On Minkowski Difference Based Contact Detection in Discrete/Discontinuous Modelling of Convex Polygons/Polyhedra: Algorithms and Implementation

Y. T. Feng[1,2,*] Yuanqiang Tan[2]
[1] Zienkiewicz Centre for Computational Engineering, Swansea University, UK
[2] The Institute of Manufacturing Engineering, Huaqiao University, China

Abstract

Contact detection for convex polygons/polyhedra has been a critical issue in discrete/discontinuous modelling, such as the discrete element method (DEM) and the discontinuous deformation analysis (DDA). The recently developed 3D contact theory for polyhedra in DDA defines the so-called entrance block of two polyhedra and reduces the contact to evaluate the distance between the reference point to the corresponding entrance block, but effective implementation is still lacking. In this paper, the equivalence of the entrance block and the Minkowski difference of two polyhedra is emphasised and two well-know Minkowski difference based contact detection and overlap computation algorithms, GJK and EPA (expending polytope algorithm), are chosen as the possible numerical approaches to the 3D contact theory for DDA, and also as alternatives for computing polyhedral contact features in DEM. The key algorithmic issues are outlined and their important features are highlighted. Numerical examples indicate that the average number of updates required in GJK for polyhedral contact is around 6, and only 1 or 2 iterations are needed in EPA to find the overlap and all the relevant contact features when the overlap between polyhedra is small.

KEYWORDS: Contact detection, Convex polygon/polyhedron, Minkowski sum/difference, support point, GJK algorithm, EPA

1 Introduction

Contact detection plays a central role in modelling contacts between geometric shapes and objects. Applications range from computational geometry, computer graphics and games, computer-aided design, robotic motion planning, to name a few. It also forms a core operation in various mechanical modellings of objects, such as Finite element methods (FEM) for contact mechanics [1], Discrete Element Methods (DEM) [2] and the Discontinuous Deformation Analysis (DDA) [3].

Contact detection is typically divided into two stages: global search and local resolution. In the first stage, geometric objects concerned, regardless of their actual shapes, are approximated by a simple enclosure or bounding geometric entity, such as a bounding sphere, or bounding box including axis-aligned (AABB) or oriented (OBB). A global search is then performed to these unified bounding shapes to establish a potential contact list for each object. In the second stage, the contact between any two objects with their bounding entities in

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

overlap is further checked based on the actual geometric shapes to establish the real contact state: separation, touch or overlap (penetration).

Many effective global search algorithms for large scale problems have been proposed over the last 50 years, including tree based, such as the k-t tree [4] and its variants [5], and grid based, such as NBS [6], C-Grid [7] or D-cell [8]. There are no major unresolved issues for the global search. Thus the local contact resolution is the focus of contact detection and particularly in discrete/discontinuous modelling of complicated objects by DEM and DDA.

In addition to the 'on-off' nature of a contact that needs to be established for two objects, the distance in separation, or the overlap in penetration between them is often required. In both DEM and DDA, other contact features, such as the contact normal and contact points, are also needed. Although local resolution is a pure geometric problem, which contact features to be evaluated and what corresponding optimal numerical procedures to be adopted are often dictated by underlying contact theories for the problem concerned.

In DEM, modelling contact between non-spherical objects has been an active research area over the last 30 years. For polygons/polyhedra, further to conventional overlap based models, the so-called 'common-plane' approach is proposed in [9]. These models and those for modelling other non-spherical objects are, however, often developed in an *ad-hoc* manner. To address the issue, an energy based general contact framework for arbitrarily shaped objects, including polygons and polyhedra, is established in [10, 11]. Within this framework, when a contact feature related energy function is properly defined, the energy conservation in an elastic contact is guaranteed, which provides better numerical stability particularly for polygons and polyhedra. In [10, 11] the overlap area/volume based energy function has been proposed and the corresponding contact model for polygons/polyhedra exhibits some preferred properties. It can also be theoretically proved that an " *ad-hoc*" overlap based contact model, particularly for 3D cases, may not obey the principle of energy conservation.

Recently, Shi [12, 13] proposed a contact theory for 3D objects (mainly polyhedron) in DDA. A lack of rigorous contact theories has hindered the further development of DDA for realistic applications. The centre to Shi's contact theory is the definition of the so-called entrance block of two polygons in 2D and polyhedra in 3D, and then the contact between two polygons/polyhedra, which needs to be performed in each 'open-close' iteration, is reduced to determine the distance between the reference point to the corresponding entrance block, or its boundary polygons. Thus, the key to the implementation aspect of this theory is to find effective computational procedures to evaluate the distance and other contact features. Effective computational implementation of the theory in 3D seems to be lacking.

Parallel to the developments in the discrete/discontinuous modelling of contacting objects, there has been considerable research work on collision detection between geometric shapes mainly in the fields of computer graphics and games, and robotics over the last 30 years (see a review on the topic in [14]). Apart from simple shapes with analytical or parametric surface descriptions, such as spheres and ellipsoids, general shapes are commonly represented by a triangulated surface mesh. Thus, contact between two convex polyhedra often becomes the core operation in many applications. Among the proposed collision detection methods for polyhedra, the most popular algorithms and their implementations include I-Collid [15] and V-Clip [16].

Another well-known collision detection algorithm is called the GJK algorithm, originally developed for robotic path planning by Gilbert, Johnson and Keerthi [17]. The algorithm is based on the fact that the contact between two shapes is equivalent to the enclosure of the origin in the Minkowski difference of the two shapes. It dynamically constructs/updates a simplex and returns it when the simplex encloses the origin, or reports a non-contact state when such a state is found. The algorithm crucially relies on the definition of the so-called *support function* or *point* for each convex shape involved to avoid explicitly constructing the Minkowski difference. Unlike I-Collid and V-Clip which are restricted to (convex) polyhedra, the GJK algorithm is applicable to any convex shape where the support function can be (numerically) defined, and therefore offers greater versatility. This, along with the simplicity of the algorithm, makes its implementation straightforward. The use of the method in many applications has proved its effectiveness and robustness. However, the GJK algorithm cannot provide the exact overlap distance when the contact of two objects is detected. Then the so-called Expending Polytope Algorithm (EPA) [18] can be utilised to fulfil the task. For polygons/polyhedra, both GJK and EPA only require support points that can be obtained from the vertices. No other topological properties are needed.

By recognising the fact that the contact based on the entrance block is exactly equivalent to the contact based on the Minkowski difference, both GJK and EPA should be an ideal algorithmic and implementation solution to Shi's 3D contact theory for DDA. In addition, it can be proved that the overlap based contact model where the overlap is computed from both algorithms fits in the energy conversed contact theory [10, 11], thereby offering a viable alternative contact model for polyhedra in DEM, although the related theoretical development is beyond the scope of this paper.

This work aims to highlight the equivalence between the entrance block and the standard Minkowski difference of two polygons/polyhedra, and to introduce the key points of both GJK and EPA for the effective implementation for convex shapes in general, and polygons/polyhedra in particular. Furthermore, the applicability of both GJK and EPA to DEM contact modelling within the energy based contact framework will be demonstrated through examples.

The paper is organised as follows: The Minkowski sum and difference of two geometric objects are reviewed in the next section, together with the introduction of basic properties for convex objects. Particular reference is made to the connection between the Minkowski difference and the entrance block in Shi's contact theory. The support point of a shape, which plays a leading role in the GJK and EPA, is also described. Section 3 introduces the GJK algorithm for detecting the contact state of two convex shapes. The computation of contact features of two contacting polygons/polyhedra is introduced in Section 4 by using the EPA. Some uniqueness and ambiguity issues are briefly discussed. Numerical examples are presented in Section 5 to demonstrate the uniqueness issue and further assess the effectiveness of the algorithms for 3D problems involving polyhedra. Conclusions are drawn in Section 6.

2 Minkowski Sum/Difference and Contact of Two Shapes

Consider two geometric shapes, A and B, in Euclidean space $\mathbb{E}^n (n = 2, 3)$ with boundaries ∂A and ∂B respectively. Their contact state can be classified as

$$\text{Contact State:} \begin{cases} \text{separation} & \equiv \quad A \cap B = \emptyset \\ \text{touch} & \equiv \quad A \cap B = \partial A \cap \partial B \\ \text{overlap} & = \quad A \cap B \neq \emptyset \end{cases}$$

The objective of the contact detection is to determine the contact state of two shapes statically or dynamically, and compute some contact features if required. This is a pure geometric problem, but different solution strategies may result in a significantly different computational efficiency.

2.1 Minkowski sum and difference

A shape can be viewed as a set of points in the shape. The Minkowski sum of the shapes A and B is a mathematical set operation that produces a new set or geometric shape, denoted as $A \oplus B$, by adding each point in A to each point in B:

$$A \oplus B = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A; \mathbf{b} \in B\} \tag{1}$$

The Minkowski difference of A and B, denoted as $A \ominus B$, is defined here to be the Minkowski sum of A and $(-B)$:

$$A \ominus B = \{\mathbf{a} - \mathbf{b} : \forall \mathbf{a} \in A; \forall \mathbf{b} \in B\} \tag{2}$$

where $(-B)$ can be viewed as the origin reflected image of B:

$$-B = \{-\mathbf{b}; \mathbf{b} \in B\}$$

Note that there are different definitions for Minkowski difference in the literature. The boundary of $A \ominus B$ is denoted as $\partial(A \ominus B)$. In robotic path planning, the Minkowski difference is also termed the (translational) configuration space obstacle as A represents an obstacle and B is a moving robot in translational motion.

Define the translation of a shape A by a vector $\mathbf{v}$ in real space $\mathbb{R}^n$ using Minkowski sum as

$$A^{\mathbf{v}} \equiv A \oplus \{\mathbf{v}\} = \{\mathbf{a} + \mathbf{v} : \mathbf{a} \in A\} \tag{3}$$

Some basic properties of Minkowski sum and Minkowski difference can be found elsewhere. For convex shapes, additional properties hold. It can be proved that the Minkowski sum or difference of two convex objects is also convex.

In particular, the Minkowski sum of two convex polygons/polyhedra is a convex polygon/polyhedron. In addition, the boundary of $A \oplus B$ or $A \ominus B$ is the convex hull of the boundary/vertex sum of A and B or A and $-B$. This property provides an effective approach to explicitly construct the Minkowski sum or difference of two convex polygons/polyhedra if needed, as many effective convex hull algorithms exist. See for instance the quick hull algorithm [19], or a simpler optimal output-sensitive algorithm [20]. Another explicit construction algorithm can be found in [21]. Furthermore, every boundary geometric feature (vertex, edge or face for 3D) of $A \ominus B$ is formed by two boundary features, one from each shape.

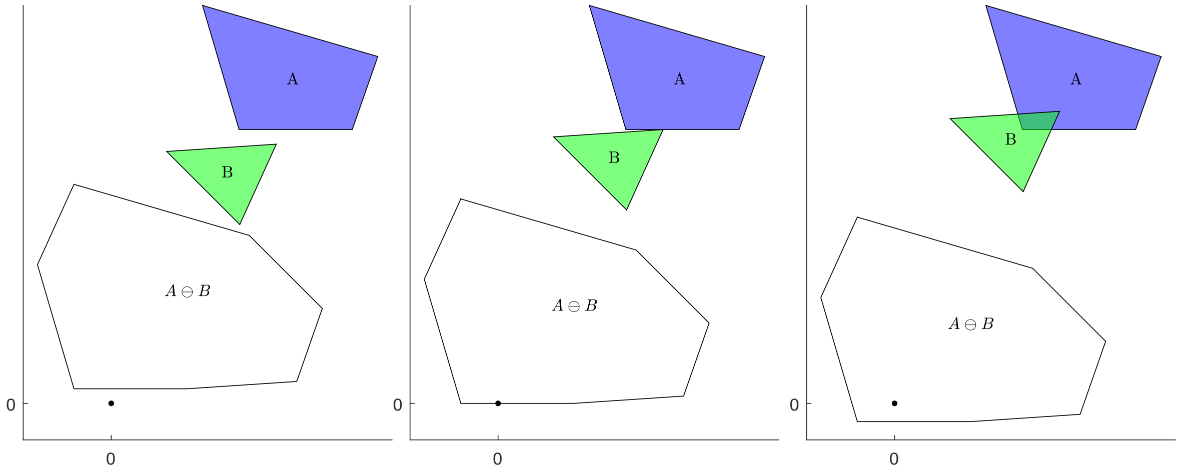
For general non-convex objects, when they can be decomposed into convex components, their Minkowski sum or difference can be obtained as the union of individual sums or differences of the convex components.

2.2 Minkowski difference and contact state

When A and B are in contact (in touch or overlap/penetration), there must exist at least one point $\mathbf{p}$ that is shared by the two objects such that $\mathbf{p} \in A; \mathbf{p} \in B$. Hence the origin $\mathbf{o}$ must be enclosed in the Minkowski difference $A \ominus B$,

$$\mathbf{o} \in A \ominus B, \qquad \text{if } A \cap B \neq \emptyset \tag{4}$$

Consequently, the contact state of two objects A and B is equivalent to the following statement:



(a) Separation (b) Touch (c) Overlap

Figure 1: Minkowski difference of quadrilateral A and triangle B and their contact states

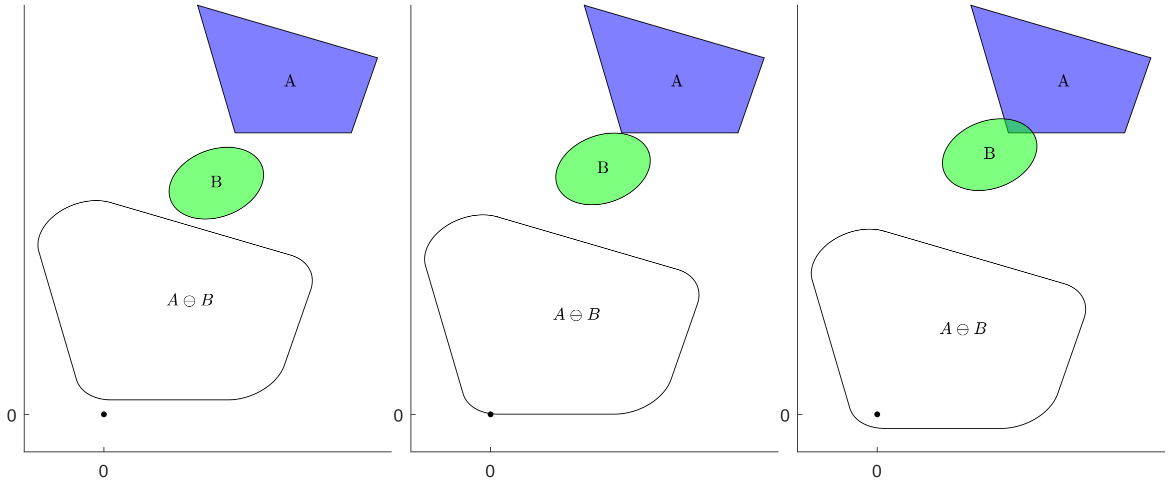
$$\text{Contact State:} \quad \begin{cases} \text{separation} & \equiv \quad \mathbf{o} \notin A \ominus B \\ \text{touch} & \equiv \quad \mathbf{o} \in \partial(A \ominus B) \\ \text{overlap} & \equiv \quad \mathbf{o} \in A \ominus B \end{cases}$$

i.e. the contact state of two objects is now reduced to check if the origin is either outside, on, or inside their Minkowski difference, which is a point inclusion test. For some contact detection problems, this contact state check is sufficient. However, for other applications such as in the FEM, DEM and DDA, additional contact features may also be required, among which the contact distance is the most important.

Assuming that A is fixed, the vector distance between A and B is defined as the minimum translational vector applied to B such that A and B just touch, i.e

$$\mathbf{d} = \underset{\mathbf{v} \in \mathbb{R}^n}{\operatorname{argmin}} \{ \|\mathbf{v}\| : \mathbf{o} \in \partial(A \ominus B^{\mathbf{v}}) \} \tag{5}$$

The vector distance $\mathbf{d}$ is also the minimum vector distance from the origin $\mathbf{o}$ to the boundary



(a) Separation (b) Touch (c) Overlap

Figure 2: Minkowski difference of quadrilateral A and ellipse B and their contact states

of $A \ominus B$:

$$\mathbf{d} = \underset{\mathbf{v}\in\mathbb{R}^n}{\operatorname{argmin}}\{\|\mathbf{v}\| : \mathbf{v} \in \partial(A \ominus B)\} \tag{6}$$

Express the vector distance $\mathbf{d}$ as

$$\mathbf{d} = d\mathbf{n}$$

where $d = \|\mathbf{d}\|$ is the scalar distance which is normally called the gap when the shapes are in separation, or the overlap when they are in penetration; $\mathbf{n}$ is a unit vector called the contact normal, another important contact feature required in the DEM. When the two shapes are in touch, $d = 0$.

Figures 1 and 2 illustrate the Minkowski differences of a fixed quadrilateral (A) with a triangle (B) and an ellipse (B) respectively, with each B having three different vertical positions. These figures clearly demonstrate that the relationship between the Minkowski difference and the origin is equivalent to the contact state of the two shapes. When B is moving in translation only, the shape and orientation of the Minkowski difference remain unchanged, while the position is changing with the same distance but in the opposite direction.

2.3 Minkowski difference and entrance block in Shi's contact theory

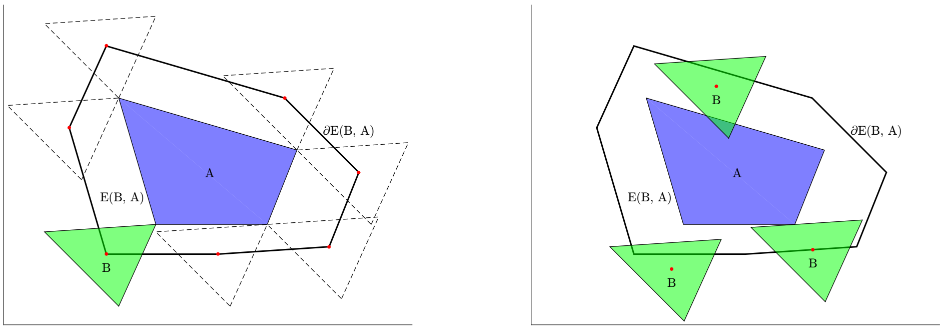
In Shi's contact theory for DDA [12, 13], the entrance block of two objects A and B is defined as

$$E(A, B) = \{\mathbf{b} - \mathbf{a} + \mathbf{v} : \mathbf{a} \in A, \mathbf{b} \in B\} \tag{7}$$

where $\mathbf{v}$ is a reference point on A that can be chosen arbitrarily and is suggested to be the centroid of A. Then the contact state between A and B can be reduced to check the distance between this reference point $\mathbf{v}$ and the boundary of $E(A, B)$, $\partial E(A, B)$.

Clearly, the entrance black $E(A, B)$ is essentially the same as $B \ominus A$ when choosing the reference point $\mathbf{v} = \mathbf{o}$. As different choices of the reference point do not alter the shape of $E(A, B)$, so $E(A, B) \cong B \ominus A$. Furthermore, the (vector) distance between $\mathbf{v}$ and $E(A, B)$ is exactly the same as the distance between $\mathbf{o}$ and $B \ominus A$. Therefore choosing $\mathbf{v} \neq \mathbf{o}$ provides no computational advantage over the standard Minkowski difference, but may offer better geometric visualisation of the difference.

The definition of Minkowski difference (2) offers no intuitive geometric visualisation of the Minkowski difference of two objects. A much simpler way of constructing $B \ominus A$ or the



(a) Entrance block

(b) Contact states

Figure 3: (a) Entrance block of quadrilateral A and triangle B constructed by sliding B along ∂A and taking the centroid of B as the reference point; (b) Three contact states between A and B

entrance block $E(A,B)$ is to fix B and slide A, i.e. in translate motion only, along the boundary of B in a full cycle so that A is always in touch with B. Then the complete trajectory of any (reference) point on A forms the boundary of $E(A,B)$, which is also the boundary of $B \ominus A$ when further setting the reference point to be the origin of the coordinate system.

Taking the two shapes in Figure 1 as an example, Figure 3 (a) shows the entrance block $E(B,A)$ and its boundary formed by sliding the triangle B against the boundary of the quadrilateral A where the centroid of B is chosen to be the reference point. It clearly indicates that $E(B,A)$ and $A \ominus B$ are identical in both shape and orientation. Figure 3 (b) further demonstrates that the three contact states coincide with the reference point being outside, on, or inside $\partial E(B,A)$.

For the above reason, the standard Minkowski difference $A \ominus B$, or $E(A,B)$ with $\mathbf{v} = \mathbf{o}$ will be assumed in the subsequence discussion. No further explicit reference to the entrance block will be made. The resulting algorithms and conclusions are, however, equally applicable to entrance blocks with $\mathbf{v} \neq 0$.

2.4 Support Point

In order to efficiently determine if the Minkowski difference of two shapes contains the origin, many contact detection algorithms make use of a mathematical operation called the support function or support mapping that obtains the support point as the furthest point of a shape along a given direction.

Denote $\mathbf{p}_s(A,\mathbf{v})$ as the support point of the shape A along the search direction $\mathbf{v} \neq 0$ as

$$\mathbf{p}_s(A,\mathbf{v}) = \mathbf{p} \in A: \ \mathbf{p} \cdot \mathbf{v} \geq \mathbf{q} \cdot \mathbf{v}, \ \forall \mathbf{q} \in A \tag{8}$$

where $\forall q \in A$ means for all the points in the shape A. Note that $\mathbf{v}$ does not need to be a unit vector. Since $\mathbf{q} \cdot \mathbf{v}$ can be viewed as the (projected) coordinate of the point $\mathbf{q}$ in the search direction $\mathbf{v}$, the support point is thus the point in A with the maximum projected coordinate along the direction.

For the reflected shape of A about the origin, the support function is as follows:

$$\mathbf{p}_s(-A,\mathbf{v}) = -\mathbf{p}_s(A,-\mathbf{v}) \tag{9}$$

The additive nature of the Minkowski sum leads to the fact that the support point for the Minkowski sum of A and B can be expressed as the addition of the support points of the two shapes:

$$\mathbf{p}_s(A \oplus B,\mathbf{v}) = \mathbf{p}_s(A,\mathbf{v}) + \mathbf{p}_s(B,\mathbf{v}) \tag{10}$$

Similarly, the support point for the Minkowski difference of A and B by utilising (9) is:

$$\mathbf{p}_s(A \ominus B,\mathbf{v}) = \mathbf{p}_s(A,\mathbf{v}) - \mathbf{p}_s(B,-\mathbf{v}) \tag{11}$$

The support point of the Minkowski difference $A \ominus B$ has the following properties:

1. Any support point of $A \ominus B$ is on (the convex hull of) the boundary $\partial(A \ominus B)$.

2. A and B are in separation or not in contact if

$$\mathbf{v} \cdot \mathbf{p}_s(A \ominus B,\mathbf{v}) = \mathbf{v} \cdot \mathbf{p}_s(A,\mathbf{v}) - \mathbf{v} \cdot \mathbf{p}_s(B,-\mathbf{v}) < 0 \tag{12}$$

   which geometrically states that the origin is further than any point on $\partial(A \ominus B)$ along the direction, so must be outside $A \ominus B$.

The significance of both (10) and (11) is that the non-explicit construction of $A \oplus B$ or $B \ominus A$ is necessary when computing the support point. Therefore as long as a support function can be defined for any (composite/complex) shape, the support point for the Minkowski sum or Minkowski difference of two such shapes can be obtained without first explicitly generating the sum or difference which can be very complicated and tedious, if not impossible.

The support function plays an utmost role in both GJK and EPA introduced in the next two sections for contact detection between two convex shapes.

Note that the support point of a shape for a particular direction may not be unique. For instance, any point on an edge or even on a face can be the support point when the whole edge or face is the furtherest.

3 The GJK Algorithm for Contact Between Convex Shapes

The GJK algorithm is an iterative procedure that attempts to dynamically find a simplex (triangle in 2D and tetrahedron in 3D) from the boundary or vertices of $\partial(A \ominus B)$ that contains the origin, or reports a non-contact case when such a state is detected. The outline of the algorithm is presented in Box 1.

The algorithm returns a simplex which is empty when the two shapes are not in contact, and is non-empty otherwise. The non-empty simplex can be used as the input for the EPA to be presented in the next section to compute other contact features between the two shapes, if needed.

The non-contact state check in step 2.b is based on the second property of the support point (12). Step 2.c is the key to the algorithm. The relevant algorithmic detail can be found elsewhere [17, 22, 23] as no attempt is made to cover the algorithmic aspect of the method in this paper. An effective implementation of this step, however, requires a systematic approach that can reduce the cases to be examined for the inclusion check of the origin to

Box 1: The GJK algorithm - Outline

1. Initialisation
    - a. Choose an (arbitrary) initial search direction $\mathbf{v} \neq 0$
    - b. Set Simplex $\mathbf{S}$ to be empty: $\mathbf{S} = \emptyset$
2. Iteration
    - a. Compute the support point $\mathbf{a} = \mathbf{p}_s(A \ominus B, \mathbf{v})$
    - b. If $\mathbf{a} \cdot \mathbf{v} < 0$, **return** $\mathbf{S} = \emptyset$ (NO CONTACT)
    - c. Update Simplex $\mathbf{S}$
        - i. Add $\mathbf{a}$ to $\mathbf{S}$ as a new vertex
        - ii. If $\mathbf{S}$ contains the origin $\mathbf{o}$, **return** $\mathbf{S}$ (CONTACT);
        - iii. Reduce $\mathbf{S}$ to the lowest dimension possible that still contains the closest feature to the origin by discarding vertices.
        - iv. Compute a new search direction $\mathbf{v}$ that is towards the origin normal to the new simplex; and go to 2.a

a minimum, and voids any unnecessary numerical operations. There are several (slightly) different implementations available in the literature or online sources.

The GJK algorithm can, in principle, be applied to any two convex shapes as long as the support function of each shape can be numerically defined to compute the furthest boundary point along a given direction. Thus the GJK algorithm offers greater versatility than other similar approaches mentioned in the introduction.

When applied to convex polygons/polyhedra, one of the most favourable features of the GJK is that the method only requires support points that can be obtained from the vertices of the polygons/polyhedra. No other topological properties, such as edges and faces, are needed. Thus the support function for a polygon/polyhedron can be implemented straightway. The complexity of computing one support point is proportional to the sum of the numbers of vertices of the two shapes concerned. A higher efficiency may be achieved, however, if the topological properties are also utilised to find the support point of polygon/polyhedron [22], particularly when the number of vertices is large.

4 Expanding Polytope Algorithm for Computing Contact Features

The GJK algorithm outlined above can determine whether the two convex shapes are in contact or not, but cannot provide further geometric features of the contact required by the DEM or DDA. The Enhanced GJK algorithm [22] may be used to compute the overlap, but only in an approximate fashion though. Therefore new approaches must to be found to address the issue. The Expanding Polytope Algorithm, EPA, is such an effective approach.

Although the vertices of the output simplex from the GJK are on the boundary (vertices) of the Minkowski difference of the two convex shapes, the edges and faces of the simplex are not necessarily located on the boundary. Thus, the shortest distance of the origin to the faces of the simplex is the lower bound of the actual overlap, unless the face closest to the origin is also a face of the Minkowski difference for polygon/polyhedron, or in some special degenerated cases.

The main idea of the EPA is to iteratively expand the initial simplex to form a polytope (polygon or polyhedron) until the face of this expanded polytope closest to the origin is also a boundary face of the Minkowski difference of the polygon/polyhedron, or within a prescribed tolerance to the actual boundary for general convex shapes.

EPA uses the output simplex $\mathbf{S}$ from the previous GJK step as the initial polytope $\mathbf{P}$ and expands it if $\mathbf{S}$ is a degenerated simplex. This can be achieved by adding additional support points similar to the procedure adopted in the GJK algorithm.

The outline of the EPA for general convex objects is presented in Box 2. The two key steps involved are discussed in detail in the next two subsections. Some uniqueness and ambiguity issues associated with the algorithm are also discussed in Section 4.3.

4.1 Polytope expansion and iteration termination

In Step 2, the face closest to the origin in the polytope $\mathbf{P}$ is found and its outward normal $\mathbf{n}_f$ is used as the search direction to compute a new support point on $\partial(A \ominus B)$. The corresponding distance of the face to the origin is denoted as d_f. Next, check if this new point will increase the distance from $\mathbf{o}$ to the simplex faces. If not, the shortest distance or overlap is now found, Box 2: The EPA - Outline

1. Initialisation

   a. Take the simplex $\mathbf{S}$ returned from the GLK algorithm as the start polytope $\mathbf{P} = \mathbf{S}$.

   b. Expand $\mathbf{P}$ to be a full simplex by adding new vertices if $\mathbf{S}$ is a degenerated simplex.

   c. Compute the distance d_i of each face $i \in \mathbf{P}$ to the origin.

2. Iteration

   a. Find the closest face $f \in \mathbf{P}$ to the origin $\mathbf{o}$, and set the search direction to be $\mathbf{n}_f$, where $\mathbf{n}_f$ is the outward normal of the closest face f. Let d_f be the distance of the face f to the origin.

   b. Compute the support point $\mathbf{a} = \mathbf{p}_s(A \ominus B, \mathbf{n}_f)$

   c. Compute the distance between $\mathbf{a}$ and the origin $\mathbf{o}$ along the direction $\mathbf{n}_f$: $d_a = \mathbf{n}_f \cdot \mathbf{a}$

   d. Check for convergence: if $d_a - d_f < \epsilon$ (given tolerance), **go to 3**

   e. Expand $\mathbf{P}$:
   For each face $f \in \mathbf{P}$, if $\mathbf{n}_f \cdot (\mathbf{a} - \mathbf{v}_f) > 0$ (where $\mathbf{v}_f$ is a vertex of the face)

      i. Remove the face f from $\mathbf{P}$

      ii. Add new faces formed by the vertices of the face and the new point $\mathbf{a}$ to $\mathbf{P}$; and also calculate the distance of each new face to the origin.

   f. Go back to 2.a

3. Compute Contact Features

   a. Find the projected point $\mathbf{c}$ of the origin $\mathbf{o}$ on the closest face f.

   b. Set Contact overlap $d_c = d_f$, and compute Contact normal $\mathbf{n}_c = \mathbf{c}/d_c$

   b. Compute Contact points on the two shapes

      i. Compute the local or barycentric coordinates $\boldsymbol{\xi}_c$ of $\mathbf{c}$ in the face using its vertices.

      ii. Assume the same local $\boldsymbol{\xi}_c$ on the corresponding face of each shape to compute the local point contact $\mathbf{c}_k, (k = A, B)$

and the iteration is terminated.

For polygon/polyhedron cases, the shortest distance will be obtained in a finite number of iterations when the new point found is one of the vertices on the face, i.e. the closest face of the simplex $\mathbf{S}$ is also a face of $\partial(A \ominus B)$, thus $d_a = d_f$. For general convex shapes, the new point may never be an existing vertex of $\mathbf{P}$, so a pre-defined tolerance ϵ has to be introduced to control the termination of the iteration.

The key implementation issue in this step is how to expand the polytope. It is achieved by removing any face in $\mathbf{P}$ that is *visible* from the new point $\mathbf{a}$ and then adding new faces formed by the point and the vertices of the face. A face with outward normal $\mathbf{n}_f$ is visible to a point $\mathbf{a}$ if the point is on the positive side of the face: $\mathbf{n}_f \cdot (\mathbf{a} - \mathbf{v}_f) > 0$ (where $\mathbf{v}_f$ is any vertex or point on the face).

4.2 Compute contact features

When the iteration is terminated in the previous step, the closest face f on $\mathbf{P}$ to the origin $\mathbf{o}$ is determined together with the (approximate) shortest distance d_f which is also the scalar distance or overlap $d_c = d_f$. Other contact features are calculated in Step 3, including the contact normal $\mathbf{n}_c$, contact point pair $\mathbf{c}_k(k = A, B)$ on the two shapes, and contact type if applicable.

4.2.1 Contact points

The key to compute these contact features is to find the closest point $\mathbf{c}$ on f to the origin. As the closest face has been identified in the previous step, the point $\mathbf{c}$ is just the projected point of the origin onto the face. Its local or barycentric coordinates $\boldsymbol{\xi}_c$ on the face are obtained using the vertices of the face. In the 2D case, the face is a segment, and $\boldsymbol{\xi}_c = (\xi_1, \xi_2)$(with $\xi_1 + \xi_2 = 1$); in the 3D case, the face is a triangle, and $\boldsymbol{\xi}_c = (\xi_1, \xi_2, \xi_3)$ are the barycentric coordinates.

Let $\mathbf{v}_f^i(i = 1, 2$ for 2D or $1, 2, 3$ for 3D) be the vertices of the face f, then the point $\mathbf{c}$ can be expressed as the linear combination of these vertices using the local or barycentric coordinates

$$\mathbf{c} = \begin{cases} \xi_1 \mathbf{v}_f^1 + \xi_2 \mathbf{v}_f^2; & \text{(2D case)} \\ \xi_1 \mathbf{v}_f^1 + \xi_2 \mathbf{v}_f^2 + \xi_2 \mathbf{v}_f^2; & \text{(3D case)} \end{cases} \tag{13}$$

By virtue of the definition (11), each support point $\mathbf{p}_s$ on $\partial(A \ominus B)$ is associated with two support points $\mathbf{p}_A$ and $\mathbf{p}_B$ on the boundaries ∂A and ∂B respectively by

$$\mathbf{p}_s = \mathbf{p}_A - \mathbf{p}_B$$

So the vertices $\mathbf{v}_f^i$ of the closest face f have their *associated* counterparts on ∂A and ∂B. The same is applicable to $\mathbf{c}$, i.e. there also exist two associated points $\mathbf{c}_A$ and $\mathbf{c}_B$, which are called the *contact points*, in A and B (but may not be on ∂A and ∂B if the shapes are not polygon/polyhedron). Furthermore, the local/barycentric coordinates of these two associated points are assumed to be the same as $\boldsymbol{\xi}_c$.

Let $\mathbf{v}_A^i \in \partial A$ and $\mathbf{v}_B^i \in \partial B$ be the two associated vertices of $\mathbf{v}_f^i \in \partial(A \ominus B)$. The two associated contact points $\mathbf{c}_A$ and $\mathbf{c}_B$ can be computed as

$$\mathbf{c}_{A,B} = \begin{cases} \xi_1 \mathbf{v}_{A,B}^1 + \xi_2 \mathbf{v}_{A,B}^2; & \text{(2D case)} \\ \xi_1 \mathbf{v}_{A,B}^1 + \xi_2 \mathbf{v}_{A,B}^2 + \xi_2 \mathbf{v}_{A,B}^2; & \text{(3D case)} \end{cases} \tag{14}$$

from which the contact distance $\mathbf{c}$, overlap d_c and contact normal $\mathbf{n}_c$ can be recovered by

$$\mathbf{c} = \mathbf{c}_A - \mathbf{c}_B = d\mathbf{n} \tag{15}$$

In DEM formulations, one single or common contact point $\mathbf{c}_p$ for both A and B in contact may be needed. In this case, a certain combination of $\mathbf{c}_A$ and $\mathbf{c}_B$ may be sufficient. For instance, $\mathbf{c}_p = (\mathbf{c}_A + \mathbf{c}_B)/2$.

4.2.2 Contact type for polygon/polyhedron shapes

For polygonal/polyhedral shapes, each face on $\partial(A \ominus B)$ has two associated faces on ∂A and ∂B. In what follows, the closest face f is termed the *(master) contact face*, and its two associated faces are the *associated contact faces.* Based on the geometric nature of these two faces, i.e. if they are a vertex, edge or face, the *contact type* of the two contacting shapes can be established.

Since several support points on $\partial(A \ominus B)$ may be associated with the same point on ∂A or ∂B, it is possible that for a triangular face on $\partial(A \ominus B)$, the associated face on A or B may be degenerated into an edge or even a single vertex. By considering all possible degenerated cases of the two associated faces where one or more vertices are identical, a contact between A and B can be classified into one of the following possible contact types (where 'v' stands for vertex, 'e' for edge and 'f' for face):

$$\text{Contact Types: } \begin{cases} \text{v-v} \quad \text{v-e} \quad \text{e-e} & \text{(2D)} \\ \text{v-v} \quad \text{v-e} \quad \text{e-e} \quad \text{v-f} \quad \text{e-f} \quad \text{f-f} & \text{(3D)} \end{cases}$$

However, since $A \ominus B$ is convex and contains the origin, the 'v-v' type cannot occur unless A and B are just in touch at a pair of vertices. Similarly the 'v-e' type in 3D can only appear in a touch contact. Also, the 'e-e' type for 2D and the 'f-f' type for 3D are possible only for special cases when the two edges or faces are parallel. Thus the common contact types are reduced to only one type 'v-e' for 2D, and three types 'e-e', 'v-f' and 'e-f' for 3D.

4.3 Uniqueness and Ambiguity Issues

The EPA may face some uniqueness and ambiguity issues in some special cases where either the vector contact distance $\mathbf{d}_c$ may not be unique, or the contact points $\mathbf{c}_A$ and/or $\mathbf{c}_B$ may have multiple values.

A uniqueness issue may occur in Step 2.i when the origin has an equal (shortest) distance to multiple faces of the simplex. This alone may not impose any difficulty on the outcome of the EPA if the actual contact is unique in the sense that only one possible contact normal direction exists. However, there are cases where multiple points on $\partial(A \ominus B)$ may have the same minimum distance with the origin, leading to possible multiple contact points and directions. For instance, in polygon/polygon or polyhedron/polyhedron contact in FEM, DEM and DDA, there often have a large number of so called "conner-conner" contact cases where the origin may have the equal distance to two edges (2D) or multiple faces (3D) of the Minkowski difference.

For these cases, the EPA will obtain only one solution as expected, depending (randomly) on numerical round-off errors and also on algorithmic implementation details. This is acceptable in theory. In practice, this situation may not be hold in the next iteration/time step due to the motion of A and/or B.

The real issue in numerical simulations for these cases is that the contact features between the objects concerned may be subjected to a non-smooth switch between two or more possible contact solutions - causing convergence difficulties for FEM and DDA or introducing fictitious energy in DEM. Often, smaller time steps may have to be adopted to ensure solution accuracy and/or numerical stability.

Thus, this particular uniqueness issue stems not from the EPA methodology itself, but from Minkowski difference based contact theories. In DEM, a contact area or volume based contact theory can resolve the issue [10, 11].

Another ambiguity scenario arises when the contact point $\mathbf{c}$ on $\partial(A \ominus B)$ is uniquely determined, and so are the contact overlap and normal direction, but the associated contact points on the two contacting shapes may not be unique. This situation appears when the contact occurs not on a point but along a segment or on a surface area.

For the contact of polygons/polyhedra, this occurs when the contact type is 'e-e' for 2D and 'f-f' for 3D where the pair of edges or faces involved are in parallel. In technical or computational terms, this situation mainly occurs when the support point of A or B (or both) along the final contact normal direction is not unique, as explained in Section 2.4. In this case, although the local coordinates $\boldsymbol{\xi}_c$ can also be uniquely determined, the computed contact point $\mathbf{c}_A$ or $\mathbf{c}_B$ (or both) are obtained in a rather arbitrary manner due to the ambiguity of the associated face vertices of A or B used in (14). In consequence, the two associated contact points, or contact type, may undergo a large shift between consecutive iteration/time steps which leads to similar adverse effects as the previous case. A solution to this problem is to define the centroid of the two overlapping edges/faces as the contact point.

5 Numerical Examples

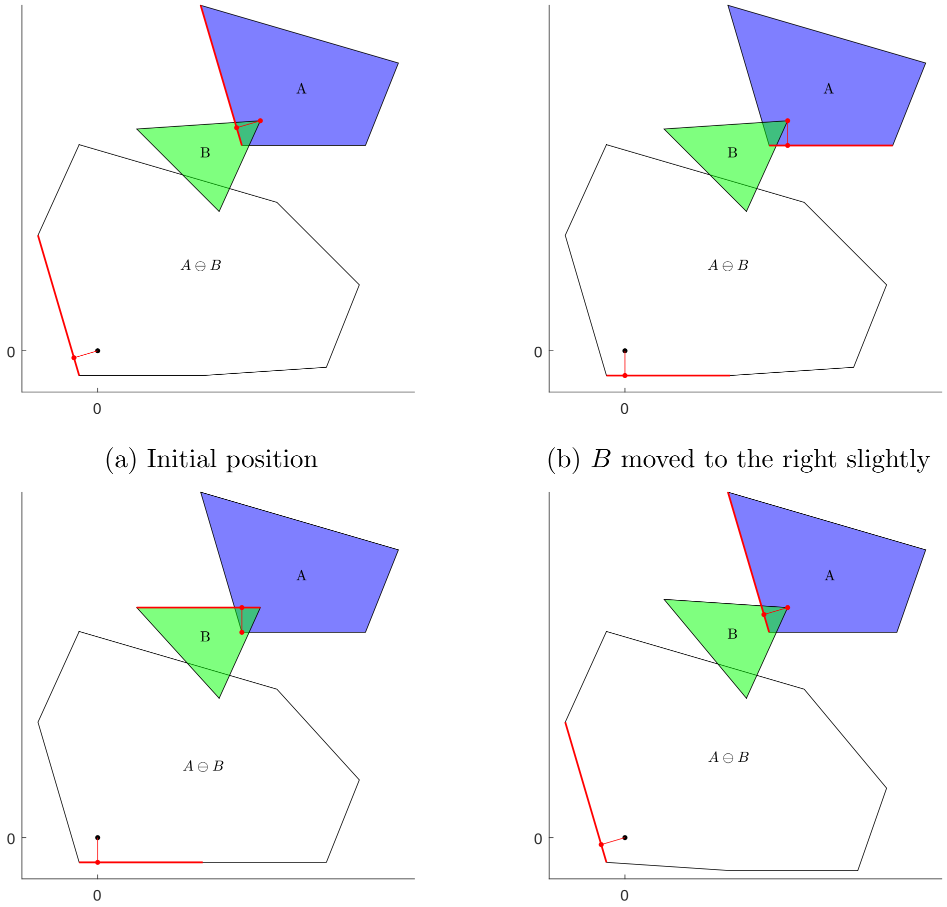
Several numerical examples will be presented in this section to illustrate a number of issues discussed above and further assess and demonstrate the efficiency of both GJK and EPA methods in terms of number of updates/iterations required to converge and terminates the procedures.

5.1 Contact feature evolution and uniqueness issues

Firstly, consider the contact between a fixed quadrilateral A and a moving triangle B at four configurations as shown in Figure 4. For each configuration, both GJK and GPA algorithms are applied to compute all the contact features. The two contact faces, one on $\partial(A \ominus B)$ and the other on A or B, together with the three contact points, one on $\partial(A \ominus B)$, and one on ∂A and ∂B each, are highlighted (in RED) in the figure. The corresponding vector contact distances are also drawn.

All four contacts have the 'v-e' contact type. In the first case (a), A and B are located in such a configuration that the origin is on the verge of having an equal distance to the two edges of $\partial(A \ominus B)$. When B is moved slightly to the right in the second case (b), the contact face on A is switched from the left to the bottom edge. Next, in the third case (c), B is rotated clockwise to the position so that its top edge is parallel to the bottom edge of A, but the contact points and direction completely change. Note that in this configuration, the actual contact should be a 'f-f' type and the contact points should be along the whole contact segment. The one shown in the figure may be considered as a possible option, but by slightly changing the length of the top edge of B or the bottom edge of A, for instance, the contact pair 'v' and 'e' can change back to the case (b), indicating the existence of the uniqueness issue discussed in Section 4.3. When B is further rotated clockwise in the case (d), the contact feature changes back to the case (a). Thus, slightly alerting the position of B result in total 3 different contact feature changes.

This example clearly highlights the non-smoothing evolution nature of contact features for polygon/polyhedron contacts. The uniqueness issue for the special 'f-f' contact type in the case (c) is also covered.



(a) Initial position

(b) B moved to the right slightly

(c) B rotated clockwise (parallel sides)

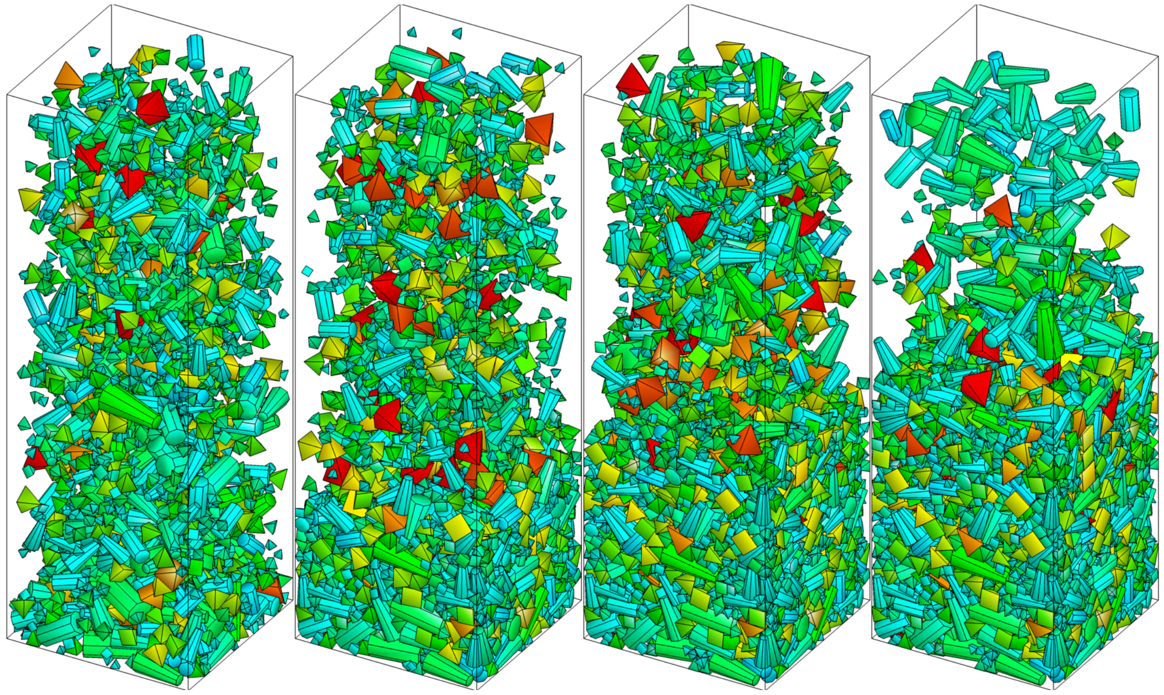
(c) B rotated clockwise further

Figure 4: Contact features between quadrilateral A and triangle B at four slightly altered configurations

5.2 Performance Assessments of GJK and EPA

The performance of the GJK for the contact detection and the GPA for computing the contact features has been tested previously [17, 22, 23]. Further assessments are conducted below by two 3D simulations. The performance of GJK is measured by the number of updates required to either find a non-contact case, or construct a simplex that contains the origin, while that of EPA is measured by the number of iterations needed to obtain the shortest distance (and all the related contact features).

Note that the simulations are conducted in the context of DEM where the overlap based linear contact model is used within the framework of the energy based general contact theory [5, 11]. The overlap is evaluated by both GJK and EPA and the corresponding linear model is properly derived although the theoretical development in this aspect is omitted. Because the focus of the following numerical simulations is on the performance of GJK and EPA, the conclusions to be drawn should be equally applicable to both DEM and DDA.



(a) Stage 1 (b) Stage 2 (c) Stage 3 (d) Stage 4

Figure 5: Snap shots of the 4-type polyhedral system at four stages - Simulation I

5.2.1 Simulation I: Polyhedron contact

In the first simulation, the system consists of three types of polyhedra: square pyramids (5 vertices, 5 faces and labelled as 'P5'), octagonal prisms (16 vertices, 10 faces, 'P16'), and octagonal frustums (16 vertices, 10 faces, 'F16'). These three types of polyhedra are randomly injected into a box-like region with different numbers and sizes. The injection takes place periodically over the whole course of the simulation which lasts 10s. The final number of polyhedra contained in the system is about 5000.

The system has a good mixture of polyhedra with different types and sizes. It is mainly loosely packed in the early stage of the simulation, meaning that there are not many contact pairs and contact overlap may be very small in general, but the system will increasingly become tightly packed under the action of gravity over the course of the simulation, and thus more contact pairs and larger overlap exist in the system.

The mechanical contact between polyhedra is modelled using a linear contact model in the context of DEM. Linear viscous contact damping and friction are applied to dissipate the energy. The GJK is employed to detect the contact between the polyhedra, followed by using the EPA to compute the contact features, based on which the contact forces are obtained within the framework of the energy based contact theory [11]. The snapshots of the system configuration at four stages/time instances of the simulation are displayed in Figure 5.

The average number of updates for the GJK and the average number of iterations for the EPA are recorded at some time instances. For the GJK, two numbers are calculated, one for non-contact cases, the other for contact cases. Also, the numbers for different combinations of the three types of polyhedra, in total of 6 contact groups, are also obtained individually. The results at the stages coinciding with the snapshot time instances in Figure 5 are presented in Table 1.

Table 1: GJK/EPA performance for different contact groups - Simulation I

Stage	GJK + EPA	Contact Group					
		P5-P5	P5-P16	P5-F16	P16-P16	P16-F16	F16-F16
1	GJK(0)	3.65	4.02	4.00	3.98	4.14	4.08
	GJK(1)	5.30	6.18	6.30	6.86	6.72	6.70
	EPA	1.29	1.69	1.61	1.94	2.18	1.91
2	GJK(0)	3.69	3.86	3.89	4.10	3.81	3.78
	GJK(1)	5.29	6.26	6.17	6.69	6.51	6.45
	EPA	1.54	1.84	1.84	2.34	2.32	2.33
3	GJK(0)	3.67	3.86	3.85	3.81	3.93	3.82
	GJK(1)	5.25	6.26	6.21	6.46	6.56	6.42
	EPA	1.56	1.90	1.90	2.41	2.28	2.23
4	GJK(0)	3.67	3.88	3.82	3.95	3.81	3.78
	GJK(1)	5.22	6.23	6.13	6.48	6.49	6.45
	EPA	1.62	1.97	1.97	2.48	2.45	2.41

Based on the results presented in the table the following observations may be made:

1. The average number of updates in the GJK to detect a non-contact case is around 4, while the average number for detecting a contact case is around 6 $\sim$ 7; both are almost independent of polyhedral shapes and sizes. For contact cases, however, the number is increased for polyhedra with more vertices/faces but only marginally.

2. The average number of iterations in the EAP to find the correct contact feature is only around 2. Such a low number may be associated with the face that in those contact cases, overlap is very small so the origin is very close to the boundary of the Minkowski difference and the closest face of the simplex output from the GJK is almost the real contact face. Compared to the GJK, the number of polyhedral vertices/faces has a slightly larger impact on the convergence of EPA.

3. It appears that when the polyhedra are packed more tightly, and therefore the overlap is generally getting larger, less GJK updates are required, while more EPA iterations are needed instead.

5.2.2 Simulation II: Polyhedron and sphere contact

The second simulation is almost the same as the previous one. The main difference is that the octagonal frustums are replaced by spheres in order to access the performance of the GJK for non-polyhedral shapes. Also, the number of polyhedra for each polyhedral type is different from the previous simulation. The total number of objects in the system is over 5500.

A sphere is the shape with the simplest support function and thus its contact with polyhedron can be handled by the GJK algorithm. Note, however, that although the contact overlap between sphere and polyhedron can also be evaluated by the EPA in principle, the convergence will be poor as compared to pure polygon/polyhedron cases, and is also influenced by the level of tolerance specified. For this reason, the performance of the EPA for sphere related contact groups will not be taken into account, and the contact feature computation between a sphere and a polyhedron is specially treated in a separate function when their contact is found by the GJK.

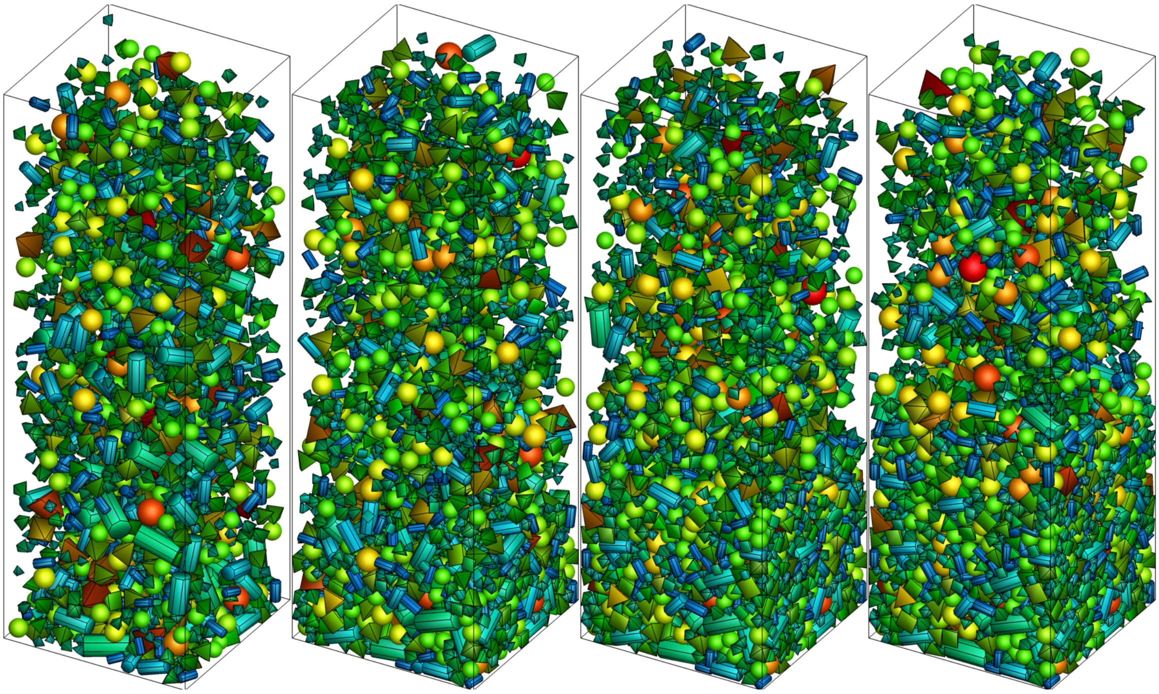

(a) Stage 1 (b) Stage 2 (c) Stage 3 (d) Stage 4

Figure 6: Snap shots of the 3-type polyhedra + sphere system at four stages - Simulation II

The sphere group is denoted as 'S'. There are five contact groups in this simulation as the sphere-sphere group is not included. The snapshots of the simulation at four stages or time instances are depicted in Figure 6. The GJK and EPA performance results at the same stages are given in Table 2.

The results for the polyhedral contacts in Table 2 are generally consistent with the previous simulation, and further confirms the observations made. The results related to the sphere group show that the average GJK updates for the no-contact case are slightly larger than

Table 2: GJK/EPA performance for different contact groups - Simulation II

stage	GJK + EPA	Contact Group				
		P5-P5	P5-P16	P16-P16	P5-S	P16-S
1	GJK(0)	3.51	3.84	3.81	4.05	4.26
	GJK(1)	5.40	6.43	6.79	11.2	10.6
	EPA	1.20	1.30	1.82	-	-
2	GJK(0)	3.85	3.81	3.93	4.38	4.47
	GJK(1)	5.23	6.13	6.55	9.90	9.89
	EPA	1.52	1.91	2.38	-	-
3	GJK(0)	3.65	3.85	3.93	4.56	4.65
	GJK(1)	5.27	6.25	6.67	9.70	9.65
	EPA	1.51	1.84	2.14	-	-
4	GJK(0)	3.65	3.80	3.89	4.46	4.59
	GJK(1)	5.27	6.25	6.59	9.62	9.53
	EPA	1.55	1.85	2.15	-	-

the non-sphere counterparts, while the updates for the contact case has a marked increase of over 50%. This may indicate that the GJK algorithm works better for polyhedra, which have a limited number of vertices and flat faces, than shapes with curved and smooth surfaces. However, it does not imply that the GJK for the sphere-polyhedron contact is inferior than other methods designed for such a case, although no attempt is made in the current work to address this issue.

6 Concluding Remarks

In this work, based on the definition of the standard Minkowski difference of two shapes, its equivalence to the entrance block defined in Shi's contact theory has been highlighted. Consequently, any existing Minkowski difference based contact detection method for polyhedra can be utilised as the computational approach to his 3D contact theory for DDA, and this also fits the energy based contact theory for DEM.

As an effective contact detection method for convex shapes based the concept of Minkowski difference, the GJK algorithm avoids the explicit construction of the Minkowski difference as long as the support functions of the shapes can be defined. This algorithm is generic, independent of the geometric shapes involved, simple to be implemented, and has proved to be effective, robust and versatile. The numerical simulations indicate that the average number of updates required to detect contact or no contact is around 6 for polyhedra. The performance of the GJK is only weakly affected by the number of vertices that polyhedra have, and may not be affected by polyhedral shapes and sizes. Note that the performance may be enhanced by taking advantage of temporal and geometric coherence that is typically presented in discrete/discontinuous problems.

The EPA is designed to be the complementary to the GJK for evaluating the overlap for a pair of contacting shapes detected by the GJK algorithm, and is also easily implemented and applicable to convex shapes. The numerical examples show that only 1 or 2 iterations are needed to find the overlap and all the relevant contact features for polyhedra, when their overlap is small which is the case for all problems solved by DEM and DDA. The performance is almost not affected by the number of vertices and shapes of polyhedra.

As tested in the simulations, both GJK and EPA are less effective for smooth surfaced shapes. Thus specially designed methods may need to be developed for these cases. Nevertheless the generic and simple nature of the two algorithms may still be a favourable option or viable alternative for more complicated shapes where the specially developed contact algorithms may be less robust, too sophisticated or costly.

Acknowledgment

The work was partially supported by the National Natural Science Foundation of China (Project No. 11772135) and the special fund from the Institute of Manufacturing Engineering of Huaqiao University, China. This support is greatly acknowledged.

References

[1] P. Wriggers. *Computational Contact Mechanics.* (2nd ed.), Springer-Verlag, Berlin, Germany, 2006.

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

[3] G. H. Shi. *Discontinuous Deformation Analysis - A New Model for the Statics and Dynamics of Block Systems.* Ph.D thesis, University of California, Berkeley, 1988.

[4] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM.* 18(9): 509, 1975.

[5] Y. T. Feng, D. R. J. Owen. An sugumented spatial digital tree algorithm for contact detection in computational mechanics. *Int. J. Numer. Meth. Engng.*, **55**: 159-176, 2002.

[6] A. Munjiza. *The Combined Finite-Discrete Element Method.* England, Wiley & Sons, 2004.

[7] J. R. Williams, E. Perkins, B. Cook. A contact algorithm for partitioning N arbitrary sized objects. *Engineering Computations.* 21: 215-234, 2004.

[8] K. Han, Y. T. Feng, D. R. J. Owen. Performance Comparisons of Tree Based and Cell Based Contact Detection Algorithms. *Engineering Computations*, 24(2):165-181, 2004.

[9] P. A. Cundall. Formulation of a three-dimensional distinct element model - Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics* 25:107-116, 1988.

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

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

[12] G. H. Shi. Basic equations of two and three dimensional contacts. Proceedings of the 47th US Rock Mechanics/Geo-mechanics Symposium, San Francisco, ARMA: 253-269, 2013.

[13] G. H. Shi. Basic theory of two dimensional and three dimensional contacts. Frontiers of Discontinuous Numerical Methods and Practical Simulation in Engineering and Disaster Prevention, Taylor & Francis Group, London, 2013.

[14] Y. Kim, M. Lin, D. Manocha. Chapter 39. Collision and proximity queries. *Handbook of Discrete and Computational Geometry: Geometric data structures and searching* (eds. J. E. Goodman, J. O'Rourke, C. Toth), 3rd edition, CRC Press, 2018.

[15] J. Cohen, M. Lin, D. Manocha, K. Ponamgi. I-COLLIDE: An Interactive and Exact Collision Detection System for Large Scale Environments. *ACM Int. 3D Graphics Conference.* pages 189-196, 1995.

[16] B. Mirtich V-clip: Fast and Robust Polyhedral Collision Detection. Mitsubishi Electric Research Laboratory TR-97-05, July 1997.

[17] E. G. Gilbert, D. W. Johnson and S. S. Keerthi. A Fast Procedure for Computing the Distance between Complex Objects in Three-Dimensional Space. *IEEE Trans. Robotics and Automation.* 4(2):193-203, 1988.

[18] G. van de Bergen. Proximity queries and penetration depth computation on 3d game objects. Game Developers Conference, 2001.

[19] C. B. Barber, D. P. Dobkin, H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*. 22(4): 469–483, 1996.

[20] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete and Computational Geometry*. 16:361–368, 1996.

[21] D. Halperin, E. Fogel Exact and Efficient Construction of Minkowski Sums of Convex Polyhedra with Applications. *Computer Aided Design*, 39(11): 929-940, 2007.

[22] C. Cameron. Enhancing GJK: Computing Minimum and Penetration Distances between Convex Polyhedra. *Int Conf Robotics and Automation*, pages 3112-3117, April 1997.

[23] G. van de Bergen. A fast and robust GJK implementation for collision detection of convex objects. Journal of Graphics Tools, 4(2):7-25, 1999.