

**A binary-tree element subdivision method for evaluation of  
nearly singular domain integrals with continuous or discontinuous kernel**

**Jianming Zhang<sup>1\*</sup>, Baotao Chi<sup>1</sup>, Krishna M. Singh<sup>2</sup>, YuDong Zhong<sup>1</sup>, Chuanming Ju<sup>1</sup>**

*<sup>1</sup>State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China*

*<sup>2</sup>Department of Mechanical and Industrial Engineering, Indian Institute of Technology, Roorkee, India*

\*Correspondence to: Jianming Zhang

College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China

Telephone: +86-731-88823061

E-mail: zhangjm@hnu.edu.cn

**Abstract**

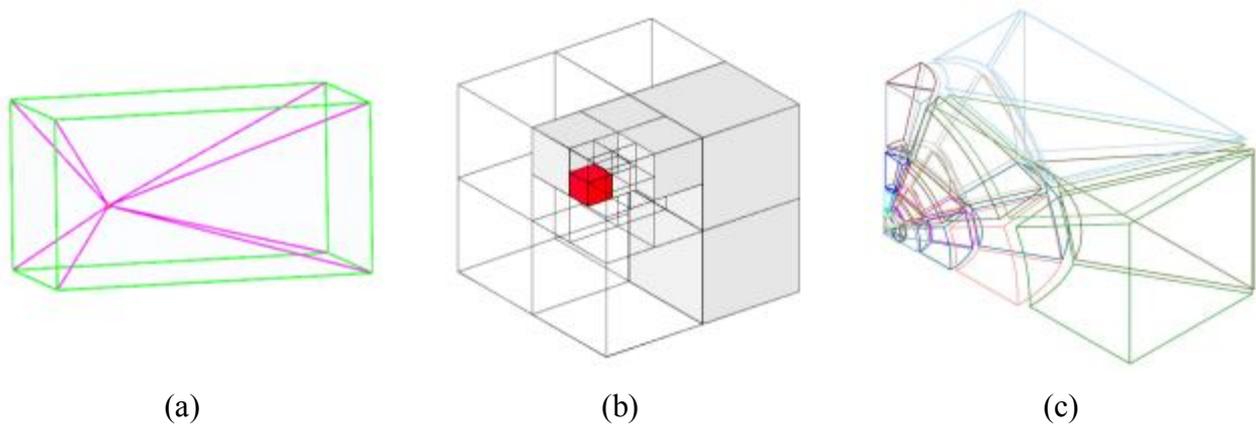
An adaptive and efficient volume element subdivision method using binary tree for evaluation of nearly singular domain integrals with continuous or discontinuous kernel in three-dimensional (3-D) boundary element method (BEM) has been presented. In the Conventional Subdivision Method (CSM) for evaluation of nearly singular integrals, the patches are obtained by simply connecting the source point with each vertex of the element. Thus, the accuracy of the integral obtained with CSM is easily affected by the shape of the element and the location of the source point. In contrast, the proposed Binary-Tree Subdivision Method (BTSM) is more convenient to implement and can guarantee successful patch generation under any circumstances for accurate evaluation of nearly singular domain integrals with continuous or discontinuous kernel. Numerical results for volume elements of arbitrary type with various relative locations of the source point demonstrate robustness, accuracy and efficiency of the proposed method.

**Keywords:** BEM, nearly singular integral, binary-tree, element subdivision, Gaussian quadrature, Hammer quadrature

**1. Introduction**

With the distinct feature that only the boundary is required to be discretized into elements for linear and exterior problems, the boundary element method (BEM) [1] has been developed rapidly to such a level that it can be widely applied to solve very complicated engineering problems, such as potential problems [2-4], elastostatic problems [5-7], contact problems [8], fracture mechanics problems [9] and acoustics problems [10,11]. Application of the BEM to a boundary value problem

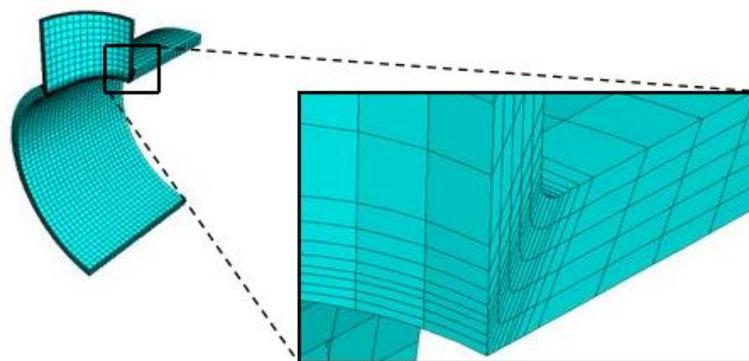
with body forces, time dependent effects or certain class of non-linearities generally leads to an integral equation which contains domain integrals [12]. Nearly singular integrals arise when the source point is close to but not inside or on the boundary of element. In BEM, nearly singular integrals come up in the following physical situations [13]: (1) Computing the interior quantities close to the boundary; (2) There are great differences among the sizes of adjoining element meshes; (3) For narrow and thin domains; (4) For non-linear problems in which the integrals in the domains near the boundary need to be calculated. It should be noted that, theoretically, the domain integrals are actually regular since the value of their integrand is finite. However, for evaluation of nearly singular integrals with continuous or discontinuous kernel, the value of the integrand varies dramatically if the field point is at some discontinuous point of the fundamental solution or the source point is close to the element of integral. Hence, it is of crucial importance to evaluate nearly singular domain integrals accurately and efficiently in BEM implementation.



**Fig. 1.** Various element subdivision methods: (a) conventional subdivision method, (b) quad-tree subdivision method, and (c) spherical element subdivision method.

The main approaches for evaluation of nearly singular domain integrals in BEM have been the use of element subdivision methods such as the Conventional Subdivision Method (CSM) [14] (Fig. 1(a)), the Quad-Tree Subdivision Method (QTSM) [15] (Fig. 1(b)), the Spherical Subdivision Method (SSM) [16] (Fig. 1(c)), etc. The most widely used method is the CSM, in which the patches are obtained by simply connecting the source point with each vertex of the element. Note that the requirement for achieving the connectivity of the desirable mesh can still be a bottleneck for complex geometries. Inevitably, there are some slender volume elements in the process of mesh generation [17] (see Fig.2). As such, the CSM may produce poor-quality patches in cases where the

element is distorted or irregular in shape. Based on the quad-tree data structure, the QTSM may generate more pleonastic patches for integration, thus resulting in low efficiency of computation. These methods are implemented in the local coordinate system of the element rather than in the physical coordinate system. Thus, the accuracy of integral evaluated by the CSM or the QTSM is easily affected by the shape of the element and the location of the source point. In addition, the ultimate patches obtained by these methods can hardly guarantee the accuracy of domain integration with discontinuous kernel. The SSM proposed by Zhang [16], which has the ability to subdivide an element into a number of well-shaped patches through a sequence of spheres with decreasing radius. Similar to the Advancing Front Method (AFM) in mesh generation, the SSM is an empirical subdivision method without strict theory demonstration, but has been verified by a large amount of experiments. This method is applicable to arbitrary fundamental solution, arbitrary shape of element and arbitrary location of the source point. Although the SSM overcomes all the difficulties associated with integration in BEM, it does not guarantee convergence of successful element subdivision for some situations. To take care of these situations, we propose an adaptive and efficient element subdivision method based on binary tree [18] for nearly singular domain integrals with continuous or discontinuous kernel in BEM. The proposed method is an essential complement to the SSM, which can generate patches successfully under any circumstances.



**Fig. 2.** Adaptive mesh generation of thin structures.

This paper is organized as follows. Detailed description of an adaptive and efficient volume element subdivision method using binary tree is presented in Section 2. Section 3 presents the key ideas of the BTSM for nearly singular domain integrals with continuous or discontinuous kernel. Section 4 presents construction of valid jagged core cavities for projection which is an important part

of our method. Section 5 details how the cavity projection algorithm works to generate high-quality patches. Numerical examples for different types of volume elements with various relative locations of the source point are given in Section 6 followed by conclusions in Section 7.

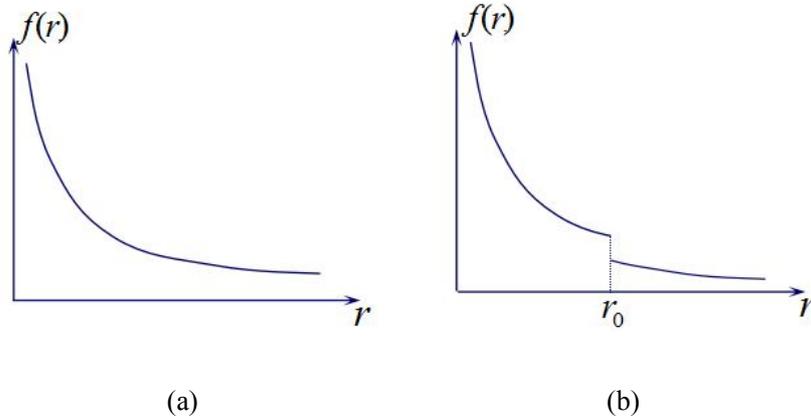
## 2. Volume element subdivision method using binary tree

Consider a general fundamental solution  $f(P, Q)$  for three-dimensional problems. The kernel  $f(P, Q)$  can be classified into two types: continuous function and discontinuous function (see Fig.3).

If the kernel  $f(P, Q)$  is bounded everywhere in the domain for the source point  $P$ , the domain integral is regular. On the other hand, if the kernel  $f(P, Q)$  becomes infinite at some points, it is singular or nearly singular. The following domain integral is dealt with in this paper:

$$I(P) = \int_{\Omega} f(P, Q) N(Q) d\Omega = \int_{\Omega} \frac{\bar{f}(P, Q)}{r^{\beta}(P, Q)} N(Q) d\Omega \quad (1)$$

where  $P$  is the source point,  $Q$  is the field point,  $\Omega$  is the domain of problem,  $\bar{f}(P, Q)$  is the non-singular part of the kernel, and  $r(P, Q)$  is the distance between the points  $P$  and  $Q$ .

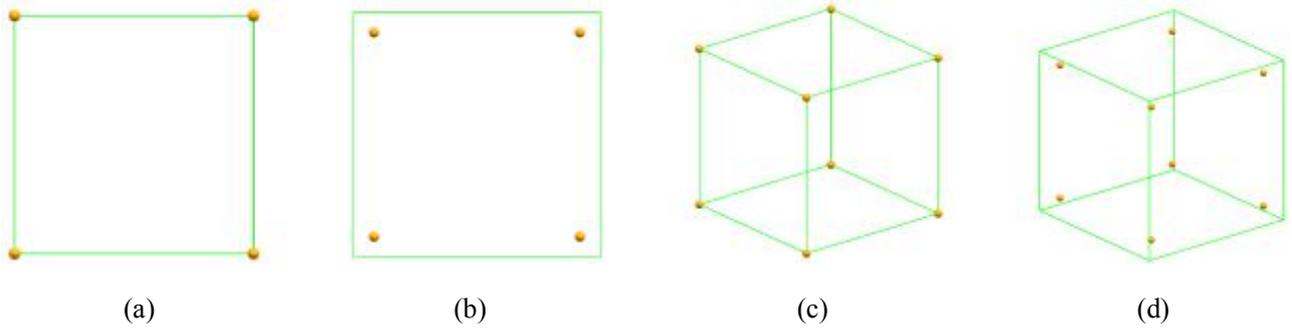


**Fig. 3.** Different types of fundamental solution: (a) the continuous kernel, (b) the discontinuous kernel.

### 2.1 Element Subdivision Rules

Spatial decomposition methods such as binary tree, quadtree and octree, were originally proposed as a way to represent approximation of geometric objects. These methods have been designed to meet the needs of fully automatic spatial decomposition and employed extensively to arbitrary complex non-manifold geometries. The spatial decomposition methods also reduce the extensive amount of time and effort required to generate sub-regions with semi-automatic methods. These

decomposition approaches are robust and reliable in theory and have a wide range of applications in engineering. Hence, we opt for binary-tree subdivision to improve the computational accuracy of the domain integrals and overcome disadvantages of the CSM, the QTSM and the SSM. In this method, the volume element is firstly subdivided into a number of patches by binary tree and is intersected with sphere of specified radius if necessary. A significant advantage of this structure is that a single data structure can handle volume element subdivision very efficiently.

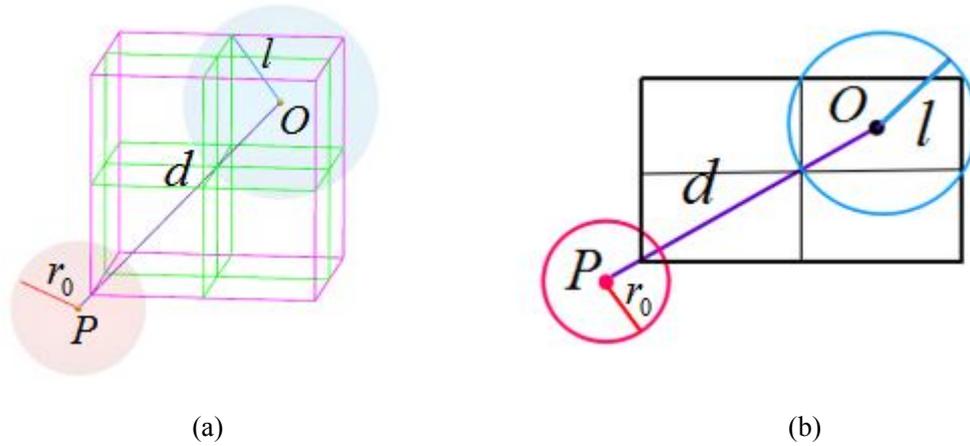


**Fig. 4.** The standard surface and volume elements in BEM: (a) the continuous surface elements,

(b) the discontinuous surface elements, (c) the continuous volume elements, (d) the discontinuous volume elements.

Application of the BEM to solve a boundary integral equation with domain integrals, the surface mesh and the domain mesh are usually both required to be generated. As is illustrated in Fig. 4, unless otherwise mentioned, the non-mesh source points in this paper mainly refer to the nodes (the sphere marked in yellow) of the standard surface or volume elements, such as the continuous and discontinuous elements in BEM. In the binary-tree subdivision scheme, the subdivision rules are based on the local geometric properties of sub-element and sphere or the reference subdivision ratio. The schematic description in Fig. 5 explains the element subdivision rule in three dimensions and in two dimensions, respectively. For nearly singular domain integrals, the subdivision rules are only based on the subdivision ratio  $\eta$ . The subdivision criterion of element subdivision for nearly singular domain integrals can be expressed as  $\eta < \eta_{Ref}$ . The subdivision ratio  $\eta$ , by definition, is the ratio of circumradius  $l$  of sub-element to the distance  $d$  between the source point  $P$  and the geometric center  $O$  of sub-element, i.e.  $\eta = l / d$ . The reference subdivision ratio  $\eta_{Ref}$  is defined on the basis of experience and different types of volume element. In the process of implementation, the reference subdivision ratio  $\eta_{Ref}$  is set to around 1.0. For regular volume elements, the reference

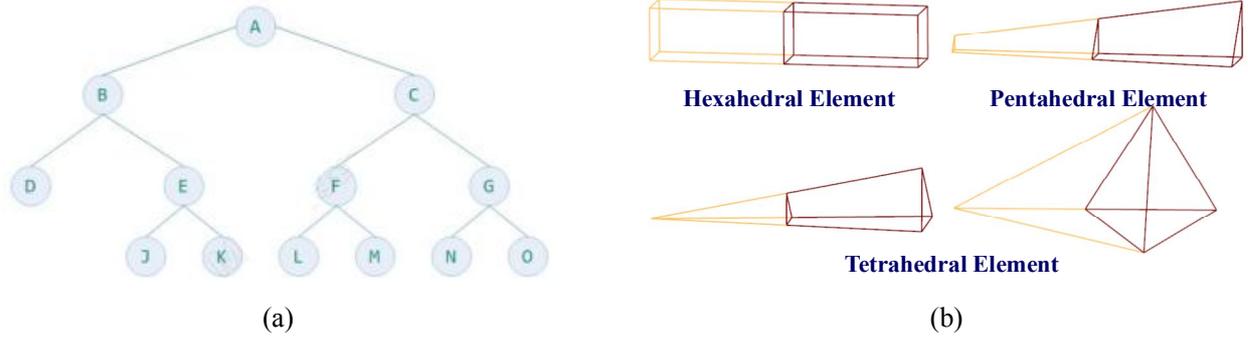
subdivision ratio  $\eta_{Ref}$  is greater than or equal to 1.0. For slender volume elements, the reference subdivision ratio  $\eta_{Ref}$  is less than or equal to 1.0. The specific parameter of the reference subdivision ratio  $\eta_{Ref}$  is determined based on extensive experimental data. The subdivision will be executed continuously when the subdivision ratio is greater than the reference value. For nearly singular domain integrals with continuous kernel, the specified radius  $r_0$  is set to 0. For nearly singular domain integrals with discontinuous kernel, the specified radius  $r_0$  denotes the distance between the source point and the discontinuous point.



**Fig. 5.** The element subdivision criterion: (a) description of the element subdivision rule in three dimensions, (b) schematic of the element subdivision rule in two dimensions.

## 2.2 Element Subdivision Techniques

A binary tree is a tree data structure in which each node has at most two children. Hence, during the process of element subdivision, the root volume element is subdivided into two equal-sized or different kinds of sub-elements, and then each of them may be recursively refined until the ultimate sub-elements meet the stopping criterion. Base on the size and type of volume element, different element subdivision techniques can be applied to them. As is illustrated in Fig. 6, hexahedral, pentahedral and regular tetrahedral elements can be subdivided into two sub-elements of the same shape as their ancestor. A slender tetrahedral element can be subdivided into a tetrahedral sub-element and a pentahedral sub-element to achieve best performance of element subdivision.

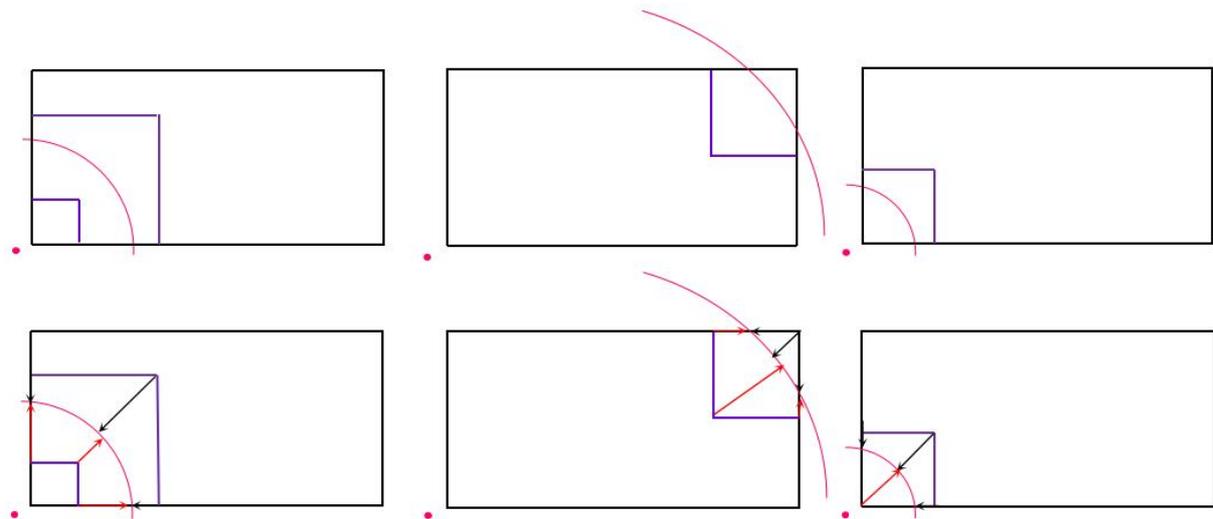


**Fig. 6.** Element subdivision techniques: (a) the binary-tree data structure, (b) subdivision techniques of different types of elements.

### 3. The key ideas of the BTSM for nearly singular domain integrals

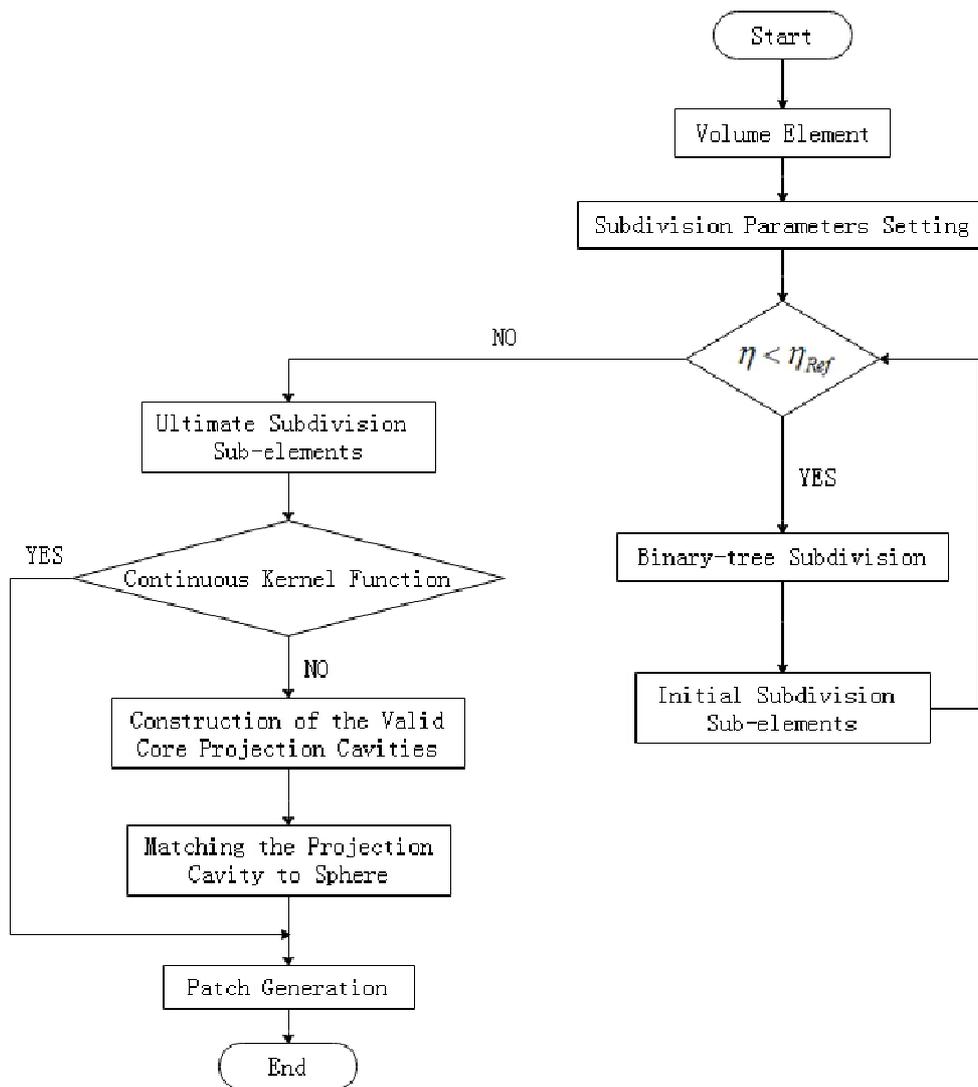
For nearly singular domain integrals with continuous kernel, the fundamental solution is continuous everywhere in the domain of problem. It is unnecessary to perform any additional operation for element subdivision other than the binary-tree subdivision scheme. A reasonable result can be achieved with the above mentioned subdivision algorithm for different types of elements.

For nearly singular domain integrals with discontinuous kernel, the fundamental solution may be discontinuous at some points, called discontinuous points. Theoretically, the domain integrals are actually regular since the value of their integrand is finite. However, the value of the integrand varies dramatically if the field point is at the discontinuous points of the fundamental solution or the source point is close to the element of integral. The ultimate patches are naturally divided into two parts by the sphere with specified radius  $r_0$  which is the distance between the source point and the discontinuous point. To better understand, the algorithm will be explained in two dimensions. The projection cavity construction algorithm can be regarded as an extension of the algorithm for the two-dimensional case. An elimination process is necessary in which the straddling sub-elements are thrown away, and remaining sub-elements are taken as the eligible patches. Thus, the inner and outer projection cavities are constructed after the elimination process. There is a separation between the cavity faces and the boundary of sphere. It is reasonable to generate patches by projecting the cavity faces along the specified direction until they reach the sphere (see Fig. 7). The key ideas of the BTSM for nearly singular domain integrals with continuous or discontinuous kernel is further described in the flow diagram shown in Fig. 8.



(a) Both inner cavity and outer cavity      (b) Only the inner cavity      (c) Only the outer cavity

**Fig. 7.** The key ideas of the BTSM with different types of projection cavity construction for nearly singular domain integrals with discontinuous kernel.

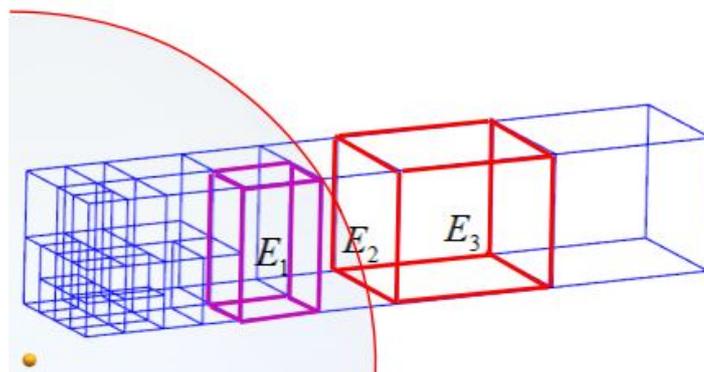


**Fig. 8.** Flow diagram of the BTSM for nearly singular domain integrals with continuous or discontinuous kernel.

#### 4. Construction of the valid core projection cavities

An important part of the binary-tree element subdivision method is construction of valid jagged core cavities for projection. For nearly singular domain integrals with discontinuous kernel, it is necessary to construct the inner and outer projection cavities. A key aspect of the projection cavity construction is to pick up the faces of sub-elements obtained in the ultimate refinement structure [19]. One of the basic requirements of projection cavity construction is that the cavity faces should be as near to the boundary of sphere as possible. Further, the faces of the core projection cavity must be matched to the boundary of sphere. Hence, construction of the valid core projection cavities is of crucial importance for successful implementation of patch generation.

To construct the projection cavity conveniently and automatically, the status of ultimate sub-element entities should first be set. As is illustrated in Fig. 9, the ultimate subdivision sub-elements obtained and their related topology items, including vertices, edges and faces, can be classified into three types: exterior, interior and straddling. The status of vertices (whether they are in the interior or exterior of sphere) is determined by the distance to the sphere center. If the distance is less than the radius of sphere, the vertex is an interior one. Otherwise, the status of the vertex is exterior. The status of sub-elements can be obtained simply after the judgment of vertex status. A sub-element will be named as an interior/exterior element if its vertices are all interior/exterior. The remaining sub-elements and their related topology items that are not entirely within the sphere are named as the straddling entities. Thus, the sub-elements  $E_1$ ,  $E_2$  and  $E_3$  (marked in purple, blue and red in Fig. 9, respectively) are interior sub-element, straddling sub-element and exterior sub-element, respectively.

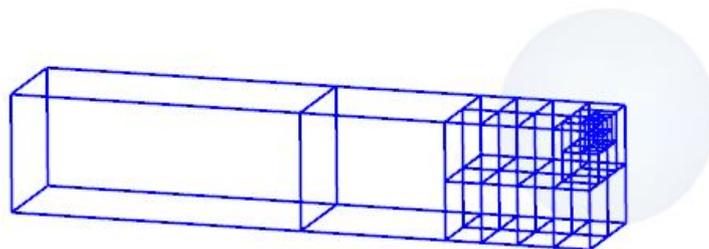


**Fig. 9.** The status of ultimate sub-elements topology items.

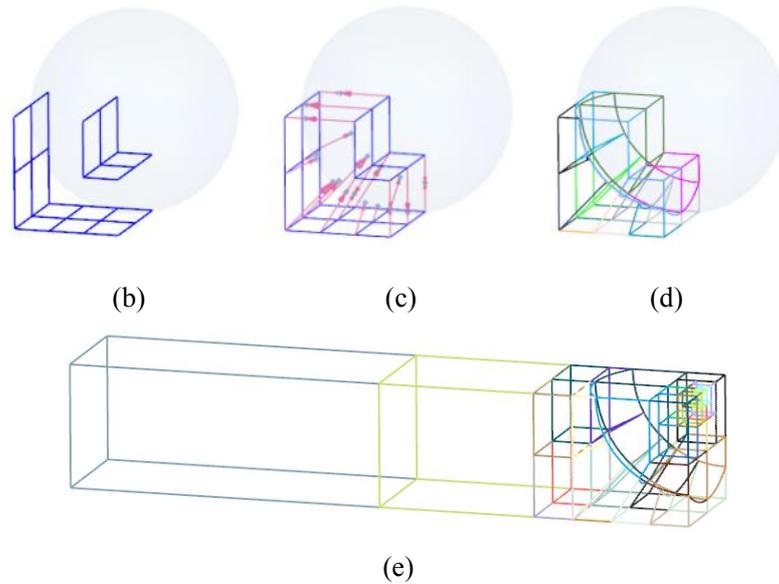
#### 4.1 The projection cavity construction algorithm

The projection cavity construction algorithm is achieved by eliminating some undesired sub-elements that intersected with sphere from the ultimate refinement result. The elimination process is conducted by superimposing ultimate refinement result. The straddling sub-elements are thrown away. Remaining sub-elements are taken as the eligible patches. In this process, the jagged core cavity faces should be as near to the sphere as possible. However, there is a separation between the core cavity and the sphere which nearly equals to the edge of a patch in length. The spatial region is named as the *cavity-gap* in this paper (see Fig. 10).

As is illustrated in the top row of figure 7, three different types of projection cavity construction have been identified based on the intersection between the sphere and faces of sub-elements: (i) both inner cavity and outer cavity, (ii) only the inner cavity and (iii) only the outer cavity. The first projection cavity type are all regular cavity faces (i.e., quadrilaterals for all hexahedral sub-elements or triangles for all tetrahedral sub-elements), that do not intersect with the sphere. The second and third projection cavity types are due to the radius of sphere centered at the source point being too large or too small, respectively. In these two cases, only the inner cavity or the outer cavity can be obtained. For this situation, the projection cavity construction algorithm is invalid and unenforceable to perform the cavity projection algorithm for patch generation. Thus, a projection cavity construction technique by virtual operation, called virtual cavity face zoning algorithm is proposed in the next subsection.



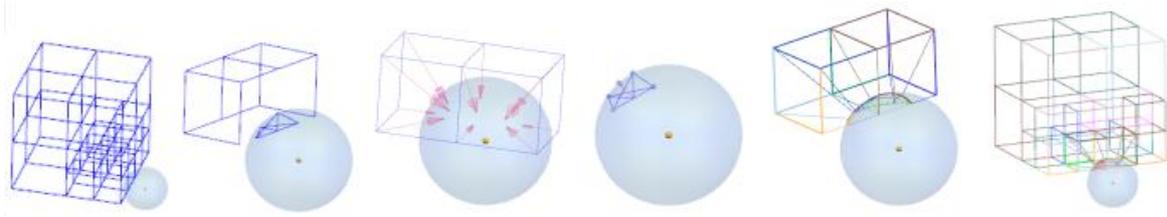
(a)



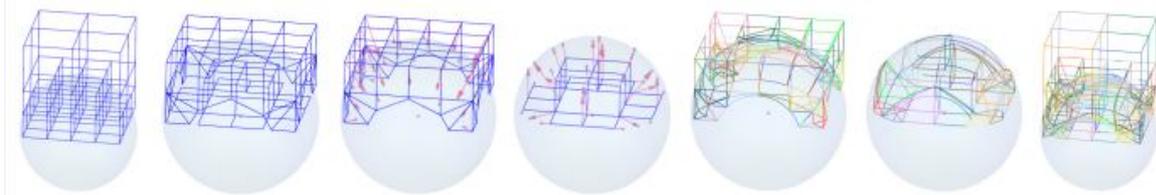
**Fig. 10.** Construction of valid jagged core cavities for projection: (a) ultimate subdivision sub-elements, (b) the core projection cavities, (c) matching the core projection cavity to sphere, (d) convert relative sub-elements into well-shaped serendipity patches, and (e) the resulting patch generation.

#### 4.2 Virtual cavity face zoning algorithm

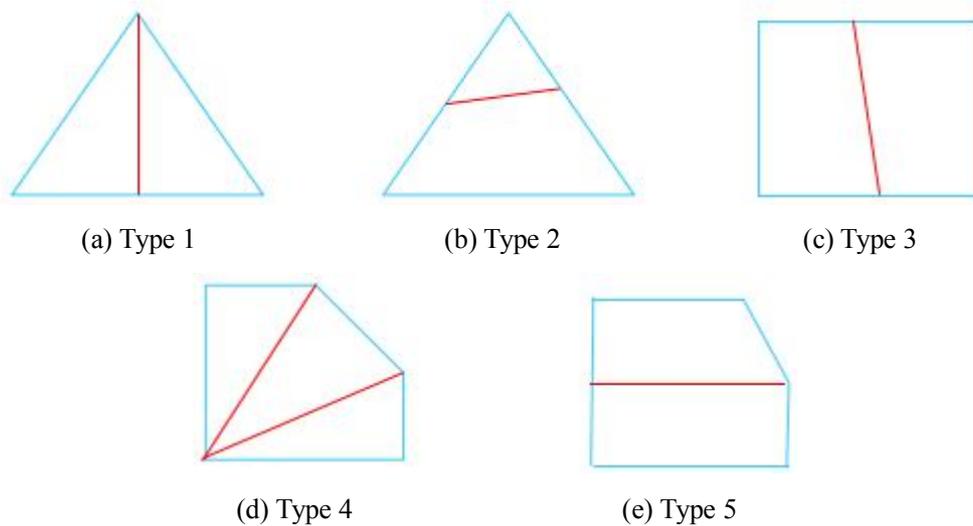
In order to ensure the validity of the projection cavity, an adaptive virtual cavity face zoning algorithm is proposed. As depicted in Fig.11 and Fig.12, there is an intersection loop between the sphere and the volume element. The intersection loop will subdivide the sub-element faces into a set of polygonal faces, such as triangles, quadrilaterals or pentagons. The triangle faces and quadrilateral faces are both regular cavity faces for patch generation. We can use the triangle faces and quadrilateral faces to generate pentahedral and hexahedral patches, respectively. The pentagonal faces are unable to perform the cavity projection algorithm directly, since there are no corresponding patches for this cavity face type. Therefore, cavity faces of this kind should be converted to several regular cavity faces, such as quadrilaterals or triangles. Five different types of polygonal face partitions are summarized in Fig. 13 according to the intersection between the sphere and the volume element. Type 1 and Type 2 indicate the cases of a triangle cavity face intersected with sphere. Type 3 to Type 5 represent that four vertices of a quadrilateral cavity face fall on either side of sphere. The partitioning techniques are easily applied to reasonable cavity face configuration for cases of arbitrary intersection between the sphere and element.



**Fig. 11.** An example of the inner virtual cavity construction.



**Fig. 12.** An example of the outer virtual cavity construction.

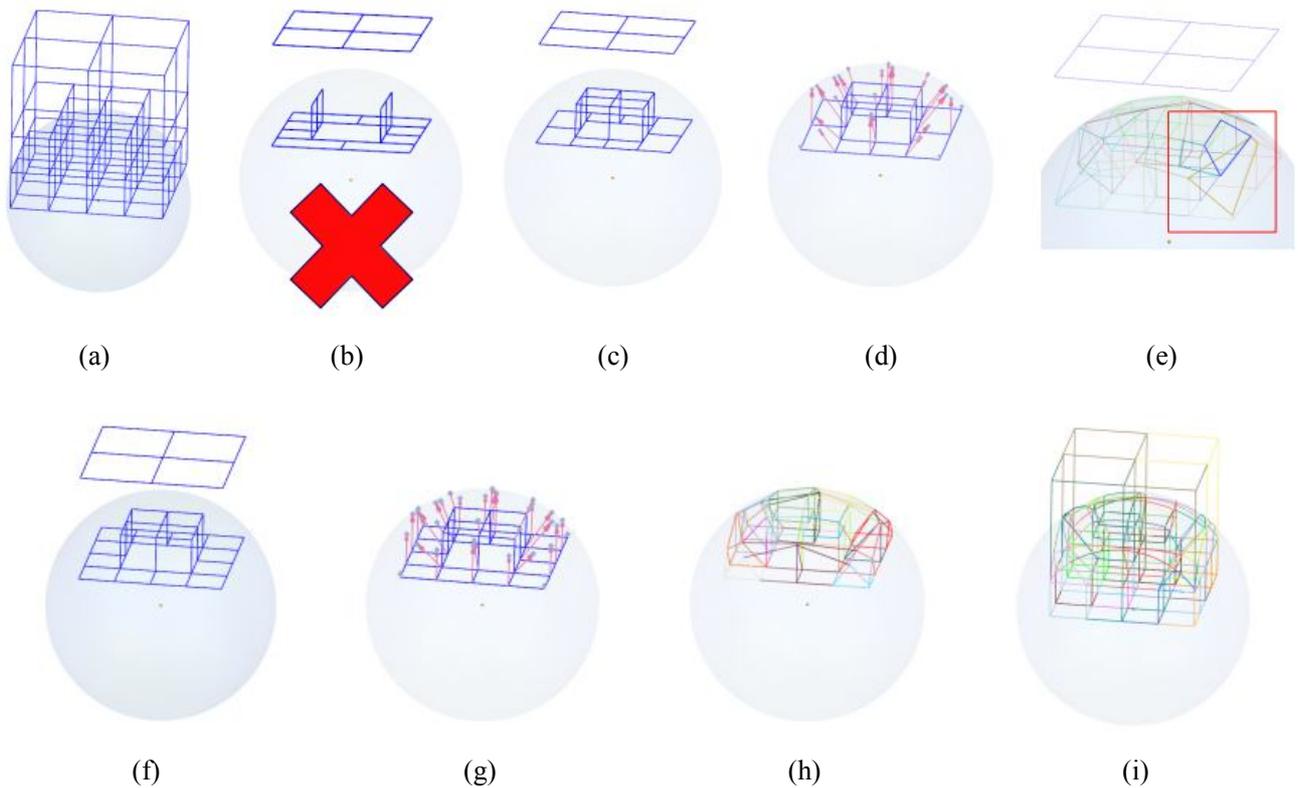


**Fig. 13.** Five types of polygonal faces partition based on intersection between the sphere and element.

The following techniques are employed to construct the valid projection cavities:

- (1) Pick up the exterior faces of the straddling sub-elements as the component of outer projection cavity. If there are no outer cavity faces, a virtual outer cavity is constructed using the virtual cavity face zoning algorithm.
- (2) If the inner projection cavity is needed, pick up the interior faces of the straddling sub-elements or the faces of interior sub-elements that contact with straddling sub-elements. If there are no interior cavity faces, construct a virtual interior cavity.
- (3) Check the validity of inner or outer projection cavity construction. A valid projection cavity requires that the continuity of the cavity faces should be guaranteed. It means that

there are no gaps between the cavity faces. Figure 14(b) shows an example of invalid projection cavity. Further, the cavity faces should be balanced at the same level to avoid the projection interference, as is shown in Fig. 14(e). Another requirement is that if the filtered straddling sub-elements contain corner vertices; these corner vertices should be included in the projection cavity.



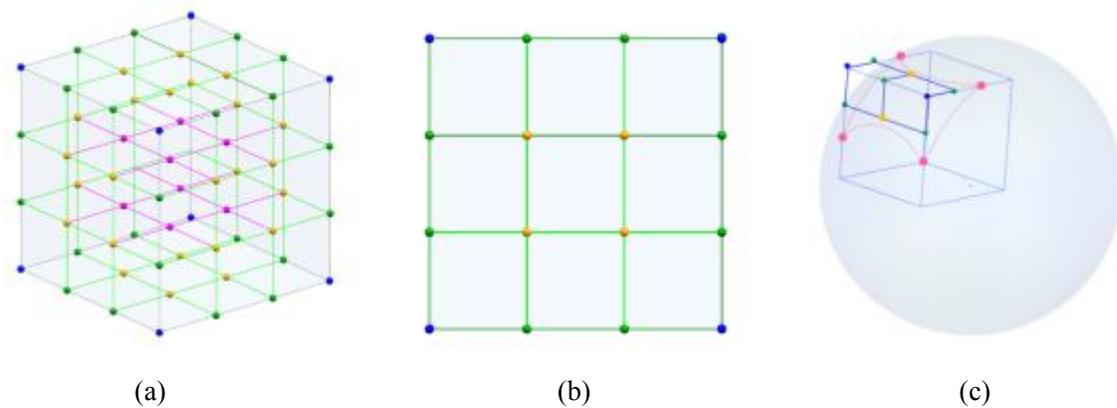
**Fig. 14.** Schematic of the valid core jagged projection cavities construction: (a) ultimate subdivision sub-elements, (b) the invalid projection cavity, (c) (d) the projection cavity without balance and its relative projection points, (e) the projection interference, (f) (g) the balanced projection cavity and its relative projection points, (h) convert relative sub-elements into well-shaped patches, and (i) the resulting patch generation.

## 5. Matching the projection cavity to sphere

Taking into account the definition of *cavity-gap*, it is reasonable to generate patches by projecting the cavity faces along the specified direction or sweeping path until they reach the sphere (see the bottom row of figure 7 for a simplified two-dimensional illustration). A general projection algorithm and a sweeping projection algorithm are proposed and illustrated in this section. These algorithms are used to fill the *cavity-gap* with a new layer of well-shaped patches, such as tetrahedral or pyramidal, pentahedral or hexahedral patches.

## 5.1 The general projection algorithm

The general projection algorithm is based on the location of vertices and the number of cavity faces around the vertex. Before discussing the general projection algorithm, some vocabulary of cavity vertices according to their location is illustrated in Fig. 15. The blue vertices on the corners of an element are called the *corner vertices*. The green vertices at the edges are called the *frame vertices*. The yellow vertices on the boundary faces are called *boundary vertices* and the purple vertices inside the cell are called *interior vertices*. The red vertices over the cavity loop and located on the root element edge are called *inflex vertices*.

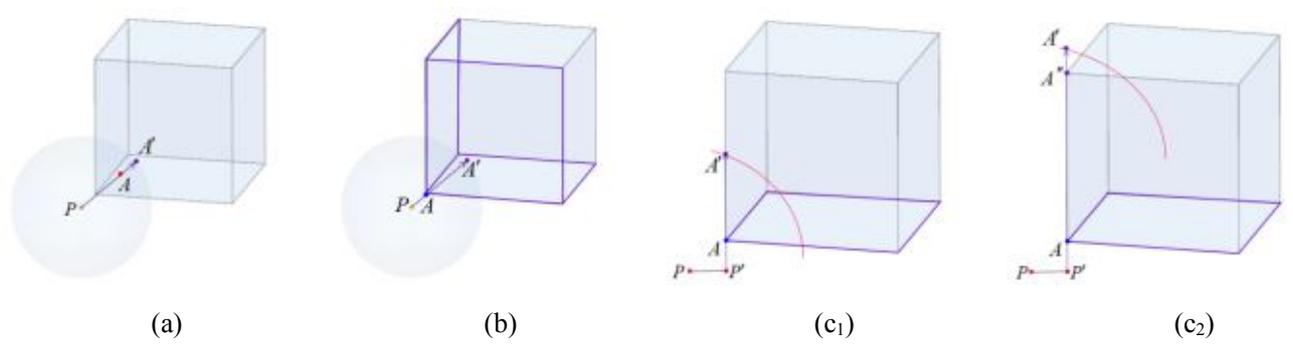


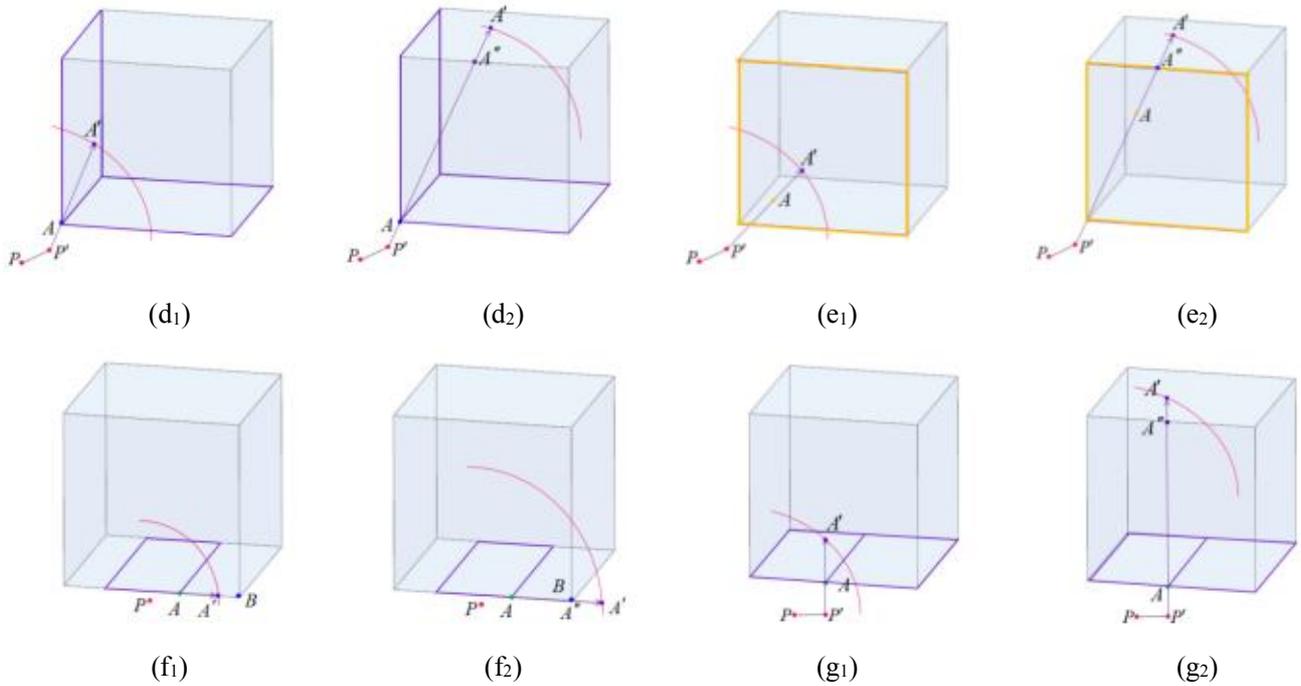
**Fig. 15.** Different types of projection vertices according to the location: (a) schematic of different types of vertices, (b) the partial enlarged view of a boundary face, and (c) schematic of the inflex vertices and the intersection loop.

In order to ensure that the general projection algorithm can be applied, the following requirements have to be satisfied. A vertex labeled  $P'$  is the projection source point, which is obtained by projecting the source point  $P$  to the boundary face or edge of element (see Fig. 16). The vertices labeled  $A$  colored purple, yellow, blue, green and red represent interior, boundary, corner and frame projection points, respectively. The vertices labeled  $A'$  on the sphere are called the target projection points, which is obtained by projecting the projection points  $A$  along the specified direction  $\overline{P'A}$  to the spherical surface. If the target projection point  $A'$  lies beyond the range of the boundary face or edge of an element, it should be moved to points  $A''$  which is within the element. Different target projection points  $A'$  can be obtained according to the projection point type and the number of cavity faces around the projection point  $A$ .

As is illustrated in Fig. 16, detailed description of the projection of different types of vertices using

the general projection algorithm is presented. The faces with purple lines are the cavity faces. The faces with yellow lines are the boundary faces of the volume element. For the interior vertex and the corner vertex that belongs to all its surrounding cavity root faces, the target projection point  $A'$  is obtained by projecting the projection point  $A$  along the radical direction (Fig. 16(a) and Fig. 16(b)). For other situations, we should first project the source point  $P$  on the boundary root face or edge to get the projection source point  $P'$ . For the corner vertex that belongs to only one boundary root cavity face, the target projection point  $A'$  is obtained by projecting the projection point  $A$  along the boundary root edge that does not belong to the cavity root face (Fig. 16(c<sub>1</sub>)). For the corner vertex that belongs to two boundary cavity root faces, the target projection point  $A'$  is obtained by projecting the projection point  $A$  along the boundary root face which is not the cavity face (Fig. 16(d<sub>1</sub>)). For the boundary vertex, project the projection point  $A$  along its belonging boundary root face to the spherical surface to get the target projection point  $A'$  (Fig. 16(e<sub>1</sub>)). For the frame vertex that belongs to only one cavity face, project the projection point  $A$  along the boundary root edge to the spherical surface to get the target projection point  $A'$  (Fig. 16(f<sub>1</sub>)). For the frame vertex that belongs to multiple cavity faces, the target projection point  $A'$  is obtained by projecting the projection point  $A$  along the boundary root face which is not the cavity face (Fig. 16(g<sub>1</sub>)). If the target projection point  $A'$  lies beyond the range of the boundary root edge of an element, it should be moved to points  $A''$  which is within the element (see (c<sub>2</sub>) and (f<sub>2</sub>) in Fig. 16). If the target projection point  $A'$  lies beyond the range of the boundary root face of an element, it should be moved to points  $A''$  which is within the element (see (d<sub>2</sub>), (e<sub>2</sub>) and (g<sub>2</sub>) in Fig. 16).



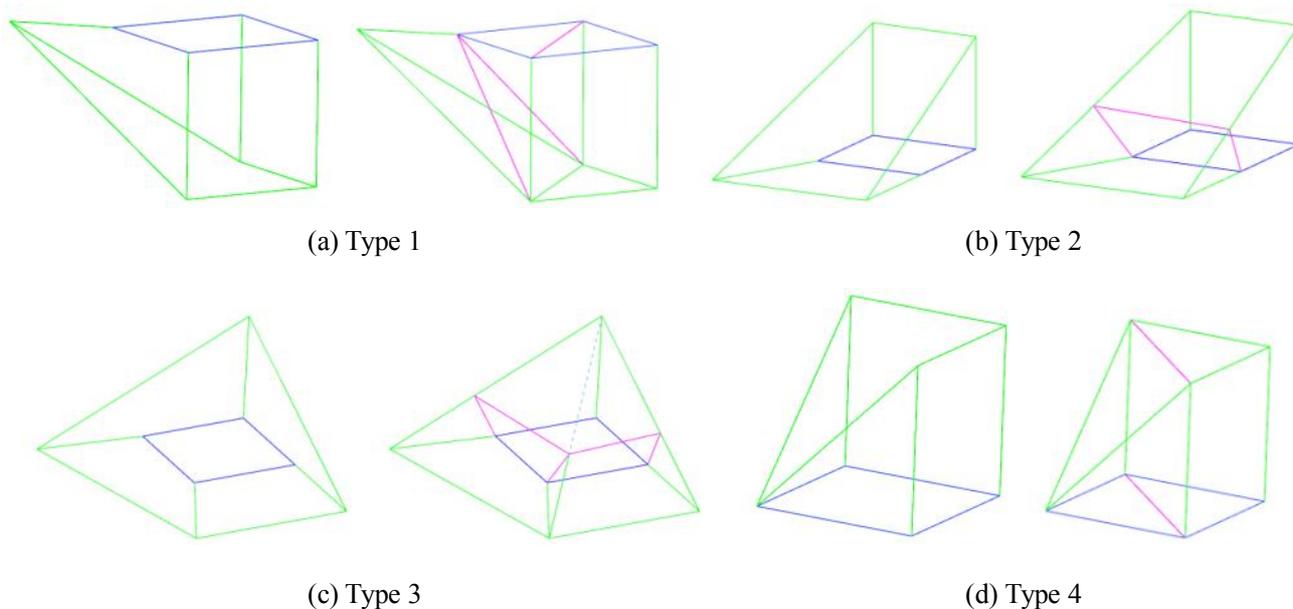


**Fig. 16.** Projection of different types of vertices using the general projection algorithm: (a) projection of the interior vertex, (b) projection of the corner vertex that belongs to all its surrounding cavity root faces, (c<sub>1</sub>)(c<sub>2</sub>) projection of the corner vertex that belongs to only one boundary cavity root face, (d<sub>1</sub>)(d<sub>2</sub>) projection of the corner vertex that belongs to two boundary cavity root faces, (e<sub>1</sub>)(e<sub>2</sub>) projection of the boundary vertex, (f<sub>1</sub>)(f<sub>2</sub>) projection of the frame vertex that belongs to only one cavity face, (g<sub>1</sub>)(g<sub>2</sub>) projection of the frame vertex that belongs to multiple cavity faces.

The general projection algorithm is used for the projection of cavity vertices over the cavity loop in the nearly singular domain integrals with discontinuous kernel. It is reasonable to generate patches by projecting the cavity faces along the specified direction to fill the *cavity-gap*. For nearly singular domain integrals with discontinuous kernel, the vertices over the cavity loop including boundary vertices, frame vertices and corner vertices, should be projected along the boundary of element.

Inevitably, there are a limit number of degenerate patches generated when using the general projection algorithm. These degenerate patches can be decomposed into several regular well-shaped patches. Figure 17 lists corresponding decomposition templates for the different kinds of degenerate patches. The decomposition templates as the segmentation of the degenerate patches into regular ones are not unique. In addition to these basic decomposition templates, other available decomposition templates can be created and developed according to specified requirements or the intention of designers. The blue vertices and green vertices in Fig. 17 denote the projection cavity

points and target projection points, respectively. In our implementation, Type 1 describes the situation that one target projection point is coplanar with the cavity face, which is decomposed into a tetrahedral patch, a pyramid patch and a pentahedral patch. Type 2 depicts that two target projection points are coplanar with the cavity face, which is decomposed into a pentahedral patch and a hexahedral patch. The situation described in Type 3 is that three target projection points are coplanar with the cavity face, which is decomposed into a hexahedral patch and two pentahedral patches. Type 4 illustrates that the target projection point and its projection point may be in the same position, which is decomposed into a pyramid patch and a pentahedral patch. For symplectic domains, the hexahedral patches and the pyramid patches are integrated using classical Gauss quadrature. The tetrahedral patches are integrated using Hammer quadrature. More details of the proper quadrature rules for symplectic domains can be found in [20-22].

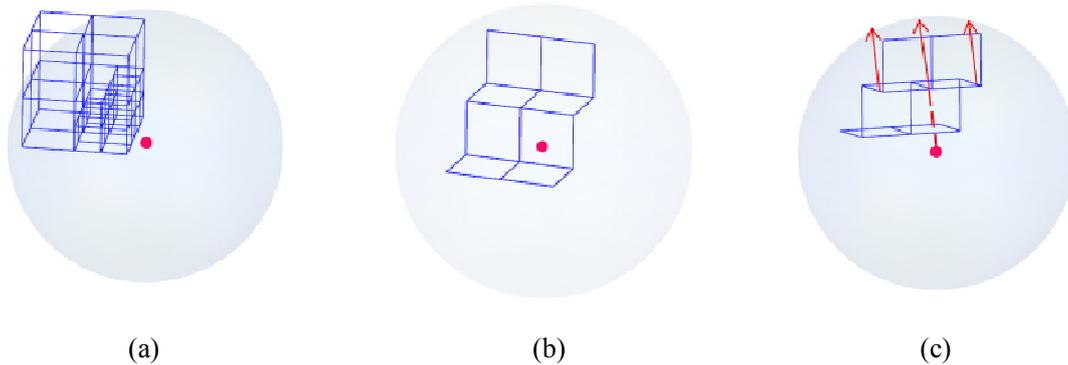


**Fig. 17.** Decomposition templates for degenerate patches.

## 5.2 The sweep projection algorithm

In most cases, a mass of target projection points is projected outside of the volume element when the sphere has a larger radius or the element is irregular, as shown in Fig. 18. It is very difficult to solve this problem by the general projection algorithm. In this section, the sweep projection algorithm is proposed for projection of cavity vertices in nearly singular domain integrals with discontinuous kernel. Sweep method is one of the most robust mesh techniques to generate volume meshes in extrusion volumes, which has an outstanding performance in the field of mesh generation.

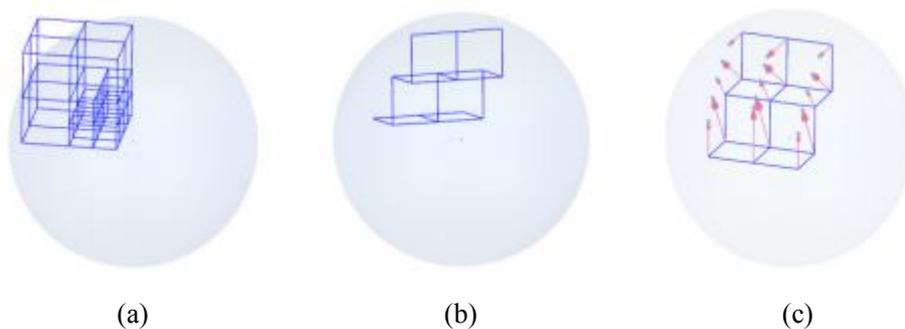
More details of the sweep method can be found in [23].

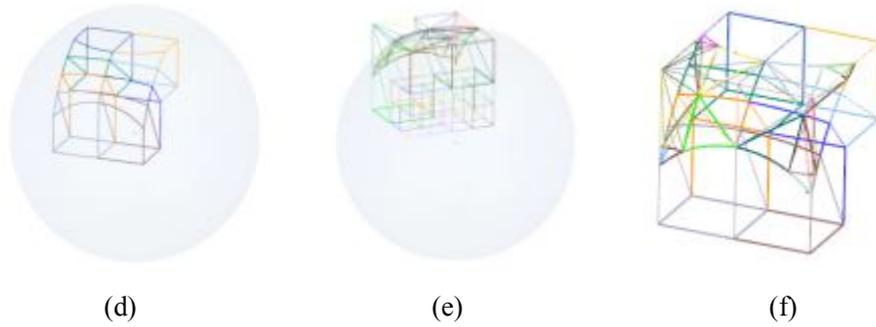


**Fig. 18.** A tricky situation that a mass of target projection points are projected outside of the volume element: (a) ultimate subdivision sub-elements, (b) construction of the inner cavity, and (c) relative target projection points.

One of the main issues to be dealt by any sweep algorithm is the projection of a source surface mesh onto the target surface. For cases of arbitrary intersection between sphere and the volume element, the sweep projection algorithm is used for projection of inner or outer cavity faces. This projection is carried out by means of a least-squares approximation of an affine mapping defined between the 3D spaces of the cavity faces. The main advantage of this method is that the sweep projection algorithm does not require solution of any root finding problem to ensure that the target projection points are on the target surface.

The sweep projection algorithm presented in this paper regards the inner or outer cavity as the source surface and a piece of sphere with the intersection loop as the target surface. A particular requirement is that the source cavity surface and target spherical surface must be topologically equivalent and have same connectivity. Taking into account any situation of arbitrary intersection between sphere and the volume element, it is reasonable to generate patches successfully under any circumstances by sweeping a given discretization of the source cavity faces along the sweep path. An example in Fig. 19 illustrates the feasibility and advantages of the sweep projection algorithm.





**Fig. 19.** Filling the cavity-gap using the sweep projection algorithm: (a) ultimate subdivision sub-elements, (b) construction of the inner cavity, (c) relative target projection points, (c) relative target projection points, (d) the interior serendipity patches, (e) the resulting patch generation, and (f) the partial enlarged view of relative serendipity patches.

The main algorithm for patch generation by sweeping can be decomposed into the following five steps:

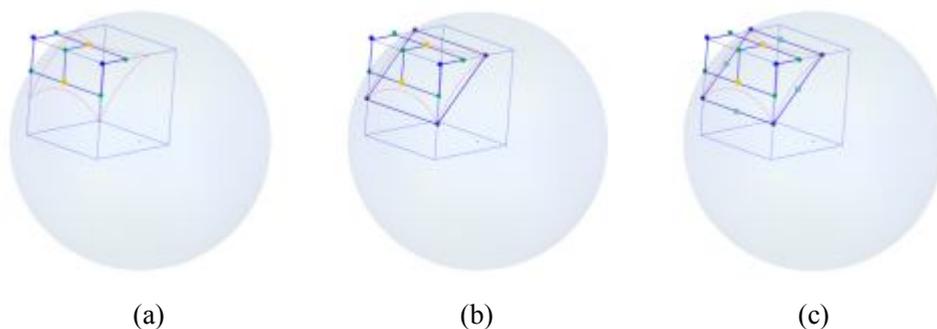
Step 1. Construct the projection cavities using the core cavity construction algorithm and check their validities, as illustrated in Fig. 20(a).

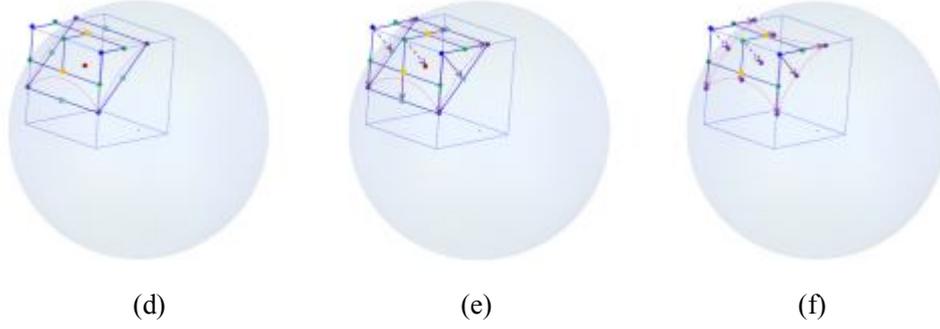
Step 2. Form an intersection loop as shown in Fig. 20(b) (i.e., form a polygon whose edges are parallel to the boundary of element) as the boundary loop of target spherical surface for sweeping.

Step 3. Project the vertices over the outer loop of source cavity using the general projection algorithm. Thus, an affine mapping can be established by the boundary data between the source cavity surface and the target spherical surface, as shown in Fig. 20(c).

Step 4. Project the interior vertices of source cavity faces onto the target spherical surface by sweeping, as shown in Fig. 20(d).

Step 5. Once all the target projection points on sphere are obtained, the *cavity-gap* can be filled with a new layer of pentahedral or hexahedral and other kinds of degenerate but well-shaped patches. Convert the degenerate patches to regular patches based on the standard decomposition templates (see Fig. 20(e) and Fig. 20(f)).





**Fig. 20.** Schematic of the sweep projection algorithm: (a) construction of the projection cavity, (b) forming an intersection loop, (c) projection of the vertices over the outer loop of source cavity, (d) projection of the interior vertices of source cavity, and (e) (f) relative target projection points.

## 6. Numerical examples

Three groups of numerical examples and corresponding adaptive element subdivision for nearly singular domain integrals with continuous and discontinuous kernel function are presented in this section. Comparisons are made between the BTSM and the CSM for volume element integrals. For the purpose of error estimation, relative error is defined as follows:

$$\delta = \left| \frac{I_{(n)} - I_{(e)}}{I_{(e)}} \right| \quad (2)$$

where  $\delta$  is the relative error, and the subscripts  $(e)$  and  $(n)$  of  $I$  refer to the exact and numerical solution of the volume element integral, respectively.

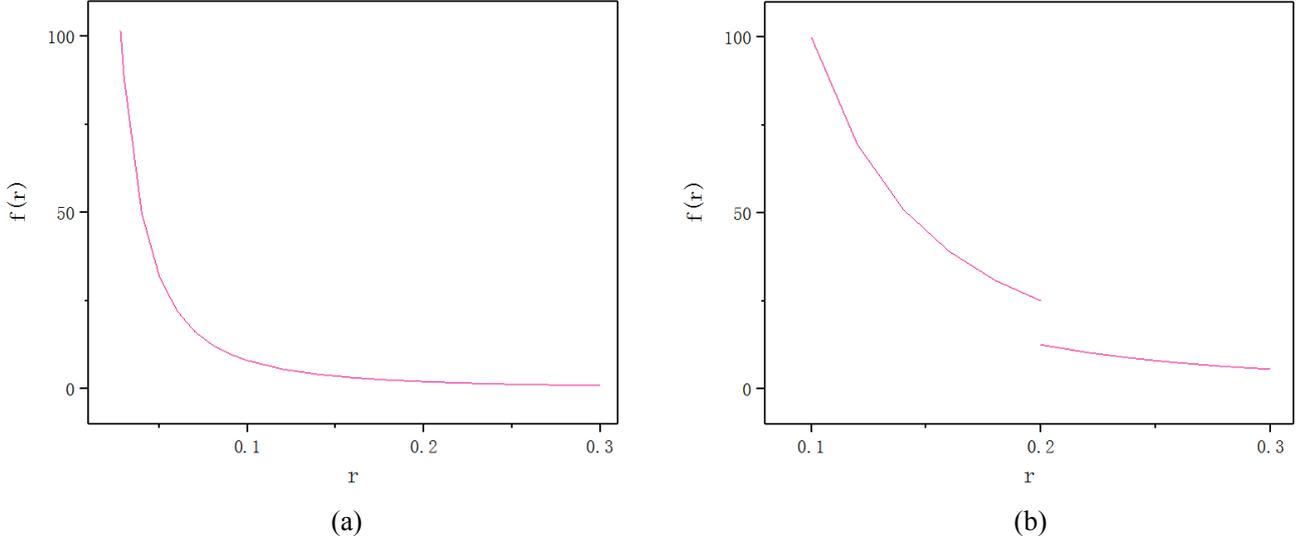
The following integral is considered to evaluate the nearly singular domain integrals with continuous kernel function:

$$I = \int_{\Omega} \frac{1}{4\pi r^2} N d\Omega \quad (3)$$

We consider the following integral for numerical evaluation of the nearly singular domain integrals with discontinuous kernel function:

$$I = \begin{cases} \int_{\Omega} \frac{1}{r^2} N d\Omega & r \leq 0.2 \\ \int_{\Omega} \frac{1}{2r^2} N d\Omega & r > 0.2 \end{cases} \quad (4)$$

In preceding equations,  $N$  is the shape function and  $r$  denotes the distance between the source point and the field point. Figure 21 provides a graphical illustration of the continuous and discontinuous kernels in Eq. (3) and Eq. (4).



**Fig. 21.** The graph of the continuous and discontinuous kernel: (a) the continuous kernel, (b) the discontinuous kernel.

For demonstration purposes, adaptive element subdivision and corresponding numerical results for volume elements of arbitrary type with various relative locations of the source point in the local coordinate system of the element are included. In all numerical examples, the number of the integration points used and the accuracy obtained by both the BTSM and the CSM have been compared. In the physical coordinate system, vertex coordinates of a slender hexahedral element are  $(1, 0, 0)$ ,  $(1, 0, 5)$ ,  $(1, 1, 5)$ ,  $(1, 1, 0)$ ,  $(0, 0, 0)$ ,  $(0, 0, 5)$ ,  $(0, 1, 5)$ ,  $(0, 1, 0)$ . Vertex coordinates of a slender tetrahedral element are  $(-3, -3, -3)$ ,  $(0, 1, 0)$ ,  $(0, 0, 1)$ ,  $(1, 0, 0)$ , while vertex coordinates of a regular tetrahedral element are  $(-2, -2, -2)$ ,  $(0, 2, 0)$ ,  $(0, 0, 2)$ ,  $(2, 0, 0)$ . Vertex coordinates of a slender pentahedral element are  $(0, 0.25, 0)$ ,  $(0, 0, 0)$ ,  $(0.25, 0, 0)$ ,  $(0, 1, 5)$ ,  $(0, 0, 5)$ ,  $(1, 0, 5)$ . For each example, we have included pictures of the patches obtained using our method. On the right of each figure is the partial enlarged view of the part containing source point, from which we can see if the patches of volume elements are acceptable in shape and size.

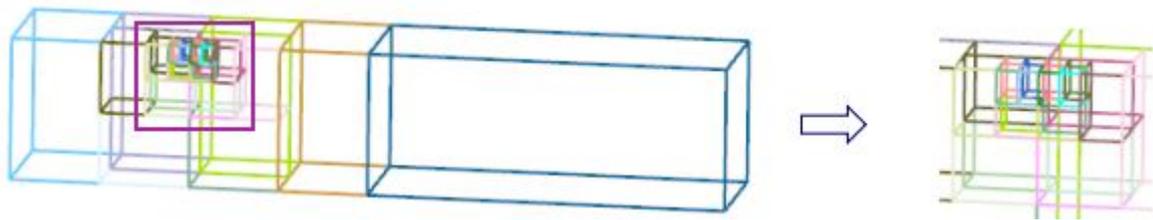
### 6.1 Evaluation of convergence performance of the BTSM

Three examples of different types of volume elements are presented to verify the accuracy and convergence performance of the BTSM for nearly singular domain integrals. Unless otherwise mentioned, Eq. (3) is used as the kernel for integration in this section. For the specific reference subdivision ratio  $\eta_{Ref}$ , the ultimate element subdivision of volume elements is unique for the given source point. In the BTSM implementation, different levels of the integration errors are guaranteed by different number of the integration points for the given source point. To study the convergence of

the BTSM, nearly singular domain integrals are evaluated by the BTSM with increasing number of the integration points for the same source point. Numerical results have been obtained for integration with different number of the integration points. A comparison of convergence of the CSM and the BTSM as shown below.

### Slender hexahedral element

Figure 22 shows the element subdivision of slender hexahedral element for the specified source point for nearly singular domain integrals. Numerical results are presented in Table 1 which clearly show the superior convergence performance of the proposed method as compared to the conventional subdivision method.



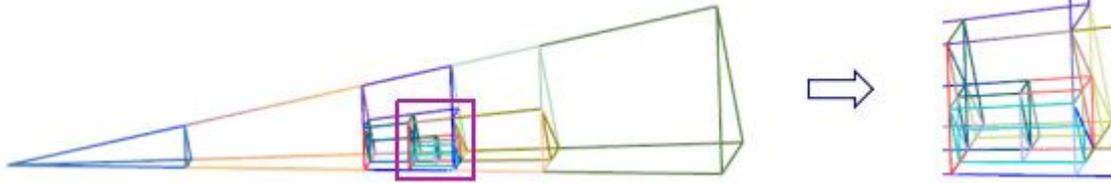
**Fig. 22.** The local parametric coordinate of source point is (1.01, -0.5, -0.75).

Table 1: Convergence of BTSM and CSM for nearly singular domain integrals  
with the source point at (1.01, -0.5, -0.75)

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM
Slender hexahedral element	(1.01, -0.5, -0.75)	625	434	1.33e-001	8.25e-002
		1080	904	3.93e-002	1.98e-003
		2560	2372	5.30e-002	1.63e-004
		5000	4790	1.97e-002	3.99e-005

### Slender tetrahedral element

Figure 23 shows the element subdivision of slender tetrahedral element for the specified source point for nearly singular domain integrals. Numerical results are presented in Table 2 which clearly show the superior convergence performance of the proposed method as compared to the conventional subdivision method.



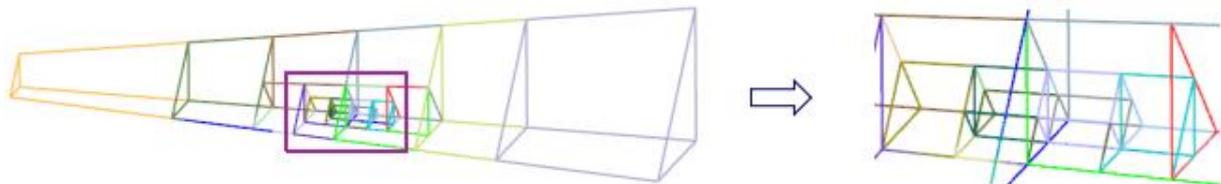
**Fig. 23.** The local parametric coordinate of source point is  $(-0.01, 0.3, 0.3)$ .

Table 2: Convergence of BTSM and CSM for nearly singular domain integrals  
with the source point at  $(-0.01, 0.3, 0.3)$

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM
Slender tetrahedral element	$(-0.01, 0.3, 0.3)$	375	355	$1.29e-001$	$2.83e-002$
		1536	1315	$2.74e-001$	$5.01e-003$
		3000	2426	$1.74e-001$	$1.95e-004$
		3993	3502	$4.89e-002$	$2.47e-005$

### Slender pentahedral element

Figure 24 shows the element subdivision of slender pentahedral element for the specified source point for nearly singular domain integrals. Numerical results are presented in Table 3 which clearly show the superior convergence performance of the proposed method as compared to the conventional subdivision method.



**Fig. 24.** The local parametric coordinate of source point is  $(0.0, 0.75, -0.01)$ .

Table 3: Convergence of BTSM and CSM for nearly singular domain integrals  
with the source point at  $(0.0, 0.75, -0.01)$

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM

		686	594	4.20e-001	9.56e-002
Slender		1458	1219	2.70e-001	1.51e-003
pentahedral	(0.0, 0.75, -0.01)	2662	2355	1.88e-001	3.26e-004
element		5288	4953	1.35e-003	1.12e-005

Numerical results in Figs. 22-24 verify the superior convergence performance of the BTSM and the quality of the element subdivision results in comparison with the CSM. Table 1 through Table 3 demonstrate that the accuracy of BTSM for integration is greatly improved with the increasing number of the integration points. With similar number of the integration points, it is appealing that significantly better accuracy can be achieved by the BTSM than the CSM. And above all, the results of the BTSM for integration are stable and acceptable for different types of volume elements. It clearly shows that the BTSM for nearly singular domain integrals has excellent properties of high accuracy and superior convergence.

## 6.2 Evaluation of nearly singular integrals with continuous kernel

### Slender hexahedral element

Figure 25 shows a typical example of the element subdivision of slender hexahedral element for nearly singular domain integrals with continuous kernel. Numerical results presented in Table 4 clearly show the superior performance of the proposed method as compared to the CSM.



**Fig. 25.** The local parametric coordinate of source point is (1.01, -0.9, -0.95).

Table 4: Numerical evaluation of nearly singular domain integrals with continuous kernel  
for slender hexahedral element

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM
Slender	(1.01, -0.9, -0.95)	1715	1497	1.81e-002	9.44e-005
hexahedral	(1.01, -0.5, -0.75)	2560	2372	5.30e-002	1.63e-004

element	(1.01, 0.5, -0.25)	2560	2488	1.43e-002	1.06e-004
	(1.1, 0.5, -0.25)	1715	1488	4.82e-003	4.18e-005
	(1.001, 0.5, -0.25)	3645	3136	3.80e-002	6.53e-004

### Slender tetrahedral element

A typical example of the element subdivision of slender tetrahedral element is presented in Fig. 26 for nearly singular domain integrals with continuous kernel. Numerical results with various relative locations of the source point in the local coordinate system of the element in Table 5 clearly show the superior performance of the proposed method as compared to the CSM.



**Fig. 26.** The local parametric coordinate of source point is (-0.01, 0.1, 0.1).

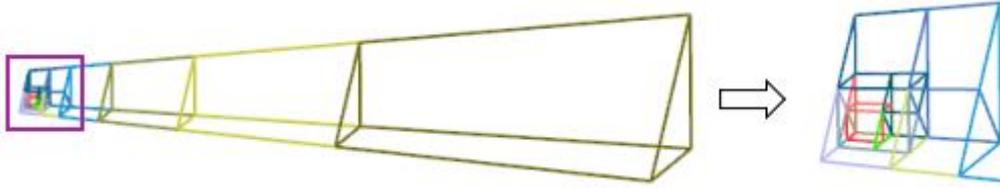
Table 5: Numerical evaluation of nearly singular domain integrals with continuous kernel  
for slender tetrahedral element

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM
Slender tetrahedral element	(-0.01, 0.1, 0.1)	2187	2051	3.94e-001	4.06e-004
	(-0.01, 0.3, 0.3)	3993	3502	4.89e-002	2.47e-005
	(-0.01, 0.5, 0.5)	2662	2673	4.69e-003	1.44e-005
	(-0.11, 0.3, 0.3)	2187	1763	3.57e-003	4.05e-005
	(-0.001, 0.3, 0.3)	3993	4042	2.43e-002	5.92e-004

### Slender pentahedral element

A typical example of the element subdivision of slender pentahedral element is presented in Fig. 27 for nearly singular domain integrals with continuous kernel. Numerical results with various relative locations of the source point in the local coordinate system of the element in Table 6 clearly show the

superior performance of the proposed method as compared to the CSM.



**Fig. 27.** The local parametric coordinate of source point is  $(-1.0, 1.0, -0.01)$ .

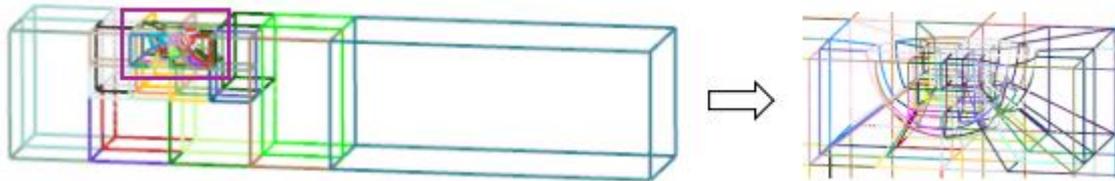
Table 6: Numerical evaluation of nearly singular domain integrals with continuous kernel for slender pentahedral element

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM
	$(-1.0, 1.0, -0.01)$	729	628	1.15e-002	8.57e-004
Slender pentahedral element	$(-0.5, 0.875, -0.01)$	1458	1652	1.66e-001	3.63e-004
	$(0.0, 0.75, -0.01)$	2000	1744	6.32e-002	3.64e-004
	$(0.0, 0.75, -0.11)$	1458	1398	1.77e-001	3.11e-004
	$(0.0, 0.75, -0.001)$	4394	4235	8.08e-002	2.32e-004

### 6.3 Evaluation of nearly singular integrals with discontinuous kernel

#### Slender hexahedral element

Figure 28 shows a typical example of the element subdivision of slender hexahedral element for nearly singular domain integrals with discontinuous kernel. Numerical results with various relative locations of the source point in the local coordinate system of the element presented in Table 7 clearly show the superior performance of the proposed method as compared to the CSM.



**Fig. 28.** The local parametric coordinate of source point is  $(1.01, -0.5, -0.75)$ .

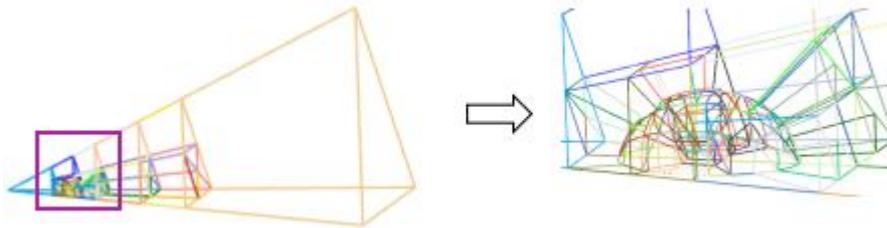
Table 7: Numerical evaluation of nearly singular domain integrals with discontinuous kernel for slender hexahedral element

Volume element	Source point	The total number of the integration points	Relative Error
----------------	--------------	--	----------------

		CSM	BTSM	CSM	BTSM
Slender hexahedral element	(1.01, -0.9, -0.95)	3645	3168	3.82e-002	1.02e-005
	(1.01, -0.5, -0.75)	3645	3236	6.62e-002	1.77e-004
	(1.01, 0.0, -0.5)	3645	3562	5.14e-002	4.60e-004
	(1.02, -0.5, -0.75)	3645	3188	6.08e-002	5.08e-004
	(1.001, -0.5, -0.75)	5000	4728	1.77e-002	2.19e-004

### Regular tetrahedral element

A typical example of the element subdivision of regular tetrahedral element is presented in Fig. 29 for nearly singular domain integrals with discontinuous kernel. Numerical results with various relative locations of the source point in the local coordinate system of the element in Table 8 clearly show the superior performance of the proposed method as compared to the CSM.



**Fig. 29.** The local parametric coordinate of source point is (-0.01, 0.1, 0.1).

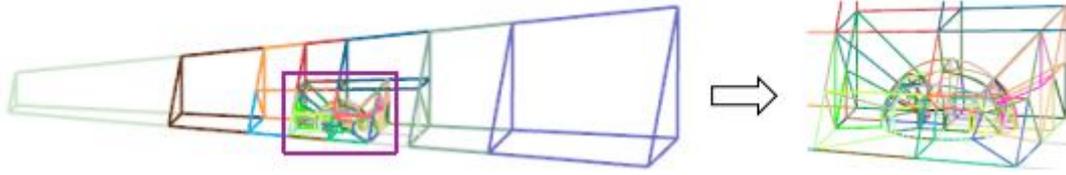
Table 8: Numerical evaluation of nearly singular domain integrals with discontinuous kernel for regular tetrahedral element

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM
Regular tetrahedral element	(-0.01, 0.1, 0.1)	3000	3190	3.48e-002	5.73e-004
	(-0.01, 0.5, 0.5)	3456	3669	4.45e-003	2.15e-005
	(-0.02, 0.1, 0.1)	3000	3123	5.72e-002	1.81e-005
	(-0.001, 0.1, 0.1)	5184	4832	3.88e-002	5.23e-004

### Slender pentahedral element

A typical example of the element subdivision of slender pentahedral element is presented in Fig. 30

for nearly singular domain integrals with discontinuous kernel. Numerical results with various relative locations of the source point in the local coordinate system of the element in Table 9 clearly show the superior performance of the proposed method as compared to the CSM.



**Fig. 30.** The local parametric coordinate of source point is  $(0.0, 0.75, -0.01)$ .

Table 9: Numerical evaluation of nearly singular domain integrals with discontinuous kernel for slender pentahedral element

Volume element	Source point	The total number of the integration points		Relative Error	
		CSM	BTSM	CSM	BTSM
Slender pentahedral element	$(-0.9, 0.975, -0.01)$	2662	2568	$6.05e-002$	$3.81e-004$
	$(0.0, 0.75, -0.01)$	3456	3382	$1.12e-001$	$3.54e-004$
	$(0.5, 0.625, -0.01)$	3456	3569	$2.91e-002$	$3.31e-005$
	$(0.0, 0.75, -0.02)$	2000	2046	$1.46e-001$	$6.08e-004$
	$(0.0, 0.75, -0.001)$	4394	4524	$1.22e-001$	$5.79e-004$

Numerical results in Figs. 25-30 demonstrate that well-shaped patches for hexahedral, tetrahedral and pentahedral element can be obtained with the BTSM. These examples verify the robustness of our algorithm and the quality of the element subdivision results in comparison with the CSM. Table 4 through Table 9 show that when the number of the integration points used is almost the same, the accuracy obtained by our method is 2 to 3 orders of magnitude higher than that by the CSM. In comparison with the CSM, significantly better accuracy is obtained by the BTSM for various locations of the source point.

## 7. Conclusions and future work

We have proposed a binary tree subdivision method (BTSM) for evaluation of nearly singular domain integrals with continuous or discontinuous kernel in BEM. By using the techniques of the binary-tree subdivision scheme, construction of the projection cavities and the cavity projection algorithm, well-shaped patches and excellent numerical results have been successfully obtained for

nearly singular domain integrals with continuous or discontinuous kernel. This subdivision method is more convenient to implement for arbitrary shape of the element and arbitrary location of the source point. Besides, the proposed method can guarantee the convergence of recursive subdivision based on a given terminating condition and generate patches successfully and efficiently under any circumstances.

In comparison with the CSM, the patches obtained are automatically refined as they approach the source point. The patches in the neighborhood of the source point are smaller than the distant ones. Besides, the average number of the quadrature points of patches are similar based on the quadrature rules. Thus, the integration points are set denser around the source point, and are sparsely distributed away from it. With the distinct feature that extensive unnecessary integration points can be avoided, the computational efficiency of domain integration is increased significantly by the BTSM. Significantly better accuracy is obtained by the BTSM for various locations of the source point. Based on excellent properties of the spline rules for integration [24-27], the work of Gaussian rules for spline space over simplices is ongoing, and the new results will be reported in near future. In addition, extension of our work to 3D curved volume element for nearly singular domain integrals and evaluating nearly singular domain integrals with discontinuous kernel including multiple discontinuous points using the proposed method is ongoing.

### **Acknowledgement**

This work was supported by National Science Foundation of China under grant number 11772125.

### **References**

- [1] J.H. Kane, *Boundary Element Analysis in Engineering Continuum Mechanics*, Prentice-Hall, Engelwood Cliffs, NJ, 1994.
- [2] J.M. Zhang, Qin XY, Han X, Li GY. A boundary face method for potential problems in three dimensions. *Int J Numer Meth Eng* 2009; 80(3): 320-337.
- [3] Zieniuk E. Potential problems with polygonal boundaries by a BEM with parametric linear functions[J]. *Engineering Analysis with Boundary Elements*, 2001, 25(3): 185-190.
- [4] Coda H B, Venturini W S. Three-dimensional transient BEM analysis[J]. *Computers & structures*, 1995, 56(5): 751-768.
- [5] Liu YJ. Analysis of shell-like structures by the boundary element method based on 3-D elasticity: formulation and verification. *Int J Numer Meth Eng* 1998; 41(3): 541-558.

- [6] ChenHaibo. Singular solutions of anisotropic plate with an elliptical hole or a crack. *Acta Mechanica Sinica*, 2005, 18(2):130-141.
- [7] Ma H, Kamiya N. A general algorithm for the numerical evaluation of nearly singular boundary integrals of various orders for two- and three-dimensional elasticity. *Computational mechanics* 2002; 29; 277–288.
- [8] Blázquez A, París F. Effect of numerical artificial corners appearing when using BEM on contact stresses[J]. *Engineering analysis with boundary elements*, 2011, 35(9): 1029-1037.
- [9] Dong C Y, Yang X, Pan E. Analysis of cracked transversely isotropic and inhomogeneous solids by a special BIE formulation [J]. *Engineering analysis with boundary elements*, 2011, 35(2): 200-206.
- [10] Peake M J, Trevelyan J, Coates G. Extended isogeometric boundary element method (XIBEM) for three-dimensional medium-wave acoustic scattering problems [J]. *Computer Methods in Applied Mechanics and Engineering*, 2015, 284: 762-780.
- [11] J.T. Chen, Y.T. Lee, Y.J. Lin, Analysis of multiple-spheres radiation and scattering problems by using null-field integral equations, *Appl Acoust*, 71 (2010;) 690–700.
- [12] Guo S, Li X, J.M. Zhang, et al. A triple reciprocity method in Laplace transform boundary element method for three-dimensional transient heat conduction problems[J]. *International Journal of Heat and Mass Transfer*, 2017, 114: 258-267.
- [13] Niu Zhongrong, Wang Xiuxi, et al. A new semi-analytical algorithm for the evaluation of the nearly singular integrals in 3-dimensional boundary element methods. *Computer Methods in Applied Mechanics and Engineering*, 2005, Vol. 195: 1057-1074.
- [14] Klees R. Numerical calculation of weakly singular surface integrals. *J Geodesy* 1996; 70(11):781-797.
- [15] Tanaka M, J.M. Zhang, Matsumoto T. Boundary-type meshless solution of potential problems: comparison between singular and regular formulations in hybrid BNM *Transactions of JASCOME, Journal of Boundary Element Methods* 2003; Vol.20, pp.21-26.
- [16] J.M. Zhang, Lu C, Zhang X, et al. An adaptive element subdivision method for evaluation of weakly singular integrals in 3D BEM. *Engineering Analysis with Boundary Elements*, 2015, 51:213-219.
- [17] Yin L, Luo X, Shephard M S. Identifying and meshing thin sections of 3-d curved domains[C]//*Proceedings of the 14th International Meshing Roundtable*. Springer, Berlin, Heidelberg, 2005: 33-54.
- [18] Bentley J L. Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the Acm*, 1975, 18(9):509-517.
- [19] Zhang H, Zhao G, Ma X. Adaptive generation of hexahedral element mesh using an improved grid-based method. *Computer-Aided Design*, 2007, 39(10):914-928.
- [20] Hammer, P. C., & Stroud, A. H. Numerical integration over simplexes. *Mathematical tables and other aids to computation*, 1956, 10(55), 137-139.
- [21] Stroud A H. Approximate calculation of multiple integrals[J]. 1971.
- [22] Kosinka J, Bartoň M. Gaussian quadrature for C1 cubic Clough–Tocher macro-triangles[J]. *Journal of Computational and Applied Mathematics*, 2019, 351: 6-13.
- [23] Roca X, Sarrate J, Huerta A. Surface Mesh Projection for Hexahedral Mesh Generation by Sweeping. *Proceedings International Meshing Roundtable*, 2004:169-180.

- [24] Barton, M., Calo, V. M., 2016. Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 305, 217-240.
- [25] Johannessen, K., 2017. Optimal quadrature for univariate and tensor product splines. *Computer Methods in Applied Mechanics and Engineering* 316, 84-99.
- [26] Barton, M., Calo, V. M., 2017. Gauss-Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis. *Computer-Aided Design* 82, 57-67.
- [27] Hiemstra, R., Calabro, F., Schillinger, D. and Hughes, T. J. R., 2017. Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 316, 966-1004.