 CMES, vol.70, no.2, pp.123-151, 2010

The Hybrid Boundary Node Method Accelerated by Fast Multipole Expansion Technique for 3D Elasticity

Qiao Wang[1], Yu Miao[1,2] and Junjie Zheng[1]

Abstract: In this paper, a fast formulation of the hybrid boundary node method (Hybrid BNM) for solving 3D elasticity is presented. Coupling modified variational principle with the Moving Least Squares (MLS) approximation, the Hybrid BNM only requires discrete nodes constructed on the surface of a domain. The preconditioned GMERS is employed to solve the resulting system of equations. At each iteration step of the GMERS, the matrix-vector multiplication is accelerated by the fast multipole method (FMM). The fundamental solution of three-dimensional elasticity problem is expanded in terms of series. An oct-tree data structure is adopted to subdivide the computational domain into well-separated cells hierarchically and to invoke the multipole expansion approximation. Formulations for the local and multipole expansions and conversion of multipole to local expansion are given. Nearly one million of total unknowns can be computed on a PC with 2.67GHz CPU and 2.0GB RAM. All the formulations are implemented in a computer code written in C++. Numerical examples demonstrate the accuracy and efficiency of the proposed approach.

Keywords: meshless method; hybrid boundary node method; modified variational principle, moving least squares approximation; fast multipole method; 3D elasticity

1 Introduction

There are two representative methods for numerical analysis. The first one is the finite element method (FEM) and the second is the boundary element method (BEM). The FEM is a very powerful method and has been well-developed, while a discretization of the domain is needed, which will result in difficulties with remesh-

[1] School of Civil Engineering and Mechanics, Huazhong University of Science and Technology, Wuhan 430074, China

[2] Corresponding Author. Tel: 86 27 87540172; Fax: 86 27 87542231; Yu Miao Email: my_miaoyu@163.com

ing in problems involving moving boundaries, large deformations or crack propagation. When dealing with complex geometries, it is often time-consuming and prone to errors while meshing. The BEM can reduce the dimensionality of a problem by one and requires a discretization of the boundary only. However, the BEM will lead to a dense and unsymmetrical $N \times N$ system of linear equation, where N is the total number of degrees of freedom (DOF). The computational cost and memory requirement for directly factoring

such system increase with $O(N^3)$ and $O(N^2)$, respectively, which limits the development of the BEM.

In the past decades, a new class of numerical methods, namely, the meshfree or meshless methods were developed. This kind of methods can reduce the human-labor costs required for meshing the domains of complex-shape. There are a number of meshless methods been investigated by several researchers and they can be classified into two categories: the domain type and the boundary type. The element free Galerkin method (EFG) (Belytschko et al., 1994) is a representative method of the domain type, which uses a global symmetric weak form and the shape function comes from the moving least-squares (MLS) approximation (Belytschko et al., 1996). However, the EFG method needs background cells for the integration. Another domain type method is the meshless local Petrov-Galerkin (MLPG) approach (Atluri and Zhu, 1998; Atluri and Shen, 2002a,b; Atluri, 2004). The method uses local weak form over local sub-domains to avoid the generation of the background cells. The MLPG method has also been applied to boundary integral equations, as MLPG/BIE (Atluri, Han and Shen, 2003; Han and Atluri, 2003). In a series of efforts to simplify and speed up the meshless implementation, the so-called MLPG "mixed" finite volume method was proposed (Atluri, Han and Rajendran, 2004). Recently, Atluri et al. proposed a MLPG "mixed" collocation method (Atluri, Liu and Han, 2006a) and a MLPG "mixed" finite difference method (Atluri, Liu and Han, 2006b).

One of the boundary type meshless methods is the boundary node method (BNM) (Mukherjee and Mukherjee, 1997; Chati, Mukherjee and Mukherjee, 1999), which inserts the MLS approximation into the boundary integral equations (BIE). However, the method needs cell structure for numerical integration like the EFG. Another boundary type meshless method is the hybrid boundary node method (Hybrid BNM). The Hybrid BNM (Zhang and Yao, 2001) is proposed by Zhang et al. for potential problems (Zhang, Tanaka, M and Matsumoto, 2004) and elasticity problems (Zhang and Yao, 2004) and has been developed by Miao et al. (2005, 2006, 2009), which combines the MLS approximation scheme with the hybrid displacement variational formula. The Hybrid BNM not only has the advantage of reducing the spatial dimensions like BEM or BNM, but also does not require any

cells for interpolation nor for integration. However, like the traditional BEM, the Hybrid BNM has a dense and unsymmetrical system matrix, which requires $O(N^2)$ memory and $O(N^3)$ operations. Therefore, the Hybrid BNM must be speeded up while dealing with large scale problems. The fast multipole method (FMM) was introduced by Rokhlin (1985) as a fast solution method for integral equations for two dimensional Laplace's equation and then developed by Greengard and Rokhlin (1987) as an algorithm for the rapid evaluation of potential and force fields in a large scale ensemble of charged particles. It can reduce the computational cost for the pair-wise force calculation from $O(N^2)$ to $O(N)$, the computation for large scale problems becomes possible.

Applying the FMM to accelerate the BEM computation has been investigated by many researchers in potential problems (Nabors et al., 1994; Nishida and Hayami, 1997; Liu and Nishimura, 2006), and the computational cost of the FM-BEM, including the memory and CPU time, have been successfully reduced to $O(N)$. However, the implementation of FMM for elasticity problems is more difficult than that for potential problems as the fundamental solution is much more complex to expand in terms of series. There are mainly two types of expansions: Taylor series and spherical harmonic series. Peiece and Napier (1995) developed a Taylor series multipole expansion algorithm to solve the two-dimensional problem of multi-cracks in elastic media. Hayami and Sauter (1998a) presented an efficient algorithm for 3D elastostatics use Taylor series. Popov et al. (2001) developed an $O(N)$ Taylor series multipole boundary element methods for three-dimensional elasticity problems. The algorithms for 3D elasticity problems, which based on spherical harmonic expansions, were also investigated. Fu et al. (1998) decomposed the original 3D elasticity fundamental solution into five terms and each of them can be expanded in terms of spherical harmonic series with their corresponding duality principle. Hayami and Sauter (1998b) proposed an expansion in terms of the spherical harmonic expansion of $1/R$ and the derivatives of R. Starting from the kernel expansion of the fundamental solution of the Laplacian, Yoshida et al. (2001) obtained a new multipole expansion together with the corresponding translations for the fundamental solution of linear elastostatics. The FMM also has some implementations in boundary-based meshfree methods. Kulkarni et al. (2003) combined the BNM with FMM for 2D potential problems, in which high accuracy and efficiency are observed.

The Hybrid BNM accelerated by FMM for 3D potential problems has been proposed by Zhang et al. (2005), called FM-HBNM. Zhang also has used the FM-HBNM to study the thermal behavior of carbon nanotubes (CNT) composites (Zhang and Tanaka, 2007; Zhang and Tanaka, 2008), which is a very complex problem and almost impossible to obtain a reasonable discretization for the geometry with FEM.

In this paper, the Hybrid BNM for 3D elasticity problems accelerated by the FMM using the spherical harmonic series is proposed, and an efficient algorithm that can reduce both the computer costs and the human-labor costs is derived.

The paper is organized as follows. In the second part, the Hybrid BNM for three-dimensional elasticity problems is reviewed. The detail of the FM-HBNM is presented next. This is followed by the procedure of the FM-HBNM in practical implementation. Finally, in the fifth part, numerical results are given. Results from the Hybrid BNM and the FM-HBNM are compared, with respect to accuracy and computational efficiency.

2 The Hybrid Boundary Node Method for 3D Elasticity

In this section, the Hybrid BNM for 3D elasticity is reviewed. The Hybrid BNM is based on a modified variational principle (DeFigueredo and Brebbia, 1989). In 3D elasticity, the functions in the modified variational principle that assumed to be independent are: displacements $\tilde{u}_i$ and tractions $\tilde{t}_i$ on the boundary and displacements u_i inside the domain. Consider a domain Ω enclosed by $\Gamma = \Gamma_u + \Gamma_t$ with $\bar{u}_i$ and $\bar{t}_i$ are the prescribed displacements and tractions, respectively. The corresponding variational functional Π_{HB} is defined as follows:

$$\Pi_{HB} = \int_\Omega \frac{1}{2} u_{i,j} C_{ijkl} u_{k,l} \mathrm{d}\Omega - \int_\Gamma \tilde{t}_i (u_i - \tilde{u}_i) \mathrm{d}\Gamma - \int_{\Gamma_t} \bar{t}_i \tilde{u}_i \mathrm{d}\Gamma \tag{1}$$

where, the boundary displacements $\tilde{u}_i$ satisfy the essential boundary conditions, i.e. $\tilde{u}_i = \bar{u}_i$ on Γ_u.

The integral equations can be obtained by $\delta \Pi_{HB} = 0$ over the domain and its boundary as follows:

$$\int_\Gamma (t_i - \tilde{t}_i) \delta u_i \mathrm{d}\Gamma - \int_\Omega \sigma_{ij,j} \delta u_i \mathrm{d}\Omega = 0 \tag{2}$$

$$\int_\Gamma (u_i - \tilde{u}_i) \delta \tilde{t}_i \mathrm{d}\Gamma = 0 \tag{3}$$

$$\int_\Gamma (\tilde{t}_i - \bar{t}_i) \delta \tilde{u}_i \mathrm{d}\Gamma = 0 \tag{4}$$

Equation (4) will be satisfied if the traction boundary conditions $\tilde{t}_i = \bar{t}_i$ are imposed. So it can be ignored in the following discussion.

The modified variational principle holds both in the whole domain Ω and any sub-domain Ω_I with its boundary Γ_I and L_I. Define the sub-domain Ω_I as an intersection of the domain and a small sphere centered at node $\mathbf{s}_I$, with $\Gamma_I = \partial\Omega_I \cap \Gamma$

and $L_I = \partial\Omega_I - \Gamma_I$, respectively. We can obtain the following weak forms for the sub-domains and its boundaries to replace Equations (2) and (3):

$$\int_{\Gamma_I+L_I} (t_i - \tilde{t}_i) h_I \mathrm{d}\Gamma - \int_{\Omega_I} \sigma_{ij,j} h_I \mathrm{d}\Omega = 0 \tag{5}$$

$$\int_{\Gamma_I+L_I} (u_i - \tilde{u}_i) h_I \mathrm{d}\Gamma = 0 \tag{6}$$

where h_I is a weight function.

The displacements $\tilde{u}$ and tractions $\tilde{t}$ at the boundary Γ are approximated by the MLS approximation as follows:

$$\tilde{u}(\mathbf{s}) = \sum_{J=1}^{n} \Phi_J(\mathbf{s}) \hat{u}_J \tag{7}$$

$$\tilde{t}(\mathbf{s}) = \sum_{J=1}^{n} \Phi_J(\mathbf{s}) \hat{t}_J \tag{8}$$

where n is the number of nodes for MLS approximation which located on the surface; $\hat{u}_J$ and $\hat{t}_J$ are nodal values, and $\Phi_J(\mathbf{s})$ is the shape function of the MLS approximation, corresponding to node $\mathbf{s}_J$, which is given by

$$\Phi_J(\mathbf{s}) = \sum_{j=1}^{m} p_j(\mathbf{s}) [A^{-1}(\mathbf{s}) B(\mathbf{s})]_{jJ} \tag{9}$$

and the matrices $A(\mathbf{s})$ and $B(\mathbf{s})$ are defined by

$$A(\mathbf{s}) = \sum_{J=1}^{n} w_J(\mathbf{s}) \mathbf{p}(\mathbf{s}_J) \mathbf{p}^{\mathbf{T}}(\mathbf{s}_J) \tag{10}$$

$$B(\mathbf{s}) = [w_1(\mathbf{s})\mathbf{p}(\mathbf{s}_1), w_2(\mathbf{s})\mathbf{p}(\mathbf{s}_2), ..., w_n(\mathbf{s})\mathbf{p}(\mathbf{s}_n)] \tag{11}$$

In the above three equations, $p_j(\mathbf{s})$ provide a basis of order m consisting of monomials in(s_1, s_2), which is the represent parametric coordinates on a surface. In this paper, we take m as 6, namely, $\mathbf{p}^{\mathbf{T}}(\mathbf{s}) = [1, s_1, s_2, s_1^2, s_1 s_2, s_2^2]$. In Equations (10) and (11), $w_J(\mathbf{s})$ are the weight functions and in this paper we choose Gaussian weight function (Zhang andYao, 2004).

In Equations (5) and (6), $\tilde{u}_i$ and $\tilde{t}_i$ on Γ_I can be expressed by Equations (7) and (8) since Γ_I is a portion of Γ, but $\tilde{u}_i$ and $\tilde{t}_i$ on L_I has not been defined yet. In order to solve this problem, we deliberately select h_I such that all integrals over L_I vanished. This can be easily accomplished by using the weight function in the

MLS approximation for h_I, with replacing the radius of the support of the weight function by the radius r_I of the sub-domain Ω_I, i.e.

$$h_I(Q) = \begin{cases} \frac{\exp[-(d_I/c_I)^2]-\exp[-(r_I/c_I)^2]}{1-\exp[-(r_I/c_I)^2]}, & 0 \le d_I \le r_I \\ 0, & d_I \ge r_I \end{cases} \tag{12}$$

where d_I is the distance between a field point Q and the nodal point $\mathbf{s}_I$; c_I is a constant controlling the shape of the function. With h_I vanishing at L_I, the integrals over L_I in Equations (5) and (6) are zero and the two equations can be rewritten as

$$\int_{\Gamma_I} (t_i - \tilde{t}_i) h_I \mathrm{d}\Gamma - \int_{\Omega_I} \sigma_{ij,j} h_I \mathrm{d}\Omega = 0 \tag{13}$$

$$\int_{\Gamma_I} (u_i - \tilde{u}_i) h_I \mathrm{d}\Gamma = 0 \tag{14}$$

The $\mathbf{u}$ and $\mathbf{t}$ inside the domain can be approximated by fundamental solutions as

$$\mathbf{u} = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \sum_{J=1}^{N} \begin{bmatrix} u_{11}^J & u_{12}^J & u_{13}^J \\ u_{21}^J & u_{22}^J & u_{23}^J \\ u_{31}^J & u_{32}^J & u_{33}^J \end{bmatrix} \begin{Bmatrix} x_1^J \\ x_2^J \\ x_3^J \end{Bmatrix} \tag{15}$$

$$\mathbf{t} = \begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \sum_{J=1}^{N} \begin{bmatrix} t_{11}^J & t_{12}^J & t_{13}^J \\ t_{21}^J & t_{22}^J & t_{23}^J \\ t_{31}^J & t_{32}^J & t_{33}^J \end{bmatrix} \begin{Bmatrix} x_1^J \\ x_2^J \\ x_3^J \end{Bmatrix} \tag{16}$$

where $u_{ij}^J = u_{ij}(\mathbf{s}_J, Q)$ and $t_{ij}^J = t_{ij}(\mathbf{s}_J, Q)$ are the fundamental solutions; x_i^J are unknown parameters; N is the total number of boundary nodes for approximation of $\mathbf{u}$ and $\mathbf{t}$ and this set of nodes is contained by the set of nodes for MLS approximation, so we have $N \le n$. For 3D elasticity problems, the fundamental solutions are

$$u_{ij}^J = \frac{-1}{16\pi r(1-\nu)\mu} \{(3-4\nu)\delta_{ij} - r_{,i} r_{,j}\} \tag{17}$$

$$t_{ij}^J = \frac{-1}{8\pi(1-\nu)r^2} \{[(1-2\nu)\delta_{ij} + 3r_{,i} r_{,j}] \frac{\partial r}{\partial n} + (1-2\nu)(r_{,i} n_j - r_{,j} n_i)\} \tag{18}$$

where $r = r(\mathbf{s}_J, Q)$ and Q is the field point while $\mathbf{s}_J$ is the source point.

From the fundamental solution, one can see that it leads to singularities in the integrals of Equations (13) and (14) if the field point Q and source point $\mathbf{s}_J$ are coincide. The singularities in Equation (14) are weak and can be evaluated directly. However,

the integrals in Equation (13) are strong singularities, in order to avoid direct numerical integration of these terms, the rigid body movement is utilized (Miao and Wang, 2006).

By substituting Equations (7), (8), (12), (15) and (16) into Equations (13) and (14), we can obtain

$$\begin{aligned}&\sum_{J=1}^{N}\int_{\Gamma_I}\begin{bmatrix}u_{11}^J & u_{12}^J & u_{13}^J\\ u_{21}^J & u_{22}^J & u_{23}^J\\ u_{31}^J & u_{32}^J & u_{33}^J\end{bmatrix}\begin{Bmatrix}x_1^J\\ x_2^J\\ x_3^J\end{Bmatrix}h_I(Q)\mathrm{d}\Gamma\\ &=\sum_{J=1}^{N}\int_{\Gamma_I}\begin{bmatrix}\Phi_J(\mathbf{s}) & 0 & 0\\ 0 & \Phi_J(\mathbf{s}) & 0\\ 0 & 0 & \Phi_J(\mathbf{s})\end{bmatrix}\begin{Bmatrix}\hat{u}_1^J\\ \hat{u}_2^J\\ \hat{u}_3^J\end{Bmatrix}h_I(Q)\mathrm{d}\Gamma\end{aligned} \tag{19}$$

$$\begin{aligned}&\sum_{J=1}^{N}\int_{\Gamma_I}\begin{bmatrix}t_{11}^J & t_{12}^J & t_{13}^J\\ t_{21}^J & t_{22}^J & t_{23}^J\\ t_{31}^J & t_{32}^J & t_{33}^J\end{bmatrix}\begin{Bmatrix}x_1^J\\ x_2^J\\ x_3^J\end{Bmatrix}h_I(Q)\mathrm{d}\Gamma\\ &=\sum_{J=1}^{N}\int_{\Gamma_I}\begin{bmatrix}\Phi_J(\mathbf{s}) & 0 & 0\\ 0 & \Phi_J(\mathbf{s}) & 0\\ 0 & 0 & \Phi_J(\mathbf{s})\end{bmatrix}\begin{Bmatrix}\hat{t}_1^J\\ \hat{t}_2^J\\ \hat{t}_3^J\end{Bmatrix}h_I(Q)\mathrm{d}\Gamma\end{aligned} \tag{20}$$

Using Equations (19) and (20) for all nodes, the final system equations in matrix form can be written as

$$\mathbf{U}\mathbf{x}=\mathbf{H}\hat{\mathbf{u}} \tag{21}$$

$$\mathbf{T}\mathbf{x}=\mathbf{H}\hat{\mathbf{t}} \tag{22}$$

where

$$U_{IJ}=\int_{\Gamma_I}\begin{bmatrix}u_{11}^J & u_{12}^J & u_{13}^J\\ u_{21}^J & u_{22}^J & u_{23}^J\\ u_{31}^J & u_{32}^J & u_{33}^J\end{bmatrix}h_I(Q)\mathrm{d}\Gamma \tag{23}$$

$$T_{IJ}=\int_{\Gamma_I}\begin{bmatrix}t_{11}^J & t_{12}^J & t_{13}^J\\ t_{21}^J & t_{22}^J & t_{23}^J\\ t_{31}^J & t_{32}^J & t_{33}^J\end{bmatrix}h_I(Q)\mathrm{d}\Gamma \tag{24}$$

$$H_{IJ}=\int_{\Gamma_I}\begin{bmatrix}\Phi_J(\mathbf{s}) & 0 & 0\\ 0 & \Phi_J(\mathbf{s}) & 0\\ 0 & 0 & \Phi_J(\mathbf{s})\end{bmatrix}h_I(Q)\mathrm{d}\Gamma \tag{25}$$

$$\mathbf{x}^{\mathbf{T}}=[x_1^1,x_2^1,x_3^1,\cdots,x_1^N,x_2^N,x_3^N] \tag{26}$$

$$\hat{\mathbf{u}}^{\mathbf{T}}=[\hat{u}_1^1,\hat{u}_2^1,\hat{u}_3^1,\cdots,\hat{u}_1^N,\hat{u}_2^N,\hat{u}_3^N] \tag{27}$$

$$\hat{\mathbf{t}}^{\mathbf{T}} = [\hat{t}_1^1, \hat{t}_2^1, \hat{t}_3^1, \cdots, \hat{t}_1^N, \hat{t}_2^N, \hat{t}_3^N] \tag{28}$$

For a well-posed problem, either $\tilde{u}_i$ or $\tilde{t}_i$ are known at each nodes on the boundary. For the faces where $\tilde{u}_i$ are prescribed, $\hat{u}_i^I$ can be computed by

$$\hat{u}_i^I = \sum_{J=1}^{N} R_{IJ} \tilde{u}_i^J = \sum_{J=1}^{N} R_{IJ} \bar{u}_i^J \tag{29}$$

and for the faces where $\tilde{t}_i$ are prescribed, $\hat{t}_i^I$ can be computed by

$$\hat{t}_i^I = \sum_{J=1}^{N} R_{IJ} \tilde{t}_i^J = \sum_{J=1}^{N} R_{IJ} \bar{t}_i^J \tag{30}$$

where $R_{IJ} = [\Phi_J(\mathbf{s}^I)]^{-1}$ (Atluri, Kim and Cho, 1999).

For a general problem, either $\tilde{u}_i$ or $\tilde{t}_i$ are known at each node on the boundary and by rearranging Equations (21) and (22), a final algebraic equation in terms of $\mathbf{x}$ only can be obtained as below:

$$\mathbf{A}\mathbf{x} = \mathbf{d} \tag{31}$$

For the node $\mathbf{s}_I$, if $\tilde{u}_i$ is known, select the correspond row in $\mathbf{U}$ to $\mathbf{A}$, otherwise, select the correspond row in $\mathbf{T}$ to $\mathbf{A}$, and the corresponding term of $\mathbf{d}$ comes from the matrix-vector product of $\mathbf{H}\hat{\mathbf{u}}$ or $\mathbf{H}\hat{\mathbf{t}}$. Then the unknown vector $\mathbf{x}$ is obtained by solving the final algebraic equation. The nodal values $\hat{\mathbf{u}}$ and $\hat{\mathbf{t}}$ on the boundary can be computed by the back-substitution of $\mathbf{x}$ into Equations (21) and (22), then use Equations (7) and (8) the displacements and tractions on the boundary can be obtained. The displacements and tractions at interior points can be evaluated by the traditional boundary integral equations.

3 The Formulations of Hybrid BNM With FMM

The Hybrid BNM has a dense and unsymmetrical system matrix, which will restrict the method in small scale problems if we use the conventional direct solvers, such as, Gaussian elimination to solve the system matrix. For 3D elasticity, every node has three degrees of freedom which makes it take more memory and time than potential problems. It is quite necessary to combine the Hybrid BNM for 3D elasticity with the FMM, which can dramatically reduce the computational cost. In this section, the detail of the implementation of FMM techniques in Hybrid BNM for 3D elasticity is presented.

The FMM is an algorithm which can achieve fast products for particular dense matrices with vectors, and also reduce the memory complexity. The FMM uses

multipole expansions to approximate the effects of a distant group of particles (in Hybrid BNM they are nodes) on a local group, and thus achieve faster summation. However, it bounds the error analytically and can achieve a certain guaranteed level of accuracy by determining how many terms in a multipole expansion. The FMM uses a hierarchical decomposition of space to define ever-larger groups as distances increase. In 3D problems, an oct-tree is usually employed to decompose the domain.

In this paper, in order to utilize the FMM, an iterative equation solver is necessary, and the restarted preconditioned GMRES (1986) is employed. An adaptive version of the FMM (Greegard, 1988) is adopted; a hierarchy of boxes which refine the computational domain is constructed first. The refinement level 0 contains the entire computational domain. Refinement level $l+1$ is obtained from level l by subdividing of a box into eight equal boxes. The eight boxes at level $l+1$ obtained by subdivision of the box at level l are considered as its children. Stop the subdivision of a box while the number of nodes included in the box is smaller than a prescribed number. Delete the child box if it contains no node. A childless box we call it leaf.

Here we have some definitions first.

Definition 1

Two boxes are said to be neighbors if they are at the same level and share at least a vertex (and we also said a box is a neighbor of itself).

Definition 2

Two boxes are said to be well separated if they are at the same level but not neighbors.

Definition 3

We said each box b has an interaction list, whose members are the children of the neighbors of b's parent which are well separated from box b.

Now consider the formula below, it is the inner product between one row of the matrix $\mathbf{A}$ in Equation (31) and an iteration vector $\mathbf{x}'$ corresponding to the solution vector $\mathbf{x}$, which is given by either

$$\sum_{J=1}^{N} \int_{\Gamma_J} u_{ij}^J h_I(Q) x'_{Ji} \mathrm{d}\Gamma(Q) \tag{32}$$

or

$$\sum_{J=1}^{N} \int_{\Gamma_J} t_{ij}^J h_I(Q) x'_{Ji} \mathrm{d}\Gamma(Q) \tag{33}$$

The first formula will be computed for convenience. Suppose that the boundary node $\mathbf{s}_I$ belongs to a leaf of boxes in the oct-tree structure. The sum of expression (32) can be divided into two parts. The first part is the sum of the contributions of the nodes contained in the neighborhoods of the leaf (we call them *near nodes*); the other part is the sum of all the rest nodes (we call them *far nodes*). The sum of (31) can be expressed as

$$\sum_{J=1}^{N}\int_{\Gamma_I} u_{ij}^{J} h_I(Q) x'_{Ji} \mathrm{d}\Gamma(Q) = \sum_{J=1}^{N_{\text{near}}}\int_{\Gamma_I} u_{ij}^{J} h_I(Q) x'_{Ji} \mathrm{d}\Gamma(Q) + \sum_{J=1}^{N_{\text{far}}}\int_{\Gamma_I} u_{ij}^{J} h_I(Q) x'_{Ji} \mathrm{d}\Gamma(Q) \tag{34}$$

where N_{near} and N_{far} are the numbers of the near nodes and far nodes, respectively. The sum of the near nodes' contributions will be computed directly, while the far nodes' contributions will be computed by multipole expansion to speed up. Consider a leaf B_{local} contains node $\mathbf{s}_I$ and another leaf B_{far} which is on the interaction list of B_{local} contains N_b nodes. The node $\mathbf{s}_J$ is one of the N_b nodes.

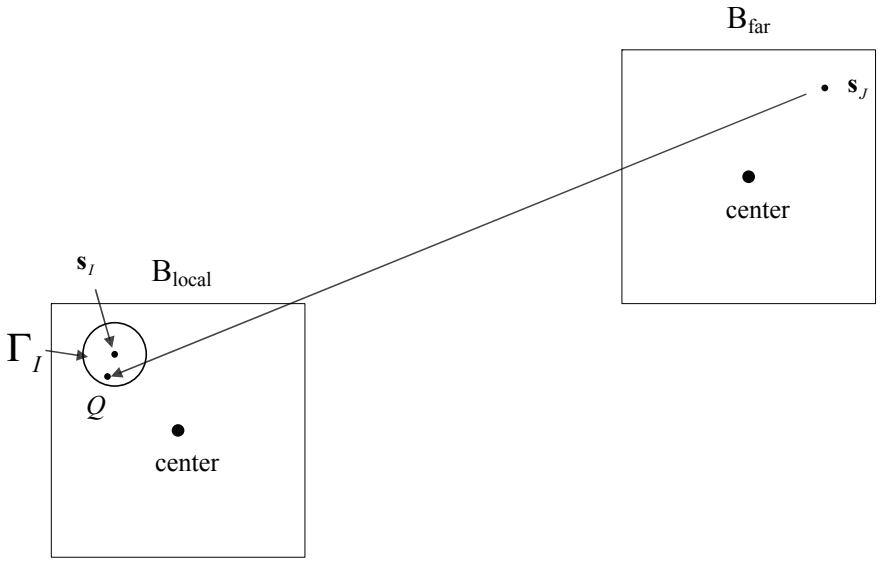

Figure 1: Conventional evaluation of contribution from node $\mathbf{s}_J$

Figure 1 is the conventional evaluation of the contribution from node $\mathbf{s}_J$. In FM-HBNM it will be computed indirectly with the tree structure.

In Hybrid BNM of 3D elasticity, the expansion of the fundamental solution used in this paper has some difference from that in BEM (Yoshida, Nishimura and Kobayashi,2001; Liu, Nishimura, Otani and Takahashi, 2005). We will derive the expansion and all the other formulations needed in the following text.

Start with the following form of the fundamental solution:

$$u_{ij}^J = u_{ij}(\mathbf{s}_J, Q) = \frac{1}{8\pi\mu}(\delta_{ij}\frac{2}{r(\mathbf{s}_J, Q)} - \frac{\lambda+\mu}{\lambda+2\mu}\frac{\partial}{\partial Q_i}\frac{\partial}{\partial Q_j}r(\mathbf{s}_J, Q)) \tag{35}$$

Rewrite the above equation as

$$u_{ij}(\mathbf{s}_J, Q) = \frac{1}{8\pi\mu}(\delta_{ij}\frac{2}{r(\mathbf{s}_J, Q)} - \frac{\lambda+\mu}{\lambda+2\mu}\frac{\partial}{\partial Q_j}\frac{Q_i - \mathbf{s}_{Ji}}{r(\mathbf{s}_J, Q)}) \tag{36}$$

Set the centre of B_{far} as the origin of a spherical co-ordinates system (see Figure 2). Since the condition $|O_1\mathbf{s}_J| < |O_1Q|$ holds, the following identity holds:

$$r(\mathbf{s}_J, Q) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n}\overline{S_{n,m}}(\overrightarrow{O_1Q})R_{n,m}(\overrightarrow{O_1\mathbf{s}_J}) \tag{37}$$

By substituting Equation (37) into Equation (36), one can obtain

$$u_{ij} = \frac{1}{8\pi\mu}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}(\overline{F^s_{ij,n,m}}(\overrightarrow{O_1Q})R_{n,m}(\overrightarrow{O_1\mathbf{s}_J}) + \overline{G^s_{j,n,m}}(\overrightarrow{O_1Q})(\overrightarrow{O_1\mathbf{s}_J})_i R_{n,m}(\overrightarrow{O_1\mathbf{s}_J})) \tag{38}$$

where

$$F^s_{ij,n,m}(\overrightarrow{Ox}) = \frac{\lambda+3\mu}{\lambda+2\mu}\delta_{ij}S_{n,m}(\overrightarrow{Ox}) - \frac{\lambda+\mu}{\lambda+2\mu}(\overrightarrow{Ox})_i\frac{\partial}{\partial x_j}S_{n,m}(\overrightarrow{Ox}) \tag{39}$$

$$G^s_{j,n,m}(\overrightarrow{Ox}) = \frac{\lambda+\mu}{\lambda+2\mu}\frac{\partial}{\partial x_j}S_{n,m}(\overrightarrow{Ox}) \tag{40}$$

$$R_{n,m}(\overrightarrow{Ox}) = \frac{1}{(n+m)!}P_n^m(\cos\theta)e^{im\varphi}r^n$$

$$S_{n,m}(\overrightarrow{Ox}) = (n-m)!P_n^m(\cos\theta)e^{im\varphi}\frac{1}{r^{n+1}}$$

Here (r,θ,φ) are the polar coordinates of the point x, P_n^m is the associated Legendre function and a superposed bar indicates the complex conjugate, respectively.

Similarly, we have

$$F^R_{ij,n,m}(\overrightarrow{Ox}) = \frac{\lambda+3\mu}{\lambda+2\mu}\delta_{ij}R_{n,m}(\overrightarrow{Ox}) - \frac{\lambda+\mu}{\lambda+2\mu}(\overrightarrow{Ox})_i\frac{\partial}{\partial x_j}R_{n,m}(\overrightarrow{Ox}) \tag{41}$$

$$G^R_{j,n,m}(\overrightarrow{Ox}) = \frac{\lambda+\mu}{\lambda+2\mu}\frac{\partial}{\partial x_j}R_{n,m}(\overrightarrow{Ox}) \tag{42}$$

The functions $R_{n,m}$ and $S_{n,m}$ satisfy the relations given by the next two equations. If $\overrightarrow{Ox}$ and $\overrightarrow{Oy}$ are two vectors such that $|\overrightarrow{Oy}| < |\overrightarrow{Ox}|$, then

$$S_{n,m}(\overrightarrow{yx}) = \sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}\overline{R_{n',m'}}(\overrightarrow{Oy})S_{n+n',m+m'}(\overrightarrow{Ox}) \tag{43}$$

If $\overrightarrow{Ox}$ and $\overrightarrow{Oy}$ are two arbitrary vectors, then

$$R_{n,m}(\overrightarrow{yx}) = \sum_{n'=0}^{n}\sum_{m'=-n'}^{n'}R_{n',m'}(\overrightarrow{yO})R_{n-n',m-m'}(\overrightarrow{Ox}) \tag{44}$$

Using Equation (38), the sum in Equation (34) for the nodes included in B_{far} can be expressed by

$$\begin{aligned}&\sum_{J=1}^{N_b}\int_{\Gamma_I}u^J_{ij}h_I(Q)x'_{Ji}\mathrm{d}\Gamma(Q)\\ &=\frac{1}{8\pi\mu}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}\Big(\int_{\Gamma_I}\overline{F^s_{ij,n,m}}(\overrightarrow{O_1Q})h_I(Q)\mathrm{d}\Gamma(Q)M^1_{i,n,m}(O_1)\\ &+\int_{\Gamma_I}\overline{G^s_{j,n,m}}(\overrightarrow{O_1Q})h_I(Q)\mathrm{d}\Gamma(Q)M^2_{i,n,m}(O_1)\Big)\end{aligned} \tag{45}$$

where $M^1_{i,n,m}(O_1)$ and $M^2_{i,n,m}(O_1)$ are multipole moments centered at O_1, expressed as

$$M^1_{i,n,m}(O_1) = \sum_{J=1}^{N_b}R_{n,m}(\overrightarrow{O_1\mathbf{s}_J})x'_{Ji} \tag{46}$$

$$M^2_{i,n,m}(O_1) = \sum_{J=1}^{N_b}(\overrightarrow{O_1\mathbf{s}_J})_iR_{n,m}(\overrightarrow{O_1\mathbf{s}_J})x'_{Ji} \tag{47}$$

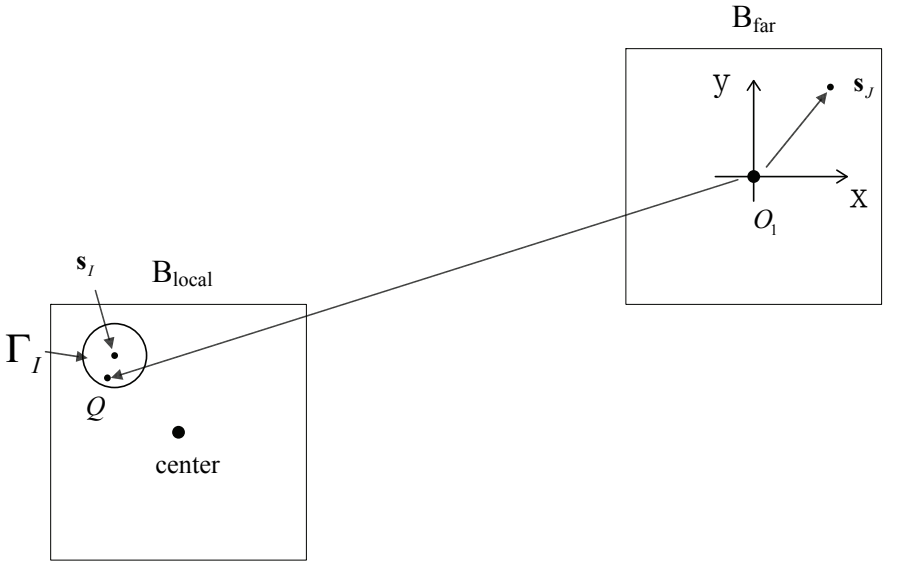

Figure 2: Co-ordinate system for multipole expansion

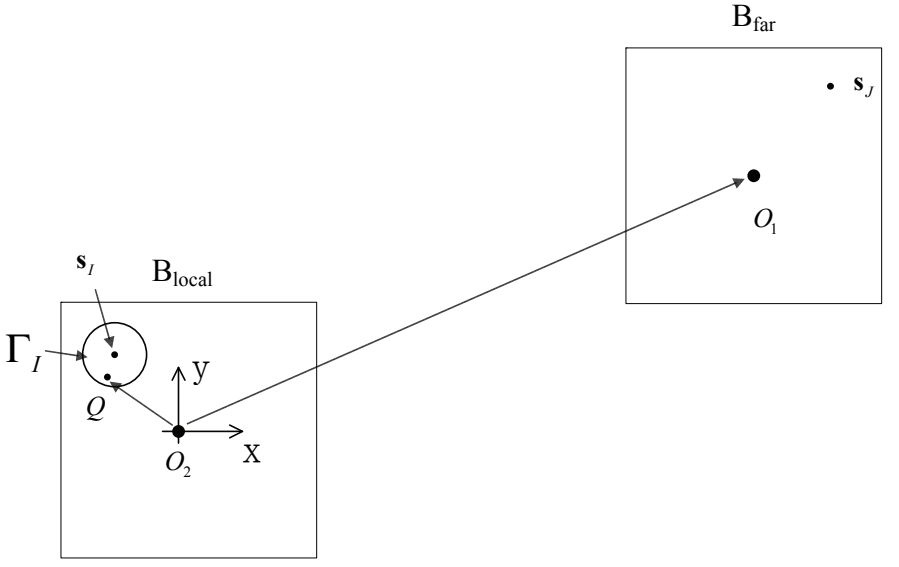

Figure 3: Co-ordinate system for local expansion

Move the origin of the spherical co-ordinate system to B_{local}'s center (see Figure 3). The Equation (43) is used to obtain the following series

$$\begin{aligned}
&\overline{F^s_{ij,n,m}}(\overrightarrow{O_1Q}) \\
&\quad=(-1)^n\sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}\overline{S_{n+n',m+m'}}(\overrightarrow{O_2O_1})\Big(\frac{\lambda+3\mu}{\lambda+2\mu}\delta_{ij}R_{n',m'}(\overrightarrow{O_2Q}) \\
&\quad+\frac{\lambda+\mu}{\lambda+2\mu}(\overrightarrow{O_2O_1})_j\frac{\partial}{\partial x_i}R_{n',m'}(\overrightarrow{O_2Q})-\frac{\lambda+\mu}{\lambda+2\mu}(\overrightarrow{O_2Q})_j\frac{\partial}{\partial x_i}R_{n',m'}(\overrightarrow{O_2Q})\Big)
\end{aligned} \tag{48}$$

$$\begin{aligned}
&\overline{G^s_{j,n,m}}(\overrightarrow{O_1Q}) \\
&=(-1)^n\sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}\overline{S_{n+n',m+m'}}(\overrightarrow{O_2O_1})\frac{\lambda+\mu}{\lambda+2\mu}\frac{\partial}{\partial x_j}R_{n',m'}(\overrightarrow{O_2Q})
\end{aligned} \tag{49}$$

Substituting Equations (48) and (49) into Equation (45), we can obtain

$$\begin{aligned}
&\sum_{J=1}^{N_b}\int_{\Gamma_I}u^J_{ij}h_I(Q)x'_{Ji}\mathrm{d}\Gamma(Q) \\
&=\frac{1}{8\pi\mu}\sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}\int_{\Gamma_I}F^R_{ij,n',m'}(\overrightarrow{O_2Q})h_I(Q)\mathrm{d}\Gamma(Q)L^1_{i,n',m'}(O_2) \\
&+\frac{1}{8\pi\mu}\sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}\int_{\Gamma_I}G^R_{j,n',m'}(\overrightarrow{O_2Q})h_I(Q)\mathrm{d}\Gamma(Q)L^2_{i,n',m'}(O_2)
\end{aligned} \tag{50}$$

where

$$L^1_{i,n',m'}(O_2)=\sum_{n=0}^{\infty}\sum_{m=-n}^{n}(-1)^{n'}\overline{S_{n+n',m+m'}}(\overrightarrow{O_1O_2})M^1_{i,n,m}(O_1) \tag{51}$$

$$L^2_{i,n',m'}(O_2)=\sum_{n=0}^{\infty}\sum_{m=-n}^{n}(-1)^{n'}\overline{S_{n+n',m+m'}}(\overrightarrow{O_1O_2})(M^2_{i,n,m}(O_1)-(\overrightarrow{O_1O_2})_iM^1_{i,n,m}(O_1)) \tag{52}$$

The above two equations are known as the multipole to local (M2L) translation (see Figure 3).

It has been so far discussed the evaluation of influences of a cluster of far nodes included in a leaf B_{far} on a node of another leaf B_{local}, where B_{far} is on B_{local}'s interaction list.

If B_{far}'s parent belongs to the interaction list of B_{local}'s parent (see Figure 4), we first translate the multipole moments about the center of B_{far} to the center of B_{far}'s parent by using the following two equations, which can be obtained from Equations (44), (46) and (47):

$$M^1_{i,n,m}(O') = \sum_{n'=0}^{n} \sum_{m'=-n'}^{n'} R_{n',m'}(\overrightarrow{O'O}) M^1_{i,n-n',m-m'}(O) \tag{53}$$

$$M^2_{i,n,m}(O') = \sum_{n'=0}^{n} \sum_{m'=-n'}^{n'} R_{n',m'}(\overrightarrow{O'O})(M^2_{i,n-n',m-m'}(O) - (\overrightarrow{OO'})_i M^1_{i,n-n',m-m'}(O)) \tag{54}$$

The above two equations are known as multipole to multipole (M2M) translation.

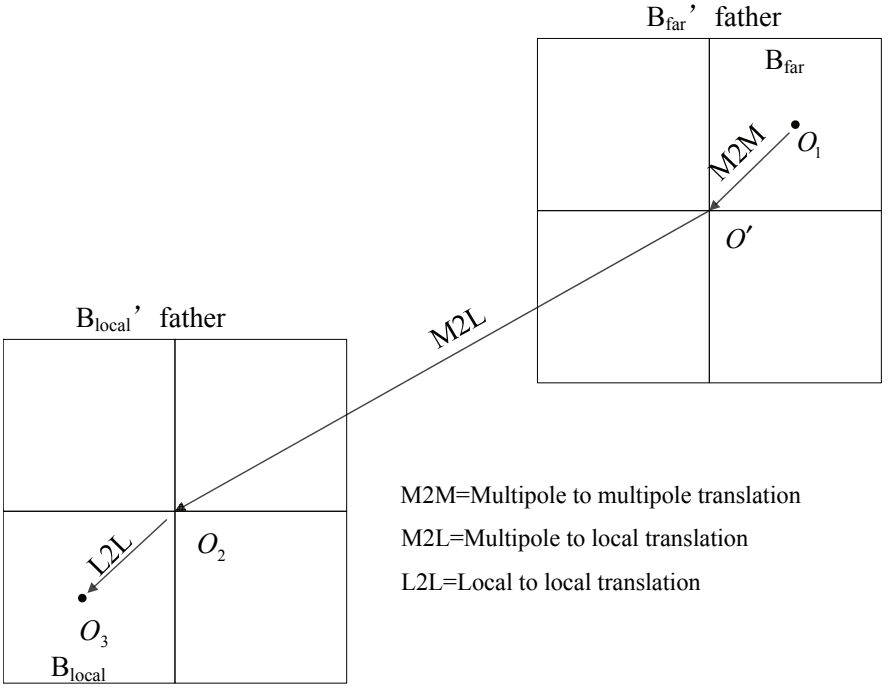

Figure 4: Conversions of multipole to local expansions

Then by using Equations (51) and (52), we translate the multipole moments about the center of B_{far}'s parent to the local moments of B_{local}'s parent. Finally the center of the local moments will be shift from the center of B_{local}'s parent to B_{local}. It can

be obtained from Equations (44) and (50):

$$\begin{aligned}&\sum_{J=1}^{N_b}\int_{\Gamma_I} u_{ij}^J h_I(Q)x'_{Ji}d\Gamma(Q)\\&=\frac{1}{8\pi\mu}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}A^1_{ij,n',m'}L^1_{i,n',m'}(O_3)+\frac{1}{8\pi\mu}\sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}A^2_{j,n',m'}L^2_{i,n',m'}(O_3)\end{aligned}\tag{55}$$

where

$$A^1_{ij,n',m'}=\int_{\Gamma_I}F^R_{ij,n',m'}(\overrightarrow{O_3Q})h_I(Q)\mathrm{d}\Gamma(Q)\tag{56}$$

$$A^2_{j,n',m'}=\int_{\Gamma_I}G^R_{j,n',m'}(\overrightarrow{O_3Q})h_I(Q)\mathrm{d}\Gamma(Q)\tag{57}$$

$$L^1_{i,n',m'}(O_3)=\sum_{n=n'}^{\infty}\sum_{m=-n}^{n}R_{n-n',m-m'}(\overrightarrow{O_2O_3})L^1_{i,n,m}(O_2)\tag{58}$$

$$L^2_{i,n',m'}(O_3)=\sum_{n=n'}^{\infty}\sum_{m=-n}^{n}R_{n-n',m-m'}(\overrightarrow{O_2O_3})(L^2_{i,n,m}(O_2)-(\overrightarrow{O_2O_3})_iL^1_{i,n,m}(O_2))\tag{59}$$

The above Equations (58) and (59) are used to shift the center of the local moments from the center of B_{local}'s parent to B_{local}, which are known as local to local (L2L) translation.

If B_{far}'s grandparent belongs to the interactions list of B_{local}'s grandparent, we first translate the multipole moments $M^1_{i,n,m}$ and $M^2_{i,n,m}$ from the centre of B_{far} to the centre of B_{far}'s parent, then from the centre of B_{far}'s parent to the centre of B_{far}'s grandparent by Equations (53) and (54). Then, we convert the multipole moments $M^1_{i,n,m}$ and $M^2_{i,n,m}$ into $L^1_{i,n,m}$ and $L^2_{i,n,m}$ from the centre of B_{far}'s grandparent to the centre of B_{local}'s grandparent by Equations (51) and (52). Finally, we translate the local moments $L^1_{i,n,m}$ and $L^2_{i,n,m}$ from the centre of B_{local}'s grandparent to the centre of B_{local}'s parent, and next to the centre of B_{local}, by Equations (58) and (59). In general, if one of B_{far}'s ancestors at level l belongs to the interaction list of one of B_{local}'s ancestors, the above process is repeated recursively until level l.

The process for computing the sum (33) is exactly the same as the sum (32). Using the relation between t^J_{ij} and u^J_{ij}:

$$t^J_{ij}(s_J,Q)=\frac{\partial}{\partial Q_l}u^J_{ik}(s_J,Q)c_{klpj}n_p(Q)\tag{60}$$

where

$$c_{klpj}=\lambda\delta_{kl}\delta_{pj}+\mu(\delta_{kp}\delta_{lj}+\delta_{kj}\delta_{lp})$$

it can obtain exactly the same formulations about the multipole moments and M2M, M2L and L2L translations, and Equation (55) is replaced by

$$\begin{aligned}&\sum_{J=1}^{N_b}\int_{\Gamma_I} t_{ij}^J h_I(Q)x'_{Ji}d\Gamma(Q)\\&=\frac{1}{8\pi\mu}\sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}B^1_{ij,n',m'}L^1_{i,n',m'}(O_2)+\frac{1}{8\pi\mu}\sum_{n'=0}^{\infty}\sum_{m'=-n'}^{n'}B^2_{j,n',m'}L^2_{i,n',m'}(O_2)\end{aligned}\tag{61}$$

where

$$B^1_{ij,n',m'}=\int_{\Gamma_I}\frac{\partial}{\partial Q_l}F^R_{ik,n',m'}(\overrightarrow{O_2Q})c_{klpj}n_p(Q)h_I(Q)\mathrm{d}\Gamma(Q)\tag{62}$$

$$B^2_{j,n',m'}=\int_{\Gamma_I}\frac{\partial}{\partial Q_l}G^R_{k,n',m'}(\overrightarrow{O_2Q})c_{klpj}n_p(Q)h_I(Q)\mathrm{d}\Gamma(Q)\tag{63}$$

In practical computations, the sum in the infinite series (51), (52), (53), (54), (55), (58), (59) and (61) are truncated after p terms. The estimation of errors and proofs can be found in the work of Greegard (1988).

4 The Algorithm for FM-HBNM

In this section, the detail of the procedure for the algorithm proposed is summarized. The restarted preconditioned GMRES is employed as the iterative equation solver. An adaptive version of the FMM (Greegard, 1988) with a hierarchy of boxes which refine the computational domain is used.

Step 1: Discretization: Create nodes which disturbed on the boundary of the domain in the same manner as in the original Hybrid BNM.

Step 2: *Construction of oct-tree structure*: Consider a smallest cube that can contain the entire domain needed to compute. Use it as the root box (which is considered as level 0) of the hierarchical decomposition of the domain. Choose the maximum number of nodes contained in a leaf. Construct the hierarchy of boxes using an oct-tree data structure.

Step 3: *Determination of matrixes associated with some boxes*: Choose the desired multipole expansion order p. With each box that $l>1$, associate six $p\times p$ matrixes which describe the multipole moments $M^1_{i,n,m}$ and $M^2_{i,n,m}$ about the box center. With each box that $l>2$, associate six $p\times p$ matrixes which describe the multipole moments $L^1_{i,n,m}$ and $L^2_{i,n,m}$ about the box center. With each leaf associate two matrices of size $p\times p\times 3\times 3\times N_b$ described $A^1_{ij,n',m'}$ and $B^1_{ij,n',m'}$ and two matrices of size $p\times p\times 3\times N_b$ described $A^2_{j,n',m'}$ and $B^2_{j,n',m'}$, where N_b is the number of nodes contained in that leaf. Associate other two matrices U_L and T_L of size $3\times 3\times N_b\times N_n$

to the leaf described the near-field coefficients of U and T which will be computed directly, with N_n being the number of nodes contained in the neighborhood of that leaf.

Step 4: *Computational of the integral associated with leaf*: For each leaf, compute U_L, T_L, $A^1_{ij,n',m'}$, $A^2_{j,n',m'}$, $B^1_{ij,n',m'}$ and $B^2_{j,n',m'}$ by Equations (23), (24), (56), (57), (62) and (63), respectively.

Step 5: *computation of the multipole moments (upward)*: For an iteration vector x'_{Ji} form multipole moments $M^1_{i,n,m}$ and $M^2_{i,n,m}$ about the centre of each leaf from all the nodes included in that leaf by Equations (46) and (47). Now consider a non leaf box of level l. Compute multipole moments $M^1_{i,n,m}$ and $M^2_{i,n,m}$ about the centre of each box at level l by merging multipole moments from its children using Equations (53) and (54) (M2M in Figure 4). This procedure is repeated for $l \geq 2$ tracing the tree structure of boxes obtained in step upward (decreasing l).

Step 6: *Computation of the coefficients of the local expansions (downward)*: Consider boxes at level l from level 2 to the finest level. For each box a at level l, convert the multipole moments $M^1_{i,n,m}$ and $M^2_{i,n,m}$ of each box b in the interaction list of box a to a local expansion about the centre of box a, using Equations (51) and (52) (M2L in Figure 4).

If $l > 2$, then shift the local expansion of a's parent to itself, using Equations (58) and (59) (L2L in Figure 4).

Add these two local expansions together.

Step 7: *Evaluation of the integral in sum (32) and (33)*: The contribution of the far field is computed by Equations (55) and (61) while the contribution of the near field deduced by the nodes contained in the neighborhood of the leaf is computed directly. Add the two parts.

Step 8: Update: Update the candidate vector and go back to step 5.

In the above algorithm, the multipole and local moments associated with a box at each level are reused to the full extent. The local expansion of a leaf is reused for all the nodes belonging to the leaf in Equations (55) and (61); and in the conversion of multipole moments into a local expansion (Equations (51) and (52)), the multipole moments of a box is repeatedly used for the boxes whose interaction list contains that box.

5 Numerical Results

The proposed techniques have been implemented in C++. In this section, three numerical examples are presented to demonstrate the performance of the method.

For the purpose of error estimation, a formula is defined as

$$e = \frac{1}{|u|_{\max}} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (u_i^{(e)} - u_i^{(n)})^2} \tag{64}$$

where $u_i^{(e)}$ and $u_i^{(n)}$ refer to the exact and numerical solutions respectively and $|u|_{\max}$ is the maximum value of $u$over N nodes.

In all three examples, the properties are given by: Young's modules E=1.0 and Poisson's ratio v=0.25. The problems are solved using two solvers: direct (Gaussian elimination method) and indirect (FMM). The Gaussian elimination method is adopted when it is capable of solving the problem. In the FMM, a restarted preconditioned GMRES(m) with m=25 is employed as the preconditioner being the inverse of the blocked diagonal matrix corresponding to the nodes in leaves. All the infinite expansions are truncated after p=8 and the maximum number of boundary nodes in a leaf box is set to be 60. The iteration is terminated when the relative error is less than 10^{-5}. All the computations are performed on a PC with a 2.67 GHz CPU and 2.0 GB RAM.

5.1 One-dimensional strain problem

Figure 5 describes the one-dimensional strain problem in a cube. The cube is bounded by the planes $x = \pm 1, y = \pm 1, z = \pm 1$. In Figure 5, the boundary conditions on the faces are:

On the face $z = 1$: $u_z = 0$, $t_x = 0$ and $t_y = 0$;

On the face $y = 1$: $u_y = 0$, $t_x = 0$ and $t_z = 0$;

On the face $x = -1$: $u_x = 0$, $t_y = 0$ and $t_z = 0$;

On the face $x = 1$: $t_x = \sigma_0 = 1.0$, $t_y = 0$ and $t_z = 0$;

On the other faces: $t_x = 0$, $t_y = 0$ and $t_z = 0$.

Table 1: Results for the one-dimensional strain problem

DOF	Levels	Iterations	T_{FMM}(s)	T_{dir}(s)	E_{FMM}	E_{dir}	T_iter(s)
1 800	3	24	35	7	4.1×10^{-4}	4.1×10^{-4}	1.5
4 050	3	31	43	85	2.7×10^{-4}	2.7×10^{-4}	1.4
7 200	4	38	64	477	2.3×10^{-4}	2.3×10^{-4}	1.7
16 200	4	46	197	(5 440)	2.1×10^{-4}	——	4.3
28 800	5	56	553	(30 565)	1.3×10^{-4}	——	9.9
64 800	5	62	963	(347 130)	9.6×10^{-5}	——	15.5
115 200	6	68	2 796	(1 956 191)	6.8×10^{-5}	——	41.1
180 000	6	76	4 023	(7 462 277)	4.9×10^{-5}	——	52.9

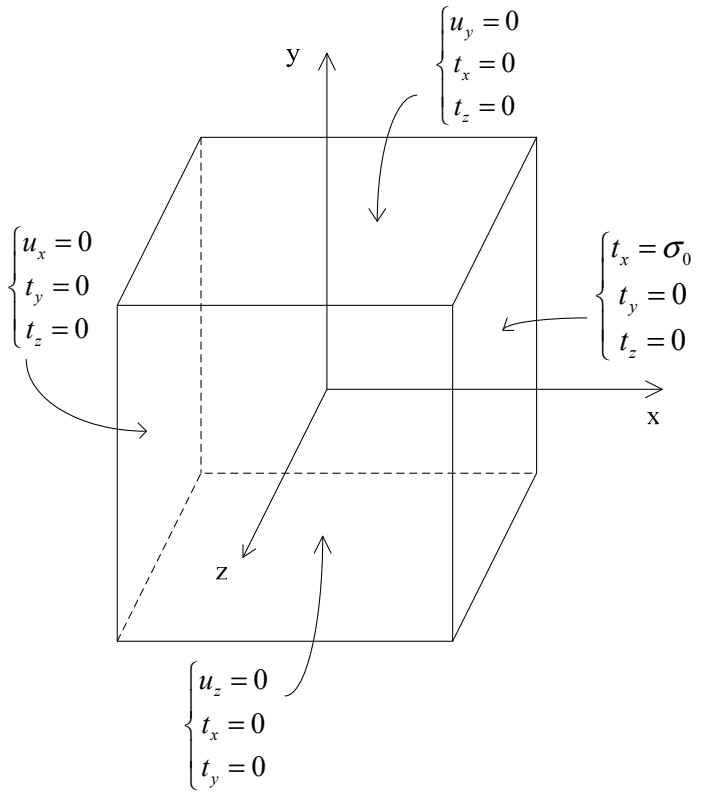

Figure 5: One-dimensional strain problem

Table 1 shows the results of the one-dimensional strain problem solved by direct and indirect solvers. The first, second and third columns list the degrees of freedom (DOF), number of levels used in the multipole hierarchy, and number of iterations of GMRES, respectively. The fourth and fifth columns indicate the time consumption of FM-HBNM and the conventional Hybrid BNM with direct solver. Only the cases, of which the unknowns are less than 10000, were considered in the computation when the direct solver was used. The stated times in parentheses are estimated by extrapolation. In the sixth and seventh columns, the relative errors of u_x are presented. The relative error is evaluated over 20 sample points uniformly distributed along a line segment from (0.0, 0.0, 0.0) to (1.0, 0.0, 0.0). In the eighth column, the time of FM-HBNM for per iteration is presented. From Table 1, we can see that when the DOF is more than 4050, the FM-HBNM has an advantage over the conventional Hybrid BNM. The results demonstrate that the FM-HBNM is extremely

effective for large-scale computation and the computational cost is nearly proportional to the problem size. It can also be concluded that FM-HBNM overcomes the restriction of the memory.

5.2 Thick-walled cylindrical pressure vessel

The problem of a thick-walled pressure vessel provides another test case for the validation of the FM-HBNM. Consider a thick-walled cylinder with an open end as shown in Figure 6(a). It is loaded by internal pressure p_i and external pressure p_o as shown in Figure 6(b). It has inner radius r_i and outer radius r_o.

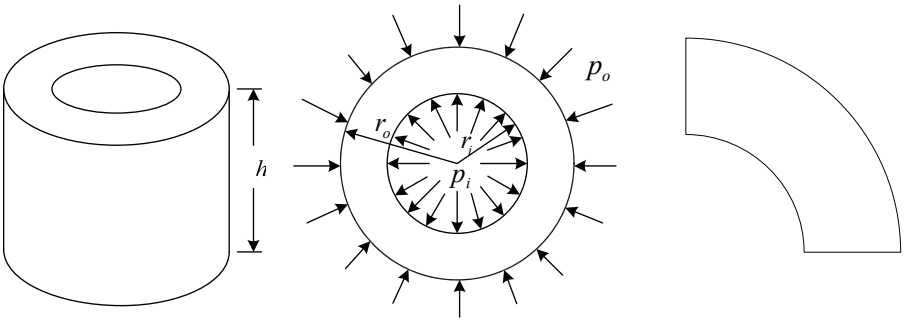

Figure 6: Thick-walled cylindrical pressure vessel

The analytical solution for the circumferential stress (σ_θ) of a thick-walled cylindrical pressure vessel is

$$\sigma_\theta = -\frac{r_i^2 r_o^2}{r_o^2 - r_i^2}\frac{p_o - p_i}{r^2} + \frac{r_i^2 p_i - r_o^2 p_o}{r_o^2 - r_i^2} \tag{65}$$

The problem was solved with $r_i = 10$, $r_0 = 25$, $h = 10$, $p_o = 0$, $p_i = 10$. Due to the symmetry, only one quarter of the geometry is considered (see Figure 6(c)).

The results of the thick-walled cylindrical pressure vessel problem are shown in Table 2, which has the same structure as Table 1. In the sixth and seventh columns, the relative error of σ_θ is evaluated over 20 sample points uniformly distributed along the radius from 10 to 25 and the high accuracy of FM-HBNM can be observed. In Table 2, it can be seen that when the DOF is more than 1878, the FM-HBNM is more effective than the conventional Hybrid BNM. Compare the results in Table 1 with that in Table 2, we can see that there is a little difference in the last columns. In Table 1, the time consumption per iteration for the case of 1800 DOF is much

Table 2: Results for the thick-walled cylindrical pressure vessel problem

DOF	Levels	Iterations	T_{FMM}(s)	T_{dir}(s)	E_{FMM}	E_{dir}	T_iter(s)
1 878	4	25	6	8	9.3×10^{-3}	9.3×10^{-3}	0.24
5 154	5	22	29	172	5.1×10^{-3}	5.1×10^{-3}	1.3
11 724	5	23	129	(2 024)	3.2×10^{-3}	——	5.6
20 496	6	33	234	(10 816)	2.4×10^{-3}	——	7.1
46 050	6	37	830	(122 682)	1.5×10^{-3}	——	22.4
81 786	7	40	987	(687 277)	1.1×10^{-3}	——	24.7
183 876	7	52	3 863	(7 810 347)	7.6×10^{-4}	——	74.3

larger than that of 1878 DOF in Table 2. One of the reasons may due to the numbers of boxes in the interaction lists. In the first case, the max number of boxes in the interaction lists is 49 while that is 8 in the second case. Actually, if one truncates the infinite series in the multipole expansion taking p terms then the computational costs for M2M, M2L and L2L translations are proportional to $O(p^4)$, $O(np^4)$ and $O(p^4)$, respectively, where n is the number of boxes in the interaction list. In the worst case, n will be 189 and the M2L translation would be the bottleneck in the algorithm of FMM. This may explain the results discussed above. However, Greengard and Rokhlin (1997) have developed a new version of FMM, in which they use a new diagonal form for translation operators and further reduce the computational cost of M2L translation to $O(p^3)$.

5.3 Cubes with holes

In order to show the advantages of FM-HBNM further, the third example deals with complicated geometries. The models are cubes with different number of spherical holes, which are distributed uniformly (Figure 7) or randomly (Figure 8). Models of the cubes with increasing number of holes, are considered. Each model of the cube has the same dimension of $2\times2\times2$, containing a total of $m\times m\times m$ holes, with m=2, 4, 6, 8, 10 and 12. The radiuses of the holes are $0.4/m$ for the models contain $m\times m\times m$ holes. In each model, only unit displacement along the direction of the outer normal is applied on the boundary of the cube (the boundary conditions of the other directions are traction-free). Traction-free boundary conditions are applied on all edges of the holes.

Table 3 shows the results of different cases for cubes with holes by FM-HBNM. The first, second and third columns list the number of holes, the total DOF and the different models (uniformly distributed holes or randomly distributed holes), respectively. The fourth and fifth columns indicate the number of levels used in the multipole hierarchy and iteration steps of GMRES. In the sixth column, the total time consumption of FM-HBNM is presented. From Table 3, one can observe that

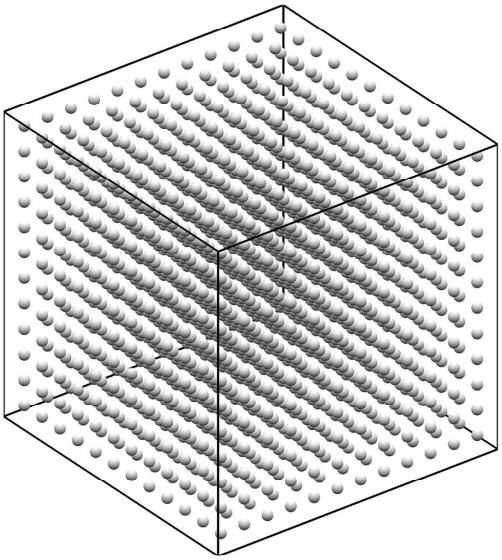

Figure 7: A cube with 1728 uniformly distributed holes

Table 3: Results for FM-HBNM for cubes with many holes

Holes	DOF	Case	Levels	Iterations	T_{FMM}(s)
8	5 928	uniform	3	12	24
		random	4	16	35
64	34 824	uniform	4	16	549
		random	5	18	470
216	113 256	uniform	6	16	1 108
		random	6	19	1 442
512	265 992	uniform	6	16	4 388
		random	6	19	3 623
1000	517 800	uniform	7	19	7 898
		random	7	20	7 274
1728	893 448	uniform	7	23	14 941
		random	7	21	13 114

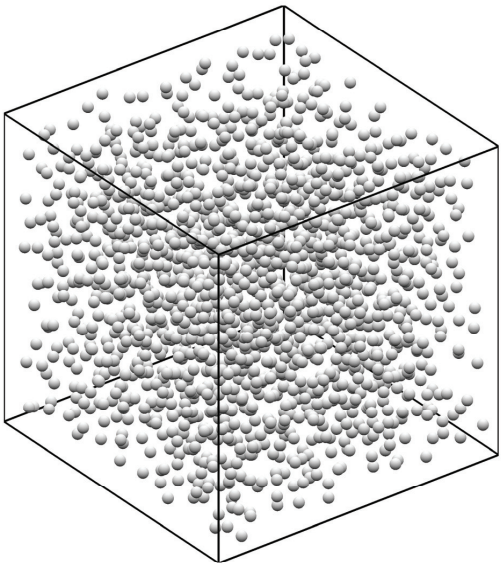

Figure 8: A cube with 1728 randomly distributed holes

the algorithm proposed converged fast: in the largest model with a total DOF of 893448, the iteration steps are 23 and 21 for the uniform case and random case, respectively. Figure 9 shows the CPU time per iteration with different number of unknowns.

6 Conclusions and Discussion

In this paper, the FMM has been successfully implemented into the Hybrid BNM for 3D elasticity problems. Formulations for the local and multipole expansions and the conversion of multipole to local expansion are presented. The proposed techniques have been implemented in C++. Three numerical examples are presented to demonstrate the nearly linear complexity of the FM-HBNM and high accuracy.

The FM-HBNM for 3D elasticity in this paper retains the advantages of both the meshless method and the fast solver. It not only can save computing time and memory, but also can simplify the discretization tasks for problems with complicated geometries. Therefore, the proposed method is especially applicable for large-scale problems and problems with complicated geometries. The research work in this

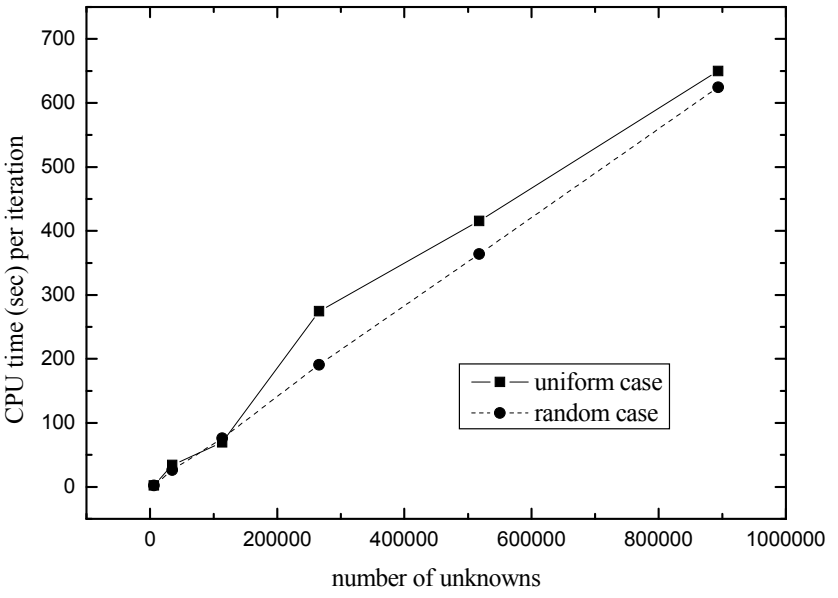

Figure 9: CPU time (sce) per iteration with different number of unknowns

paper is a basis for the implementation of the FM-HBNM for 3D elasticity to more complex problems.

As it can be seen in the numerical examples, the conventional FMM used in this paper has one bottleneck, that is, the M2L translation. The new version of FMM can further reduce the computational time of the M2L translation. The incorporation of the new FMM into Hybrid BNM is a subject of our future research.

Acknowledgement: Financial support for the project from the Natural Science Foundation of China (No. 50808090).

References

Atluri, S.N. (2004): The Meshless Local Petrov-Galerkin (MLPG) Method for Domain & Boundary Discretizations, Tech Science Press, 665 pages.

Atluri, S.N.; Han, Z.D.; Rajendran, A.M. (2004): A New Implementation of the Meshless Finite Volume Method, Through the MLPG "Mixed" Approach, *CMES:*

Computer Modeling in Engineering & Science, vol. 6, no. 6, pp. 491-514.

Atluri, S.N.; Han, Z.D.; Shen, S. (2003): Meshless Local Petrov-Galerkin (MLPG) approaches for weakly-singular traction & displacement boundary integral equations, *CMES: Computer Modeling in Engineering & Science*, vol. 4, no. 5, pp. 507-517.

Atluri, S.N.; Kim, H.G.; Cho, J.Y. (1999): A critical assessment of the truly meshless local Petrov-Galerkin (MLPG), and local boundary integral equation (LBIE) methods. *Computational Mechanics*, vol. 24, pp. 348-372.

Atluri, S.N.; Liu, H.T.; Han, Z.D. (2006a): Meshless Local Petrov-Galerkin (MLPG) Mixed Collocation Method for Elasticity Problems, *CMES: Computer Modeling in Engineering & Science*s, vol. 14, no. 3, pp. 141-152.

Atluri, S.N.; Liu, H.T.; Han, Z.D. (2006b): Meshless Local Petrov-Galerkin (MLPG) Mixed Finite Difference Method for Solid Mechanics, *CMES: Computer Modeling in Engineering & Science*s, vol. 15, no. 1, pp. 1-16.

Atluri, S.N.; Shen S.P. (2002a): The meshless local Petrov-Galerkin (MLPG) method. Tech. Science Press, 440 pages.

Atluri, S.N.; Shen S.P. (2002b): The meshless local Petrov-Galerkin (MLPG) method: A simple & less-costly alternative to the finite element and boundary element methods, *CMES: Computer Modeling in Engineering & Science*, vol. 3, no. 1, pp. 11-52.

Atluri, S.N.; Zhu, T. (1998): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, vol. 22, pp. 117-127.

Atluri, S.N.; Zhu, T. (1998): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, vol. 22, pp. 117-127.

Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P. (1996): Meshless method: an overview and recent development. *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 3-47.

Belytschko, T.; Lu, Y.Y.; Gu, L. (1994): Element free Galerkin methods. *International Journal for Numerical Methods in Engineering*, vol. 37, pp. 229-256.

Chati, M.K.; Mukherjee, S.; Mukherjee, Y.X. (1999): The boundary node method for three-dimensional linear elasticity. *International Journal for Numerical Methods in Engineering*, vol. 46, pp. 1163-1184.

DeFigueredo, T.G.B.; Brebbia, C.A. (1989): A new hybrid displacement variational formulation of BEM for elastostatics. In *Advances in Boundary Elements*, Brebbia CA, Conner JJ (eds), vol. 1. Southampton: *Computational Mechanics*

Publications, pp. 47-57.

Fu, Y.H.; Klimkowski, K.J.; Rodin, G.J.; et al. (1998): A fat solution method for three dimensional many-particle problems of linear elasticity. *International Journal for Numerical Methods in Engineering*, vol. 42, pp. 1215-1229.

Greegard, L. (1988): *The Rapid Evaluation of Potential Fields in Particle Systems*. Cambridge, MIT Press.

Greengard, L.; Rokhlin, V. (1987): A fast algorithm for particles simulations. *Journal of Computational Physics*, vol. 73, pp. 325-348.

Greengard, L.; Rokhlin, V. (1997): A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, vol. 6, pp. 229-269.

Hayami, K.; Sauter, S.A. (1998a): Cost estimation of the panel clustering method applied to 3-D elastostatics. In: Brebbia CA, eds. *Proceedings of the Second European Boundary Element Method Symposium*, EUROBEM 98. Southampton: *Computational Mechanics* Publications, pp. 33-42.

Hayami, K.; Sauter, S.A. (1998b): Panel clustering for 3-D elastostatics using spherical harmonics. In: Kassab A, Brebbia CA and Chopro M, eds. *Proceedings of boundary elements*. Southampton: *Computational Mechanics* Publications, pp. 289-298.

Kulkarni, S.S.; Telekunta, S.; Mukherjee, S. (2003): Application of an accelerated boundary-based mesh-free method to two-dimensional problems in potential theory. *Computational Mechanics*, vol. 32, pp. 240-249.

Liu, Y.J.; Nishimura, N. (2006): The fast multipole boundary element method for potential problems: A tutorial. *Engineering Analysis with Boundary Elements*, vol. 30, pp. 371-381.

Liu, Y.J.; Nishimura, N.; Otani, Y.; Takahashi, T.; Chen, X.L.; Munakata, H. (2005): A fast boundary element method for the analysis of fiber-reinforced composites based on a rigid-inclusion model. *Journal of Applied Mechanics*, vol. 72, pp.115-128.

Miao, Y.; Wang, Q.; Liao, B.H.; Zheng, J.J. (2009): A Dual Hybrid Boundary Node Method for 2D elastodynamics Problems. *CMES: Computer Modeling in Engineering and Sciences*, vol. 53, no. 1, pp. 1-22.

Miao, Y.; Wang, Y.H. (2006): Meshless analysis for three-dimensional elasticity with singular hybrid boundary node method. *Applied Mathematics and Mechanics*, vol. 27, no. 6, pp. 673-681.

Miao, Y.; Wang, Y.H.; Yu, F. (2005): Development of hybrid boundary node method in two-dimensional elasticity. *Engineering Analysis with Boundary Elements*, vol. 29, pp. 703-712.

Mukherjee, Y.X.; Mukherjee, S. (1997): The boundary node method for potential problems. *International Journal for Numerical Methods in Engineering*, vol. 40, pp. 797-815.

Nabors, K.; Korsmeyer, F.T.; Leighton, F.T.; White, J. (1994): Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *SIAM Journal on Scientific Computing*, vol. 15, pp. 713-735.

Nishida, T.; Hayami, K. (1997): Application of the fast multipole to the 3D BEM analysis of electron guns. In *Boundary Elements XIX*, Marchettia M, Brebbia CA, Aliabadi MH (eds). Southampton: *Computational Mechanics* Publications, pp. 613-622.

Peirce, A.P.; Napier, J.A.L. (1995): A spectral multipole method for efficient solutions of large scale boundary element models in elastostatics. *International Journal for Numerical Methods in Engineering*, vol. 38, pp. 4009-4034.

Popov, V.; Power, H. (2001): An $O(N)$ Taylor series multipole boundary element methods for three-dimensional elasticity problems. *Engineering Analysis with Boundary Elements*, vol. 25, pp. 7-18.

Rokhlin, V. (1985): Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, vol. 60, pp. 187-207.

Saad, Y.; Schultz, M.H. (1986): GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear system. *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp.856-869.

Yoshida, K.; Nishimura, N.; Kobayashi, S. (2001): Application of fast multipole Galerkin boundary integral equation method to elastostatic crack problems in 3D. *International Journal for Numerical Methods in Engineering*, vol. 50, pp. 525-547.

Zhang, J.M.; Tanaka, M. (2007): Systematic study of thermal properties of CNT composites by the fast multipole hybrid boundary node method. *Engineering Analysis with Boundary Element*, vol. 31, pp. 388-401.

Zhang, J.M.; Tanaka, M. (2008): Fast HdBNM for large-scale thermal analysis of CNT-reinforced composites. *Computational Mechanics*, vol. 41, pp. 777-787.

Zhang, J.M.; Tanaka, M.; Endo, M. (2005): The hybrid boundary node method accelerated by fast multipole expansion technique for 3D potential problems. *International Journal for Numerical Methods in Engineering*, vol. 63, pp. 660-680.

Zhang, J.M.; Tanaka, M.; Matsumoto, T. (2004): Meshless analysis of potential problems in three dimensions with the hybrid boundary node method. *International Journal for numerical methods in Engineering*, vol. 59, pp. 1147-1160.

Zhang, J.M.; Yao, Z.H. (2001): Meshless regular hybrid boundary node method.

CMES: Computer Modeling in Engineering and Sciences, vol. 2, pp. 307-318.

Zhang, J.M.; Yao, Z.H. (2004): The regular hybrid boundary node method for three-dimensional linear elasticity. *Engineering Analysis with Boundary Element*, vol. 28, pp. 525-534.