Computers and Geotechnics 121 (2020) 103459

Contents lists available at ScienceDirect

Computers and Geotechnics

journal homepage: www.elsevier.com/locate/compgeo

Research Paper

Discontinuous deformation analysis for ellipsoids using cone complementary formulation

Huo Fan[a], Duruo Huang[b,*], Gang Wang[a,*], Feng Jin[b]

[a] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[b] Department of Hydraulic Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Discontinuous deformation analysis
Ellipsoid
Semi-analytical geometric iteration
Frictional contact
Cone complementarity

ABSTRACT

A new computational framework for discontinuous deformation analysis (DDA) of ellipsoidal particles is developed by taking full advantage of the geometric convexity of the ellipsoid. To identify the contact points and contact directions between ellipsoidal bodies analytically and efficiently, a semi-analytic geometry iteration (SAGI) algorithm is proposed based on parametric equation of the ellipsoid, which is also extended to cylindrical and conical boundaries. The controlling equations of motion of the bodies are established in the context of discontinuous deformation analysis. To reinforce the constraint of frictional contacts in the discrete system, a linearized cone complementarity formulation is proposed to solve the contact forces using a fixed point iteration algorithm, which is the key ingredient to conserve energy, linear and angular momentums in the new numerical framework. The accuracy, computational efficiency, and application prospects of the proposed methodologies are demonstrated through some numerical examples.

1. Introduction

To numerically simulate a discrete system containing a large number of discrete bodies, the geometry description of bodies, governing equations of motions, and treatment of contact between bodies play a very important role. As a typical representation of discontinuum-based methods, the discontinuous deformation analysis [1] (DDA) regards individual body as a geometrically independent element. And, it is based on the principle of minimum potential energy to set up governing equations of motion of the bodies, such that the translation, rotation, contact, and separation of these discrete bodies can be captured straightforwardly by the DDA. Due to these attractive features, the DDA have been a widely recognized especially in the fields of geotechnical engineering and computational geomechanics.

After more than 30 years, the method has been significantly improved in many aspects, including correction of false volume expansion of blocks in large rotation [2–5], enhancement of the deformation analysis and stress field within the bodies [6–10], improvement of vertex-vertex contact [11–14], open-close iteration [15] and contact force calculation between bodies [16,17]. The method is also implemented for coupling of fluid and solid [18,19]. In addition, particle shapes, such as disk [20–22], sphere [23], an assembly of spheres, cylinders and cones [24], has been modeled by the DDA. Most recently, the spherical harmonic function is employed to describe complex shaped particles [25], such that micromorphology of particles can be considered in the DDA analysis.

To further improve geometry modeling in DDA, rigid ellipsoidal particles are introduced for the first time in this study, rendering the commonly used spherical particles as its special case. Specifically, the principle of minimum potential energy and the Newmark time integration scheme are employed in this study, as shown in Section 4. A semi-analytic geometry iteration (SAGI) algorithm is proposed to find the closest points between a point and an ellipsoid based on the geometric convexity of ellipsoid and its parametric equation. Then, the SAGI algorithm is applied to detect the contact points and contact directions between ellipsoidal particles. The SAGI algorithm is further extended for identifying the geometric contact information between ellipsoid and a cylindrical or conical container without using meshes. Our numerical tests show that the convergence rate of the proposed SAGI algorithm is increased by more than an order of magnitude compared with the traditional optimization algorithm. One can refer to Section 5 for more detailed comparisons.

On the other hand, the conventional open-close iteration for solving contact force in the original DDA heavily relies on values of contact springs that are usually specified by experience. To bypass this limitation, some sophisticated strategies, such as the complementary theory [26], the variational inequality formation [27,28], and the second-order cone programming [29], have been proposed along with the

* Corresponding authors.
E-mail addresses: huangduruo@tsinghua.edu.cn (D. Huang), gwang@ust.hk (G. Wang).

https://doi.org/10.1016/j.compgeo.2020.103459
Received 29 October 2019; Received in revised form 13 December 2019; Accepted 19 January 2020

development of DDA to calculate the contact force as accurately as possible. However, convergence rate of these solution algorithms, i.e. the path Newton method in [26] or the compatibility iteration in [27,28], is still an obstacle to their practical application. Meanwhile, the second-order cone programming in [29] is difficult to implement.

In this study, the linearized cone complementarity formulation is deduced in detail for reformulated the frictional contact phenomenon without using contact springs. The so-called fixed point iteration algorithm [30] is resorted to solve the linearized cone complementarity problem in order to obtain the correct contact forces. To save the memory space and improve the computational speed, in the actual implementation, two types of concise and compact data structures are employed, which are constructed and organized in the terms of the contact-pairs and the bodies, respectively. After verifying the conservations of energy, linear and angular momentums, the accuracy, computational efficiency, and application prospects of the presented methodologies are demonstrated by some interesting and challenging examples. Consequently, an accurate, efficient, implementable DDA method is established, which is expected to provide a new numerical simulation platform for modeling the mechanical response of particulate media [31–33].

2. Geometry of ellipsoid

2.1. Function of a standard ellipsoid

By setting the centroid of an ellipsoid to be at the origin of the spherical coordinate system, we have the following parametric function of a *standard ellipsoid* (see Fig. 1(a) and (b))

$$\begin{cases} u = r_u \cos a \cos e \\ v = r_v \sin a \cos e \\ w = r_w \sin e \end{cases}, \quad \begin{matrix} r_u, r_v, r_w > 0 \\ a \in [0, 2\pi] \\ e \in [-\pi/2, \pi/2] \end{matrix}, \tag{1}$$

where r_u, r_v, and r_w are the three semi-axes of the ellipsoid. Without loss of generality, we define $r_u \geq r_v \geq r_w$ in this study. Moreover, the two angle parameters a and e are called as the azimuth angle and elevation angle, respectively, see Fig. 1(a). A pair of angle parameters (a, e) determine uniquely a point on the standard ellipsoid. Further, Eq. (1) can be rewritten in a matrix format as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} r_u & 0 & 0 \\ 0 & r_v & 0 \\ 0 & 0 & r_w \end{bmatrix} \begin{bmatrix} \cos a \cos e \\ \sin a \cos e \\ \sin e \end{bmatrix} = \mathrm{Diag}(r_u, r_v, r_w)\boldsymbol{e}_a, \tag{2}$$

where $\mathrm{Diag}(r_u, r_v, r_w)$ denotes a diagonal matrix. Applying a rotation $\boldsymbol{R}$, scaling $\boldsymbol{S}$, and non-zero centroid $\boldsymbol{x}_c = (x_c, y_c, z_c)^T$ to Eq. (2) for the standard ellipsoid, the *transformed ellipsoid* with the centroid located at $C(x_c, y_c, z_c)^T$ (shown in Fig. 1(c)) can be expressed as

$$\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{e}_a + \boldsymbol{x}_c, \tag{3}$$

where $\boldsymbol{x} = (x, y, z)^T$ and

$$\boldsymbol{\Phi} = \boldsymbol{S}\cdot\boldsymbol{R}\cdot\mathrm{Diag}(r_u, r_v, r_w). \tag{4}$$

In this study, the 3×3 matrix Φ is called as the *shape matrix* of the ellipsoid, while the 3×1 vector $\boldsymbol{x}_c$ is referred to as the *position vector* of the same ellipsoid. The rotation matrix $\boldsymbol{R}$ is given by

$$\boldsymbol{R} = \begin{bmatrix} \cos r_y \cos r_z & \sin r_x \sin r_y \cos r_z - \cos r_x \sin r_z & \cos r_x \sin r_y \cos r_z + \sin r_x \sin r_z \\ \cos r_y \sin r_z & \sin r_x \sin r_y \sin r_z + \cos r_x \cos r_z & \cos r_x \sin r_y \sin r_z - \sin r_x \cos r_z \\ -\sin r_y & \sin r_x \cos r_y & \cos r_x \cos r_y \end{bmatrix}, \tag{5}$$

where r_x, r_y, and r_z are the three Euler angles around x-, y-, and z-axis, respectively. In addition, the scaling matrix $\boldsymbol{S}$ is defined as

$$\boldsymbol{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}, \tag{6}$$

where s_x, s_y, and s_z are the scaling factors along x-, y-, and z-axis, respectively. It is worth mention that scaling operation is needed to enlarge or reduce the particle size during preparation process of a packing, and we set $s_x = s_y = s_z$ in this paper.

In our dynamic simulation, an ellipsoidal particle is treated as a rigid body. Therefore, only rotation and translation will be involved. From the time steps k to $k + 1$, the new transformed ellipsoid can be expressed by

$$\boldsymbol{x}^{k+1} = \boldsymbol{\Phi}^{k+1}\boldsymbol{e}_a + \boldsymbol{x}_c^{k+1}, \tag{7}$$

where

$$\begin{aligned} \boldsymbol{\Phi}^{k+1} &= \boldsymbol{R}^k\boldsymbol{\Phi}^k, \\ \boldsymbol{x}_c^{k+1} &= \boldsymbol{R}^k\boldsymbol{x}_c^k + \boldsymbol{t}^k, \end{aligned} \tag{8}$$

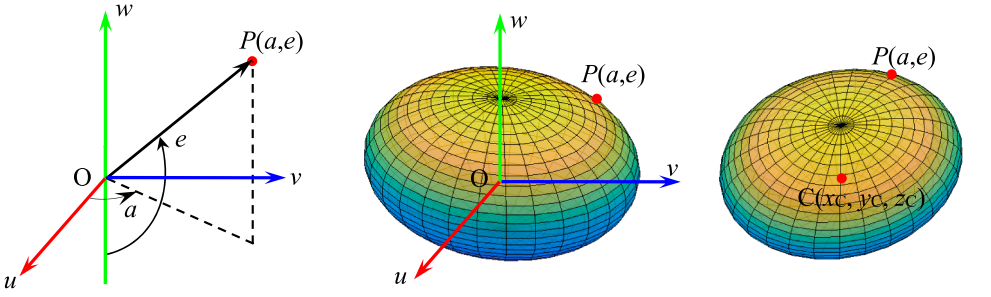
where $\boldsymbol{\Phi}^{k+1}$ and $\boldsymbol{x}_c^{k+1}$ are the new shape matrix and new centroid, respectively, and $\boldsymbol{t}^k = (\Delta x^k, \Delta y^k, \Delta z^k)^T$ represents the translation component. As we can see, the shape matrix is only affected by the rotation transformation. Similar to the standard ellipsoid, a pair of angle parameters (a, e) also specify uniquely a point on the transformed ellipsoid. For a sphere ($r_u = r_v = r_w$), at any time step, the shape matrix will be reduced to the 3×3 identity matrix. Equation (7) is the mathematical foundation of the proposed semi-analytic geometry iteration algorithm.

3. Semi-analytic geometry iteration algorithm

In this section, we will detail the semi-analytic geometry iteration (SAGI) algorithm for finding the closest points between two ellipsoidal particles. For this purpose, we discuss firstly how to determine the closest points between a point and an ellipsoidal particle.

3.1. The closest points between a point and an ellipsoidal particle

The closest point between an ellipsoidal surface and a given point P is the orthogonal projection $I^{Closest}(a_I, e_I)$ of point P onto the ellipsoidal surface, see Fig. 2. This means that the angle θ between vector $\overrightarrow{I^{Closest}P}$ and the unit outward normal vector n at point $I^{Closest}$ will approach zero. Here, the unit outward normal vector n is calculated analytically



(a) Spherical coordinate system (b) a standard ellipsoid (c) a transformed ellipsoid

Fig. 1. Spherical coordinate system and ellipsoid.

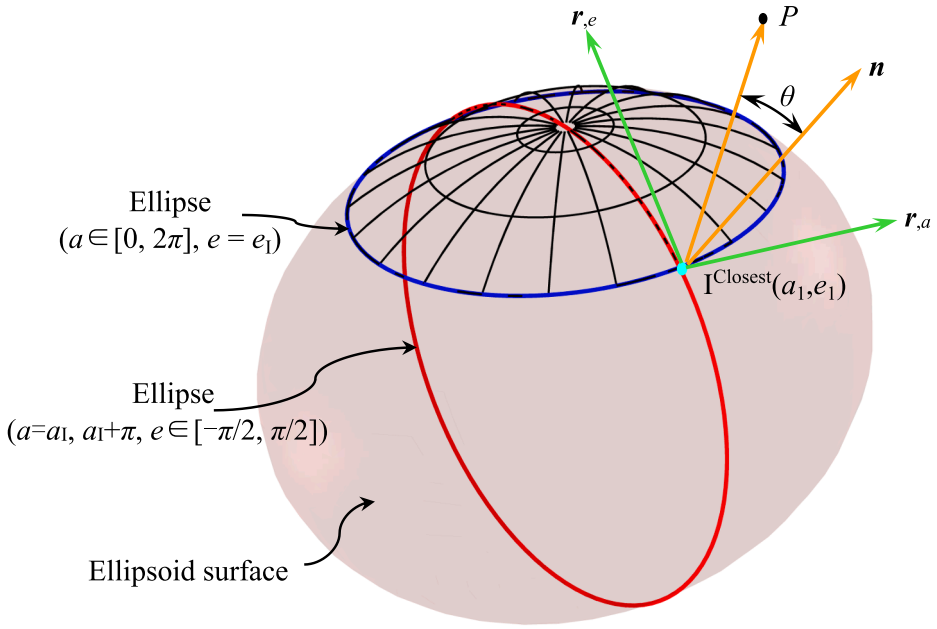


Fig. 2. Closest point between point P and an ellipsoidal particle.

$$\begin{aligned}\boldsymbol{r}_{,a} &= \left\{\frac{\partial x}{\partial a}, \frac{\partial y}{\partial a}, \frac{\partial z}{\partial a}\right\},\\ \boldsymbol{r}_{,e} &= \left\{\frac{\partial x}{\partial e}, \frac{\partial y}{\partial e}, \frac{\partial z}{\partial e}\right\},\\ \boldsymbol{n} &= \left.\frac{\boldsymbol{r}_{,a}\times\boldsymbol{r}_{,e}}{|\boldsymbol{r}_{,a}\times\boldsymbol{r}_{,e}|}\right|_{(a_1,e_1)}.\end{aligned} \tag{9}$$

For the convenience of calculating the unit outward normal vector $\boldsymbol{n}$, the parametric equation of transformed ellipsoid can be further explicitly expanded as

$$\begin{cases} x = \phi_{11} r_u \cos a \cos e + \phi_{12} r_v \sin a \cos e + \phi_{13} r_w \sin e + x_c \\ y = \phi_{21} r_u \cos a \cos e + \phi_{22} r_v \sin a \cos e + \phi_{23} r_w \sin e + y_c \\ z = \phi_{31} r_u \cos a \cos e + \phi_{32} r_v \sin a \cos e + \phi_{33} r_w \sin e + z_c \end{cases} \tag{10}$$

where ϕ_{ij} $(i, j = 1, 2, 3)$ is the element of the shape matrix $\boldsymbol{\Phi}$ at the i-th row and j-th column, and (x_c, y_c, z_c) is the centroid of the ellipsoid.

From Fig. 2 we can conclude that there are two basic kinds of ellipses on an ellipsoid: The first type is determined by "a fixed azimuth angle a and the elevation angle $e \in [-\pi/2, \pi/2]$", which means that the azimuth angle is a constant, while the elevation angle traverses all of its possible values. This type ellipse is called as "ellipse-a" in this study. The second type is described by "a fixed elevation angle e and the azimuth angle $a \in [0, 2\pi]$", which implies that the elevation angle is a constant, while the azimuth angle can be any of its possible values. This kind of ellipse is referred to as "ellipse-e" in this paper. Inspired by this, we propose the following semi-analytic geometry iteration algorithm for determining the closest point between a point and an ellipsoidal surface.

The semi-analytic geometry iteration algorithm for finding the closest point between a point and an ellipsoid surface is stated as follows:

(1) Set the elevation angle $e = e_1$.
(2) Generate the *second* type of ellipse, e.g. ellipse-e_1, by setting $a \in [0, 2\pi]$ and $e = e_1$.
(3) Taking the azimuth angle a as a unknown parameter to determine the closest point I_1 between point P and ellipse-e_1, which can be solved analytically and will be discussed in more details later. Assume the closest point I_1 has the angle parameters (a_1, e_1).
(4) Construct the *first* kind of ellipse, e.g. ellipse-a_2, by using $a = a_1$, $a_1 + \pi$ and $e \in [-\pi/2, \pi/2]$.
(5) Taking the elevation angle e as an undetermined parameter to estimate the closest point I_2 between point P and ellipse-a_2, assume the closest point I_2 is corresponding to the angle parameters (a_1, e_2).
(6) Let $e_1 = e_2$, repeat steps (2) to (5) until the angle θ (refer to Fig. 2) between vector $\overrightarrow{\mathrm{I}_k\mathrm{P}}$ and the unit outward normal vector $\boldsymbol{n}$ at point I_k is less than a tolerance (e.g. 1×10^{-4}, which is enough for an angular value), point I_k is the desired closest point. Usually, after 3–5 iterations, the closest points can be found. One can see Fig. 3 to understand intuitively this iteration procedure. In Fig. 3, note that the elevation angles e_2 and e_3 are almost equal to each other.

It should be pointed out that the closest point between a point and an ellipse-a or ellipse-e, which is involved in steps (3) and (5) in the above iteration process, can be determined analytically. Just because of this, we term iteration algorithm as "semi-analytic". Next, we will focus on the analytical algorithm in more details.

One can use the following parametric function of t to define the two types of ellipse afore-mentioned

$$\boldsymbol{x} = \boldsymbol{C}_{3\times 2}\begin{bmatrix}\cos t\\ \sin t\end{bmatrix} + \boldsymbol{x}_t, \quad t \in [0, 2\pi], \tag{11}$$

For the first type of ellipse, ellipse-a with a given azimuth angle, the 3×2 matrix $\boldsymbol{C}$ is given by

$$\boldsymbol{C} = \begin{bmatrix} \phi_{11} r_u \cos a + \phi_{12} r_v \sin a & \phi_{13} r_w \\ \phi_{21} r_u \cos a + \phi_{22} r_v \sin a & \phi_{23} r_w \\ \phi_{31} r_u \cos a + \phi_{32} r_v \sin a & \phi_{33} r_w \end{bmatrix}, \tag{12}$$

where $\phi_{ij}(i, j = 1, 2, 3)$ has the same meanings as in Eq. (10), and the vector $\boldsymbol{x}_t = \boldsymbol{x}_c = (x_c, y_c, z_c)^{\mathrm{T}}$. While for the second type of ellipse, namely ellipse-e with a fixed elevation angle, the 3×2 matrix $\boldsymbol{C}$ is expressed as

$$\boldsymbol{C} = \begin{bmatrix} \phi_{11} r_u \cos e & \phi_{12} r_v \cos e \\ \phi_{21} r_u \cos e & \phi_{22} r_v \cos e \\ \phi_{31} r_u \cos e & \phi_{32} r_v \cos e \end{bmatrix}, \tag{13}$$

and the vector $\boldsymbol{x}_t$ becomes

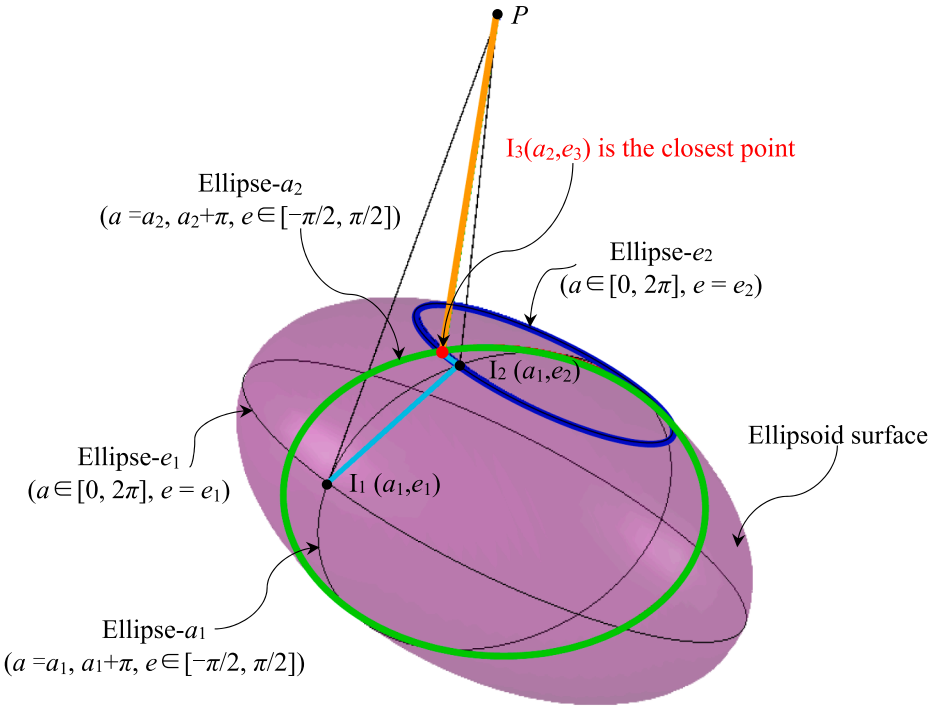


Fig. 3. Iteration process of finding the closest point between a point and an ellipsoid.

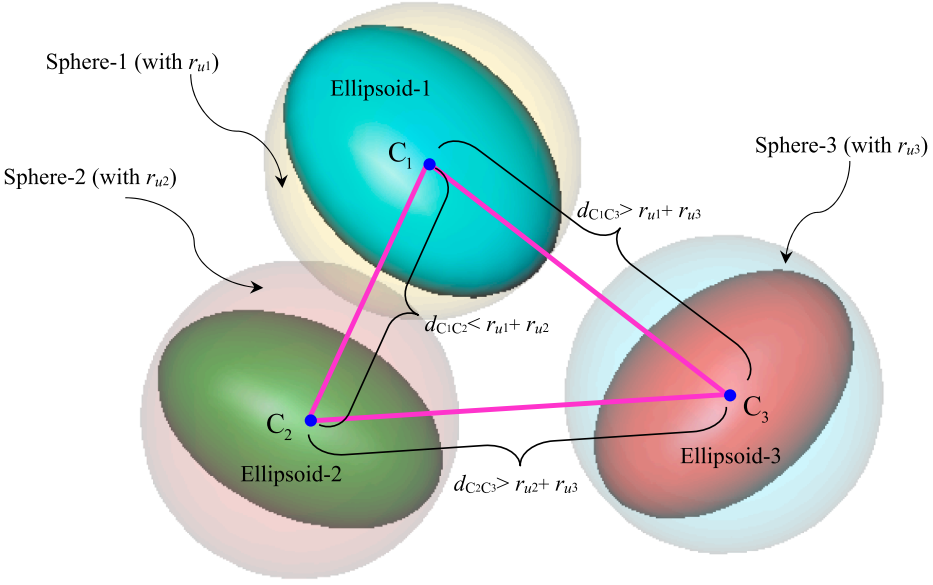


Fig. 4. External sphere estimated for two ellipsoids.

$$\boldsymbol{x}_t = \boldsymbol{x}_c + r_w \sin e\ [\phi_{13},\ \phi_{23},\ \phi_{33}]^{\mathrm{T}}. \tag{14}$$

Based on Eq. (11), on an ellipse, the closest point I_k to a specific point $P(x_P, y_P, z_P)$ should satisfy the following stationary condition

$$\frac{d\ |\vec{I_k P}|^2}{dt} = \frac{d}{dt}[(x(t) - x_P)^2 + (y(t) - y_P)^2 + (z(t) - z_P)^2] = 0. \tag{15}$$

Note that in the above equation, $\cos t = \frac{1-\tan^2(t/2)}{1+\tan^2(t/2)}$ and $\sin t = \frac{2\tan(t/2)}{1+\tan^2(t/2)}$, so it can be transformed to a quadratic equation of $\tan(t/2)$. There is already existing efficient method to solve this type of equation analytically. Generally speaking, we can obtain 4 complex number solutions for $\tan(t/2)$. Further, by picking out these real solutions to calculate the corresponding parameters t and candidate points. Finally, by comparing the relevant distances we can find out the closest point. Following a similar procedure, the closet points between a point and cylindrical/conical surface can be established (cf. Appendix A1-A3).

3.2. The closest points between two ellipsoids

Now, we will extend the above semi-analytic geometry iteration algorithm to determine the closest points between two ellipsoidal surfaces/particles.

Fig. 4 shows three transformed ellipsoid surfaces ellipsoid-1, ellipsoid-2, and ellipsoid-3. Their centroids are C_1, C_2, and C_3, respectively. For any ellipsoid we can take its r_u ($r_u \geq r_v \geq r_w$ in this study) as radius

to construct an external sphere surrounding it. In Fig. 4, the corresponding external spheres are sphere-1(with r_{u1}), sphere-2(with r_{u2}), and sphere-3(with r_{u3}), respectively. These spheres are employed to check preliminarily whether the relevant ellipsoids are possibly in contact with each other or not. At the initial stage of the contact detection, for the sake of narration, we define two possible cases: *external-sphere-intersection* and *external-sphere-nonintersection*. The so-called external-sphere-intersection means that the distance between the two centroids of any two ellipsoids is less than or equal to the sum of their r_u. For example in Fig. 4, ellipsoid-1 and ellipsoid-2 ($d_{C1C2} < r_{u1} + r_{u2}$) can be classified as this case. Whereas, the alleged external-sphere-nonintersection implies that, for instance in Fig. 4, $d_{C1C3} > r_{u1} + r_{u3}$ between ellipsoid-1 and ellipsoid-3, or $d_{C2C3} > r_{u2} + r_{u3}$ between ellipsoid-2 and ellipsoid-3. For the case of external-sphere-nonintersection, the corresponding ellipsoids do not contact each other. In contrast, the case of external-sphere-intersection should be further disposed at the narrow phase of the contact detection. This procedure is called as the *external sphere estimating* in this study.

Under the premise of external-sphere-intersection, if two corresponding ellipsoids might be in contact, there are two possible cases, namely *two non-overlap ellipsoids* and *two overlap ellipsoids* (see Fig. 5) that should be distinguished firstly.

Two ellipsoids, ellipsoid-a and ellipsoid-b with centroids C_a and C_b respectively, are sketched in Fig. 6 (only 2D view is given for clarity). To determine whether the two ellipsoids overlap each other or not, we propose the following semi-analytic geometry iteration, which is called as the *overlap checking* procedure:

(1) Determine the closest point A between point C_a and ellipsoid-b using algorithm in Section 3.1.
(2) Determine the closest point B between point C_b and ellipsoid-a.
(3) Choose initial iteration point I_0 according to the following consideration: Because point A is just the orthogonal projection of point C_a onto ellipsoid-b, vector $\overrightarrow{AC}_a$ is collinear with the unit outward vector n_A of ellipsoid-b at point A. Due to same reason, vector $\overrightarrow{BC}_b$ is collinear with the unit outward vector n_B of ellipsoid-a at point B. Therefore, $\angle ABC_b$ denotes the angle between vectors $\overrightarrow{BA}$ and n_B. Similarly, $\angle BAC_a$ is the angle between vectors $\overrightarrow{AB}$ and n_A. Based on the fact that if points A and B are the closest points between the ellipsoids, both $\angle ABC_b$ and $\angle BAC_a$ approach to zero. Thus, if $\angle ABC_b < \angle BAC_a$, point B is closer to the target point in some degree. In this case, point B can be chosen as the initial iteration point. Otherwise, point A is used as the initial iteration point. In Fig. 6, point B is picked as the initial iteration point I_0 for illustration.
(4) Find the closest point I_1 between point I_0 and the other ellipsoid. One can refer to Fig. 6 to understand this step intuitively.
(5) Compute the closest point I_2 between point I_1 and the other ellipsoid (see Fig. 7).
(6) Repeat steps (4) and (5) until $\angle ABC_b$ and $\angle BAC_a$ are both close to the preset tolerance (e.g. 1×10^{-5}). For example, points I_3 and I_4 in Fig. 6 are the desired closest points.
(7) Calculate the distance d_{min} between the closest points. If d_{min} is greater than a tolerance (e. g. 1×10^{-6}), we get two non-overlapped ellipsoids. Otherwise, we have two overlapped ellipsoids.

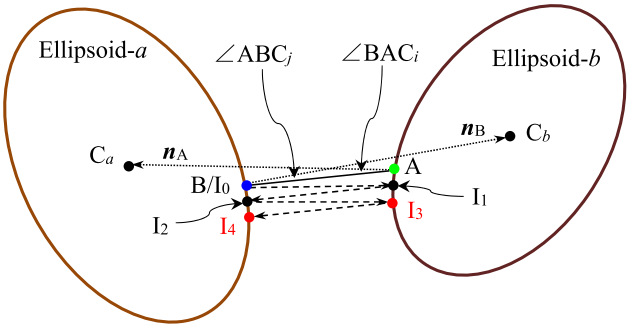
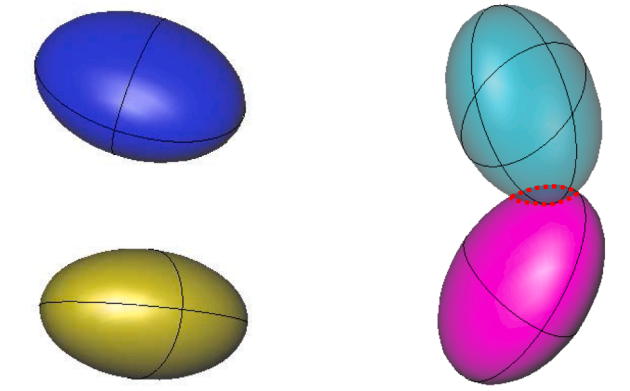


Fig. 6. Closest points between two non-overlapped ellipsoids (only 2D view for clarity).

For two non-overlapped ellipsoids, assume that points A and B are the closest points belonging to ellipsoid-a and ellipsoid-b ($a < b$), respectively, as shown in Fig. 7. We can define the contact points and contact directions by resorting to the following *contact determining* procedure:

(1) Take the mid-point of line segment AB as the contact point C.
(2) Choose one of closest points from the ellipsoid with a relative bigger index, for example point B of ellipsoid-b in Fig. 7, and calculate the unit outward normal $\boldsymbol{n}$ vector of this ellipsoid at point B, which is defined as the contact normal vector, see Fig. 7. Note that the contact normal vector points to the interior of the ellipsoid with a relative smaller index, namely ellipsoid-a here.
(3) Construct the tangent plane by using the unit normal $\boldsymbol{n}$ and passing through the contact point C. Generate another point D on the tangent plane to form vector $\overrightarrow{CD}$. Establish the first contact tangent unit vector $\boldsymbol{s}$ by $\boldsymbol{s} = \overrightarrow{CD}/|\overrightarrow{CD}|$ and the second contact tangent unit vector $\boldsymbol{t}$ by $\boldsymbol{t} = \boldsymbol{n} \times \boldsymbol{s}$, respectively.

Aiming at two non-overlapped ellipsoids, after executing the contact determining procedure we can obtain the contact points and contact directions, which is the base of solving interaction involved in a discrete



(a) Two non-overlapped ellipsoids (b) Two overlapped ellipsoids

Fig. 5. Two non-overlapped and overlapped ellipsoids.

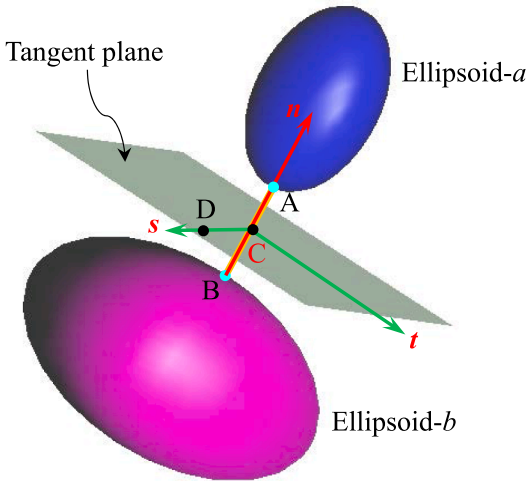


Fig. 7. Contact points and contact direcitons between two ellipsoids.

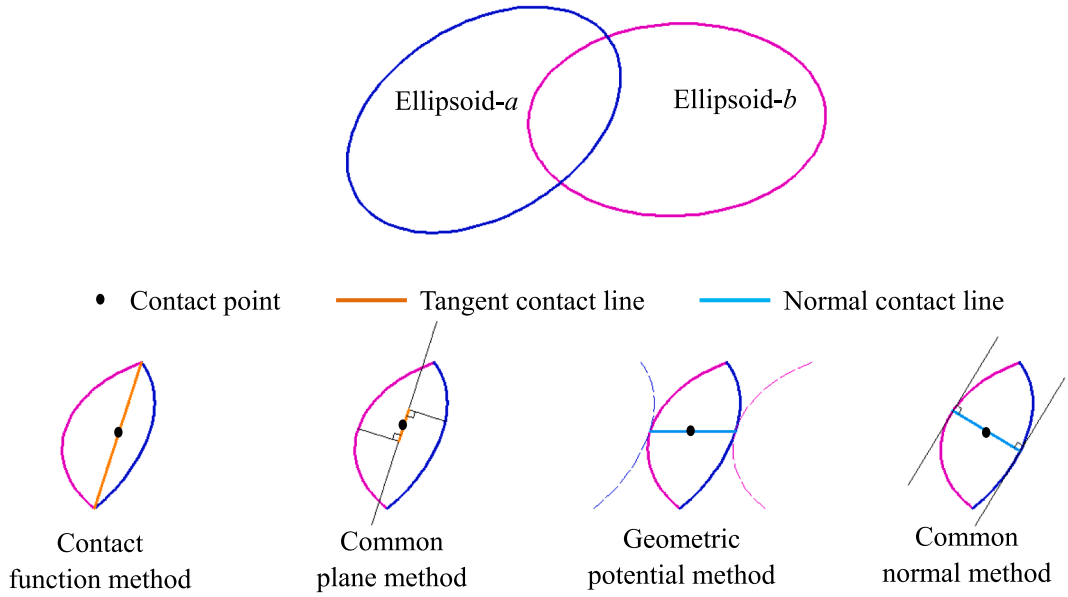


Fig. 8. "Overlap-based" schemes for two overlapped ellipsoid (only 2D view for clarity).

medium system.

A so-callled "*overlap-based*" scheme is used for disposing two overlapped ellipsoids or bodies. The method is summarized as follows: (1) Let two ellipsoids overlap each other; (2) based on the overlapped region to find the contact points and the contact directions. This tactic has been used widely in some approaches, such as the contact function method [34], the common plane method [35], the geometric potential method [36], and the common normal algorithm [37]. In general, the contact points and the contact directions obtained from these above-mentioned approaches are different even though they are all based on the overlap-based schemes (see Fig. 8). Specifically speaking, in the contact function method, it chooses the intersection line of two ellipsoids as the tangent contact line, whose mid-point plays the part of the contact point. As for the common plane method, the normal vector of the common plane is taken as the contact normal vector. Correspondingly, the tangent contact line is on the common plane. Whereas, in the geometric potential method, we determine firstly the minimum geometric potential line that serves as the contact normal line, whose mid-point is the contact point. Regarding to the common normal method, two specific points on each of two ellipsoid surfaces are found in a condition such that the normal directions at these points are parallel to the line that passes through the two points. The mid-point of the common line is defined as the contact point, as exhibited in Fig. 8. Consider the two facts that the overlap between two rigid bodies is not allowed from a realistic point of view and that the displacement of any individual ellipsoid is a small value within a time increment from a computational point of view, we utilize another strategy, named as the "*separation-based*" scheme, as demonstrated in Fig. 9, to determine the contact between two overlapped ellipsoids.

For two overlapped ellipsoids, we perform the following "separation-based" procedure to "drag" the two ellipsoids to their respective temporary locations, i.e. ellipsoid-a* and ellipsoid-b*.

(1) Change the positions of the two ellipsoids by $-k\Delta\boldsymbol{d}_a^{\text{last}}$ and $-k\Delta\boldsymbol{d}_b^{\text{last}}$, where $\Delta\boldsymbol{d}_a^{\text{last}}$ and $\Delta\boldsymbol{d}_b^{\text{last}}$ are the increments of the basic known vector of the two ellipsoids within the last time step. The coefficient k (recommended value $k \in [0.3, 0.7]$) is a positive real number. In current study, we set $k = 0.4$. The negative sign $(-)$ indicates that the "separation-based" scheme is a reverse "operation".

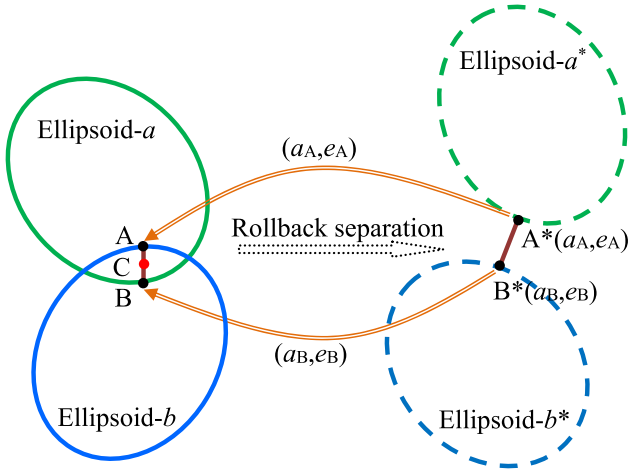


Fig. 9. "Separation-based" scheme for two overlap ellipsoids.

(2) Invoke the overlap checking procedure (described previously) to determine whether there is at least one intersection point between the two ellipsoids.
(3) Repeat the steps (1) and (2) until the two ellipsoids become separated. Usually, we can successfully separate the two bodies within two or three cycles.
(4) Once the two ellipsoids are separated, the overlap checking procedure gives the temporary closest points A*(a_A, e_A) and B*(a_B, e_B) between ellipsoid-a* and ellipsoid-b*, see Fig. 9.
(5) Determine the line AB on ellipsoid-a and ellipsoid-b by using two sets of angle parameters (a_A, e_A) and (a_B, e_B), respectively. Then, the contact point C and the contact directions $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{t}$ can be obtained by the contact determining procedure.

The separation-based scheme can avoid some difficulties that might exist in the overlap-based scheme, such as the increased computation cost, degraded stability, decreased accuracy, and even calculation failure especially for handling two highly complex particles. In addition, the separation-based scheme is more in line with the physical reality.

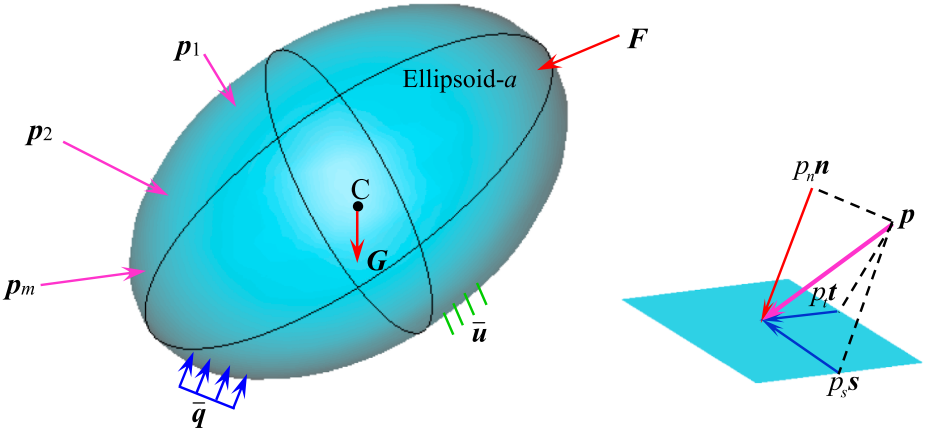


Fig. 10. Constraints and loading on an ellipsoidal particle.

4. Dynamics of ellipsoidal particles

Assume that an ellipsoidal particle is a rigid body; next, we will describe the movement of ellipsoidal particles in the framework of DDA.

4.1. Equation of motion of an ellipsoidal particle

In Fig. 10 point C(x_c, y_c, z_c) is the centroid of the ellipsoid-a, and $\boldsymbol{F}$, $\boldsymbol{G}$, and $\bar{\boldsymbol{q}}$ are the point loading, the gravity, and the distributed loading, respectively. In addition, $\bar{\boldsymbol{u}}$ is the specified displacement constraints. Assume that there are m contact-pairs on this ellipsoid, the corresponding contact forces are $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, …, and $\boldsymbol{p}_m$. Any contact force $\boldsymbol{p}$ can be expressed as $\boldsymbol{p} = [p_n\boldsymbol{n}, p_s\boldsymbol{s}, p_t\boldsymbol{t}]^{\mathrm{T}}$. Note that the vectors $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{t}$ are not necessarily parallel to the global Cartesian coordinate system. Following the origianl DDA and the Newmark time integration (algorithm parameters $\gamma = 1.0$ and $\beta = 0.5$), we have the equation of motion for ellipsoid-a considering these contact forces as

$$\frac{2}{\Delta t^2}\boldsymbol{m}_a\Delta\boldsymbol{d}_a - \bar{\boldsymbol{h}}_a\bar{\boldsymbol{p}}_a = \boldsymbol{f}_a, \tag{16}$$

where $\boldsymbol{m}_a$ and $\boldsymbol{f}_a$ are the 6×6 equivalent mass matrix and the 6×1 equivalent force vector, respectively, and Δt is the time step. The increment of displacement or the based unknown vector of ellipsoid-a is defined as

$$\Delta\boldsymbol{d}_a = [\Delta u_c,\ \Delta v_c,\ \Delta w_c,\ \Delta r_x,\ \Delta r_y,\ \Delta r_z]^{\mathrm{T}} \tag{17}$$

where Δu_c, Δv_c, and Δw_c are three translational increments of the centroid of the ellipsoidal particle, Δr_x, Δr_y, and Δr_z are three Euler rotation angle increments of the same particle. A non-negative contact force vector $\bar{\boldsymbol{p}}_a$ reads

$$\bar{\boldsymbol{p}}_a = [p_{1n}, p_{1s}, p_{1t}, p_{2n}, p_{2s}, p_{2t}, \cdots, p_{m\,n}, p_{m\,s}, p_{m\,t}]^{\mathrm{T}}. \tag{18}$$

Moreover, the transform matrix $\bar{\boldsymbol{h}}_a$, whose functions is to transfer variables from the local contact coordinate system to the global Cartesian coordinate system, is obtained from

$$\bar{\boldsymbol{h}}_a = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_j, \cdots, \boldsymbol{h}_m], \tag{19}$$

where m is assumed to be the total number of these contact forces on the ellipsoid-a. And each $\boldsymbol{h}_j$ in $\bar{\boldsymbol{h}}_a$ is the 6×3 matrix associated with the j-th contact-pair on the ellipsoid-a, reading

$$\boldsymbol{h}_j(x_j, y_j, z_j) = sign(\cdot)\boldsymbol{T}_a^{\mathrm{T}}(x_j, y_j, z_j)[\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}]_j, \tag{20}$$

where (x_j, y_j, z_j) are the coordinates of the j-th contact point at the beginning of the current time step. Note that the contact forces always come in pairs and they are equal and opposite in direction. Moreover, the vector $\bar{\boldsymbol{p}}_a$ is non-negative. We introduce a sign function $sign(\cdot)$ to represent that one contact force is positive and the other one is negative. Complying with the definitions of the unit vectors $\boldsymbol{n}$ (refer to Fig. 7) of a certain contact-pair between ellipsoid-a and ellipsoid-b, if $a < b$, $sign(\cdot)$ associated with ellipsoid-a is the sign $(+)$, while $sign(\cdot)$ related to ellipsoid-b is the sign $(-)$. Meanwhile, the interpolation matrix $\boldsymbol{T}_a$ is given by

$$\boldsymbol{T}_a(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & 0 & z - z_c{}^a & y_c{}^a - y \\ 0 & 1 & 0 & z_c{}^a - z & 0 & x - x_c{}^a \\ 0 & 0 & 1 & y - y_c{}^a & x_c{}^a - x & 0 \end{bmatrix}, \tag{21}$$

where $(x_c{}^a, y_c{}^a, z_c{}^a)$is the centroid of ellipsoid-a. Additionally, the 3×3 matrix $[\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}]_j$ is defined as

$$[\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}]_j = \begin{bmatrix} \cos\alpha_n & \cos\alpha_s & \cos\alpha_t \\ \cos\beta_n & \cos\beta_s & \cos\beta_t \\ \cos\gamma_n & \cos\gamma_s & \cos\gamma_t \end{bmatrix}_j, \tag{22}$$

where ($\cos\alpha_n$, $\cos\beta_n$, $\cos\gamma_n$), ($\cos\alpha_s$, $\cos\beta_s$, $\cos\gamma_s$), and ($\cos\alpha_t$, $\cos\beta_t$, $\cos\gamma_t$) are the direction cosines of the vectors $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{t}$, respectively.

In order to generate the equivalent mass matrix and equivalent force vector in Eq. (16), we have to calculate ten formulas: $F = \iiint dV$, $F_x = \iiint x dV$, $F_y = \iiint y dV$, $F_z = \iiint z dV$, $F_{xx} = \iiint x^2 dV$, $F_{yy} = \iiint y^2 dV$, $F_{zz} = \iiint z^2 dV$, $F_{yz} = \iiint yz dV$, $F_{zx} = \iiint zx dV$, and $F_{xy} = \iiint xy dV$. From time steps k to $k + 1$, the new coordinates $\boldsymbol{x}^{k+1}(x^{k+1}, y^{k+1}, z^{k+1})$ of any point $\boldsymbol{x}^k(x^k, y^k, z^k)$ on an ellipsoid particle can be obtained by

$$\boldsymbol{x}^{k+1} = \boldsymbol{R}\,\boldsymbol{x}^k + (\boldsymbol{I} - \boldsymbol{R})\boldsymbol{x}_c^k + \Delta\boldsymbol{u}_c \tag{23}$$

where $\boldsymbol{I}$ is identity matrix, $\boldsymbol{R}$ is a rotation matrix defined in Eq. (5) and is calculated by replacing r_x, r_y, and r_z with Δr_x, Δr_y, and Δr_z, respectively, and $\Delta\boldsymbol{u}_c = [\Delta u_c, \Delta v_c, \Delta w_c]^{\mathrm{T}}$. Note that $F = \iiint dV$ is the volume of ellipsoid particle, which is constant for rigid body. Moreover, the three integrals F_x, F_y, and F_z can be computed from

$$\begin{aligned}&[F_x^{k+1},\ F_y^{k+1},\ F_z^{k+1}]^{\mathrm{T}} \\ &= \int \boldsymbol{x}^{k+1}dV = \int \boldsymbol{R}x^k dV + \int (\boldsymbol{I} - \boldsymbol{R})\boldsymbol{x}_c^k dV + \int \Delta\boldsymbol{u} dV.\end{aligned} \tag{24}$$

By simplifying Eq. (24), one obtains

$$[F_x^{k+1},\ F_y^{k+1},\ F_z^{k+1}]^{\mathrm{T}} = \boldsymbol{R}[F_x^k,\ F_y^k,\ F_z^k]^{\mathrm{T}} + F[(\boldsymbol{I} - \boldsymbol{R})\boldsymbol{x}_c^k + \Delta\boldsymbol{u}]. \tag{25}$$

As for the last six integrals, based on Eq. (23) we construct an interim matrix

$$\begin{bmatrix} F_{xx}^{k+1} & F_{xy}^{k+1} & F_{xz}^{k+1} \\ F_{yx}^{k+1} & F_{yy}^{k+1} & F_{yz}^{k+1} \\ F_{zx}^{k+1} & F_{zy}^{k+1} & F_{zz}^{k+1} \end{bmatrix} = \int \boldsymbol{x}^{k+1}(\boldsymbol{x}^{k+1})^{\mathrm{T}} dV$$
$$= \int \boldsymbol{R}\boldsymbol{x}^{k}(\boldsymbol{R}\boldsymbol{x}^{k})^{\mathrm{T}} dV + \int (\boldsymbol{I}-\boldsymbol{R})\boldsymbol{x}_{\mathrm{c}}^{k}[(\boldsymbol{I}-\boldsymbol{R})\boldsymbol{x}_{\mathrm{c}}^{k}]^{\mathrm{T}} dV + \int \Delta\boldsymbol{u}(\Delta\boldsymbol{u})^{\mathrm{T}} dV. \tag{26}$$

After some mathematical derivations, we have

$$\begin{bmatrix} F_{xx}^{k+1} & F_{xy}^{k+1} & F_{xz}^{k+1} \\ F_{yx}^{k+1} & F_{yy}^{k+1} & F_{yz}^{k+1} \\ F_{zx}^{k+1} & F_{zy}^{k+1} & F_{zz}^{k+1} \end{bmatrix} = \boldsymbol{R}\begin{bmatrix} F_{xx}^{k} & F_{xy}^{k} & F_{xz}^{k} \\ F_{yx}^{k} & F_{yy}^{k} & F_{yz}^{k} \\ F_{zx}^{k} & F_{zy}^{k} & F_{zz}^{k} \end{bmatrix}\boldsymbol{R}^{\mathrm{T}} + F\{(\boldsymbol{I}-\boldsymbol{R})\boldsymbol{x}_{\mathrm{c}}^{k}[(\boldsymbol{I}-\boldsymbol{R})\boldsymbol{x}_{\mathrm{c}}^{k}]^{\mathrm{T}} + \Delta\boldsymbol{u}(\Delta\boldsymbol{u})^{\mathrm{T}}\}. \tag{27}$$

The elements in the upper triangular matrix of the interim matrix are the desired last six integrals. Eqs. (25) and (27) imply that the aforementioned ten integrations need to be calculated only at the initial time step (i.e. $t = 0$). During the simulation, the subsequent volume integrals can be obtained recursively according to Eqs. (25) and (27).

From Eq. (3), we know that for the intial time step ($t = 0$), an arbitrary transformed ellipsoid can be regarded as transfromed from its corresponding standard ellipsoid with the same three semi-axes, e.g. r_u, r_v, and r_w. As for the standard ellipsoid, the ten integrals can be calulated analytically, reading

$$F = \frac{4\pi}{3} r_u r_v r_w, \quad F_x = F_y = F_z = 0,$$
$$F_{xx} = \frac{4\pi}{15} r_u^3 r_v r_w, \quad F_{yy} = \frac{4\pi}{15} r_u r_v^3 r_w, \quad F_{zz} = \frac{4\pi}{15} r_u r_v r_w^3. \tag{28}$$

For the transformed ellipsoid corresponding to the initial time step ($t = 0$), we have

$$\begin{aligned} F^0 &\equiv F, \quad F_x^0 = x_{\mathrm{c}}^0 F, \quad F_y^0 = y_{\mathrm{c}}^0 F, \quad F_z^0 = z_{\mathrm{c}}^0 F \\ F_{yz}^0 &= \phi_{21}\phi_{31}F_{xx} + \phi_{22}\phi_{32}F_{yy} + \phi_{23}\phi_{33}F_{zz} + y_{\mathrm{c}}^0 z_{\mathrm{c}}^0 F \\ F_{zx}^0 &= \phi_{31}\phi_{11}F_{xx} + \phi_{32}\phi_{12}F_{yy} + \phi_{33}\phi_{13}F_{zz} + z_{\mathrm{c}}^0 x_{\mathrm{c}}^0 F \\ F_{xy}^0 &= \phi_{11}\phi_{21}F_{xx} + \phi_{12}\phi_{22}F_{yy} + \phi_{13}\phi_{23}F_{zz} + x_{\mathrm{c}}^0 y_{\mathrm{c}}^0 F \ , \\ F_{xx}^0 &= \phi_{11}^2 F_{xx} + \phi_{12}^2 F_{yy} + \phi_{13}^2 F_{zz} + (x_{\mathrm{c}}^0)^2 F \\ F_{yy}^0 &= \phi_{21}^2 F_{xx} + \phi_{22}^2 F_{yy} + \phi_{23}^2 F_{zz} + (y_{\mathrm{c}}^0)^2 F \\ F_{zz}^0 &= \phi_{31}^2 F_{xx} + \phi_{32}^2 F_{yy} + \phi_{33}^2 F_{zz} + (z_{\mathrm{c}}^0)^2 F \end{aligned} \tag{29}$$

where $\phi_{ij}(i, j = 1, 2, 3)$ is the element of the shape matrix ϕ^0 (at $t = 0$) in the i-th row and j-th column, and $\boldsymbol{x}^0 = (x_{\mathrm{c}}^0, y_{\mathrm{c}}^0, z_{\mathrm{c}}^0)^{\mathrm{T}}$ is the centroid of the transformed ellipsoid at $t = 0$. Based upon Eqs. (25), (27), and (29), we can compute these desired integrals expediently.

4.2. Treatment of contact based on cone complementary formulation

Consider any pair of contact, as shown in Fig. 11, point C is the contact point, vectors $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{t}$ are the three directions of the contact. The friction force $\boldsymbol{p}$ is given by $\boldsymbol{p} = \boldsymbol{p}_{\mathrm{N}} + \boldsymbol{p}_{\mathrm{T}} = p_n\boldsymbol{n} + p_s\boldsymbol{s} + p_t\boldsymbol{t}$, where $p_n \geq 0$, $\boldsymbol{p}_{\mathrm{T}} = p_s\boldsymbol{s} + p_t\boldsymbol{t}$. The Coulomb friction law states that

$$p_n \geqslant 0, \ \ g_n \geqslant 0, \ \ p_n g_n = 0, \tag{30}$$

$$\mu p_n + c \geqslant \sqrt{p_s^2 + p_t^2}\ , \ ||\boldsymbol{g}_{\mathrm{T}}||(\mu p_n + c - \sqrt{p_s^2 + p_t^2}) = 0, \tag{31}$$

$$<\boldsymbol{p}_{\mathrm{T}}, \boldsymbol{g}_{\mathrm{T}}> = -\,||\boldsymbol{p}_{\mathrm{T}}||\,||\boldsymbol{g}_{\mathrm{T}}||, \tag{32}$$

where g_n is the normal contact gap, $\boldsymbol{g}_{\mathrm{T}}$ is the total relative movement along the contact tangent direction, i.e. $\boldsymbol{g}_{\mathrm{T}} = g_s\boldsymbol{s} + g_t\boldsymbol{t}$. Moreover, μ (typically $\mu \in [0,1]$) and c are the friction factor and the inner cohesion, respectively. Eq. (30) denotes the complementary relationship in the normal direction. Eq. (31) refers to the constrain condition in the tangent direction. And Eq. (32) requires that the tangential contact force be opposite to the tangential relative movement, namely, the tangential reaction force is dissipative. The maximum dissipation principle can be

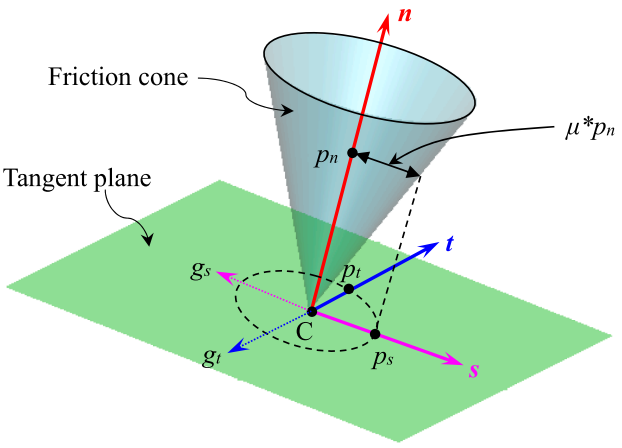


Fig. 11. A pair of contact.

used to express Eqs. (30), (31), and (32) equivalently, reading

$$(p_s, p_t) = \arg \min_{\mu p_n + c \geqslant \sqrt{p_s^2 + p_t^2}} (p_s\boldsymbol{s} + p_t\boldsymbol{t})^{\mathrm{T}}(g_s\boldsymbol{s} + g_t\boldsymbol{t}), \tag{33}$$

where the superscript "T" means "transpose". As for the optimization problem representing by Eq. (33), the Lagrangian multiplier method declares that there exists a non-negative constant λ that makes the following relationships hold

$$\lambda p_s = -g_s, \tag{34}$$

$$\lambda p_t = -g_t, \tag{35}$$

$$(0 \leqslant \lambda) \ \perp \ (\mu p_n + c - \sqrt{p_s^2 + p_t^2} \geqslant 0). \tag{36}$$

Eqs. (34) and (35) means that

$$\lambda \sqrt{p_s^2 + p_t^2} = \sqrt{g_s^2 + g_t^2}. \tag{37}$$

Note that $\sqrt{p_s^2 + p_t^2} \geqslant 0$, from Eq. (36) one can obtain

$$\lambda \sqrt{p_s^2 + p_t^2}\,(\mu p_n + c - \sqrt{p_s^2 + p_t^2}) = 0. \tag{38}$$

Considering Eq. (37) and Eq. (38) yields

$$(\mu p_n + c)\sqrt{g_s^2 + g_t^2} - \lambda(p_s^2 + p_t^2) = 0. \tag{39}$$

To consider cohesion effect, a non-negative number $\bar{\mu} = c/p_n$ can be introduced. Eqs. (34) and (35), Eq. (39) can be reformulated as

$$p_n\mu^*\sqrt{g_s^2 + g_t^2} - \lambda(p_s^2 + p_t^2) = p_n\mu^*\sqrt{g_s^2 + g_t^2} + p_s g_s + p_t g_t = 0, \tag{40}$$

where $\mu^* = \mu + \bar{\mu}$ and is called as the *equivalent friction factor* in this study. Note corresponding to the j-th contact-pair, consider two vectors

$$\boldsymbol{p}_j = (p_n, p_s, p_t)^{\mathrm{T}}$$
$$\tilde{\boldsymbol{g}}_j = (g_n + \mu^*\sqrt{g_s^2 + g_t^2}, g_s, g_t)^{\mathrm{T}} \tag{41}$$

where g_n is the normal contact gap, we immediately have

$$<\boldsymbol{p}_j, \tilde{\boldsymbol{g}}_j> = \boldsymbol{p}_j \cdot \tilde{\boldsymbol{g}}_j = p_n g_n + p_n\mu^*\sqrt{g_s^2 + g_t^2} + p_s g_s + p_t g_t = 0$$
$$\Leftrightarrow \boldsymbol{p}_j \perp \tilde{\boldsymbol{g}}_j. \tag{42}$$

For the friction cone $\mathscr{C}_j = \{\boldsymbol{p}_j | \mu^* p_n \geqslant \sqrt{p_s^2 + p_t^2}\}$, its dual cone is given by $\tilde{\mathscr{C}}_j^{\,*} = \{\tilde{\boldsymbol{g}}_j | <\boldsymbol{p}_j, \tilde{\boldsymbol{g}}_j> \geqslant 0, \forall\, \boldsymbol{p}_j \in \mathscr{C}_j\}$, and its polar cone is defined as $\tilde{\mathscr{C}}_j^{\,\circ} = -\tilde{\mathscr{C}}_j^{\,*}$ [38]. Thus, for any contact-pair, the Coulomb friction law, i.e. Eq. (33), can be written as a *cone complementarity* problem

$$\boldsymbol{p}_j \in \mathscr{C}_j \perp \tilde{\boldsymbol{g}}_j \in \tilde{\mathscr{C}}_j^{\,*} \text{or } \boldsymbol{p}_j \in \mathscr{C}_j \perp -\tilde{\boldsymbol{g}}_j \in \tilde{\mathscr{C}}_j^{\,\circ}. \tag{43}$$

It should be pointed out that the similar result in terms of the relative velocity had been obtained in [39] using the De Saxcé-Feng

bipotential [40].

On the other hand, assume that the j-th contact-pair occurs between ellipsoid-a and ellipsoid-b ($a < b$), one has

$$\boldsymbol{g}_j = \begin{bmatrix} g_n \\ g_s \\ g_t \end{bmatrix} = [\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}]_j^{\mathrm{T}} [\boldsymbol{T}_a(x_j, y_j, z_j), -\boldsymbol{T}_b(x_j, y_j, z_j)] \begin{bmatrix} \Delta \boldsymbol{d}_a \\ \Delta \boldsymbol{d}_b \end{bmatrix} = [\boldsymbol{h}_a^{\mathrm{T}}, \boldsymbol{h}_b^{\mathrm{T}}]_j \begin{bmatrix} \Delta \boldsymbol{d}_a \\ \Delta \boldsymbol{d}_b \end{bmatrix}, \tag{44}$$

where the 3×3 matrix $[\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}]_j$ is given by Eq. (22), which should be a constant matrix once the contact point has been determined. So do the interpolation matrices $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$. Namely, Eq. (44) indicates that g_n, g_s, and g_t are all the linear function of the basic unknown vectors $\Delta \boldsymbol{d}_a$ and $\Delta \boldsymbol{d}_b$, however, the first term $g_n + \mu^* \sqrt{g_s^2 + g_t^2}$ of the vector $\tilde{\boldsymbol{g}}_j$ (see Eq. (41)) is a non-linear function of the same basic unknown vectors, which implies that Eq. (43) is in fact a *non-linear cone complementarity* problem associated with the basic unknown vectors. To avoid the associated numerical difficulties, in this study, a linearized vector $\boldsymbol{g}_j = (g_n, g_s, g_t)^{\mathrm{T}}$ is actually adopted, thus we have the following *linearized cone complementarity* problem

$$\boldsymbol{p}_j \in \mathscr{C}_j \perp \boldsymbol{g}_j \in \mathscr{C}_j^* \text{ or } \boldsymbol{p}_j \in \mathscr{C}_j \perp -\boldsymbol{g}_j \in \mathscr{C}_j^\circ \tag{45}$$

where $\mathscr{C}_j^* = \{\boldsymbol{g}_j | <\boldsymbol{p}_j, \boldsymbol{g}_j> \geqslant 0, \forall\, \boldsymbol{p}_j \in \mathscr{C}_j\}$ and $\mathscr{C}_j^\circ = -\mathscr{C}_j^*$. In Section 5, we will demonstrate the validity of the solution for the linearized cone complementarity problem.

For the whole discrete system, if there are n ellipsoid particles and m contact-pairs in current time step, by collecting all basic unknown vectors of these ellipsoid particles and all contact-pairs among these ellipsoids, we obtain the global governing equation of motion with Coulomb friction law (or the *global linearized cone complementarity* problem) as follows

$$\frac{2}{\Delta t^2} \boldsymbol{M}_{6n \times 6n} \Delta \boldsymbol{D}_{6n \times 1} - \boldsymbol{H}_{6n \times 3m} \boldsymbol{P}_{3m \times 1} = \boldsymbol{F}_{6n \times 1}, \tag{46}$$

where the global equivalent mass matrix $\boldsymbol{M}$ is written as $\boldsymbol{M} = \mathrm{Diag}(\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n)$, which is a block diagonal matrix. The global incremental displacement vector $\Delta \boldsymbol{D}$ is given by $\Delta \boldsymbol{D} = [\Delta \boldsymbol{d}_1, \Delta \boldsymbol{d}_2, \cdots, \Delta \boldsymbol{d}_n]^{\mathrm{T}}$, the global contact force vector $\boldsymbol{P}$ reads $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_m]^{\mathrm{T}}$ (refer to Eq. (41) for each $\boldsymbol{p}$ in $\boldsymbol{P}$), and the $\boldsymbol{F}$ is the global equivalent force vector considering the effect of initial velocity. Note that the global contact gap vector, defined as $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_m]^{\mathrm{T}}$, can be cast into a general form $\boldsymbol{G} = \boldsymbol{H}^{\mathrm{T}} \Delta \boldsymbol{D}$ by assembling Eq. (44). The global transform matrix $\boldsymbol{H}$ is constructed as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_1^1 & \cdots & \boldsymbol{h}_j^1 & \cdots & \boldsymbol{h}_m^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{h}_1^i & \cdots & \boldsymbol{h}_j^i & \cdots & \boldsymbol{h}_m^i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{h}_1^n & \cdots & \boldsymbol{h}_j^n & \cdots & \boldsymbol{h}_m^n \end{bmatrix}_{6n \times 3m}. \tag{47}$$

For any element $\boldsymbol{h}_j^i (i = 1, \ldots, n; j = 1, \ldots, m)$ in $\boldsymbol{H}$, the superscript "i" corresponds to the global index of one ellipsoid, while the subscript "j" indicates the global index of one contact-pair. If the j-th contact-pair is on the i-th ellipsoid, $\boldsymbol{h}_j^i$ will be determined by Eq. (20), otherwise, $\boldsymbol{h}_j^i$ is equal to $\boldsymbol{0}$. Therefore, the global contact gap vector $\boldsymbol{G}$ can be determined using $\Delta \boldsymbol{D}$ solution from Eq. (46), leading to

$$\boldsymbol{G} = \boldsymbol{H}^{\mathrm{T}} \Delta \boldsymbol{D} = \boldsymbol{N} \boldsymbol{P} + \boldsymbol{b} \tag{48}$$

where $\boldsymbol{N} = \frac{\Delta t^2}{2} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{M}^{-1} \boldsymbol{H}$, $\boldsymbol{b} = \frac{\Delta t^2}{2} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{M}^{-1} \boldsymbol{F}$. Finally, we have [38]

$$\boldsymbol{P} \in \mathscr{C} \perp \boldsymbol{G} \in \mathscr{C}^* \text{ or } \boldsymbol{P} \in \mathscr{C} \perp -\boldsymbol{G} \in \mathscr{C}^\circ \tag{49}$$

$$\mathscr{C} = \oplus_{j=1..m} \mathscr{C}_j, \quad \mathscr{C}^* = \oplus_{j=1..m} \mathscr{C}_j^*, \quad \mathscr{C}^\circ = \oplus_{j=1..m} \mathscr{C}_j^\circ. \tag{50}$$

Moreover, from the time steps k to $k+1$, the global acceleration, velocity, and displacement vectors are updated by Newmark integration

$$\boldsymbol{A}^{k+1} = 2\Delta \boldsymbol{D}/\Delta t^2 - 2\boldsymbol{V}^k/\Delta t, \quad \boldsymbol{V}^{k+1} = 2\Delta \boldsymbol{D}/\Delta t - \boldsymbol{V}^k, \quad \boldsymbol{D}^{k+1} = \boldsymbol{D}^k + \Delta \boldsymbol{D}. \tag{51}$$

4.3. Fixed-point iteration algorithm

In order to solve the global linearized cone complementarity problem (see Eqs. (46), (49), and (50)), inspired by [30], we have the following fixed-point iteration algorithm

$$\boldsymbol{P}_{r+1}^k = \tau\, \mathrm{Proj}\{\boldsymbol{P}_r^k - \omega \boldsymbol{A}_r^k [\boldsymbol{N} \boldsymbol{P}_r^k + \boldsymbol{b}_r^k + \boldsymbol{B}_r^k (\boldsymbol{P}_{r+1}^k - \boldsymbol{P}_r^k)]\} + (1-\tau) \boldsymbol{P}_r^k, \tag{52}$$

where the subscript "r" or "$r+1$" denotes the iteration step, the superscript "k" is the current time step. τ and ω are two algorithm parameters. Proj{} is a project operator, which will be detailed in Eqs. (56) and (57). Moreover, the matrix $\boldsymbol{A}_r^k$ is a diagonal matrix, in which each 3×3 subblock $(\boldsymbol{A}_r^k)_j$ corresponds to the j-th contact-pair and defined as

$$(\boldsymbol{A}_r^k)_j = \eta_j \boldsymbol{I} = \frac{3}{\mathrm{Trace}(\boldsymbol{H}_j^{\mathrm{T}} \boldsymbol{M}^{-1} \boldsymbol{H}_j)} \boldsymbol{I}, \tag{53}$$

where $\boldsymbol{I}$ is a 3×3 identify matrix, and $\boldsymbol{H}_j$ is a $6n \times 3$ sub-matrix extracted from the matrix $\boldsymbol{H}$ (see Eq. (47)) corresponding to the j-th contact-pair. In addition, $\boldsymbol{B}_r^k$ is taken as the lower block structure of the matrix $\boldsymbol{N}$ (see Eq. (48)).

Consider the fact that any contact-pair relates only to two ellipsoids, i.e. ellipsoid-a and ellipsoid-b ($a < b$), in practice, we can use a proper data structure to save storage space and improve computational efficiency. For this purpose, we introduce the following data organization:

In Fig. 12, the matrices $\boldsymbol{m}_i$, $\boldsymbol{f}_i$, and $\triangle \boldsymbol{d}_i$ are given by Eq. (16). In Fig. 13, the symbols "$\rightarrow \Delta \boldsymbol{d}_a$" and "$\rightarrow \Delta \boldsymbol{d}_b$" denote two pointer variables pointing to $\Delta \boldsymbol{d}_a$ and $\Delta \boldsymbol{d}_b$, respectively. And the matrices $\boldsymbol{h}_{ja}$ and $\boldsymbol{h}_{jb} (a < b)$ are given by

$$\boldsymbol{h}_{ja} = +\boldsymbol{T}_a^{\mathrm{T}}(x_j, y_j, z_j)[\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}]_j, \quad \boldsymbol{h}_{jb} = -\boldsymbol{T}_b^{\mathrm{T}}(x_j, y_j, z_j)[\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}]_j, \tag{54}$$

where (x_j, y_j, z_j) are the coordinates of the j-th contact point at the beginning of the current time step. And

$$\boldsymbol{\Lambda}_{ja} = \boldsymbol{m}_a^{-1} \boldsymbol{h}_{ja}, \quad \boldsymbol{\Lambda}_{jb} = \boldsymbol{m}_b^{-1} \boldsymbol{h}_{jb}, \quad \eta_j = \frac{3}{\mathrm{Trace}(\boldsymbol{h}_{ja}^{\mathrm{T}} \boldsymbol{m}_a^{-1} \boldsymbol{h}_{ja}) + \mathrm{Trace}(\boldsymbol{h}_{jb}^{\mathrm{T}} \boldsymbol{m}_b^{-1} \boldsymbol{h}_{jb})}. \tag{55}$$

Based on the data structures, the following *fixed point iteration algorithm* can be employed to solve the proposed global linearized cone complementarity problem for a certain time step.

(1) Set algorithm parameters $\tau = 1$, $\omega = 0.2$ similar to that in [30], the maximum iteration number $N_{\max}$ (i.e. 3000), and the tolerance of the maximum of contact force increment $\Delta p_{\max}$ (i.e. 1.0×10^{-6}).
(2) Generate the ellipsoid-based data structure (see Fig. 12) and the contact-pair-based data structure (see Fig. 13).
(3) For $j = 1$ to m // traverse all of contact-pairs
$\boldsymbol{p}_j^0 = [0,0,0]^{\mathrm{T}}$; //initialize the contact force vector
End j
For $r = 1$ to $N_{\max}$
$\Delta p_{\max} = -99999$; //initialize the maximum of contact force increment
For $j = 1$ to m //traverse all of contact-pairs
$\tilde{\boldsymbol{p}}_j^r = \boldsymbol{p}_j^r - \omega \eta_j [\boldsymbol{h}_{ja}^{\mathrm{T}} \Delta \boldsymbol{d}_a^r + \boldsymbol{h}_{jb}^{\mathrm{T}} \Delta \boldsymbol{d}_b^r]$; //calculate a temporary value $\tilde{\boldsymbol{p}}_j^r$
$\tilde{\boldsymbol{p}}_j^r = \mathrm{Proj}\{\tilde{\boldsymbol{p}}_j^r\}$; //calculate the projection of $\tilde{\boldsymbol{p}}_j^r$ (see Eqs. (56) and (57))

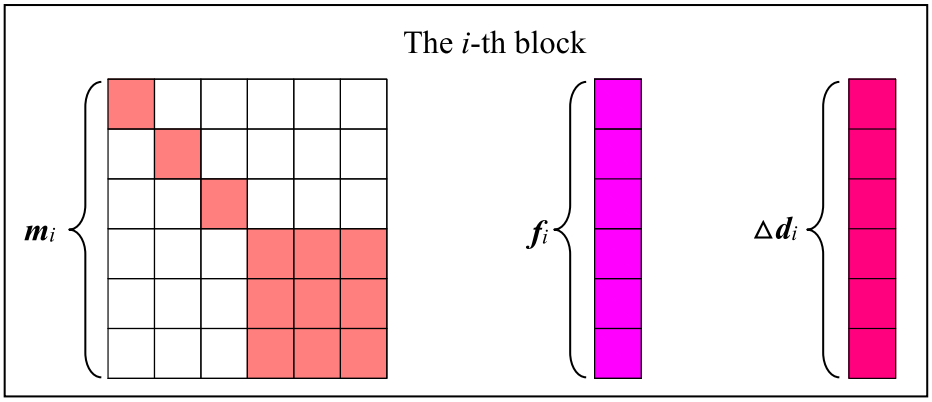


Fig. 12. Body-based data structure for fixed point iteration.

$\boldsymbol{p}_j^{r+1} = \tau\bar{\boldsymbol{p}}_j^{r} + (1-\tau)\boldsymbol{p}_j^{r}$; //update the contact force vector

$\Delta\boldsymbol{p}_j^{r} = \boldsymbol{p}_j^{r+1} - \boldsymbol{p}_j^{r}$; //compute the contact force increment vector

$\begin{cases}\Delta\boldsymbol{d}_a^{r} = \Delta\boldsymbol{d}_a^{r} + \boldsymbol{\Lambda}_{ja}\Delta\boldsymbol{p}_j^{r} \\ \Delta\boldsymbol{d}_b^{r} = \Delta\boldsymbol{d}_b^{r} + \boldsymbol{\Lambda}_{jb}\Delta\boldsymbol{p}_j^{r}\end{cases}$; //update the basic unknown vectors

If $\Delta p_{\max} < |\max(\Delta\boldsymbol{p}_j^{r})|$; //update the maximum of contact force increment

$\Delta p_{\max} = |\max(\Delta\boldsymbol{p}_j^{r})|$;

End j

If $\Delta p_{\max}$ < the tolerance of $\Delta p_{\max}$; //convergence criterion

Stop the fixed point iteration.

End r

Note that we use pointer variables "$\rightarrow \Delta\boldsymbol{d}_a$" and "$\rightarrow \Delta\boldsymbol{d}_b$" to "read" and "write" the basic unknown vectors $\Delta\boldsymbol{d}_a$ and $\Delta\boldsymbol{d}_b$, which therefore can be updated in real time. Additionally, similar to [41], projection of the contact forces are obtained from the following rule:

For the normal contact force

$$\begin{cases}\text{if } p_n < 0, & \text{Proj}\{p_n\} = 0 \\ \text{else}, & \text{Proj}\{p_n\} = p_n\end{cases}. \tag{56}$$

And for the tangent contact forces

$$\begin{cases}\text{if } (p_s^2 + p_t^2) \leqslant (\mu p_n + c)^2, & \text{Proj}\{p_s\} = p_s, \quad \text{Proj}\{p_t\} = p_t. \\ \text{else}, & \text{Proj}\{p_s\} = \frac{p_s(\mu p_n + c)}{\sqrt{p_s^2 + p_t^2}}, \quad \text{Proj}\{p_t\} = \frac{p_t(\mu p_n + c)}{\sqrt{p_s^2 + p_t^2}}.\end{cases} \tag{57}$$

As a geometrical interpretation of the projection, Eq. (57) stated that if $(p_s^2 + p_t^2) > (\mu p_n + c)^2$, the shear contact force will be radially returned onto the friction cone. The flowchat of the proposed DDA method is summarized in Fig. 14.

5. Numerical examples

5.1. Test of the closest point between point and ellipsoid

For any transformed ellipsoid with three semi-axes r_u, r_v, and r_w ($r_u \geq r_v \geq r_w$), we can construct its internal and external spheres by using r_w and r_u respectively, as shown in Fig. 15. To validate the proposed semi-analytic geometry iteration algorithm, we firstly generate 1000 random checking point $\mathrm{P}^{\mathrm{Random}}$ on the internal or external spheres, then find out the closest point $\mathrm{I}^{\mathrm{Closest}}$ by using our algorithm. In Fig. 15, the vector $\boldsymbol{n}$ is the unit outward vector of the ellipsoid at the closest point $\mathrm{I}^{\mathrm{Closest}}$. For the internal random checking point, if the angle between vectors $\mathrm{I}^{\mathrm{Closest}}\overrightarrow{\mathrm{P}^{\mathrm{Random}}}$ and $\boldsymbol{n}$ is close to π or 180°, abbreviated as "angle($\mathrm{I}^{\mathrm{Closest}}\overrightarrow{\mathrm{P}^{\mathrm{Random}}}$, $\boldsymbol{n}$) = $\pi \pm 1.0 \times 10^{-4}$", the iteration will be terminated. Whereas, for the case of the external random checking point, the iteration termination condition is defined as "angle ($\mathrm{I}^{\mathrm{Closest}}\overrightarrow{\mathrm{P}^{\mathrm{Random}}}$, $\boldsymbol{n}$) = 1.0×10^{-4}", which means that the two vectors $\mathrm{I}^{\mathrm{Closest}}\overrightarrow{\mathrm{P}^{\mathrm{Random}}}$ and $\boldsymbol{n}$ are almost in the same direction. It should be pointed out that for an angle value, a tolerance value of 10^{-4} is

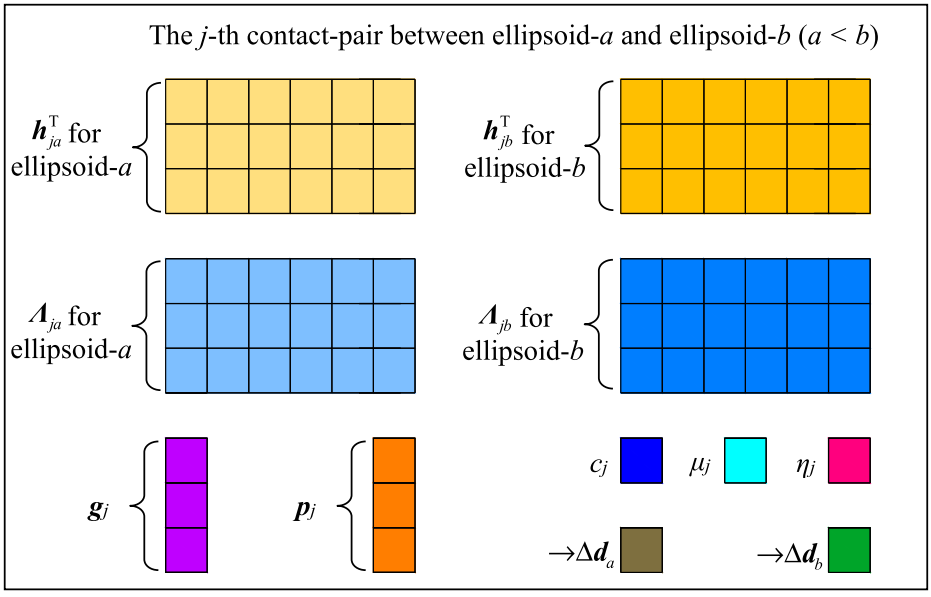


Fig. 13. Contact-pair-based data structure for fixed point iteration.

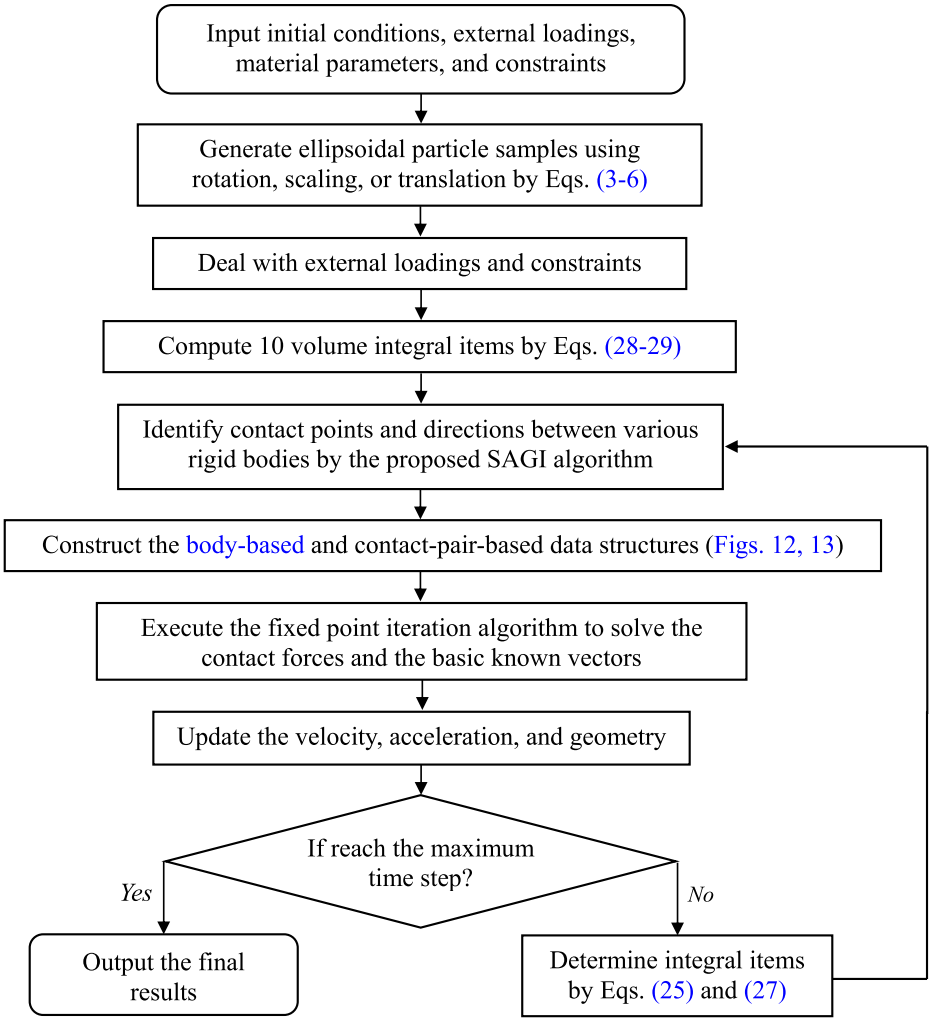


Fig. 14. Flowchart of the proposed DDA method.

sufficiently accurate. Moreover, the maximum number of iterations is set to be 20 for all cases.

The iteration curve of a random internal and external checking point is depicted in Fig. 16(a) and (b). As we can see, for the random internal checking point, the angle($I^{Closest}\vec{P}^{Random}$, $\boldsymbol{n}$) is about equal to 3.14156 rad at 6-th iteration step, which implies that the iteration termination condition is achieved. Meanwhile, the corresponding nearest distance given by the proposed algorithm is about equal to 0.15303 m (see Fig. 16(a)). On the other hand, after 4 iteration steps, the angle($I^{Closest}\vec{P}^{Random}$, $\boldsymbol{n}$) and the nearest distance are respectively about equal to 0.00001 rad and 0.07896 m, satisfying the convergence criterion (see Fig. 16(b)). For the 2000 random checking points, the numbers of iteration steps are further summarized in Fig. 17.

For the 1000 random internal checking points, the mean, minimum,

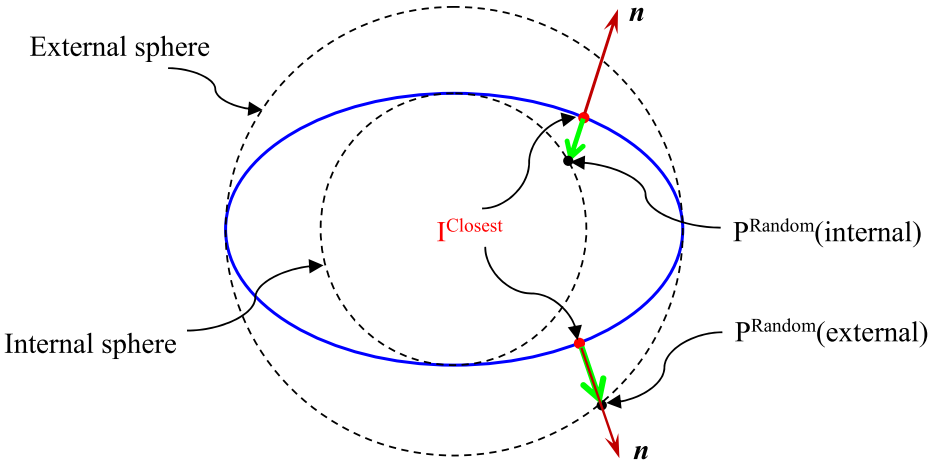
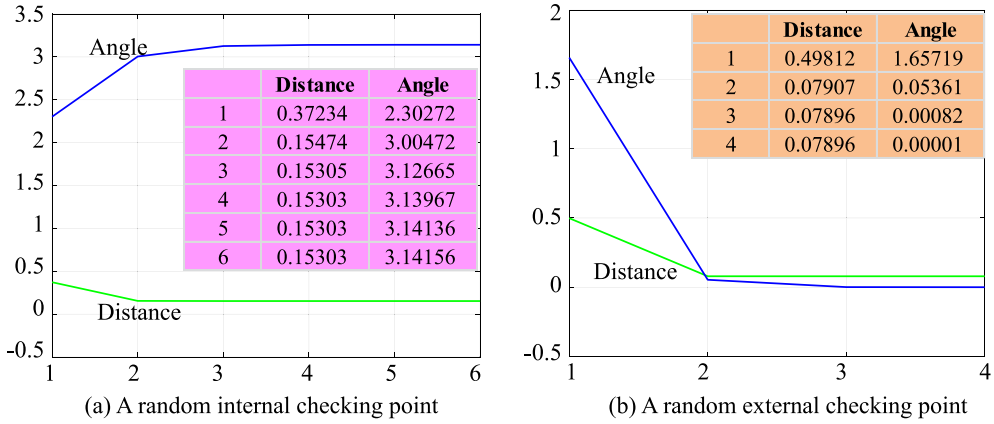
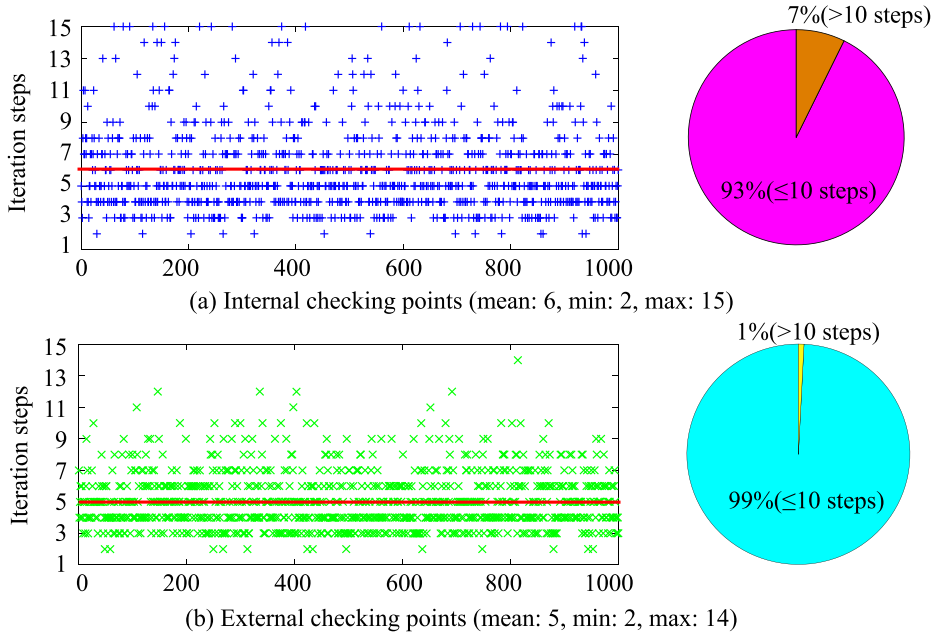


Fig. 15. Ellipsoid and its internal and external spheres.



(a) A random internal checking point (b) A random external checking point

Fig. 16. Iteration curve for a random checking point. (unit: m, rad).



(a) Internal checking points (mean: 6, min: 2, max: 15)

(b) External checking points (mean: 5, min: 2, max: 14)

Fig. 17. Iteration steps for 1000 random checking points.

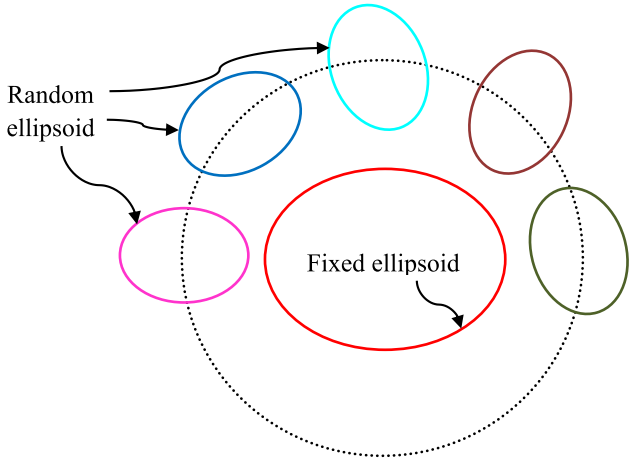


Fig. 18. 1000 pairs of random ellipsoids.

and maximum of iteration steps are about equal to 6, 2, and 15, respectively. While for the 1000 random external checking points, the corresponding values are about equal to 5, 2, and 14, respectively. By doing statistical analysis, we find that there are 93% (or 99%) of random internal (or external) checking points, whose iteration steps are less than 10 steps (see Fig. 17, in which the red line marks the mean of iteration steps). This indicates that for the great majority of random checking points the proposed algorithm can be achieved convergence within 10 iteration steps.

5.2. Test of the closest point between two ellipsoids

In order to validate the proposed algorithm for the case of two ellipsoids, we generate randomly 1000 ellipsoids around a fixed ellipsoid, such that we obtain 1000 pairs of random ellipsoids, as sketched in Fig. 18. As is known, based on the parametric equation (see Eq. (7)) of ellipsoid, finding the closest points between a point and an ellipsoid or between two ellipsoids can be converted to an optimization problem by setting up the squared distance function between two objects. In the MATLAB environment, the procedure can be implemented easily by using a build-in subroutine "*fminsearch*". For both the 1000 pairs of ellipsoids and the 2000 random checking points in the Section 5.1, we find that the closest points between any pair of geometric objects given by *fminsearch* and the proposed algorithm are almost identical. However, the computational efficiency of the proposed algorithm is significantly superior than *fminsearch*. Therefore, we only demonstrate

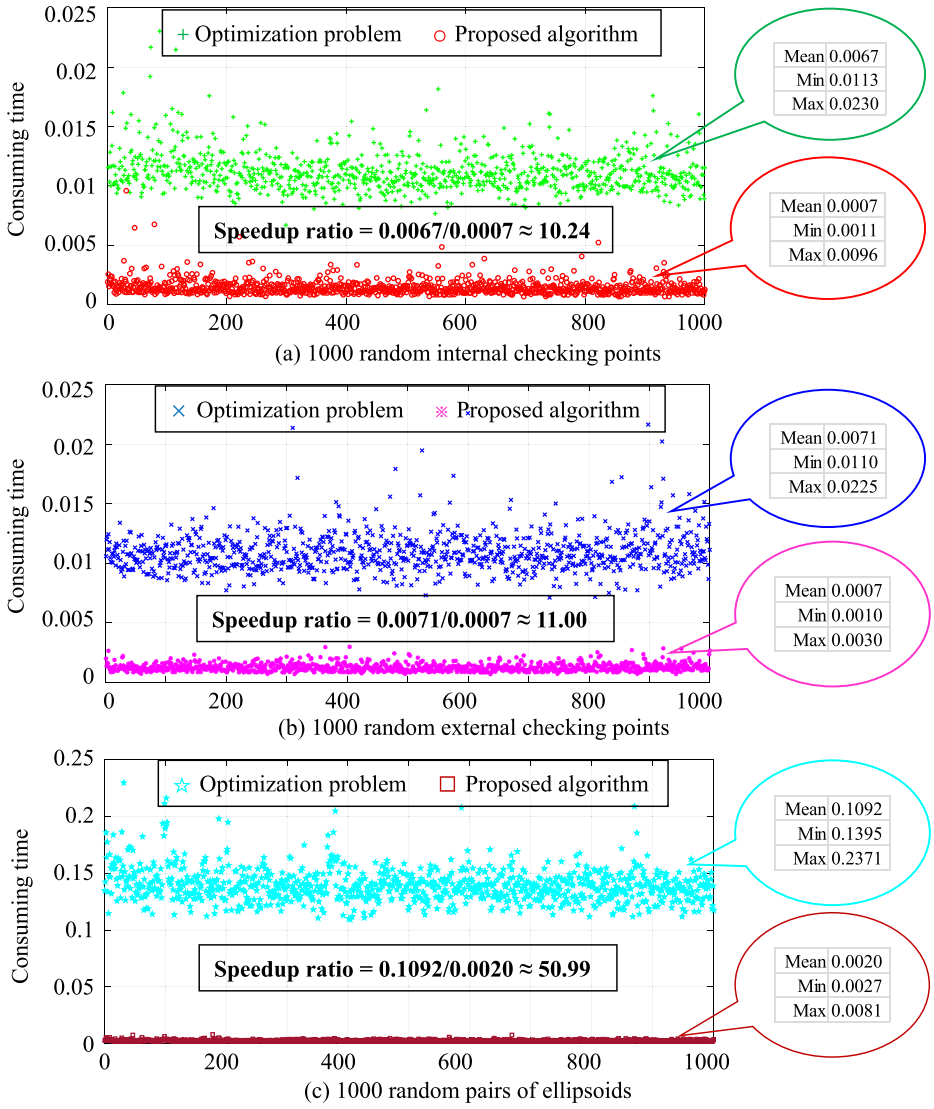


Fig. 19. Comparisons of consuming time. (unit: s).

comparisons in computational efficiency, as shown in Fig. 19.

From Fig. 19, we can observe, for the 1000 random internal checking points, the consuming time of the *fminsearch* optimization (see Fig. 19(a)) falls into the range of [0.0113, 0.0230] s, the corresponding mean is about equal to 0.0067 s. By contrast, that of the proposed SAGI algorithm is within the interval of [0.0011, 0.0096] s, with a mean close to 0.0011 s. As for the 1000 random external checking points, the consuming time of the *fminsearch* (see Fig. 19(b)) is between 0.0110 s and 0.0225 s, the mean is about 0.0110 s. Whereas, the corresponding result associated with the proposed algorithm is between 0.0010 s and 0.0030 s, the mean is about 0.0007 s. Moreover, for the 1000 random pairs of ellipsoids, the consuming time of the *fminsearch* (see Fig. 19(c)) is limited to interval of [0.1395, 0.2371] s. Accordingly, the mean is about equal to 0.1092 s. Comparatively speaking, the consuming time of the proposed algorithm is within range of [0.0027, 0.0081] s. The mean is about 0.0020 s. By comparing the mean of consuming time of the two methods, for the three cases we can estimate the corresponding *speedup ratios* are about equal to 10.24, and 11.00, and 50.99, respectively. In the Appendix, algorithms for detecting the closest points between ellipsoid and cylinder/truncated cone are also presented. The algorithms also have speedup ratios about 50 compared with the Matlab subroutine. It testified that our algorithm has a much faster rate of convergence compared with the traditional method.

5.3. Conservation of energy and momentum

In this example, the conservation of energy, linear momentum, and angular momentum will be demonstrated. Firstly, the free falling of an ellipsoid with the centroid C_1 is simulated. Under the action of the gravity $\boldsymbol{G}$ the ellipsoid will move downwards. Due to the contact force, it will rebound after touching the fixed plane underneath. The distance between the lowest point A and the fixed plane will be observed and its value is equal to 1 m at the initial time, as shown in Fig. 20(a). Moreover, let the time step $\Delta = 0.001$ s, the material density of body $\rho = 1$ kg/m^3, the acceleration of gravity $g = -10$ m/s^2, and the total time is 10 s. Some results are depicted in Fig. 21.

The calculation time when the ellipsoid contacts the fixed plane is equal to 0.447 s, which is almost the same as the theoretical solution, see Fig. 21. Meanwhile, the period of motion obtained from the proposed DDA method is about 0.894 s, which is very close to the analytical one. Within 10 s, we can observe 11 complete cycles. In addition, the maximum distance between point A and the fixed plane is always equal to 1 m. This is to say that our DDA method can ensure the conservation of energy.

In order to test the conservation of linear momentum, we assign the initial translational velocity $\boldsymbol{V}_0(1,0,1)$ m/s to a sphere C_2, which is moving within a fixed cubic boundary, as shown Fig. 20(b). We set the

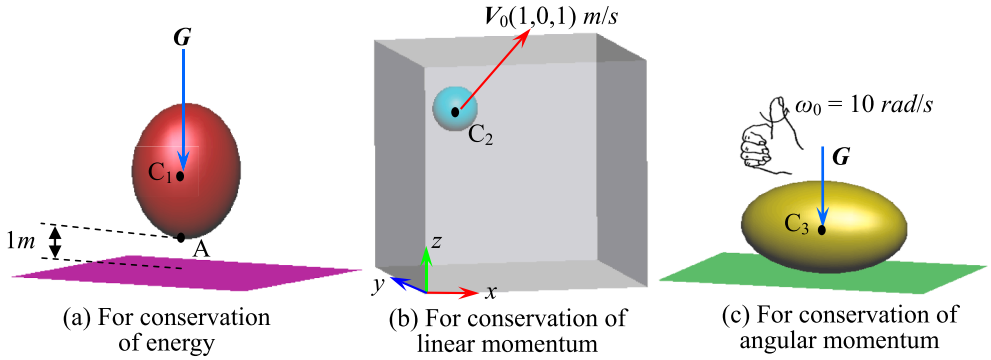


Fig. 20. Configuration for conservation of energy and momentum.

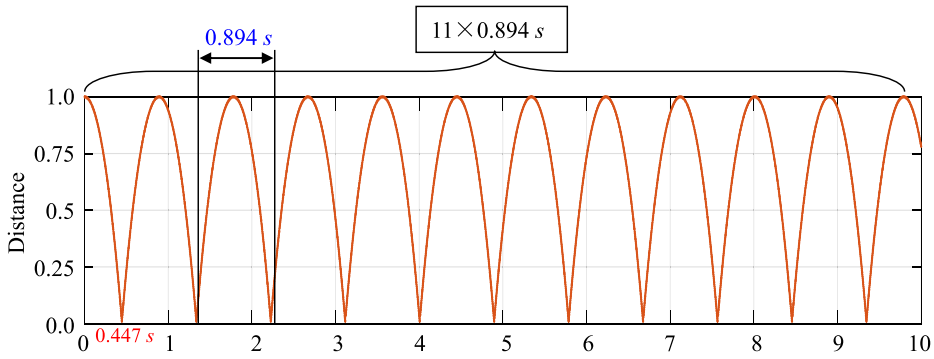


Fig. 21. Free falling of an ellipsoid (unit: m, s. Refer to Fig. 21(a)).

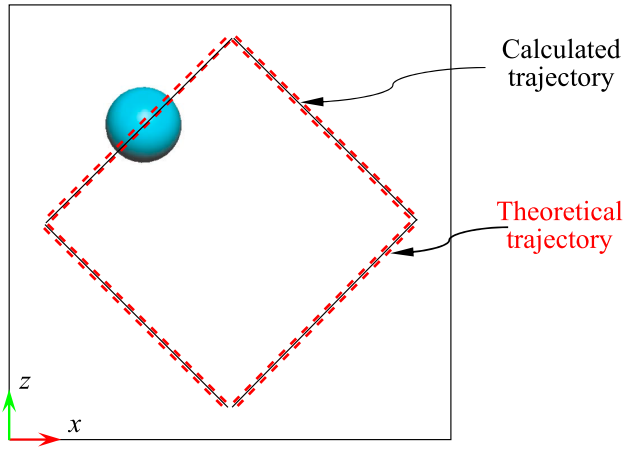


Fig. 22. Trajectory of the centroid C_2 (refer to Fig. 20(b)).

time step $\Delta = 0.001$ s, the total time 10 s, and the material density of body $\rho = 1$ kg/m^3. Gravity as well as friction between the sphere and boundary are not considered. Under these conditions, the sphere will move periodically due to conservation of linear momentum. The trajectory of the centroid C_2 is sketched in Fig. 22 (black solid line), which coincides with the theoretical trajectory (red double dot line). This indicates that the linear momentum is conserved in the proposed DDA method.

Next, we consider the spinning of an ellipsoid in Fig. 20(c). An initial angular velocity $\omega_0 = 10$ rad/s is specified to an ellipsoid around its central axis on a frictionless fixed plane. The material density of body $\rho = 1$ kg/m^3, the acceleration of gravity $g = -10$ m/s^2, the time step $\Delta = 1$ s, and the total calculation step is set to be 1000 s. Because the ellipsoid contacts with the fixed plane at its lowest point, if the contact force can be predicted correctly by the proposed DDA method, the ellipsoid will rotate around z-axis with a constant angular velocity ω_0, meanwhile, the angular velocities around x- and y-axes both should be equal to zero. Based on this reason, the calculated angular velocities around x, y, and z-axes will be tested, shown in Fig. 23. We can find that the absolute error of angular velocities around x-, y-, and z-axes is in the order of 10^{-12}, which are sufficiently small, and that the angular velocity around z-axis is almost equal to the preset constant value of 10 rad/s. Consequently, the conservation of angular momentum has been verified.

5.4. Frictional rolling of a sphere

A sphere with the radius $R = 1$ m is on a fixed plane, as shown in Fig. 24. $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{T}$ are the constant point loading, the gravity, and the friction, respectively. $d \in [0, 2]$ m is the distance between the action line of $\boldsymbol{F}$ and the static plane, and $\boldsymbol{V}_0$ is the initial velocity of the centroid of the sphere. We let the material density of body $\rho = 1$ kg/m^3, the acceleration of gravity $g = -10$ m/s^2, and the time step $\Delta = 0.001$ s.

Now, consider the first case: Set the friction factor to be 0.1, the initial velocity $\boldsymbol{V}_0 = 0$, the point loading $\boldsymbol{F} = 1$ N, and the distance d ranges from 0 to 2 m. In this case, the maximum of the friction between the sphere and the plane $\boldsymbol{T}_{\max} = 4\pi/3 = 4.18879$ N, which is greater than $\boldsymbol{F}$. In this case, the sphere will do a pure rolling motion rightwards. Theoretically, the real friction $\boldsymbol{T}$ is analytically given by $\boldsymbol{T} = \boldsymbol{F}(1 - 5d/7)$. For comparison, the calculated values of the friction $\boldsymbol{T}$ are summarized in Fig. 25 as well as the corresponding theoretical values.

In Fig. 25, we can observe that the friction decreases gradually along with the increase in distance d. When $d = 1.4$ m, the corresponding friction is equal to zero. Then, the friction will change its direction, i.e. from the leftward changes to the rightward. Meanwhile, it

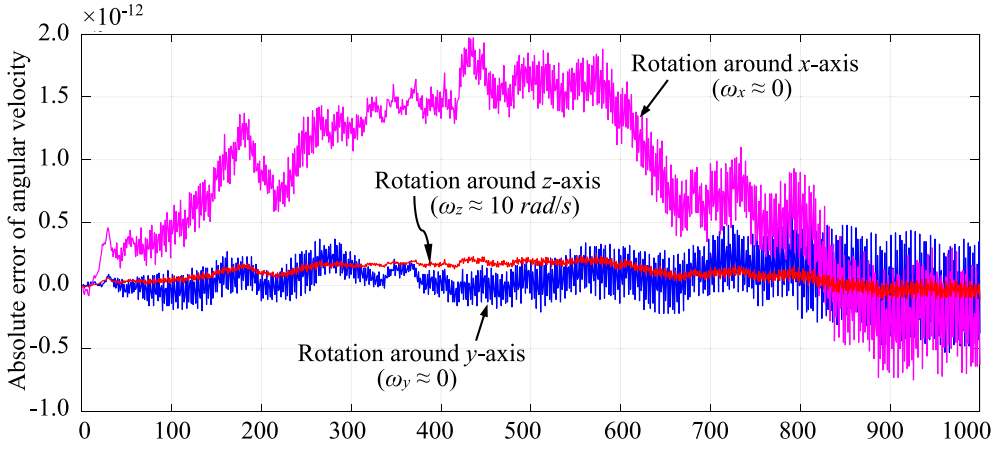


Fig. 23. Angular velocities of an ellipsoid (unit: rad/s, s; refer to Fig. 20(c)).

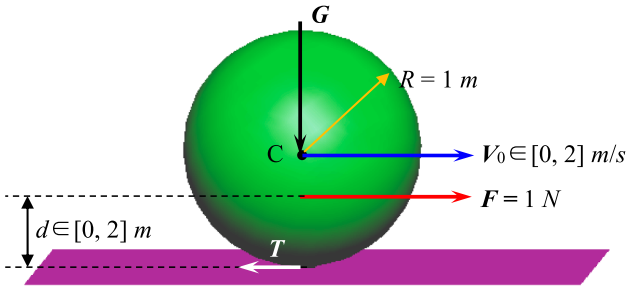


Fig. 24. Configuration for fictional rolling of a sphere.

can be noted that the calculated frictions obtained from the proposed DDA method are almost coincide with that given by analytical formula. Namely, as we can see from Fig. 25, the data points (theoretical frictions) marked by the red "+" symbols and the data points (calculated frictions) denoted by the blue "○" symbols overlap with each other. This is to say that the computational accuracy of our DDA method is highly satisfactory.

Next, set the friction factor to be 0.1, the initial velocity $\boldsymbol{V}_0$ increase gradually from 0 to 2 m/s by 0.2 m/s, and the point loading $\boldsymbol{F} = 0$ *N*. In this case, at the first stage of the motion, because the relative sliding exists the between the sphere and the fixed plane, the friction must be up to the maximum value, and the sphere decelerates gradually. Then, at a

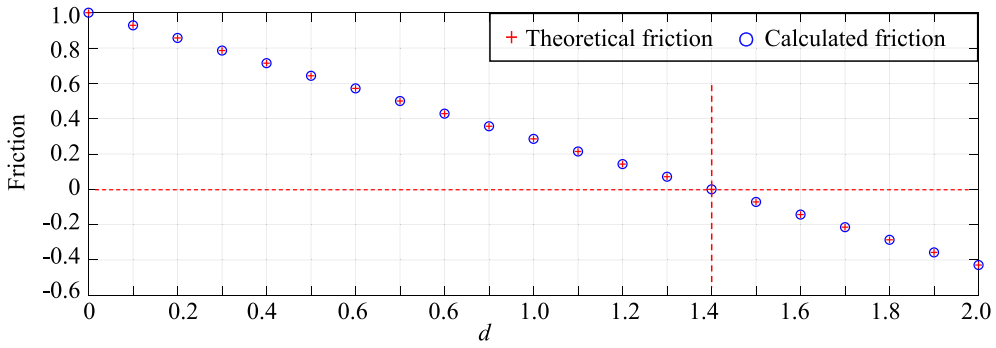


Fig. 25. Friction corresponding to the different *d* (unit: N, m).

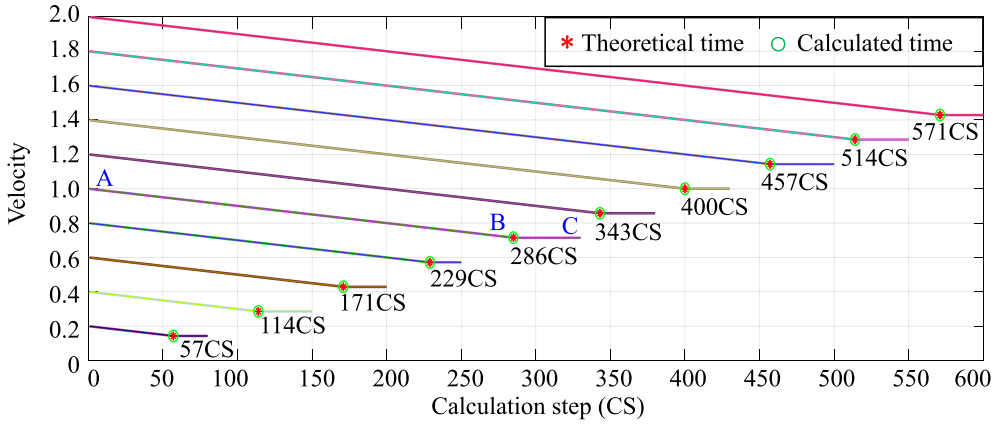


Fig. 26. Velocity *vs.* calculation step (unit: m/s).

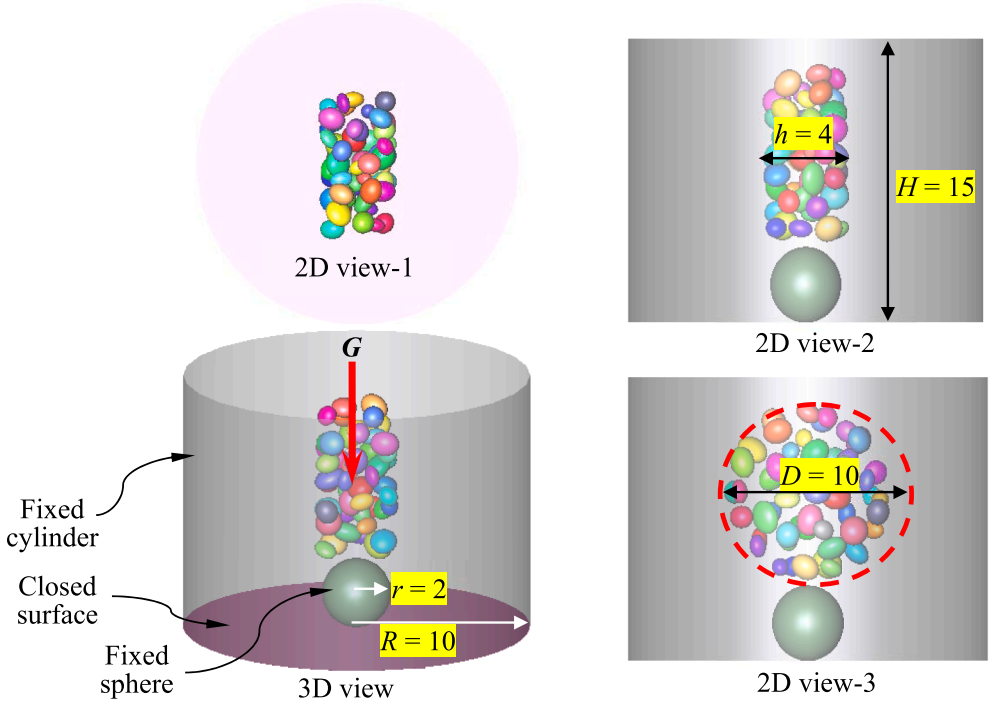


Fig. 27. Configuration for falling of ellipsoids in a cylinder (unit: m).

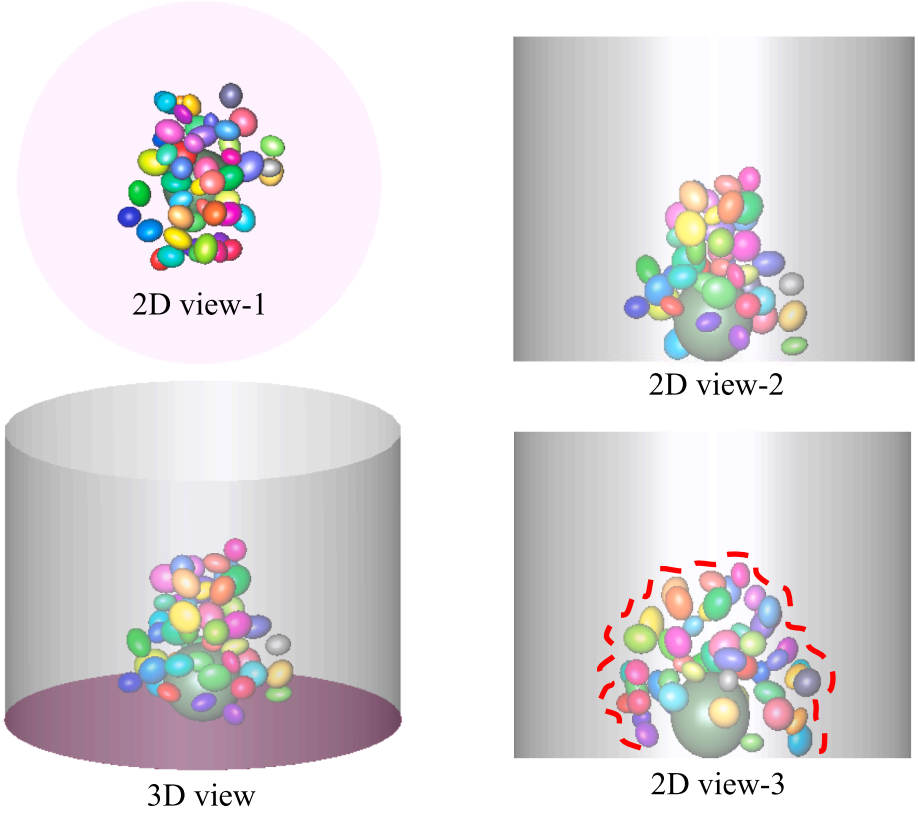


Fig. 28. Configuration of 50 ellipsoids in a cylinder (CS = 1000).

crucial moment, the sphere begins to do pure rolling motion without sliding. This means that the acceleration value $|\boldsymbol{a}_C|$ of the centroid should be R times of the angular acceleration β around the contact point between the sphere and the fixed plane. Namely, $|\boldsymbol{a}_C| = R\beta$. Therefore, the critical time can be computed by $t = 2|\boldsymbol{V}_0|/7$. We will predict the critical time by using the proposed DDA method. When a relatively constant velocity is achieved, the corresponding calculation step is considered as the critical time. Some results are illustrated in Fig. 26.

The ten velocity-calculation step (CS) curves are plotted in Fig. 26. At CS = 0, the velocities are the given initial velocities of the centroid

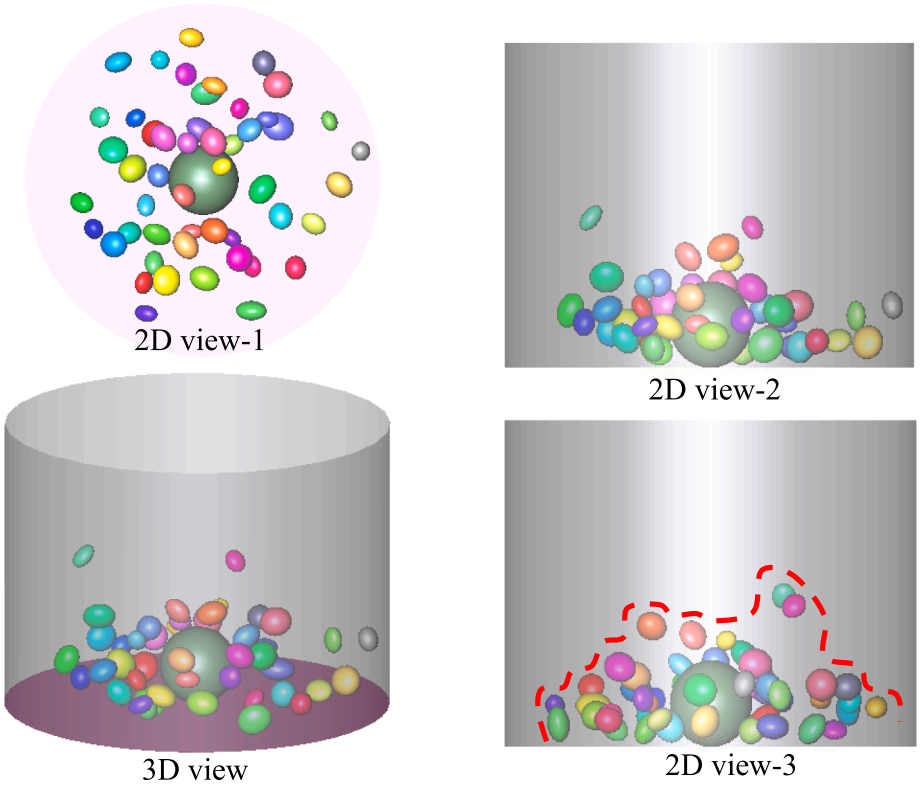


Fig. 29. Configuration of 50 ellipsoids in a cylinder (CS = 1500).

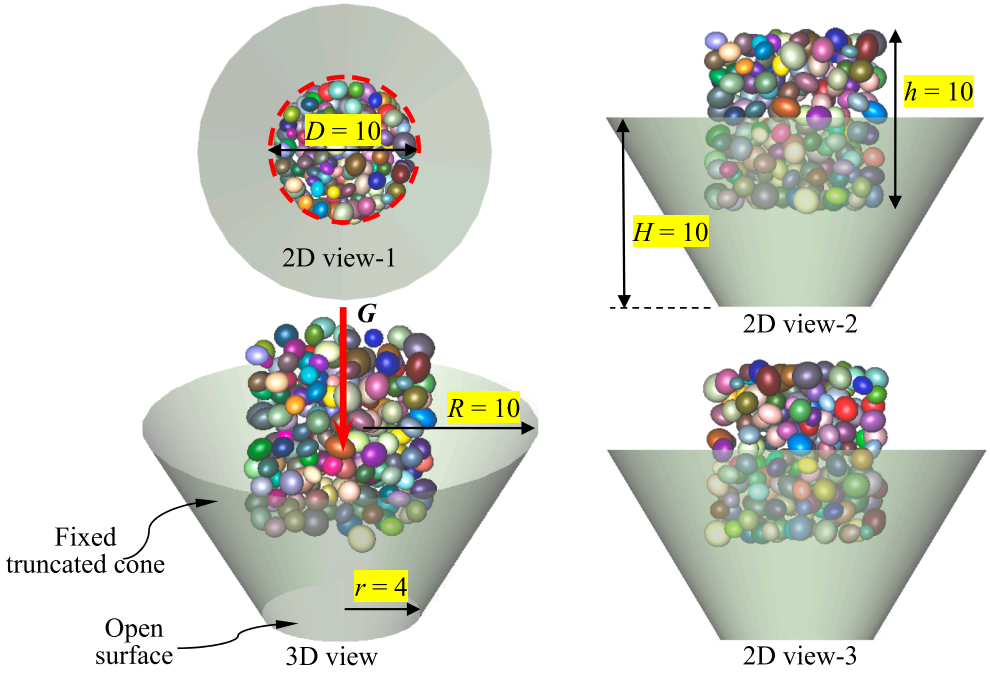


Fig. 30. Configuration for motion of ellipsoids in a truncated cone (unit: m).

of the sphere. Taking the initial velocity $\boldsymbol{V}_0$ = 1.0 m/*s* as an example, i.e. the polyline ABC in Fig. 26, the line segment AB denotes the decelerated motion of the sphere, while at the stage represented by the line segment BC the sphere attains a relative stable velocity. Obviously, the point B is turning point, where the sphere has just begun doing the pure roll without sliding. For the case of $\boldsymbol{V}_0$ = 1.0 m/*s*, the calculation step corresponding to the point B is about equal to 285CS, i.e. 285 × 0.001 = 0.285 s that is very close to the theoretical time. In the same fashion, we can scrutinize the other nine curves. As we can find, the calculated critical times are in good agreement with the theoretical ones. In other words, the propose DDA method has the ability to capture complex movement process.

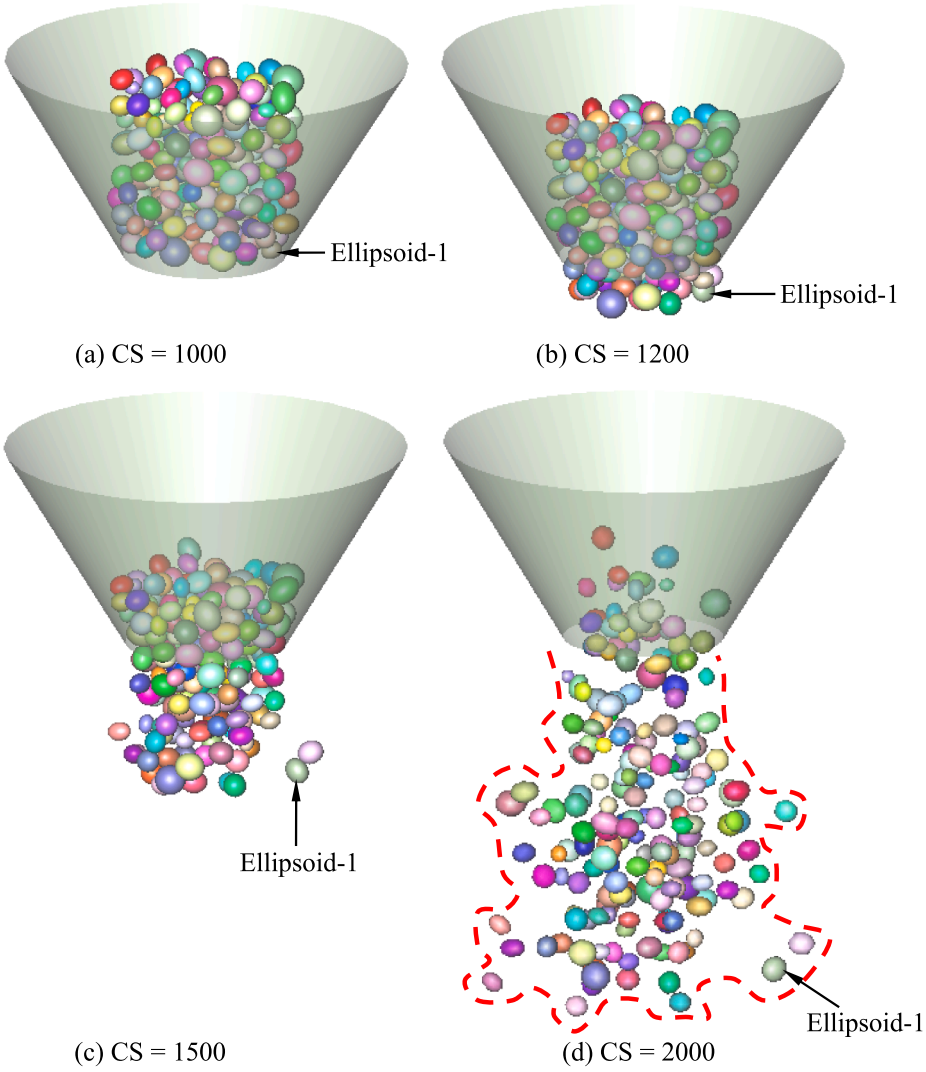


Fig. 31. Configuration of 200 ellipsoids in a truncated cone (3D view).

5.5. Movement of ellipsoids in a cylinder

There are 50 ellipsoids in a fixed cylinder with a closed bottom surface, as shown in Fig. 27. On the bottom of the cylinder there is a static sphere. The heigh and radius of the cylinder are $H = 15$ m and $R = 10$ m, respectively. The radius of the sphere is $r = 2$ m. The 50 ellipsoids are generated ramdomly with the constraint that the three semi-axes r_u, r_v, and r_w ($r_u \geq r_v \geq r_w$) are all within the interval of [0.6, 1.0] m. From the perspective of the outside outline, the 50 ellipsoids form a cylindrical shape with the diameter $D = 10$ m and the heigh $h = 4$ m, as exhibited in Fig. 27. Under the action of the gravity, these ellipsoids fall downwards. Additionally, the material density of body $\rho = 1$ kg/m^3, the acceleration of gravity $g = -10$ m/s^2, the time step $\Delta = 0.001$ s, the total calculation step (CS) is 1500, and the friction is ignored. Some results are demonstrated in Figs. 28 and 29.

At CS = 1000, among the 50 ellipsoids, some of ellipsoids are colliding with the fixed sphere. At the same time, some of them have been contacted with the closed bottom surface of the cylinder. Because of the interactions of these rigid bodies, the outline of the 50 ellipsoids is now highly irregular (see the red solid line in Fig. 28) and no longer looks like a cylinder.

Due to the gravity, the contact force, and the inertia, the 50 ellipsoids keep moving. When CS = 1500, these ellipsoids are scattered around the static sphere, as presented in Fig. 29. Meanwhile, we can find, as expected, that the closed bottom surface of the cylinder serves as a barrier that restricts these ellipsoids to be within the cylinder. This example shows that the interactions between ellipsoids and cylinders can be simulated by the proposed DDA method.

5.6. Motion of ellipsoids in a truncated cone

Now, the movement of 200 ellipsoids is simulated. Their three semi-axes r_u, r_v, and r_w ($r_u \geq r_v \geq r_w$) are all distributed randomly within the interval of [0.6, 1.0] m. The initial configuration is described in Fig. 30. The radiuses of the smaller and larger surfaces of the fixed truncated cone are given by $r = 4$ m and $R = 10$ m, and its heigh $H = 10$ m. Intially, the 200 ellipsoid assmeble like a cylinder with the diameter $D = 10$ m and the heigh $h = 10$ m. At the initial time, the nearest distance between these ellipsoids and the bottom surface of the truncated cone is equal to 5 m, i.e. $d = 5$ m (see Fig. 30). Note that the smaller surface of the truncated cone does not be closed. The gravity ($g = -10$ m/s^2) is the only external loading. And the material density of body $\rho = 1$ kg/m^3, the time step $\Delta = 0.001$ s, the total calculation step (CS) is 2000. Some results are demonstrated in Figs. 31 and 32.

At CS = 1000, the ellipsoids located at the lowest layer reach at the level of bottom surface of the truncated cone, see Fig. 31(a). During their descent, because that the diameter of the truncated cone is getting smaller and smaller along the direction of the gravity, these ellipsoids

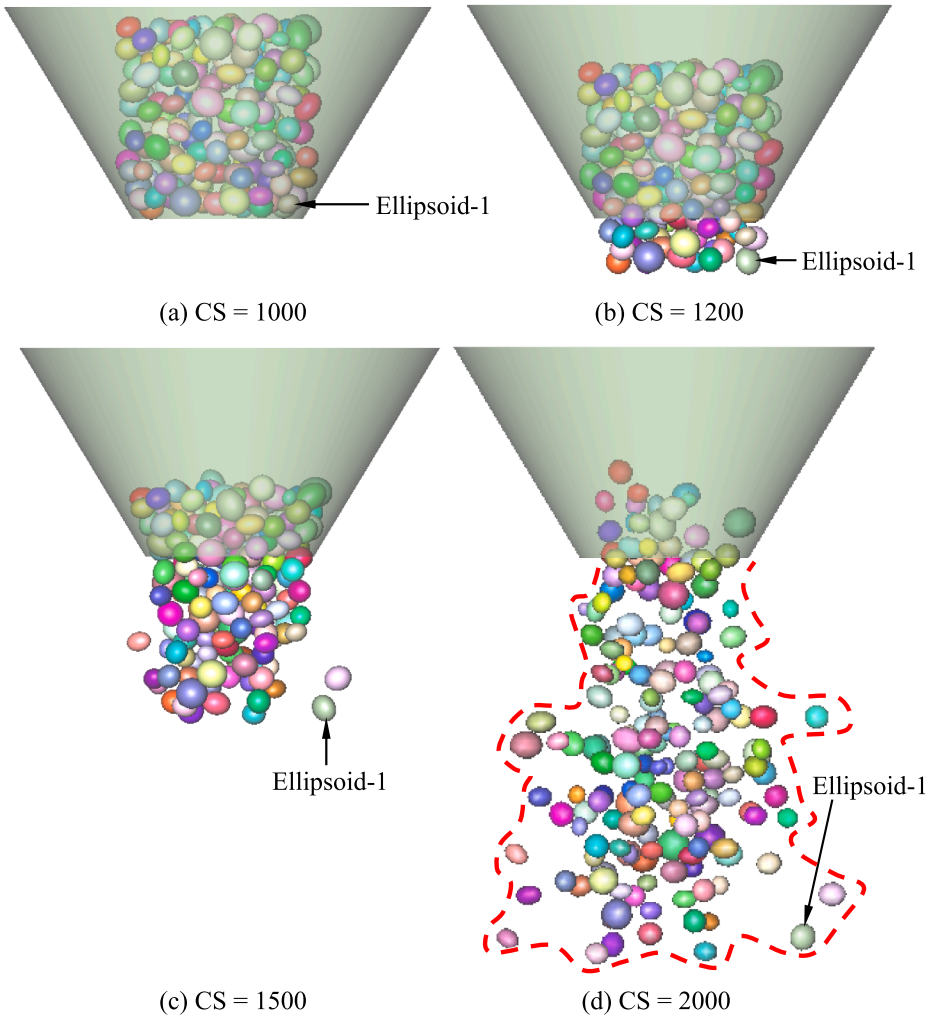


Fig. 32. Configuration of 200 ellipsoids in a truncated cone (2D view).

are forced to pass through a reduced section at the bottom open surface (refer to the two red solid curves in Fig. 31(a)). As time goes on, these ellipsoids continue to fall down. At CS = 1200, 1500, and 2000, their positions are presented in Fig. 31(b), (c), and (d), respectively. In our simulation, the motion of any interested body can be investigated in detail. For instance, we can observe the orientation of the ellipsoid-1 in Fig. 31(a), (b), (c), and (d). Further, the two-dimensional views of these results are exhibited in Fig. 32.

Fig. 32 can help us to understand the falling process of this group of ellipsoids. As we can see, these ellipsoids pass step by step through the bottom open surface of the truncated cone. At CS = 2000, i.e. Fig. 32(d), it can be found out that these ellipsoids outside of the truncated cone are scattered to a larger region because of repulsive contact forces between particles. This example further demonstrates the ability of the proposed DDA method to modeling complex motion.

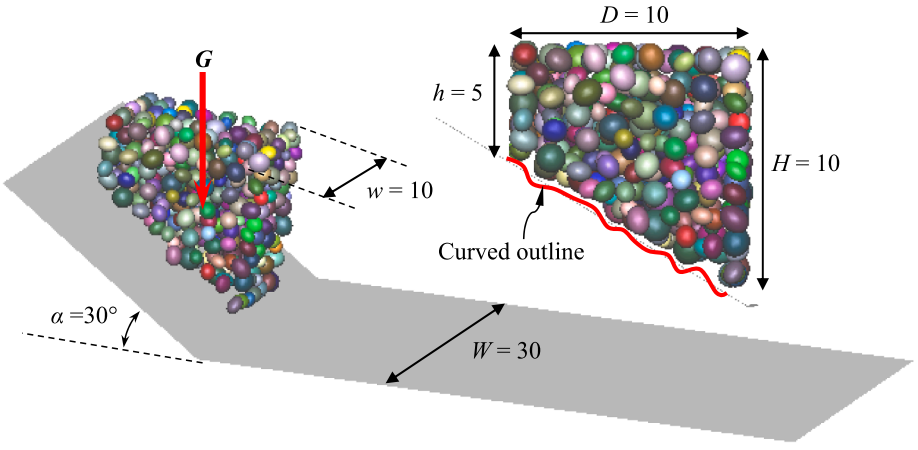


Fig. 33. Initial configuration for sliding of ellipsoids on a slope (unit: m).

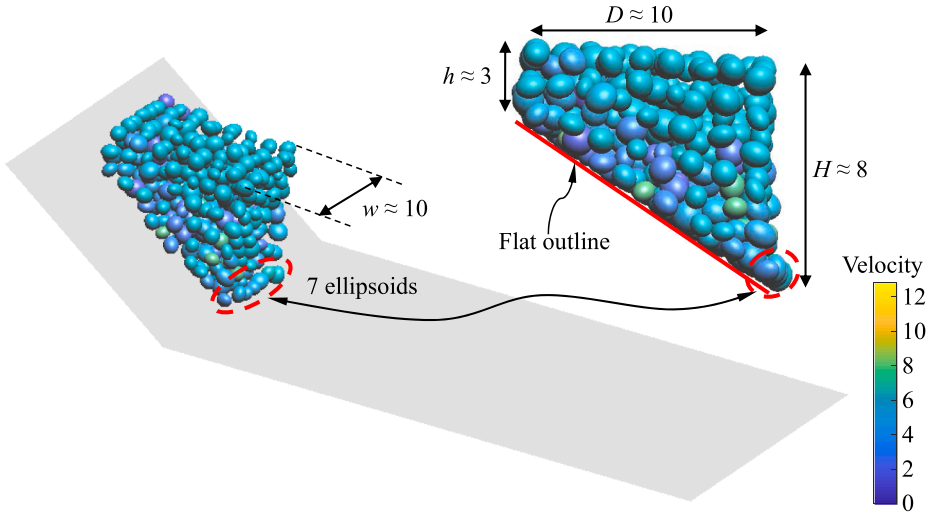


Fig. 34. Configuration of a group of ellipsoids on a slope (CS = 600, unit: m, m/s).

5.7. Collapse of ellipsoids on a slope

Placed on a slope with an angle $\alpha = 30^{\circ}$ and width $W = 30$ m are 400 ellipsoids, whose three semi-axes r_u, r_v, and r_w ($r_u \geq r_v \geq r_w$) are generated randomly from the range of [0.6, 1.0] m. The assembly has an initial configuration of $D = 10$ m, $H = 10$ m, $w = 10$ m, and $h = 5$ m, as outlined in Fig. 33. A curved red outline is also shown to indicate the boundary of ellipsoids that are immediately close to the slope. Under the action of the gravity ($g = -10$ m/s^2) and friction ($\mu = 0.1$), the collapse of the group of ellipsoids is modeled. Moreover, the material density of body $\rho = 1$ kg/m^3, the time step $\Delta = 0.001$ s, and the total calculation step (CS) is set to be 1500. Some results are presented in Figs. 34 and 35.

About at CS = 600, the group of ellipsoids become denser. Now, we estimate the corresponding dimensions as: $D \approx 10$ m, $H \approx 8$ m, $w \approx 10$ m, and $h \approx 3$ m. Apparently, along the direction of gravity the two

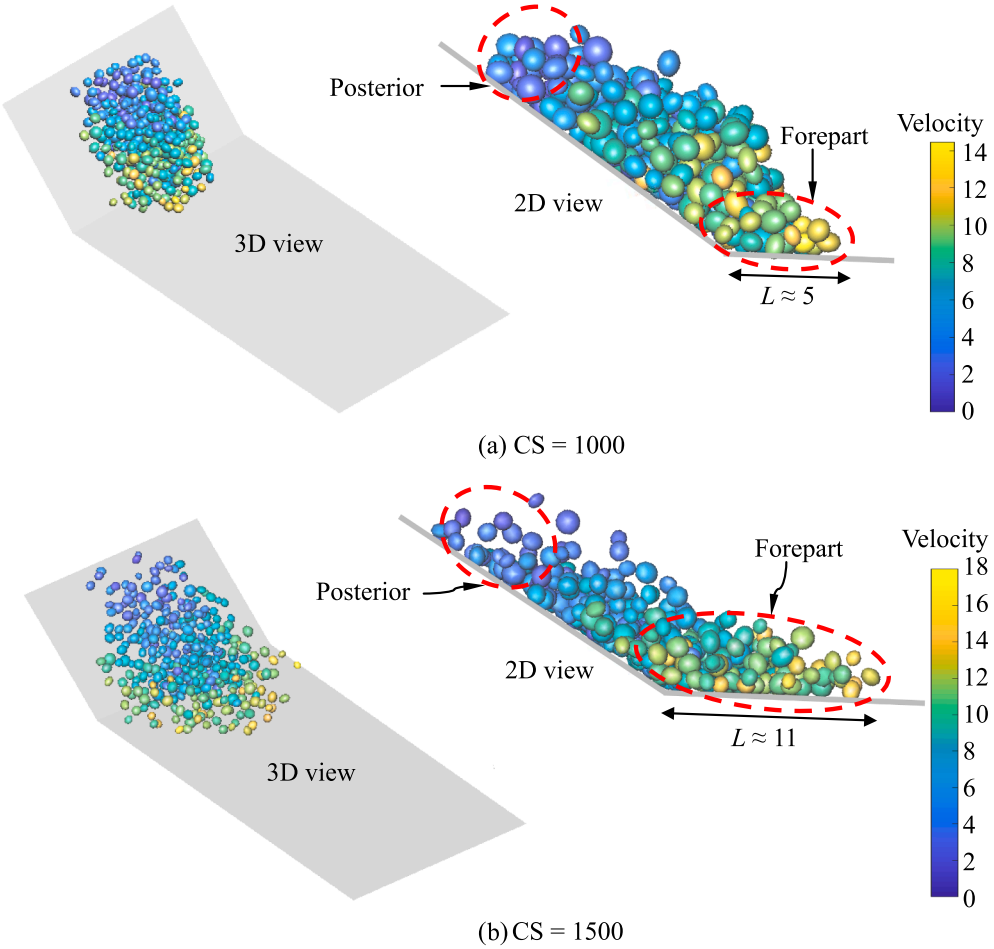


Fig. 35. Configuration of a group of ellipsoids on a slope (CS = 1000 and 1500, unit: m, m/s).

heights H and h are smaller than the initial values. Relatively speaking, along the ellipsoids, which are immediately close to the slope, we can draw a flat outline (see the red solid line in Fig. 34) rather than an irregular outline (see the red solid line in Fig. 33). In addition, from Fig. 34 it is can be found that there seem to be 7 ellipsoids that are squeezed out firstly. At this moment, among these ellipsoids the maximum of the velocity of centroids is about equal to 12.51 m/s. The positions of these ellipsoids corresponding to CS = 1000 and CS = 1500 are further described in Fig. 35.

Due to the action of the gravity, contact with friction, and inertia, these ellipsoids continue to collapse and slide along the inclined slope and the horizontal plane, as shown in Fig. 35. As CS = 1000, the maximum of the velocity of centroids is about equal to 14.46 m/s among these ellipsoids. Meanwhile, the posterior of the ellipsoids occur collapse towards the back of the slope. When CS = 1500, the velocity of centroids reaches as fast as 17.33 m/s. At the same time, the collapse becomes more apparent. As for the forepart of these ellipsoids, the sliding distances towards the front of the slope are about equal to 5 m and 10 m respectively corresponding to CS = 1000 and 1500. Simultaneously, these ellipsoids also run off towards the two sides of the slope. Through this example, the validity and application prospect of the proposed DDA method is further illustrated. Further, the proposed method can be used to solve more complex, real engineering problems, such as landslide hazard assessment, in which, reliability method such as interval non-probabilistic reliability method [42] becomes important to handle uncertainty in material properties.

6. Conclusions

In this study, an accurate and efficient numerical simulation platform was developed for modeling ellipsoidal particles and frictional contact between them. In the proposed framework, firstly, the geometric convexity of ellipsoid itself has been fully exploited to create a semi-analytic geometry iteration (SAGI) algorithm for identifying the contact positions and contact directions between rigid bodies. In the appendix, algorithms for the closest points between ellipsoid and cylindrical/truncated conical boundaries are also presented. In terms of the computational efficiency, some designed examples have demonstrated that the SAGI algorithm is significantly superior than the traditional optimization algorithm without any loss of accuracy. Secondly, based on the principle of minimum potential energy, the discontinuous deformation analysis (DDA) was employed to set up the governing equation of motion of these rigid bodies. Thirdly, the linearized cone complementarity formulation has been deduced in detail to further address the frictional contact problems. By using the fixed point iteration algorithm, the contact forces between bodies can be calculated accurately. These inherent characteristics of the proposed methodology can ensure the conservation of the energy, linear momentum, and angular momentum of the system, which have been validated through a set of numerical experiments. Some challenging problems have been simulated to further demonstrate the advantages and application prospects of the numerical method. In the near future, we intend to extend the linearized cone complementarity formation to general polyhedral block system and to develop the corresponding parallel iteration algorithm.

CRediT authorship contribution statement

Huo Fan: Methodology, Software, Formal analysis, Writing - original draft. **Duruo Huang:** Conceptualization, Project administration, Writing - original draft, Writing - review & editing. **Gang Wang:** Methodology, Supervision, Funding acquisition, Writing - original draft, Writing - review & editing. **Feng Jin:** Methodology, Writing - review & editing.

Acknowledgements

The study is supported by NSFC/RGC Joint Research Scheme 51861165102 from National Natural Science Foundation of China, N_HKUST621/18 and General Research Fund 16204618 from Hong Kong Research Grants Council.

Appendix A

A1. The closest point between a point and a cylindrical surface

In Fig. A1, points C_1 and C_2 are the centroids of the bottom and upper surfaces of a cylinder, respectively. Namely, line C_1C_2 is the center axis. To find the closest point between point P and the cylinder surface, the following geometric algorithm can be employed:

(1) Find the projection A onto line C_1C_2.
(2) Determine the closest point Q between point P and the cylinder surface by

$$\boldsymbol{x}_Q = \boldsymbol{x}_A + \frac{R(\boldsymbol{x}_P - \boldsymbol{x}_A)}{|AP|} \tag{A1}$$

where $\boldsymbol{x}_A = (x_A, y_A, z_A)$, $\boldsymbol{x}_P = (x_P, y_P, z_P)$, and $\boldsymbol{x}_Q = (x_Q, y_Q, z_Q)$ are the coordinates of points A, P, and Q, respectively. R is the radius of the bottom surface of the cylinder. And $|AP|$ is the distance between points A and P.

A2. The closest point between a point and a truncated cone surface

Line C_1C_2 is the center axis of the truncated cone. In other words, points C_1 and C_2 are the centroids of the bottom and upper surfaces of a truncated cone, respectively, as exhibited in Fig. A2. We give the following geometric algorithm to compute the closest point between point P and the truncated cone surface:

(1) Find the projections A and B onto the bottom and upper surfaces of the truncated cone, respectively.
(2) Determine points D and E on the bottom and upper circles of the truncated cone by using

$$\boldsymbol{x}_D = \boldsymbol{x}_{C1} + \frac{R(\boldsymbol{x}_A - \boldsymbol{x}_{C_1})}{|C_1A|}, \quad \boldsymbol{x}_E = \boldsymbol{x}_{C2} + \frac{r(\boldsymbol{x}_B - \boldsymbol{x}_{C_1})}{|C_2B|} \tag{A2}$$

where $\boldsymbol{x}_A = (x_A, y_A, z_A)$, $\boldsymbol{x}_B = (x_B, y_B, z_B)$, $\boldsymbol{x}_{C1} = (x_{C1}, y_{C1}, z_{C1})$, $\boldsymbol{x}_{C2} = (x_{C2}, y_{C2}, z_{C2})$, $\boldsymbol{x}_D = (x_D, y_D, z_D)$, and $\boldsymbol{x}_E = (x_E, y_E, z_E)$ are the coordinates of points A, B, C_1, C_2, D, and E, respectively. Moreover, R and r are the radius of the bottom and upper surfaces of the truncated cone. $|C_1A|$ is the distance between points C_1 and A. Whereas, $|C_2B|$ is the distance between points C_2 and B.

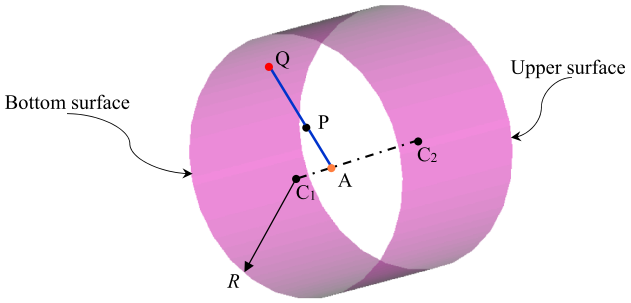


Fig. A1. The closest point from point to cylinder surface.

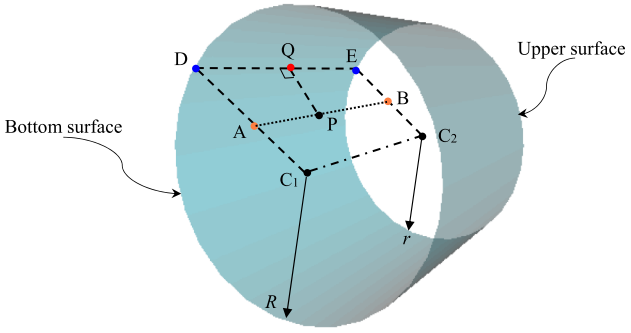


Fig. A2. The closest point from point to truncated cone surface.

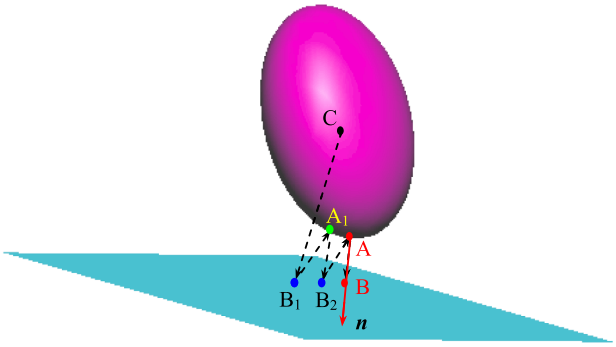


Fig. A3. Projection iteration for the closest point between an ellipsoid and a plane.

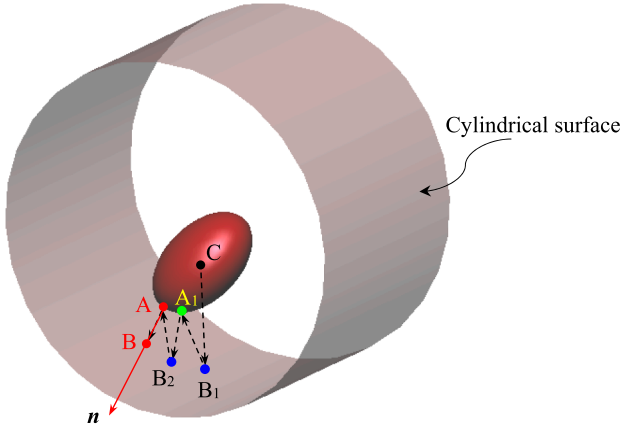


Fig. A4. Projection iteration for the closest point between ellipsoid and cylinder surface.

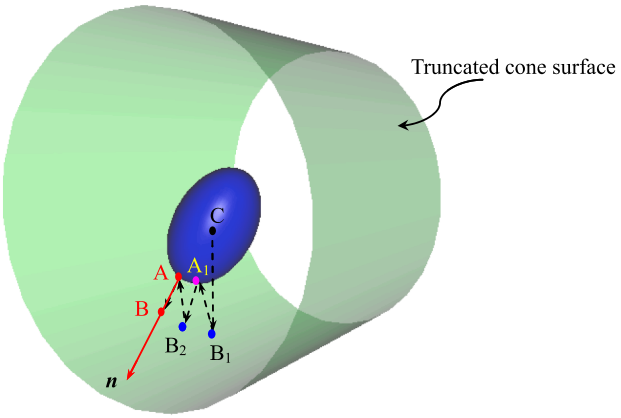


Fig. A5. Projection iteration for the closest point between an ellipsoid and a truncated cone surface.

(3) Determine projected point Q of the point P onto line DE. The point Q is the closest point to the truncated cone.

A3. The closest point between an ellipsoid and a plane/cylindrical surface/truncated cone surface

For the closest point between an ellipsoid and a plane/cylindrical surface/truncated cone surface as shown in Figs. A3–A5, the following projection iteration algorithm can be used.

(1) Find the projection B_1 of the centroid C of the ellipsoid onto the plane/cylindrical surface/truncated cone surface (see Sections A1 and A2 for more details).
(2) Compute the closest point A_1 between point B_1 and the ellipsoid.
(3) Calculate the projection B_2 of point A_1 onto the plane/cylindrical surface/truncated cone surface.
(4) Repeat steps (2)–(3) until $|A_{n-1}A_n|/|A_nB_{n+1}| \le 10^{-4}$ or the angler between vector $\vec{AB}$ and the outwards normal vector of the ellipsoid at pint A is about equal to 10^{-4}.

References

[1] Shi GH. Discontinuous deformation analysis: a new numerical model for the statics and dynamics of block system. Berkeley, California: University of California Berkeley; 1988.
[2] Cheng YM, Zhang YH. Rigid body rotation and block internal discretization in DDA analysis. Int J Numer Anal Meth Geomech 2000;24(6):567–78.
[3] Wu JH, Ohnishin Y, Nishiyama S. A development of the discontinuous deformation analysis for rock fall analysis. Int J Numer Anal Meth Geomech 2005;29(10):971–88.
[4] Fan H, Zheng H, Zhao JD. Discontinuous deformation analysis based on strain-rotation decomposition. Int J Rock Mech Min Sci 2017;92:19–29.
[5] Fan H, Zheng H, Zhao JD. Three-dimensional discontinuous deformation analysis based on strain-rotation decomposition. Comput Geotech 2018;95:191–210.
[6] Shyu K. Nodal-based discontinuous deformation analysis. PhD thesis. Univ Calif Berkeley, Berkeley, California; 1993.
[7] Zhang YH, Cheng YM. Coupling of FEM and DDA methods. Int J Geomech 2002;2(4):503–17.
[8] Miki S, Sasaki T, Koyama T, Nishiyama S, Ohnishi Y. Development of coupled discontinuous deformation analysis and numerical manifold method (NMM-DDA). Int J Comput Methods 2010;7(1):1–20.
[9] Zhao ZY, Gu J. Stress recovery procedure for discontinuous deformation analysis. Adv Eng Softw 2009;40(1):52–7.
[10] Fu GY, Ma GW, Qu XL. Boundary element based discontinuous deformation analysis. Int J Numer Anal Meth Geomech 2017;41(7):994–1015.
[11] Bao HR, Zhao ZY. The vertex-to-vertex contact analysis in the two-dimensional discontinuous deformation analysis. Adv Eng Softw 2012;45(1):1–10.
[12] Fan H, He SM. An angle-based method dealing with vertex-vertex contact in the two-dimensional discontinuous deformation analysis (DDA). Rock Mech Rock Eng 2015;48(5):2031–43.
[13] Shi GH. Contact theory. Sci China Tech Sci 2015;58(8):1–47.
[14] Fan H, Zheng H, Wang JF. A generalized contact potential and its application in discontinuous deformation analysis. Comput Geotech 2018;99:104–14.
[15] Wu JH, Lin HM. Improvement of open-close iteration in DDA. Frontiers of discontinuous numerical methods and practical simulations in engineering and disaster prevention. London: McGraw-Hill; 2013. p. 185–91.
[16] Lin CT, Amadei B, Jung J, Dwyer J. Extensions of discontinuous deformation analysis for jointed rock masses. Int J Rock Mech Min Sci 1996;33(7):671–94.
[17] Cai YE, Liang GP, Shi GH, Cook NGW. Studying an impact problem by using LDDA method. In: Proceedings of the first international forum on discontinuous deformation analysis (DDA) and simulations of discontinuous media. Berkeley, CA. TSI Press: Albuquerque; 1996. p. 288–95.
[18] Koyama T, Nishiyama S, Yang M, Ohnishi Y. Modeling the interaction between fluid flow and particle movement with discontinuous deformation analysis (DDA) method. Int J Numer Anal Meth Geomech 2011;35(1):1–20.
[19] Morgan WE, Aral MM. An implicitly coupled hydro-geomechanical model for hydraulic fracture simulation with the discontinuous deformation analysis. Int J Rock Mech Min Sci 2015;73:82–94.
[20] Ke TC, Bray J. Modeling of particulate media using discontinuous deformation analysis. J Eng Mech 1995;121(11):1234–43.
[21] Beyabanaki SAR, Bagtzoglou AC. Non-rigid disk-based DDA with a new contact model. Comput Geotech 2013;49:25–35.
[22] Fan H, Wang JF, Zheng H. Variational inequality-based particle discontinuous deformation analysis. Int J Numer Anal Meth Geomech 2019;43(11):1995–2019.
[23] Beyabanaki SAR, Bagtzoglou AC. Three-dimensional discontinuous deformation analysis (3D DDA) method for particulate media applications. Geomech Geoeng 2012;7(4):239–53.
[24] Fan H, Wang JF. Dynamic modeling of sphere, cylinder, done, and their assembly. Arch Comput Methods Eng 2019:1–48.
[25] Fan H, Huang D, Wang G, et al. Discontinuous deformation analysis for SH-body. Comput Geotech 2020;117:103234.
[26] Zheng H, Jiang W. Discontinuous deformation analysis based on complementary theory. Sci China Tech Sci 2009;29:2547–54.
[27] Zheng H, Zhang P, Du XL. Dual form of discontinuous deformation analysis. Comput Methods Appl Mech Eng 2016;305:196–216.
[28] Fan H, Zhao JD, Zheng H. Variational inequality-based framework of discontinuous deformation analysis. Int J Numer Meth Eng 2018;115(3):358–94.
[29] Meng JJ, Cao P, Huang JS, et al. Three-dimensional spherical discontinuous deformation analysis using second-order cone programming. Comput Geotech 2019;112:319–28.
[30] Tasora A, Anitescu M. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. Comput Methods Appl Mech Eng 2011;200(5–8):439–53.
[31] Wei J, Huang D, Wang G. Micro-scale descriptors for particle-void distribution and jamming transition in pre- and post-liquefaction of granular soils. J Eng Mech 2018;144(8):04018067.
[32] Wang G, Wei J. Microstructure evolution of granular soils in cyclic mobility and post-liquefaction process. Granular Matt 2016;18(3):51.
[33] Wei J, Wang G. Discrete-element method analysis of initial fabric effects on pre-and post-liquefaction behavior of sands. Géotechnique Lett 2017;7(2):161–6.
[34] Perram JW, Wertheim MS. Statistical mechanics of hard ellipsoids. I. Overlap

algorithm and the contact function. J Comput Phys 1985;58(3):409–16.
[35] Cundall PA. 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. Int J Rock Mech Min Sci Geomech Abstr 1988;25:107–16.
[36] Ting JM. A robust algorithm for ellipse-based discrete element modelling of granular material. Comput Geotech 1992;13:175–86.
[37] Lin XS, Ng TT. Contact detection algorithms for three-dimensional ellipsoids in discrete element modelling. Int J Numer Anal Meth Geomech 1995;19(9):653–9.
[38] Hiriart-Urruty JB, Lemarechal C. Convex analysis and minimization algorithms. Berlin: Springer Verlag; 1993.
[39] Beatini V, Royer-Carfagni G, Tasora A. A regularized non-smooth contact dynamics approach for architectural masonry structures. Comput Struct 2017;187:88–100.
[40] De Saxc G, Feng ZQ. Recent advances in contact mechanics the bipotential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. Math Comput Modell 1998;28(4):225–45.
[41] Jiang W, Zheng H, Sun GH, et al. 3D DDA based on variational inequality theory and its solution scheme. Int J Comput Methods 2018;15(08):1850081.
[42] Dong L, Sun D, Li X, et al. Interval non-probabilistic reliability of surrounding jointed rockmass considering microseismic loads in mining tunnels. Tunn Undergr Sp Tech 2018;81:326–35.