Title

Generalized contact model for polyhedra in three-dimensional discontinuous deformation analysis

Permalink
https://escholarship.org/uc/item/3pc3n3g9

Journal
International Journal for Numerical and Analytical Methods in Geomechanics, 42(13)

ISSN
0363-9061

Authors
Zheng, Fei
Jiao, Yu-Yong
Sitar, Nicholas

Publication Date
2018-09-01

DOI
10.1002/nag.2798

Peer reviewed

Generalized contact model for polyhedra in three-dimensional discontinuous deformation analysis

Fei Zheng[1] | Yu-Yong Jiao[2] | Nicholas Sitar[3]

1 Research Associate, Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China 2 Professor, Faculty of Engineering, China University of Geosciences, Wuhan 430074, China 3 Edward G. and John R. Cahill Professor, Department of Civil and Environmental Engineering, UC Berkeley, Berkeley CA 94720-1710, USA

Correspondence Yu-Yong Jiao, Faculty of Engineering, China University of Geosciences, Wuhan 430074, China. Email: yyjiao@cug.edu.cn

Summary

We present a generalized contact computation model for arbitrarily shaped polyhedra to simplify the contact analysis in discontinuous deformation analysis. A list of generalized contact constraints can be established for contacting polyhedra during contact detection. Each contact constraint contains information for 2 contact points, unique contact plane, and related contact modes (open, locked, or sliding). Computational aspects of the generalized contact model include identification of contact positions and contact modes, uniform penalty formulation of generalized contact constraint, and uniform updating of contact modes and contact planes in the open-close iteration. Compared with previous strategies, the generalized contact computation model has a simpler data structure and fewer memory requirements. Meanwhile, it simplifies the penalty formulation and facilitates the open-close iteration check while producing enough accuracy. Illustrative examples show the ability of the method to handle the full range of polyhedral shapes.

1 INTRODUCTION

Correctly modeling contact interaction of discrete bodies is the key issue in discontinuous computation methods.[1-5] Focusing on polyhedral bodies, resolving the contact interaction is difficult as the non-smooth change of polyhedron face normal. In DEM and FDEM modeling, the common plane model,[2] the energy-conserving contact interaction model,[6] the triangulated rounded bodies model,[7] the polygon-based description,[8] the potential particle model,[9, 10] dilated polyhedra[11, 12] and the potential function model[13, 14] have been applied to treat contact of polyhedra. These contact models are mostly for granular materials such as ballast and soils, and the contact interaction law is based on the overlap of 2 contact bodies. However, for the 3-dimensional (3-D) discontinuous deformation analysis (DDA) method,[3, 4] a strict “no penetration” requirement leads to very small overlap during contact interaction. The resolution algorithms for both convex and concave polyhedral blocks are mostly depending on the classification of the contact types.[15-17] The final contact constraint information can be obtained according to the first entrance rule or shortest exist rule.[5] A 4-type identification phase[15-17] and the separate treating of the vertex-to-face model[18, 19] and edge-to-edge models[20-22] are used, which may complicate the data structures and contact analysis procedure of the program. A more general contact model for polyhedral blocks that strictly fits the contact constraint requirement and the implicit solution approach is necessary. We address 3 issues for successful contact analysis in DDA, namely: (1) detection of contact constraints on all potential contacting blocks; (2) penalty formulation of contact force-displacement function in equilibrium equations; and (3) the open-close iteration (OCI) to revise the penetration/no penetration constraints.

Finding all contact constraints of 2 polyhedral blocks is the basic issue. The distance criterion and the entrance criterion[15, 23] are usually applied to locate the contact position and contact normal. Some commonly used contact detection algorithms are for convex polyhedral blocks, such as the common plane method and its extensions,[2, 24, 25] the linear programming method,[26] the potential particle method,[9, 10] the approaching face method[27] and the fast direct search method.[23] Typical algorithms to detect arbitrarily shaped polyhedra include the direct search algorithm,[15] the multi-shell cover contact detection algorithm[16] and the angle-based contact detection algorithm.[17] For most algorithm in 3-D DDA, 4 basic types will be recorded and the formulation is based on vertex-to-face model and edge-to-edge model. To go further, we found that a simple contact model that generalize the detection and formulation of all basic contact types for polyhedra can be established, as an extension of the general point-to-plane contact model for convex polyhedron.[28]

When formulating the contact constraints, attention has to be given to the choice of the correct contact points and the contact plane. Generally, there are 7 different types of contact between 2 polyhedra: vertex-to-vertex (v-v), vertex-to-edge (v-e), vertex-to-face (v-f), cross edge-to-edge (cross e-e), parallel edge-to-edge (parallel e-e), edge-to-face (e-f), and face-to-face (f-f). All these contact types can be grouped into 3 categories: (1) contacts that have a unique contact plane, but with infinite number of contact point pairs, ie, parallel e-e, e-f, and f-f contacts, as shown in Figure 1; (2) contacts consisting of a pair of points, but non-unique contact plane, ie, pure v-v and pure v-e contacts, as shown in Figure 2C,D; and (3) contacts types that have a defined contact point pair and a definite contact plane, ie, v-f, cross e-e and v-v and v-e that belong to e-f or f-f, as shown in Figure 2A,B. In our approach, all these contact types are generalized into a single contact computation model. Specifically, a list of contact constraints can be established for 2 polyhedral blocks. Each constraint then provides the complete information for the formulation of the equilibrium equation: coordinates of the 2 contact points, the contact plane (normal direction and tangential direction), and the contact mode (open, locked, or sliding). The concept of “contact mode” is used to describe the penetration/no penetration constraint. Open mode represents no penetration, while locked and sliding modes represent penetration. Methods to determine the initial contact mode for contact analysis of 3-D DDA were rarely discussed. Based on the complete constraint model, strategies to determine contact mode and related formulation in the step-wise discontinuous computation are proposed.

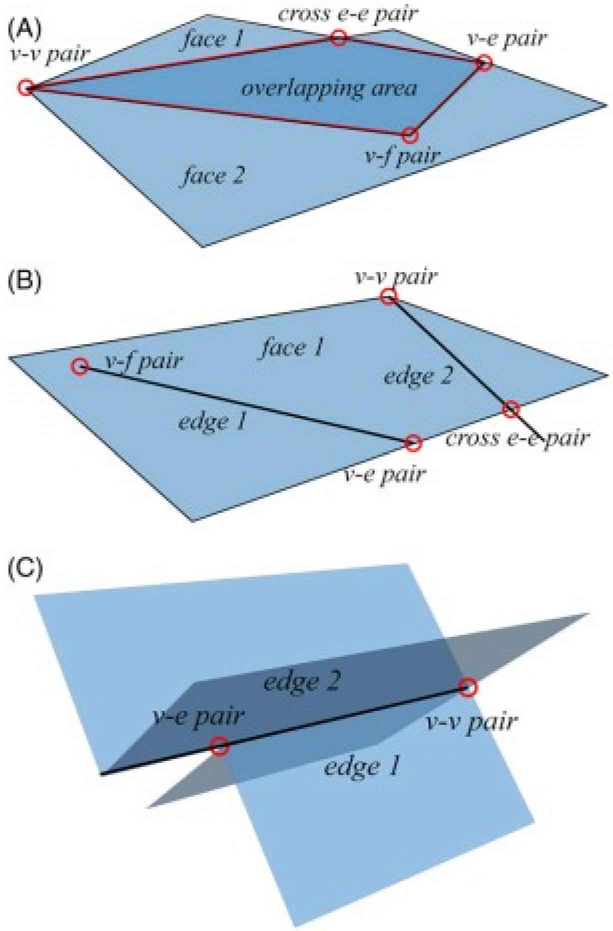


Figure 1

Contact types that include infinite contact point pairs: A, face-to-face type; B, edge-to-face type; C, parallel edge-to-edge type

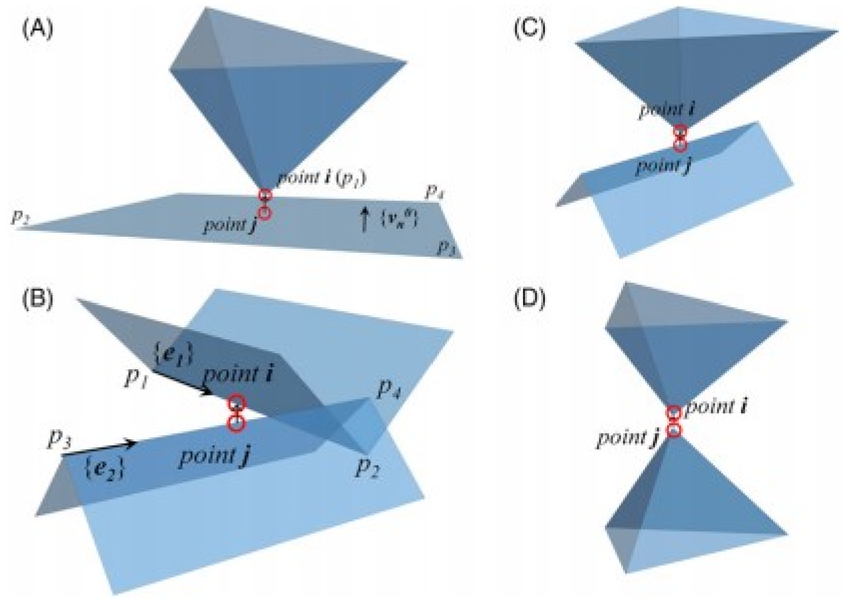


Figure 2

Four basic contact types: A, vertex-to-face type; B, crossing edge-to-edge type; C, vertex-to-edge type; D, vertex-to-vertex type

Correct determination of the penetration/no penetration contact mode is very important in solving contact of a discrete block system. The OCI is applied in DDA to locate all closed entrances at the end of each computation step. In details, 2 requirements should be satisfied: (1) the penalty constraint should be applied for contact pair from open to close; (2) any penalty function should be removed for contact pair from close to open. The final contact mode in each time step is determined by the criteria in OCI, which is limited in tension and limited in penetration at each first entrance or shortest exit position,[5] along with the friction law to separate the locked and sliding modes. Based on the generalized contact model, the OCI process can be implemented more conveniently.

To this end, we first present the general contact computation model with particular attention on the choice of contact points, contact planes, and contact modes. Then, the general contact constraint model and its formulation are shown and compared with other formulations. Next, the execution of OCI is discussed in detail, potential issues that may affect the accuracy of the contact analysis are discussed, and related control methods are presented. Finally, examples are provided to illustrate the effectiveness of the generalized contact analysis model.

2 GENERAL CONTACT CONSTRAINT MODEL

Contact computation of DDA is achieved by applying contact constraints, ie, no penetration in normal direction, Coulomb friction law in shear direction, on all contact positions. According to Shi,[5] contact of 2 polyhedral can be represented by a list of contact covers which define unique contact positions and contact planes. Each contact cover corresponds to a contact type and provides geometrical information for contact constraints to formulate the contact equations. Then, the first entrance rule is used to locate the contact position when 2 blocks contact from open to closed. Meanwhile, the shortest exit rule is used to locate the contact position when 2 blocks already contact. The choices of contact points and contact planes for all contact types between 2 polyhedra are discussed here based on the 3 categories already introduced. For type (a) contacts with infinite contact points, the approach is to either integrate over the area or to use a set of finite points that represent the contact area. Most DDA algorithms use the latter scheme, in which the boundary vertices of the overlapping polygon area are chosen as contact positions to apply the contact constraints. As shown in Figure 1, the parallel edge-to-edge, edge-to-face, and face-to-face contact types are thus decomposed into combinations of the previous 4 types (v-v, v-e, v-f, and cross e-e).[29, 30] For type (b) contacts with indeterminate contact plane, inscribed sphere scheme,[30] potential particle scheme,[9] common plane,[2] or shortest exit scheme[3]can be used to choose the appropriate contact plane. The contact plane may change suddenly for polyhedron when the contact type changes from v-v or v-e to v-f or e-e. The inscribed sphere scheme[30] and potential particle scheme[9] allow a smooth change of contact plane for all contact types of convex polyhedra including type (b), while the accuracy is controlled by the size of the inscribed sphere or the parameters that determine the curvature of the particle faces. The first entrance approach and shortest exit approach were proposed by Shi[5] to define the contact position. The shortest exit is extended to find the contact plane for both v-v and v-e pair that include either convex or concave angles (or edges). For type (c), both the contact points and contact planes are unique.

Finally, all first entrance positions can be represented by 4 contact types (covers): v-v, v-e, v-f, and e-e.[9-14] To simplify and generalize the representation of contact constraints and the formulation of contact terms in equilibrium equations, we represent all contact types by 2 contact points p_i^0 (x_i^0,y_i^0,z_i^0) and p_j^0 (x_j^0,y_j^0,p_j^0) in block *i* and block *j*, respectively, a unique plane with unit normal vector , and its contact mode: open, locked, or sliding, as shown in Figure 3A.

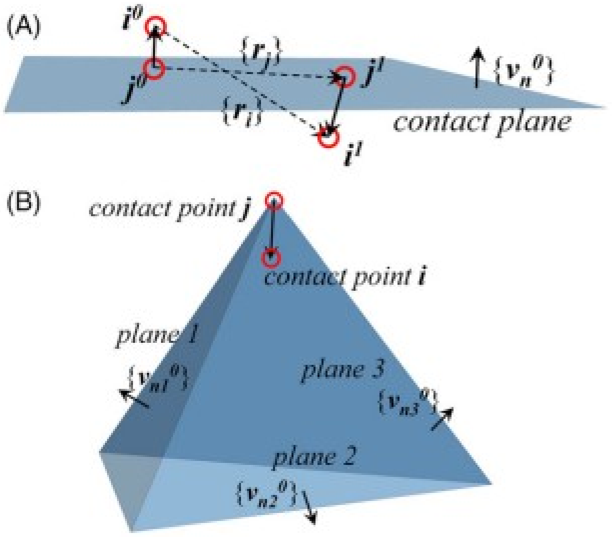


Figure 3

Generalized contact models: A, the model with 1 contact plane; B, the model with multiple potential contact planes

2.1 Contact points

Contact points, ie, first entrance position, are identified to apply the normal constraint, the tangent constraint, or frictional force. The auxiliary-simplex method and vector analysis have been used to derive the formulation of DDA for the vertex-to-face model[18, 19] and the edge-to-edge model.[20-22] For a constraint in the normal direction, the vertex and 3 points on the face are used to derive the normal constraint matrix for the vertex-to-face model; 2 points on each edge are used to derive the normal constraint for the edge-to-edge model. For constraint in shear direction (shear spring or frictional force), 2 closest points on a vertex-to-face pair or an edge-to-edge pair are used to derive the formulation. Beyabanaki et al[28] proposed a point-to-face contact model to derive contact matrix for all contact types, but their solution was limited to convex polyhedra. Contact models using the common plane method simplify the contact of 2 convex polyhedra to contact of convex polyhedra and contact plane. The reference point on the common-plane defines the line of action of the contact force and is taken as the contact location.[31]

In our proposed contact model, each polyhedra pair has a list of constraints. Each constraint is defined by one of the v-v, v-e, v-f, and crossing e-e contact types. For each contact type, the closest points are chosen as the contact points if blocks do not penetrate and 2 points through which the shortest exit is obtained to separate the penetrated blocks are chosen as the contact points if the blocks penetrate. Figure 2 shows the closest pair of points for each contact type.

For vertex-face pairs, whether they penetrate or not, the contact points can be obtained by projecting the vertex on the face. One contact point is the vertex itself, the other is its projection point on the face. From Figure 2A, the projection point $\{p_j^0\}$ of the vertex $\{p_i^0\}$ on the contact face can be computed by

$$\left\{p_j^0\right\} = \{p_i^0\} - (\overrightarrow{p_2p_1}) \cdot \{v_n^0\}. \quad (1)$$

For cross edge-edge pair, whether they penetrate or not, the contact points can be obtained by calculating the closest point pair that defines a line perpendicular to both edges. From Figure 2B, the 2 contact points on the contacting edges can be computed by

$$\left\{p_i^0\right\} = \{p_1\} + t_1 \cdot \overrightarrow{p_1p_2}$$

$$\left\{p_j^0\right\} = \{p_3\} + t_2 \cdot \overrightarrow{p_3p_4} \quad (2)$$

where t_1 and t_2 can be computed by

$$t_1 = \frac{(\overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_3}) \cdot (\overrightarrow{p_3p_4} \cdot \overrightarrow{p_3p_4}) - (\overrightarrow{p_3p_4} \cdot \overrightarrow{p_1p_3}) \cdot (\overrightarrow{p_1p_2} \cdot \overrightarrow{p_3p_4})}{(\overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_2}) \cdot (\overrightarrow{p_3p_4} \cdot \overrightarrow{p_3p_4}) - (\overrightarrow{p_3p_4} \cdot \overrightarrow{p_1p_2}) \cdot (\overrightarrow{p_1p_2} \cdot \overrightarrow{p_3p_4})}$$

$$t_2 = \frac{(\overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_3}) \cdot (\overrightarrow{p_3p_4} \cdot \overrightarrow{p_1p_2}) - (\overrightarrow{p_3p_4} \cdot \overrightarrow{p_1p_3}) \cdot (\overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_2})}{(\overrightarrow{p_3p_4} \cdot \overrightarrow{p_3p_4}) \cdot (\overrightarrow{p_1p_2} \cdot \overrightarrow{p_1p_2}) - (\overrightarrow{p_1p_2} \cdot \overrightarrow{p_3p_4}) \cdot (\overrightarrow{p_3p_4} \cdot \overrightarrow{p_1p_2})}. \quad (3)$$

For a vertex-vertex pair in Figure 2D, the contact point pair may be: (a) the 2 vertices; (b) 1 vertex and its projection point on a face neighboring the other vertex; or (c) the closest point on 2 edges that neighboring the 2 vertices, respectively. In contact detection process, a vertex-to-vertex pair is usually detected when their distance is smaller than a tolerance. To avoid complicated judging and computing of 2 closest points for the vertex-to-vertex pair, the 2 vertices themselves are used as contact points.

For vertex-edge pair in Figure 2C, the contact point pair can be: (a) the vertex and its projection point on the edge; (b) the closest points on 2 edges neighboring the vertex and the edge, respectively; and (c) the vertex and its projection point on either face neighboring the edge. In contact detection process, a small distance tolerance is usually set in locating vertex-edge pair and method (a) is chosen to calculate the contact points.

During changes of contact modes, from open to close, from close to open, or from close to close, contact points change in a single computing step. In this sense, the representation of contact points is an explicit form that uses the configuration of contact points at the beginning of the time step. Revised approximation of contact point coordinates within the computational step can be obtained using vector analysis during the contact process, as is discussed later.

2.2 Contact planes

Contact planes are defined and distinct for the following contact types: (1) v-f, (2) cross e-e, and (3) v-v and v-e that belong to e-f or f-f. However, for pure v-v, pure v-e contact a unique contact plane does not exist, as already mentioned. For 2 convex polyhedra, the common plane method uses a bisection plane to be the contact plane through which 2 nearest vertices have the smallest gap distance.[2] To use the common plane algorithm, penetrating polyhedra should be separated along a certain path. Cundall[2] proposed an iterative method to find the contact plane, while Nezami et al find it using the fast common plane identification process[24] or shortest link searching procedure.[25] Beyabanaki et al[28] use a mid-plane for the v-v or v-e pairs as the contact plane, the normal of which is computed by taking an average of faces neighboring the vertex.

Normals of the contact plane may change abruptly when the contact type changes from v-v or v-e to v-f or cross e-e and vice versa. In this case, the block penetration should be carefully controlled. The inscribed sphere scheme[30] and the potential particle scheme[9] can be used to obtain a smooth change of contact normal for convex blocks. The first entrance approach and shortest exit approach proposed by Shi[5] are adopted here to deal with both convex and concave blocks.

2.2.1 Shortest exit strategy

For convex v-v and v-e contact pair, the shortest exit scheme is used as follows: (1) Locate all potential contact planes from vertex-face pairs and edge-edge pairs that neighbor the contacting vertex or contacting edge using no-overlap check[15, 17]; (2) Choose the contact plane with shortest normal penetration for the contact points. For each contact plane, the separation distance is computed as Equation 4.

$$d_{ni}^{0} = \left(\left\{p_i^0\right\} - \left\{p_j^0\right\}\right) \cdot \left\{v_{ni}^0\right\} \quad (4)$$

Then, the contact plane with the largest separation distance value is chosen. Through this strategy, physically implausible large normal penetration can be avoided when contact plane abruptly changes for v-v and v-e pairs or contact type switches between v-v, v-e, and v-f, cross e-e. As shown in Figure 3B, 3 potential contact planes are already established, and a unique plane with the shortest normal penetration distance will be chosen as the contact plane.

The contact plane algorithm then consists of 2 steps: (1) initial coordinates of the contact points are used in determining the contact plane for the first cycle; and (2) updated coordinates of the contact points are used for subsequent cycles.

2.2.2 Concave angles and concave edges

The scheme for choosing a contact plane using the shortest exit approach has to be modified for concave vertex and concave edges. Zhang et al[15] and Zheng et al[17] mention using multi-contact plane scheme for contacts that include concave angles and concave edges.

Here, we propose a decomposition strategy for managing concave objects. To prevent penetration of concave objects, multiple contact planes are established based on the decomposition of the concave object. The strategy is as follows: (1) Obtain all potential contact planes from angle-to-face pair and convex edge to convex edge pair by checking for physical overlap while making sure that all potential contact planes point from block j to block i; and (2) Check the convexity of the angle bounded by all potential contact planes. If it is a convex angle, use the single contact plane scheme; otherwise, decompose this angle into the union of several convex angles, and choose 1 contact plane for each sub-convex angle.

An example of determining contact plane for vertex-to-edge contact that includes a concave edge is shown in Figure 4. By checking for overlap, 2 potential contact planes forming a concave angle are found. This concave angle is then decomposed into 2 half-plane angles, and both plane 1 and plane 2 are regarded as contact planes.

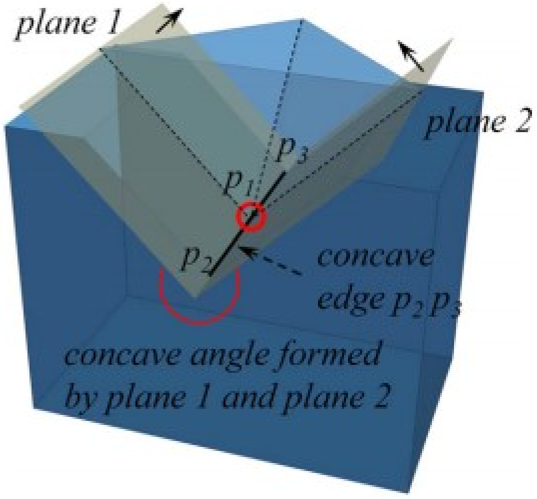


Figure 4

A vertex to concave edge model with multiple contact planes

Another example of determining the contact plane for vertex-to-vertex contact including concave angle is shown in Figure 5. By checking for overlap for v-f and e-e pair, 4 contact planes forming a concave angle are found. This angle can be decomposed into 2 convex angles, sub-angle 1 (plane 1 and plane 2) and sub-angle 2 (plane 3 and plane 4). During computation, 1 contact plane is chosen for each sub-angle, according to the shortest exit scheme.

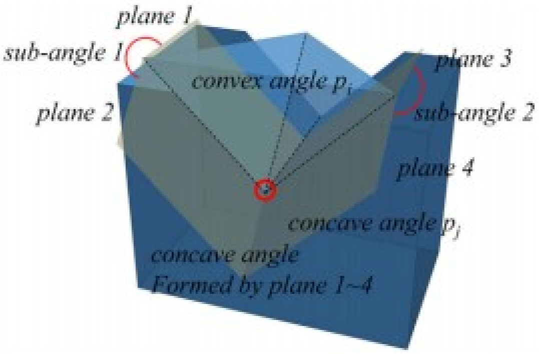


Figure 5

A vertex to concave vertex model with multiple contact planes

2.3 Tangent direction

When sliding between 2 contact points occurs on the contact plane, its direction is unknown prior to the computation of current step. As the formulation of the frictional force term needs an explicit representation of frictional force vector, the relative sliding direction needs be assumed. An accurate approximation of the tangent sliding direction is important to guarantee correct friction force computation. Jiang and Yeung[19] proposed an iterative procedure for determining the relative sliding direction along with OCI, while Huang et al.[32]presented an algorithm to compute the direction of frictional force for contacting spheres.

We provide 3 ways to calculate the tangent direction: (1) using the initial velocity of contact points; (2) using the displacement of contact points in the previous time step; and (3) using the displacement of contact points in the previous iteration of the OCI process.

For computation of the first cycle in a time step, (1) the initial velocity of current time step or (2) the relative displacement in previous time step can be used to compute the initial tangent direction.

The initial velocity of the 2 contact points can be computed as follows:

$$\{v_i^0\} = [T_i(x_i, y_i, z_i)]\{v_{oi}\} \quad (5)$$

where $T_i(x_i, y_i, z_i)$ represents transfer matrix defined in Equation 11, and $\{v_{oi}\}$ represents initial velocity term (translation, rotation, strain rate) of the block centroid. Assuming and are initial velocities of the 2 contact points p_i and p_j, respectively, and is the unit normal vector of the contact plane, the relative sliding direction can be computed by

$$\{v_t^0\} = \{v_i^0\} - \left\{v_j^0\right\} - \left(\{v_i^0\} - \left\{v_j^0\right\}\right) \cdot \{v_n^0\}. \quad (6)$$

Replacing $\{v_i^0\}$ and $\{v_j^0\}$ by the displacement $\{r_i^0\}$ and $\{r_j^0\}$ of point *i* and *j* in the previous time step, the tangent direction can be obtained using the displacement in the previous step.

$$\{v_t^0\} = \{r_i^o\} - \{r_j^o\} - \left(\{r_i^o\} - \{r_j^o\}\right) \cdot \{v_n^0\} \quad (7)$$

Equation 7 will be used only if the output tangent sliding vector in Equation 6 is smaller than a tolerance. For example, for static analysis that the initial velocity is set to zero, Equation 7is used.

For computation of later cycles in the time step, the tangent direction can be updated in OCI judgment. Using the displacement $\{r_i^*\}$ and $\{r_j^*\}$ of point *i* and pint *j* in the previous iteration, the tangent direction can be updated by

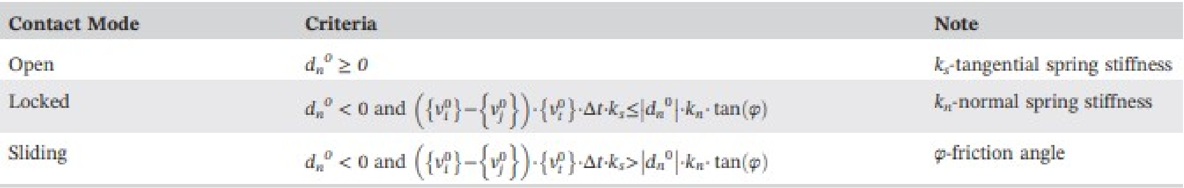
$$\{v_t^*\} = \{r_i^*\} - \{r_j^*\} - \left(\{r_i^*\} - \{r_j^*\}\right) \cdot \{v_n^0\}. \quad (8)$$

2.4 Contact mode

Contact modes refer to 1 of the 3 modes: open, locked, and sliding, for each contact pair. At each time step, the contact modes must converge to a single mode. In general, there are 3 options to determine the contact modes: (1) In the initial approximation by using the normal penetration distance d_n^0 and relative tangential velocity of the contact pair; (2) By inheriting the closed contact modes from the same contact pair in the previous time step; and (3) By testing against the criteria in the OCI.

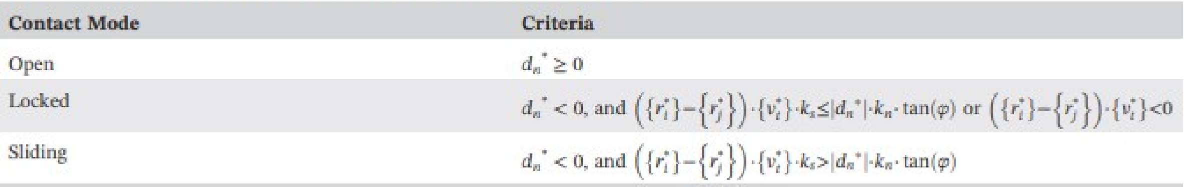
If the contact pair is first detected in the current step, method (1) is applied to make an initial assumption, using initial normal penetration distance d_n^0, the initial velocity $\{v_i^0\}$ and $\{v_j^0\}$ of point *i* and point *j*, initial tangential direction vector $\{v_t^0\}$, and time step size Δ*t*. Table 1 shows the criteria for initial assumption of contact mode.

TABLE 1 The criteria for initial assumption of contact mode



Contact Mode	Criteria	Note
Open	$d_n^o \geq 0$	k_s-tangential spring stiffness
Locked	$d_n^o < 0$ and $\left(\{v_i^0\}-\{v_j^0\}\right)\cdot\{v_t^0\}\cdot\Delta t\cdot k_s \leq \lvert d_n^o\rvert\cdot k_n\cdot\tan(\varphi)$	k_n-normal spring stiffness
Sliding	$d_n^o < 0$ and $\left(\{v_i^0\}-\{v_j^0\}\right)\cdot\{v_t^0\}\cdot\Delta t\cdot k_s > \lvert d_n^o\rvert\cdot k_n\cdot\tan(\varphi)$	φ-friction angle

TABLE 2 The criteria for updating contact mode in open-close iteration



Contact Mode	Criteria
Open	$d_n^* \geq 0$
Locked	$d_n^* < 0$, and $\left(\{r_i^*\}-\{r_j^*\}\right)\cdot\{v_t^*\}\cdot k_s \leq \lvert d_n^*\rvert\cdot k_n\cdot\tan(\varphi)$ or $\left(\{r_i^*\}-\{r_j^*\}\right)\cdot\{v_t^*\}<0$
Sliding	$d_n^* < 0$, and $\left(\{r_i^*\}-\{r_j^*\}\right)\cdot\{v_t^*\}\cdot k_s > \lvert d_n^*\rvert\cdot k_n\cdot\tan(\varphi)$

If d_n^0 and the same closed contact pair already exist in the contact list from the previous time step, the same closed modes are used for that contact type in the current time step, ie, method (2).

In method (3), the contact mode is updated in the OCI process. The determination of the mode of contact is based on the computed displacement vector $\{r_i^*\}$ and $\{r_j^*\}$ of contact points i and j in the current cycle, where d_n^* and $\{v_t^*\}$ are the normal penetration distance and the tangential sliding direction vector obtained in previous OCI cycle. The criteria are shown in Table 2.

3 CONTACT FORMULATION

The main assumption during the formulation of contact submatrices and contact force vectors is that the unit normal vector of the contact plane remains constant during the time step. Based on contact modes, formulations of normal spring constraint, tangential spring constraint, and frictional force vectors are derived next, and this formulation is then compared with that obtained using the determinant form by Wu.[18]

3.1 Basic assumption

Adopting a first-order approximation of block displacement function in each step and assuming the blocks to be linear elastic hawse have 12 degrees of freedom, which can be represented as

$$\{d_i\} = \left\{u\ \ v\ \ w\ \ rx\ \ ry\ \ rz\ \ \varepsilon_x\ \ \varepsilon_y\ \ \varepsilon_z\ \ \gamma_{yz}\ \ \gamma_{zx}\ \ \gamma_{xy}\right\}^T \quad (9)$$

where u,v,w represent the translation of the block centroid along 3 axes, rx,ry,rz represent the rotation of the block along 3 Cartesian axes, and ε_x, ε_y, ε_z, γ_{yz}, γ_{zx}, γ_{xy} represent the normal and shear strain of the block.

The displacement of a point (x,y,z) in a block in 1 step can be represented by

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = [T]\{d_i\} \quad (10)$$

where

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 & z_t & -y_t & x_t & 0 & 0 & 0 & z_t/2 & y_t/2 \\ 0 & 1 & 0 & -z_t & 0 & x_t & 0 & y_t & 0 & z_t/2 & 0 & x_t/2 \\ 0 & 0 & 1 & y_t & -x_t & 0 & 0 & 0 & z_t & y_t/2 & x_t/2 & 0 \end{bmatrix} \quad (11)$$

and

$$x_t = x - x_0$$

$$y_t = y - y_0$$

$$z_t = z - z_0 \quad (12)$$

x_0, y_0, z_0 represent the coordinate of block centroid.

An energy function can be established by considering the elastic deformation energy, inertia potential energy, contact energy, and potential energy of loadings and constraints. Then, by minimizing the total potential energy, the global equilibrium equation can be established in the following form

$$\begin{bmatrix} [k_{11}] & [k_{12}] & \cdots & [k_{1n}] \\ [k_{21}] & [k_{22}] & \cdots & [k_{2n}] \\ \vdots & \vdots & \ddots & \vdots \\ [k_{n1}] & [k_{n2}] & \cdots & [k_{nn}] \end{bmatrix} \begin{Bmatrix} \{D_1\} \\ \{D_2\} \\ \vdots \\ \{D_n\} \end{Bmatrix} = \begin{Bmatrix} \{F_1\} \\ \{F_2\} \\ \vdots \\ \{F_n\} \end{Bmatrix} \quad (13)$$

where $\{D_i\}$ and $\{F_i\}$ are 12 × 1 vectors that represent the displacement unknowns and generalized force vectors for block i, $[k_{ii}]$ is a 6 × 6 matrix that represents the contribution of elastic deformation, inertial force, and some other terms to block i, $[K_{ij}]$ is a 6 × 6 matrix represents the contact of block i and block j. Detailed formulation of submatrices can be found in Shi.[4]

3.2 Geometry representation of the contact model

The contact matrix is formulated based on the geometrical model of the contact constraint. Assume coordinates of the contact points i and j at the beginning of the current time step are $\{p_i^0\} = \{x_i^0,y_i^0,z_i^0\}^T$ and $\{p_j^0\} = \{x_j^0,y_j^0,p_j^0\}^T$, the unit normal vector of the contact plane and unit vector of tangential sliding direction at the beginning of the time step are $\{v_n^0\} = \{n_x^0,n_y^0,n_z^0\}^T$ and $\{v_t^0\} = \{t_x^0,t_y^0,t_z^0\}^T$, displacements of contact point i and j in current computation cycle are$\{r_i\} = \{r_{ix},r_{iy},r_{iz}\}^T$, $\{r_j\} = \{r_{jx},r_{jy},r_{jz}\}^T$, the geometrical relationship in contact process can be established.

Relative displacement of contact points p_i and p_j in 1 step is

$$\{\Delta r\} = \{r_i\} - \{r_j\} \quad (14)$$

where

$$\begin{gathered} \{r_i\} = [T_i(x_i,y_i,z_i)]\{d_i\} \\ \{r_j\} = \left[T_j\left(x_j,y_j,z_j\right)\right]\{d_j\}\,. \end{gathered} \quad (15)$$

The value of total relative displacement is

$$\Delta d_r = \sqrt{\{\Delta r\}^T \cdot \{\Delta r\}} = \sqrt{\left(\{r_i\} - \{r_j\}\right)^T \cdot \left(\{r_i\} - \{r_j\}\right)}\,. \quad (16)$$

Relative displacement in normal direction is

$$\Delta d_n = \{\Delta r\}^T \cdot \{v_n^0\} = \left(\{r_i\} - \{r_j\}\right)^T \cdot \{v_n^0\}\,. \quad (17)$$

Assume the tangential direction is known in advance, relative displacement in tangential direction is

$$\Delta d_t = \{\Delta r\}^T \cdot \{v_t^0\} = \left(\{r_i\} - \{r_j\}\right)^T \cdot \{v_t^0\}\,. \quad (18)$$

If the tangential direction is unknown in advance, the relative displacement in tangential direction can be represented using normal displacement and total displacement

$$\Delta d_t = \sqrt{\Delta d_r{}^2 - \Delta d_n{}^2} = \sqrt{\left(\{r_i\}-\{r_j\}\right)^T\cdot\left(\{r_i\}-\{r_j\}\right)-\left(\{r_i\}-\{r_j\}\right)^T\cdot\{v_n^0\}\cdot\{v_n^0\}^T\cdot\left(\{r_i\}-\{r_j\}\right)}$$

(19)

3.3 Formulation of normal constraint

The normal penetration distance at the beginning of the time step is

$$d_n^0 = \left(\{p_i^0\}-\left\{p_j^0\right\}\right)^T\cdot\{v_n^0\} = \left\{\begin{matrix} x_i-x_j \\ y_i-y_j \\ z_i-z_j \end{matrix}\right\}^T\cdot\{v_n^0\}. \quad (20)$$

The normal penetration distance at the end of the time step is

$$d_n^1 = d_n^0 + \Delta d_n = d_n^0 + \left(\{r_i\}-\{r_j\}\right)^T\cdot\{v_n^0\}. \quad (21)$$

The potential energy of normal contact spring is

$$\pi = \frac{1}{2}\cdot k_n\cdot\left(d_n^1\right)^2 = \frac{1}{2}\cdot k_n\cdot\left\{\begin{matrix} \left(d_n^0\right)^2 \\ +2\cdot d_n^0\cdot\{d_i\}^T\cdot\{e_{ni}\} \\ -2\cdot d_n^0\cdot\{d_j\}^T\cdot\{e_{nj}\} \\ +\{d_i\}^T\cdot\{e_{ni}\}\cdot\{e_{ni}\}^T\cdot\{d_i\} \\ -\{d_i\}^T\cdot\{e_{ni}\}\cdot\{e_{nj}\}^T\cdot\{d_j\} \\ -\{d_j\}^T\cdot\{e_{nj}\}\cdot\{e_{ni}\}^T\cdot\{d_i\} \\ +\{d_j\}^T\cdot\{e_{nj}\}\cdot\{e_{nj}\}^T\cdot\{d_j\} \end{matrix}\right\} \quad (22)$$

where

$$\{e_{ni}\} = [T_i(x_i, y_i, z_i)]^T\cdot\{v_n^0\}$$

$$\{e_{nj}\} = \left[T_j\left(x_j, y_j, z_j\right)\right]^T\cdot\{v_n^0\}. \quad (23)$$

The first order variation of potential energy toward $\{d_i\}$ or $\{d_j\}$ contributes the force vector term, while the second-order variation of potential energy contributes the matrix term. Then, contribution of this contact pair to global equation is

$$\begin{pmatrix} +k_n\cdot\{e_{ni}\}\cdot\{e_{ni}\}^T\rightarrow[k_{ii}] \\ -k_n\cdot\{e_{ni}\}\cdot\{e_{nj}\}^T\rightarrow[k_{ij}] \\ -k_n\cdot\{e_{nj}\}\cdot\{e_{ni}\}^T\rightarrow[k_{ji}] \\ +k_n\cdot\{e_{nj}\}\cdot\{e_{nj}\}^T\rightarrow[k_{jj}] \\ -k_n\cdot d_n^0\cdot\{e_{ni}\}\rightarrow\{F_i\} \\ +k_n\cdot d_n^0\cdot\{e_{nj}\}\rightarrow\{F_j\} \end{pmatrix}. \quad (24)$$

3.4 Formulation of tangent locked constraint

The tangential direction is usually unknown in advance. The relative tangential displacement of a contact pair can be computed using the total relative displacement and relative displacement in normal direction.

The previous contact modes may be open, sliding, or locked. If the locked contact mode is inherited from the previous time step, an accumulated tangential displacement may exist. If this accumulated tangential displacement term cannot be omitted, 2 strategies can be used to eliminate these errors. One scheme is to use the first locked contact points as the contact points for this pair. Then, information about the first locked contact points should always be kept and their coordinates should be updated at each step before the mode change. The other scheme is to store the accumulated displacement vector since the first locked mode is established.

The accumulated tangential displacement for locked contact type is taken into consideration by extending Equation 14 to

$$\{\Delta r^m\} = \{r_t^a\} + \{\Delta r\} = \{r_t^a\} + \{r_i\} - \{r_j\} \quad (25)$$

where $\{r_t^a\}$ represent the accumulated displacement for this locked contact pair. Then, the total relative tangential displacement at the end of the time step is

$$(\Delta d_t)^2 = \left(\Delta d_r^m\right)^2 - (\Delta d_n)^2 = \{\Delta r^m\}^T\{\Delta r^m\} - \{\Delta r^m\}^T\cdot\{v_n^0\}\cdot\{v_n^0\}^T\cdot\{\Delta r^m\} = \{\Delta r^m\}^T[N]\{\Delta r^m\}$$
(26)

where

$$[N] = [I] - \{v_n^0\}\cdot\{v_n^0\}^T = \begin{bmatrix} 1-n_x^2 & -n_xn_y & -n_xn_z \\ -n_yn_x & 1-n_y^2 & -n_yn_z \\ -n_zn_x & -n_zn_y & 1-n_z^2 \end{bmatrix}. \quad (27)$$

The potential energy of tangential contact spring is

$$\pi = \frac{1}{2} k_t \cdot (\Delta d_t)^2 = \begin{pmatrix} \{r_t^a\}^T \cdot [N] \cdot \{r_t^a\} \\ +2 \cdot \{r_t^a\}^T \cdot [N] \cdot [T_i(x_i, y_i, z_i)] \cdot \{d_i\} \\ -2 \cdot \{r_t^a\}^T \cdot [N] \cdot \left[T_j\left(x_j, y_j, z_j\right)\right] \cdot \{d_j\} \\ +\{d_i\}^T \cdot [T_i(x_i, y_i, z_i)]^T \cdot [N] \cdot [T_i(x_i, y_i, z_i)] \cdot \{d_i\} \\ -\{d_i\}^T \cdot [T_i(x_i, y_i, z_i)]^T \cdot [N] \cdot \left[T_j\left(x_j, y_j, z_j\right)\right] \cdot \{d_j\} \\ -\{d_j\}^T \cdot \left[T_j\left(x_j, y_j, z_j\right)\right]^T \cdot [N] \cdot [T_i(x_i, y_i, z_i)] \cdot \{d_i\} \\ +\{d_j\}^T \cdot \left[T_j\left(x_j, y_j, z_j\right)\right]^T \cdot [N] \cdot \left[T_j\left(x_j, y_j, z_j\right)\right] \cdot \{d_j\} \end{pmatrix}. \quad (28)$$

Contribution to the global equation

$$\begin{pmatrix} +k_t \cdot [T_i(x_i, y_i, z_i)]^T \cdot [N] \cdot [T_i(x_i, y_i, z_i)] \rightarrow [k_{ii}] \\ -k_t \cdot [T_i(x_i, y_i, z_i)]^T \cdot [N] \cdot \left[T_j\left(x_j, y_j, z_j\right)\right] \rightarrow [k_{ij}] \\ -k_t \cdot \left[T_j\left(x_j, y_j, z_j\right)\right]^T \cdot [N] \cdot [T_i(x_i, y_i, z_i)] \rightarrow [k_{ji}] \\ +k_t \cdot \left[T_j\left(x_j, y_j, z_j\right)\right]^T \cdot [N] \cdot \left[T_j\left(x_j, y_j, z_j\right)\right] \rightarrow [k_{jj}] \\ -k_t \cdot [T_i(x_i, y_i, z_i)]^T \cdot [N]^T \cdot \{r_t^a\} \rightarrow \{F_i\} \\ +k_t \cdot \left[T_j\left(x_j, y_j, z_j\right)\right]^T \cdot [N]^T \cdot \{r_t^a\}^T \rightarrow \{F_j\} \end{pmatrix}. \quad (29)$$

3.5 Formulation of tangent slide constraint

To keep the symmetric, positive define feature of the global matrix in equilibrium equation, an explicit representation form of friction force is used, with a priori assumed sliding direction at beginning of the time step.

The magnitude of friction force is

$$F_{friction} = k_n \cdot |d_n{}^*| \cdot \tan\varphi. \quad (30)$$

The consumed energy due to friction force is

$$\pi = F_{friction} \cdot \Delta d_t^* = k_n \cdot |d_n{}^*| \cdot \tan\varphi \cdot \begin{pmatrix} +\{d_i\}^T [T_i(x_i, y_i, z_i)]^T \cdot \{v_t^*\} \\ -\{d_j\}^T \left[T_j\left(x_j, y_j, z_j\right)\right]^T \cdot \{v_t^*\} \end{pmatrix}. \quad (31)$$

The contribution to global equation is

$$\begin{gathered} -k_n \cdot |d_n{}^*| \cdot \tan\theta \cdot [T_i(x_i, y_i, z_i)]^T \cdot \{v_t^*\} \rightarrow \{F_i\} \\ +k_n \cdot |d_n{}^*| \cdot \tan\theta \cdot \left[T_j\left(x_j, y_j, z_j\right)\right]^T \cdot \{v_t^*\} \rightarrow \{F_j\} \end{gathered} \quad (32)$$

where d_n^* for the first computation cycle in the OCI is d_n^0, and d_n^* for the subsequent cycle is d_n^1 computed in the previous iteration cycle. $\{v_t^*\}$ for the first iteration cycle is $\{v_t^0\}$, and $\{v_t^*\}$ for subsequent iteration cycle is $\{v_t^1\}$ computed in the previous iteration.

3.6 Comparison to other formulations

In addition to the formulation model that is based on 2 closest points and contact plane,[28]other formulations have been proposed based on vertex-to-face contact[18, 19] and edge-edge contact.[20-22] A major difference in the contact models is the representation of normal penetration distance.

Wu[18] used a determinant to represent the normal penetration distance for vertex-to-face contact, which can be given exactly by

$$d=\frac{\Delta}{A}=\frac{-1}{\left|\overrightarrow{p_2p_3}\times\overrightarrow{p_2p_4}\right|}\cdot\begin{vmatrix}1 & x_1+u_1 & y_1+v_1 & z_1+w_1\\ 1 & x_2+u_2 & y_2+v_2 & z_2+w_2\\ 1 & x_3+u_3 & y_3+v_3 & z_3+w_3\\ 1 & x_4+u_4 & y_4+v_4 & z_4+w_4\end{vmatrix}. \quad (33)$$

Then, omitting the second-order and third-order parts, Δ can be represented by

$$\Delta \approx -\left(\begin{vmatrix}1 & x_1 & y_1 & z_1\\ 1 & x_2 & y_2 & z_2\\ 1 & x_3 & y_3 & z_3\\ 1 & x_4 & y_4 & z_4\end{vmatrix}+\begin{vmatrix}1 & u_1 & y_1 & z_1\\ 1 & u_2 & y_2 & z_2\\ 1 & u_3 & y_3 & z_3\\ 1 & u_4 & y_4 & z_4\end{vmatrix}+\begin{vmatrix}1 & x_1 & v_1 & z_1\\ 1 & x_2 & v_2 & z_2\\ 1 & x_3 & v_3 & z_3\\ 1 & x_4 & v_4 & z_4\end{vmatrix}+\begin{vmatrix}1 & x_1 & y_1 & w_1\\ 1 & x_2 & y_2 & w_2\\ 1 & x_3 & y_3 & w_3\\ 1 & x_4 & y_4 & w_4\end{vmatrix}\right).$$

(34)

And A can be represented by

$$A \approx \sqrt{\begin{vmatrix}1 & y_2 & z_2\\ 1 & y_3 & z_3\\ 1 & y_4 & z_4\end{vmatrix}^2+\begin{vmatrix}1 & z_2 & x_2\\ 1 & z_3 & x_3\\ 1 & z_4 & x_4\end{vmatrix}^2+\begin{vmatrix}1 & x_2 & y_2\\ 1 & x_3 & y_3\\ 1 & x_4 & y_4\end{vmatrix}^2}. \quad (35)$$

The normal distance at the end of current time step can be represented in a simple form.

$$d=\frac{Volu}{A}+\{e_i\}\cdot\{d_i\}+\left\{g_j\right\}\cdot\{d_j\} \quad (36)$$

As unit normal vector of the contact plane can be computed using 3 points on the face,

$$\left\{v_n^0\right\} = \frac{\overrightarrow{p_2p_3} \times \overrightarrow{p_2p_4}}{\left|\overrightarrow{p_2p_3} \times \overrightarrow{p_2p_4}\right|} = \frac{\begin{vmatrix} i & j & k \\ x_3-x_2 & y_3-y_2 & z_3-z_2 \\ x_4-x_2 & y_4-y_2 & z_4-z_2 \end{vmatrix}}{\sqrt{\begin{vmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix}^2 + \begin{vmatrix} 1 & z_2 & x_2 \\ 1 & z_3 & x_3 \\ 1 & z_4 & x_4 \end{vmatrix}^2 + \begin{vmatrix} 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{vmatrix}^2}}. \quad (37)$$

Comparing formula 21 and 36, we found the following equation

$$\frac{Volu}{A} = d_n^0 \quad (38)$$

$$\{e_i\} = [T_1(x_1, y_1, z_1)]\{v_1\} = [T_i(x_i, y_i, z_i)]\left\{v_n^0\right\} = \{e_{ni}\} \quad (39)$$

$$\begin{aligned}\left\{g_j\right\} &= [T_2(x_2, y_2, z_2)]\{v_2\} + [T_3(x_3, y_3, z_3)]\{v_3\} + [T_4(x_4, y_4, z_4)]\{v_4\} \\ &\approx \left[T_j\left(x_j, y_j, z_j\right)\right]\left\{v_n^0\right\} = \{e_{nj}\}.\end{aligned} \quad (40)$$

The 2 formulations given by Equations 21 and 36 produce the same translation and rotation terms for vertex-to-face contact.

4 IMPLEMENTATION IN DDA

4.1 The proposed procedure

DDA is a step-based computation method for dynamic and static problems. The implementation of the proposed contact analysis model in DDA procedure is shown in Figure 6. At the beginning of each step, the contact detection algorithm is executed to find the 4 basic contact types, then the contact constraint list can be established by identifying the contact point, contact normal direction, tangential direction, and the contact mode. Then, the contact terms are implemented into the equilibrium equation. The contact mode of all contact pairs is verified in the open-close check. After the convergence of the contact mode, the displacement, velocity, stress, strain, and other terms are updated for the next step.

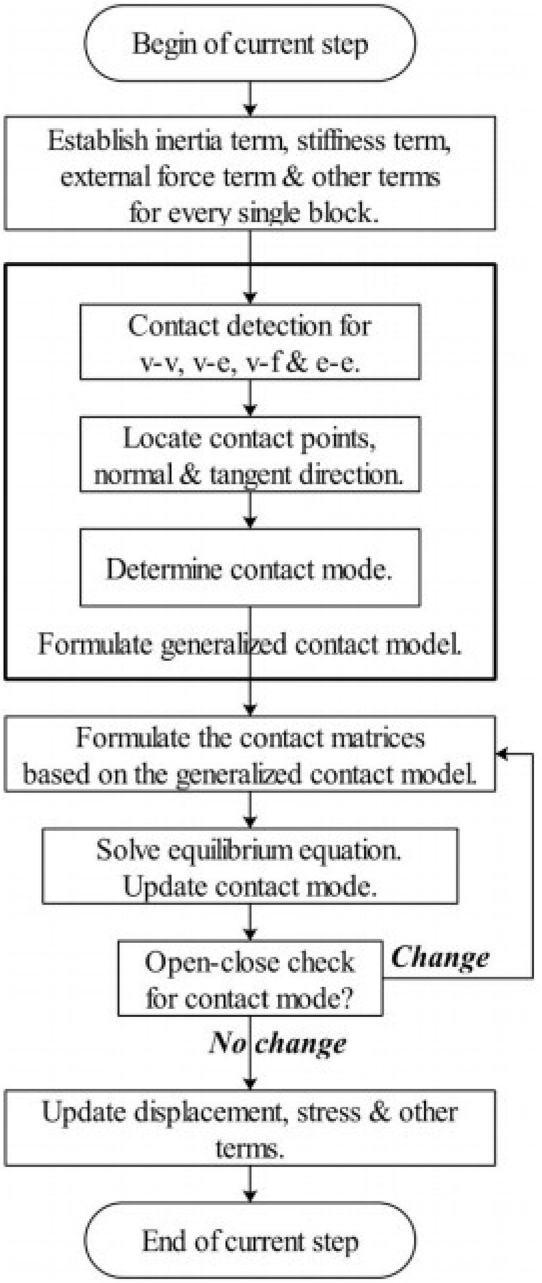
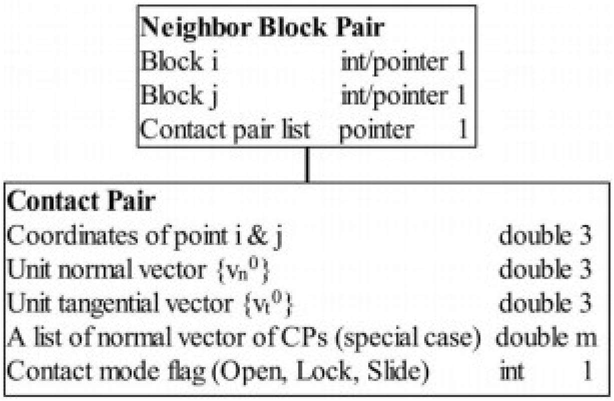

Begin of current step
Establish inertia term, stiffness term, external force term & other terms for every single block.
Contact detection for v-v, v-e, v-f & e-e.
Locate contact points, normal & tangent direction.
Determine contact mode.
Formulate generalized contact model.
Formulate the contact matrices based on the generalized contact model.
Solve equilibrium equation. Update contact mode.
Open-close check for contact mode?
Change
No change
Update displacement, stress & other terms.
End of current step

Figure 6

Implementation of the generalized contact analysis procedure in DDA framework

4.2 Data structure for contact list

For traditional algorithms that detect contact with 4 basic types and formulate the penalty function with vertex-to-face and edge-to-edge types, the data structure is usually in one of the following formats: (1) based on a list with lots of pointers that point to the basic elements (vertices, edges, faces) in blocks; (2) based on a list that save lots of geometrical data. For method 1, the extensive usage of pointers in a program tends to be less robust, especially for parallel computation models. For method 2, at least coordinates of 4 points should be used for traditional v-f and cross e-e penalty formulation; For v-v or v-e types to be identified as v-f or cross e-e types, a list of 4-point pairs should be stored for the formulation. By contrast, the generalized contact analysis model maintains a list of class that contains the coordinates of 2 contact points, the vectors of the contact normal and contact tangential directions, an integer for contact mode, and a vector list for potential contact normal directions for special cases. The data structure and memory requirement for basic elements of the contact list are shown in Figure 7. Only some integer, double, and pointer data are stored during contact analysis. The vector list is only activated when pure v-v or pure v-e type is detected. The data structure based on the generalized contact model is simpler and less memory-intensive compared with algorithms that store 4 basic types and formulate contact constraints based on v-f and cross e-e types. Meanwhile, it makes the program more robust and facilitates the OCI check.



Neighbor Block Pair		
Block i	int/pointer	1
Block j	int/pointer	1
Contact pair list	pointer	1

Contact Pair		
Coordinates of point i & j	double	3
Unit normal vector $\{v_n^0\}$	double	3
Unit tangential vector $\{v_t^0\}$	double	3
A list of normal vector of CPs (special case)	double	m
Contact mode flag (Open, Lock, Slide)	int	1

Figure 7

The data structure for basic elements in the contact list

4.3 Revision in open-close iteration

DDA uses an OCI process to update contact modes until reaching the convergence of penetration/no penetration constraints. The criteria for judging the contact modes were already presented in Table 2. A flowchart to exhibit the OCI process is presented in Figure 8. By using the proposed contact computation model, only coordinates of the 2 contact points of each contact constraint need to be updated in the OCI. Then, the total, normal, and tangential displacements can be updated to determine the new contact mode. In the current version, only the mode transfer between open and closed (locked, sliding) status is checked to fulfill the OCI convergence.

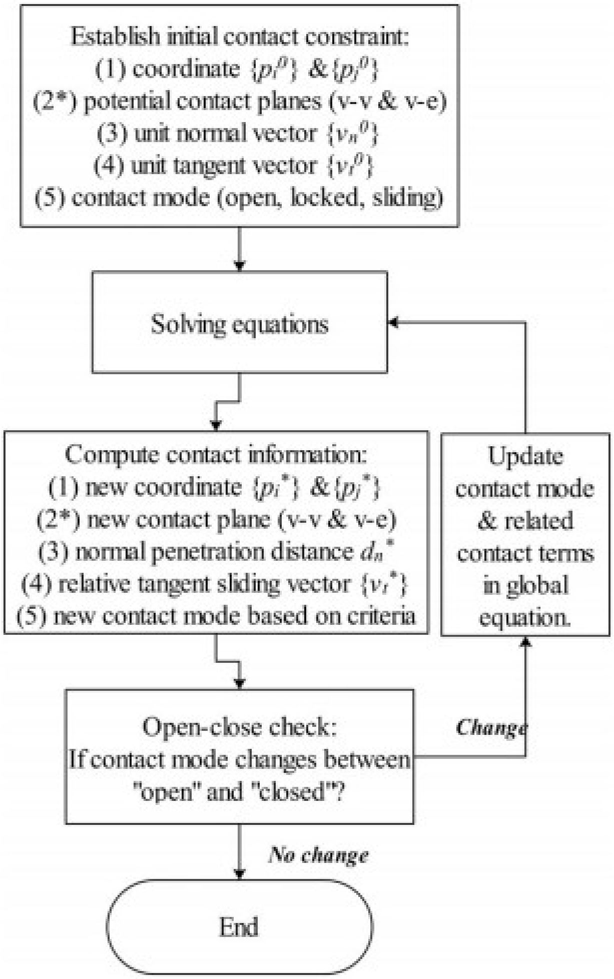

Establish initial contact constraint:
(1) coordinate $\{p_i^0\}$ & $\{p_j^0\}$
(2*) potential contact planes (v-v & v-e)
(3) unit normal vector $\{v_n^0\}$
(4) unit tangent vector $\{v_t^0\}$
(5) contact mode (open, locked, sliding)
Solving equations
Compute contact information:
(1) new coordinate $\{p_i^*\}$ & $\{p_j^*\}$
(2*) new contact plane (v-v & v-e)
(3) normal penetration distance d_n^*
(4) relative tangent sliding vector $\{v_t^*\}$
(5) new contact mode based on criteria
Update contact mode & related contact terms in global equation.
Open-close check: If contact mode changes between "open" and "closed"?
Change
No change
End

Figure 8

The open-close iteration process

Additional operations can also be executed during the OCI to increase the computation accuracy, such as (1) using the newly computed coordinates of contact points and the shortest exit scheme to update the contact plane for v-v and v-e pair that have multiple potential contact planes, (2) using relative displacement to update the coordinates of contact points for contact pairs from open to close, and (3) using the computed displacement to update tangent sliding vector (friction force direction) and normal penetration distance for computing the friction force.

5 POTENTIAL ISSUES IN CONTACT ANALYSIS

Considering the potential change in contact types in the contact analysis of discrete block systems, the identification algorithm should be robust enough to deal with the change of contact points and contact plane during the change of contact types. The formulation of the contact constraint should also be accurate enough to satisfy the convergence of OCI.

5.1 Revised contact location

In Section 2, we presented algorithms to calculate contact points for the generalized model. However, the contact points need revision when the contact mode changes from open to close, to make the formulation more accurate. Zhang et al[22] presented an iterative method to compute the first entrance point in order to make the representation more accurate. Here, we present a vector analysis method to revise the contact points based on the vertex-to-face pair and the edge-to-edge pair in the OCI.

For vertex-to-face contact model shown in Figure 2A, assuming $\{p_i\}$ is the vertex from block i, $\{p_j\}$ is the projection point of $\{p_i\}$ on the contact face from block j, contact point $\{p_j\}$ can be revised to the penetrating point on the face.

The relative displacement of the contact point can be represented by

$$\left\{\Delta r^*\right\} = \left\{\Delta r_n^*\right\} + \left\{\Delta r_t^*\right\}. \quad (41)$$

Assuming the relative displacement changes linearly, a linear approximation of the penetration point $\{p_j^*\}$ can be represented by

$$\left\{p_j^*\right\} = \left\{p_j^0\right\} + t^* \cdot \left(\left\{\Delta r^*\right\} - \left\{\Delta r_n^*\right\}\right) \quad (42)$$

where $\{\Delta r^*\}$ represents the relative displacement of the contact points, $\{\Delta r_n^*\}$ represents the relative displacement along normal direction, and t can be computed by

$$t^* = \frac{\left|d_n^0\right|}{\left|d_n^0\right| + \left|d_n^*\right|} \quad (43)$$

where d_n^0 represents the normal distance before penetration, and d_n^* represents the penetration distance in the current computing cycle.

For edge-to-edge contact type shown in Figure 2B, the relative displacement of 2 contact points can be represented in the following form,

$$\left\{\Delta r^*\right\} - \left\{\Delta r_n^*\right\} = \left\{\Delta r_t^*\right\} = t_1^*\{e_1\} + t_2^*\{e_2\} \quad (44)$$

where $\{e_1\}$ and $\{e_2\}$ represent unit vectors along the contact edges p_1p_2 and p_3p_4, respectively. Parameters t_1^* and t_2^* can be obtained from Equation 45:

$$t_1^* = \frac{\Delta r_{tx}^* \cdot (e_{2y} + e_{2z}) - \left(\Delta r_{ty}^* + \Delta r_{tz}^*\right) \cdot e_{2x}}{e_{1x} - e_{1y} - e_{1z}}$$

$$t_2^* = \frac{\Delta r_{tx}^* \cdot (e_{1y} + e_{1z}) - \left(\Delta r_{ty}^* + \Delta r_{tz}^*\right) \cdot e_{1x}}{e_{2x} - e_{2y} - e_{2z}}. \quad (45)$$

Using the linear displacement assumption, the revised contact points on 2 edges can be represented by

$$\{p_i^*\} = \{p_i^0\} + t^* \cdot t_1^* \cdot \{e_1\}$$
$$\left\{p_j^*\right\} = \left\{p_j^0\right\} + t^* \cdot t_2^* \cdot \{e_2\} \quad (46)$$

where t^* can be computed using Equation 43.

It should be mentioned that when the contact points are revised in OCI with the contact mode changing from open to closed, the term $\{\Delta r^m\}$ in Equation 25 should be replaced by the Equation 47:

$$\{\Delta r^m\} = \left(\{p_i^*\} - \left\{p_j^*\right\}\right) + \{\Delta r^*\} = \left(\{p_i^*\} - \left\{p_j^*\right\}\right) + \{r_i^*\} - \left\{r_j^*\right\}. \quad (47)$$

5.2 Error due to rotation of contact plane

An error in normal penetration may appear in consecutive step, when contact plane rotates and sliding occurs. This error is mainly due to the assumption that the normal vector of the contact plane is constant during the computation.

Figure 9 shows a section along contact sliding direction. Assuming the rotation angle of contact plane in this section is θ, the relative displacement is d, and the displacement errors in normal direction d_{ne} and tangent direction d_{te} can be estimated as:

$$d_{ne} = \left|d_n^r - d_n^1\right| = |d \cdot \sin\alpha - d \cdot \sin(\alpha + \theta)| \approx d \cdot \cos\alpha \cdot \theta \quad (48)$$

$$d_{te} = \left|d_t^r - d_t^1\right| = |d \cdot \cos\alpha - d \cdot \cos(\alpha + \theta)| \approx d \cdot \sin\alpha \cdot \theta. \quad (49)$$

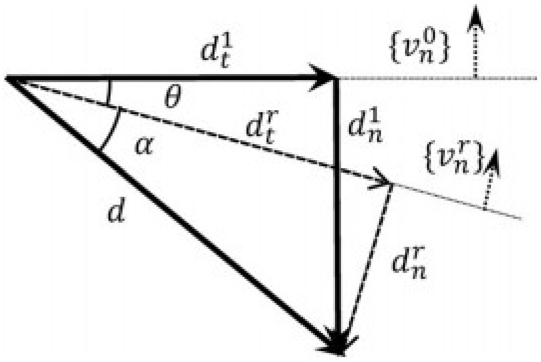


Figure 9

A schematic diagram for analysis of contact plane rotation

θ is set to 1 to 3° in computation, and in the case of sliding, α is usually smaller than 45°, which is related to the ratio of tangent sliding distance and normal penetration distance. In sliding cases, $d_{te} < d_{ne} < d$ is usually satisfied. Large d_{ne} may cause unrealistic initial contact force that leads to irrational movement of the contacting blocks. To control the error caused by using constant normal vector in contact formulation, the displacement and block rotation angle should be controlled by a tolerance, and very large penalty parameters should be avoided. In the test examples presented herein, θ is set 1 to 3°, and the displacement d is controlled by Equation 50:

$$d<\eta\cdot l_{emin}\cdot \sin\theta \quad (50)$$

where l_{emin} represents the minimum edge length of blocks from the block system, and η is a coefficient that is set 0.1 to 1.

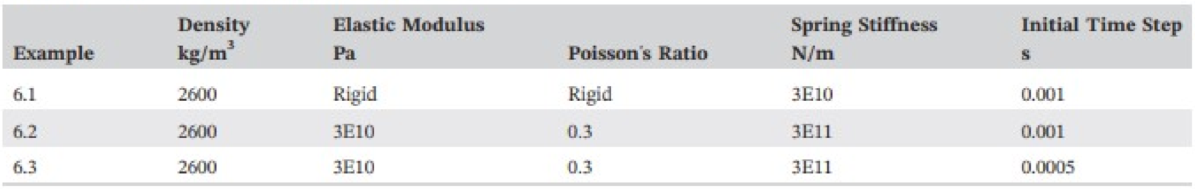
6 ILLUSTRATIVE EXAMPLES

The generalized contact model is implemented into our 3-D DDA program, and 3 typical examples are tested to verify the model.

6.1 Wedge sliding

Wedge sliding is a typical failure mode in rock slope engineering. In this example, the wedge block sliding case is modeled to show contact model representation and to test the accuracy of the computation model for concave blocks. A wedge block is generated by considering 2 joints (joint 1: dip = 50°, dip direction = 110°; joint 2: dip = 55°, dip direction = 240°). The computation parameters are shown in Table 3.

TABLE 3 Physical parameters in 3 illustrative examples



Example	Density kg/m^3	Elastic Modulus Pa	Poisson's Ratio	Spring Stiffness N/m	Initial Time Step s
6.1	2600	Rigid	Rigid	3E10	0.001
6.2	2600	3E10	0.3	3E11	0.001
6.3	2600	3E10	0.3	3E11	0.0005

Without enough cohesion and friction angle value, the wedge block will slide along the base block. The sliding process without cohesion or friction is shown in Figure 10. The sliding velocity is recorded with unit m/s. The contact information for the initial model is listed in Table 4. Six contact constraints are established for the 4 contact positions as shown in Figure 11. The formation of the contact constraints is according to the identification rule in chapter 2. Based on the generalized model, we only need to record and update the coordinate of 2 points in contact analysis, while the contact normal direction keeps constant. Tracking of 4 points in vertex-to-face and edge-to-edge model is avoided.

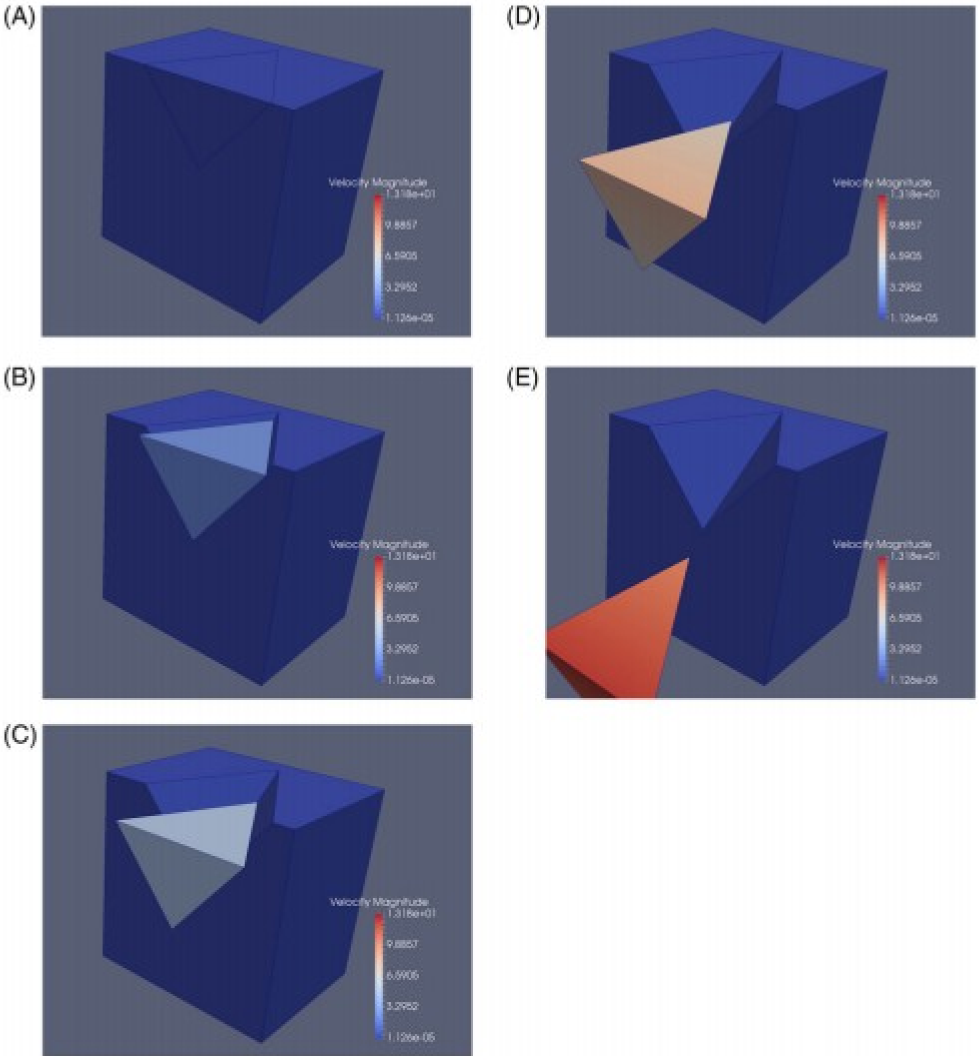
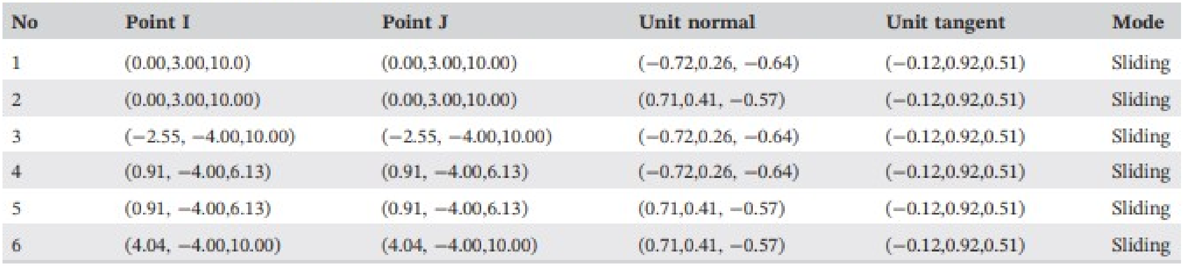


Figure 10

A wedge sliding model: A, *t* = 0 s, step = 0; B, *t* = 0.5 s, step = 600; C, *t* = 1 s, step = 2000; D, *t* = 1.5 s, step = 4000; E, *t* = 2 s, step = 7400

TABLE 4 Contact information in example 6.1



No	Point I	Point J	Unit normal	Unit tangent	Mode
1	(0.00,3.00,10.0)	(0.00,3.00,10.00)	(−0.72,0.26, −0.64)	(−0.12,0.92,0.51)	Sliding
2	(0.00,3.00,10.00)	(0.00,3.00,10.00)	(0.71,0.41, −0.57)	(−0.12,0.92,0.51)	Sliding
3	(−2.55, −4.00,10.00)	(−2.55, −4.00,10.00)	(−0.72,0.26, −0.64)	(−0.12,0.92,0.51)	Sliding
4	(0.91, −4.00,6.13)	(0.91, −4.00,6.13)	(−0.72,0.26, −0.64)	(−0.12,0.92,0.51)	Sliding
5	(0.91, −4.00,6.13)	(0.91, −4.00,6.13)	(0.71,0.41, −0.57)	(−0.12,0.92,0.51)	Sliding
6	(4.04, −4.00,10.00)	(4.04, −4.00,10.00)	(0.71,0.41, −0.57)	(−0.12,0.92,0.51)	Sliding

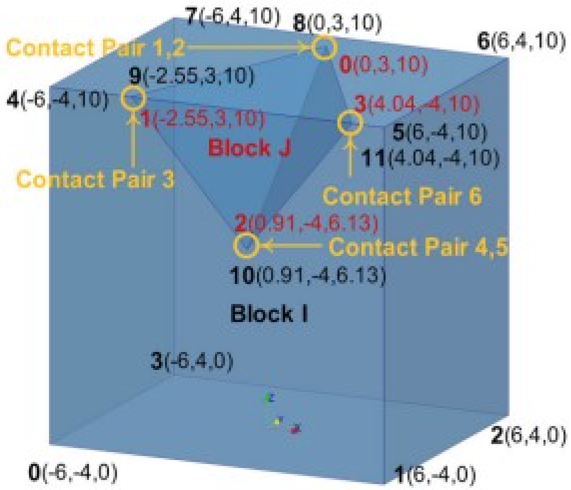


Figure 11

Geometrical details of the wedge block example

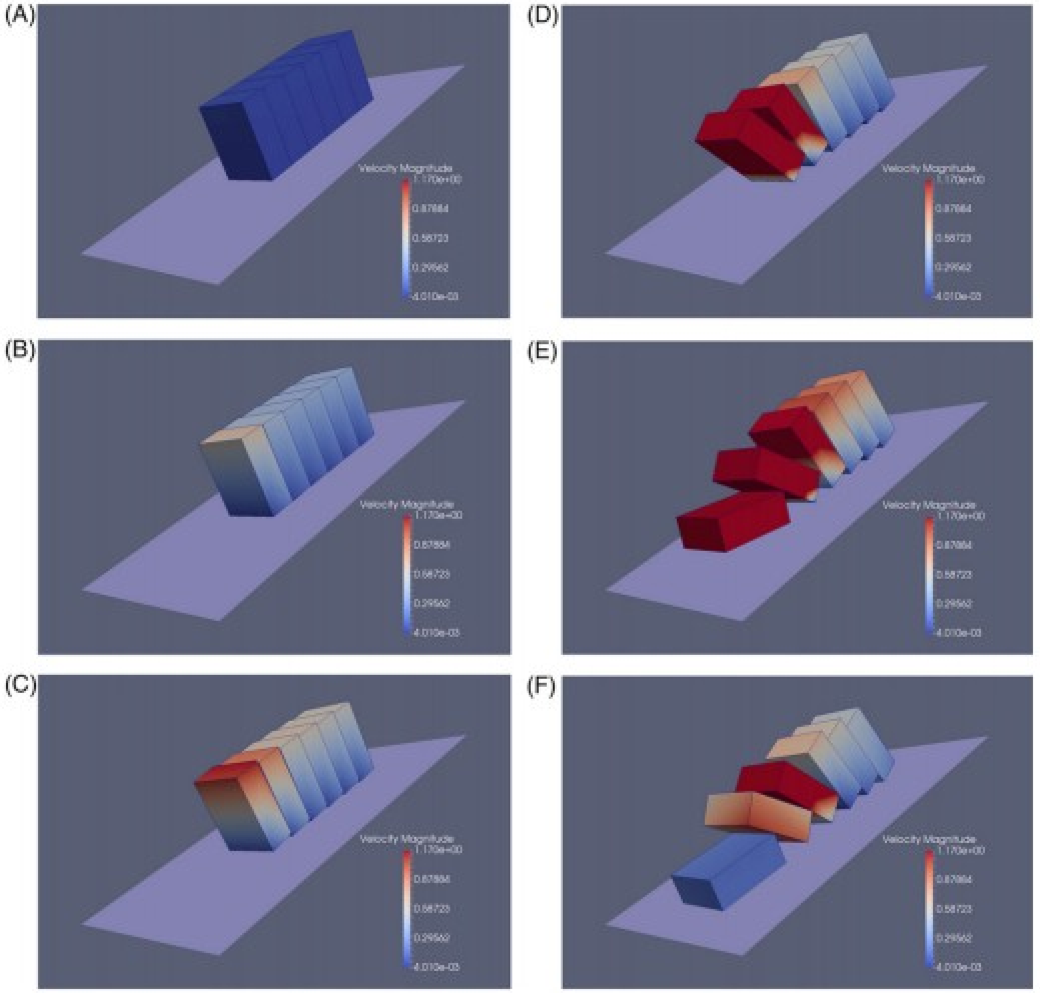


Figure 12

A block toppling model: A, t = 0 s, step = 0; B, *t* = 0.365 s, step = 1000; C, *t* = 0.761 s, step = 2000; D, *t* = 1.520 s, step = 3000; E, *t* = 2.003 s, step = 4000; F, *t* = 2.546 s, step = 5000

Considering friction angle only, an analytical solution of the factor of safety for the wedge block case can be computed by Equation 51:

$$Fs = \frac{\sin\beta}{\sin(\xi/2)} \cdot \frac{\tan\phi}{\tan\psi_i}. \quad (51)$$

The friction angles for the 2 joints are set at 40°, and the *Fs* is 2.198 according to Equation 51. In 3-D DDA, a bisection method is used to trace the critical friction angle φ_r at which sliding of all contact pair between the wedge and base block starts. The factor of safety computed using DDA is given by

$$Fs = \frac{\tan\phi}{\tan\phi_r} \quad (52)$$

where φ is the initial friction angle, and φ_r is the reduced friction angle. The critical friction angle given by DDA is around 21.25, and related *Fs* is 2.158.

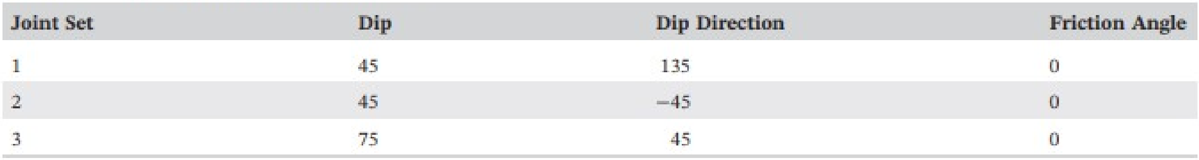
6.2 Toppling

Toppling involves rotation of blocks of rock about a fixed base. In this example, a simple block toppling model is built to verify the validity of contact computation model. In this model, the slope angle is set at 30°. There are 6 blocks in total, with dimensions 3 m × 1.5 m × 4 m. Friction angle between the blocks and the surface is set to 40°, while the friction angle between blocks is set to 15°. Other physical parameters are shown in Table 3. The toppling process is shown in Figure 12, and block velocity is exhibited with the unit m/s. This simulation result shows the generalized contact model can treat rotating block correctly.

6.3 Failure of a block system

A model of block system is created by cutting a hexahedral region with 3 sets of joints. Dip, dip direction, and friction angle for the 3 joints are summarized in Table 5. The overall dimensions of the hexahedral region are 10 m × 10 m × 10 m. A total of 18 blocks are generated during the cutting process and the friction angle for all joints is set to zero in order follow the progression of the block movements under the influence of gravity. The computation parameters are shown in Table 3. The velocity of each block is recorded with the unit m/s, as shown in Figure 13. This simulation shows the potential of the proposed contact computation model for treating complex polyhedral blocks.

TABLE 5 Joint set information in example 6.3



Joint Set	Dip	Dip Direction	Friction Angle
1	45	135	0
2	45	−45	0
3	75	45	0

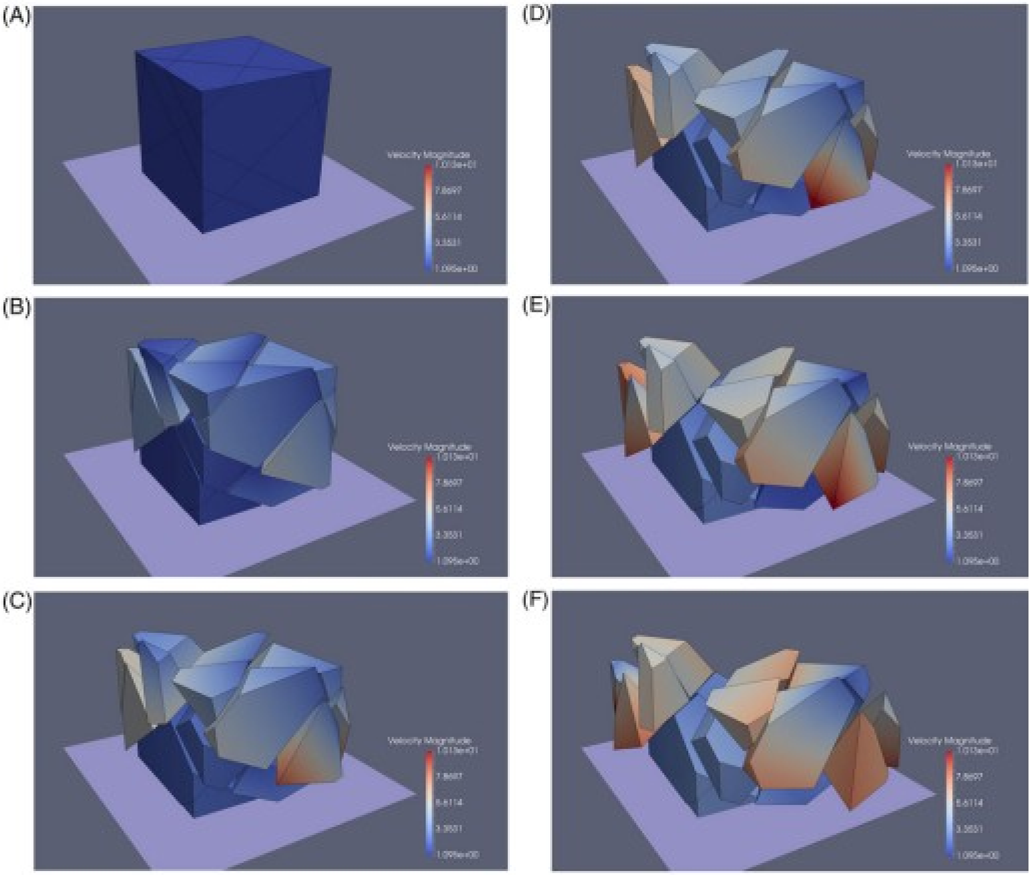


Figure 13

Failure process of a block system: A, t = 0 s, step = 0; B, *t* = 0.45 s, step = 1000; C, *t* = 0.67 s, step = 2000; D, *t* = 0.8 s, step = 3000; E, *t* = 0.925 s, step = 4000; F, *t* = 1.05 s, step = 5000

7 DISCUSSIONS AND CONCLUSIONS

We present a generalized contact computation model for arbitrary polyhedron in 3-D DDA. A list of contact constraints can be established by identification of 4 basic contact types (v-v, v-e, v-f, and cross e-e) for arbitrary polyhedron. Each constraint contains 2 contact points, unique contact plane, and unique contact mode. The following advantages make this generalized model promising in 3-D DDA: (1) The generalized contact constraints result in a simpler and less memory-intensive data structure compared with previous algorithms that record 4 basic types in contact list; (2) The penalty formulation used in this generalized contact constraint model avoids the necessity of separate treatment of the vertex-to-face and edge-to-edge contacts; (3) The initial assumption and updated estimate of the contact mode are all based on 2 contact points and a constant contact plane, which eliminates the number of points to be computed in the OCI; and (4) By considering multiple contact planes, this generalized model extends the point-to-plane model presented by Beyabanaki et al[28] to both convex and concave polyhedral shapes. Additionally, to maintain the accuracy and stability of the contact analysis for blocks with linear displacement assumption, the displacement of the block system in each step should be carefully controlled, and the strategy in chapter 5.2 can be used as a reference.

ACKNOWLEDGEMENTS

This study was supported by the National Natural Science Foundation of China (51479191,11672360, 41731284) and Hubei Provincial Natural Science Foundation of China (2016CFA023). The first author was supported by China Scholarship Council during his study in UC Berkeley. The third author was supported in part by the Edward G. and John R. Cahill Chair endowment.

References

1. Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. Geotechnique. 1979;29(1):47-65.

2. 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(3):107-116.

3. Shi GH. Discontinuous deformation analysis—a new numerical model for the statics and dynamics of block system. Ph.D. 1988, University of California, Berkeley, 1988.

4. Shi GH. Three dimensional discontinuous deformation analysis. Fourth International Conference on Discontinuous Deformation Analysis. Glasgow, 2001; 1–21.

5. Shi GH. Contact Theory. Sci China Technol Sci. 2015;58:1-47.

6. Feng YT, Han K, Owen DRJ. Energy-conserving contact interaction models for arbitrarily shaped discrete elements. Comput Methods Appl Mech Eng. 2012;205:169-177.

7. Smeets B, Odenthal T, Keresztes J, et al. Modeling contact interactions between triangulated rounded bodies for the discrete element method. Comput Methods Appl Mech Eng. 2014;277:219-238.

8. Smeets B, Odenthal T, Vanmaercke S, Ramon H. Polygon-based contact description for modeling arbitrary polyhedra in the discrete element method. Comput Methods Appl Mech Eng. 2015;290:277-289.

9. Houlsby GT. Potential particles: a method for modelling non-circular particles in DEM. Comput Geotech. 2009;36(6):953-959.

10. Ahmed S, Harkness J, len Pen L, et al. Numerical modelling of railway ballast at the particle scale. Int J Numer Anal Methods Geomech. 2016;40(5):713-737.

11. Ji S, Sun S, Yan Y. Discrete element modeling of dynamic behaviors of railway ballast under cyclic loading with dilated polyhedra. Int J Numer Anal Methods Geomech. 2017;41(2):180-197.

12. Senseney CT, Duan Z, Zhang B, et al. Combined spheropolyhedral discrete element (DE)–finite element (FE) computational modeling of vertical plate loading on cohesionless soil. Acta Geotech. 2017;1-11.

13. Munjiza A, Andrews KRF, White JK. Penalty function method for combined finite-discrete element systems comprising large number of separate bodies. Int J Numer Methods Eng. 2000;49(11):1377-1396.

14. Yan CZ, Zheng H. A new potential function for the calculation of contact forces in the combined finite-discrete element method. Int J Numer Anal Methods Geomech. 2017;41(2):265-283.

15. Zhang H, Chen GQ, Zheng L, et al. Detection of contacts between three-dimensional polyhedral blocks for discontinuous deformation analysis. Int J Rock Mech Min Sci. 2015;78:57-73.

16. Wu W, Zhu HH, Zhuang XY, Ma GW, Cai YC. A multi-shell cover algorithm for contact detection in the three dimensional discontinuous deformation analysis. Theor Appl Fract Mech. 2014;72:136-149.

17. Zheng F, Jiao YY, Zhang XL, Tan F, Wang L, Zhao Q. Object-oriented contact detection approach for three-dimensional discontinuous deformation analysis based on entrance block theory. Int J Geomech. 2016. E4016009

18. Wu JH, Juang CH, Lin HM. Vertex-to-face contact searching algorithm for three-dimensional frictionless contact problems. Int J Numer Methods Eng. 2005;63(6):876-897.

19. Jiang QH, Yeung MR. A model of point-to-face contact for three-dimensional discontinuous deformation analysis. Rock Mech Rock Eng. 2004;37(2):95-116.

20. Wu JH. New edge-to-edge contact calculating algorithm in three-dimensional discrete numerical analysis. Adv Eng Softw. 2008;39(1):15-24.

21. Yeung MR, Jiang QH, Sun N. A model of edge-to-edge contact for three-dimensional discontinuous deformation analysis. Comput Geotech. 2007;34(3):175-186.

22. Zhang H, Liu SG, Zheng L, et al. Extensions of edge-to-edge contact model in three-dimensional discontinuous deformation analysis for friction analysis. Comput Geotech. 2016;71:261-275.

23. Zheng F, Jiao YY, Gardner M, Sitar N. A fast direct search algorithm for contact detection of convex polygonal or polyhedral particles. Comput Geotech. 2017;87:76-85.

24. Nezami E, Hashash Y, Zhao D, Ghaboussi J. A fast contact detection algorithm for 3-D discrete element method. Comput Geotech. 2004;31(7):575-587.

25. Nezami E, Hashash Y, Zhao D, Ghaboussi J. Shortest link method for contact detection in discrete element method. Int J Numer Anal Methods Geomech. 2006;30(8):783-801.

26. Boon CW, Houlsby GT, Utili S. A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method. Comput Geotech. 2012;44:73-82.

27. Keneti AR, Jafari A, Wu JH. A new algorithm to identify contact patterns between convex blocks for three-dimensional discontinuous deformation analysis. Comput Geotech. 2008;35(5):746-759.

28. Beyabanaki S, Mikola RG, Hatami K. Three-dimensional discontinuous deformation analysis (3-D DDA) using a new contact resolution algorithm. Comput Geotech. 2008;35(3):346-356.

29. Liu XL, Lemos JV. Procedure for contact detection in discrete element analysis. Adv Eng Softw. 2001;32(5):409-415.

30. Ahn TY, Song JJ. New contact-definition algorithm using inscribed spheres for 3D discontinuous deformation analysis. Int J Comput Methods. 2011;8(02):171-191.

31. Hart R, Cundall PA, Lemos J. Formulation of a three-dimensional distinct element model—part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks. Int J Rock Mech Min Sci Geomech Abstr. 1988;25(3):117-125.

32. Huang GH, Jiao YY, Wang L, Zhao Q. Three-dimensional spherical DDA method for modeling friction problems. Int J Geomech. 2016; E4016016.