used in [26, 28], while the third one is proposed in [29] and used in [30].

If all triangles have similar areas, the above three weighting options give similar results. Otherwise, the second and third options will produce better results. Because the quadric error of a triangle describes the squared distance of a point to the triangle (or its support plane), when the quadric is weighted by the squared area of the triangle, the resulting error will be the squared tetrahedral volume formed by the triangle and the point (multiplied by a constant factor of 36). Thus, the third weighting method has a clear geometric interpretation. In particular, when this weighted quadric error is used in the subsequent edge contraction based mesh simplification, a better volume preservation can be achieved than the second area weighted method. This observation, which is highlighted in [29, 30], is also numerically confirmed in our experiments. However, if the volume preservation has to be ensured, another vertex replacing scheme can be derived [29], by imposing this condition as a constraint to the third weighting method, the position of the replaced vertex can be determined.

The addition of two quadrics Q_1 and Q_2 is performed component-wise:

$$Q_1 + Q_2 = (\mathbf{A}_1, \mathbf{b}_1, c_1) + (\mathbf{A}_2, \mathbf{b}_2, c_2) = (\mathbf{A}_1 + \mathbf{A}_2, \mathbf{b}_1 + \mathbf{b}_2, c_1 + c_2) \tag{22}$$

Thus the addition of two quadric errors of a point $\mathbf{v}$ is equal to the quadric error of the added quadric:

$$Q_1(\mathbf{v}) + Q_2(\mathbf{v}) = (Q_1 + Q_2)(\mathbf{v}) \tag{23}$$

Note that the quadric error of each vertex of a triangular mesh is zero. The vertex quadrics of a given triangular mesh can be conveniently constructed by processing all the triangles of the mesh.

3.2 Edge contraction

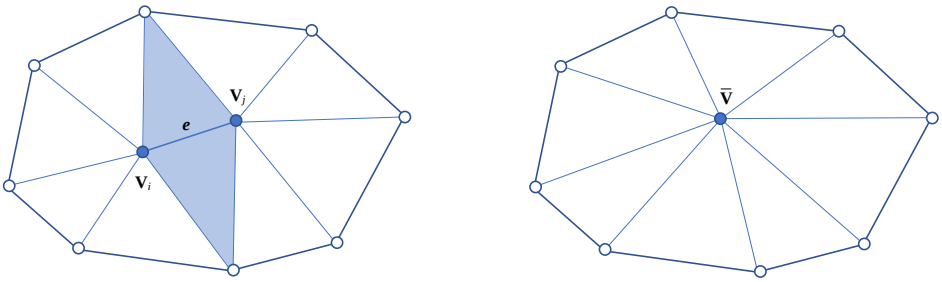
An edge of a triangular mesh is incident to two vertices. Let $e = (i, j)$ denote an edge e with two vertices i and j, and their coordinates $\mathbf{v}_i$ and $\mathbf{v}_j$, respectively. The quadric of an edge e is defined as the summation of the quadrics of its two vertices i and j:

$$Q_e = Q_i + Q_j = (\mathbf{A}_e, \mathbf{b}_e, c_e) \tag{24}$$

An edge contraction is to replace its two end vertices $(\mathbf{v}_i, \mathbf{v}_j)$ by another vertex $\bar{\mathbf{v}}$. The associated cost is the edge quadric error of $\bar{\mathbf{v}}$

$$Q_e(\bar{\mathbf{v}}) = Q_i(\bar{\mathbf{v}}) + Q_j(\bar{\mathbf{v}}) \tag{25}$$

The replacing vertex $\bar{\mathbf{v}}$ can be obtained in different ways, depending on the replacing criterion used. Three criteria are given below.



(a) Before (b) After

Figure 4: Edge contraction: (a) edge e with vertices $\mathbf{v}_i$ and $\mathbf{v}_j$ is to be contracted; (b) e is replaced by $\bar{\mathbf{v}}$ and all adjacent triangles are updated

Criterion 1: $\bar{\mathbf{v}}$ is chosen as either $\mathbf{v}_i$ or $\mathbf{v}_j$ depending on which one has a smaller quadric error. So

$$\bar{\mathbf{v}} = \underset{\mathbf{v}\in\{\mathbf{v}_i,\mathbf{v}_j\}}{\operatorname{argmin}} Q_e(\mathbf{v}) \tag{26}$$

Criterion 2: $\bar{\mathbf{v}}$ is a vertex located on the edge (i, j) that has the minimum quadric error. A vertex $\mathbf{v}$ on the edge can be expressed as a function of parameter $t \in [0, 1]$

$$\mathbf{v}(t) = t\mathbf{v}_i + (1-t)\mathbf{v}_j, \quad 0 \le t \le 1 \tag{27}$$

Its quadric error can be expressed as a quadric function of t:

$$Q_e(\mathbf{v}(t)) = t^2\mathbf{v}_i^T\mathbf{A}_e\mathbf{v}_i + (1-t)^2\mathbf{v}_j^T\mathbf{A}_e\mathbf{v}_j + 2t(1-t)\mathbf{v}_i^T\mathbf{A}_e\mathbf{v}_j + 2t\mathbf{b}_e^T(\mathbf{v}_i - \mathbf{v}_j) + 2\mathbf{b}_e^T\mathbf{v}_j + c_e \tag{28}$$

from which the t that minimises the error can be obtained as

$$\bar{t} = \left(\mathbf{b}_e^T(\mathbf{v}_j - \mathbf{v}_i) - \mathbf{v}_i^T\mathbf{A}_e\mathbf{v}_j\right) / \left(\mathbf{v}_i^T\mathbf{A}_e\mathbf{v}_i + \mathbf{v}_j^T\mathbf{A}_e\mathbf{v}_j - 2\mathbf{v}_i^T\mathbf{A}_e\mathbf{v}_j\right) \tag{29}$$

When $\bar{t} \notin [0, 1]$, it is set to be the nearest end point 0 or 1. Then the replacing vertex is obtained as

$$\bar{\mathbf{v}} = \bar{t}\mathbf{v}_i + (1-\bar{t})\mathbf{v}_j \tag{30}$$

Criterion 3: $\bar{\mathbf{v}}$ minimises the quadric error $Q_e(\mathbf{v})$ with the following value [26]

$$\bar{\mathbf{v}} = -\mathbf{A}_e^{-1}\mathbf{b}_e \tag{31}$$

and the corresponding quadric error can be attained by

$$Q_e(\bar{\mathbf{v}}) = \mathbf{b}_e^T\bar{\mathbf{v}} + c_e \tag{32}$$

For the above three replacing criteria, the last one provides the smallest quadric error. The second criterion restricts the minimisation on the edge and therefore gives a larger error. The first criterion produces the largest error, but because one of the vertices is retained, the edge contraction with this criterion may be equivalent to a simple vertex decimation scheme without local re-triangulation [31]. Our experiments show that Criterion 3 will generally lead to a better simplified mesh, and thus will be used in this work.

For an edge $e = (i, j)$ to be contracted, a number of operations are performed to update the geometry and topology of the mesh. Figure 4 illustrates an edge contraction operation, where the highlighted edge in (a) is contracted or collapsed into a single point $\bar{\mathbf{v}}$ shown in (b), while the two shaded triangles are removed. Only the remaining vertices, edges and triangles in (b) are affected by the contraction and their quadrics need to be updated.

The procedure is summarised in Algorithm 1. The updates involved can be done efficiently. The effect of an edge contraction is therefore small and highly localised.

Algorithm 1: Edge Contraction Operations

1. Given edge $e = (i, j)$ and replacing vertex $\bar{\mathbf{v}}$
2. Set $\mathbf{v}_i = \bar{\mathbf{v}}$; Delete $\mathbf{v}_j$
3. Remove the two adjacent triangles of e
4. Merge the adjacent triangles of vertices i and j to form a new adjacent triangle set: $\mathcal{F}_i = \{f_{i_1}, f_{i_2}, ..., f_{i_k}\}$
5. Update edge connectivities of the adjacent triangles in $\mathcal{F}_i$
6. Recalculate triangle quadrics in F_i and update the relevant vertex quadrics

3.3 Mesh Consistency Check

An edge contraction will alter the adjacent triangles of the two vertices of the edge, and also change their normals. When the change to a triangle is too large, the triangle may fold over on some neighbouring triangles, resulting in an unacceptable mesh configuration [26].

To avoid such an event, a mesh consistency check will be performed before the edge contraction is permitted to take place.

Let $\mathcal{F}_e$ be a set of triangles that will be affected by the candidate edge to be contracted; and $\mathbf{n}_i^0$ and $\mathbf{n}_i$ be the normals of the i-triangle in $\mathcal{F}_e$ before and after the contraction. If the following condition is not met for a triangle in $\mathcal{F}_e$

$$\mathbf{n}_i^T \mathbf{n}_i^0 > \tau \tag{33}$$

where τ is a prescribed tolerance, then the mesh consistency check fails and the proposed edge contraction is rejected. In addition, the cost of this edge is heavily penalised to prevent the edge from being selected again in the same iterative cycle, or the edge is disallowed for further contraction.

This simple pre-edge contraction check is applied in [26] where $\tau = 0$ is used. To ensure that no mesh alteration will compromise the subsequent discrete element simulation, a fairly large value $\tau = 0.75$ is chosen in this work.

3.4 Iterative Edge Contraction Algorithm

For a given triangular mesh, mesh simplification can now be performed by iteratively following the steps outlined in the previous two subsections. The procedure is summarised in Algorithm 2. Because each edge contraction removes one vertex and two triangles, the exact size of the final mesh produced by the simplification procedure can be fully controlled.

Note that the quadrics of the affected triangles, vertices and edges are re-calculated after an edge contraction operation. Thus this procedure is the same as the so-called memoryless simplification used in [29, 30]. Their tests show that this procedure performs better than the original version [26].

Also note that no special treatments are considered for possible degenerated cases that may occur in the evaluation of the replacing vertex using (31), where matrix $\mathbf{A}$ may be (near)

singular. A possible solution is to use the pseudo-inverse of $\mathbf{A}$ from its singular value decomposition but ignore the smallest singular value.

Algirithm 2: Iterative Edge Contractions

1. Input: a triangular mesh and the targeted number of vertices.
2. Initialisation
   a. Compute quadrics of all triangles and accumulate vertex quadrics by (21);
   b. Compute all edge costs and replacing vertices by (30) or (31).
3. Iteration
   a. Find the edge with the least contraction cost: $e : (\mathbf{v}_i, \mathbf{v})$ with replacing vertex $\bar{\mathbf{v}}$.
   b. Check the mesh consistency condition (33); If not met, penalise the edge contraction cost by adding a large value, and go back to 2.a
   c. Locally update mesh geometry, topology and quadrics (Algorithm 1).
   d. If the number of remaining vertices/triangles is equal to the required value, exit.
   e. Update affected edge costs and replacing vertices.

3.5 Illustrative Example

The performance of the edge contraction based mesh simplification scheme has been extensively tested [26, 30]. This subsection employs the same star-shaped spherical harmonic representation as used in Figure 3 to further demonstrate the high quality of simplified meshes that the algorithm can produce. To quantitatively measure the accuracy of different levels of simplification, the relative errors of the total volume and three principal moments of inertia (MOI) of a simplified mesh are compared with the original mesh represented shape. The volume and MOI are chosen because they play an important role in the discrete element modelling. The performance of the simplified meshes will be further assessed in the particle simulations in Section 5.

The original meshed surface is generated by the golden spiral lattice algorithm and has $20,000$ vertices and 39,996 triangles. This mesh is progressively simplified down to the final mesh with 100 vertices and 196 triangles, which is only 0.5% of the original mesh. Table 2 lists the relative errors of the total volume and MOIs of the intermediately simplified meshes. The volume and MOIs of the original shape serve as the exact values for the error calculations. Clearly, excellent approximations have been achieved by the edge contraction operation: even for the mesh with 100 vertices, the errors are only around 1%.

Figures 5(b)-(d) show three simplified meshes with the number of vertices being 10,000, 1,000, and 100 respectively. It can be seen from (b) that the first 10,000 edge contractions, which remove 10,000 vertices, mainly take place around the central region where the vertices are clustered in the original mesh. It indicates that the edge contraction can indeed act as a mesh optimiser that removes/redistributes vertices to achieve a more effective mesh representation. The mesh in (c) has 1000 vertices (which is only 5% of the initial number), but it still captures well the main features of the original surface. Even with 0.5% of the original vertices, the

Table 2: Approximation errors of volume and principal moments of inertia (MOI) for meshes simplified by edge contraction and without simplification (marked by * in the first column)

No.	Volume		MOI	
vertices	value	error	value	error
20,000	647.300	-	(25.3128, 8.91566, 20.1384)	-
10,000	647.294	1.00e-05	(25.3129, 8.91557, 20.1384)	(4.27, 9.69, 3.93)e-06
5,000	647.251	7.60e-05	(25.3123, 8.91526, 20.1378)	(2.02, 4.49, 2.97)e-05
2,000	647.029	4.19e-04	(25.3079, 8.91344, 20.1343)	(1.94, 2.49, 2.04)e-04
1,000	646.616	1.06e-03	(25.2990, 8.90916, 20.1287)	(5.45, 7.29, 4.78)e-04
200	643.113	6.46e-03	(25.2756, 8.86869, 20.1267)	(1.47, 5.26, 0.57)e-03
100	638.392	1.37e-02	(25.3230, 8.81467, 20.1943)	(0.04, 1.13, 0.27)e-02
$10,000^*$	646.399	1.39e-03	(25.2696, 8.89849, 20.1094)	(1.71, 1.93, 1.44)e-03
$1,000^*$	632.473	2.29e-02	(24.6172, 8.66333, 19.6447)	(2.75, 2.83, 2.45)e-02
100^*	551.426	1.48e-01	(20.8757, 7.84251, 16.6276)	(1.75, 1.20, 1.74)e-01

coarsest mesh shown in (d) can still reasonably represent the original surface.

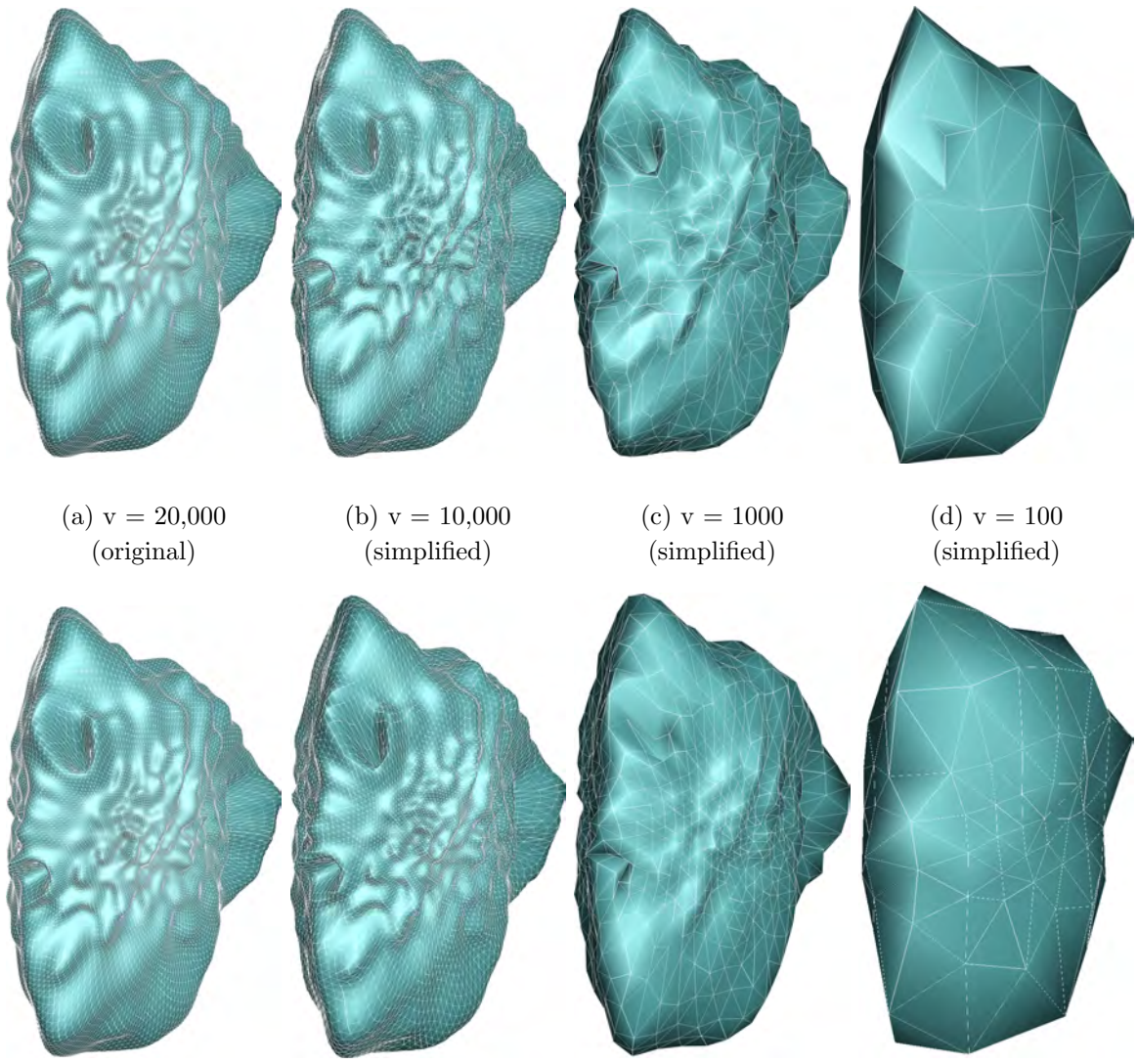
To further highlight the benefit that edge contraction offers, three triangular meshes with the number of vertices 10,000, 10000 and 100, directly generated based on the golden spiral lattice without simplification, are also showed in Figures 5 (f)-(h), respectively. Their visually inferior approximation compared to the counterparts with mesh simplification in (b)-(d) is clearly demonstrated, and this is quantitatively assessed by the relative errors of the total volumes and MOIs shown in Table 2. It is evident that the mesh simplified from an initial fine mesh exhibits a much higher approximation accuracy than the one with the same size but directly generated by the golden spiral lattice.

In summary, the edge contraction based mesh simplification method is highly effective, robust and simple to implement. It can be readily incorporated into an existing discrete element modelling framework to serve as a mesh optimiser. As will be demonstrated in later examples in Section 5, particle simulations using simplified meshes can attain as good modelling accuracy as using fine meshes, but with much reduced computational costs. An additional benefit of using a simplified mesh is that the actual volume and MOIs can be taken from the initial mesh to further reduce the simulation error.

4 Contact Detection and Energy-Conserving Normal Contact Model for Surface Triangulated 3D Shapes

Consider a general discrete element system with particles of any shape (convex or concave) and represented by surface triangular meshes. In a standard discrete element modelling procedure, as outlined in [32], contact detection plays the most important role. Broadly speaking, contact detection takes three steps: 1) global search - using bounding boxes of particles to establish a potential contact list for each particle; 2) local contact resolution - for any pair of particles in the contact list, using their actual geometric shapes to check if they are in real contact. Contact geometric features are then computed for the pair in contact; and 3) contact force computation - for each pair of particles in contact, apply contact interaction laws to evaluate the contact forces based on the contact geometric features.

The surface triangular mesh of a particle can have a large number of vertices/triangles, while



(a) v = 20,000 (original) (b) v = 10,000 (simplified) (c) v = 1000 (simplified) (d) v = 100 (simplified)

(e) v = 20,000 (same as (a)) (f) v = 10,000 (no simplification) (g) v = 1000 (no simplification) (h) v = 100 (no simplification)

Figure 5: (a, e) Original mesh generated from the golden spiral lattice; (b)-(d) simplified meshes with different numbers of vertices; and (f)-(g) different meshes generated from the golden spiral lattice without simplification

the computational cost associated with the local resolution for two such particles is at least proportional to the number of vertices involved. In order to improve the overall efficiency of the contact detection, strategies to balance the computational cost of the global search against the local resolution have been proposed in [18].

The next subsection will address the evaluation of the contact geometric features. The normal contact model that is essential to evaluate the contact forces will be discussed in Section 4.2.

4.1 Contact Geometry

Consider a general contact between two arbitrary 3D shapes represented by surface triangulated meshes $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively. For most contact models employed by the discrete element method, the contact overlap is the most important contact geometric feature. How-

ever, in the energy-conserving contact modelling framework established in [20], the contact volume plays a paramount role. To explicitly compute the contact volume is computationally intensive, and thus makes this special energy-conserving contact model practically less appealing than contact-overlap-based contact models, despite the fact that this energy-conserving contact model offers superior numerical accuracy and stability, and is also applicable to both convex and concave shapes.

This computationally intensive issue has been resolved in [18] with an alternative scheme to evaluate the contact geometric features. Instead of computing the contact volume, only the intersection boundary of two contacting particles needs to be computed. As a result, the computational cost associated with this energy-conserving contact model is significantly reduced.

In the energy-conserving contact model with the new approach, the major computations involve the construction of the intersection boundary $\Gamma = \mathcal{T}_1 \cap \mathcal{T}_2$ and two associated line (vector) integrals $\mathbf{S}_n$ and $\mathbf{G}_n$ over Γ,

$$\mathbf{S}_n = \frac{1}{2} \oint_\Gamma \mathbf{x} \times d\boldsymbol{\Gamma} \tag{34}$$

$$\mathbf{G}_n = -\frac{1}{3} \oint_\Gamma \mathbf{x} \cdot \mathbf{x} d\boldsymbol{\Gamma} \tag{35}$$

The computational issues are briefly discussed below. Full details can be found in [18].

4.1.1 Construction of Intersection Line Segments

As $\mathcal{T}_1$ and $\mathcal{T}_2$ are triangular meshes, their intersection boundary Γ is a closed 3D polyline for a single contact, or multiple closed polylines for multiple contacts. A single contact case will be considered next, while multiple contacts can be treated as individual polylines in a similar fashion.

Since Γ is a polyline, its segments are formed by the intersections of the individual triangles from the two surfaces. The basic numerical procedure is to determine the line segment of the intersection (if exists) between two given triangles from each surface. This operation can be performed in a straightforward manner. There are two intersection points if two triangles are in contact. If the two triangles lie on the same plane, it is treated as no contact. The start and end points of the segment should be properly ordered to ensure a correct orientation of Γ. The positive direction $\boldsymbol{\tau}$ of the segment is determined by

$$\boldsymbol{\tau} = \mathbf{n}_1 \times \mathbf{n}_2 \tag{36}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are respectively the outer normal directions of the two triangles.

4.1.2 Computations of $\mathbf{S}_n$ and $\mathbf{G}_n$

Assume that $\mathbf{x}_i, (i = 1, ..., m+1)$ with $\mathbf{x}_1 = \mathbf{x}_{m+1}$ are m vertices that form m line segments of Γ. Denote the i-th segment as $\Delta\Gamma_i = [\mathbf{x}_i, \mathbf{x}_{i+1}]$, and $\Delta\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$.

In order to compute $\mathbf{S}_n$ and $\mathbf{G}_n$, each intersection segment can be locally parameterised. By introducing a local coordinate system on the i-th segment with parameter $t \in [0, 1]$, the parametric equation and the infinitesimal increment of the segment are

$$\mathbf{x}(t) = \mathbf{x}_i + t\,\Delta\mathbf{x}_i; \quad d\boldsymbol{\Gamma} = \Delta\mathbf{x}_i\, dt \tag{37}$$

Then the integral of $\mathbf{S}_n$ reduces to a simple summation over the line segments

$$\mathbf{S}_n = \frac{1}{2}\sum_{i=1}^{m} \mathbf{x}_i \times \mathbf{x}_{i+1} \tag{38}$$

and $\mathbf{G}_n$ becomes

$$\mathbf{G}_n = -\frac{1}{3}\sum_{i=1}^{m} \left(\mathbf{x}_i \cdot \mathbf{x}_{i+1} + \frac{1}{3}\Delta\mathbf{x}_i \cdot \Delta\mathbf{x}_i\right) \Delta\mathbf{x}_i \tag{39}$$

Both $\mathbf{S}_n$ and $\mathbf{G}_n$ are additive as explained in [18]. Due to this property, their computations can be undertaken in a segment by segment fashion and in any order. Thus the procedures outlined above, including the local contact search, the triangle-triangle intersection check and computation, can be fully parallelised using multiple CPU, GPU or GPGPU computing. As only a minimal level of communication overhead is incurred, a high parallel efficiency is expected.

4.1.3 Contact normal and contact point/line

Once $\mathbf{S}_n$ and $\mathbf{G}_n$ are evaluated, two contact geometric features, i.e. the contact normal $\mathbf{n}$ and contact point $\mathbf{x}_c$, can be determined by

$$\mathbf{n} = -\mathbf{S}_n/S_n; \quad S_n = |\mathbf{S}_n| \tag{40}$$

and

$$\mathbf{x}_c = \mathbf{n} \times \mathbf{G}_n/S_n + \lambda\mathbf{n} \tag{41}$$

where λ is a free parameter, meaning that it can take any value in principle. In other words, the contact point is not unique. When λ takes different values, the point forms a line, and it is called the contact line. This non-uniqueness does not cause any issue for the normal contact, but the location of the point on the contact line will have an impact on the tangential force. By introducing the minimal contact surface, a simple formula has been derived in [18] to determine the unique contact point. Alternatively, as the contact normal line will normally intersection a point on each surface and the middle point of these two pints can be taken as the contact point.

4.2 The Energy-Conserving Linear Normal Contact Model

The contact-volume-based energy-conserving linear normal contact model, or the linear contact model for short, is developed in [17, 18, 20] based on the assumption that the normal contact energy w can be expressed as a linear function of the contact volume $V_c = |\mathcal{T}_1 \cap \mathcal{T}_2|$ between two contacting particles

$$w = k_n V_c \tag{42}$$

and the normal contact force $\mathbf{F}_n$ is the negative gradient of this energy function

$$\mathbf{F}_n = -\nabla_{\mathbf{x}} w = k_n \mathbf{S}_n = k_n S_n \mathbf{n} \tag{43}$$

where $\mathbf{x}$ is taken as the coordinates of a reference point on the first particle. Further derivations lead to a liner expression for the contact force in terms of $\mathbf{S}_n$

$$\mathbf{F}_n = k_n \mathbf{S}_n = F_n \mathbf{n} \tag{44}$$

where $F_n = k_n S_n$ is the magnitude of the force. The condition that no contact moment exists at the contact point leads to the formula (41), where $\mathbf{G}_n$ is introduced.

Geometrically, $\mathbf{S}_n$ is a signed area of the contact surface of the two particles, and $\mathbf{G}_n$ can be interpreted as the "moment" of the contact surface. $\mathbf{S}_n$, instead of the contact overlap, plays the leading role in the current contact-volume-based linear contact model, which is summarised in Algorithm 3.

Algorithm 3: Normal Contact Model between Two Surface Triangulated Shapes

1. Input: two triangular meshes $\mathcal{T}_1$ and $\mathcal{T}_2$; contact stiffness k_n
2. Compute contact geometry
   a. Construct the intersection polyline $\Gamma = \mathcal{T}_1 \cap \mathcal{T}_2$. If $\Gamma = \varnothing$, exit (NO CONTACT);
   b. Compute $\mathbf{S}_n$ and $\mathbf{G}_n$ using (30) and (31).
3. Compute contact features
   a. Contact normal: $\mathbf{n} = \mathbf{S}_n / S_n$, $S_n = |\mathbf{S}_n|$.
   b. Contact force magnitude: $F_n = k_n S_n$
   c. Contact point/line: $\mathbf{x}_c = \mathbf{n} \times \mathbf{G}_n / S_n + \lambda \mathbf{n}$

This linear normal contact model has the following distinct properties [18]: 1) energy conservation guaranteed for any elastic impact; 2) applicable to any shape, 2D or 3D, convex or concave; 3) capable of handling both single or multiple contact scenarios; and 4) highly parallelizable.

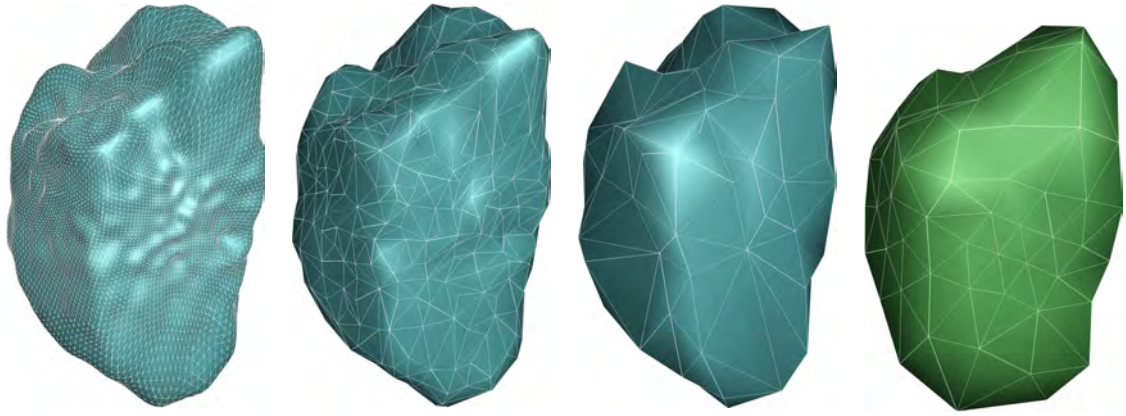
5 Discrete Element Validation of Particles with Simplified Meshes

Two examples will be presented in this section. The first example will quantitatively assess the performance of simplified meshes in the context of the discrete element modelling where pair-wise particles represented by such meshes are subject to elastic and inelastic impacts. It will also demonstrate the energy-conserving nature of the linear contact model for the elastic impact case. The second example attempts to further quantify the mechanical behaviour of simplified meshes in a more complex setting.

The classic central difference scheme is employed as the time integrator to solve the translational motion of the particles, while the rotational motion is resolved by a symplectic time integration scheme [33] in conjunction with the quaternion representation of the particle orientation. The time step is chosen such that both numerical stability and accuracy can be achieved following an empirical formula proposed in [34].

5.1 Pair-wise Impact of Star-Shaped Particles

A star-shaped particle is represented by a triangular mesh with 10,000 vertices as the initial input mesh. Then it is simplified to two meshes with 1,000 and 100 vertices, respectively.



(a) v = 10,000 (original) (b) v = 1000 (simplified) (c) v = 100 (simplified) (d) v = 100 (original)

Figure 6: Four triangular meshes of a star-shaped surface: from the original mesh with vertices v = 10,000 (a); simplified to v = 1000 (b), and further down to v = 100 (c); another original mesh with v = 100

The fourth mesh is directly generated from the shape with 100 vertices. The four meshes, denoted as 1,2,3 and 4, are shown in Figure 6. Similar to the assessment made in Section 3.5, the relative errors of the total volume and MOIs are computed and given in Table 3, where the values of the original mesh with 10,000 vertices are used as the reference values. Again, the two simplified meshes 2 and 3, with only 10% and 1% vertices of the original mesh 1 respectively, have a high approximation accuracy, while Mesh 4 has much larger approximate errors.

As mentioned in Section 3.5, the volume and MOIs (and their directions) of any simplified mesh can be set to be those of the initial mesh to further reduce the geometric errors. However, this cannot be done for Mesh 4 as the mesh is directly imported and no properties of the original shape is supposed to be available. Since the total volume and MOIs directly affect the motion of a particle, Mesh 4 is expected to have a large modelling error.

To quantitatively assess the impact of mesh simplification on the mechanical behaviour of particles, two identical particles with the same mesh representation in each mesh case are subject to impact, leading to four pair-wise test cases with identical initial conditions. In each case, the two particles have the same initial velocity but in the opposite direction. They are initially separated, then bought into contact as a result of the specified initial velocities ([1, 0, 1] and [-1, 0, -1]). After bouncing off, they become separated again.

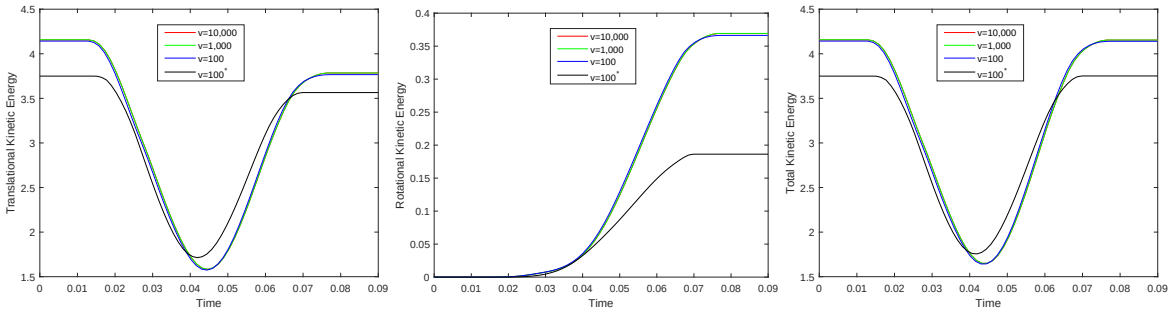
Table 3: Relative rrors of volume and principal moments of inertia (MOIs) for meshes simplified by edge contraction and without simplification (mesh 4)

Mesh No.:	Volume		MOIs	
No. vertices	value	error	value	error
1: 10,000	1876.64	-	(34.3312 36.0142 18.3829)	-
2: 1,000	1875.63	5.40e-04	(34.3184 35.9992 18.3748)	(3.74 4.16 4.42)e-04
3: 100	1869.03	4.06e-03	(34.2184 35.8656 18.3498)	(3.28 4.12 1.80)e-03
4: 100*	1686.92	1.01e-01	(29.8903 31.0186 16.9758)	(1.29 1.39 0.76)e-01

A small value of the contact stiffness k_n is used in order to induce a large contact overlap for a more complex impact process. Another reason of using a small stiffness is to prolong the impact duration so that the rotational motion of particles can be fully developed to make the contact more sophisticated.

For each test case, three impact scenarios are considered: 1) elastic impact; 2) inelastic impact with friction; and 3) inelastic impact with friction and damping. The coefficient of friction is 0.5 for the 2nd and 3rd cases. The viscous damping ratio ξ, equivalent to a restitution of 0.1, is applied to both normal and tangential directions in case 3. Gravity is ignored. The other settings are the same in all the four cases.

1) Elastic impact. Time evolutions of translational, rotational and total energy for the frictionless contact are compared in Figures 7(a), (b) and (c). As the same volume and MOIs are used in the first three cases, their initial energy is the same, but this is not the case for the fourth case. It can be seen that the first three cases exhibit almost identical evolutions, while the fourth case has a significantly different behaviour. The difference is particularly pronounced in the rotational motion as shown in Figure 7(b)



(a) Translational (b) Rotational (c) Total

Figure 7: Time evolutions of translational (a), rotational (b) and total (c) kinetic energies of the four pair-wise elastic impact cases without friction

As expected, energy conservation of the elastic impact without friction is achieved in all cases as shown in Figure 7(c), where the total energy for each case remains unchanged before and after the impact.

Th snap shots of the frictionless contact at four different time instants for each case are depicted in Figure 8. The four rows from the top correspond to four impact stages: 1) initial contact; 2) advanced stage; 3) maximum penetration; and 4) bouncing off. The contact profiles of the first three cases are similar, but the fourth one is significantly different.

2). Inelastic impact with friction $\mu = 0.5$.

For the frictional contact, the system energy is expected to lose during the impact. Due to the surface irregularity, multiple contacts occur particularly in the initial and near end contact, as can be seen from the configurations in Figure 8. The classic Coulomb friction model is applied to each independent contact. Details can be found in [18, 34].

The simulation results are plotted in Figures 9(a), (b) and (c) respectively. Again the first three cases exhibit almost identical energy evolutions, while the fourth case behaves differently.

3). Inelastic impact with friction $\mu = 0.5$ **and damping ratio** $\xi = 0.1$.

In addition to the friction effect, the damping effect is also considered in both normal and

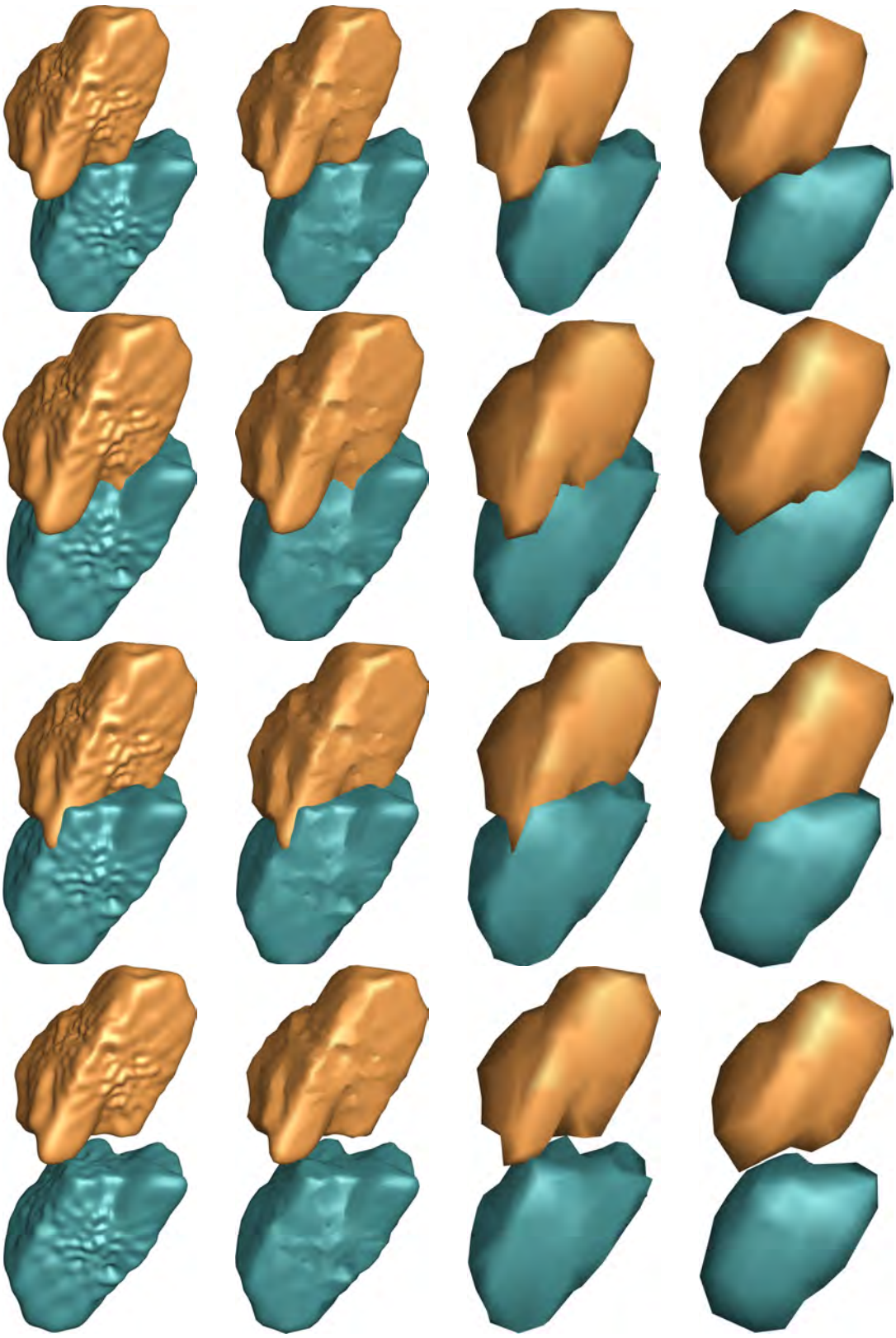
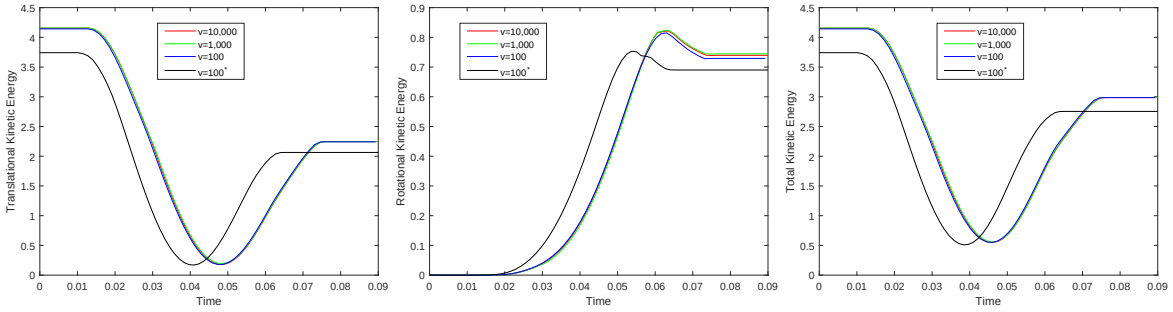


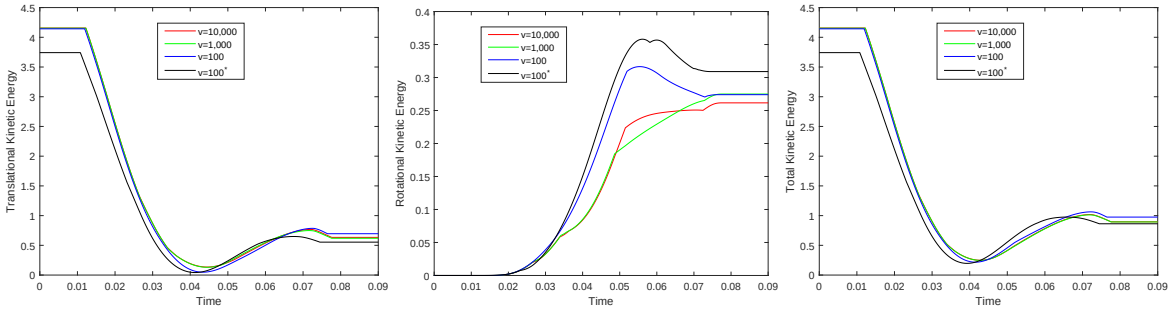
Figure 8: Particle configurations of the four cases at different time instants. Each row represents the four cases. Columns (from top to bottom) correspond to four impact stages: initial contact, advanced contact, maximum penetration, and bouncing off



(a) Translational (b) Rotational (c) Total

Figure 9: Time evolutions of translational (a), rotational (b) and total (c) kinetic energies of the four pair-wise inelastic impact cases with friction $\mu = 0.5$

tangential contacts. However, the damping treatment for multiple contacts is different from that of the friction. Instead of applying a viscous damping term to each contact, it is weighted by the ratio of the S_n for this contact to the total S_n summed up from all the contacts [34]. The simulated results are plotted in Figures 10(a), (b) and (c). The first three cases still show very good agreement in the translational and total energy curves, while the rotational energy of the third case is diverted in the middle of the impact from the other two more noticeably, but the exit value is almost the same as the second case. The second case also shows some small discrepancy from the first case at a later stage of the simulation. This indicates that damping has a larger effect than friction on the mechanical response of a simplified mesh. In comparison, the fourth case shows the largest error.



(a) Translational (b) Rotational (c) Total

Figure 10: Time evolutions of translational (a), rotational (b) and total (c) kinetic energies of the four pair-wise inelastic impact cases with friction $\mu = 0.5$ and damping ratio $\xi = 0.1$ for both normal and tangential contacts

The results from the three impact scenarios demonstrate that the edge contraction based mesh simplification generates a very small geometric approximation error, and the discrete element modelling with the simplified meshes can achieve a very similar mechanical response as using the original fine mesh.

5.2 Random Deposition of Star Shaped Particles in a Container

The previous example confirms that the mesh simplification by edge contraction generates a much smaller sized mesh but maintain small geometric and mechanical errors for a pair-wise impact under various conditions. One more example will be presented to examine the

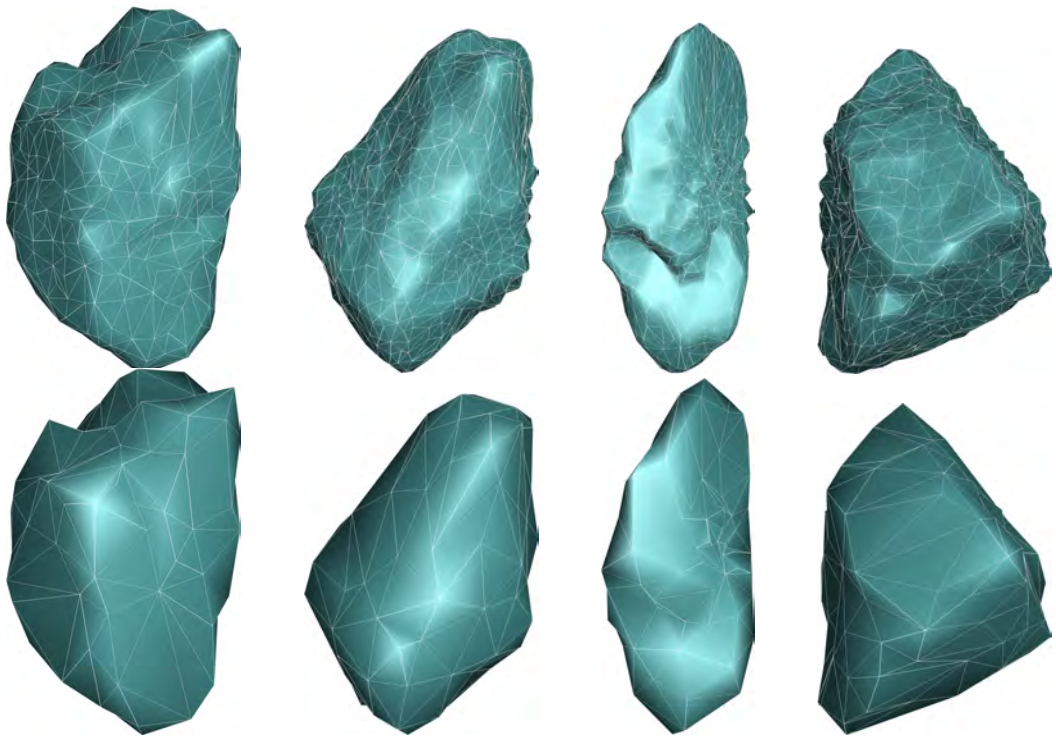


Figure 11: Triangular meshes of four star-shaped surfaces (from left to right) simplified from the original meshes with vertices v = 10,000 to v = 1000 (first row) and further down to v = 100 (second row)

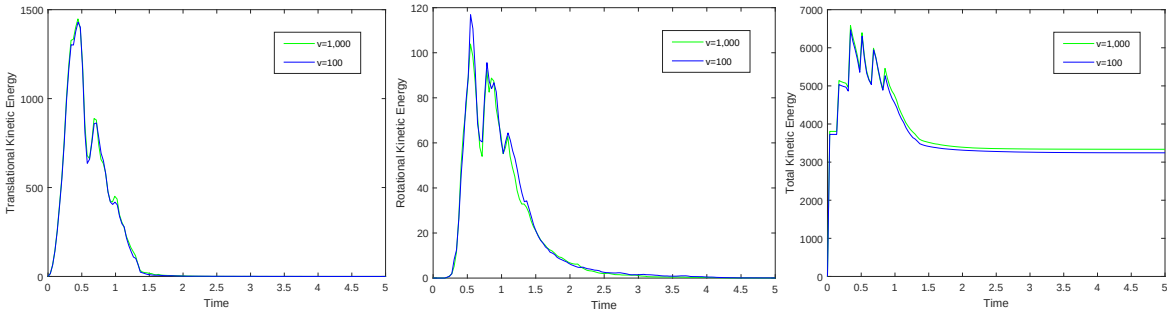
bahaviour of the simplified meshes under more realistic modelling conditions.

This example considers settling four different types of star-shaped particles into a container. Each particle type is represented by two levels of simplified meshes as shown in Figure 11, where each mesh is simplified from an original mesh with 10,000 vertices.

Two simulations, each consisting of these four types of star-shaped particles but one simulation using 1000 vertices for every mesh while the other using 100 vertices, are conducted. Particles are randomly generated in six time instants and fall under gravity towards a container which is also surface triangulated. A coefficient of friction $\mu = 0.5$ is applied. The damping ratio ξ equivalent to a restitution coefficient of 0.1 is also applied to both normal and tangential contacts. The time step is calculated as 3.41×10^{-4}s, and $15,000$ steps are performed, which covers a total physical time of 5.115s.

In each simulation, 300 particles for each shape are generated, giving 1200 particles in total. Due to the randomness involved in the particle generation stage, the two simulations cannot have identical conditions, which makes a quantitative comparison between the two results difficult. Nevertheless, the evolutions of the three energies of the particle system in the two simulations are given in Figure 12. Note that the particle injection takes place in the time interval $t \in [0, 1]$s, and each injection causes an increase in the total energy as shown in Figure 12(c). After the injection is complete, both translational and rotational energies start to decrease due to the dissipative effect of friction and damping, and are eventually fully dissipated. The steady convergence of the energy curves also indicates the robustness and stability of the energy-conserving normal contact model used. The packing configurations at three time instants are illustrated in Figure 13.

The evolution curves of the three energies in Figure 12 clearly show good agreement between



(a) Translational (b) Rotational (c) Total

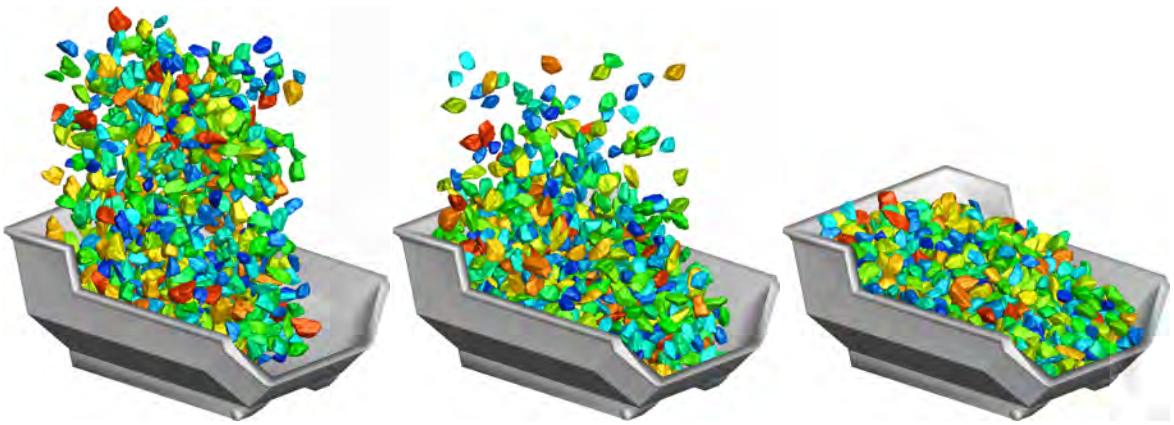
Figure 12: Time evolutions of translational (a), rotational (b) and total (c) kinetic energies of the two systems with friction $\mu = 0.5$ and damping ratio $\xi = 0.1$ for both normal and tangential contacts

the two simulations with different levels of mesh simplification. A small difference in the final total energy in Figure 12(c) exists, indicating that the two final packing configurations are slightly different, which can also be observed in Figure 13. This, however, should be expected and acceptable since the contact modelling of irregular particles is highly nonlinear and the numerical randomness plays a significant part. The general behaviour of the two systems are similar though.

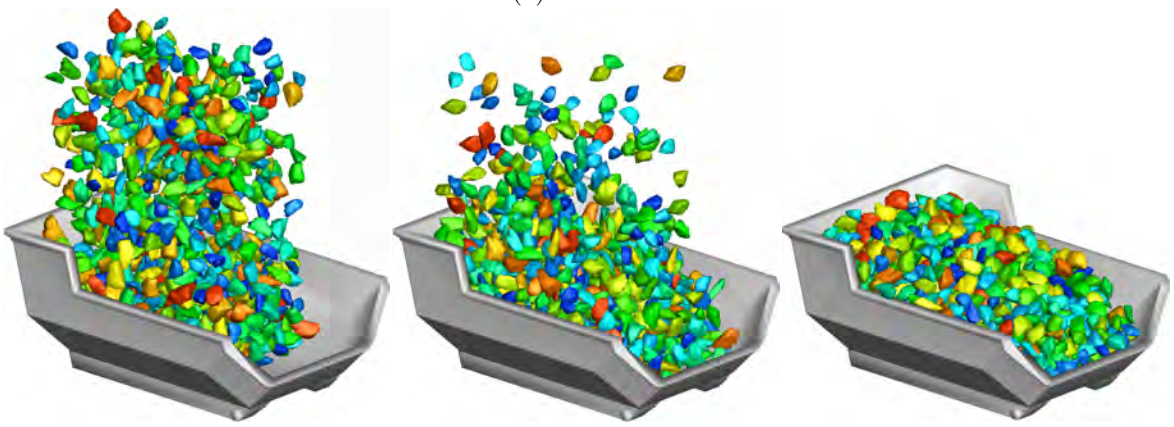
More importantly, the CPU speed-up of using the mesh of 100 vertices over using that of 1000 vertices is around 6.75. The main reason for achieving a speed-up less than 10 is that the computational complexity of some procedures involved in the DEM is not linearly proportional to the number of triangles, and some is only related to the number of particles which is the same in the two simulations.

6 Concluding Remarks

An effective contact modelling strategy for triangular mesh represented spherical harmonic particles has been proposed. This strategy features: 1) using a golden spiral lattice on the unit sphere to generate an initial triangular mesh with any number of vertices/triangles for a star-shaped surface; 2) applying the edge contraction based mesh simplification algorithm to reduce the mesh size to any desired level; and 3) adopting an energy-conserving linear normal contact model to compute the contact geometric and force features of two triangular mesh represented particles. In particular, the edge contraction algorithm is applicable to any triangular mesh. It is algorithmically very simple and highly effective, and can be easily implemented and incorporated into an existing discrete element software. The numerical experiments conducted in the DEM simulations for pair-wise impacts and a more realistic problem have quantitatively assessed the mechanical response error from simplified meshes, and have convincingly demonstrated that these meshes can not only have a very low geometric approximation error but also achieve a high mechanical modelling accuracy. Consequently, the edge contraction based mesh simplification approach can serve as an ideal pre-processor to optimise large triangular meshes for complex shaped particles in order to significantly reduce the computational cost in discrete element simulations.



(a) v = 1000



(b) v = 100

Figure 13: Settling 1,200 star-shaped particles into a container at three stages with two different simplified triangular meshes: (a) v = 1000; and (b) v = 100

Acknowledgement

The $a_{n,m}$ coefficients of the spherical harmonic shapes used in the current work are obtained from the following database created by Dr Garboczi at Materials and Structural Systems Division, National Institute of Standards and Technology, USA: *ftp.nist.gov/pub/bfrl/garbocz/Particle-shape-database/MA111-7/*. This public service is gratefully acknowledged.

This work is partially supported by National Natural Science Foundation of China under Grant Nos. 12072217 and 11772135. The support is acknowledged.

References

[1] P. A. Cundall, and O. D. L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47-65, 1979.

[2] S. Ji. and L. Lu. *Computational Granular Mechanics and Its Engineering Applications.* Springer, 2020. ISBN 978-981-15-3304-4.

[3] A. H. Barr. Superquadrics and Angle-Preserving Transformations. *IEEE Computer Graphics and Applications.* 1(1):11-23, 1981.

[4] J. R. Williams, A. Pentland. Superquadrics and model dynamics for discrete elements in interactive design. *Engineering Computations*. 9:115-128, 1992.

[5] C. Wellmann, C. Lillie, and P. Wriggers. A contact detection algorithm for superellipsoids based on the common-normal concept. *Engineering Computations* 25(5):432-442, 2008.

[6] J. Andrade, K. Lim, C. Avila, I. Vlahinic. Granular element method for computational particle mechanics. *Comput. Methods Appl. Mech. Engrg.* 241:262-274, 2012.

[7] T. J. Hughes, J. A. Cottrell, Y Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Engrg., 194(3941):41354195, 2005.

[8] W. Gao, J. Wang, S. Yin, Y. T. Feng. A coupled 3D isogeometric and discrete element approach for modeling interactions between structures and granular matters. *Comput. Methods Appl. Mech. Engrg.* 354:441-463, 2019.

[9] W. Gao, Y. T. Feng. A coupled 3D discrete elements/isogeometric method for particle/structure interaction problems. *Comp. Part. Mech.*, 2019. https://doi.org/10.1007/s40571-019-00267-8.

[10] D. H. Ballard, C. M. Brown. *Computer Vision.* Englewood Cliffs, NJ:Prentice-Hall; 1982.

[11] E. J. Garboczi, J. W. Bullard. Contact function, uniform-thickness shell volume, and convexity measure for 3D star-shaped random particles. *Powder Technology*, 237:191-201, 2013.

[12] Z. Zhu, H. Chen, W. Xu, L. Liu. Packing simulation of three-dimensional multi-sized star-shaped particles. *Modelling and Simulation in Materials Science and Engineering*, 22(3), article id. 035008, 2014.

[13] C. Brechbuhler, G. Gerig, O. Kubler. Parametrization of closed surfaces for 3D shape description. Computer Vision and Image Understanding, 61(2): 15470, 1995.

[14] 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):325-339, 2017.

[15] J. K. Anochie-Boateng, J. J. Komba, G. M. Mvelase. Three-dimensional laser scanning technique to quantify aggregate and ballast shape properties. *Construction and Building Materials.* 43:389-398, 2013.

[16] P. J. Schneider, D. H. Eberly *Geometric Tools for Computer Graphics.* San Mateo, CA:Morgan Kaufmann Publishers, 2003.

[17] Y. T. Feng. An energy-conserving contact theory for discrete element modelling of arbitrarily shaped particles: Basic framework and general contact model. *Comput. Methods Appl. Mech. Engrg.* 2021, 373:113454.

[18] Y. T. Feng. An energy-conserving contact theory for discrete element modelling of arbitrarily shaped discrete elements: Contact volume based model and computational issues. *Comput. Methods Appl. Mech. Engrg.* 2021, 373:113493.

[19] Y. T. Feng, D. R. J. Owen. A 2D polygon/polygon contact model: algorithmic aspects. *Engineering Computations*, 21:265-277, 2004.

[20] Y. T. Feng, K. Han, D. R. J. Owen. Energy-conserving contact interaction models for arbitrarily shaped discrete elements. *Comput. Methods Appl. Mech. Engrg.* 205-208: 169-177, 2012.

[21] M. Abramowitz, I. A. Stegun (Eds.). "Legendre Functions" and "Orthogonal Polynomials." Ch. 22 in Chs. 8 and 22 in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th printing. New York: Dover, pp. 331-339 and 771-802, 1972.

[22] J.H. Hannay, J. F. Nye. Fibonacci numerical integration on a sphere. *J. Phys. A: Math. General*. 37:1159111601, 2004.

[23] Y. T. Feng, K. Han, D. R. J. Owen. An accurate evaluation of geometric view factors for modelling radiative heat transfer in randomly packed beds of equally sized spheres. *International Journal of Heat and Mass Transfer*. 55:63746383, 2012.

[24] M. Roberts. How to evenly distribute points on a sphere more effectively than the canonical Fibonacci Lattice. http://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/, 09 June 2020 (last accessed: 07/09/2020)

[25] S. H. Lo. *Finite Element Mesh Generation.* CRC Press, Taylor & Francis Group, 2015.

[26] M. Garland, P. S. Heckbert. Surface simplification using quadric error metrics. *SIGGRAPH 97 Proceedings*, pp. 209216, August 1997.

[27] M. Garland, P. S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. *IEEE Visualization 98 Conference Proceedings*, pp. 263-269, October 1998.

[28] P. S. Heckbert, M. Garland. Optimal triangulation and quadric-based surface simplification. *Computational Geometry* 14:49-65, 1999.

[29] P. Lindstrom, G. Turk. Fast and memory efficient polygonal simplification. In Visualization 98 Proceedings (1998), IEEE. *IEEE Visualization 98 Conference Proceedings*, pp. 279-286, October 1998.

[30] H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. *IEEE Visualization 99 Conference Proceedings*, pp. 59-66, 1999.

[31] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *SIGGRAPH 97 Proceedings*, 26(2):6570, July 1992.

[32] D. R. Owen, Y. T. Feng, E. de Souza Neto, F. Wang, M. G. Cottrel, F. A. Pires, J. Yu. The modelling of multi-fracturing solids and particulate Media. *International Journal for Numerical Methods in Engineering*. 60(1): 317-340, 2004.

[33] A. Dullweber, B. Leimkuhlera and R. McLachlanb. Symplectic splitting methods for rigid body molecular dynamics. *J. Chem. Phys.* 107(15):5831-5851, 1997.

[34] Y. T. Feng. A generic energy-conserving discrete element modelling strategy for concave particles represented by surface triangular meshes. *International Journal of Numerical Methods for Engineering*. 2021; 1-17. https://doi:10.1002/nme.6633.