International Journal of Computational Methods
Vol. 12, No. 5 (2015) 1550029 (30 pages)

DOI: 10.1142/S0219876215500292

Boundary Condition Related Mixed Boundary Element and its Application in FMBEM for 3D Elastostatic Problem

Yingjun Wang*, Xiaowei Deng*,§, Qifu Wang†, Zhaohui Xia† and Hua Xu‡

*Department of Structural Engineering
University of California, San Diego, 9500 Gilman Drive
Mail Code 0085, La Jolla, CA 92093, USA

†National CAD Support Software Engineering Research Center
Huazhong University of Science and Technology
Wuhan 430074, P. R. China

‡School of Civil Engineering
Southwest Jiaotong University
Chengdu 610031, P. R. China
§x8deng@eng.ucsd.edu

Received 7 July 2014
Accepted 25 March 2015
Published

A boundary condition (BC) related mixed element method is presented to address the corner problem in boundary element method (BEM) for 3D elastostatic problems. In this method, noncontinuous elements (NCEs) are only used at the displacement-prescribed corners/edges and continuous elements (CEs) in other places, which can decrease the degrees of freedom (DOFs) compared to the approach using NCEs at all corners/edges. Moreover, an automatic generation algorithm of BC related mixed linear triangular elements is implemented with the help of 3D modeling engine ACIS, and the boundary element analysis (BEA) is integrated into CAD systems. In order to solve large scale problems, the fast multipole BEM (FMBEM) with mixed elements is proposed and utilized in the BEA. The examples show that the node shift scheme adopting 1/4 is optimal and the BEM/FMBEM using mixed elements can produce more accurate results by only increasing a small number of DOFs.

Keywords: Boundary element method; fast multipole method; mixed boundary elements; corner/edge problems; 3D elastostatics.

1. Introduction

The boundary element method (BEM) [Mukherjee and Liu (2013)], due to its high accuracy and dimension reduction characteristics, has been widely applied in a variety of engineering problems including potential problem [Liu and Nishimura (2006)], elastostatics [Wilde and Aliabadi (1998)], fluid dynamics [Chahine and Kalumuck (1998)], thermodynamics [Dong *et al.* (2014); Prochazka and Lok (2012)]

and acoustics [Tadeu *et al.* (2001); Zheng *et al.* (2012)]. However, its application to some kinds of problems (e.g., elastostatics) poses great challenge when solving domains have sharp corners and/or edges. It is because that the outward normals are all different at the elements adjacent to the nodes at the corners or edges.

Over the past years, a number of methods have been presented to deal with these difficulties. Jaswon and Symm [1977] circumvented these difficulties by rounding off the sharp corners and edges, but this method could not get accurate results along original sharp corners and edges where stress concentration often occurs. Another method is "unique traction" assumption which assumes that the tractions at a node of an edge or a corner are unique [Lachat and Watson (1976)], but this method can give rise to significant errors [Gao and Davies (2002)]. Nevertheless, the most prevalent and efficient methods fall into the following two categories: multiple node methods and mixed element methods [Subia *et al.* (1995)]. The multiple node methods introduce additional nodes then use auxiliary equations to compute extra unknown tractions. Some researchers, such as Chaudonneret [1978], Zhang and Mukherjee [1991] obtained the auxiliary equations from Hooke's law, and some others, for example, Duenser [2001], Gao and Davies [2000] derived the auxiliary equations from stress symmetry and the equilibrium of differential equations. However, the multiple node methods are somehow artificial and could not be used for problems with variable geometries/boundary conditions (BCs) and multi-region problems [Beer *et al.* (2008)]. The mixed element methods use noncontinuous elements (NCEs) along the sharp edges and corners to circumvent different outward normals at those locations. Chai *et al.* and Mi and Aliabadi [1992] used these methods in 3D fracture mechanics analysis, Guzina *et al.* used these methods in 3D solid mechanics, and Gao and Davies [2000] adopted them in 3D multi-region problems.

Although the mixed element methods have more advantages than the other methods as the statement above, the automatic generation of mixed elements for any geometry has to be figured out before their application to practical engineering problems. Despite plenty of meshing approaches have been used in quadrilateral or triangular element generation, such as geometry decomposition approach, node connection approach, mapped element approach and grid-based approach [Ho-Le (1988); Owen (1998)], there is no fully automatic meshing approach which has been published to obtain noncontinuous/continuous mixed elements for arbitrary geometries as far as we know. Considering that element generation is one of the most important factors that influence the analytical time, and in some cases it may take up majority of the analytical time, any improvements or acceleration in the meshing procedure are of great industrial interest [Simpson *et al.* (2012)].

In this paper, BC related mixed triangular boundary elements and their automatic generation algorithm are presented for 3D elastostatic problems. The mixed boundary elements consist of two kinds of elements: NCEs are just distributed at the surface edges or corners where the displacements are prescribed, and continuous elements (CEs) are distributed in other places. In order to decrease total degrees

of freedom (DOFs) as many as possible, the NCEs are further divided into three types: one-node NCEs, two-node NCEs and three-node NCEs. The element generation algorithm meshes geometries with CEs based on ACIS Faceting function [Spatial (2004)], and then automatically uses different NCEs to replace the CEs at displacement-prescribed edges or corners with the help of ACIS feature recognition. Therefore, this element generation algorithm can be directly integrated into any CAD software with ACIS kernel. The linear triangular elements are used because they can accurately represent the geometries [Cayol and Cornet (1997)] and automatically be generated for geometries [Lee and Joun (2008)]. In order to solve large scale problems, the mixed element method is applied to the fast multipole BEM (FMBEM) [Wang and Yao (2005); Chen and Xiao (2009); Wang *et al.* (2013, 2015)] which can reduce the BEM's complexities of both time and space from $O(N^2)$ to $O(N)$.

The outline of this paper is as follows: In Sec. 2, the BEM in 3D elastostatics is presented including boundary integral equations, the element integration formulations for both linear triangular CEs and linear triangular NCEs and the explanation of problems of edges and corners. In Sec. 3, the detailed automatic generation algorithm for BC related mixed boundary elements is described. In Sec. 4, the theory and algorithm of the FMBEM with mixed elements is proposed. In Sec. 5, examples demonstrate the efficiency of the BC related mixed boundary elements and their application in FMBEM for 3D elastostatic problems. Finally, we summarize this paper and make further discussions in Sec. 6.

2. BEM in 3D Elastostatics

2.1. *Boundary integral equations*

Assuming the body forces are zero, the discrete boundary integral equations for 3D elastostatics can be written as [Gao and Davies (2002)]

$$\begin{aligned} c_{ij}(P)u_j(P) &= \sum_{e=1}^{M}\int_{S_e} U_{ij}(P,Q)t_j(Q)dS(Q) \\ &\quad - \sum_{e=1}^{M}\int_{S_e} T_{ij}(P,Q)u_j(Q)dS(Q) \\ &= \sum_{e=1}^{M}\left\{\sum_{a=1}^{k} t_j^a(Q)\int_{S_e} U_{ij}(P,Q)N_a(Q)dS(Q)\right\} \\ &\quad - \sum_{e=1}^{M}\left\{\sum_{a=1}^{k} u_j^a(Q)\int_{S_e} T_{ij}(P,Q)N_a(Q)dS(Q)\right\}, \end{aligned} \tag{1}$$

where a summation is to be carried out over the range of two identical subscripts, $i,j = 1$–$3, \alpha$ denotes the αth node in the k-node element, M is the number of

elements, S_e represents the integration domain of the element e, $N_\alpha(Q)$ is the interpolation function of the αth node at point Q, $u_j^\alpha(Q)$ and $t_j^\alpha(Q)$ are components in j direction of displacements and tractions at the αth node of the element e where Q belong to, $c_{ij}(P)$ are free-term coefficients, $U_{ij}(P,Q)$ and $T_{ij}(P,Q)$ are the Kelvin's fundamental solutions for displacements and tractions as follows [Brebbia *et al.* (1984)]

$$U_{ij}(P,Q) = \frac{1}{16\pi\mu(1-v)r}[(3-4v)\delta_{ij} + r_{,i}r_{,j}], \tag{2a}$$

$$T_{ij}(P,Q) = -\frac{1}{8\pi(1-v)r^2}\left\{\frac{\partial r}{\partial n}[(1-2v)\delta_{ij} + 3r_{,i}r_{,j}] - (1-2v)(r_{,i}n_j - r_{,j}n_i)\right\}, \tag{2b}$$

where v is Poisson's ratio, μ is shear modulus, δ_{ij} is the Kronecker delta ($i = j$, $\delta_{ij} = 1$; else, $\delta_{ij} = 0$), n is the outward normal of the S_e, r is the distance between the source point P and the field point Q, $(\ )_{,k}$ is the partial derivative in the direction k of coordinates.

To produce a closed set of equations, Eq. (1) is written for each node P in turn. It can be observed that the integrals in Eq. (1) are carried out over each element. Let us assume that the integrals have been computed with the result that

$$\int_{S_e} U_{ij}(P,Q)N_\alpha(Q)dS(Q) = G_{ij}^e(P,Q), \tag{3a}$$

$$\int_{S_e} T_{ij}(P,Q)N_\alpha(Q)dS(Q) = H_{ij}^e(P,Q). \tag{3b}$$

Equation (1) can be described as

$$c_{ij}(P)u_j(P) = \sum_{e=1}^{M}\sum_{\alpha=1}^{k} t_j^\alpha(Q)G_{ij}^e(P,Q) - \sum_{e=1}^{M}\sum_{\alpha=1}^{k} u_j^\alpha(Q)H_{ij}^e(P,Q). \tag{4}$$

Equation (1) also has the matrix representation

$$[\mathbf{c}]\{\mathbf{u}\} = [\mathbf{G}]\{\mathbf{t}\} - [\mathbf{H}]\{\mathbf{u}\}. \tag{5}$$

By merging the corresponding entries of $[\mathbf{c}]$ and $[\mathbf{H}]$, Eq. (5) can be condensed into the following form

$$[\mathbf{H}']\{\mathbf{u}\} = [\mathbf{G}]\{\mathbf{t}\}, \tag{6}$$

where $[\mathbf{H}'] = [\mathbf{c}] + [\mathbf{H}]$. Merging unknowns in vectors $\{\mathbf{u}\}$ and $\{\mathbf{t}\}$ and exchanging their coefficients in matrices $[\mathbf{H}']$ and $[\mathbf{G}]$, the final linear equation system is set up as

$$[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{b}\}, \tag{7}$$

where the vectors $\{\mathbf{x}\}$ and $\{\mathbf{b}\}$ contain all the unknown and known nodal components, and matrix $[\mathbf{A}]$ is the influence matrix. Since matrix $[\mathbf{A}]$ is dense and nonsymmetric, it needs $O(N^2)$ operations to compute the coefficients and $O(N^2)$ operations to solve the system by iterative solvers, which limits the scale of engineering

problems that the conventional BEM can solve. Therefore, fast manipole technique is combined with BEM to solve this problem (see Sec. 4).

2.2. *Element integration formulations*

According to location relationship between node P and element S_e, the element integrals in Eqs. (3a) and (3b) can be divided into nonsingular integrals (P is out of S_e) and singular integrals (P intersects S_e). Since triangular elements give the maximum meshing flexibility for complex geometries and are commonly used in the BEM, we will discuss the element integration formulations for both linear triangular CEs and linear triangular NCEs in this section.

2.2.1. *CE integration*

CEs are the elements which use the same functions for the element shape and the interpolations of physical quantities inside the elements. A linear triangular CE (vertices coincide with nodes) is shown in Fig. 1(a), and the relationships between a source point P (a node of an element) and an element S_e are shown in Figs. 1(b) and 1(c).

When point P is out of element S_e, the integrals in Eqs. (3a) and (3b) are nonsingular which can be computed by Hammer numerical quadrature method [Dunavant (1985)] after mapping the element to a dimensionless local coordinate system (ξ_1, ξ_2) as Fig. 2, then Eqs. (3a) and (3b) can be written as

$$\int_{S_e} U_{ij}(P,Q)N_\alpha(Q)dS(Q) = \int_0^1 \int_0^{1-\xi_2} U_{ij}(P,Q)N_\alpha(\xi_1,\xi_2)J(\xi_1,\xi_2)d\xi_1 d\xi_2, \tag{8a}$$

$$\int_{S_e} T_{ij}(P,Q)N_\alpha(Q)dS(Q) = \int_0^1 \int_0^{1-\xi_2} T_{ij}(P,Q)N_\alpha(\xi_1,\xi_2)J(\xi_1,\xi_2)d\xi_1 d\xi_2, \tag{8b}$$

where $J(\xi_1, \xi_2)$ is the Jacobian of the parametric domain.

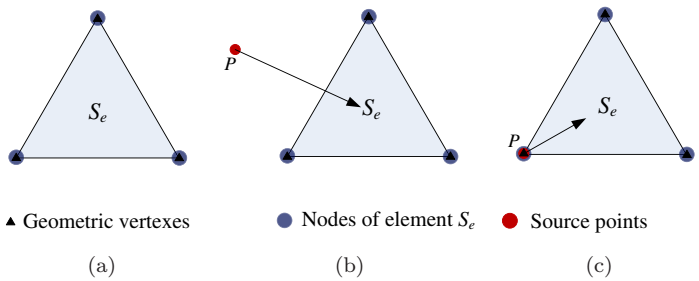

Fig. 1. A CE and its relationships with a source point: (a) a linear triangular CE S_e, (b) source point P is out of the element S_e, and (c) source point P coincides with a node of element S_e.

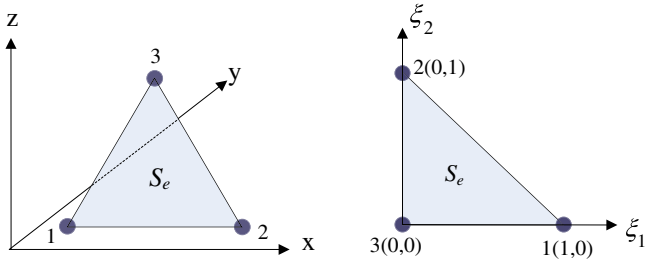

Fig. 2. Triangular element mapped from global coordinate system to local coordinate system.

The shape functions of element nodes are [Beer *et al.* (2008)].

$$N_1(\xi_1, \xi_2) = \xi_1, \tag{9a}$$

$$N_2(\xi_1, \xi_2) = \xi_2, \tag{9b}$$

$$N_3(\xi_1, \xi_2) = 1 - \xi_1 - \xi_2, \tag{9c}$$

which can be applied in the interpolations of geometric shape and physical quantities inside the elements.

When point P intersects element S_e, the integrals in Eqs. (3a) and (3b) are singular where Hammer numerical quadrature method cannot be used directly. An efficient singularity subtraction approach for CEs presented by Hu and Chen [1997] is used here, which maps triangular area from the local coordinate to another local coordinate system (ρ_1, ρ_2) (see Fig. 3) then uses the Jacobian of the coordinate transformation to remove $O(1/r)$ singularity, and we call this approach coordinate transformation singularity subtraction (CTSS).

The coordinate transformation formulations is shown as

$$\xi_1 = 1 - \rho_1^t, \tag{10a}$$

$$\xi_2 = \frac{1}{2}\rho_1^{t-1}(\rho_1 - \rho_2), \tag{10b}$$

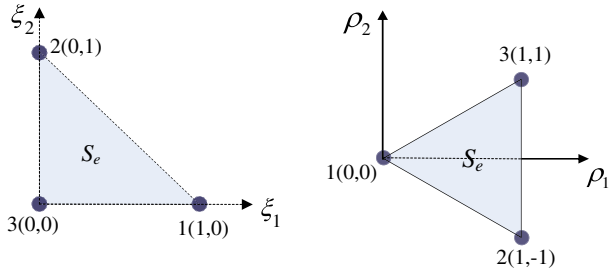

Fig. 3. Triangular element mapped from coordinate system (ξ_1, ξ_2) to another coordinate system (ρ_1, ρ_2).

where $t > 1$ (t is chosen to 2 in this paper, more details can be found in Hu and Chen [1997]), the Jacobian of the coordinate transformation from (ξ_1, ξ_2) to (ρ_1, ρ_2) is $J_2(\rho_1, \rho_2)$ which can remove $O(1/r)$ singularity, and

$$J_2(\rho_1, \rho_2) = \frac{1}{2} t \rho_1^{2(t-1)}. \tag{11}$$

Finally, the triangular element needs to be mapped to a coordinate system (ξ_1', ξ_2') with formulations

$$\rho_1 = 1 - \xi_1', \tag{12a}$$

$$\rho_2 = 1 - \xi_1' - 2\xi_2', \tag{12b}$$

and the Jacobian of the coordinate transformation $J_3(\xi_1', \xi_2')$ equals to 2. Hammer numerical quadrature formulations are used to compute all $G_{ij}^e(P, Q)$ and most of $H_{ij}^e(P, Q)$ except point P is coincident with the αth node of element S_e by the formulations as follows

$$\begin{aligned} &\int_{S_e} U_{ij}(P, Q) N_\alpha(Q) dS(Q) \\ &\quad = \int_0^1 \int_0^{1-\xi_2'} U_{ij}(P, Q) N_\alpha(\xi_1', \xi_2') J(\xi_1, \xi_2) \\ &\quad \quad \times J_2(\rho_1, \rho_2) J_3(\xi_1', \xi_2') d\xi_1' d\xi_2^1, \end{aligned} \tag{13a}$$

$$\begin{aligned} &\int_{S_e} T_{ij}(P, Q) N_\alpha(Q) dS(Q) \\ &\quad = \int_0^1 \int_0^{1-\xi_2'} T_{ij}(P, Q) N_\alpha(\xi_1', \xi_2') J(\xi_1, \xi_2) \\ &\quad \quad \times J_2(\rho_1, \rho_2) J_3(\xi_1', \xi_2') d\xi_1' d\xi_2^1. \end{aligned} \tag{13b}$$

The final shape functions of element nodes (under coordinate system (ξ_1', ξ_2')) are

$$N_1(\xi_1', \xi_2') = 1 - (1 - \xi_1')^t, \tag{14a}$$

$$N_2(\xi_1', \xi_2') = (1 - \xi_1')^{t-1} \xi_2', \tag{14b}$$

$$N_3(\xi_1, \xi_2) = (1 - \xi_1')^t - (1 - \xi_1')^{t-1} \xi_2'. \tag{14c}$$

When point P is coincident with the αth node of element S_e, the integral in Eq. (3b) has $O(1/r^2)$ singularity. However, we can avoid this $O(1/r^2)$ singular integral to get the diagonal block coefficients of $[\mathbf{H}']$ in Eq. (6) by rigid body movement method (RBMM) as [Gao and Davies (2002)]

$$[\mathbf{H}']_{ij}^{pp} = (\delta_{pq} - 1) \sum_{q=1}^{N} [\mathbf{H}']_{ij}^{pq}, \tag{15}$$

where N is the number of nodes, δ_{pq} is the Kronecker delta, the subscripts i and j range from 1 to 3, and the superscripts p and q refer to the nodes.

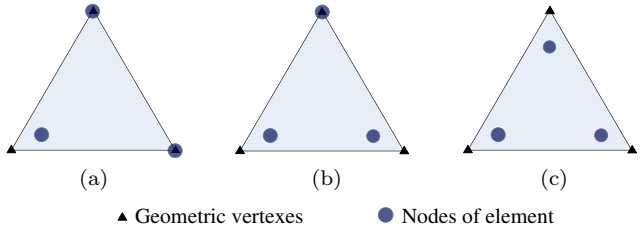

Fig. 4. Different types of NCEs: (a) one-node NCE, (b) two-node NCE and (c) three-node NCE.

2.2.2. *NCE integration*

The geometric vertices of NCEs used to define element geometric shape are the same as CEs, but the one or some nodes used for interpolations of physical quantities are moved inside the element. There are three types of NCEs shown in Fig. 4: one-node NCEs, two-node NCEs and three-node NCEs. The nodes inside the elements are called discontinuous nodes and the nodes coincide with vertices are called continuous nodes. Similar to CE integration, NCE integration also can be divided into nonsingular, $O(1/r)$ singular and $O(1/r^2)$ singular.

When point P is out of a NCE S_e, the integrals in Eqs. (3a) and (3b) are nonsingular which can be computed similar to Eqs. (8a) and (8b) by Hammer numerical quadrature method, but the shape functions for element nodes are different. We map the element to a dimensionless local coordinate system (ξ_1, ξ_2) as Fig. 2, and then can get the local coordinates of the nodes in different cases as Fig. 5.

In Fig. 5, $[n, m]$ is a marker which record the type of a NCE and locations of discontinuous node directly or indirectly, and s_1 and s_2 are used to represent local coordinates of discontinuous nodes. Letter n represents the n-node NCE, letter m means the mth node is a discontinuous node when n equals 1, is a continuous node when n equals 2, and is set to 3 when n equals 3 (all nodes are discontinuous nodes). The definition of s_1 and s_2 is

$$s_1 = \frac{l_2}{l_1 + l_2}, \tag{16a}$$

$$s_2 = \frac{l_1}{2(l_1 + l_2)}, \tag{16b}$$

where l_1 and l_2 are the node-vertex and node-side distances defined in Fig. 5. The shape functions of element nodes are derived in Table 1.

When source point P coincides with a geometric vertex of a NCE S_e, the integrals in Eqs. (3a) and (3b) are singular. The CTSS approach used in CEs (Eqs. (13a) and (13b)) can be directly used in element S_e to remove $O(1/r)$ singularity, and the final shape functions of element nodes are obtained from Table 1, respectively, where ξ_1 and ξ_2 are replaced with $1 - (1 - \xi_1')^t$ and $(1 - \xi_1')^{t-1}\xi_2'$. The RBMM in Eq. (15) can be used as the entries of $[\mathbf{H}']$'s diagonal block submatrices to circumvent $O(1/r^2)$ singular integrals.

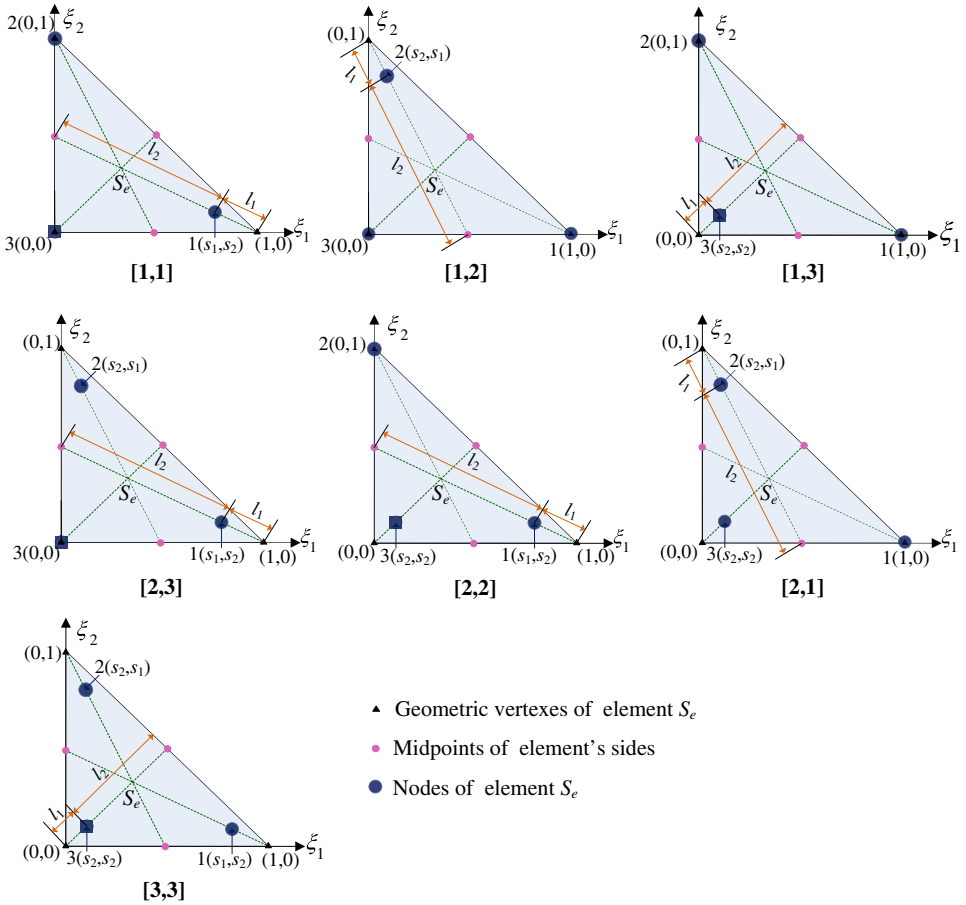

Fig. 5. Local coordinates of NCE nodes.

When point P coincides with a discontinuous node of a NCE S_e, the integrals in Eqs. (3a) and (3b) are also singular. In this case, the nodes and the interpolation points of Hammer numerical quadrature may be so close (distance of source point to an interpolation point is too small) that it may cause wrong results. In order to solve this problem, we divided element S_e into three continuous sub-elements $S_{e,1}, S_{e,2}$ and $S_{e,3}$ as Fig. 6, then the integrals are described as

$$\int_{S_e} U_{ij}(P,Q)N_\alpha(Q)dS(Q) = \sum_{k=1}^{3}\int_{S_{e,k}} U_{ij}(P,Q_k)N_\alpha^k(Q_k)dS(Q_k), \qquad (17a)$$

$$\int_{S_e} T_{ij}(P,Q)N_\alpha(Q)dS(Q) = \sum_{k=1}^{3}\int_{S_{e,k}} T_{ij}(P,Q_k)N_\alpha^k(Q_k)dS(Q_k), \qquad (17b)$$

Table 1. Shape functions of element nodes in different NCEs.

$[n,m]$	$N_1(\xi_1,\xi_2)$	$N_2(\xi_1,\xi_2)$	$N_3(\xi_1,\xi_2)$
$[1,1]$	$\frac{1}{s_1}\xi_1$	$-\frac{s_2}{s_1}\xi_1+\xi_2$	$\frac{s_2-1}{s_1}\xi_1-\xi_2+1$
$[1,2]$	$\xi_1-\frac{s_2}{s_1}\xi_2$	$\frac{1}{s_1}\xi_2$	$-\xi_1+\frac{s_2-1}{s_1}\xi_2+1$
$[1,3]$	$\frac{s_2-1}{2s_2-1}\xi_1-\frac{s_2}{2s_2-1}\xi_2+\frac{s_2}{2s_2-1}$	$-\frac{s_2}{2s_2-1}\xi_1+\frac{s_2-1}{2s_2-1}\xi_2+\frac{s_2}{2s_2-1}$	$\frac{1}{2s_2-1}\xi_1+\frac{1}{2s_2-1}\xi_2-\frac{1}{2s_2-1}$
$[2,1]$	$-\frac{1}{s_2-1}\xi_1+\frac{s_2}{s_2-1}$	$-\frac{s_2}{(s_1-s_2)(s_2-1)}\xi_1+\frac{1}{s_1-s_2}\xi_2+\frac{s_2}{(s_1-s_2)(s_2-1)}$	$\frac{s_1}{(s_1-s_2)(s_2-1)}\xi_1-\frac{1}{s_1-s_2}\xi_2-\frac{s_1}{(s_1-s_2)(s_2-1)}$
$[2,2]$	$\frac{1}{s_1-s_2}\xi_1-\frac{s_2}{(s_1-s_2)(s_2-1)}\xi_2+\frac{s_2}{(s_1-s_2)(s_2-1)}$	$-\frac{1}{s_2-1}\xi_2+\frac{s_2}{s_2-1}$	$-\frac{1}{s_1-s_2}\xi_1+\frac{s_1}{(s_1-s_2)(s_2-1)}\xi_2-\frac{s_1}{(s_1-s_2)(s_2-1)}$
$[2,3]$	$\frac{s_1}{s_1^2-s_2^2}\xi_1-\frac{s_2}{s_1^2-s_2^2}\xi_2$	$-\frac{s_2}{s_1^2-s_2^2}\xi_1+\frac{s_1}{s_1^2-s_2^2}\xi_2$	$-\frac{1}{s_1+s_2}\xi_1-\frac{1}{s_1+s_2}\xi_2+1$
$[3,3]$	$\frac{1}{s_1-s_2}\xi_1-\frac{s_2}{s_1-s_2}$	$\frac{1}{s_1-s_2}\xi_2-\frac{s_2}{s_1-s_2}$	$-\frac{1}{s_1-s_2}\xi_1-\frac{1}{s_1-s_2}\xi_2+\frac{s_1+s_2}{s_1-s_2}$

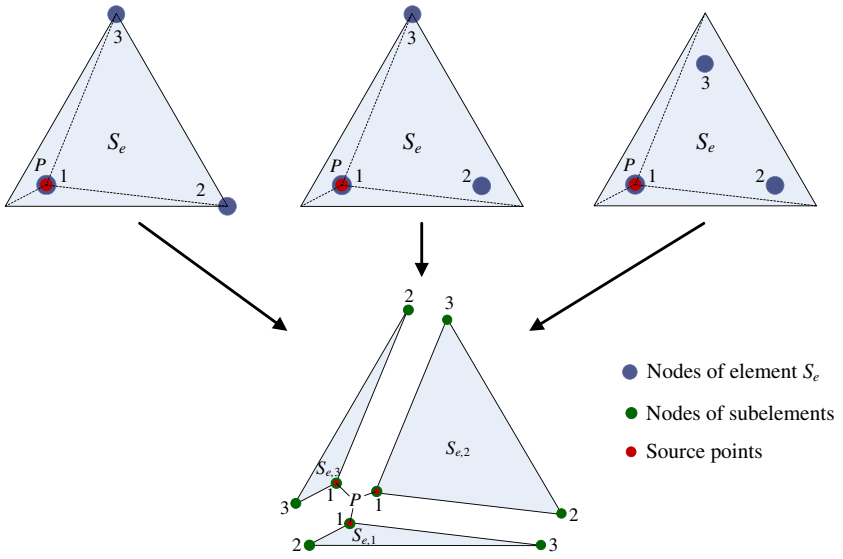

Fig. 6. Triangle subdivision.

where $N_\alpha^k(Q_k)$ is the interpolation function of the αth node of original NCE S_e at point Q_k of the sub-element $S_{e,k}$. In each sub-element, nodal displacements and tractions are represented by nodal value interpolation of element S_e with shape functions under the local coordinate system of element S_e, then $O(1/r)$ singular integrals of each sub-element can be computed by the CTSS approach as a CE (refer Eqs. (13a) and (13b)). The $O(1/r^2)$ singular integrals can also be circumvented by RBMM.

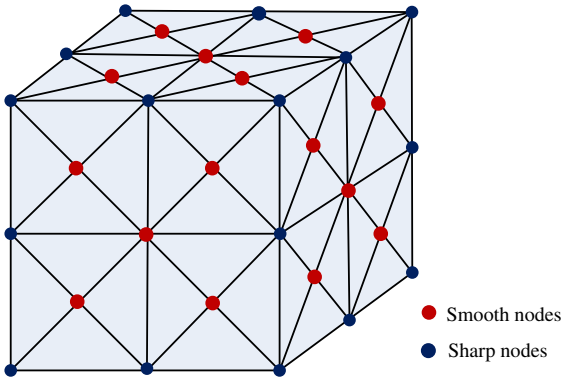

Fig. 7. (Color online) An example of smooth nodes and sharp nodes.

2.3. *Analysis of corner/edge problems*

The nodes on smooth surfaces (red nodes in Fig. 7), named smooth nodes by us, have unique displacements and tractions, and each node has three displacement components and three traction components (three components are unknown) for 3D elastostatic problems. The other nodes (blue nodes in Fig. 7) on sharp corners or edges, called sharp nodes in this paper, have unique displacements but multi-valued tractions, and each node has three displacement components and more than three traction components (three or more components are unknown). Since the boundary integral equations can generate only three equations per node, the number of equations at corner or edge nodes is insufficient if more than three unknown components need to be obtained at those nodes, which causes corner/edge problems.

Actually, when a domain with curved surfaces, such as a sphere or a cylinder, is discretized into linear elements, the normal vectors of the elements around a node are also not equal. It means that the element discretization error will also cause discontinuity of normal vectors at nodes. Fortunately, this discontinuity can be ignored in the BEM, since the normal vectors can be considered the same for the elements around a node when the element size is small enough, and the element refinement can reduce this discontinuity. Sometimes, using high-order elements still cannot completely remove this discontinuity, but a new computational method called isogeometric analysis [Hah *et al.* (2013); Hughes (2005)] can solve this discontinuous problems thoroughly, since the isogeometric analysis uses exact CAD model instead of discrete model for the analysis.

3. Generation Algorithm of BC Related Mixed Boundary Elements

3.1. *BC related mixed boundary elements*

In Sec. 2.3, the nodes are divided into smooth nodes and sharp nodes according to their locations. Although the tractions are multi-valued at sharp nodes because of

different outward normals on the intersectant surfaces, in most practical cases, the tractions are prescribed so that the unknowns are displacements which are uniquely defined and the number of equations is sufficient for the unknowns. It means that the corner/edge problems can be avoided by appropriately assembling the system equations according to BCs when tractions are prescribed at sharp nodes. Therefore, it may cause corner/edge problems when only displacements are prescribed at sharp nodes. According to the relationships between BCs and corner/edge problems, we present a BC related mixed boundary element approach to deal with corner/edge problems. In our approach, NCEs are just distributed at the surface edges or corners where the displacements are prescribed and CEs are distributed in other places.

A cube example of different element distribution strategies is shown in Fig. 8. Compared to the approach that distributes NCEs at all edges and corners, one obvious advantage of our approach is that the number of CEs which will increase nodes are greatly reduced (see Figs. 8(b) and 8(c)). When an original node is surrounded by discontinuous nodes, the node should be removed (see the removed nodes in Fig. 8).

3.2. *Automatic element generation algorithm*

The mixed element generation algorithm is implemented by developing the 3D CAD software InterSolid, whose geometric engine is ACIS, developed by National CAD Support Software Engineering Research Center of China. The 3D ACIS Modeler is a 3D modeling engine owned by Spatial Corporation, which is the geometry kernel of AutoCAD, Autodesk Inventor and many other well-known CAD applications [Kaufmann (2009)]. The mixed element generation algorithm can be divided into three major steps:

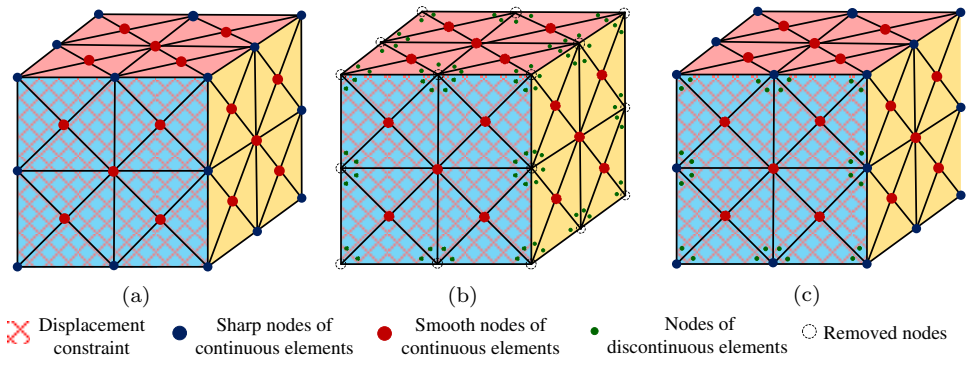

Fig. 8. Different element distribution strategies for a cube example: (a) distributing CEs on all surfaces, (b) distributing NCEs at all corners/edges, and (c) distributing NCEs at displacement-prescribed surface corners/edges.

(1) Meshing geometries with CEs (CE generation);
(2) Adding BCs where the displacement-prescribed nodes and elements at corners and edges are picked up (BC adding);
(3) Using NCEs to automatically replace the CEs at displacement-prescribed edges and corners (NCE generation). Triangular elements are used in our algorithm because they can accurately represent and automatically be generated for any geometry.

We clarify some common parameters that will be used throughout the following sections. The number of nodes in CEs is N_1, the number of nodes in NCEs is N_2, the number of elements is M. Table 2 lists general notations as follows.

3.2.1. *CE generation*

In ACIS, a component called Faceter generates and controls approximate polygonal representations. The faceting function of the Faceter component can generate

Table 2. General definition of notations.

Notation	Description
Node1[i][j]	Array elements, the jth global coordinate component of the ith node in CEs ($j = 0, 1$ and 2 present x-, y- and z-coordinate components).
Node2[i][j]	Array elements, the jth global coordinate component of the ith node in NCEs ($j = 0, 1$ and 2 present x-, y- and z-coordinate components).
Je[i][j]	Array elements. Je[i][0]–Je[i][2] are node numbers of element i. Je[i][3] is the type marker of element i (Je[i][3] = 0 for CE, Je[i][3] = 1, 2 and 3 for one-node NCE, two-node NCE and three-node NCE). Je[i][4] stores node distribution of element i (if Je[i][3] = 1, Je[i][4] is the local number of the only discontinuous node in element i; if Je[i][3] = 2, Je[i][4] is the local number of the only continuous node in element i; and Je[i][4] is set to 0 for all other cases). Je[i][5] is set to -1 if element i is a CE, and is the number in NCEs if element i is a NCE.
Aelist(i)	A list, adjacent elements of ith node in CEs.
U1[i][j]	Array elements, the jth displacement component of the ith node in CEs ($j = 0, 1$ and 2 present x-, y- and z-direction components).
U2[3*$i + k$][j]	Array elements, the jth displacement component of the kth node in the ith NCE ($j = 0, 1$ and 2 present x-, y- and z-direction components).
T[i][3*$k + j$]	Array elements, the jth traction component of the kth node in element i.
Enorm[i][j]	Array elements, the jth outward normal component of element i ($j = 0, 1$ and 2 present x-, y- and z-direction components).
Vnode[i]	Array elements, a marker marks that ith node in CEs should be removed (removed nodes in Fig. 8) or not when NCEs are generated (remove = 1, reserve = 0).
Nd[i]	Array elements, store node number of the ith displacement-prescribed corner/edge node.
Ed[i]	Array elements, store element number of the ith displacement-prescribed corner/edge element.

approximate polygonal representations for the surfaces of a body while maintaining edge consistency between adjacent surfaces. Surface faceting is performed by subdividing the surface in parameter space with a grid whose increments are determined through refinement parameters. The faceted representation of a surface is also called a mesh [Spatial (2004)] which are applied directly for meshing geometries to avoid writing a mesh algorithm and integrating it with CAD software. However, the original ACIS face meshing function does not completely satisfy the boundary element analysis (BEA) requirements. Some improvements are listed as follows: (1) optimize element quality by merging and remeshing poor quality elements; (2) define data structures to extract vertices' coordinates and relationships among elements and nodes; (3) compute outward normals of all elements. The CE generation is completed by ACIS inner functions which will not be discussed in detail here.

3.2.2. *BC adding*

This part interacts with user's operations between CE generation and NCE generation. In our code, the displacement BCs are assigned to nodes and the traction BCs are assigned to elements, which is similar to the BC assignation in BEMECH code written by Gao and Davies [2002]. The flowchart of a BC adding operation is shown in Fig. 9.

In the whole BC adding process, people interaction operations just exist in the surface selection, BC position selection and BC value assignation. ACIS inner function *api_entity_point_distance* [Spatial (2004)] which can obtain the distance between nodes and edges based on B-Rep [Marchetta and Forradellas (2010)] of geometries is used to identify displacement-prescribed nodes on edges. If the distance is 0, the

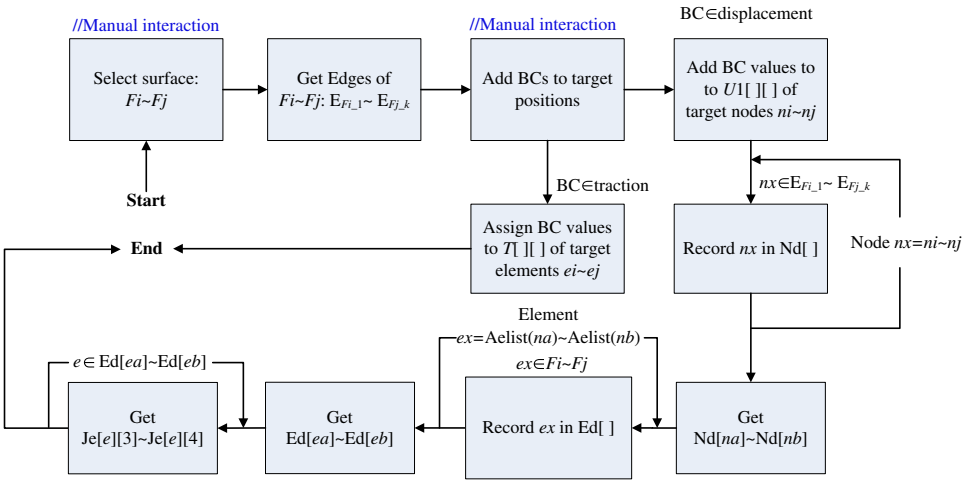

Fig. 9. The flowchart of a BC adding operation.

node is on an edge ($nx \in \mathrm{E}_{Fi\_1} \sim \mathrm{E}_{Fj\_k}$ in Fig. 9), and there is no need to identify nodes at corners because corners are edge intersections.

3.2.3. *NCE generation*

NCE generation is implemented by the element replacing algorithm (ERA) designed by us. Assume the number of nodes in CEs is N_1, the number of nodes in NCEs is N_2, the number of elements is M, and the number of CEs at corners/edges is nse and the number of nodes in Nd[ ] is *nsn*. Since all elements are CEs before ERA, Je[i][5] is set to -1 and Vnode[0]–Vnode[$N_1 - 1$] are set to 0. The pseudocode of ERA is as shown in Fig. 10.

ERA pseudocode

Start algorithm

If *nse*≠0 ***then***//Exist displacement-prescribed CEs at corners/edges
  Discontinuous node number flag N_d= N_1-1.
  For *i*=0 to *nse*-1 ***do***//Loop the elements which will be replaced by NCEs
    Get element number *oi*=Ed[*i*].
    Get original node numbers of element *oi*: *n*[*j*]=Je[*oi*][*j*], *j*=0, 1, 2.
    Transfer BCs: U2[*i**3+k][*j*]=U1[*n*[k]][*j*], *j*, *k*=0, 1, 2.
    Change element type: Je[*oi*][5]=*i*. //Element *oi* is a NCE and stored the NCE number
    If Je[*oi*][3]=1***then***//One-node NCE
      Get the discontinuous node's local number: *ni*=Je[*oi*][4].
      For *j*=0 to 2 ***do*** //Loop nodes of element *oi*
        Compute Node2[*i*][*j*] through interpolation of Node1[*n*[*k*]][*j*], *k*=0, 1, 2.
      Update node number: Je[*oi*][*ni*]=N_d+1, then $N_d = N_d$ +1.
    If Je[*oi*][3]=2 ***then***//Two-node NCE
      Get the continuous node's local number:*ni*=Je[*oi*][4].
      Get discontinuous node's local number: *ndi*1 and *ndi*2.
      For *j*=0 to 2 ***do*** //Loop nodes of element *oi*
        Compute Node2[*i*][*j*] through interpolation of Node1[*n*[*k*]][*j*], *k*=0, 1, 2.
      Update node number: Je[*oi*][*ndi*1]=N_d+1, Je[*oi*][*ndi*2]=N_d+2, then $N_d = N_d$ +2.
    If Je[*oi*][3]=3 ***then***//Three-node NCE
      For *j*=0 to 2 ***do*** //Loop nodes of element *oi*
        Compute Node2[*i*][*j*] through interpolation of Node1[*n*[*k*]][*j*], *k*=0, 1, 2.
      Update node number: Je[*oi*][*j*]=N_d+*j*+1, then N_d= N_d +3.
  The number of nodes in NCEs: N_2=N_d+1.
If *nsn*≠0 ***then***//Exist displacement-prescribed nodes at corners/edges
  For *i*=0 to *nsn*-1 ***do*** //Loop nodes of displacement-prescribed corner/edge elements
    Get node number *nn*=Nd[*i*].
    Set a flag of node removing: *nflag*=0.
    For each element e*i*Aelist(*nn*) ***do*** //Loop adjacent elements of node *nn*
      Loop nodes of element *ei*, if *nn* does not belong to any nodes of element *ei*,
      remove node *nn*.

End algorithm

Fig. 10. The pseudocode of ERA.

The ERA is automatically implemented by a computer, after which different types of NCEs are generated to replace the displacement-prescribed CEs at corners and edges. It needs to notice that the NCEs which will increase the total number of nodes but will not increase the number of elements. Subsequently, the data can be used for BEA with BC related mixed boundary elements. The whole generation algorithm of BC related mixed boundary elements is completed.

4. FMBEM with BC Related Mixed Elements

The FMBEM is a combination of BEM and fast multipole method (FMM), which adopts the hierarchical tree data structure and reduces the BEM's complexities of both time and space from $O(N^2)$ to $O(N)$. The key techniques in the FMBEM are the Multipole Expansions (ME), Local Expansions (LE) and the translations between different expansions including Moment-to-Moment (M2M) translation, Moment-to-Local (M2L) translation and Local-to-Local (L2L) translation. The formulations of the FMBEM for 3D elastostatics are listed briefly below.

4.1. *Fundamental formulations*

In this section, we only give a brief introduction to the FMBEM formulations, and more detail can be found in Yoshida [2001] and Liu [2009]. ME is a series expansion to separate a source point from a field point by a middle point that is close to the field point, which converts the integration of fundamental solutions in Eq. (1) on an element S_e, as follows

$$\begin{aligned}
&\int_{S_e} U_{ij}(P,Q)t_j(Q)\,dS(Q)\\
&\quad = \frac{1}{8\pi\mu}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}[F^S_{ij,n,m}(\overrightarrow{Q_cP})\overline{M^1_{j,n,m}}(Q_c)\\
&\qquad + G^S_{i,n,m}(\overrightarrow{Q_cP})\overline{M^1_{n,m}}(Q_c)], \qquad (18a)\\
&\int_{S_e} T_{ij}(P,Q)u_j(Q)\,dS(Q)\\
&\quad = \frac{1}{8\pi\mu}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}[F^S_{ij,n,m}(\overrightarrow{Q_cP})\overline{M^2_{j,n,m}}(Q_c)\\
&\qquad + G^S_{i,n,m}(\overrightarrow{Q_cP})\overline{M^2_{n,m}}(Q_c)], \qquad (18b)
\end{aligned}$$

where $(\overrightarrow{Q_cP})$ is a vector from Q_c to P, $\overline{M^k_{n,m}}(\boldsymbol{Q}_c)$ and $\overline{M^k_{j,n,m}}(\boldsymbol{Q}_c)$ are the complex conjugates of $M^k_{n,m}(\boldsymbol{Q}_c)$ and $M^k_{j,n,m}(\boldsymbol{Q}_c)$ ($k = 1$ or 2), and the formulations of ME moments and their coefficients $F^S_{ij,n,m}(\overrightarrow{\boldsymbol{Q}_c\boldsymbol{P}})$ and $G^S_{i,n,m}(\overrightarrow{\boldsymbol{Q}_c\boldsymbol{P}})$ are given in Appendix A.

The ME center can be moved from Q_c to $Q_{c'}$ by M2M translations

$$M^k_{n,m}(Q_{c'}) = \sum_{n'=0}^{n} \sum_{m'=-n'}^{n'} R_{n',m'}(\overrightarrow{Q_{c'}Q_c})[M^k_{n-n',m-m'}(Q_c) + (\overrightarrow{Q_{c'}Q_c})_j M^k_{j,n-n',m-m'}(Q_c)], \tag{19a}$$

$$M^k_{j,n,m}(Q_{c'}) = \sum_{n'=0}^{n} \sum_{m'=-n'}^{n'} R_{n',m'}(\overrightarrow{Q_{c'}Q_c}) M^k_{j,n-n',m-m'}(Q_c). \tag{19b}$$

LE is similar to ME, which expands the U/T kernel integrals in Eq. (1) as

$$\int_{S_e} U_{ij}(P,Q)t_j(Q)\,dS(Q) = \frac{1}{8\pi\mu} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} [F^R_{ij,n,m}(\overrightarrow{P_cP}) L^1_{j,n,m}(P_c) + G^R_{i,n,m}(\overrightarrow{P_cP}) L^1_{n,m}(P_c)], \tag{20a}$$

$$\int_{S_e} T_{ij}(P,Q)u_j(Q)\,dS(Q) = \frac{1}{8\pi\mu} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} [F^R_{ij,n,m}(\overrightarrow{P_cP}) L^2_{j,n,m}(P_c) + G^R_{i,n,m}(\overrightarrow{P_cP}) L^2_{n,m}(P_c)], \tag{20b}$$

where $G^R_{i,n,m}$ and $F^R_{ij,n,m}$ are obtained from Eqs. (A.1) and (A.2) with $S_{n,m}$ replaced with $R_{n,m}$, and the local moments are prescribed by the M2L translations

$$L^k_{n,m}(P_c) = (-1)^n \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} \overline{S_{n+n',m+m'}}(\overrightarrow{Q_cP_c})[M^k_{n',m'}(Q_c) - (\overrightarrow{Q_cP_c})_j M^k_{j,n',m'}(Q_c)], \tag{21a}$$

$$L^k_{j,n,m}(P_c) = (-1)^n \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} \overline{S_{n+n',m+m'}}(\overrightarrow{Q_cP_c}) M^k_{j,n',m'}(Q_c). \tag{21b}$$

When the LE point is moved from P_c to $P_{c'}$, we can use the following L2L translations

$$L^k_{n,m}(P_{c'}) = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} R_{n'-n,m'-m}(\overrightarrow{P_cP_{c'}})[L^k_{n',m'}(P_c) - (\overrightarrow{P_cP_{c'}})_j L^k_{j,n',m'}(P_c)], \tag{22a}$$

$$L^k_{j,n,m}(P_{c'}) = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} R_{n'-n,m'-m}(\overrightarrow{P_cP_{c'}}) L^k_{j,n',m'}(P_c). \tag{22b}$$

4.2. *The algorithm of the FMBEM*

The FMBEM algorithm consists of tree construction, upward pass and downward pass. The tree structure is used to store data and the upward pass and downward pass are the major processes of the FMBEM.

4.2.1. *Tree construction*

A improved dual-information adaptive tree [Wang *et al.* (2013)] is used here, the tree of boxes containing both node and element information (including both continuous and discontinuous nodes and elements) is constructed by dividing the problem domain into smaller sub-domains recursively until the number of nodes in each sub-domain is less than a prescribed number. At the level 0, we define a cube box containing the whole domain. Boxes on level $l+1$ are obtained by subdividing each box on level l into eight equal child boxes, then trim off empty child boxes. In order to implement the adaptive scheme, four lists associated with box b are defined as follows:

(1) *List* 1 (only for childless box b), denoted by $L_1(b)$, is the set consisting of b and all childless boxes adjacent to b.
(2) *List* 2, denoted by $L_2(b)$, is the set consisting of all children of the adjacent boxes of b's parent that are well separated from b.
(3) *List* 3 (only for childless box b), denoted by $L_3(b)$, is the set consisting of all descendents of b's adjacent boxes that are not adjacent to b.
(4) *List* 4, denoted by $L_4(b)$, is the counterpart of *List* 3, if $b \in L_3(c)$ then $c \in L_4(b)$. Note that all boxes in $L_4(b)$ are childless.

Fig. 11 shows the four lists for a box b in 2D where f represents the far field boxes of b (f will not be computed directly), and the lists for 3D problems can be obtained in the same way by replacing squares with cubes.

4.2.2. *Upward pass*

Upward pass can be divided into two steps: (1) compute multipole moment of each childless box b at its center by accumulating all element integrals in b; and (2) compute multipole moments of other boxes whose child boxes are not empty by M2M translation from the penultimate level to level 2. All the multipole moments are obtained except the boxes belong to level 0 and level 1.

4.2.3. *Downward pass*

Downward pass can be divided into following steps:

(1) From level 2 to the lowest level, loop over each box b of level l, M2L is used to translate the multipole moments of $L_2(b)$ to the local moment of b. Then, from

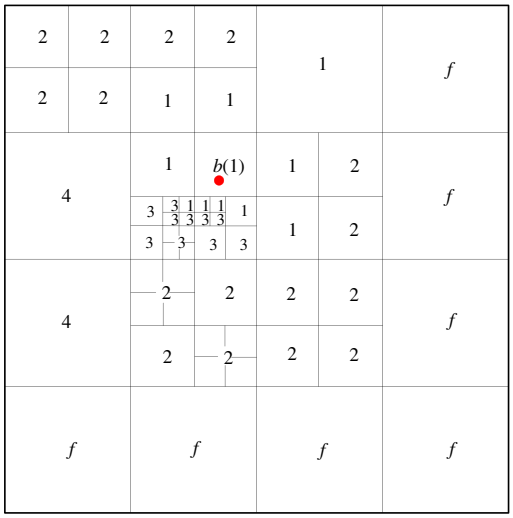

Fig. 11. Four lists for box b.

level 3 to the lowest level, L2L translation is used to translate local moment of b's parent to b for each box b of level l, and the translated local moment is added to the original local moment of b. Finally, LE are used to compute the U/T kernel integrals at each node of each childless box b.

(2) Loop over each childless box b, direct computation as conventional BEM is used to compute the U/T kernel integrals from elements of each box in $L_1(b)$ to each node in b, which is added to the original integrals.

(3) Loop each childless box b, ME (the number of nodes in b is larger than the square of expansion term p^2) or direct computation as conventional BEM (the number of nodes in b is not larger than p^2) is used to compute the U/T kernel integrals from elements of each box in $L_3(b)$ to each node in b, which will be added to the original integrals.

(4) From level 3 to the lowest level, loop each box b of level l, direct computation as conventional BEM is used to compute the U/T kernel integrals from elements of each box in $L_4(b)$ to each node in b, which will be added to the original integrals.

The major difference between the FMBEM with mixed elements and the conventional FMBEM is that the direct computation needs to utilize different strategies according to different types of nodes and elements (see Sec. 2.2. Element integration formulations). The node-element integrals can be divided into eight types as Fig. 12. For types (1) and (5), directly use the CE integration strategies in Sec. 2.2.1. CE integration. Note that there is not any singular integral in type (5). For types (2)–(4) and (6)–(8), the NCE integration strategies in Sec. 2.2.2. are used. The difference

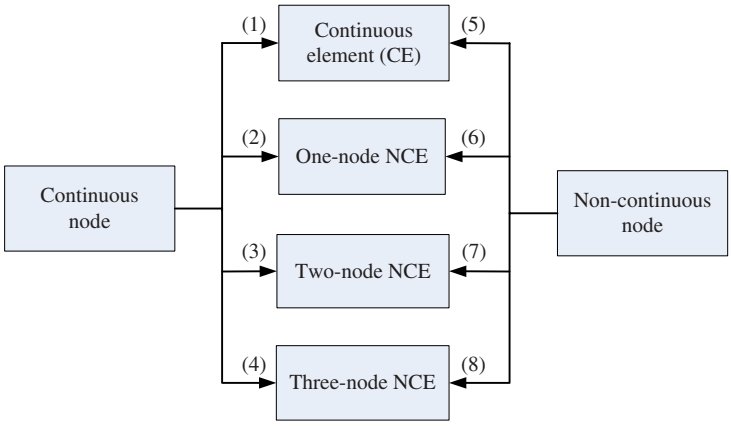

Fig. 12. Different types of node-element integrals.

between types (2)–(4) and types (6)–(8) is that it needs to divide the element into sub-elements to solve the singular integral part in types (6)–(8) as Fig. 6.

4.2.4. *Solution*

Generalized minimum residual (GMRES) method is used to solve Eq. (7) because of its high efficiency for solving dense nonsymmetric systems [Xiao and Chen (2007)]. When GMRES method is used for the FMBEM, the only difference to the conventional GMRES method is that the matrix–vector multiplication $[\mathbf{A}]\{\mathbf{v}_i\}$ in each iterative step ($\mathbf{v}_i$ is orthonormal basis vector, details can be found in Saad Y, Schultz [1986]) is computed by the FMBEM. Block diagonal matrices based on each childless box in the FMBEM are used as the preconditioner according to the description in Wang *et al.* [2005] to accelerate convergences.

5. Examples and Analysis

After the generation of BC related mixed boundary elements, the main program, called MixBEM or MixFMBEM by us, we directly adopts the mixed boundary elements to compute the unknown displacements and tractions by conventional BEM or FMBEM according to the chosen method, and then compute the stresses of boundary nodes. In MixBEM and MixFMBEM, 7 point Hammer numerical quadrature method is used for triangular element integration. When the source point is very close to the element being integrated, the integration is nearly singular [Beer *et al.* (2008)], and 16 point Hammer quadrature rule is used to reduce the integration error. The GMRES method is used to solve the equation system as Eq. (7), and the preconditioner is the diagonal matrix of matrix $[\mathbf{A}]$ in Eq. (7) for BEM and block diagonal submatrices based on each childless box for FMBEM.

MixBEM and MixFMBEM are integrated into the 3D CAD software InterSolid whose kernel is ACIS that the CAD modeling, BC adding, mixed boundary element generation and BEA can be implemented in the same software environment. All the programs are executed on a laptop: CPU is Intel Core i7-3630QM (2.40GHz, 1 core is used), OS is Windows 7, RAM is DDR3 SDRAM (4 GB), the compiler is Microsoft Visual Studio 2008 (C/C++).

5.1. *Example* 1

In the first example, we solve a cube model ($20 \times 20 \times 20$ mm) by conventional BEM in which the traction is 10 MPa on the top surface, and the bottom surface, Line 1 and Line 2 are constrained at Z-direction, X-direction and Y-direction, respectively (see Fig. 13). Elastic modulus is 37.5 GPa and Possion's ratio is 0.25. The cube surfaces are meshed with 84 linear triangular elements (see Fig. 14), the NCEs are only generated at the edges of the bottom surface, and all nodes of CEs are reserved in this case. The tolerance for convergence of the GMRES solution is set to 10^{-5}.

The node shift c, which is the ratio of the node-moved distance to half the length of the node-moved route in an element ($c = 2l_1/(l_1 + l_2)$ in Fig. 15), determines how the node locations of NCEs influences the computational effort and numerical accuracy. For quadrilateral elements, the node shift is presented with a similar definition as triangular elements shown in Fig. 15, Mi and Aliabadi [1994] chose the node shift as 1/2 and the error less 4%, Subia *et al.* [1995] considered that the node shift equals to 1/3 is the best but between 1/20 and 3/5 is relatively constant for the results, Guzina *et al.* [2006] regarded the node shift in the range 1/20–1/4 is optimal. In view of the above node shift selection, we select node shift as 1/20, 1/10, 1/6, 1/4, 1/3, 1/2 and 3/5, and compare the stress results at Z-direction (traction adding direction) with analytical solution. We adopt two ways to compare accuracy of the results: one is selecting 14 different location nodes (marked in Fig. 14) to

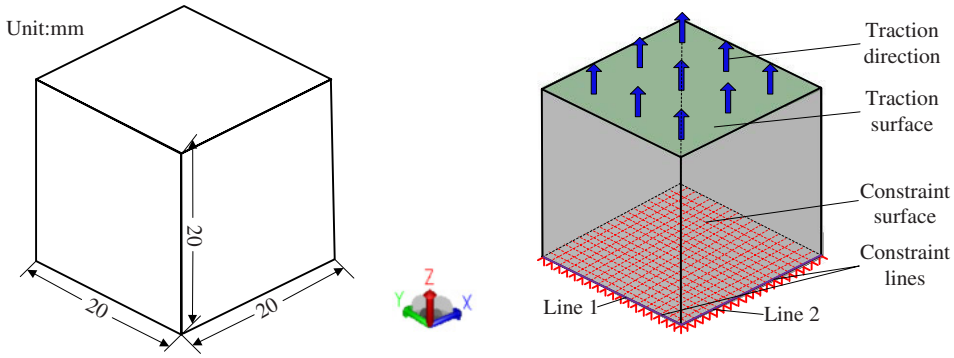

Fig. 13. Cube model and its BCs.

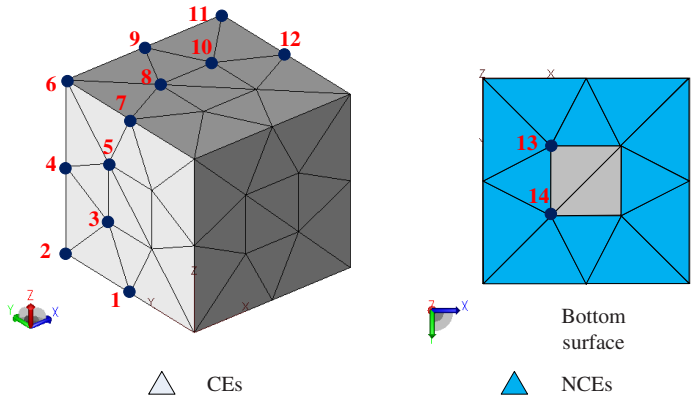

Fig. 14. Mesh of cube and nodal path.

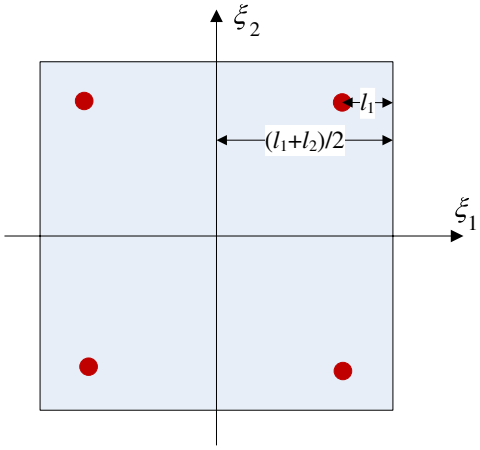

Fig. 15. Node shift of quadrilateral elements.

compare the detailed errors (see Fig. 16), and the other is using the mean relative error (MRE) for all the 14 nodes by the equation

$$\mathrm{Err}\sigma_z(\%) = \frac{1}{N_1}\sum_{i<N_1}\left|\frac{\sigma_z^{\mathrm{BEM}} - \sigma_z^{\mathrm{Analytical}}}{\sigma_z^{\mathrm{Analytical}}}\right| \times 100\%. \tag{23}$$

From Table 3, it can be found out that the MRE is minimum (0.47%) when the node shift is 1/4, and all MREs remain relatively stable for node shifts in the range 1/20–3/5 (less than 4%). The solution is convergent in small iterative number when node shift is in the range 1/4–3/5, and iterative number increases when node shift is longer or shorter. Considering both MRE and iterative number, node shift should be selected to 1/4 as the best choice. From Fig. 16, we find out that big errors occur at node 1, node 2, node 13 and node 14. The reason of the big errors is that these

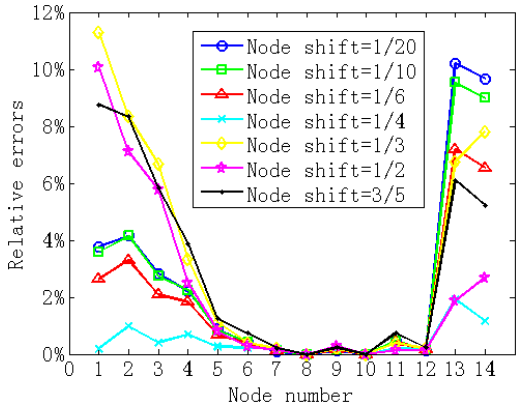

Fig. 16. Relative errors of selected nodes with different node shift schemes.

Table 3. MREs and solution iterative number of different node shift schemes.

Node shift	1/20	1/10	1/6	1/4	1/3	1/2	3/5
MRE (%)	2.49	2.38	1.82	0.47	3.33	2.26	2.97
Iterative number	53	43	35	33	33	29	33

nodes are located at the intersection between CEs and NCEs where the accuracy is lower.

5.2. *Example* 2

The second example is a valve part which is solved by FMBEM. The size and BCs are shown in Fig. 17, and the traction is 3 MPa on the irregular torus surface and all displacements are constrained on the four axial symmetry hole surfaces. Elastic

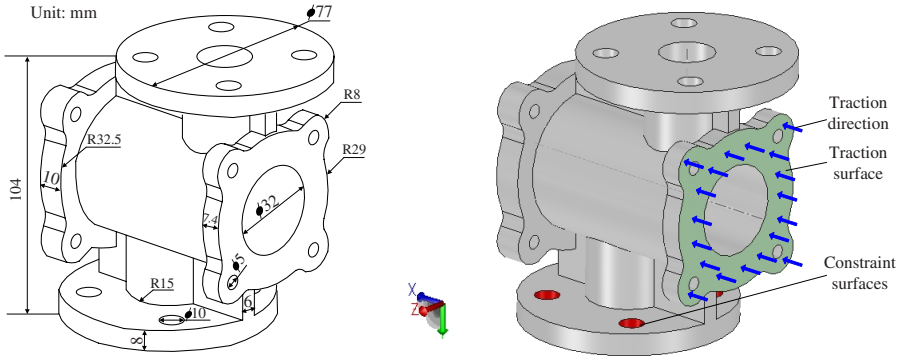

Fig. 17. Valve part model and its BCs.

modulus is 260 GPa, Possion's ratio is 0.3. The surfaces are meshed with 33,344 linear triangular elements for both MixFMBEM and "unique traction" FMBEM (only has linear triangular CEs and assumes the tractions of different elements at the same node are equal), and the node shift of MixFMBEM is selected to 1/4 according to the MRE and iterative number in Table 3. The total numbers of DOFs generated by MixFMBEM and "unique traction" BEM are 53,127 and 52,551, respectively. It can be seen that the increment of the number of DOFs is small by our MixFMBEM. The max node number allowed in a childless box is 25 and the number of expansion terms is 6. The tolerance for convergence of the GMRES solution is set to 10^{-3}. Since there is no analytical solution for such an irregular solid, we use finite element solution by commercial software ANSYS as benchmark solution. In ANSYS, 210,361 linear and 199,014 quadratic tetrahedral elements are generated with the same element size control as the FMBEMs, respectively, which are denoted by ANSYS-1 and ANSYS-2 in follows.

The iterative number, total time and total memory of different methods are listed in Table 4. The memory of ANSYS-2 contains two numbers which is divided by symbol "‖", the front one is actual memory allocated for solver which uses the out-of-core memory mode (data need to be stored in hard disk temporarily), and the other one is memory required for in-core mode (data only are stored in memory) which is adopted in ANSYS-1. Symbol "—" means ANSYS uses direct solver where iterative number does not exist.

From Table 4, it can be seen that the iterative number, total time and total memory of MixFMBEM are larger than those of FMBEM but the increment is not big. The reason for this is that the NCEs are introduced in MixFMBEM will increase the DOFs and complexity of direct computation part in the FMBEM (see Sec. 4.2.3). The memory use in FMBEMs is much lower than that in ANSYS because the dimension reduction characteristics help FMBEMs generate fewer elements. Note that most of memory in the FMBEM are used to store the coefficients of the preconditioner and solver. Taking the MixFMBEM as example, the memory used for the coefficients of the preconditioner and solver is nearly 270MB, and the memory shared by the FMM coefficients (multipole and local moments) is approximately 15MB and the other data such as nodes, elements BCs share the left memory.

The displacement results are shown in Fig. 18, in which, we can get that the displacement distributing tendency of the three ways is the same. In order to compare the detailed displacement results computed by the different methods,

Table 4. The iterative number, total time and total memory of different solution methods.

Solution method	Iterative number	Total time (s)	Total memory (MB)
MixFMBEM	80	1412.6	296.8
"Unique traction" FMBEM	66	1029.2	231.2
ANSYS-1	—	236.9	957.4
ANSYS-2	—	618.3	1221.8‖7150.2

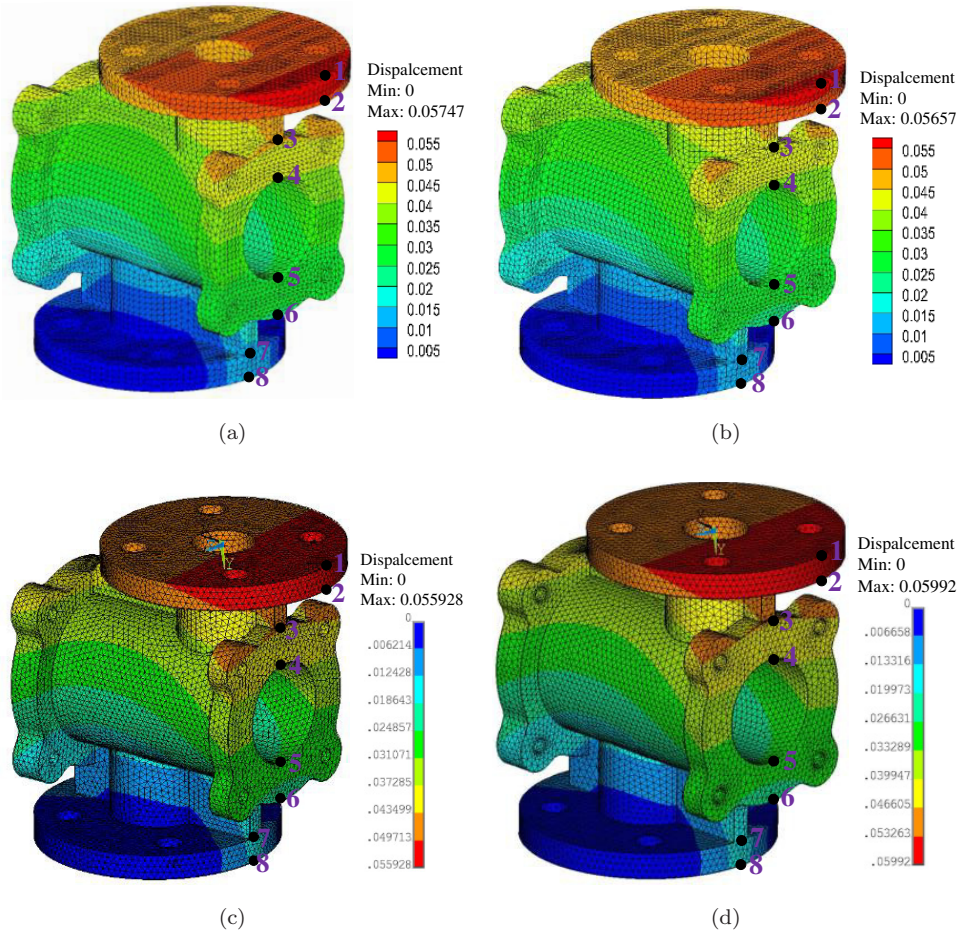

Fig. 18. (Color online) Displacement at the traction adding direction. (a) MixFMBEM, (b) "unique traction" FMBEM, (c) ANSYS-1 and (d) ANSYS-2.

we select eight nodes at different places (see Fig. 18) and compare their results in Fig. 19, and MREs of the selected nodes (MixFMBEM/ANSYS-2, "unique traction" FMBEM/ANSYS-2 and ANSYS-1/ANSYS-2) are 3.1, 5.6 and 7.5%, respectively. In general, the quadratic element is more accurate than the linear element. Taking ANSYS-2 result as criterion, we can sequence the accuracy of the results from different methods as follows: ANSYS-1<"unique traction" FMBEM<MixFMBEM<ANSYS-2. This proves that MixFMBEM can obtain higher accuracy than the "unique traction" FMBEM. Note that both of the FMBEMs use linear elements obtain higher accuracy than ANSYS-1, which reflects that the BEM is more accurate than the finite element method (FEM).

The Von Mises stress results are shown in Fig. 20. We come to the similar conclusion as that of the displacement results but the stress error is bigger,

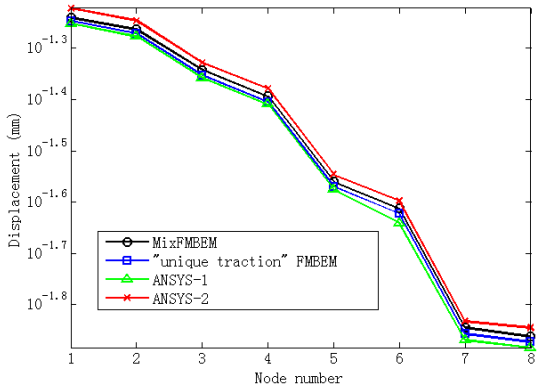

Fig. 19. Displacement of the selected nodes in Fig. 18.

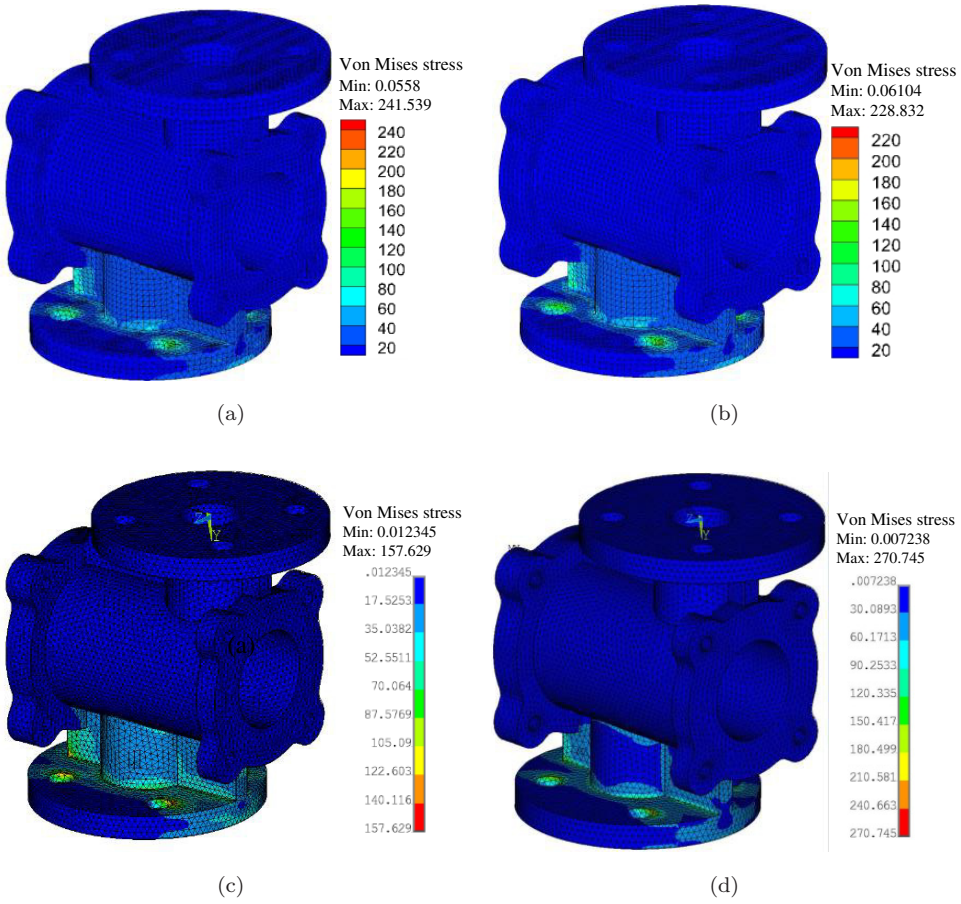

Fig. 20. (Color online) Von Mises stress. (a) MixFMBEM, (b) "unique traction" FMBEM, (c) ANSYS-1 and (d) ANSYS-2.

and the maximal stress errors of MixFMBEM/ANSYS-2, "unique traction" FMBEM/ANSYS-2 and ANSYS-1/ANSYS-2 are 10.8, 15.5 and 41.7%, respectively. The reason for this situation is that the accuracy of the stress is 1 order lower than the displacement accuracy. It means that the stress is only C^0 continuous for linear elements (used in FMBEMs and ANSYS-1), whose accuracy is much lower than the quadratic element (C^1 stress continuous) used in ANSYS-2. Comparing the max stress, it can be found that MixFMBEM is more accurate than the "unique traction" FMBEM, and the BEM is obviously more accurate than the FEM when the order of element is the same.

6. Conclusions

This paper presents a type of BC related mixed boundary elements and their efficient automatic generation technique to deal with corner/edge problems in 3D elastostatics. In this technique, NCEs are only generated at corners and edges where displacements are prescribed and CEs are distributed at all other places, which can decrease the increment of DOFs caused by NCEs as much as possible. The integrals in both CEs and NCEs are discussed in detail. With the help of ACIS, all CEs and NCEs can be automatically generated through some interacting operations (e.g., picking up surface and adding BCs) and the element generation algorithm presented in this paper. In order to solve large scale problems, the mixed boundary elements are introduced in the FMBEM and the algorithm is proposed. The BEA program is directly integrated into the 3D CAD software InterSolid whose geometric engine is ACIS, and therefore the CAD modeling and the BEA can run in the same environment that some time-consuming interacting operations between CAD models and boundary element models such as model simplification, model conversion and model modification can be easily realized. The examples show that the results are stable and reliable when the node shift in the range 1/20–3/5 and the accuracy of the BEA with BC related mixed boundary elements is higher than that only with CEs.

In the future, we will further work on the computational accuracy improvement and computational speed acceleration. In the accuracy improvement part, we want to use high accurate quadratic elements and evaluate the nearly singular integrals by some special techniques like subdivision of region of integration [Beer *et al.* (2008)] or semi-analytical algorithm [Niu *et al.* (2005)]; in the speed acceleration part, we will further optimize our program and expect to use the BC related mixed boundary element in the new FMBEM [Liu (2006)] to further improve the efficiency.

Acknowledgments

This research has been supported by two National Natural Science Foundation of China (51475180 and 51275182) and sponsored in part by the National Key Technologies R&D Program of China (Grant No. 2013ZX04005-011).

Appendix A

$$M^1_{n,m}(Q_c) = \int_{S_e} (\overrightarrow{Q_cQ})_j R_{n,m}(\overrightarrow{Q_cQ}) t_j(Q)\, dS(Q), \tag{A.1}$$

$$M^1_{j,n,m}(Q_c) = \int_{S_e} R_{n,m}(\overrightarrow{Q_cQ}) t_j(Q)\, dS(Q), \tag{A.2}$$

$$M^2_{n,m}(Q_c) = E_{jpkl} \int_{S_e} \frac{\partial}{\partial x_p}[(\overrightarrow{Q_cQ})_j R_{n,m}(\overrightarrow{Q_cQ})] n_k(Q) u_l(Q)\, dS(Q), \tag{A.3}$$

$$M^2_{j,n,m}(Q_c) = E_{jpkl} \int_{S_e} \frac{\partial}{\partial x_p}[R_{n,m}(\overrightarrow{Q_cQ})] n_k(Q) u_l(Q)\, dS(Q), \tag{A.4}$$

$$F^S_{ij,n,m}(\overrightarrow{Q_cP}) = \frac{\lambda+3\mu}{\lambda+2\mu}\delta_{ij} S_{n,m}(\overrightarrow{Q_cP}) - \frac{\lambda+\mu}{\lambda+2\mu}(\overrightarrow{Q_cP})_j \frac{\partial}{\partial x_i} S_{n,m}(\overrightarrow{Q_cP}), \tag{A.5}$$

$$G^S_{i,n,m}(\overrightarrow{Q_cP}) = \frac{\lambda+\mu}{\lambda+2\mu}\frac{\partial}{\partial x_i} S_{n,m}(\overrightarrow{Q_cP}), \tag{A.6}$$

where λ and μ are Lame coefficients, E_{jpkl} is elastic modulus tensor and equals to $\lambda\delta_{ik}\delta_{lp} + \mu(\delta_{il}\delta_{jp} + \delta_{ip}\delta_{kl})$, $S_{n,m}$ and $R_{n,m}$ are solid harmonic functions [Yoshida (2001)].

References

Beer, G., Smith, I. and Duenser, C. [2008] *The Boundary Element Method with Programming: For Engineers and Scientists* (Springer Verlag, Berlin-Heidelberg-New York).

Brebbia, C. A., Telles, J. C. F. and Wrobel, L. C. [1984] *Boundary Element Techniques: Theory and Applications in Engineering* (Springer-Verlag, Berlin).

Cayol, V. and Cornet, F. H. [1997] "3D mixed boundary elements for elastostatic deformation field analysis," *International Journal of Rock Mechanics and Mining Sciences* **34**(2), 275–287.

Chahine, G. L. and Kalumuck, K. M. [1998] "BEM software for free surface flow simulation including fluid-structure interaction effects," *International Journal of Computer Applications in Technology* **11**(3), 177–198.

Chaudonneret, M. [1978] "On the discontinuity of the stress vector in the boundary integral equation method for elastic analysis," *Recent Advances in Boundary Element Methods* (Pentech Press, Ltd, London), pp. 185–194.

Chen, Z. J. and Xiao, H. [2009] "The vectorization expressions of Taylor series multipole-BEM for 3D elasticity problems," *Computational Mechanics* **43**(2), 297–306.

Dong, Y., Zhang, J., Xie, G., Lu, C., Li, Y., Han, X. *et al.* [2014] "A general algorithm for evaluating domain integrals in 2D boundary element method for transient heat conduction," *International Journal of Computational Methods* **12**(2), 1550010-1–13.

Duenser, C. [2001] *Simulation of Sequential Tunnel Excavation with the Boundary Element Method*, Doctoral Thesis, Universitat Graz.

Dunavant, D. A. [1985] "High degree efficient symmetrical Gaussian quadrature rules for the triangle," *International Journal for Numerical Methods in Engineering* **21**(6), 1129–1148.

Gao, X. W. and Davies, T. G. [2000] "3D multi-region BEM with corners and edges," *International Journal of Solids and Structures* **37**(11), 1549–1560.

Gao, X. W. and Davies, T. G. [2002] *Boundary Element Programming in Mechanics* (Cambridge University Press, Cambridge).

Guzina, B. B., Pak, R. Y. S. and Martínez-Castro, A. E. [2006] "Singular boundary elements for three-dimensional elasticity problems," *Engineering Analysis with Boundary Elements* **30**(8), 623–639.

Hah, Z.-H., Kim, H.-J. and Youn, S.-K. [2013] "A hierarchically superimposing local refinement method for isogeometric analysis," *International Journal of Computational Methods* **11**(6), 1350074-1–28.

Ho-Le, K. [1988] "Finite element mesh generation methods: A review and classification," *Computer-Aided Design* **20**(1), 27–38.

Hu, S. R. and Chen, G. H. [1997] "A new triangle polar coordinate transformation," *Chinese Journal of Computational Mechanics* **14**(3), 372–376.

Hughes, T. J. R., Cottrell, J. A. and Bazilevs, Y. [2005] "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Computer Methods in Applied Mechanics and Engineering* **194**(39), 4135–4195.

Jaswon, M. A. and Symm, G. T. [1977] *Integral Equation Methods in Potential Theory and Elastostatics* (Oxford University Press, Oxford).

Kaufmann, H. [2009] "Dynamic differential geometry in education," *Journal for Geometry and Graphics* **13**(2), 131–144.

Lachat, J. C. and Watson, J. O. [1976] "Effective numerical treatment of boundary integral equations: A formulation for three-dimensional elastostatics," *International Journal for Numerical Methods in Engineering* **10**(5), 991–1005.

Lee, M. C. and Joun, M. S. [2008] "Adaptive triangular element generation and optimization-based smoothing, Part 1: On the plane," *Advances in Engineering Software* **39**(1), 25–34.

Liu, Y. J. [2006] "A new fast multipole boundary element method for solving large-scale two-dimensional elastostatic problems," *International Journal for Numerical Methods in Engineering* **65**(6), 863–881.

Liu, Y. J. [2009] *Fast Multipole Boundary Element Method*: *Theory and Applications in Engineering* (Cambridge University Press, Cambridge).

Liu, Y. J. and Nishimura, N. [2006] "The fast multipole boundary element method for potential problems: A tutorial," *Engineering Analysis with Boundary Elements* **30**(5), 371–381.

Marchetta, M. G. and Forradellas, R. Q. [2010] "An artificial intelligence planning approach to manufacturing feature recognition," *Computer-Aided Design* **42**(3), 248–256.

Mi, Y. and Aliabadi, M. H. [1992] "Dual boundary element method for three-dimensional fracture mechanics analysis," *Engineering Analysis with Boundary Elements* **10**(2), 161–171.

Mi, Y. and Aliabadi, M. H. [1994] "Discontinuous crack-tip elements: Application to 3D boundary element method," *International Journal of Fracture* **67**(3), 67–71.

Mukherjee, S. and Liu, Y. [2013] "The boundary element method," *International Journal of Computational Methods* **11**(6), 1350037-1–91.

Niu, Z., Wendland, W. L., Wang, X. and Zhou, H. [2005] "A semi-analytical algorithm for the evaluation of the nearly singular integrals in three-dimensional boundary element methods," *Computer Methods in Applied Mechanics and Engineering* **194**(9), 1057–1074.

Owen, S. J. [1998] "A survey of unstructured mesh generation technology" 7th International Meshing Roundtable, 239–267, 1998 (Citeseer).

Prochazka, P. P. and Lok, T. S. [2012] "Effect of elevated temperature on concrete structures by boundary elements," *International Journal of Computational Methods* **9**(1).

Saad, Y. and Schultz, M. H. [1986] "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing* **7**(3), 856–869.

Simpson, R. N., Bordas, S. P. A., Trevelyan, J. and Rabczuk, T. [2012] "A two-dimensional isogeometric boundary element method for elastostatic analysis," *Computer Methods in Applied Mechanics and Engineering* **209–212**(1), 87–100.

Spatial [2004] ACIS R15 Online Help.

Subia, S. R., Ingber, M. S. and Mitra, A. K. [1995] "A comparison of the semidiscontinuous element and multiple node with auxiliary boundary collocation approaches for the boundary element method," *Engineering Analysis with Boundary Elements* **15**(1), 19–27.

Tadeu, A. J. B., Godinho, L. and Santos, P. [2001] "Performance of the BEM solution in 3D acoustic wave scattering," *Advances in Engineering Software* **32**(8), 629–639.

Wang, Y. J., Wang, Q. F., Deng, X. W., Xia, Z. H., Yan, J. H. and Xu, H. [2015] "Graphics processing unit (GPU) accelerated fast multipole BEM with level-skip M2L for 3D elasticity problems," *Advances in Engineering Software* **82**, 105–118.

Wang, Y. J., Wang, Q. F., Wang, G., Huang, Y. B. and Wang, S. T. [2013] "An adaptive dual-information FMBEM for 3D elasticity and its GPU implementation," *Engineering Analysis with Boundary Elements* **37**(2), 236–249.

Wang, H. T. and Yao, Z. H. [2005] "A new fast multipole boundary element method for large scale analysis of mechanical properties in 3D particle-reinforced composites," *CMES*: *Computer Modeling in Engineering & Sciences* **7**(1), 85–96.

Wang, H. T., Yao, Z. H. and Wang, P. B. [2005] "On the preconditioners for fast multipole boundary element methods for 2D multi-domain elastostatics," *Engineering Analysis with Boundary Elements* **29**(7), 673–688.

Wilde, A. J. and Aliabadi, M. H. [1998] "Direct evaluation of boundary stresses in the 3D BEM of elastostatics," *Communications in Numerical Methods in Engineering* **14**(6), 505–517.

Xiao, H. and Chen, Z. J. [2007] "Numerical experiments of preconditioned Krylov subspace methods solving the dense non-symmetric systems arising from BEM," *Engineering Analysis with Boundary Elements* **31**(12), 1013–1023.

Yoshida, K. [2001] *Applications of Fast Multipole Method to Boundary Integral Equation Method*, Ph.D. Dissertation, Department of Global Environment Eng, Kyoto Univ, Japan.

Zhang, Q. and Mukherjee, S. [1991] "Design sensitivity coefficients for linear elastic bodies with zones and corners by the derivative boundary element method," *International Journal of Solids and Structures* **27**(8), 983–998.

Zheng, C., Chen, H., Matsumoto, T. and Takahashi, T. [2012] "3D acoustic shape sensitivity analysis using fast multipole boundary element method," *International Journal of Computational Methods* **9**(1), 1240004-1–11.