



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 373 (2021) 113493

**Computer methods
in applied
mechanics and
engineering**

www.elsevier.com/locate/cma

An energy-conserving contact theory for discrete element modelling of arbitrarily shaped particles: Contact volume based model and computational issues

Y.T. Feng

Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, UK

Received 7 July 2020; received in revised form 4 September 2020; accepted 6 October 2020

Available online 2 November 2020

Abstract

The contact volume based energy-conserving contact model is presented in the current paper as a specialised version of the general energy-conserving contact model established in the first paper of this series (Feng, 2020). It is based on the assumption that the contact energy potential is taken to be a function of the contact volume between two contacting bodies with arbitrary (convex and concave) shapes in both 2D and 3D cases. By choosing such a contact energy function, the full normal contact features can be determined without the need to introduce any additional assumptions/parameters. By further exploiting the geometric properties of the contact surfaces concerned, more effective integration schemes are developed to reduce the evaluation costs involved. When a linear contact energy function of the contact volume is adopted, a linear contact model is derived in which only the intersection between two contact shapes is needed, thereby substantially improving both efficiency and applicability of the proposed contact model. A comparison of this linear energy-conserving contact model with some existing models for discs and spheres further reveals the nature of the proposed model, and provides insights into how to appropriately choose the stiffness parameter included in the energy function. For general non-spherical shapes, mesh representations are required. The corresponding computational aspects are described when shapes are discretised into volumetric meshes, while new developments are presented and recommended for shapes that are represented by surface triangular meshes. Owing to its additive property of the contact geometric features involved, the proposed contact model can be conducted locally in parallel using GPU or GPGPU computing without occurring much communication overhead for shapes represented as either a volumetric or surface triangular mesh. A set of examples considering the elastic impact of two shapes are presented to verify the energy-conserving property of the proposed model for a wide range of concave shapes and contact scenarios, followed by examples involving large numbers of arbitrarily shaped particles to demonstrate the robustness and applicability for more complex and realistic problems.

© 2020 Elsevier B.V. All rights reserved.

Keywords: Concave shapes; Energy conservation; Contact volume based contact model; Volumetric mesh representation; Triangular mesh representation

1. Introduction

In [1], a generic energy conserving contact theory for arbitrarily shaped particles is comprehensively presented. The resulting general contact model outlines the procedure to fully determine the features of the normal contact

E-mail address: y.feng@swansea.ac.uk.

force between any two shapes, including the magnitude, normal direction and contact point/line. The detail of the model solely depends on what form of the contact energy function will be adopted. In this part, the contact energy function is taken to be a function of the volume of the contact region between the two contact bodies. The resulting contact model is termed the *contact volume based energy-conserving contact model*.

This special energy-conserving contact model is first proposed in [2] for polygons and used for super-quadrics in [3] then extended to polyhedra in [4,5]. The model is presented in a more general form in [6]. Applications of this model to achieve stable DEM simulations for non-spherical or irregular particles can be found, for instance, in [7] for 2D polygons, [8] for convex polyhedra, and in [9] for decomposable non-convex shapes. However, as the contact volume is involved either explicitly or implicitly, the associated computational cost can be high for other shapes, such as super-quadrics [10–12] and dilated particles [13–15]. This high computational cost can hinder the wider application of the model in DEM. In the current part of the series, this contact volume based model will be developed and presented in a more rigorous and concrete fashion. More features are exploited and new computational schemes are proposed to significantly improve the overall performance of the model. A similar idea of using the energy function for contact is also used in [16] for triangle/tetrahedron contacts in the context of the combined finite element/discrete element methodology.

All the important aspects of the model will be comprehensively addressed in the next section for general 3D shapes. In particular, it will be shown that when a linear contact energy function of the contact volume is adopted, only the intersection between two contact shapes, instead of their contact region and surfaces, is needed to fully determine the features of the normal contact force, thereby substantially improving both efficiency and applicability of the proposed contact model. A special version for 2D shapes will also be described. A comparison of this specialised energy-conserving contact model with some existing models for discs and spheres is also taken to further reveal the nature of the proposed model. This provides an insight into how to appropriately choose the stiffness parameter included in the energy function. Section 3 is devoted to the computational aspects of the model when applied to arbitrarily shaped 3D particles and particularly when shapes are discretised into volumetric meshes. Some significant new developments are presented in Section 4 for shapes represented by surface triangular meshes, resulting in a set of numerical operations only involving triangles. The additivity nature of the contact geometric features involved in the proposed contact model makes the contact model between two particles ideal to adopt parallel processing such as GPU or GPGPU computing for large scale industrial problems as already demonstrated in [9]. A set of examples considering the impact of two concave shapes is presented in Section 5 to validate the universal energy-conserving property of the proposed model for various contact scenarios. This is followed by examples involving large numbers of different concave shapes to demonstrate the robustness and applicability of the proposed model for more complex and realistic problems. Conclusions are drawn in Section 6.

Note that the same set of notations and conventions used in the first paper of this series [1] is adopted here.

2. Contact volume based energy-conserving contact model

Consider the following contact energy function

$$w = w(V_c) \quad (1)$$

where $V_c = |\Omega_c|$ is the volume of the contact region Ω_c between two contact bodies concerned. As the contact volume is independent of the global coordinate system, any monotonically increasing function of the contact volume satisfies both translational and rotational invariants as established in Section 3.4 of the first paper [1]. Thus such a function is a valid candidate for the contact energy function.

2.1. Contact volume based energy-conserving contact model for 3D shapes

Applying the particular form (1) of the contact energy to the general contact model developed in Section 3 of [1] leads to the specialised contact volume based normal contact model, which fully describes the features of a normal contact force including the normal direction, magnitude and contact point/line and also guarantees elastic energy conservation in any contact scenario. Referring to Fig. 1, the main aspects of the proposed contact model are summarised below.

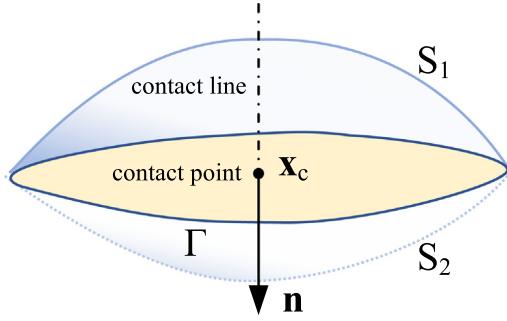


Fig. 1. Two arbitrarily shaped bodies Ω_1 and Ω_2 in contact and the contact region formed by two contact surfaces S_1 and S_2 .

Model 2 (Contact Volume Based Model). Let $w(V_c)$ be a monotonically increasing function of the contact volume V_c . Then the normal force \mathbf{F}_n exerted on Body 1 can be obtained as

$$\mathbf{F}_n = -\nabla_x w(V_c) = -w'(V_c) \mathbf{S}_n \quad (2)$$

where \mathbf{S}_n is the vector area of the contact surface S_1

$$\mathbf{S}_n = \int_{S_1} d\mathbf{S} \quad (3)$$

with the projected or scalar contact area $S_n = |\mathbf{S}_n|$. Then

(1) The unit normal contact vector \mathbf{n} is defined as the negative unit direction of \mathbf{S}_n

$$\mathbf{n} = -\mathbf{S}_n / S_n \quad (4)$$

(2) The magnitude F_n is identified as

$$F_n = w'(V_c) S_n \quad (5)$$

(3) The contact point c is determined by the coordinates \mathbf{x}_c

$$\mathbf{x}_c = \frac{\mathbf{n} \times \mathbf{G}_n}{S_n} + \lambda \mathbf{n}, \quad \text{with } \mathbf{G}_n = \int_{S_1} \mathbf{r} \times d\mathbf{S} \quad (6)$$

where λ is a free parameter, and its all possible values define the normal contact line.

The normal force \mathbf{F}'_n exerted on Body 2 by Body 1 is

$$\mathbf{F}'_n = -\mathbf{F}_n = -F_n \mathbf{n} \quad (7)$$

and can be applied to the same contact point (but in Body 2) or any point on the contact line.

In the following subsection, some key aspects of the model will be proved. Other important issues will be discussed in the subsequent subsections.

The key to the development of this contact volume based model is the derivation of the following relation

$$\nabla_{\mathbf{x}} V_c = \frac{\partial V_c}{\partial \mathbf{x}} = \int_{S_1} d\mathbf{S} \quad (8)$$

which can be readily obtained by applying the classic Leibniz integral rule with variable limits [17], or other similar approaches, such as the Reynolds transport theorem [18], to an integral with a moving boundary. The details are omitted here.

Next, we prove in a different way from [1] that \mathbf{F}_n and \mathbf{F}'_n are indeed a pair of action and reaction forces and that the contact moments at the contact point/line must be zero. The results will be utilised to derive more effective computational schemes in the next subsection.

Expression (2) for \mathbf{F}_n can be rewritten as

$$\mathbf{F}_n = - \int_{S_1} w'(V_c) d\mathbf{S} = - \int_{S_1} p_n \mathbf{n}_s dS \quad (9)$$

where \mathbf{n}_s is the outer normal of the surface S_1 , and $p_n = w'(V_c)$ is a constant. Similarly, \mathbf{F}'_n can also be expressed as

$$\mathbf{F}'_n = - \int_{S_2} w'(V_c) d\mathbf{S} = - \int_{S_2} p_n \mathbf{n}_s dS \quad (10)$$

It can be seen from (9) and (10), and first observed in [5], that the proposed contact volume based contact model is equivalent to defining a uniformly distributed normal contact pressure $\mathbf{p} = p_n \mathbf{n}$. This is similar to a hydrostatic pressure acting on the contact surface (i.e. both contact surfaces S_1 and S_2) with intensity p_n . \mathbf{F}_n and \mathbf{F}'_n are the resultant forces of the pressure on the two contact surfaces, respectively.

Similarly, the moments produced by this constant pressure p_n on surfaces S_1 and S_2 about the origin (or any point) can be expressed as

$$\mathbf{M}_n = - \int_{S_1} p_n \mathbf{r} \times \mathbf{n}_s dS; \quad \mathbf{M}'_n = - \int_{S_2} p_n \mathbf{r} \times \mathbf{n}_s dS \quad (11)$$

where \mathbf{r} is the position vector of a point on S_1 or S_2 .

The pair $(\mathbf{F}_n, \mathbf{M}_n)$ forms a force-couple system that can be replaced by a single \mathbf{F}_n acting at the contact point \mathbf{x}_c with the condition

$$\mathbf{F}_n \times \mathbf{x}_c = \mathbf{M}_n = - \int_{S_1} p_n \mathbf{r} \times \mathbf{n}_s dS = - p_n \mathbf{G}_n \quad (12)$$

which yields the solution for the contact point \mathbf{x}_c given in (6).

The summation of two expressions (9) and (10) gives the resultant force acting on the whole contact surface $S = S_1 + S_2$

$$\mathbf{F}_s = \mathbf{F}_n + \mathbf{F}'_n = - \int_{S_1} p_n \mathbf{n}_s dS - \int_{S_2} p_n \mathbf{n}_s dS = - \oint_S p_n \mathbf{n}_s dS = - p_n \oint_S d\mathbf{S} \quad (13)$$

while the moment of the pressure produced from the whole contact surface about the origin is

$$\mathbf{M}_o = \mathbf{M}_n + \mathbf{M}'_n = - \oint_S p_n \mathbf{r} \times \mathbf{n}_s dS = - p_n \oint_S \mathbf{r} \times d\mathbf{S} \quad (14)$$

It is well known in fluid mechanics that any solid object subjected to a uniform hydrostatic pressure on its entire boundary surface must be in equilibrium, i.e. $\mathbf{F}_s = \mathbf{M}_o = 0$. This can also be derived from the following two geometric relations: any closed surface S satisfies

$$\oint_S d\mathbf{S} = 0; \quad \oint_S \mathbf{r} \times d\mathbf{S} = 0 \quad (15)$$

These relations can be easily proved from the classic Gauss theorem (or the divergence theorem) and one of its variants [19].

Both (3) and (6) indicate that the contact surface S_1 (and equally S_2) solely determines the contact normal direction and the contact point or line, while the magnitude of the force may also be affected by the contact volume V_c . Thus how these contact geometric features associated with the contact region are evaluated will dictate the computational costs involved and may have a significant effect on the overall efficiency of a DEM simulation with non-spherical particles when employing the proposed contact model.

In order to compute \mathbf{S}_n and \mathbf{x}_c (or \mathbf{G}_n), it may be necessary to explicitly obtain the whole contact region Ω_c . This is clearly a challenging task to fulfil for most geometric shapes. Thus it makes the current contact model, despite offering superior numerical accuracy and stability, computationally expensive and less appealing than existing, mainly empirical-based contact models for non-spherical particles.

However, the computational issue mentioned above can be largely resolved. The following two subsections develop new schemes to effectively compute \mathbf{S}_n and \mathbf{G}_n based only on the intersection Γ between the two contact surfaces S_1 and S_2 and neither of the surfaces needs to be explicitly obtained. Consequently, the computational costs involved in evaluating \mathbf{n} and \mathbf{x}_c can be significantly reduced. In Section 2.4, the energy function $w(V_c)$ is assumed to be a linear function of V_c , then the explicit construction of the contact region Ω_c can be entirely avoided. Further implication of adopting this linear energy function is also discussed. These two developments will substantially improve the numerical efficiency of the proposed contact model.

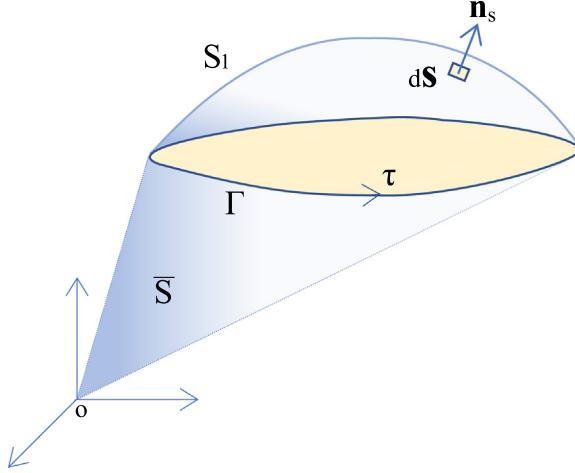


Fig. 2. The contact surface S_1 with the boundary Γ and one ‘conical’ replacement surface \bar{S} .

2.2. Evaluation of the normal contact and the contact centre

The evaluation of both the normal contact direction \mathbf{n} and the contact point \mathbf{x}_c requires the evaluation of the following two integrals over S_1

$$\mathbf{S}_n = \int_{S_1} d\mathbf{S}; \quad \mathbf{G}_n = \int_{S_1} \mathbf{r} \times d\mathbf{S}$$

The two identities in (15) lead to the following alternative expressions for \mathbf{S}_n and \mathbf{G}_n

$$\mathbf{S}_n = - \int_{S_2} d\mathbf{S}; \quad \mathbf{G}_n = - \int_{S_2} \mathbf{r} \times d\mathbf{S} \quad (16)$$

which implies that both \mathbf{S}_n and \mathbf{G}_n are *independent* of the actual shape of S_1 , provided that S_2 is fixed. The immediate consequence of this observation is that S_1 can be replaced by any other surface as long as they share the same boundary ∂S_1 (or ∂S_2).

Let $\Gamma = \partial S_1$ be the boundary of S_1 and oriented consistently with the outer normal of the surface as shown in Fig. 2. Γ is also the intersection between S_1 and S_2 . The boldfaced Γ , $\boldsymbol{\Gamma}$, is used to represent the oriented Γ below, and $-\boldsymbol{\Gamma}$ is the oriented boundary of S_2 . The orientation of $\boldsymbol{\Gamma}$, consistent with the outer normals of S_1 and S_2 , can be attained if its positive (tangential) direction τ at an intersection point is determined in the following way

$$\tau = \mathbf{n}_1 \times \mathbf{n}_2 \quad (17)$$

where \mathbf{n}_1 and \mathbf{n}_2 are the outer normals of the two surfaces S_1 and S_2 at the intersection point respectively.

2.2.1. Alternative evaluation schemes for \mathbf{S}_n

We now have a more general expression for \mathbf{S}_n : for any surface \bar{S} with the same oriented boundary $\boldsymbol{\Gamma}$

$$\mathbf{S}_n = \int_{\bar{S}} d\mathbf{S} \quad \forall \bar{S}: \partial \bar{S} = \boldsymbol{\Gamma} \quad (18)$$

This expression may lead to a more effective evaluation if a proper replacement surface \bar{S} can be chosen. In practical terms, an ideal replacement surface would be one that offers the best computational benefit in evaluating the above integral.

One possible candidate, as shown in Fig. 2, is a conical surface with the origin as its apex and $\boldsymbol{\Gamma}$ as its base or directrix. As this special surface is entirely determined by $\boldsymbol{\Gamma}$, this further implies that it is the intersection of the two contacting bodies $\boldsymbol{\Gamma}$ that dictates \mathbf{S}_n or the normal direction, rather than the contact region Ω or surfaces S_1 and S_2 . This assertion is fully supported by the following derivation which gives rise to another evaluation scheme for \mathbf{S}_n .

Stoke's theorem [19] relates the surface integral of a curl of a vector field \mathbf{F} over an open surface S to the line integral of \mathbf{F} over its consistently oriented boundary $\Gamma = \partial S$ by

$$\int_S \nabla \times \mathbf{F} \cdot d\mathbf{S} = \oint_{\Gamma} \mathbf{F} \cdot d\Gamma \quad (19)$$

If $\nabla \times \mathbf{F} = \mathbf{u}$ is a constant vector, then the left hand side of the above integral represents the projection of $\mathbf{S}_n = \int_S d\mathbf{S}$ onto \mathbf{u}

$$\int_S \nabla \times \mathbf{F} \cdot d\mathbf{S} = \mathbf{u} \cdot \mathbf{S}_n \quad (20)$$

Thus, if there exist three independent constant vectors $\mathbf{u}_i (i = 1, 2, 3)$, \mathbf{S}_n can be recovered. Obviously, the best choice of these three direction vectors will be \mathbf{i}, \mathbf{j} and \mathbf{k} , the unit vectors of the three coordinate axes. For a vector $\mathbf{F} = [F_x, F_y, F_z]^T$, its gradient is

$$\nabla \mathbf{F} = \left[\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}, \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}, \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right]^T$$

It is easy to check that the gradients of the following three vector fields $\mathbf{F}_i, \mathbf{F}_j$ and \mathbf{F}_k , as arranged in matrix form, are \mathbf{i}, \mathbf{j} and \mathbf{k} respectively

$$\mathbb{F} = [\mathbf{F}_i, \mathbf{F}_j, \mathbf{F}_k] = \frac{1}{2} \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} = \frac{1}{2} \mathbf{I} \times \mathbf{x} \quad (21)$$

For example,

$$\mathbf{F}_k = -y\mathbf{i} + x\mathbf{j} + 0\mathbf{k}, \quad \nabla \mathbf{F}_k = \frac{1}{2} \nabla(-y\mathbf{i} + x\mathbf{j} + 0\mathbf{k}) = \mathbf{k}$$

An arbitrary constant vector \mathbf{c} can be added to \mathbf{x} without altering \mathbb{F} in (21)

$$\mathbb{F} = \frac{1}{2} \mathbf{I} \times (\mathbf{x} + \mathbf{c}) \quad (22)$$

With the three vector fields $\mathbf{F}_i, \mathbf{F}_j$ and \mathbf{F}_k defined, \mathbf{S}_n can now be evaluated as a line integral over the boundary Γ

$$\mathbf{S}_n = \int_{S_1} d\mathbf{S} = \frac{1}{2} \oint_{\Gamma} \mathbb{F} \cdot d\Gamma = \frac{1}{2} \oint_{\Gamma} (\mathbf{I} \times \mathbf{x}) \cdot d\Gamma \quad (23)$$

By utilising the scalar triple product identity

$$\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$$

Eq. (23) is reduced to a simpler form

$$\mathbf{S}_n = \frac{1}{2} \oint_{\Gamma} \mathbf{x} \times d\Gamma = \frac{1}{2} \oint_{\Gamma} \mathbf{r} \times d\Gamma \quad (24)$$

where $\mathbf{r} = \mathbf{x}$ is the position vector of \mathbf{x} from the origin. Now \mathbf{S}_n can be viewed as half of the total moment produced by the intersection line Γ , treated as a line-distributed ‘force’, about the origin. Adding a non-zero constant vector \mathbf{c} to \mathbf{x} or \mathbf{r} will not alter the result as $\oint_{\Gamma} d\Gamma = 0$, but is equivalent to choosing the point $-\mathbf{c}$ as the new ‘moment’ centre.

Eq. (24) not only proves that \mathbf{S}_n is solely determined by the intersection of the two contacting bodies, but also provides an alternative and more efficient procedure to evaluate \mathbf{S}_n . The original surface integral over the contact surface S_1 is now replaced by a line integral over the intersection Γ .

Both (18) and (24) offer two alternative evaluation schemes for \mathbf{S}_n . When the conical surface \tilde{S} shown in Fig. 2 is used as the replacement surface, the two schemes are identical. Further details will be provided in Section 4 for the case where both bodies are triangulated and thus the intersection Γ is also discretised into a polyline.

2.2.2. Alternative evaluation schemes for the contact point

The key to determining the contact point \mathbf{x}_c by Eq. (6) is to evaluate the surface integral \mathbf{G}_n over S_1

$$\mathbf{G}_n = \int_{S_1} \mathbf{r} \times d\mathbf{S}$$

Following the preceding discussion for \mathbf{S}_n , a similar alternative scheme for evaluating \mathbf{G}_n is to use a *computationally* simpler surface \bar{S} to replace S_1

$$\mathbf{G}_n = \int_{\bar{S}} \mathbf{r} \times d\mathbf{S} \quad (25)$$

Again, the special conical surface \bar{S} used for computing \mathbf{S}_n is an ideal option.

The second alternative scheme can also be developed as follows. Consider another identity from Stoke's theorem over a surface S with the correctly oriented boundary Γ : for a scalar function f defined on S , it has

$$\int_S \nabla f \times d\mathbf{S} = - \oint_{\Gamma} f d\Gamma \quad (26)$$

If the function f has the form

$$f = \frac{1}{3} \mathbf{r} \cdot \mathbf{r} = \frac{1}{3} \|\mathbf{r}\|^2$$

it can be shown that

$$\nabla f = \mathbf{r}$$

Then

$$\int_S \nabla f \times d\mathbf{S} = \int_S \mathbf{r} \times d\mathbf{S} = - \frac{1}{3} \oint_{\Gamma} \mathbf{r} \cdot \mathbf{r} d\Gamma \quad (27)$$

Directly applying this formula to \mathbf{G}_n leads to the second alternative evaluation scheme

$$\mathbf{G}_n = \int_{S_1} \mathbf{r} \times d\mathbf{S} = - \frac{1}{3} \oint_{\Gamma} \mathbf{r} \cdot \mathbf{r} d\Gamma \quad (28)$$

In addition, if the boundary Γ can be parameterised by a scalar t , the above expression can be further reduced to a definitive integral over an interval

$$\mathbf{G}_n = - \frac{1}{3} \int_{t_0}^{t_1} \mathbf{r}(t) \cdot \mathbf{r}(t) dt \quad (29)$$

where $\mathbf{r}(t_0) = \mathbf{r}(t_1)$ are the starting and ending points of the boundary.

Eq. (28) reduces the evaluation of \mathbf{G}_n from the original surface integral over the contact surface S_1 to a line integral over the contact intersection Γ , while Eq. (29) further reduces it down to an integral over a parameter interval. Similar conclusions from the \mathbf{S}_n case can also be drawn in this case.

Note that the above proposed alternative schemes for evaluating \mathbf{S}_n and \mathbf{G}_n have reduced the associated computation cost to an insignificant level in comparison with finding the intersection Γ . The detailed numerical implementation of these schemes for triangulated shapes will be provided in Section 4.

2.3. Minimal contact surface and unique contact point

In the current model, $S_n = |\mathbf{S}_n|$ is taken as the contact area which can be readily used to model heat conduction between non-spherical particles. The nature of DEM makes it impossible to know the actual contact surface. Thus, S_n can be treated as an approximation to the actual contact area.

However, the contact point cannot be uniquely determined. In theory, any point along the contact line can be taken as the contact point without affecting the motion of the particles in the normal direction. Thus, it may be helpful if an additional condition can be imposed so that a unique contact point can be determined.

Here we introduce the concept of a minimal surface for a given boundary. Finding the minimal surface for a given boundary with some specified constraints is a mathematical problem [20] in the field of calculus of variations.

In the current case, for an open surface S with boundary Γ , we may define its (scalar) surface area S_a as

$$S_a = \int_S dS; \quad \partial S = \Gamma \quad (30)$$

The minimal contact surface S_m associated with the boundary Γ is characterised as the surface with the minimal surface area S_a . The problem can be described in a few equivalent ways in other fields. In geometry, a minimal surface is defined as the surface with zero mean curvature, and satisfies a partial differential equation called Lagrange's equation [21]. In architecture or structural engineering, a minimal surface is the shape of the soap film that naturally forms around the boundary as the wire frame.

If the minimal contact surface for Γ exists, the contact point can be taken as the intersection point of this surface with the normal contact line. Thus in order to find the unique contact point, we need to explicitly construct the minimal surface. However, solving this minimal surface problem for any given boundary is not trivial, and thus will not be attempted in this work. Nevertheless, when Γ is discretised as a polyline and further conditions are imposed, the problem can be significantly simplified and a simple analytical solution can be found. The details will be presented in Section 4 where triangular meshes are adopted to represent particles and the intersection boundary is a polyline.

2.4. Contact force evaluation and linear contact energy function

The main development in Section 2.2 can be summarised as follows: for the evaluation of \mathbf{S}_n and the contact point/line, the contact region or surfaces are no longer needed, and only the intersection of the two contact surfaces Γ is required. This results in a significant simplification over the original need of explicitly constructing contact surface S_1 or S_2 .

The remaining issue to be addressed is evaluating the magnitude of the normal force \mathbf{F}_n , F_n . By definition

$$F_n = w'(V_c)S_n \quad (31)$$

Thus F_n is proportional to the (scalar) contact area S_n and also to $w'(V_c)$. The actual form of $w(V_c)$ therefore needs to be specified. Typically, $w(V_c)$ may take the following power-law form

$$w(V_c) = k_n V_c^m \quad (32)$$

where k_n may be considered as a normal stiffness (the unit is different though); the exponent $m \geq 1$. Then F_n has the form

$$F_n = m k_n V_c^{m-1} S_n \quad (33)$$

In addition to k_n , the contact volume V_c , S_n and the exponent m also affect F_n . Note that the commonly used contact overlap is not presented. S_n may be interpolated as the measure of the contact area (in 3D) or width (in 2D), and thus can be useful when, for instance, heat conduction between contacting bodies is modelled.

In general, as F_n depends on the contact volume V_c , the contact region Ω_c may have to be explicitly obtained and its volume computed. The advantage gained from the new evaluation schemes developed in Section 2.2, which only use the intersection Γ for \mathbf{S}_n and \mathbf{G}_n , may therefore become much less significant.

However, when $m = 1$, i.e. w is a linear function of V_c , F_n becomes

$$F_n = k_n S_n \quad (34)$$

Now, F_n is only proportional to S_n which can be obtained solely from Γ and crucially, V_c disappears. Hence, neither the contact region Ω_c nor the contact surface S_1 or S_2 are involved in \mathbf{F}_n , and only the intersection Γ between the two contacting bodies needs to be explicitly constructed. This feature, together with the alternative schemes (24) and (28) for the contact normal and the contact point/line, gives a much simpler and effective normal contact model than those derived from other forms of the energy function. This improvement can also simplify the data structure to be used in the implementation. This is specially true when particles are represented by triangular meshes as discussed in Section 4.

Consequently, this particular contact model, referred to as the *contact volume-based linear model* or the *linear contact model* for short below, is recommended as the default model within the contact volume based modelling framework.

Note that in addition to the proposed contact model which is generally non-linear, the above recommended linear contact model is also different from the conventional overlap based linear spring model. Thus there are implications of the proposed both linear and nonlinear models on the critical time step associated with the time integration and on energy dissipation when the standard linear viscous damping model is used. The related issues will not be discussed in this work but will be addressed elsewhere.

2.5. Additivity of the contact geometric features in the contact volume based model

In the general form of the contact volume based model, the quantities that need to be evaluated include the contact volume V_c , \mathbf{S}_n and \mathbf{G}_n . These quantities are all integrals of geometric domains (regions, surfaces and lines), and are therefore naturally additive. So, when the original domains of the two bodies are divided into sub-domains, the required values can be obtained by summing all the sub-domain values. This additive property makes the current contact model flexible and highly parallelisable.

When the original shapes in contact need to be decomposed into sub-domains, V_c , \mathbf{S}_n and \mathbf{G}_n are evaluated distributively. In the first case, the shapes are composed of individual geometric entities, and naturally, V_c , \mathbf{S}_n and \mathbf{G}_n can be evaluated locally for sub-contact domains between individual sub-entities.

In the second case, the shape in contact may be geometrically complicated. For instance, it may be a triangulated surface mesh with many triangles, which could represent either a scanned realistic 3D object or a large complicated wall surface. The computational cost involved in computing V_c , \mathbf{S}_n and \mathbf{G}_n for two such shapes can be significant and therefore it may be desirable to reduce the cost through parallel computing such as GPU computing.

The additivity of the contact geometric features V_c , \mathbf{S}_n and \mathbf{G}_n in the current contact model ensures that the distributed computing for V_c , \mathbf{S}_n and \mathbf{G}_n can be correctly undertaken, sequentially or in parallel, in a straightforward way. For the contact model without using a linear energy function, the magnitude of the total normal force \mathbf{F}_n needs to be updated after the total V_c is computed. However, this step is omitted for the linear contact model where V_c is not required. The distributed computing of \mathbf{S}_n and \mathbf{G}_n is discussed below.

Suppose that the two bodies Ω_1 and Ω_2 are decomposed or divided into sub-domains Ω_{ik} :

$$\Omega_i = \sum_k \Omega_{ik} \quad (i = 1, 2) \quad (35)$$

Denote the overlapping region of two sub-domains Ω_{1i} and Ω_{2j} , one from each body, as Ω_{cij}

$$\Omega_{cij} = \Omega_{1i} \cap \Omega_{2j} \quad (36)$$

If $\Omega_{cij} \neq \emptyset$, two associated sub-contact surfaces \mathbf{S}_{1ij} and \mathbf{S}_{2ij} and the corresponding properly oriented sub-intersection Γ_{ij} can be obtained. Then the original intersection Γ can be pieced together from all the sub-intersections if needed:

$$\Gamma = \sum_{ij} \Gamma_{ij} \quad (37)$$

Utilising (24) and (28), we have

$$\mathbf{S}_n = \frac{1}{2} \int_{\Gamma} \mathbf{r} \times d\Gamma = \sum_{ij} \mathbf{S}_{1ij}; \quad \mathbf{S}_{1ij} = \sum_{ij} \int_{\Gamma_{ij}} \mathbf{r} \times d\Gamma \quad (38)$$

and

$$\mathbf{G}_n = \oint_{\Gamma} \mathbf{r} \cdot \mathbf{r} d\Gamma = \sum_{ij} \mathbf{G}_{nij}; \quad \mathbf{G}_{nij} = \int_{\Gamma_{ij}} \mathbf{r} \cdot \mathbf{r} d\Gamma \quad (39)$$

The intersection Γ may consist of multiple independent closed curves for complex contact shapes. Each closed curve may represent the outer boundary of a solid or void contact surface. Fig. 3 demonstrates a case with three closed boundaries forming a solid contact area with two inner holes on a 3D plane. The nature of the enclosed area, solid or hole, by a closed boundary curve can be identified from the boundary orientation in relation to the contact normal direction \mathbf{n} . However, the additive property of the model makes this identification unnecessary.

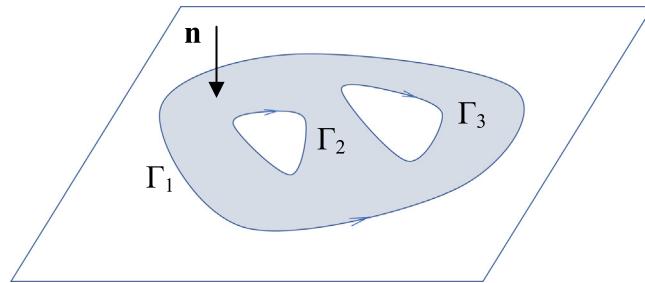


Fig. 3. A multi-boundary case: a solid contact area (in grey) and two voids formed by three disjoint boundaries Γ_1 , Γ_2 and Γ_3 , with an indicative contact normal direction \mathbf{n} .

Each closed boundary is treated equally and the orientation or the nature of the enclosed area is automatically taken into account in the definitions of \mathbf{S}_n and \mathbf{G}_n , as implied in both (38) and (39).

In Fig. 3, let \mathbf{S}_n^i ($i = 1, 2, 3$) each be \mathbf{S}_n of the three boundaries Γ_i ($i = 1, 2, 3$) respectively. \mathbf{S}_n^1 is the \mathbf{S}_n for the whole area enclosed by Γ_1 only, while \mathbf{S}_n^2 and \mathbf{S}_n^3 are those of the two holes formed by Γ_2 and Γ_3 respectively. The opposite directions of both Γ_2 and Γ_3 relative to Γ_1 ensure that the two vectors \mathbf{S}_n^1 and \mathbf{S}_n^2 are opposite to \mathbf{S}_n^1 . Thus, the resulting $\mathbf{S}_n = \mathbf{S}_n^1 + \mathbf{S}_n^2 + \mathbf{S}_n^3$ will be the same if only the solid (grey) area is considered. The same argument also applies to \mathbf{G}_n .

Formulae (38) and (39) also suggest two slightly different approaches for evaluating \mathbf{S}_n and \mathbf{G}_n . In the first approach, individual sub-intersections are first collected centrally; then (24) and (28) are used to obtain \mathbf{S}_n and \mathbf{G}_n . In the second approach, \mathbf{S}_n and \mathbf{G}_n in each sub-contact domain are evaluated locally; then their values are collected from all the sub-contact domains and added together centrally to obtain the overall values.

Using parallel computing to evaluate \mathbf{S}_n and \mathbf{G}_n , the second approach is preferable as \mathbf{S}_n and \mathbf{G}_n can be totally distributed to individual sub-contact domains. All the evaluations are performed locally and are fully independent, so the whole procedure can be perfectly parallelised without any cross-partition data exchange and communication overhead. Data transfer only takes place at the last step to sum the contributions from all sub-contact domains to recover the total values of \mathbf{S}_n and \mathbf{G}_n . However, this step may be unnecessary. In fact each sub-domain contact can be viewed as an individual contact whose contact features are obtained independently and applied to the contacting bodies directly, depending on the coding strategy. Another benefit of this approach is that the contact between the two bodies can now be treated as a distributed contact and consequently, distributed damping and frictional forces can be applied locally across the contact area. In cases where the contact may take place over a large area, this treatment may be more appropriate when considering damping and friction effects.

As mentioned earlier, the actual computational cost involved in evaluating \mathbf{S}_n and \mathbf{G}_n is relatively insignificant. The major computation occurs in finding all sub-domain intersections of the two shapes. Again, this step can be done fully in parallel, provided there is a pre-processing scheme available which can effectively exclude cases when two sub-domains do not intersect. A fully parallel but highly ineffective scheme is to check the intersection for all sub-domain combinations.

2.6. The area based contact model for 2D shapes

The contact volume based algorithm for any 3D shapes has been fully established in the previous section. The new schemes have also been proposed for improving the computational efficiency of the associated algorithm. The model can be easily reduced to 2D cases, leading to a *contact area based normal contact model* for any 2D shapes.

2.6.1. Contact geometry

The notations used for 3D shapes will be adopted, but some of the definitions are changed. V_c is now the contact area A_c ; S_1 and S_2 are two open boundaries instead of surfaces; the intersection Γ is changed from a boundary line to individual points. As shown in Fig. 4, there are only two intersection points (\mathbf{a}, \mathbf{b}) in case (a), while in a more complicated contact scenario (b), multiple pairs of intersections, $(\mathbf{a}_i, \mathbf{b}_i)$ ($i = 1, \dots, m$), may occur. For each pair, \mathbf{a} is classified as the starting point and \mathbf{b} the end point, with the following conditions

$$\mathbf{n}_1(\mathbf{a}) \times \mathbf{n}_2(\mathbf{a}) \cdot \mathbf{k} < 0; \quad \mathbf{n}_1(\mathbf{b}) \times \mathbf{n}_2(\mathbf{b}) \cdot \mathbf{k} > 0 \quad (40)$$

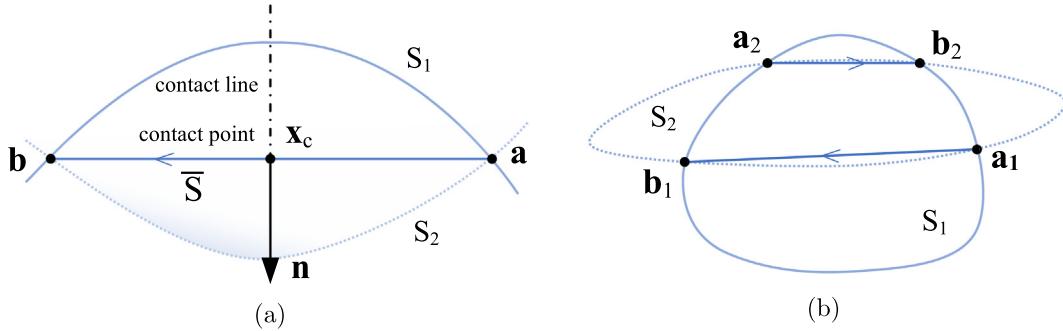


Fig. 4. 2D contact cases: (a) a simple contact with two intersection points; and (b) a complex contact with four contact points.

where $\mathbf{n}_1(\mathbf{x})$ and $\mathbf{n}_2(\mathbf{x})$ are the outer normals of S_1 and S_2 at the intersection point \mathbf{x} respectively. This condition is compatible with the definition (17) for the positive direction of Γ in 3D.

2.6.2. Contact features

The normal force \mathbf{F}_n still takes the same form as in (2), but \mathbf{S}_n becomes a line integral over the contact boundary. Furthermore, the previously developed alternative schemes for \mathbf{S}_n based on the intersection line is no longer valid, while the scheme using a simple contact boundary \bar{S} to replace S_1 is still valid and becomes even simpler: \bar{S} shares the same interaction points as S_1 .

For the two-intersection point case in Fig. 4(a), the line segment linking the two points (\mathbf{a}, \mathbf{b}) is the simplest \bar{S} , and \mathbf{S}_n can be explicitly written as

$$\mathbf{S}_n = \int_{\bar{S}} d\mathbf{S} = -w_c \mathbf{n} \quad (41)$$

where $w_c = |\mathbf{b} - \mathbf{a}|$ is the distance between the two intersection points and is therefore termed the *contact width*; and \mathbf{n} is the normal contact direction perpendicular to the line segment $\mathbf{b} - \mathbf{a}$. Thus the link direction $\mathbf{b} - \mathbf{a}$ is also the tangential direction.

By extending the definition of \mathbf{G}_n to the line segment \bar{S} and setting the origin as the reference point or moment centre (so $\mathbf{r} = \mathbf{x}$), it can be shown that

$$\mathbf{G}_n = \int_{\bar{S}} \mathbf{r} \times d\mathbf{S} = \frac{\mathbf{a} + \mathbf{b}}{2} \times (w_c \mathbf{n}) \quad (42)$$

Based on (6) and taking $\lambda = 0$, the contact point can be obtained as

$$\mathbf{x}_c = \frac{\mathbf{a} + \mathbf{b}}{2} \quad (43)$$

which is the middle point of the two intersection points.

By definition, the magnitude of the normal force is

$$F_n = w'(A_c)w_c \quad (44)$$

For a linear energy function $w = k_n A_c$, it becomes

$$F_n = k_n w_c \quad (45)$$

i.e. the contact width w_c , instead of the traditional contact overlap, primarily determines the force magnitude.

In summary, for a 2D contact with two intersection points, these two points determine all the contact geometric features: the distance is the contact width, the link defines the tangential direction, the normal direction is perpendicular to this link, and the middle point can be taken as the contact point.

For multiple intersection points with the linear contact energy function, each pair of the intersection points can be treated as an independent contact using the above procedure. If an overall contact is preferred, \mathbf{S}_n and \mathbf{G}_n can

be computed as

$$\mathbf{S}_n = \sum_i w_{c_i} \mathbf{n}_i; \quad \mathbf{G}_n = \sum_i w_{c_i} \frac{\mathbf{a}_i + \mathbf{b}_i}{2} \times \mathbf{n}_i \quad (46)$$

from which the normal direction \mathbf{n} and the contact point \mathbf{x}_c can be obtained following (4) and (6).

This energy conserving contact model for 2D shapes is first proposed in [2] for polygons and extended to 2D super-quadratics in [3], while the model is directly degenerated here from the general 3D contact model as a special case and the derivation involved is more straightforward. It is highlighted that the current model can be applied to the contact between any 2D convex or concave shapes. For the linear contact model, only the intersection points between two shapes need to be found, and the computational cost involved is comparable with some existing models, e.g. for ellipses and super-quadratics.

2.7. Comparison with standard disc/sphere contact models

The proposed contact volume based model is different from most conventional models currently employed in DEM. In its simplest form when the linear contact energy function is used, all the contact features depend on the intersection Γ of the two shapes. An analytical expression for the intersection is only available for discs and spheres. Other shapes, except for polygons (in 2D) or polyhedra (3D), may need to be discretised into polygons or polyhedra, so their contact can be treated as the contact between polygon/polygon or polyhedron/polyhedron. Computational issues of the volume based contact model for discretised non-spherical shapes will be comprehensively described in the next two sections.

The existence of an analytical solution for a pair of discs or spheres can be used to compare the current model with conventional models in order to gain an insight into the nature and differences of the model with the others.

Consider the contact of two discs/spheres with radii r_1 and r_2 and an overlap δ . Let $d = r_1 + r_2 - \delta$, $\Delta r = r_1 - r_2$. Then w_c can be derived as

$$w_c^2 = 2 \left[(r_1 + r_2) - \frac{\Delta r^2}{d} \right] \delta - \left[1 + \left(\frac{\Delta r}{d} \right)^2 \right] \delta^2 \quad (47)$$

Thus for discs, the normal force magnitude F_n is equal to

$$F_n = k_n w_c = k_n \sqrt{2 \left[(r_1 + r_2) - \frac{\Delta r^2}{d} \right] \delta - \left[1 + \left(\frac{\Delta r}{d} \right)^2 \right] \delta^2} \quad (48)$$

For spheres, the contact intersection is a circle with w_c as the diameter. Thus $S_n = \pi w_c^2 / 4$, and the magnitude F_n is

$$F_n = k_n S_n = k_n \frac{\pi}{2} \left\{ \left[(r_1 + r_2) - \frac{\Delta r^2}{d} \right] \delta - \frac{1}{2} \left[1 + \left(\frac{\Delta r}{d} \right)^2 \right] \delta^2 \right\} \quad (49)$$

To understand the nature of the two formulae (48) and (49), set $r_1 = r_2 = r$. Also recall that the actual contact radius in the Hertzian theory is equal to $a = \sqrt{r\delta}$. Thus the two formulae are reduced to

$$F_n = 2k_n \sqrt{r\delta} = 2k_n a \propto \sqrt{\delta} \quad (\text{disc}) \quad (50)$$

$$F_n = k_n \pi \left(r\delta - \frac{\delta^2}{4} \right) = k_n \pi \left(a^2 - \frac{\delta^2}{4} \right) \propto \delta \quad (\text{sphere}) \quad (51)$$

which indicate that the force for discs is proportional to the equivalent contact radius a or to the square root of the overlap δ ; while for spheres, the force is proportional to the actual contact area πa^2 or linear to δ if ignoring the higher order δ^2 term for a small overlap. For comparison, the Hertzian contact force for two monosized and linearly elastic spheres with Young's modulus E and zero Poisson's ratio is

$$F_n = \frac{4E}{3} a \delta \propto \delta^{3/2} \quad (52)$$

Clearly, the current model for spheres is very close to a linear spring model. It may not be accurate for problems where the Hertzian contact model is more appropriate, but for many practical problems where particle shapes are

non-spherical and the contact interaction between particles is far from the Hertzian contact, a linear spring model is often adopted. In these cases, the current model should produce similar results.

The equivalence of the current model to a standard linear spring model for spheres for a small overlap may also suggest a practical way to properly choose the spring stiffness in the current linear model for general 3D shapes. Eq. (51) indicates that the spring stiffness may be taken as $k_n/\pi r$, where k_n is the value used in a conventional linear spring model, and r can be considered as the (equivalent) characteristic length or radius of curvature of the two shapes at the contact zone.

3. Contact modelling of non-spherical particles with volumetric discretisation

The intersection Γ plays a pivotal role in the current contact model, as has been established in the previous sections. However, an analytical expression for Γ only exists for spheres. Thus some approximation needs to be introduced when dealing with non-spherical shapes using the proposed contact model.

For smooth shapes described by mathematical equations, such as cylinders, capsules, ellipsoids and super-quadratics, a partial solution is to construct the intersection Γ in a discrete manner. First, find any intersection point by simultaneously solving the two equations representing the two surfaces. Then trace the intersection from this point as the initial position in an incremental way until the whole intersection is completely covered. The resulting intersection is a polyline from which the contact feature can be computed. The next section will discuss the procedure in detail. Note that the above approach is not ideal as the computational cost involved is much higher than currently used overlap based methods, see for instance [22] for cylinders, [11] for super-quadratics, [14,15] for dilated polyhedra and [23] for poly-super-ellipsoids.

An alternative solution is to discretise a non-spherical shape and then to develop specialised contact volume based models to deal with the contact between two meshed shapes. There are mainly two types of mesh available: volumetric mesh and surface triangular mesh. The computational procedures for the associated contact models are different, resulting in different performance and applicability.

In this section, 3D shapes are assumed to be discretised into volumetric meshes, particularly tetrahedral meshes. The resulting contact model is termed the *volumetric mesh based normal contact model*. The next section will focus on the development of the computational procedure for handling the contact between surface triangulated shapes and the corresponding contact model is termed the *surface triangulation based normal contact model*.

3.1. The contact modelling of convex polyhedra

First consider the contact between two convex polyhedra using the proposed contact volume based model. A polyhedron can be geometrically described in different ways. A common approach is to define the position of a polyhedron by its vertex coordinates, and its topology by face connectivities in terms of vertex index. There are some variations differed by additional data, such as edges and face normals. A convex polyhedron can also be fully defined by a number of half-spaces. Each face of the polyhedron can be represented by a bounding half-space in the form of a linear inequality, and the polyhedron is the intersection or union of the half-spaces.

Combinations of the above two representations gives rise to three different computational techniques aiming at explicitly constructing the contact region. These three techniques are described below.

(1). Half-space cutting

Let the first polyhedron, P_1 , with n_f polygonal faces be represented by a set of n_f linear inequations

$$(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{n}_i \leq 0 \quad (i = 1, \dots, n_f) \quad (53)$$

where \mathbf{n}_i is the outer normal vector of the i th face; \mathbf{x}_i is any point on the face (normally one of its vertices). Let the second polyhedron, P_2 , be represented in the same way by its vertices and all edge and face connectivities.

The half-space cutting technique is to repeatedly find the intersection of each half-space of P_1 with P_2 and update P_2 as the resulting intersection. If the contact exists between P_1 and P_2 , the final updated P_2 is the contact region P_c , which is also a convex polyhedron. This technique is equivalent to using all the oriented planes of the faces of P_1 to cut P_2 . Each cutting may cut through some edges of P_2 to create a polygon as the new face and change or remove some existing vertices/edges/faces of P_2 to form an updated P_2 . The remaining of P_2 is P_c .

The new polygonal faces of P_c created by the consecutive cuttings of P_1 constitutes the contact surface S_1 . For the i th new face, compute its signed area A_i and the centroid \mathbf{c}_i , then \mathbf{S}_n and \mathbf{G}_n can be obtained:

$$\mathbf{S}_n = \sum_i \mathbf{A}_i; \quad \mathbf{G}_n = \sum_i \mathbf{c}_i \times \mathbf{A}_i \quad (54)$$

The corresponding contact features of \mathbf{F}_n can be fully determined as described by [Model 2](#). In addition to the computationally intensive part of determining all the intersections between the cutting plane and the edges for each cut, another time consuming part of this technique is some house-keeping operations required after each cut to maintain the correct topology of the updated P_2 . Note that a slightly different implementation is given in [8].

(2). 'Ray-tracing'

This alternative technique, proposed in [9], determines the intersection point between every edge of P_1 against every face of P_2 , and vice versa. These intersection points, if any, together with vertices of P_1 and P_2 that lie inside the other polyhedron are the vertices of P_c . Thus the convex-hull of these vertices gives P_c due to its convexity.

The key operations of this approach include: (1) Find the intersection of a line segment (edge) with a polygon (face) and repeat the operation for all edge–face combinations from both P_1 and P_2 ; (2) Identify those vertices from one polyhedron that are inside the other polyhedron; (3) Construct the convex-hull of the vertices obtained from the previous two steps; (4) Identify all triangles of the convex-hull that coincide with the faces of P_1 to form the contact surface S_1 ; (5) Compute the signed area and centroid of each triangle and use the formula (54) to compute \mathbf{S}_n and \mathbf{G}_n ; (6) Obtain \mathbf{F}_n and the contact point.

This ray-tracing technique avoids the constant topology update of the half-space cutting approach and replaces it by constructing the convex-hull of the vertices to attain P_c , but may more than double the computational cost in determining the vertices of P_c . It also needs additional computations to identify those triangles belonging to S_1 from the convex-hull. As a result, both techniques have a similar performance in general.

The total number of operations involved in the above two techniques for determining the vertices of P_c are roughly proportional to $n_{e_i} \times n_{f_j}$ ($i = 1, j = 2$ or $i = 2, j = 1$) where n_{e_i} and n_{f_i} are respectively the numbers of edges and faces of polyhedron i . Thus these techniques perform well for simple polyhedra, such as tetrahedra, but will be ineffective for polyhedra with large number vertices and faces.

(3). Dual-approach

This technique, proposed in [24], is based on the transformation of a convex polyhedron P into its dual polyhedron P^d , in which each face of P is mapped to a vertex of P^d and *vice versa*. Let one bounding half-space associated with a face of P be expressed as the following inequality in component form

$$ax + by + cz + d \leq 0 \quad (55)$$

where the vector (a, b, c) is the outer normal; and the constant d is assumed to be $d > 0$. Then the face is dual to the following vertex \mathbf{p}' of P^d in a dual space

$$\mathbf{p}' = (a/d, b/d, c/d) \quad (56)$$

Suppose that \mathbf{c} is a point inside the overlapping polyhedron P_c of P_1 and P_2 . Use \mathbf{c} as the origin of a coordinate system of a primal space, which ensures that the condition $d > 0$ is met for every face of P_1 and P_2 in this coordinate system. Then both P_1 and P_2 can be transformed to their duals P_1^d and P_2^d in the dual space. Further construct the convex hull, P_c^d , of all the vertices of both P_1^d and P_2^d . Now P_c^d is the dual of the contact polyhedron P_c . Thus P_c can be obtained by transforming P_c^d back to the primal space.

The key step in this dual approach is to find an initial point \mathbf{c} located inside P_c , and then update it to maintain this state in the subsequent steps as long as $P_c \neq \emptyset$. An iterative procedure [24] is available to achieve this. This approach has been adopted, for instance in [25], to explicitly construct P_c for modelling contacts between two convex polyhedra.

As the complexity of constructing a convex hull for n vertices is $O(n \log n)$, and the iterative procedure to locate a point in P_c has the same complexity, the overall complexity of this dual approach is therefore $O((n_{f_1} + n_{f_2}) \log(n_{f_1} + n_{f_2}))$, which is theoretically better than the first two approaches.

Nevertheless, the contact volume can be explicitly computed with either of the three approaches in case a nonlinear energy function of V_c is used. Note that a different approach based on the Minkowski difference is proposed in [26] for convex polyhedra, which can also ensure the energy-conserving property but is developed within the framework of using overlap based energy contact functions to derive energy-conserving normal contact models. This alternative framework will be covered in another paper of this series [27].

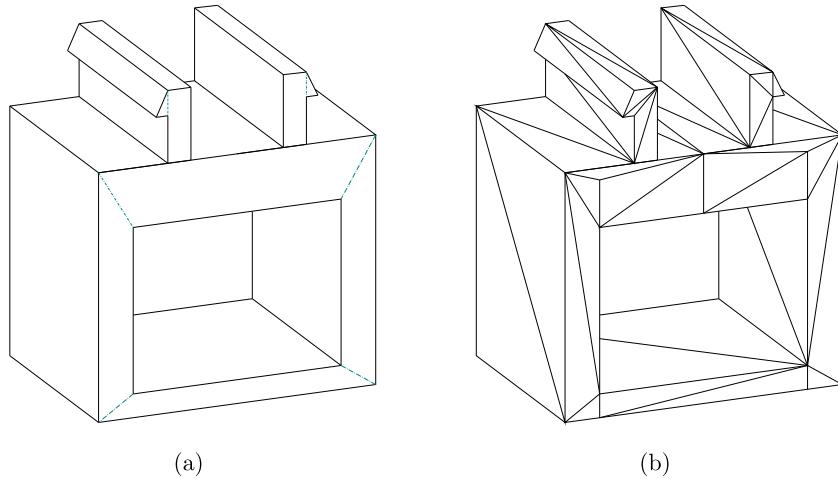


Fig. 5. A concave shape that can be manually decomposed into several convex “building blocks” (a); or discretised into a volumetric or surface mesh (b).

3.2. Contact modelling of volumetric mesh-based non-spherical particles

To apply the above three techniques to a non-spherical (convex or concave) particle, the shape needs to be decomposed or discretised into convex polyhedra. Then the contacts between individual polyhedra can be handled independently by using these three techniques. Each contact can be considered either as a single contact from which the normal contact force is computed and applied directly to the two associated particles at the contact point, or as part of the overall contact between the two particles.

If the linear contact model is employed, the contact volume V_c is not required. Then in the latter case mentioned above, only the contact surface S_1 from each contact needs to be obtained. All such surfaces are combined to form the overall contact surface S_1 , from which the required \mathbf{S}_n and \mathbf{G}_n can be evaluated. Thus the overall normal force \mathbf{F}_n and contact point \mathbf{x}_c for the contacting particle pair are computed.

Note that after a shape is decomposed/discretised into polyhedra, some faces of these polyhedra are not part of the original or exterior surface of the shape but are newly created from the interior of the shape. These are termed *interior* faces. For any face of a contact polyhedron obtained by the half-space cutting or ray tracing technique, if it belongs to an interior face of the first polyhedron, it will be cancelled when all contact surfaces are combined to form an overall contact surface S_1 . This is because there exists exactly one face from the other polyhedra that is the same but with the opposite normal. Taking advantage of this property can significantly reduce the computational cost associated with the half-space cutting technique by excluding all interior faces of a polyhedron from the cutting. However, this favourable property may not be effectively utilised in the ray-trace technique.

From the computational viewpoint, a non-spherical shape should be discretised into fewer convex polyhedra, each with a small number of faces. In case the shape is composed of identifiable convex blocks with a few flat surfaces, it should be decomposed into its “building blocks”. See such a shape in Fig. 5(a), which can be split into eight simple convex blocks along the dotted lines shown. However, this is not a common case in real applications, and very often the decomposition procedure may require a certain level of manual intervention, and therefore is not sufficiently generic.

A more universal approach is to discretise a shape into a volumetric mesh, and the most popular mesh type is tetrahedral. See Fig. 5(b) for instance. In this way, the contact between any two shapes reduces to multiple contacts between tetrahedra which can be tackled based on the half-space cutting technique, as outlined above.

Nevertheless, a tetrahedral mesh discretisation of a non-spherical shape may still not be the most convenient and effective representation. Any tetrahedron that is embedded inside the shape but deeper than the maximum possible contact overlap will have no contribution to the actual contact, and therefore is redundant. One way to reduce this type of redundant elements to achieve a better tetrahedral mesh is to tune some input parameters in the volumetric mesh generator so that all vertices of the generated mesh are located on the surface of the shape. Each surface

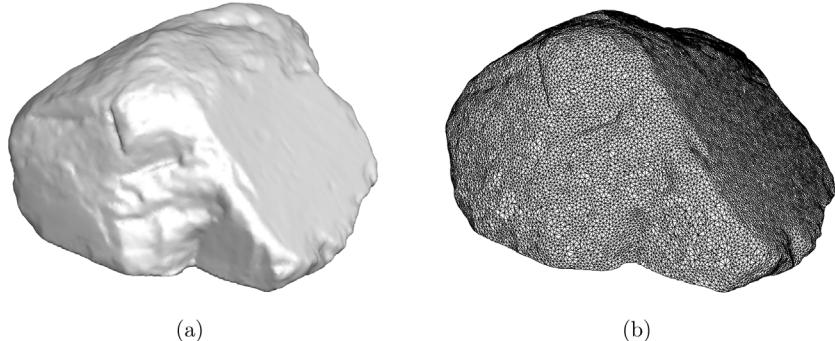


Fig. 6. (a) A scanned image of an ancient rock; and (b) its (coarsened) surface triangular mesh with 24,572 vertices and 49,140 triangles.

triangle of the shape belongs to one tetrahedron, thus the number of tetrahedra in the mesh cannot be smaller than that of the surface triangles. Then this naturally suggests that it could be more effective to only use the surface triangular mesh to represent a non-spherical shape. This idea gives rise to the surface triangulated non-spherical particle contact model, as will be fully exploited in the next section.

4. Surface triangulated non-spherical particle contact model

4.1. Triangular surface mesh representation

A triangular surface mesh is a universally applicable representation for any 3D shape and has been widely used as the default scheme for digitally representing geometric entities, including scanned objects. See Fig. 6 for a scanned image of an ancient rock and a (coarsened) surface triangular mesh with a very large number of vertices and triangles. There are many existing (CAD) toolkits available that can provide a triangular surface mesh for a 3D shape. Triangular meshes are most likely to be used to discretise any wall geometry that is not a simple flat surface in DEM simulations. Hence what to be presented below is equally applicable to a particle–wall contact scenario. Note that in a triangular mesh representing a 3D solid without holes, it can be deduced from Euler’s formula in algebraic topology that the numbers of vertices (v) and triangles (f) satisfy the relationship

$$v = f/2 + 2 \quad (57)$$

With the surface triangular mesh representation, modelling contacts between non-spherical particles becomes the handling of the contact between two triangular meshes as the core operation. Major issues to be addressed in this section include: (1) the construction of the intersection Γ between two triangular mesh described shapes; and (2) the effective evaluation of \mathbf{S}_n and \mathbf{G}_n to fully determine the normal contact force \mathbf{F}_n . Also the numerical procedure adopted to find the minimal contact surface under a simple condition and further to uniquely determine the contact point will be proposed.

4.2. Construction of intersection line segments

For two shapes with surfaces being discretised by triangular meshes, their intersection is a closed polyline Γ_h , which is a discretised version of Γ , as shown in Fig. 7. Its segments are formed by the intersection of individual triangles from both surfaces. The basic operation involved is to find the line segment of the intersection (if exists) between two given triangles from each surface, and this can be done in a straight forward manner. There are two intersection points for two triangles in contact. If two triangles lie on the same plane, there will be no contact. The start and end points of the segment should be properly identified to ensure a correct orientation. The positive direction τ of the segment, consistent with the positive orientation of Γ defined by (17), is given by

$$\tau = \mathbf{n}_1 \times \mathbf{n}_2 \quad (58)$$

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

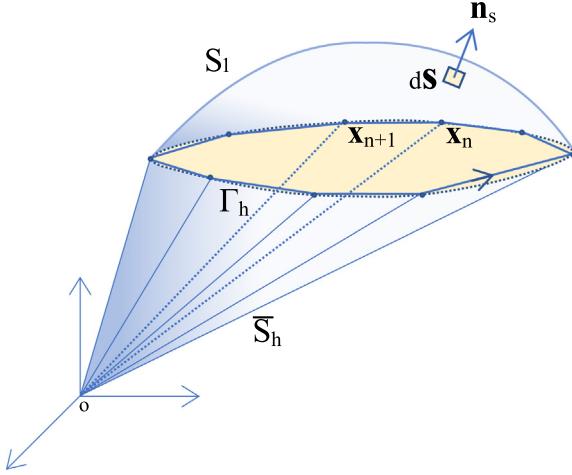


Fig. 7. The contact surface S_1 with the discretised boundary polyline Γ_h and also the discretised replacement surface \bar{S}_h .

All the segments obtained form a single or multiple closed polylines. Each closed polyline can be treated as a separate contact. Note, however, that it may not be necessary to explicitly construct the polyline(s) from these segments. This is because the additive nature of \mathbf{S}_n and \mathbf{G}_n ensures that their computations can be done in a segment by segment manner and in any order. Only when Γ_h consists of multiple polylines, and each closed polyline is considered to be a separate contact to compute the contact force, Γ_h should be pieced together from the segments in a right order.

Computations of \mathbf{S}_n and \mathbf{G}_n for the intersection segments are discussed in the next two subsections, in which for each line segment, a local parameterised form is used, as detailed below. Denote the i th line segment as $\Delta\Gamma_i$ with the start and end points being \mathbf{x}_i and \mathbf{x}_{i+1} , and set $\Delta\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$. With a local coordinate system on the segment with parameter $t \in [0, 1]$, the parametric equation and the infinitesimal increment of the segment are

$$\mathbf{x}(t) = \mathbf{x}_i + t \Delta\mathbf{x}_i; \quad d\Gamma = \Delta\mathbf{x}_i dt \quad (59)$$

4.3. Computation of \mathbf{S}_n

Section 2.2 proposes two schemes to evaluate \mathbf{S}_n as below

$$\mathbf{S}_n = \int_{\bar{S}} d\mathbf{S} = \frac{1}{2} \oint_{\Gamma} \mathbf{r} \times d\Gamma$$

where \bar{S} is a replacement surface of S , and \mathbf{r} is chosen to be $\mathbf{x} = [x, y, z]$.

The discretised version of the second scheme is

$$\mathbf{S}_n = \frac{1}{2} \sum_{i=1}^m \int_{\Delta\Gamma_i} \mathbf{x} \times d\Gamma = \sum_{i=1}^m \mathbf{S}_n^i \quad (60)$$

where m is the total number of segments; and \mathbf{S}_n^i is the contribution from the i th segment

$$\mathbf{S}_n^i = \frac{1}{2} \int_{\Delta\Gamma_i} \mathbf{x} \times d\Gamma = \frac{1}{2} \left[\int_0^1 \mathbf{x}(t) dt \right] \times \Delta\mathbf{x}_i, \quad (61)$$

while the integral in the brackets above is equal to the middle point of the segment, $\bar{\mathbf{x}}_i$:

$$\int_0^1 \mathbf{x}(t) dt = \frac{\mathbf{x}_i + \mathbf{x}_{i+1}}{2} = \bar{\mathbf{x}}_i \quad (62)$$

Thus \mathbf{S}_n^i can be computed with four identical formulae:

$$\mathbf{S}_n^i = \frac{1}{2} \bar{\mathbf{x}}_i \times \Delta\mathbf{x}_i = \frac{1}{2} \mathbf{x}_i \times \Delta\mathbf{x}_i = \frac{1}{2} \mathbf{x}_{i+1} \times \Delta\mathbf{x}_i = \frac{1}{2} \mathbf{x}_i \times \mathbf{x}_{i+1} \quad (63)$$

which are geometrically equivalent to the signed area of the triangle formed by the origin \mathbf{o} and the two end points \mathbf{x}_i and \mathbf{x}_{i+1} , or the ‘moment’ produced by the ‘force’ $\Delta\mathbf{x}_i$ about the origin. The last expression may be the most convenient to use, and leads to a simple discretised formula to evaluate \mathbf{S}_n

$$\mathbf{S}_n = \frac{1}{2} \sum_{i=1}^m \mathbf{x}_i \times \mathbf{x}_{i+1} \quad (64)$$

The triangles formed by the origin and the end points of the segments provide a triangulated mesh, \bar{S}_h , for the conical surface as the replacement surface, and \mathbf{S}_i is the same as the signed area of the i th triangle. Hence, the above result is identical to the first scheme using \bar{S}_h as the replacement surface, as shown in Fig. 7.

When a constant vector \mathbf{c} is added to all \mathbf{x} as in (22), the result of (64) will remain the same, but the formula changes to

$$\mathbf{S}_n = \frac{1}{2} \sum_{i=1}^m (\mathbf{x}_i + \mathbf{c}) \times (\mathbf{x}_{i+1} + \mathbf{c}) \quad (65)$$

which is equivalent to choosing the point $-\mathbf{c}$ instead of the origin to individual triangles or as the ‘moment’ centre. The corresponding replacement surface is still the conical surface but with $-\mathbf{c}$ as its apex. The choice of \mathbf{c} can be made to suit individual circumstances to facilitate the computation. In most cases, $\mathbf{c} = \mathbf{o}$ is the best choice.

4.4. Computation of \mathbf{G}_n

Section 2.2 proposes three schemes (25) (28) and (29) to evaluate \mathbf{G}_n , as summarised below

$$\mathbf{G}_n = \int_{\bar{S}} \mathbf{r} \times d\mathbf{S} = -\frac{1}{3} \oint_{\Gamma} \mathbf{r} \cdot \mathbf{r} d\Gamma = -\frac{1}{3} \int_{t_0}^{t_1} \mathbf{r}(t) \cdot \mathbf{r}(t) dt$$

where \bar{S} is a replacement surface of S_1 , and t is the variable used to parameterise the intersection Γ .

For the first scheme, the triangular mesh \bar{S}_h is used again as the replacement surface for \bar{S} . Let T_i be the i th triangle with \mathbf{n}_i as its unit outward normal and $\mathbf{c}_i = (\mathbf{o} + \mathbf{x}_i + \mathbf{x}_{i+1})/3$ be the centroid. The integral over T_i , \mathbf{G}_n^i , can be explicitly expressed in the following form due to the linearity of \mathbf{r} over the triangle,

$$\mathbf{G}_n^i = \int_{T_i} \mathbf{r} \times d\mathbf{S} = \int_{T_i} \mathbf{r} \times \mathbf{n}_i dA = \mathbf{c}_i \times \mathbf{S}_n^i \quad (66)$$

Then the total integral \mathbf{G}_n can be computed as

$$\mathbf{G}_n = \sum_{i=1}^m \mathbf{c}_i \times \mathbf{S}_n^i \quad (67)$$

For the second and third schemes, they are the same for each segment of Γ_h . For the i th segment $\Delta\Gamma_i$

$$\int_{\Delta\Gamma_i} \mathbf{r} \cdot \mathbf{r} d\Gamma = \left[\int_0^1 (x^2 + y^2 + z^2) dt \right] \Delta\mathbf{x}_i \quad (68)$$

Again as x , y , and z are linear functions of the parameter t on the segment $\Delta\Gamma_i$, the term in the brackets, which is the moment of inertia of the unit line segment about the origin, can be derived as

$$\int_0^1 (x^2 + y^2 + z^2) dt = \mathbf{x}_i \cdot \mathbf{x}_{i+1} + \frac{1}{3} \Delta\mathbf{x}_i \cdot \Delta\mathbf{x}_i \quad (69)$$

Therefore we have

$$\mathbf{G}_n = -\frac{1}{3} \sum_{i=1}^m (\mathbf{x}_i \cdot \mathbf{x}_{i+1} + \frac{1}{3} \Delta\mathbf{x}_i \cdot \Delta\mathbf{x}_i) \Delta\mathbf{x}_i \quad (70)$$

It can be verified that both (67) and (70) are identical, but numerically, the latter is slightly more effective as fewer vector operations are evaluated.

4.5. Determination of minimal contact surface and final contact point

In Section 2.3, the minimal contact surface is defined as the surface with a fixed boundary Γ that has a minimal total area. The intersection Γ of two triangular meshed shapes is now a polyline, but solving a minimal surface problem for this simpler boundary is still not trivial for the current contact model. Instead we further constrain the allowed surface to be a conical-like surface, similar to the conical surface \bar{S} used before: i.e. it uses Γ as its base or directrix, but its apex must be on the contact line. Thus, finding a minimal surface from this particular type of surface is equivalent to determining the final contact point. In order to derive a simple analytical solution for the problem below, we will minimise the total squared area of the surface rather than the total area. Also, we only consider the case that Γ will form a single-connected surface without holes; while when Γ consists of multiple unrelated polylines each representing an independent contact, these polylines can be dealt with as individual cases.

According to (6), a contact point on the contact line can be expressed in terms of the parameter λ as

$$\mathbf{x}_c(\lambda) = \mathbf{x}_c(0) + \lambda \mathbf{n}; \quad \text{with } \mathbf{x}_c(0) = \mathbf{n} \times \mathbf{G}_n / S_n \quad (71)$$

where $\mathbf{x}_c(0)$ is the contact point when $\lambda = 0$. Following (64), the signed area of the triangle formed by the i th segment $\Delta\Gamma_i$ and $\mathbf{x}_c(\lambda)$ takes the form

$$\mathbf{S}_i(\lambda) = \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_c) \times (\mathbf{x}_{i+1} - \mathbf{x}_c) = \mathbf{S}_i - \frac{1}{2}\mathbf{x}_c(\lambda) \times \Delta\mathbf{x}_i \quad (72)$$

Define the total squared area of the conical surface with \mathbf{x}_c as the apex by

$$S_m(\lambda) = \sum_{i=1}^m \|\mathbf{S}_i(\lambda)\|^2 \quad (73)$$

Then the surface that has the minimum total squared area when $\lambda = \lambda_m$ is the minimal contact surface, and the contact point determined by $\mathbf{x}_c(\lambda_m)$ is taken to be the final contact point.

From (71), it has

$$\frac{d\mathbf{x}_c(\lambda)}{d\lambda} = \mathbf{n} \quad (74)$$

so

$$\frac{d\mathbf{S}_i(\lambda)}{d\lambda} = -\frac{1}{2}\mathbf{n} \times \Delta\mathbf{x}_i \quad (75)$$

The condition for a minimal surface states

$$\left. \frac{dS_m(\lambda)}{d\lambda} \right|_{\lambda=\lambda_m} = - \sum_{i=1}^m \mathbf{S}_i(\lambda_m) \cdot (\mathbf{n} \times \Delta\mathbf{x}_i) = 0 \quad (76)$$

which, after some algebraic manipulation, yields the following explicit solution of λ_m

$$\lambda_m = \frac{2 \sum_i \mathbf{n} \cdot (\Delta\mathbf{x}_i \times \mathbf{S}_i) - \mathbf{x}_c(0) \cdot \sum_i (\Delta x_i^2 \mathbf{n} - \Delta x_{in} \Delta\mathbf{x}_i)}{\sum_i (\Delta x_i^2 - \Delta x_{in}^2)} \quad (77)$$

where $\Delta x_i = \|\Delta\mathbf{x}_i\|$ and $\Delta x_{in} = \Delta\mathbf{x}_i \cdot \mathbf{n}$. The final contact point can be computed as

$$\mathbf{x}_c = \mathbf{x}_c(0) + \lambda_m \mathbf{n} \quad (78)$$

4.6. Strategies to improve the overall computational performance

Finding all line segments of the intersection between two surface triangulated shapes dominates the overall performance of the current contact model, particularly for shapes with a large number of surface triangles. However, the cost occurred in this local resolution stage needs to be considered collectively with the cost occurred in the so-called global search stage of the DEM solution procedure, as outlined in [28]. The global search stage excludes those particle pairs that cannot be in contact, and produces a potential contact list between particles in the system based on their bounding box representations. Most of global search algorithms used in DEM, either tree based [29] or grid based [30,31], can effectively handle a large of number bounding boxes.

From the overall performance perspective of a DEM simulation, the following strategies can be considered in handling contacts between particles with a large number of surface triangles.

(1). *One bounding box for a particle in the global search.* This approach results in a minimal cost in the global search. Then in the local resolution stage to resolve each particle–particle contact, a local search is needed to effectively exclude most of triangle pairs (one from each particle) that do not intersect. The actual intersection computation is only performed for the remaining triangle pairs. One way to achieve this goal is to apply a local search to the bounding boxes of all the surface triangles.

(2). *One bounding box for every surface triangle of a particle in the global search.* In this way the total number of bounding boxes involved in the global search can be substantial for large scale problems, resulting in a very high computational cost at this stage. However, in the particle–particle contact stage, only triangle pairs that may be in potential contact will be involved in the intersection computation.

(3). *Multi-level bounding box representation of particles.* Each particle and its surface triangles are decomposed into a multi-level sub-domain structure using octrees [32], for instance. Each sub-domain is a bounding box of a group of triangles. Then the whole triangular mesh of a particle is represented by different sized bounding boxes depending on the level of resolution required. For a given level, non-empty sub-domains at the level will be involved in the global search. Then in the particle–particle contact stage, only triangles belonging to the sub-domains that are in the potential contact list will be checked for their intersection.

In the first option, the particle–particle resolution is computationally intensive, whereas with the second option a large portion of the computation is shifted to the global search stage. The overall performance may be similar with these two approaches depending on the implementation details. The third option attempts to balance the cost in the two stages depending on the resolution level specified. If only one level is used, this strategy becomes the first option, or the second option if a high level of resolution is applied. Thus a proper selection of the representation level based on the actual mesh of the particle may improve the overall performance of a DEM simulation for particles discretised by a large number of surface triangles.

5. Numerical verification and illustrations

In this section, the energy-conserving nature of the proposed contact model will be first verified by a set of selected examples all involving a pair-wise elastic impact between two shapes. The evolution of the kinetic energy of each system during the whole impact duration will be recorded and examined for conservation before and after the impact. Then some illustrative examples are provided to demonstrate the wide applicability of the proposed contact model and numerical methodology. All the shapes concerned will be represented by triangular meshes and thus the surface triangulated contact model and the associated numerical procedures as described in Section 4 are adopted for all the simulations in this section.

5.1. Energy conservation verification

There are four cases of elastic impact to be considered. In each case, the two shapes involved are represented by triangular surface meshes. Initially separated but under initial velocities, the shapes are subject to an elastic impact and will eventually bounce back into a separation. No friction, damping effects and gravity will be considered. To verify the energy conservation property of the proposed normal contact model, the total kinetic energy, which is the sum of translational energy and rotational energy, should be the same before and after the impact.

The central difference or leap-frog time integration scheme is applied to solve the translational motion of the two particles. As all shapes considered here are irregular and concave, the rotational motion is solved by a symplectic time integration scheme [33] in conjunction with the use of the quaternion representation for shape orientation. The recommended linear contact model (34) is used for all the cases. In addition, the time step employed for time integration is selected such that it is sufficiently small to ensure both numerical stability and reasonable accuracy. Note that the effects of the proposed model on the critical time step and other aspects of a DEM simulation will be addressed elsewhere.

In each case, the evolution of the translational, rotational and total kinetic energy of the two shapes as one system in the whole impact duration will be displayed. The total kinetic energy conservation can be validated if its value

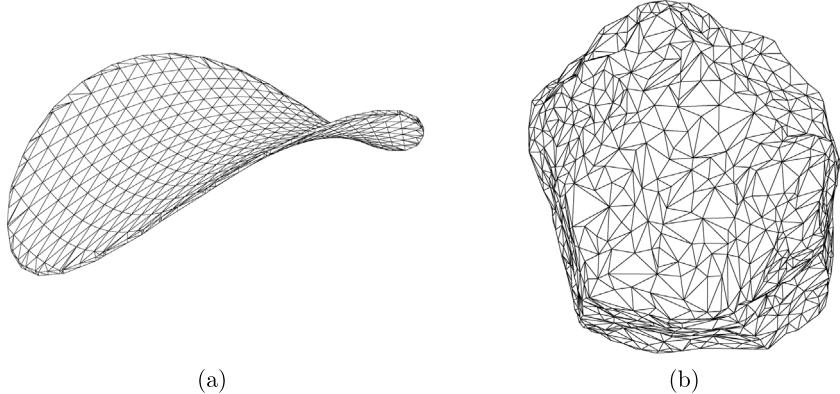


Fig. 8. Surface triangulations: (a) a crisp with 511 vertices, 1018 triangles; and (b) a rock with 625 vertices, 1246 triangles.

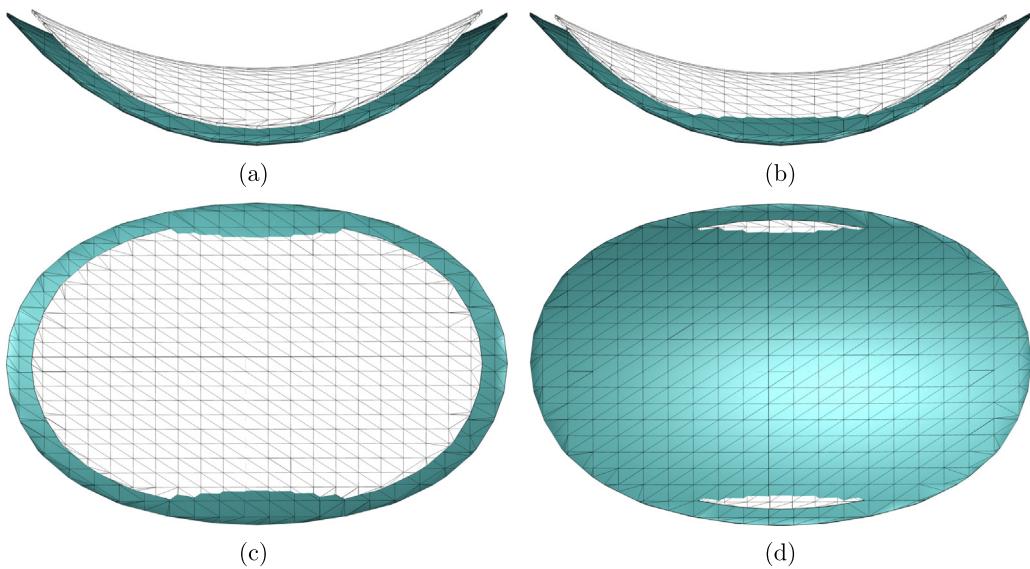


Fig. 9. Elastic impact of two crisps: (a) initial configuration; (b) configuration at the maximum penetration, and its top view (c) and bottom view (d).

remains unchanged before and after the impact. To simplify comparisons, both the energy and impact time are normalised without affecting the conclusion.

Case 1: Elastic impact of two crisps. Two crisps, both scaled versions of the sample (smoothed) crisp shown in Fig. 8(a), have the same orientation but slightly different sizes. The surface triangular mesh for each shape has 511 vertices and 1018 triangles. Note that the thickness is (relatively) very small. The smaller crisp is initially positioned exactly on top of the larger one, and both are subject to a “head-on” collision with unit but opposite initial velocities in the vertical direction, as shown in Fig. 9(a).

The snapshot of the impact at the maximum penetration is displayed in Fig. 9(b) with the top view in Fig. 9(c). There are clearly two independent contact regions developed which are not exactly the same shape because of the asymmetric surface meshes used. Also, due to the very small thickness, together with the use of a small normal contact stiffness k_n , the smaller crisp partially penetrates through the larger one, as shown in Fig. 9(d) which is the bottom view of the contact at the maximum penetration.

The evolution of the (normalised) translational, rotational and total kinetic energy of the system is shown in Fig. 10. Due to the same orientation of the two shapes and the “head-on” impact direction, almost no rotation

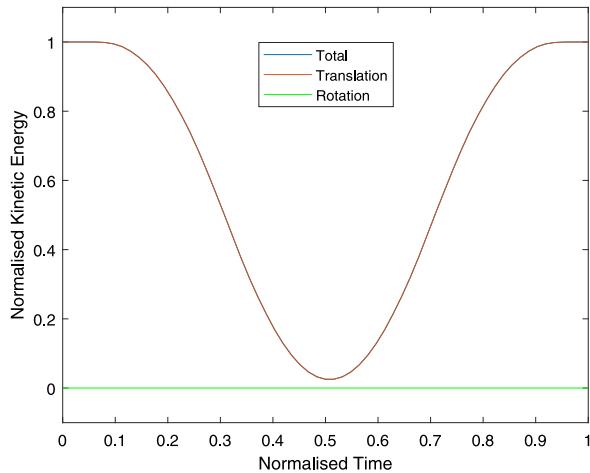


Fig. 10. Elastic impact of two crisps: evolution of translational, rotational and total kinetic energy.

occurs after the impact. It is evident that the total kinetic energy is indeed conserved, even a large penetration contact with partial pass-through is encountered in this case.

Case 2: Elastic impact of two rocks. The two rock particles, obtained by scaling the sample rock shown in Fig. 8(b), have slightly different sizes. Each surface triangular mesh has 625 vertices and 1246 triangles.

Initially, the two rocks are randomly orientated and are subject to an elastic collision with unit but opposite initial velocities in the horizontal direction, as shown in Fig. 11(a). The snapshots of the impact at an instance in the first stage of impact and at the maximum penetration are displayed in Fig. 11(b) and (c) respectively. As a small value of the contact stiffness k_n is used, the maximum contact region and penetration depth are excessive as illustrated in Fig. 9(c).

Fig. 12 shows the evolution of the (normalised) translational, rotational and total kinetic energy of the system during the impact. Because of the irregularity of the shapes, the rotational motion has developed after the impact. It is evident that the total kinetic energy is again conserved, even both an excessive penetration contact and rotational motion are observed in this case.

Case 3: Elastic impact of a nut and a bolt. The nut and the bolt used are the scaled versions of the sample nut and bolt models shown in Figs. 13(a) and (b) respectively, where the nut consists of 625 vertices and 1246 triangles, while the bolt has 1030 vertices and 2056 triangles. Note that there are many highly badly shaped triangles in both meshes.

Initially, the nut has a tilted orientation and the bolt in an up right position. They are given unit but opposite initial velocities in the horizontal direction, as shown in Fig. 14(a). The snapshots of the impact at an instance in the early stage of impact, at the maximum penetration and after the impact are displayed in Fig. 14(b), (c) and (d) respectively.

Again a small value of the contact stiffness k_n is used in order to induce a large contact region to test the proposed contact model. At the early stage of the impact (b), there are three separate contact regions but they emerge into one when the impact progresses as shown in (c). Thus the multiple contact regions in the early stage, which also re-appear in the bounce back stage, are of an associated nature, as classified in Section 5.1 of [1].

Fig. 15 depicts the evolution of the (normalised) translational, rotational and total kinetic energy of the system during the impact. Owing to the irregularity of the shapes, the rotational motion has developed after the impact as in the previous case. The total kinetic energy is clearly conserved, although an associated multiple contact scenario coupled with large penetration and rotational motion has occurred in this case.

Case 4: Elastic impact of two bolts. The bolts are the same as the one used in the previous case. Initially, they are in an up right configuration with opposite orientations and are subject to unit but opposite initial velocities in the horizontal direction, as shown in Fig. 16(a). The snapshots of the impact at an instance in the early stage, at the maximum penetration and after the bounce back are displayed in Figs. 16(b), (c) and (d) respectively.

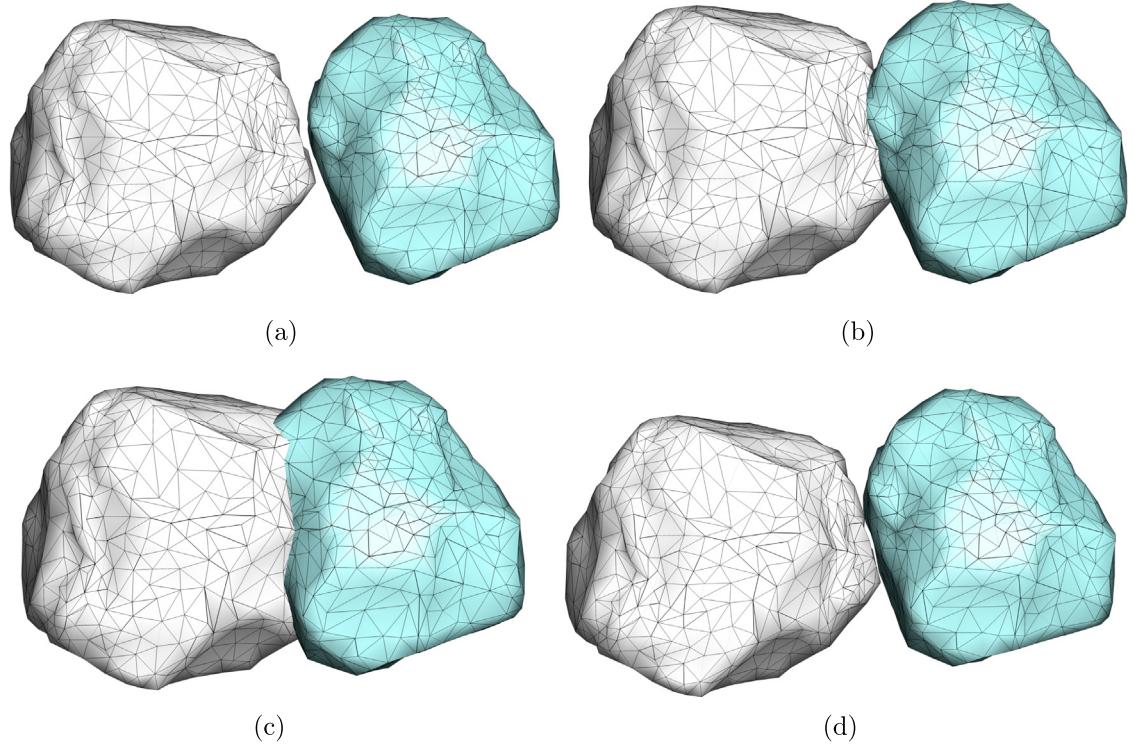


Fig. 11. Elastic impact of two rocks: (a) before impact; (b) at the increasing penetration stage; (c) at the maximum penetration; and (d) after impact.

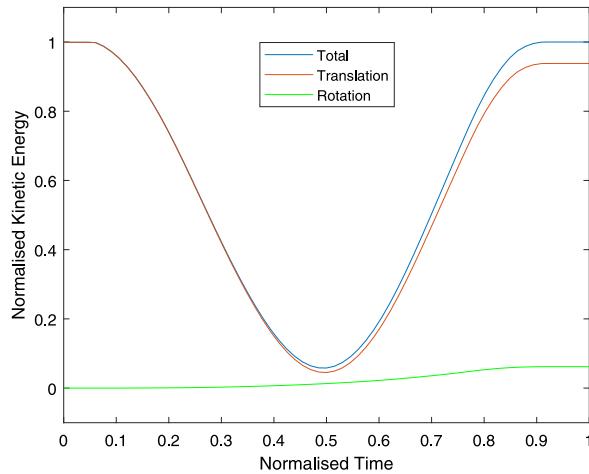


Fig. 12. Elastic impact of two rocks: evolution of translational, rotational and total kinetic energy.

As shown in (b), there exist multiple independent contact regions as soon as the contact is established between the two, but because the linear spring stiffness is not too small and the maximum penetration is not large, these contact regions remain isolated throughout the whole impact duration. Thus this is a non-associated multiple-contact case, as classified in Section 5.1 of [1].

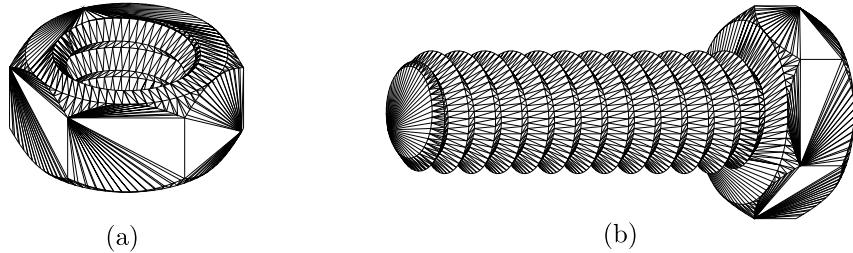


Fig. 13. Surface triangulations: (a) a nut with 645 vertices and 1286 triangles; and (b) a bolt with 1030 vertices and 2056 triangles.

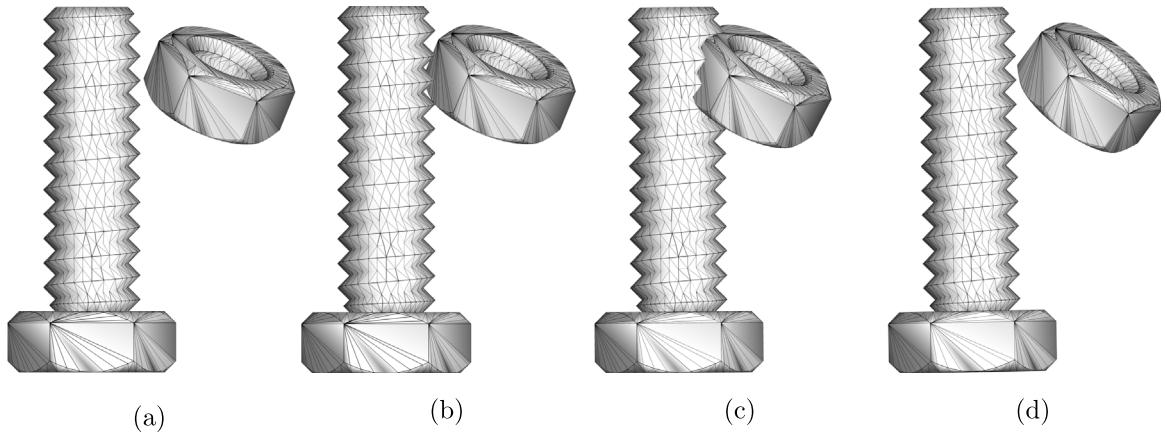


Fig. 14. Elastic impact of a nut with a bolt: (a) before impact; (b) during increasing penetration stage; (c) at the maximum penetration; and (d) after impact.

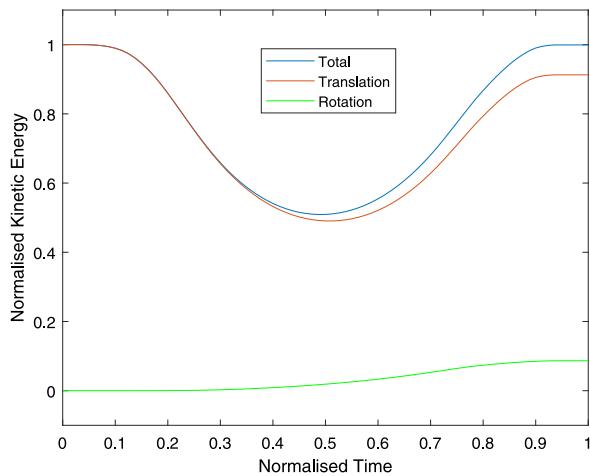


Fig. 15. Elastic impact of a nut and a bolt: evolution of translational, rotational and total kinetic energy.

Fig. 17 shows the evolution of the (normalised) translational, rotational and total kinetic energy of the system during the impact. Due to the irregularity of the shapes, the rotational motion has developed after the impact, but the total kinetic energy is still conserved in this case.

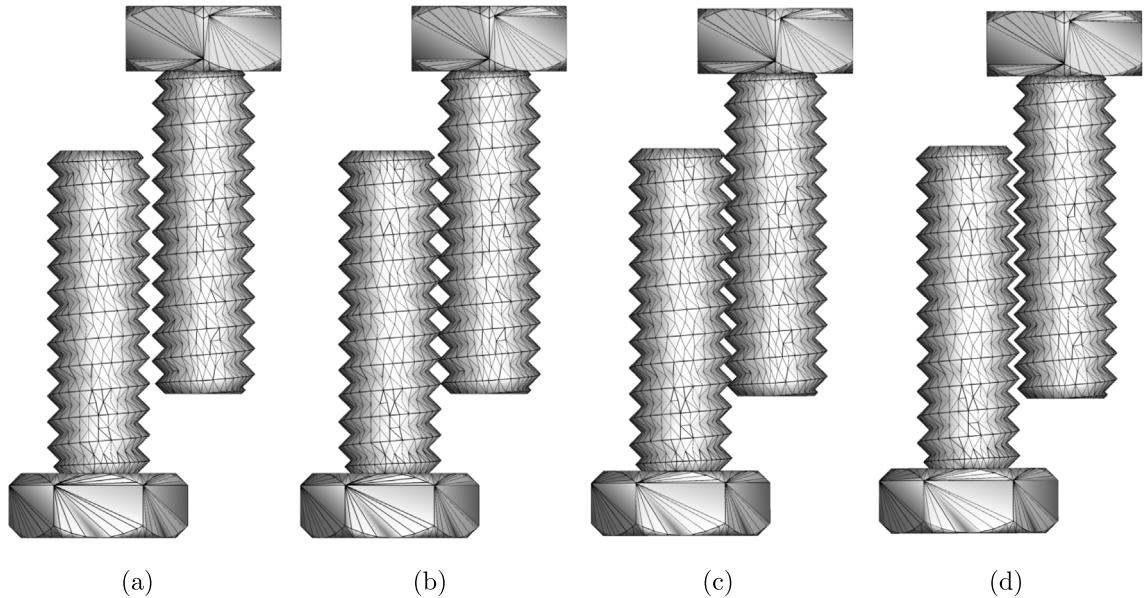


Fig. 16. Elastic impact of two identical bolts: (a) before impact; (b) during increasing penetration stage; (c) at the maximum penetration; and (d) after impact.

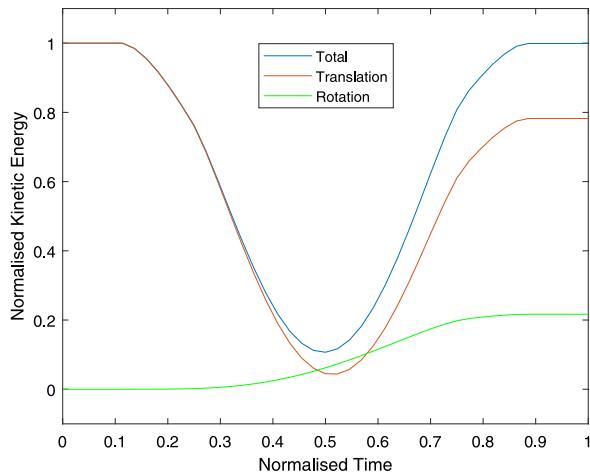


Fig. 17. Elastic impact of two bolts: evolution of translational, rotational and total kinetic energy.

From the above four cases, it is evident that energy conservation has been indeed observed in various contact scenarios. It has also been demonstrated that the proposed numerical procedures in association with the contact model are also very robust and are not sensitive to the (poor) quality of triangular meshes representing general shapes.

5.2. Further illustrative examples

After the energy-conserving nature of the proposed contact model has been numerically verified above, six more examples are presented below to further illustrate the robustness and applicability of the model in more realistic scenarios. In the first five examples, a coefficient of friction of 0.5 is used, while a value of 0.05 is used for **Example 6**. The classic Coulomb friction model is adopted for the frictional tangential contact. The standard linear

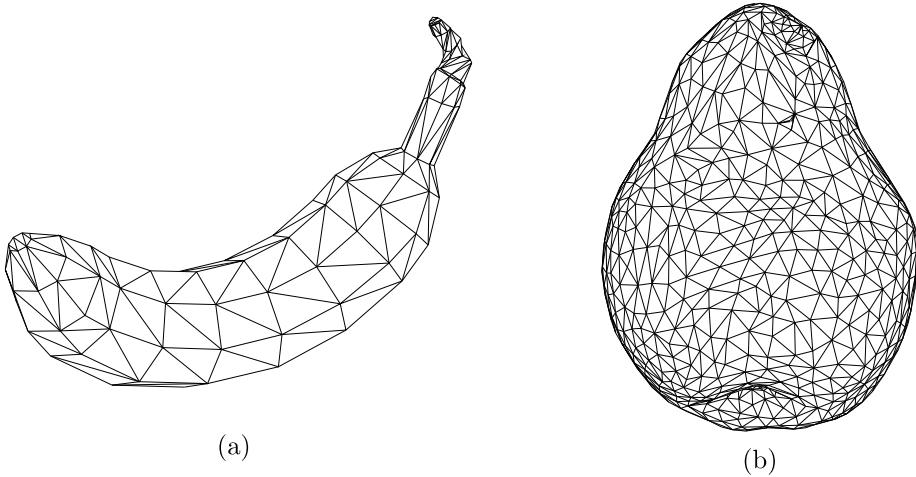


Fig. 18. Surface triangulations: (a) a banana with 152 vertices and 300 triangles; and (b) a pear with 851 vertices and 1696 triangles.

viscous damping model is extended and adopted with the current contact model. The damping ratio equivalent to a restitution of 0.1 is applied to both normal and tangential contacts. Gravity is activated.

Again, both the central difference and the symplectic time integration scheme [33] are used to solve both translational motion and rotational motion of particles respectively. The time step in each case is selected mainly to achieve a stable simulation. The contact point of each contact is determined by the minimum surface scheme presented in Section 4.5. The first strategy as described in Section 4.6 is adopted in each example where every triangle of a particle is represented by an axial aligned bounding box and is considered in the global search. No further attempt is made to optimise the overall performance of the simulation by considering the other two strategies mentioned in Section 4.6, and thus the actual computational costs involved in these examples are not included.

The main purpose of these examples is purely to demonstrate the suitability and applicability of the proposed contact model for arbitrarily shaped particles in wider simulation scenarios. Thus no effort is made to choose any geometric and physical parameter involved in a simulation to reflect any real application that the example may intend to mimic. In all cases, the energy conservation feature of the proposed contact model and the associated numerical procedure ensure that no velocity spike will appear to cause numerical instability in a simulation.

The first four examples involve the random deposition of various shaped particles or their combinations with different sizes within a box. Owing to the existence of friction and damping effects that dissipate energy, particles will eventually settle in a static packing state. In Example 5, rock particles and cylindrical particles are mixed in a ball-milling setting where the boundary is moving. The last example concerns the sliding of rock particles on the surface of a terrain to mimic a land slide.

Example 1 (Random Packing of Bananas and Pears). In this example, bananas and pears, based on the triangulated models shown in Figs 18(a) and (b) respectively, are used. The surface mesh of a banana has 152 vertices and 300 triangles; the mesh for a pear has 851 vertices and 1696 triangles. A total number of 1160 bananas and pears with slightly different sizes are randomly generated and dropped under gravity in the box. In total there are 334,993 vertices and 664,852 triangles in the system after all the particles are generated. The configurations of the system at four time instants are illustrated in Fig. 19.

Example 2 (Random Packing of Nuts and Bolts). Nuts and bolts, based on the triangulated models depicted in Figs 13(a) and (b) respectively, are used in this example. A total number of 409 nuts and bolts are randomly generated and dropped under gravity in the box, resulting in 300,332 vertices and 600,276 triangles for all the particles in the system. The configurations of the system at four different time instants are illustrated in Fig. 20.

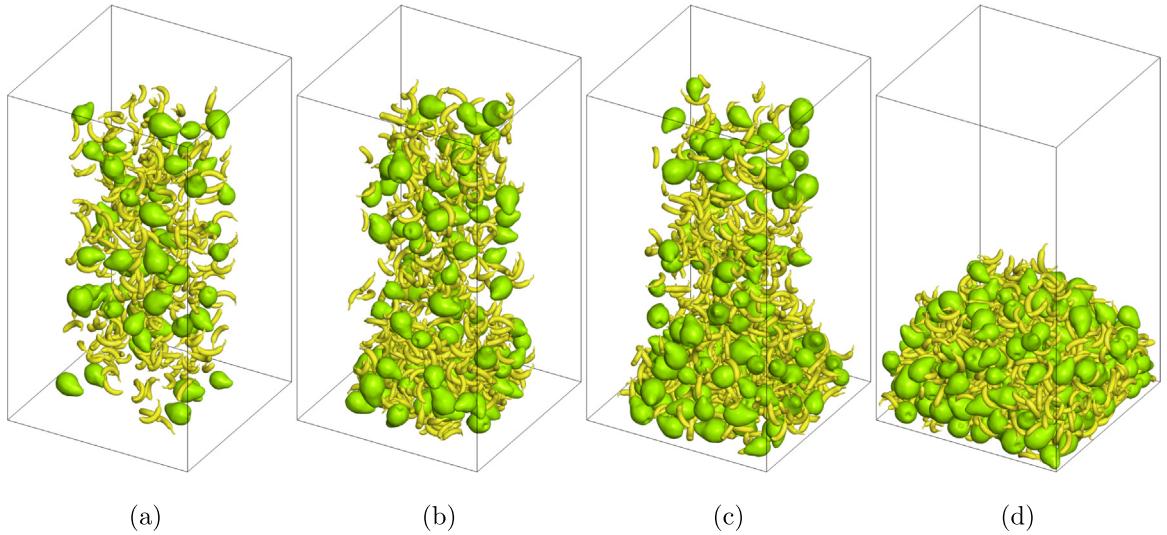


Fig. 19. Random deposition of bananas and pears in a box with their configurations at several time instances (a)–(d).

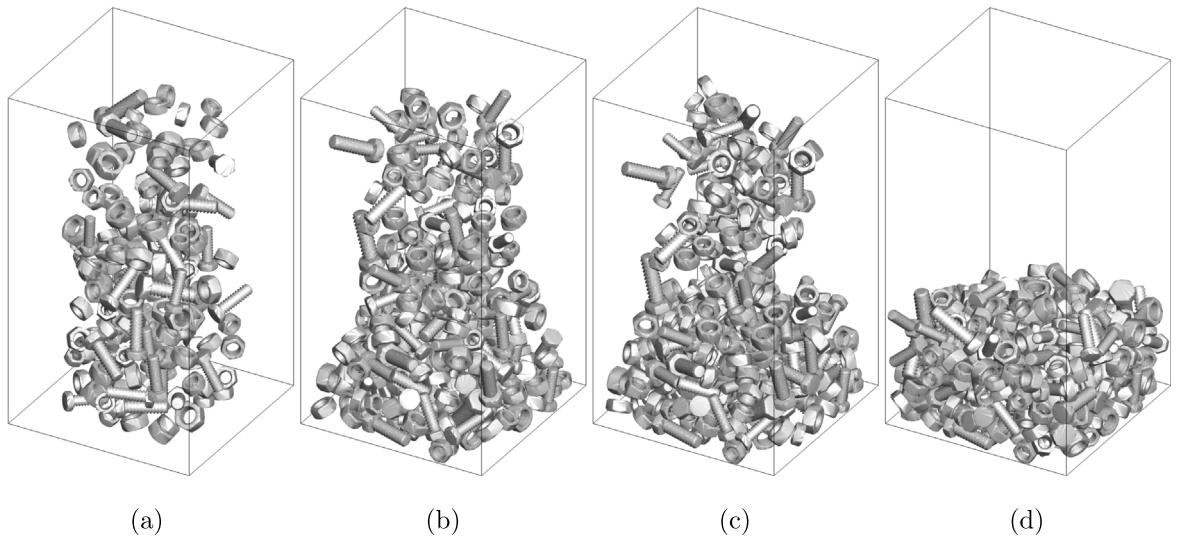


Fig. 20. Random deposition of nuts and bolts in a box with their configurations at several time instances (a)–(d).

Example 3 (Random Packing of Rock Particles). In this example, the same triangulated rock model shown in Figs 8(b) is used to generate rock particles with slightly different sizes. 2630 rock particles are randomly injected and dropped under gravity in the box. So the total number of vertices and triangles of all the rock particles are 1,643,750 and 3,276,980 respectively. The configurations of the system at four different time instants are illustrated in Fig. 21.

Example 4 (Random Packing of Letters “D”, “E” and “M”). In this example, discretised 3D letters “D” and “E” and “M” are used as rigid particles. Their surface triangulations are shown in Fig. 22. The numbers of vertices and triangles are respectively 172 and 340 for “D”, 24 and 44 for “E”; and 26 and 48 for “M”. In total, 1097 of these letters with varied sizes are randomly generated and dropped under gravity in the box. The configurations of the system at four different time instants are illustrated in Fig. 23. This is a relatively easy case as most of contacts

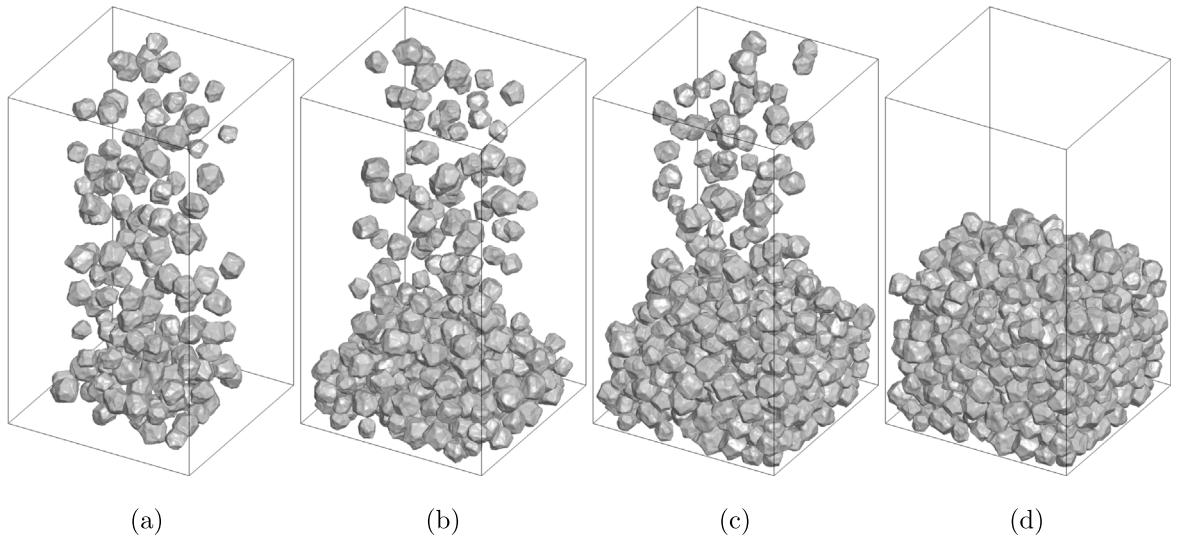


Fig. 21. Random deposition of rocks in a box with their configurations at several time instances (a)–(d).

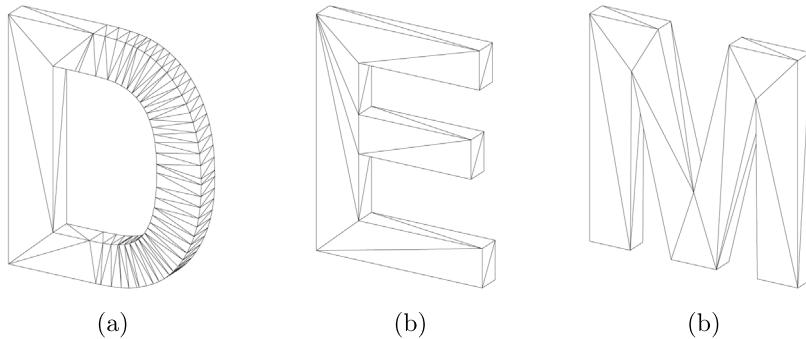


Fig. 22. Surface triangulations: (a) 3D letter “D” with 172 vertices and 340 triangles; (b) 3D letter “E” with 24 vertices and 44 triangles; and (c) 3D letter “M” with 26 vertices and 48 triangles.

are of a local convex contact nature and the total numbers of vertices and triangles of the letters involved are low compared to the other examples.

In the above four examples, the particles involved have all settled into a stable packing state and no any numerical instability has been observed, demonstrating again the superb robustness and stability of the proposed contact model.

Example 5 (Rocks and Metal Bars in a Rotating Drum with Baffles). This example attempts to simulate the operation of a ball mill in a conceptual way. A 12-sided rotating drum of 2 metres in diameter with six baffles is represented by a surface triangular mesh, and rotates at 20 rpm anti-clockwise. Over 200 rock particles with varied sizes are obtained from scaling of the triangulated rock model shown in Fig. 8(b). 40 long cylinders representing metal bars are each discretised by 74 vertices and 144 regular surface triangles. Although a cylinder is a convex shape, no distinction in terms of convexity is made between the cylinders and the rock particles in the simulation and the same numerical procedure as outlined in Section 4 is adopted.

The meshes of the drum, rock particles and metal bars are shown in Fig. 24(a). Both rock particles and cylindrical bars are randomly and continuously injected into the drum from a small cuboid region around the rotating axis of the drum. The configurations of the system at three different time instants $t = 3, 6$, and 9 s are illustrated in Figs. 24(b)–(d) respectively. The velocity magnitudes of the metal bars are also indicated visually in colours in the figures (where blue and red denote the lowest and highest velocities respectively).

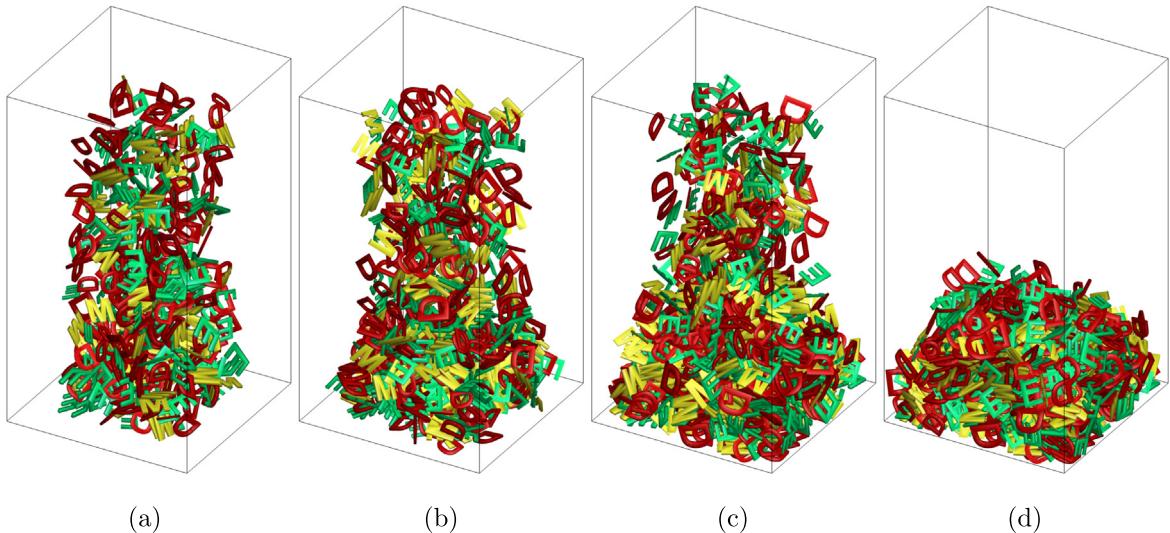


Fig. 23. Random deposition of 3D letters “D”, “E” and “M” in a box with their configurations at several time instances (a)–(d).

Example 6 (Rocks Sliding on the Surface of a Terrain). This example has a similar setting as a land slide scenario but without considering fine particles. Rock particles with varied sizes are obtained from scaling the triangulated rock model shown in Fig. 8(b). The terrain as a 3D region is represented by a closed surface triangulated mesh with 1680 vertices and 3356 triangles, and is artificially constrained by four high vertical planes around the four boundary sides to avoid the rock particles moving beyond the terrain. The rock particles are randomly and progressively released from a cuboid region above the far corner of the terrain as shown in Fig. 25(a). There are 7543 rock particles generated in total, resulting in 4,714,375 vertices and 9,398,578 triangles that are used to represent the rock particles. Thus the computational costs involved at each time step are considerable.

The coefficient of friction between the rocks and the terrain surface is taken to be a low value of 0.05 to increase the mobility of the rocks. Under the action of gravity, the rock particles start to slide on the surface of the terrain and move towards the lower valley, and eventually fill the whole valley due to the low friction and the block of the four vertical planes. The configurations together with the velocity magnitudes of the rocks at four different time instances are illustrated in Fig. 25, where the four vertical planes are not shown.

Again, in the above two examples, no any numerical instability has been observed in the simulations, showing that the proposed contact model and the corresponding numerical procedures can also effectively deal with irregular particles in contact with moving or complex shaped boundaries.

6. Concluding remarks

This paper has re-established a special form of the general energy-conserving contact model [1], based on the assumption that the contact energy potential is a function of the contact volume between two contacting bodies with general shapes in both 2D and 3D cases. By specifying such a function, the normal contact features associated with the contact model are fully determined without the need to introduce any additional parameters. The contact point, which can be any point along the contact line, can be fixed following the introduction of a simplified minimum surface concept. By further exploiting the geometric properties of the contact surfaces concerned, more effective integration schemes have been developed to reduce the evaluation costs involved in the model. When a linear contact energy function in terms of the contact volume is adopted, a linear contact model has been derived in which no contact region and surfaces need to be explicitly constructed and only the intersection between the two shapes is required to be constructed. These developments have therefore significantly improved the numerical efficiency of the proposed model compared to the previous version [5].

For general 3D shapes except for spheres, a mesh discretisation should be used. When a shape is represented by a volumetric (i.e. tetrahedral or hexahedral) mesh, the contact of two such shapes can be obtained by independently

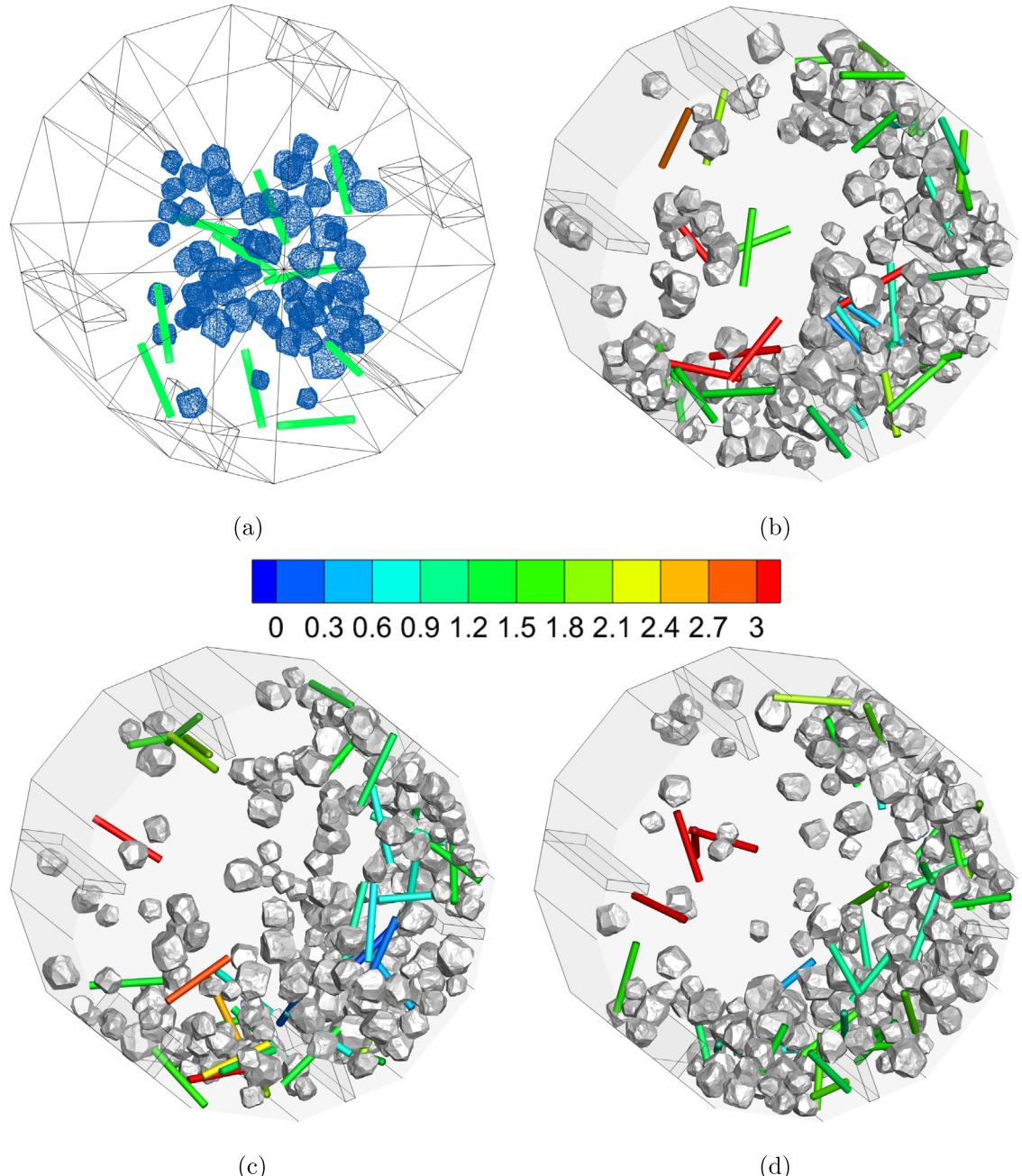


Fig. 24. Rocks and metal bars in a rotating drum with baffles: triangulations of rocks and rods and the drum (a); and configurations at several time instances $t = 3$ s (b), 6 s (c) and 9 s (d). Colours in (b)–(d) representing velocity magnitudes of bars. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

processing multiple pair-wise contacts between two convex polyhedra using the two techniques discussed. However, the surface triangulation representation is recommended and the corresponding numerical procedures have been discussed in detail. The associated core operation involves the determination of the intersections between two sets of triangles from the two shapes concerned. Based on the additive property of the proposed contact model, the modelling of two 3D shapes in contact can be conducted fully in parallel without incurring much communication overhead, using either a volumetric or surface triangular mesh representation. Consequently the proposed contact

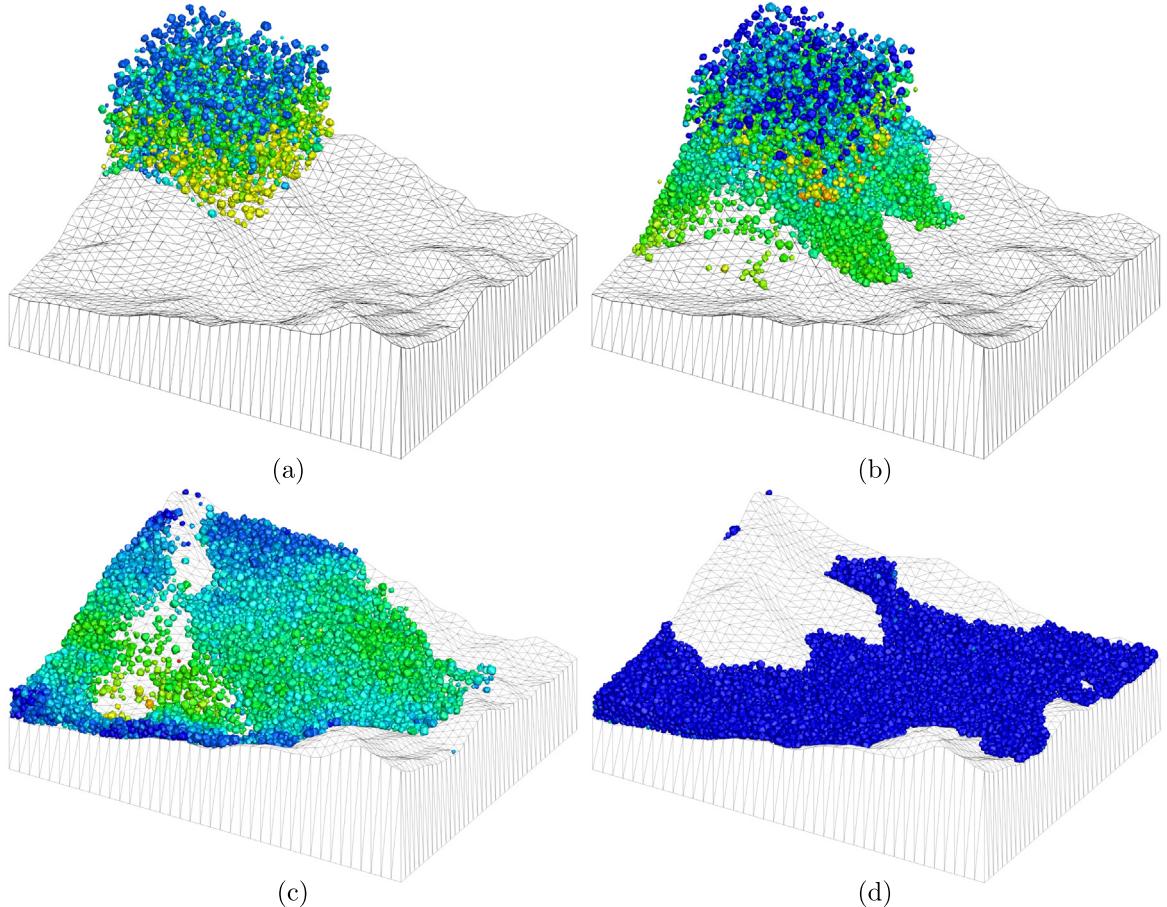


Fig. 25. A column of rock particles sliding on a terrain surface: configurations at different time instances (a)–(d). Colours representing velocity magnitudes of rocks.

model is ideal for parallel GPU or GPGPU computing, although this aspect has not been implemented in the current work.

The numerical examples conducted have verified the energy-conserving property of the proposed contact model for a wide range of shapes and contact scenarios. Additional examples have been provided to illustrate the robustness and applicability of the model for more realistic problems.

As highlighted in [1], the proposed contact volume based normal contact model is based on a particular form of the contact energy form. It is essentially a numerical based contact model, and therefore may not be able to truthfully capture the contact physics of particles. Thus further validation of the model and comparison with some existing contact models for certain types of particle shapes are needed.

Also note that the proposed contact model results in contact interaction laws that differ from traditional overlap based contact models. Consequently, additional theoretical issues arise, such as how to determinate the critical time step and how to incorporate the contact model properly with the standard (linear) viscous damping model to achieve a similar energy dissipation behaviour. These issues will be addressed in future work. Furthermore, the overall computational efficiency of DEM simulations using the current model is a major issue for large scale problems with particles represented by a large number of vertices and triangles. Although three possible strategies have been proposed, no numerical evidence has been presented to compare their relative performance. Further work is needed, particularly in conjunction with GPU or GPGPU parallel computing techniques.

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.

References

- [1] Y.T. Feng, Energy-conserving contact interaction models for arbitrarily shaped discrete elements: Basic framework and general contact model, *Comput. Methods Appl. Mech. Engrg.* 373 (2021) 113454, <http://dx.doi.org/10.1016/j.cma.2020.113454>.
- [2] Y.T. Feng, D.R.J. Owen, A 2D polygon/polygon contact model: algorithmic aspects, *Int. J. Eng. Comput.* 21 (2004) 265–277.
- [3] K. Han, Y.T. Feng, D.R.J. Owen, Contact resolution for non-circular discrete objects, *Internat. J. Numer. Methods Engrg.* 66 (3) (2006) 485–501.
- [4] Y.T. Feng, K. Han, D.R.J. Owen, An energy based polyhedron-to-polyhedron contact model, in: Proceeding of 3rd M.I.T. Conference of Computational Fluid and Solid Mechanics, MIT, USA, 2005, pp. 210–214.
- [5] 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 (2012) 169–177.
- [6] Y.T. Feng, A general contact theory for non-spherical particles, in: X. Li, Y. Feng, G. Mustoe (Eds.), *Proceedings of 7th International Conference on Discrete Element Methods, Dem 2016*, Springer, 2017, pp. 29–35.
- [7] T. Luo, E.T. Ooi, A.H.C. Chan, S.J. Fu, The combined scaled boundary finite-discrete element method: Grain breakage modelling in cohesion-less granular media, *Comput. Geotech.* 88 (2017) 199–221.
- [8] B. Nassauer, T. Liedke, M. Kuna, Polyhedral particles for the discrete element method: Geometry representation, contact detection and particle generation, *Granular Matters* 15 (2013) 85–94.
- [9] N. Govender, N. Daniel, C.-Y. Wu, J. Khinast, P. Pizette, X. Wenjie, Hopper flow of irregularly shaped particles (non-convex polyhedra): GPU-based DEM simulation and experimental validation, *Chem. Eng. Sci.* 188 (2018) 34–51.
- [10] C. Wellmann, C. Lillie, P. Wriggers, A contact detection algorithm for superellipsoids based on the common-normal concept, *Eng. Comput.* 25 (5) (2008) 432–442.
- [11] A. Podlozhnyuk, S. Pirker, C. Kloss, Efficient implementation of superquadric particles in discrete element method within an open-source framework, *Comput. Part. Mech.* 4 (2017) 101–118.
- [12] S. Wang, Y. Fan, S.Y. Ji, Interaction between super-quadric particles and triangular elements and its application to hopper discharge, *Powder Technol.* 339 (2018) 534–549.
- [13] M.A. Hopkins, Discrete element modeling with dilated particles, *Eng. Comput.* 21 (2/3/4) (2004) 422–430.
- [14] L. Liu, S.Y. Ji, A fast detection algorithm based on the envelope function of dilated polyhedron, *Sci. Sin-Phys. Mech. Astron.* 49 (2019) 064601 (in Chinese).
- [15] L. Liu, S.Y. Ji, Bond and fracture model in dilated polyhedral DEM and its application to simulate breakage of brittle materials, *Granul. Matter* 21 (2019) <http://dx.doi.org/10.1007/s10035-019-0896-4>, Article No. 41.
- [16] A. Munjiza, *The Combined Finite-Discrete Element Method*, Wiley & Sons, England, 2004.
- [17] W. Kaplan, Integrals depending on a parameter—Leibnitz's rule, in: *Advanced Calculus*, second ed., Addison-Wesley, 1973, pp. 285–288.
- [18] L. Leal, *Advanced Transport Phenomena: Fluid Mechanics and Convective Transport Processes*, Cambridge University Press, ISBN: 978-0-521-84910-4, 2007.
- [19] J. Marsden, T. Anthony, *Vector Calculus*, fifth ed., W. H. Freeman, 2003.
- [20] W.H. Meeks III, J. Pérez, The classical theory of minimal surfaces, *Bull. Amer. Math. Soc.* 48 (3) (2011) 325–407.
- [21] A. Gray, Ch. 30: Minimal surfaces and Ch. 31: Minimal surfaces and complex variables, in: *Modern Differential Geometry of Curves and Surfaces with Mathematica*, second ed., CRC Press, Boca Raton, FL, 1997, pp. 681–734.
- [22] Y.T. Feng, K. Han, D.R.J. Owen, A generic contact detection framework for cylindrical particles in discrete element modelling, *Comput. Methods Appl. Mech. Engrg.* 315 (2017) 632–651.
- [23] S. Zhao, J. Zhao, A poly-superellipsoid-based approach on particle morphology for DEM modeling of granular media, *Int. J. Numer. Anal. Methods Geomech.* 43 (2019) 2147–2169.
- [24] D.E. Muller, F.P. Preparata, Finding the intersection of two convex polyhedra, *Theoret. Comput. Sci.* 7 (2) (1978) 217–236.
- [25] S. Zhao, X. Zhou, W. Liu, Discrete element simulations of direct shear tests with particle angularity effect, *Granul. Matter* 17 (2015) 793–806.
- [26] Y.T. Feng, Y. Tan, On Minkowski difference-based contact detection in discrete-discontinuous modelling of convex polygons/polyhedral: Algorithms and implementation, *Eng. Comput.* 37 (2020) 54–72.
- [27] Y.T. Feng, Energy-conserving contact interaction models for arbitrarily shaped discrete elements: Contact overlap based model and computational issues, 2021, in preparation.
- [28] 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, *Internat. J. Numer. Methods Engrg.* 60 (1) (2004) 317–340.
- [29] Y.T. Feng, D.R.J. Owen, An augmented spatial digital tree algorithm for contact detection in computational mechanics, *Internat. J. Numer. Methods Engrg.* 55 (2002) 159–176.
- [30] A. Munjiza, K. Andrews, NBS: contact detection algorithm for bodies of similar size, *Internat. J. Numer. Methods Engrg.* 43 (1998) 131–149.
- [31] J.R. Williams, E. Perkins, B. Cook, A contact algorithm for partitioning N arbitrary sized objects, *Eng. Comput.* 21 (2004) 215–234.
- [32] D. Meagher, Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects By Computer, Technical Report IPL-TR-80-111, Rensselaer Polytechnic Institute, 1980.
- [33] A. Dullweber, B. Leimkuhler, R. McLachlan, Symplectic splitting methods for rigid body molecular dynamics, *J. Chem. Phys.* 107 (15) (1997) 5831–5851.