A NEW CONTACT DETECTION ALGORITHM FOR THREE-DIMENSIONAL NON-SPHERICAL PARTICLES

C.W. Boon[1], G. T. Houlsby[1] and S. Utili[2]

[1]Department of Engineering Science, Oxford University Parks Road, Oxford OX1 3PJ, United Kingdom

[2]School of Engineering, University of Warwick, Coventry CV4 7AL, United Kingdom; formerly at University of Oxford.

ABSTRACT

A new contact detection algorithm between three-dimensional non-spherical particles in the discrete element method (DEM) is proposed. Houlsby previously proposed the concept of potential particles where an arbitrarily shaped convex particle can be defined using a 2nd degree polynomial function [1]. The equations in 2-D has been presented and solved using the Newton-Raphson method. Here the necessary mathematics is presented for the 3-D case, which involves non-trivial extensions from 2-D. The polynomial structure of the equations is exploited so that they are second-order cone representable. Second order-cone programs have been established to be theoretically and practically tractable, and can be solved efficiently using primal-dual interior-point methods [13]. Several examples are included in this paper to illustrate the capability of the algorithm for particles of various shapes.

Keywords: DEM; non-spherical; polyhedral; contact detection; potential particles;

NOTATIONS

a, b, c, d	constants defining a plane in 3D
f	mathematical function defining a potential particle
k	fraction of the spherical term of a potential particle, and when subscripted represents that the coefficients with *k* has been factored out
p_i	slack variables for the planar terms of a potential particle
$\boldsymbol{q}$	unit quaternion
$\mathbf{Q}$	rotation matrix
r	radius of the curvature at the edges of a potential function without the spherical term
R	radius of the spherical part of the particle
s	slack variable for a potential function
x, y, z	Cartesian coordinates
$\mathbf{x}$	vector of Cartesian coordinates
w	constants for slack variables
θ	particle orientation
A	subscript identifying particle A
B	subscript identifying particle B

1 INTRODUCTION

Although spheres remain popular in the discrete element method (DEM) because of their computational efficiency in contact detection, particles in real life are largely non-spherical. Granular and powder materials are present in many shapes, most of which are non-spherical. The processing of these materials is important in many engineering applications. These encompass operations such as storage, conveying, mixing and sizing from small scale pharmaceutical or food processing operations, where composition control may be critical, to large scale industry storage where wall stresses may be important. Non-spherical granular particles, e.g. tablets, are frequently encountered in the chemical, food and pharmaceutical industries. The flow, arching and jamming mechanisms of these particles in hoppers and silos are more complex than for spherical particles. For instance, Cleary & Sawley [8] showed that the effect of particle shape on hopper discharge and stress patterns can be significant. Wu & Cocks [19] and Mack et al. [20] have compared the results of DEM simulations with real experimental data in 2-D and 3-D respectively. They showed that particle shapes can significantly influence the particle flow properties.

Various methods to model non-spherical particles have been proposed in the literature, most of which impose restrictions on the shape of the particles, i.e., either the particle has to be polyhedral or the particle shape is restricted to a particular type of function [2, 3, 4, 5, 6, 7, 9]. In applications such as powder technology, where particles may assume a wide range of shapes, it is important to have a 3-D contact detection algorithm that is as general as possible so that the same algorithm can be used repeatedly for different processes. This also allows numerical parametric studies to be performed across different particle shapes without being limited by the capability of the contact detection algorithm that has been implemented into the DEM code. The method of potential particles introduced by Houlsby [1] can model any convex particle shape from circular to roughly polygonal in 2-D and from spherical to roughly polyhedral in 3-D. In his paper, the contact detection algorithm in 2-D has been presented and solved using the Newton-Raphson method. Here, the solution for the 3-D case, which involves some non-trivial extensions from 2-D, is presented. The equations to be solved are formulated into a second-order cone program

(SOCP), which has been widely established to be theoretically and practically tractable. SOCP solvers are generally held to be robust and efficient because they can use primal-dual interior-point methods.

In the next section, the mathematical formulation of the proposed contact detection algorithm is illustrated. In the following section, three numerical examples are provided to illustrate the capabilities of the algorithm for non-spherical particles of different shapes. The robustness of the algorithm was tested for particles of both low and high aspect ratios.

2 THEORY AND METHODOLOGY

2.1 Particle Definition

Based on the notion that a convex particle can be constructed from an assembly of lines in 2-D or planes in 3-D, Houlsby [1] describes an arbitrary convex particle in terms of a 2nd degree polynomial function (with respect to a local coordinate system). In 3-D, it can be expressed as:

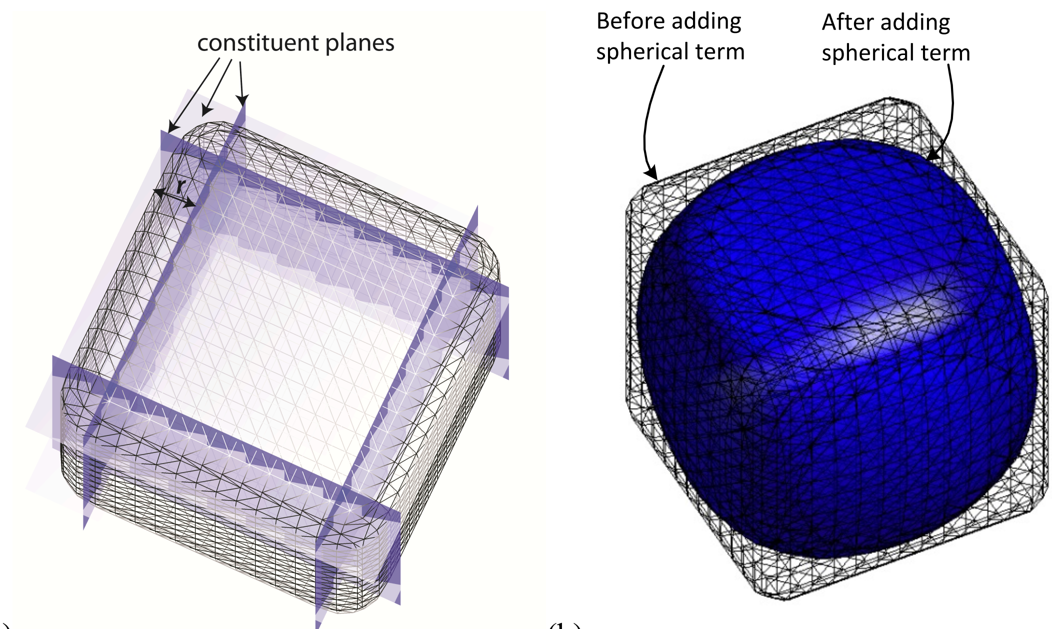
$$f = (1-k)\left(\sum_{i=1}^{N}\langle a_i x + b_i y + c_i z - d_i\rangle^2 - r^2\right) + k(x^2 + y^2 + z^2 - R^2) \tag{1}$$

where (a_i, b_i, c_i) is the normal vector of the i^{th} plane defined with respect to the particle local coordinate system, and d_i is the distance of the plane to the local origin. $\langle\ \rangle$ are Macaulay brackets, i.e., $\langle x\rangle = x$ for $x > 0$; $\langle x\rangle = 0$ for $x \leq 0$. The planes are assembled such that their normal vectors point outwards. They are summed quadratically and expanded by a distance r (see Figure 1(a)), which is also related to the radius of the curvature at the corners [1]. Further, a "shadow" spherical particle is added; R is the radius of the sphere, with $0 < k \leq 1$ denoting the fraction of sphericity of the particle (see Figure 1(b), (c) and (d)). Houlsby [1] calls this function a "potential particle", which has the following properties (see Figure 2):

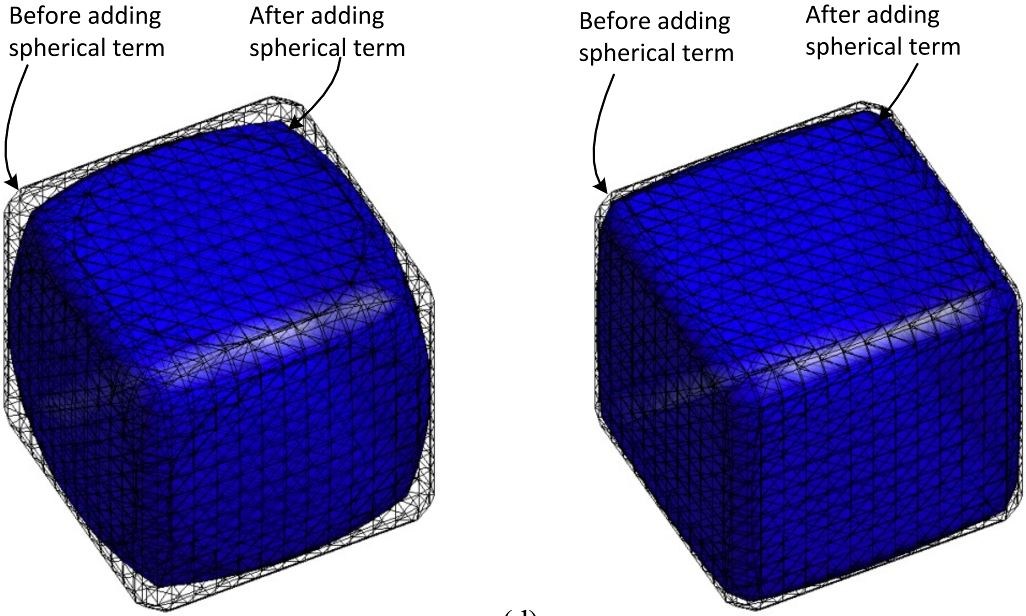
- $f = 0$ defines the particle surface,
- $f < 0$ "inside" the particle,
- $f > 0$ "outside" the particle,
- the particle is strictly convex, and any surface f=*constant* is strictly convex.

For computational reasons, the expression in Eq. (1) is normalised ~~(slightly changing the meaning of *k*)~~:

$$f = (1-k)\left(\sum_{i=1}^{N}\frac{\langle a_i x + b_i y + c_i z - d_i\rangle^2}{r^2} - 1\right) + k\left(\frac{x^2}{R^2} + \frac{y^2}{R^2} + \frac{z^2}{R^2} - 1\right) \tag{2}$$



(a) (b)



(c) (d)

Figure 1 *Construction of potential particles (a) constituent planes are squared and expanded by a constant r. A fraction of sphere is added. Particles with the spherical term are visible in (b) k = 0.9, (c) k =0.7, (d) k = 0.4*

2.2 Transforming the Reference System

Consider two potential particles, particle A $f_A(x_A, y_A, z_A) = 0$ and particle B $f_B(x_B, y_B, z_B) = 0$ defined in their local coordinates (x_A, y_A, z_A) and (x_B, y_B, z_B) respectively. For the purpose of contact detection between a pair of particles, it is necessary to work with the positions and orientations of the particles with respect to the same global coordinate system. A point $\mathbf{x}$ in the global coordinate system can be calculated from the local coordinate system $\mathbf{x}_A$ or $\mathbf{x}_B$ using the following expression:

$$\left.\begin{aligned} \mathbf{x} &= \mathbf{Q}_A \mathbf{x}_A + \mathbf{x}_{0A} \\ \mathbf{x} &= \mathbf{Q}_B \mathbf{x}_B + \mathbf{x}_{0B} \end{aligned}\right\} \quad (3)$$

where $\mathbf{x}_{0A}$ and $\mathbf{x}_{0B}$ are the particle centres of particle A and B respectively, and $\mathbf{Q}_A$ and $\mathbf{Q}_B$ are the rotation matrices which can be derived from the particle orientations with respect to the global reference system. In some 3-D DEM codes such as YADE, the orientations of the particles in 3-D are stored as unit quaternions [12]. The operation to rotate a vector from $\mathbf{x} = (x, y, z)$ to $\mathbf{x}^* = (x^*, y^*, z^*)$ by an angle θ clockwise about a vector with direction cosines (a, b, c) can be expressed as $\mathbf{x}^* = qvq^{-1}$ where q and q^{-1} are unit quaternions defined as $q = (\cos(\theta/2), a\sin(\theta/2)\mathbf{i}, b\sin(\theta/2)\mathbf{j}, c\sin(\theta/2)\mathbf{k})$ and $q^{-1} = (\cos(\theta/2), -a\sin(\theta/2)\mathbf{i}, -b\sin(\theta/2)\mathbf{j}, -c\sin(\theta/2)\mathbf{k})$. Alternatively, this operation can be expressed as a rotation matrix [10]:

$$\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \begin{bmatrix} \cos\theta + a^2F & -c\sin\theta + abF & b\sin\theta + acF \\ c\sin\theta + abF & \cos\theta + b^2F & -a\sin\theta + bcF \\ -b\sin\theta + acF & a\sin\theta + bcF & \cos\theta + c^2F \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4}$$

where $F = 2\sin^2(\theta/2) = 1 - \cos\theta$.

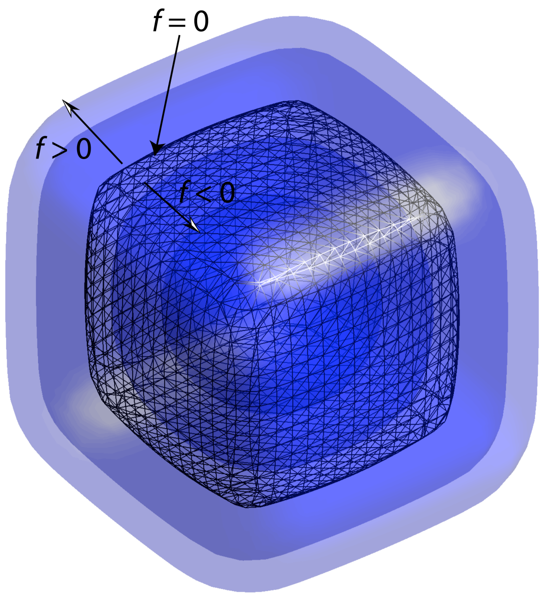


Figure 2 Definition of a potential particle in three-dimension.

2.3 Contact Detection Algorithm

To perform contact detection between a pair of potential particles f_A and f_B, Houlsby [1] proposes that one can solve one of the constrained minimisation problems below:

- minimise f_A subject to the constraint $f_B = 0$
- minimise $f_A + f_B$ subject to the constraint $f_A - f_B = 0$

Here, the second method is adopted, which corresponds to finding a point which is midway and closest to both (with respect to the potential functions of the particles). It is noteworthy that the presence of Macaulay brackets in Eq. (1) results in a discontinuity in the second derivatives which can cause convergence issues in the process of optimisation. Harkness [11] later suggested that the terms consisting of the Macaulay brackets can be raised to a 3rd degree. However, the result of formulating the optimisation problem into a second-order cone program (SOCP) makes this step unnecessary. The i^{th} term in the Macaulay brackets are each replaced with slack variables p_i and inequality constraints:

$$\left.\begin{aligned} & a_i x + b_i y + c_i z - d_i \le p_i \\ & p_i \ge 0 \end{aligned}\right\} \qquad (5)$$

After some algebraic manipulations (see Appendix A), the second-order cone program can be formulated as follows:

$$\left.\begin{aligned} & \text{minimise } s_A + s_B \\ & \text{subject to} \\ & \sqrt{\sum_{i=1}^{N_A} p_{iAk}^2 + x_{Ak}^2 + y_{Ak}^2 + z_{Ak}^2} \le s_A \\ & \sqrt{\sum_{i=1}^{N_B} p_{iBk}^2 + x_{Bk}^2 + y_{Bk}^2 + z_{Bk}^2} \le s_B \end{aligned}\right| \qquad (6)$$

$$s_{\mathrm{A}} = s_{\mathrm{B}}$$

$$w_{\mathrm{As}} x_{\mathrm{Ak}} Q_{\mathrm{A11}} + w_{\mathrm{As}} y_{\mathrm{Ak}} Q_{\mathrm{A12}} + w_{\mathrm{As}} z_{\mathrm{Ak}} Q_{\mathrm{A13}} - \left(w_{\mathrm{Bs}} x_{\mathrm{Bk}} Q_{\mathrm{B11}} + w_{\mathrm{Bs}} y_{\mathrm{Bk}} Q_{\mathrm{B12}} + w_{\mathrm{Bs}} z_{\mathrm{Bk}} Q_{\mathrm{B13}}\right) = x_{0\mathrm{B}} - x_{0\mathrm{A}}$$

$$w_{\mathrm{As}} x_{\mathrm{Ak}} Q_{\mathrm{A21}} + w_{\mathrm{As}} y_{\mathrm{Ak}} Q_{\mathrm{A22}} + w_{\mathrm{As}} z_{\mathrm{Ak}} Q_{\mathrm{A23}} - \left(w_{\mathrm{Bs}} x_{\mathrm{Bk}} Q_{\mathrm{B21}} + w_{\mathrm{Bs}} y_{\mathrm{Bk}} Q_{\mathrm{B22}} + w_{\mathrm{Bs}} z_{\mathrm{Bk}} Q_{\mathrm{B23}}\right) = y_{0\mathrm{B}} - y_{0\mathrm{A}}$$

$$w_{\mathrm{As}} x_{\mathrm{Ak}} Q_{\mathrm{A31}} + w_{\mathrm{As}} y_{\mathrm{Ak}} Q_{\mathrm{A32}} + w_{\mathrm{As}} z_{\mathrm{Ak}} Q_{\mathrm{A33}} - \left(w_{\mathrm{Bs}} x_{\mathrm{Bk}} Q_{\mathrm{B31}} + w_{\mathrm{Bs}} y_{\mathrm{Bk}} Q_{\mathrm{B32}} + w_{\mathrm{Bs}} z_{\mathrm{Bk}} Q_{\mathrm{B33}}\right) = z_{0\mathrm{B}} - z_{0\mathrm{A}}$$

$$w_{\mathrm{As}} a_{i\mathrm{A}} x_{\mathrm{Ak}} + w_{\mathrm{As}} b_{i\mathrm{A}} y_{\mathrm{Ak}} + w_{\mathrm{Ak}} c_{i\mathrm{A}} z_{\mathrm{Ak}} - w_{\mathrm{Ap}} p_{i\mathrm{Ak}} \le d_{i\mathrm{A}}, \quad i = 1, \ldots, N_{\mathrm{A}},$$

$$w_{\mathrm{Bs}} a_{i\mathrm{B}} x_{\mathrm{Bk}} + w_{\mathrm{Bs}} b_{i\mathrm{B}} y_{\mathrm{Bk}} + w_{\mathrm{Bk}} c_{i\mathrm{B}} z_{\mathrm{Bk}} - w_{\mathrm{Bp}} p_{i\mathrm{Bk}} \le d_{i\mathrm{B}}, \quad i = 1, \ldots, N_{\mathrm{B}},$$

$$s_{\mathrm{A}} \ge 0$$

$$s_{\mathrm{B}} \ge 0$$

where:

$$\left.\begin{aligned} w_{\mathrm{Ap}} &= \frac{r_{\mathrm{A}}}{\sqrt{1-k_{\mathrm{A}}}} \\ w_{\mathrm{As}} &= \frac{R_{\mathrm{A}}}{\sqrt{k_{\mathrm{A}}}} \\ w_{\mathrm{Bp}} &= \frac{r_{\mathrm{B}}}{\sqrt{1-k_{\mathrm{B}}}} \\ w_{\mathrm{Bs}} &= \frac{R_{\mathrm{B}}}{\sqrt{k_{\mathrm{B}}}} \end{aligned}\right\} \quad (7)$$

Note that the variables with subscript *k* are related to the original variables in Eq. (2), (3), (4) and (5) through:

$$\left.\begin{aligned} p_{i\mathrm{Ak}} &= \frac{\sqrt{1-k_{\mathrm{A}}}}{r_{\mathrm{A}}} p_{i\mathrm{A}} \\ x_{\mathrm{Ak}} &= \frac{\sqrt{k_{\mathrm{A}}}}{R_{\mathrm{A}}} x_{\mathrm{A}} \end{aligned}\right\} \quad (8)$$

$$y_{\mathrm{Ak}} = \frac{\sqrt{k_{\mathrm{A}}}}{R_{\mathrm{A}}} y_{\mathrm{A}}$$

$$z_{\mathrm{Ak}} = \frac{\sqrt{k_{\mathrm{A}}}}{R_{\mathrm{A}}} z_{\mathrm{A}}$$

Eq. (6) can be input directly into second-order cone optimisers such as MOSEK [14]. There is overlap if both $s_{\mathrm{A}} < 1.0$ and $s_{\mathrm{B}} < 1.0$. Notice that a linear inequality is introduced for every plane "*i*". If there is overlap, the optimal point (x_{Ak}^*, y_{Ak}^*, z_{Ak}^*) has to be transformed back to the original local coordinates of particle A (x_{A}^*, y_{A}^*, z_{A}^*) using Eq. (8). Thereafter, one can find the global Cartesian coordinates using Eq. (4). This point is then used as the contact point, i.e, the point at which contact forces are applied between two particles.

2.4 Calculating the Contact Normal

For particles of equal stiffness, the unit vector identifying the direction of the plane of contact (i.e. the normal to the plane) can be calculated as the average between the two unit normal vectors of the two interacting particles. ~~The contact normal can be assigned as a weighted average of the normal vectors of the interacting particles at the contact point based on the particle stiffnesses.~~ For each particle, the normal vector has been calculated at the point of contact with the other particle. In local coordinates, the normal vector of a particle can be calculated as:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \tag{6}$$

where

$$\left.\begin{aligned}
\frac{\partial f}{\partial x} &= \frac{2(1-k)}{r^2}\sum_{i=1}^{\mathrm{N}} a_i \left\langle a_i x + b_i y + c_i z - d_i \right\rangle + \frac{2k}{R^2} x \\
\frac{\partial f}{\partial y} &= \frac{2(1-k)}{r^2}\sum_{i=1}^{\mathrm{N}} b_i \left\langle a_i x + b_i y + c_i z - d_i \right\rangle + \frac{2k}{R^2} y \\
\frac{\partial f}{\partial z} &= \frac{2(1-k)}{r^2}\sum_{i=1}^{\mathrm{N}} c_i \left\langle a_i x + b_i y + c_i z - d_i \right\rangle + \frac{2k}{R^2} z
\end{aligned}\right\} \quad (7)$$

The normal vector can be transformed into global coordinates using Eq.(4). The overlap distance can be found by performing a line search along the contact normal and bracketing two points, i.e., one on particle A ($f_A = 0$) and the other on particle B ($f_B = 0$) (see Figure 3). The overlap distance is the distance between these two points.

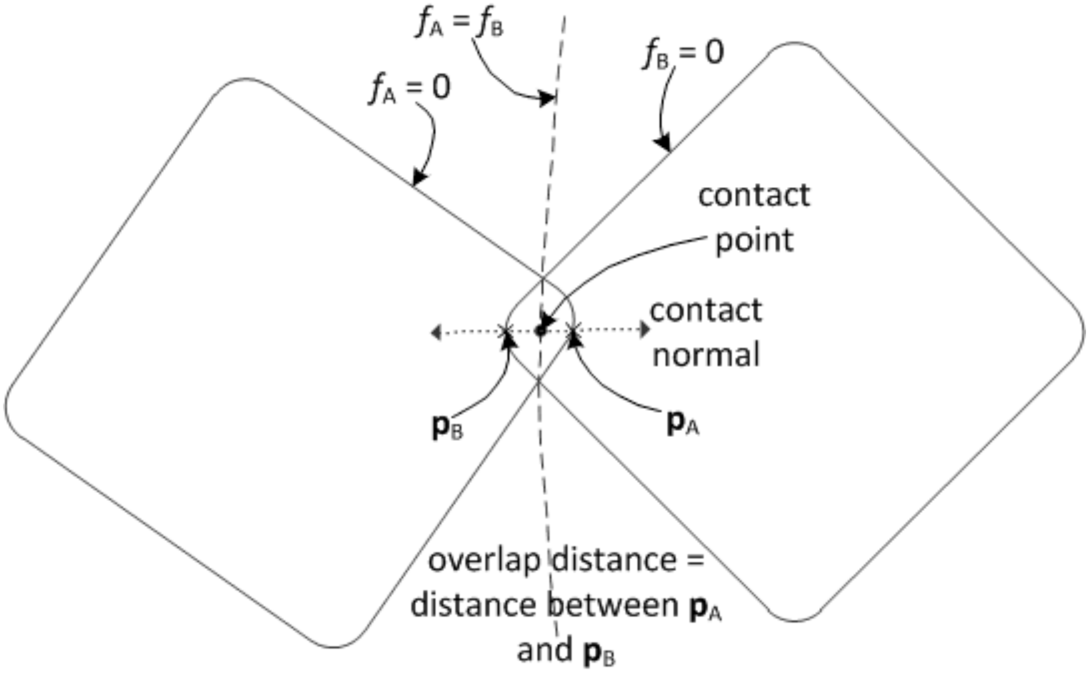


Figure 3 *Schematic of overlapping potential particles. Overlap is exaggerated for the purpose of illustration. 2-D polygons are plotted for sake of explanation.*

3 EXAMPLES

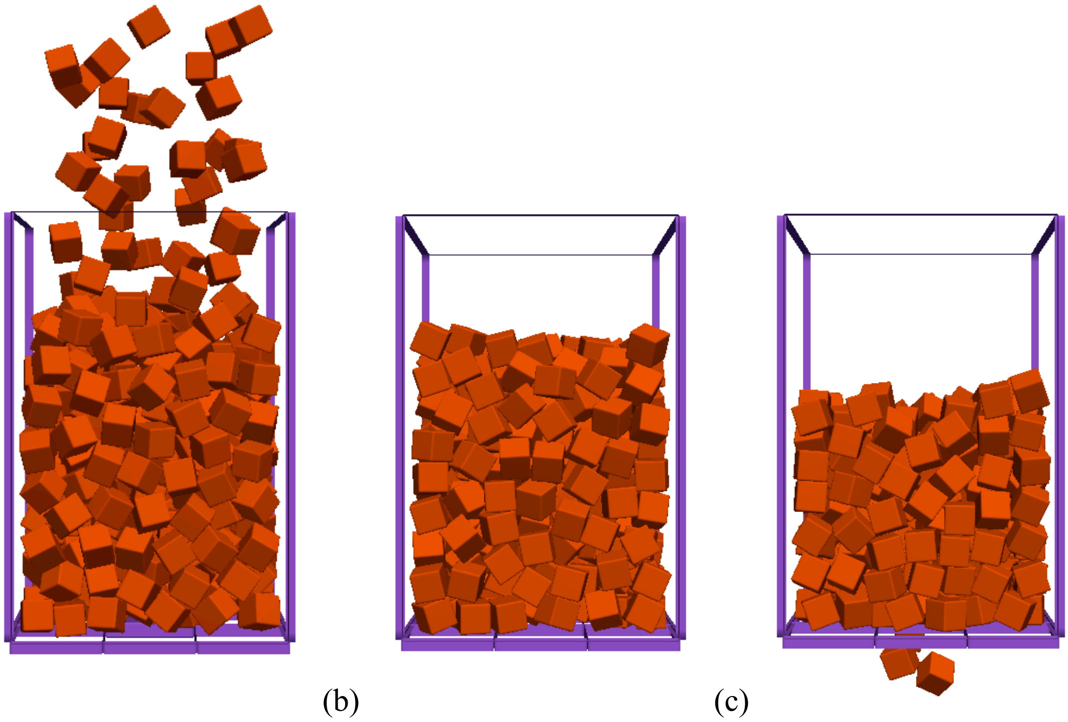
To illustrate the capability and robustness of the proposed contact detection algorithm, some example simulations were run using the open-source discrete element code YADE [12]. The second-order cone program (SOCP) was solved using the conic optimiser MOSEK [13, 14]. For every potential contact, MOSEK was called as an external library in a routine in YADE, by specifying inputs which consists of the objective function and constraints. The solution calculated by MOSEK was then used as the contact point.

3.1 TEST A

In the first simulation, 360 cubes were generated with random orientations. Subsequently, they were allowed to fall under gravity impacting the base of a prismatic container (see Figure 4(a)). All the particles were assumed to be frictionless. In this example, a combination of several contact conditions, involving angular corners, angular edges and roughly flat surfaces is present throughout the simulation so that the robustness of the algorithm can be tested. Once the cubes have settled (Figure 4(b)), an orifice at the base of the container was opened (Figure 4(c)). The size of the orifice was 3×3 times the edge length of the cubes, while the size of the base was 9×9 times the edge length of the cubes. The simulation was repeated with tetrahedral particles (see Figure 5), whose size was chosen as to be tightly inscribed in the cubes. The volume of these tetrahedra is one-third that of the circumscribing cubes, and their edge length is $\sqrt{2}$ times the length of the cubes. The adopted contact law in the normal direction is linear elasticity (elastic spring acting only in compression) whereas in the shear direction is linearly elastic-perfectly plastic (elastic spring plus a frictional slider). The contact stiffness in both directions has been assumed to be 1 GN/m. In the performed numerical experiments, the density of the particles was scaled to 10000 kg/m^3. ~~The density of the tetrahedron was assumed to be 3 times the cube density so that they have the same mass.~~ Table 1 summarises the parameters used in this test. The flow of the particles over time is shown in Figure 6. The simulation correctly shows that the flow rates of particles through an orifice are influenced by their shapes; an inaccurate algorithm is likely to have resulted in similar flow rates between shapes if the same particle size is modelled. The difference between the deposition levels after settling (before the orifice is opened) is also captured realistically by the contact detection algorithm (see Figure 4 (b) and Figure 5 (b)); note that the volume ratio for a tetrahedron inscribed in a cube is 1:3.

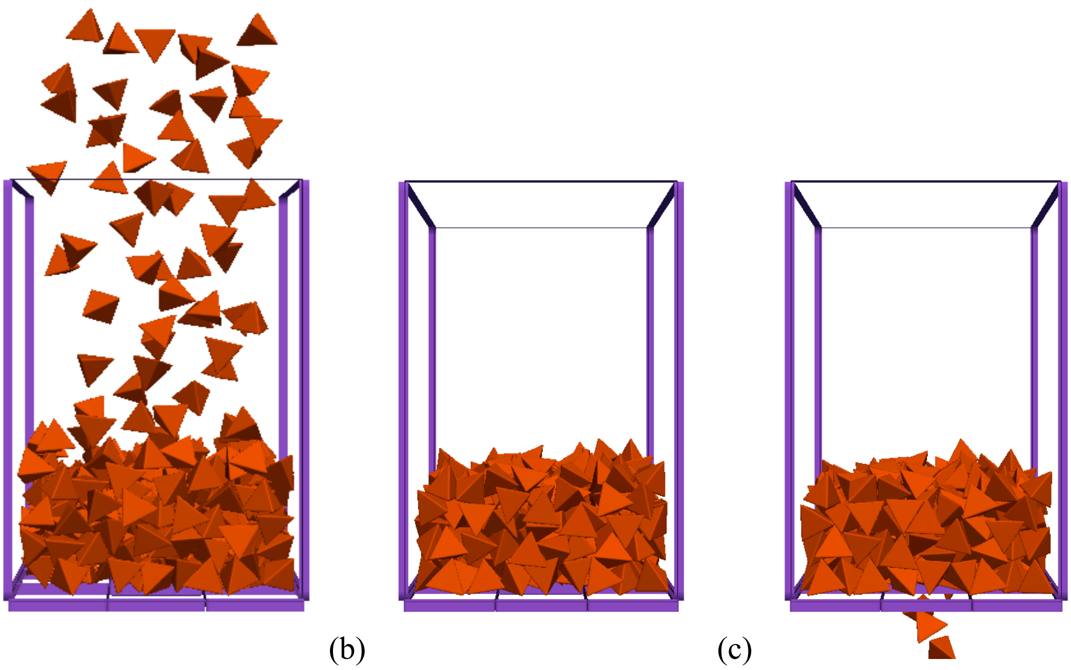
TABLE 1: Parameters for Test A

Parameters	Values
Density	10000 kg/m^3
Normal stiffness	1 GN/m
Shear stiffness	1 GN/m
Friction angle of particles and containers	0°
Container dimension	9 m × 9 m × 14 m
Orifice dimension	3 m × 3m
Cube dimension	1 m × 1m × 1 m



(a) (b) (c)

Figure 4 *Simulations of cube-shaped particles (a) filling the container (b) settling and (c) flowing through the orifice*



(a) (b) (c)

Figure 5 *Simulations of tetrahedral-shaped particles (a) filling the container (b) settling and (c) flowing through the orifice*

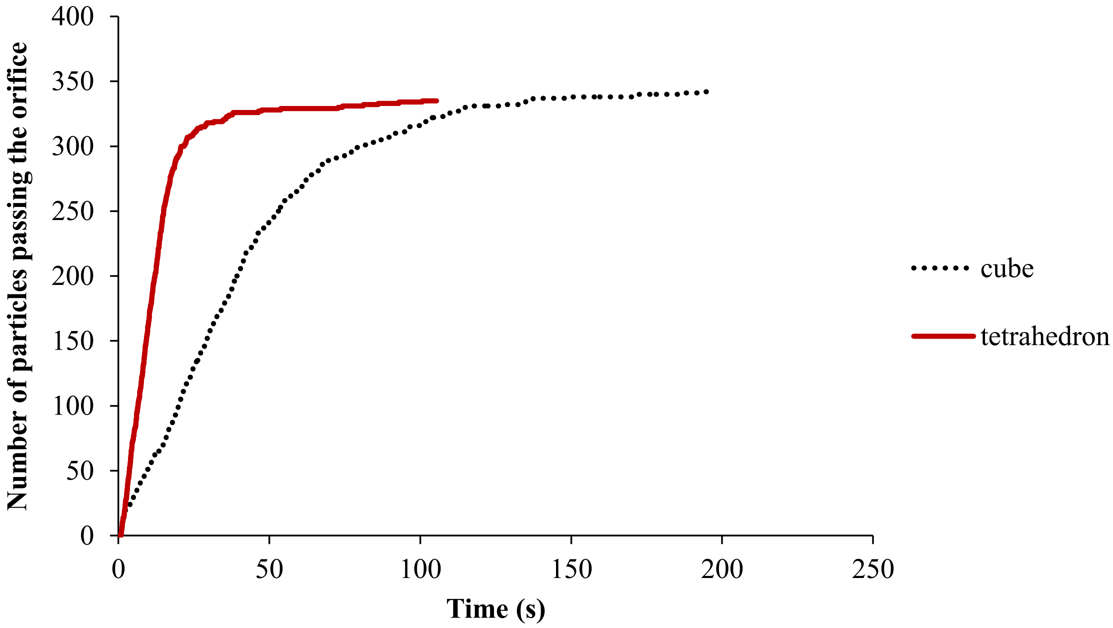
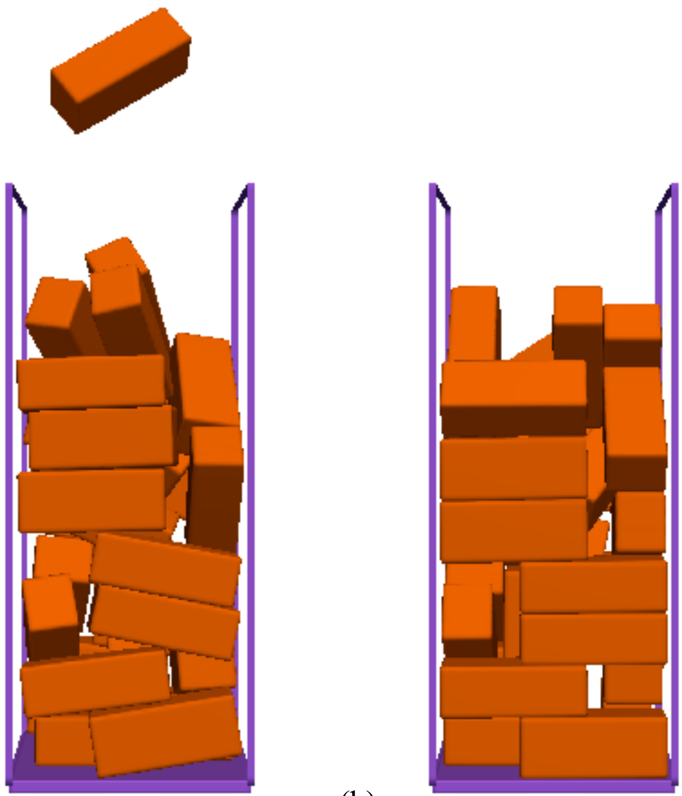


Figure 6 *Discharge flow of particles through the orifice over time. It should be noted that t=0 in this figure is the time when the orifice is opened, not the start of the simulation*

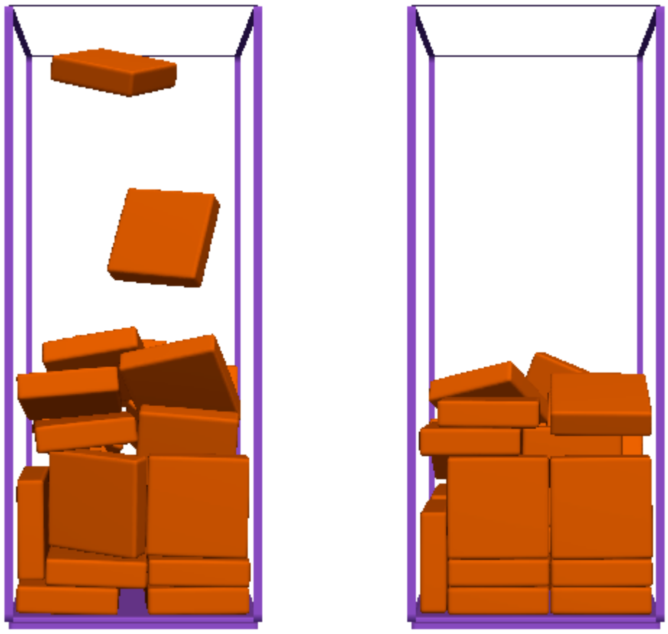
3.3 TEST B

In this test, simulations were carried out using frictionless particles of high aspect ratios. In the first test, the prisms have aspect ratios of 1:3. In the second test, the prisms have aspect ratios of 1:8. First, like in Test A, the particles were generated with random orientations and allowed to fall under gravity impacting each other inside a container. The density and contact stiffness of the particles were the same as in Test A. Figure 7 (a) and Figure 8 (a) show the particles falling under gravity and dynamically re-orienting themselves in the container. Figure 7 (b) and Figure 8 (b) show the configurations of the particles after they have settled. These particles re-aligned nicely with the container, showing that the algorithm is able to model particles of high aspect ratios realistically. The results conform to physical experience.



(a) (b)

Figure 7 *Simulations of prisms of aspect ratio 1:3 (a) filling the container and dynamically changing positions (b) after settling. Some particles are leaning against the front wall of the container (transparent in this figure). The accuracy of the figure is limited by the size of tessellations of the visualisation tool.*



(a) (b)

Figure 8 *Simulations of prisms of aspect ratio 1:8 (a) filling the container and dynamically changing positions (b) after settling. Some particles are leaning against the front wall of the container (transparent in this figure). The accuracy of the figure is limited by the size of tessellations of the visualisation tool.*

3.3 TEST C

The conic optimisation formulation in Eq. (6) can be solved using a variety of numerical techniques. The computation time for contact detection depends on the details of the numerical technique used to solve the optimisation problem. From our experience, primal-dual interior-point methods which can take advantage of second-order cone constraints are robust, e.g. MOSEK [14] and CPLEX [15]. But the same formulation can be solved using other general non-linear optimisation software [18]. The choice of the optimisation software depends on its compatibility with the DEM code in terms of programming language, operating system, cost of the licence and compiler version restrictions.

The computation time also depends on the termination criteria that are set for the optimisation task. Most well-developed optimisation softwares make use of more than two termination criteria and offer a range of "refinement" parameters. These termination criteria and "refinement" parameters are normally different between optimisation softwares since different optimisation techniques are used. Values for the termination criteria are set based on the accuracy desired by the users. Note, however, that even considering the same software, the numerical values of the termination criteria are not universal because certain particle shapes may be more sensitive to the criteria than others.

For quasi-static problems, the contact point for a pair of particles in contact may be very close to the contact point in the previous time step. With a good starting point, warm-starting allows the solver to take less Newton steps to satisfy the same termination criteria. It is worth noting that although primal-dual interior point methods are preferred in the optimisation literature to solve second-order cone programs because of their efficiency and robustness, they do not allow warm-starting, i.e. they cannot make use of user-supplied starting point information. If the modeller wishes to warm-start, the conventional primal barrier method can be used to solve the second-order cone program [16]. Depending on the experience of the modeller, he may wish to program his own Newton method for the optimisation problem to allow more flexibility in fine-tuning the parameters for the solver, e.g. aggressiveness (increment of penalty parameters), convergence tolerances, and starting point strategies. Another convenient way to warm-start is to use general non-linear optimisation softwares which can make use of user-supplied starting point information. Since different solvers (or strategies) may have different performances in terms of speed and robustness, it is recommended that more than one solver is employed in the DEM code of interest. Different solvers can be called under different circumstances, depending on the strategy of the modeller. The fine-tuning strategies usually relate to the experience of the modeller. In general, the computation time increases with the strictness of the termination criteria (normally at the expenses of accuracy) and reduces with the "tuning" aggressiveness (normally at the expenses of robustness). The overall run-time of a DEM calculation further depends on the type of simulation and its parameters which are likely to affect the number of "fortuitous encounters" of good starting points. It is also affected inherently by the particle shape; certain shapes experience higher coordination numbers (number of contacts per particle) and certain shapes can be more efficiently inscribed inside axes-aligned bounding boxes or spheres which are used before the actual contact resolution stage.

As an example, we show the computation time to solve Eq. (6) for a pair of particles in contact (see Figure 9) using MOSEK and the primal barrier method code which can be downloaded from [17]. In the primal barrier solver, we have substituted the equality constraints into the objective and constraint functions (refer to Eq. (6)) so that the equations are solved in terms of global coordinates rather than using two sets of local coordinates. Note that the formulation in Eq. (6) is proposed here because it is accepted by the majority of conic or non-linear optimisation solvers; certain conic optimisation software may impose restrictions on the mathematical expressions of the second-order cones, e.g. CPLEX and MOSEK [14, 15]. In the first simulation (refer to Figure 9), we used the primal barrier method for contact detection. These particles were fixed in space. Using one of the two cores of the Intel Core 2 Duo processor, the computation time for the barrier method with warm-starting was 366 µs; default values in [17] for the penalty increment parameter and termination criteria were used. Using a more aggressive penalty parameter with the same termination criteria, the computation time was 48 µs. In these two barrier calculations, we have used the contact point calculated at the previous time-step as the starting point. At the starting point, we have chosen the slack variables *s*'s and *p*'s in Eq. (6) such that the inequalities are satisfied to within a margin of 10-^{5}. Using exact values without perturbation may cause numerical difficulties since the inequalities are modelled inside log functions in the barrier method. These implementation details will vary with the type of numerical technique. The computation time for MOSEK using its default termination criterion was 428 µs. Table 2 shows the results of this exercise.

TABLE 2: Computation time comparison between choices of solvers

	Spheres	Computation time with non-spherical particles in Figure 9		
		Primal barrier method [17] with tuning	Primal barrier method [17] without tuning	Primal-dual interior point method (MOSEK)
Computation time per contact between two particles	0.1 µs	48 µs	366 µs	428 µs

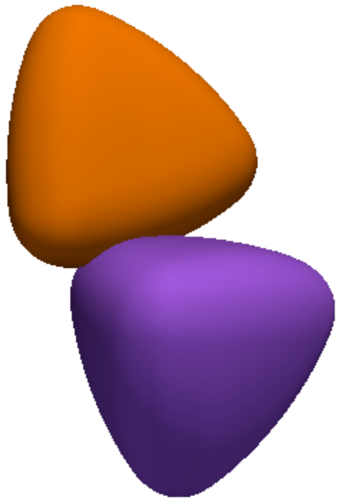


Figure 9 *Two rounded tetrahedral particles in contact*

4 CONCLUSIONS

The mathematics for the contact detection between potential particles in 3-D is presented. The optimisation problem was cast into a second-order cone program which is generally held to be one of the most robust formulations in the field of convex optimisation. Simulations were run to test the robustness and capability of the contact detection algorithm. An example involved roughly angular particles settling into a prismatic container. However, any convex particle could have been used. A wide range of contact types involving angular corners, angular edges and roughly flat surfaces were tested in this example. Then, the particles were allowed to flow through an orifice under gravity. Particles with high aspect ratios were also modelled falling and settling into a container. They were able to realign nicely among themselves inside the container upon settling. In the paper, it has been shown that potential particles together with the proposed contact detection algorithm can be used to model non-spherical particles for engineering applications. The advantage of this method is that it can model any convex shape from rounded to roughly polyhedral, and can be solved using ubiquitous optimisation software.

5 ACKNOWLEDGEMENTS

Erling Andersen from MOSEK is thanked for highlighting that the Macaulay brackets can be replaced with auxiliary variables and inequality constraints.

APPENDIX A: Derivation of the second order cone program (SOCP)

Consider the optimisation problem:

$$\left.\begin{array}{c} \text{minimise } f_{\mathrm{A}} + f_{\mathrm{B}} \\ \text{subject to} \\ f_{\mathrm{A}} = f_{\mathrm{B}} \end{array}\right\} \quad \text{(A.1)}$$

where f_{A} and f_{B} are the potential functions of Particle A and B which according to the definition in (2) can be expressed as:

$$\left.\begin{array}{l} f_{\mathrm{A}} = \dfrac{(1-k_{\mathrm{A}})}{r_{\mathrm{A}}^2}\left(\sum_{i=1}^{N_{\mathrm{A}}} \langle a_{i\mathrm{A}} x_{\mathrm{A}} + b_{i\mathrm{A}} y_{\mathrm{A}} + c_{i\mathrm{A}} z_{\mathrm{A}} - d_{i\mathrm{A}} \rangle^2 - r_{\mathrm{A}}^2\right) + \dfrac{k_{\mathrm{A}}}{R_{\mathrm{A}}^2}(x_{\mathrm{A}}^2 + y_{\mathrm{A}}^2 + z_{\mathrm{A}}^2 - R_{\mathrm{A}}^2) \\ \\ f_{\mathrm{B}} = \dfrac{(1-k_{\mathrm{B}})}{r_{\mathrm{B}}^2}\left(\sum_{i=1}^{N_{\mathrm{B}}} \langle a_{i\mathrm{B}} x_{\mathrm{B}} + b_{i\mathrm{B}} y_{\mathrm{B}} + c_{i\mathrm{B}} z_{\mathrm{B}} - d_{i\mathrm{B}} \rangle^2 - r_{\mathrm{B}}^2\right) + \dfrac{k_{\mathrm{B}}}{R_{\mathrm{B}}^2}(x_{\mathrm{B}}^2 + y_{\mathrm{B}}^2 + z_{\mathrm{B}}^2 - R_{\mathrm{B}}^2) \end{array}\right\} \quad \text{(A.2)}$$

where $(x_{\mathrm{A}}, y_{\mathrm{A}}, z_{\mathrm{A}})$ and $(x_{\mathrm{B}}, y_{\mathrm{B}}, z_{\mathrm{B}})$ are the local coordinates with respect to Particle A and B respectively. It is convenient to optimise over the global Cartesian coordinate system, so that:

$$\mathbf{Q}_{\mathrm{A}}\mathbf{x}_{\mathrm{A}} + \mathbf{x}_{\mathbf{0}\mathrm{A}} = \mathbf{Q}_{\mathrm{B}}\mathbf{x}_{\mathrm{B}} + \mathbf{x}_{\mathbf{0}\mathrm{B}} \quad \Rightarrow \quad \mathbf{Q}_{\mathrm{A}}\mathbf{x}_{\mathrm{A}} - \mathbf{Q}_{\mathrm{B}}\mathbf{x}_{\mathrm{B}} = \mathbf{x}_{\mathbf{0}\mathrm{B}} - \mathbf{x}_{\mathbf{0}\mathrm{A}} \quad \text{(A.3)}$$

where $\mathbf{x}_\mathrm{A} = (x_\mathrm{A}, y_\mathrm{A}, z_\mathrm{A})$ and $\mathbf{x}_\mathrm{B} = (x_\mathrm{B}, y_\mathrm{B}, z_\mathrm{B})$ while $\mathbf{x}_{\mathbf{0}\mathrm{A}}$ and $\mathbf{x}_{\mathbf{0}\mathrm{B}}$ denote the positions of Particle A and B. $\mathbf{Q}_\mathrm{A}$ and $\mathbf{Q}_\mathrm{B}$ are rotation matrices which can be used to transform vectors from the local reference systems of the particles to the global coordinate system.

Recalling that $\langle\ \rangle$ in (A.2) are Macaulay brackets, i.e., $\langle x\rangle = x$ for $x > 0$; $\langle x\rangle = 0$ for $x \le 0$. For the purpose of minimisation, the Macaulay brackets can be replaced with auxiliary slack variables p_i and adding additional constraints so that:

$$\left.\begin{aligned}
&f_\mathrm{A} = \frac{(1-k_\mathrm{A})}{r_\mathrm{A}^2}\left(\sum_{i=1}^{N_\mathrm{A}} p_{i\mathrm{A}}^2 - r_\mathrm{A}^2\right) + \frac{k_\mathrm{A}}{R_\mathrm{A}^2}(x_\mathrm{A}^2 + y_\mathrm{A}^2 + z_\mathrm{A}^2 - R_\mathrm{A}^2)\\
&a_{i\mathrm{A}} x_\mathrm{A} + b_{i\mathrm{A}} y_\mathrm{A} + c_{i\mathrm{A}} z_\mathrm{A} - d_{i\mathrm{A}} \le p_{i\mathrm{A}}, \quad i = 1,\dots,N_\mathrm{A},\\
&p_{i\mathrm{A}} \ge 0, \quad i = 1,\dots,N_\mathrm{A},
\end{aligned}\right\} \tag{A.4}$$

By further introducing:

$$\left.\begin{aligned}
p_{i\mathrm{Ak}} &= \frac{\sqrt{1-k_\mathrm{A}}}{r_\mathrm{A}} p_{i\mathrm{A}}, \quad i = 1,\dots,N_\mathrm{A},\\
x_\mathrm{Ak} &= \frac{\sqrt{k_\mathrm{A}}}{R_\mathrm{A}} x_\mathrm{A}\\
y_\mathrm{Ak} &= \frac{\sqrt{k_\mathrm{A}}}{R_\mathrm{A}} y_\mathrm{A}\\
z_\mathrm{Ak} &= \frac{\sqrt{k_\mathrm{A}}}{R_\mathrm{A}} z_\mathrm{A}
\end{aligned}\right\} \tag{A.5}$$

the potential function can be expressed in terms of these new variables:

$$f_\mathrm{A} = \sum_{i=1}^{N} p_{i\mathrm{Ak}}^2 + x_\mathrm{Ak}^2 + y_\mathrm{Ak}^2 + z_\mathrm{Ak}^2 - 1 \tag{A.6}$$

Introducing slack variables s_{A} and s_{B} with $s_{\mathrm{A}} \geq 0$ and $s_{\mathrm{B}} \geq 0$, and the constants $w_{\mathrm{Ap}}, w_{\mathrm{As}}$, w_{Bp} and w_{Bs}:

$$\left.\begin{aligned}
w_{\mathrm{Ap}} &= \frac{r_{\mathrm{A}}}{\sqrt{1-k_{\mathrm{A}}}} \quad \text{(planar component of particle A)} \\
w_{\mathrm{As}} &= \frac{R_{\mathrm{A}}}{\sqrt{k_{\mathrm{A}}}} \quad \text{(spherical component of particle A)} \\
w_{\mathrm{Bp}} &= \frac{r_{\mathrm{B}}}{\sqrt{1-k_{\mathrm{B}}}} \quad \text{(planar component of particle B)} \\
w_{\mathrm{Bs}} &= \frac{R_{\mathrm{B}}}{\sqrt{k_{\mathrm{B}}}} \quad \text{(spherical component of particle B)}
\end{aligned}\right\} \tag{A.7}$$

we can express the optimisation problem as a second order cone program:

$$\left.\begin{aligned}
&\text{minimise } s_{\mathrm{A}}+s_{\mathrm{B}} \\
&\text{subject to} \\
&\sqrt{\sum_{i=1}^{N_{\mathrm{A}}} p_{i\mathrm{Ak}}^2 + x_{\mathrm{Ak}}^2 + y_{\mathrm{Ak}}^2 + z_{\mathrm{Ak}}^2} \leq s_{\mathrm{A}} \\
&\sqrt{\sum_{i=1}^{N_{\mathrm{B}}} p_{i\mathrm{Bk}}^2 + x_{\mathrm{Bk}}^2 + y_{\mathrm{Bk}}^2 + z_{\mathrm{Bk}}^2} \leq s_{\mathrm{B}} \\
&s_{\mathrm{A}} = s_{\mathrm{B}} \\
&w_{\mathrm{As}} x_{\mathrm{Ak}} Q_{\mathrm{A11}} + w_{\mathrm{As}} y_{\mathrm{Ak}} Q_{\mathrm{A12}} + w_{\mathrm{As}} z_{\mathrm{Ak}} Q_{\mathrm{A13}} - \left(w_{\mathrm{Bs}} x_{\mathrm{Bk}} Q_{\mathrm{B11}} + w_{\mathrm{Bs}} y_{\mathrm{Bk}} Q_{\mathrm{B12}} + w_{\mathrm{Bs}} z_{\mathrm{Bk}} Q_{\mathrm{B13}}\right) = x_{0\mathrm{B}} - x_{0\mathrm{A}} \\
&w_{\mathrm{As}} x_{\mathrm{Ak}} Q_{\mathrm{A21}} + w_{\mathrm{As}} y_{\mathrm{Ak}} Q_{\mathrm{A22}} + w_{\mathrm{As}} z_{\mathrm{Ak}} Q_{\mathrm{A23}} - \left(w_{\mathrm{Bs}} x_{\mathrm{Bk}} Q_{\mathrm{B21}} + w_{\mathrm{Bs}} y_{\mathrm{Bk}} Q_{\mathrm{B22}} + w_{\mathrm{Bs}} z_{\mathrm{Bk}} Q_{\mathrm{B23}}\right) = y_{0\mathrm{B}} - y_{0\mathrm{A}} \\
&w_{\mathrm{As}} x_{\mathrm{Ak}} Q_{\mathrm{A31}} + w_{\mathrm{As}} y_{\mathrm{Ak}} Q_{\mathrm{A32}} + w_{\mathrm{As}} z_{\mathrm{Ak}} Q_{\mathrm{A33}} - \left(w_{\mathrm{Bs}} x_{\mathrm{Bk}} Q_{\mathrm{B31}} + w_{\mathrm{Bs}} y_{\mathrm{Bk}} Q_{\mathrm{B32}} + w_{\mathrm{Bs}} z_{\mathrm{Bk}} Q_{\mathrm{B33}}\right) = z_{0\mathrm{B}} - z_{0} \\
&w_{\mathrm{As}} a_{i\mathrm{A}} x_{\mathrm{Ak}} + w_{\mathrm{As}} b_{i\mathrm{A}} y_{\mathrm{Ak}} + w_{\mathrm{Ak}} c_{i\mathrm{A}} z_{\mathrm{Ak}} - w_{\mathrm{Ap}} p_{i\mathrm{Ak}} \leq d_{i\mathrm{A}}, \quad i = 1,\ldots, N_{\mathrm{A}}, \\
&w_{\mathrm{Bs}} a_{i\mathrm{B}} x_{\mathrm{Bk}} + w_{\mathrm{Bs}} b_{i\mathrm{B}} y_{\mathrm{Bk}} + w_{\mathrm{Bk}} c_{i\mathrm{B}} z_{\mathrm{Bk}} - w_{\mathrm{Bp}} p_{i\mathrm{Bk}} \leq d_{i\mathrm{B}}, \quad i = 1,\ldots, N_{\mathrm{B}},
\end{aligned}\right\} \tag{A.8}$$

$$p_{i\mathrm{Ak}} \geq 0, \qquad i = 1,\ldots,N_{\mathrm{A}},$$

$$p_{i\mathrm{Bk}} \geq 0, \qquad i = 1,\ldots,N_{\mathrm{B}},$$

$$s_{\mathrm{A}} \geq 0$$

$$s_{\mathrm{B}} \geq 0$$

where the constants The last two constraints $p_{i\mathrm{Ak}} \geq 0$ and $p_{i\mathrm{Bk}} \geq 0$ in (A.8) can be omitted from the formulation because they are minimised over their squared values. For any point in which they are negative, they will assume the value 0 since their quadratic expressions in the cones are minimised. Further, because MOSEK does not allow variables to be repeated in separate cones (s_{A} and s_{B} in our case), the linear constraint $s_{\mathrm{A}} = s_{\mathrm{B}}$ has to be specified. In other optimisation codes, one can remove this linear constraint and replace s_{A} and s_{B} using the same variable.

References

1 G. T. Houlsby, Potential particles: a method for modelling non-circular particles in DEM, Computers and Geotechnics 36 (6) (2009) 953-959.

2 P. A. Cundall, Formulation of a three-dimensional distinct element model--Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks, International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstract 25 (3) (1988) 107-116.

3 E. G. Nezami, Y. M. A. Hashash, D. Zhao, J. Ghaboussi, A fast contact detection algorithm for 3-D discrete element method, Computers and Geotechnics, 31 (7) (2004) 575-587.

4 E. G. Nezami, Y. M. A. Hashash, D. Zhao, J. Ghaboussi, Shortest link method for contact detection in discrete element method, International Journal for Numerical and Analytical Methods in Geomechanics, 30 (8) (2006) 783-801.

5 F. Y. Fraige, P. A. Langston, G. Z. Chen, Distinct element modelling of cubic particle packing and flow, Powder Technology, 186 (2008) 224-240.

6 S. W. Chang and C. S. Chen, A non-iterative derivation of the common plane for contact detection of polyhedral blocks, International Journal for Numerical Methods in Engineering 74 (2008) 734-753.

7 L. Pournin, T. M. Liebling, in Powders and Grains, eds. R. García-Rojo, H. J. Herrmann, S. McNamara, Stuttgart, (2005) 1375-1378.

8 P.W. Cleary, M.L. Sawley. DEM modelling of industrial hopper flows: 3D case study and the effect of particle shape on hopper discharge. Applied mathematical modelling, 26 (2) (2002) 89-111.
9 C.W. Boon, G. T. Houlsby, S. Utili. A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method, Computers and Geotechnics, 44 (2012) 73-82.
10 J. B. Kuipers, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, Princeton University Press, (2002).
11 J. Harkness, Potential particles for the modeling of interlocking media in three dimensions, International Journal for Numerical Methods in Engineering, 80 (12) (2009) 1573-1594.
12 J. Kozicki F. V. Donzé, A new open-source software developed for numerical simulations using discrete modeling methods. Computer Methods in Applied Mechanics and Engineering, 197 (2008) 4429-4443.
13 E. D. Anderson, C. Roos, T. Terlaky, On implementing a primal-dual interior point method for conic quadratic optimization, Mathematical Programming, 95 (2) (2003) 249-277.
14 MOSEK, The Optimisation Tools Manual: MOSEK ApS, (2010).
15 CPLEX, User's manual: IBM ILOG CPLEX 9.0 (2003).
16 S.P. Boyd, S.P., L. Vandenberghe, Convex Optimization, Cambridge University Press, (2004) 1-716.
17 S.P. Boyd, Website for source code examples in Convex Optimization [16] http://www.stanford.edu/~boyd/cvxbook/cvxbook_examples/ (2004)
18 A. Wächter, & L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming, 106 (1) (2006) 25-57.
19 C.-Y. Wu, A.C.F. Cocks, Numerical and experimental investigations of the flow of powder into a confined space, Mechanics of Materials 38 (2006) 304 – 324.
20 S. Mack, P. Langston, C. Webb, T. York, Experimental validation of polyhedral discrete element model, 214 (2011) 431-442.