

Liang Shen · Yijun J. Liu

An adaptive fast multipole boundary element method for three-dimensional potential problems

Received: 28 June 2005 / Accepted: 24 January 2006 / Published online: 15 March 2006
© Springer-Verlag 2006

Abstract An adaptive fast multipole boundary element method (FMBEM) for general three-dimensional (3-D) potential problems is presented in this paper. This adaptive FMBEM uses an adaptive tree structure that can balance the multipole to local translations (M2L) and the direct evaluations of the near-field integrals, and thus can reduce the number of the more costly direct evaluations. Furthermore, the coefficients used in the preconditioner for the iterative solver (GMRES) are stored and used repeatedly in the direct evaluations of the near-field contributions. In this way, the computational efficiency of the adaptive FMBEM is improved significantly. The adaptive FMBEM can be applied to both the original FMBEM formulation and the new FMBEM with diagonal translations. Several numerical examples are presented to demonstrate the efficiency and accuracy of the adaptive FMBEM for studying large-scale 3-D potential problems. The adaptive FMBEM is found to be about 50% faster than the non-adaptive version of the new FMBEM in solving the model (with 558,000 elements) for porous materials studied in this paper. The computational efficiencies and accuracies of the FMBEM as compared with the finite element method (FEM) are also studied using a heat-sink model. It is found that the adaptive FMBEM is especially advantageous in modeling problems with complicated domains for which free meshes with much more finite elements would be needed with the FEM.

Keywords Fast multipole method · Boundary element method · Three-dimensional potential problems

1 Introduction

The boundary element method (BEM) involves formation of a dense system matrix and solution of the linear system $\mathbf{A}\lambda = \mathbf{b}$. Each coefficient A_{ij} in matrix \mathbf{A} represents the interaction between source point i and field point j . Computing \mathbf{A} is a time consuming process in the BEM, which takes $O(N^2)$ operations (with N being the number of nodes/elements). This process becomes a huge obstacle that limits the use of the BEM. Solving the system of equations is another expensive task, which requires $O(N^3)$ operations with direct solvers. Many works have been devoted in extending the applicability of the BEM to large-scale problems with parallel computing or sub-domain techniques. Still the BEM has been restricted to solving relatively small-size problems with N around 10,000 on a personal computer (PC), until the recent development of the fast multipole method (FMM).

With the help of the FMM and iterative solvers such as the generalized minimal residual method (GMRES), the fast multipole BEM (FMBEM) is now capable of solving large-scale problems on a PC in a reasonable amount of time. The GMRES was originally developed in 1980s by Saad and Schultz [1]. It is an iterative projection method designed to solve large-scale and non-symmetrical linear systems $\mathbf{A}\lambda = \mathbf{b}$. The FMM was first introduced by Rokhlin [2] and Greengard [3, 4] in the 1980s to accelerate the evaluation of interactions of large ensembles of particles. The FMM reduces the operation counts from $O(N^2)$ to $O(N)$ for N -body interaction problems. The FMBEM adopts the same mechanism to accelerate the multiplication of system matrix \mathbf{A} and solution vector λ , where the element-to-element interactions are replaced by box-to-box interactions with each box containing a certain number of elements. Combining the GMRES and FMM, the FMBEM can now solve large-scale problems, for example, with N above one million using a mid-range laptop PC within a few hours for 2-D potential problems [5, 6].

The potential problem has been studied by many using the FMBEM besides the original work by Rokhlin [2] and Greengard [3, 4]. A comprehensive review of the FMBEM,

L. Shen · Y. J. Liu (✉)
Computer-Aided Engineering Research Laboratory,
Department of Mechanical Engineering,
University of Cincinnati,
Cincinnati, OH 45221-0072, USA
E-mail: yijun.liu@uc.edu
Tel.: +1-513-5564607
Fax: +1-513-5563390

including that for potential problems, can be found in Ref. [7]. Nishimura's group [7–11] has done extensive research on the development of the FMBEM to solve large-scale 3-D scalar crack problems in both static and dynamic cases. Chew's group (see, e.g., Ref. [12] and the references therein) has been modeling electromagnetic wave scatterings using the FMBEM, and have obtained remarkable results. For example, they have demonstrated that the scattering data from a full aircraft model at GHz frequencies can be obtained using several millions of boundary elements and within a few hours. Most recently, very large 3-D BEM models with more than 28 million elements have been solved successfully by the FMBEM within hours and with moderate computing resources [13, 14] in modeling composite materials.

However, there are still rooms for improvements with the FMBEM regarding the solution efficiencies. The FMM can significantly accelerate the BEM solutions, especially for 2-D problems. For example, if the elements are uniformly distributed in a 2-D domain, the FMBEM takes approximately $29(N/s)p^2 + 2Np + 9Ns$ [15] operations to finish the multiplication of matrix \mathbf{A} and vector λ , where p is the expansion order, and s the number of elements per *box* (see Sect. 3.2 for the definition of a "box"). Choosing $s = p$, the total operation count for 2-D yields $40Np$. While for 3-D problems, the FMM is considered less efficient than in 2-D problems. The operation count to perform the multiplication is approximately $2Np^2 + 189(N/s)p^4 + 27Ns$ [15]. Choosing $s = 2p^2$, the operation count yields $150Np^2$. To achieve high accuracies, the expansion order p has to be high, and the p^2 term is hard to be ignored.

To reduce the operation count for 3-D problems, White and Head-Gordon [16] came up with the idea of rotating the coordinate system based on the observation that the operations count for spherical harmonic expansion translations in the z direction is $3p^3$, instead of p^4 . The total operations count is thus reduced to $2Np^2 + 189(N/s)3p^3 + 27Ns$ [15], if one aligns the vector connecting the source expansion center and the target expansion center with the z coordinate. The implementation of this scheme requires minor modifications on the standard FMM scheme and no additional approximation is made, so the error bound for the standard FMM is retained.

Greengard and Rokhlin [17] introduced a new version of the FMM for the 3-D Laplace equation that reduces the operation count to $2Np^2 + (N/s)(20p^3 + 189p^2) + 27Ns$ [15]. This new version of the FMM has been implemented by Nishimura's group [8] for 3-D crack problems and recently by Yao's group [18] for 3-D elasticity problems. Cheng et al. [19] further introduced an adaptive algorithm to implement the new version of the FMM. However, this adaptive algorithm is only for the evaluation of pair-wise interactions of large ensembles of particles (some errors are found in this algorithm, which will be discussed later in this paper). To the best knowledge of the authors, the adaptive algorithm has not been implemented for the FMM combined with the BEM.

This paper introduces an adaptive FMBEM algorithm to solve large-scale general 3-D potential problems. It adopts

and improves the adaptive FMM algorithm introduced by Cheng et al. [19] into the FMBEM implementation. The adaptive algorithm can be applied not only to the new version of the FMBEM with diagonal translation, but also to the original FMBEM. Some new improvements regarding the integration of the FMM and GMRES are employed. The FMBEM implemented in this paper employs the new version of the FMBEM [17] and preconditioned GMRES to achieve better performance. The coefficients used in the preconditioner for GMRES are stored and used repeatedly in the direct evaluations of the near-field contributions. Thus, computational efficiency in the FMBEM solution is greatly improved.

This paper is organized as follows: the basic FMBEM formulation is introduced in Sect. 2. The adaptive FMBEM algorithm is presented in Sect. 3. Section 4 presents the numerical results, and Sect. 5 concludes the paper with further discussions.

2 Formulation of the FMBEM

2.1 Conventional BEM formulation

The boundary integral equation (BIE) for the potential problem is [20, 21]:

$$C(\mathbf{x})\phi(\mathbf{x}) = \int_S [G(\mathbf{x}, \mathbf{y})q(\mathbf{y}) - F(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})] dS(\mathbf{y}), \quad \forall \mathbf{x} \in S, \quad (1)$$

where the coefficient $C(\mathbf{x}) = 1/2$ if the boundary S is smooth at the source point \mathbf{x} , $G(\mathbf{x}, \mathbf{y})$ is the Green's function given by:

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi r}, \quad \text{with } r = |\mathbf{x} - \mathbf{y}|, \quad (2)$$

for 3-D problems and

$$F(\mathbf{x}, \mathbf{y}) = \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{y})} = -\frac{1}{4\pi r^2} \frac{\partial r}{\partial n(\mathbf{y})}, \quad (3)$$

with n being the outward normal at the field point \mathbf{y} .

Using constant elements [20], the corresponding discretized form of BIE (1) is:

$$\sum_{j=1}^N f_{ij}\phi_j = \sum_{j=1}^N g_{ij}q_j, \quad \text{for } i = 1, 2, \dots, N, \quad (4)$$

where

$$g_{ij} = \int_{\Delta S_j} G(\mathbf{x}_i, \mathbf{y}) dS(\mathbf{y}), \quad (5)$$

$$f_{ij} = \frac{1}{2}\delta_{ij} + \int_{\Delta S_j} F(\mathbf{x}_i, \mathbf{y}) dS(\mathbf{y}). \quad (6)$$

with ΔS_j being an element. Rearranging Eq. (4) with all the unknown boundary values in vector λ leads to:

$$\mathbf{A}\lambda = \mathbf{b}, \quad (7)$$

where \mathbf{A} is the coefficient matrix which needs $O(N^2)$ operations in the conventional BEM and vector \mathbf{b} is the known right-hand side vector which also needs $O(N^2)$ operations to compute.

2.2 Basic FMM formulas

In this section, the basic FMM formulas are listed for convenience, which will be used to implement the adaptive FMBEM. Most of the formulas in this section can be found in Ref. [22].

Fundamental solution G in Eq. (2) can be expanded with a series expansion as:

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi |\mathbf{x} - \mathbf{y}|} \cong \frac{1}{4\pi} \sum_{n=0}^p \sum_{m=-n}^n \overline{S_{n,m}(\mathbf{x}, \mathbf{y}_c)} \times R_{n,m}(\mathbf{y}, \mathbf{y}_c), \quad |\mathbf{x} - \mathbf{y}_c| > |\mathbf{y} - \mathbf{y}_c|, \quad (8)$$

where \mathbf{y}_c is the expansion center, p the number of expansion terms, and $\overline{(\cdot)}$ indicates the complex conjugate. The functions $S_{n,m}$ and $R_{n,m}$ are solid harmonic functions [22].

The kernel F in Eq. (3) can also be expanded as:

$$F(\mathbf{x}, \mathbf{y}) = \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{y})} \cong \frac{1}{4\pi} \sum_{n=0}^p \sum_{m=-n}^n \overline{S_{n,m}(\mathbf{x}, \mathbf{y}_c)} \times \frac{\partial R_{n,m}(\mathbf{y}, \mathbf{y}_c)}{\partial n(\mathbf{y})}, \quad |\mathbf{x} - \mathbf{y}_c| > |\mathbf{y} - \mathbf{y}_c|. \quad (9)$$

Applying expansions in Eqs. (8) and (9), one can evaluate the integrals in Eqs. (5) and (6) on a collection of elements within S_0 (a subset of S) as follows:

$$\int_{S_0} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) dS(\mathbf{y}) \cong \frac{1}{4\pi} \sum_{n=0}^p \sum_{m=-n}^n \overline{S_{n,m}(\mathbf{x}, \mathbf{y}_c)} \times M_{n,m}(\mathbf{y}_c), \quad |\mathbf{x} - \mathbf{y}_c| > |\mathbf{y} - \mathbf{y}_c|, \quad (10)$$

$$\int_{S_0} F(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) dS(\mathbf{y}) \cong \frac{1}{4\pi} \sum_{n=0}^p \sum_{m=-n}^n \overline{S_{n,m}(\mathbf{x}, \mathbf{y}_c)} \times \tilde{M}_{n,m}(\mathbf{y}_c) \quad |\mathbf{x} - \mathbf{y}_c| > |\mathbf{y} - \mathbf{y}_c|, \quad (11)$$

where $M_{n,m}$ and $\tilde{M}_{n,m}$ are multipole moments centered at \mathbf{y}_c and defined as:

$$M_{n,m}(\mathbf{y}_c) = \int_{S_0} R_{n,m}(\mathbf{y}, \mathbf{y}_c) q(\mathbf{y}) dS(\mathbf{y}), \quad (12)$$

$$\tilde{M}_{n,m}(\mathbf{y}_c) = \int_{S_0} \frac{\partial R_{n,m}(\mathbf{y}, \mathbf{y}_c)}{\partial n(\mathbf{y})} \phi(\mathbf{y}) dS(\mathbf{y}). \quad (13)$$

The multipole expansion center can be moved from \mathbf{y}_c to $\mathbf{y}_{c'}$ according to the following formula (M2M translation):

$$M_{n,m}(\mathbf{y}_{c'}) = \int_{S_0} R_{n,m}(\mathbf{y}, \mathbf{y}_{c'}) q(\mathbf{y}) dS(\mathbf{y}) = \sum_{n'=0}^n \sum_{m'=-n'}^{n'} R_{n',m'}(\mathbf{y}_{c'}, \mathbf{y}_c) M_{n-n',m-m'}(\mathbf{y}_c). \quad (14)$$

Similarly for $\tilde{M}_{n,m}$.

In the original FMM formulation, the multipole moments are translated to the local expansion coefficients using the following formula (M2L translation):

$$L_{n,m}(\mathbf{x}_1) \cong \sum_{n'=0}^p \sum_{m'=-n'}^{n'} (-1)^n \overline{S_{n+n',m+m'}(\mathbf{x}_1, \mathbf{y}_c)} \times M_{n',m'}(\mathbf{y}_c), \quad |\mathbf{y}_c - \mathbf{x}_1| > |\mathbf{x} - \mathbf{x}_1|. \quad (15)$$

In the new FMM formulations [17], the multipole expansion is translated to local expansion via the exponential expansion, where the translation operators become diagonal. For example, integrals in Eq. (5) can be expanded using the following exponential expansions:

$$\int_{S_0} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) dS(\mathbf{y}) \cong \frac{1}{4\pi} \sum_{k=0}^{S(\varepsilon)} \sum_{j=1}^{M_k} W_{k,j} \times \exp \left\{ -\frac{\lambda_k}{d} (y_3 - x_3) \right\} \exp \left\{ i \frac{\lambda_k}{d} \left((y_1 - x_1) \cos(\alpha_{j,k}) + (y_2 - x_2) \sin(\alpha_{j,k}) \right) \right\} \quad (16)$$

where $S(\varepsilon)$ can be determined by the required accuracy (with tolerance ε), $\alpha_{j,k} = 2\pi j/M_k$, and d is the cube length. The vector $\vec{x}\vec{y}$ must satisfy the following inequality to ensure Eq. (16) is valid:

$$d \leq y_3 - x_3 \leq 4d, \quad 0 \leq \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} \leq 4\sqrt{2}d.$$

The multipole expansion is translated to exponential expansion using the following formula (M2X translation):

$$W_{k,j}(\mathbf{y}_c) = \frac{\omega_k/d}{M_k} \sum_{m=-n}^p (-i)^{|m|} e^{im\alpha_{j,k}} \sum_{n=|m|}^p \frac{M_n^m(\mathbf{y}_c)}{\sqrt{(n-m)!(n+m)!}} (\lambda_k/d)^n, \quad (17)$$

where the values for the weight ω_k , the node λ_k , and integer array M_k can be found in Ref. [19].

The exponential expansion center is then shifted from point \mathbf{y}_c to point \mathbf{x}_1 (X2X translation) by:

$$V_{k,j}(\mathbf{x}_1) = W_{k,j}(\mathbf{y}_c) \exp \left\{ -\frac{\lambda_k}{d} (y_c x_1)_3 \right\} \times \exp \left\{ i \frac{\lambda_k}{d} \left((y_c x_1)_1 \cos(\alpha_{j,k}) + (y_c x_1)_2 \sin(\alpha_{j,k}) \right) \right\}. \quad (18)$$

And the coefficients for local expansion centered at \mathbf{x}_l can be obtained from the exponential expansion (X2L translation):

$$L_n^m(\mathbf{x}_1) = \frac{(-i)^{|m|}}{\sqrt{(n-m)!(n+m)!}} \sum_{k=1}^{S(\varepsilon)} \left(\frac{\lambda_k}{d} \right)^n \sum_{j=1}^{M_k} V_{k,j}(\mathbf{x}_1) e^{im\alpha}. \quad (19)$$

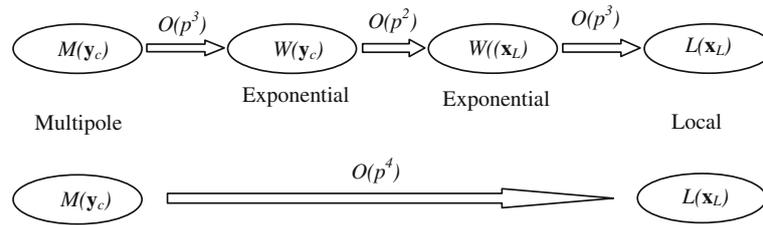


Fig. 1 M2X, X2X, X2L, and M2L translations

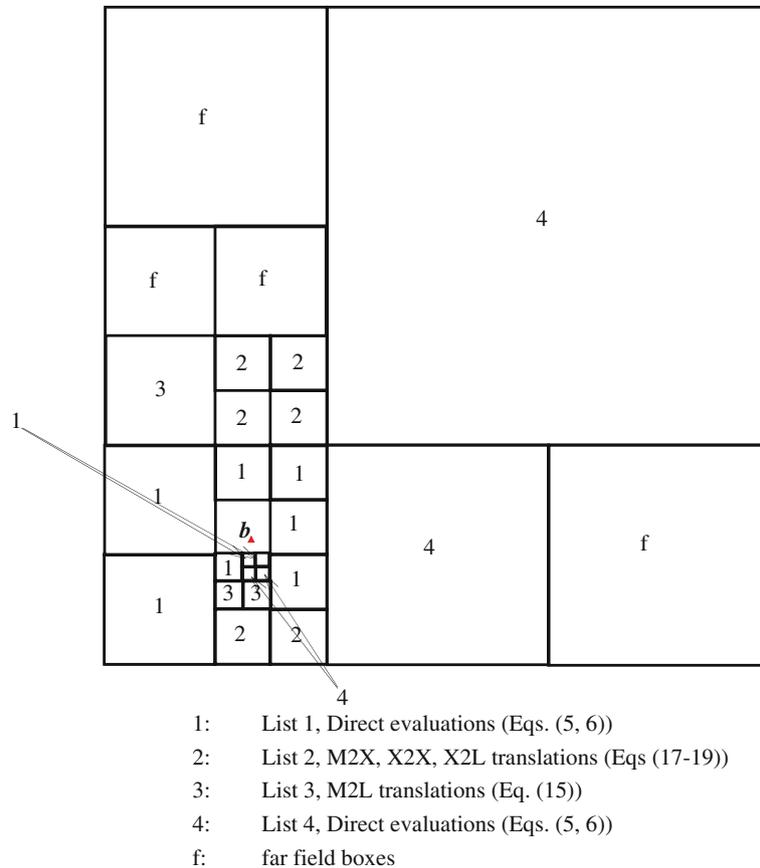


Fig. 2 Four lists of boxes associated with box *b*

To make the above translations (17), (18), and (19) valid, the inequality $d \leq y_3 - x_3 \leq 4d$ must be satisfied, which means point \mathbf{y}_c is above point \mathbf{x}_1 . If point \mathbf{y}_c is not above point \mathbf{x}_1 , the multipole moments need to be rotated to make this assumption valid. After M2X, X2X and X2L translations, the local coefficients need to be rotated back. The direct M2L translations from \mathbf{y}_c to \mathbf{x}_1 and translations via exponential expansions are illustrated in Fig. 1.

The local expansion center can be shifted from \mathbf{x}_1 to \mathbf{x}_l using the following L2L formula:

$$L_{n,m}(\mathbf{x}_l) \cong \sum_{n'=0}^p \sum_{m'=-n}^{n'} R_{n'-n,m'-m}(\mathbf{x}_l, \mathbf{x}_1) L_{n',m'}(\mathbf{x}_1). \quad (20)$$

Finally each term in Eq. (4) can be evaluated using the local expansion:

$$\sum_{j=1}^N f_{ij} \phi_j \text{ or } g_{ij} q_j = \frac{1}{4\pi} \sum_{n=0}^{\infty} \sum_{m=-n}^n R_{n,m}(\mathbf{x}_i, \mathbf{x}_1) L_{n,m}(\mathbf{x}_1), \quad (21)$$

for far-field contributions.

In the following section, we present a new adaptive algorithm to implement the FMBEM for general 3-D potential problems, which can further improve the efficiency of the FMBEM as compared with the current approaches (the standard non-adaptive FMBEM) reported in the literature.

3 The adaptive algorithm

The adaptive algorithm for accelerating the FM-BEM for 3-D potential problems is described in the following subsections. The proposed adaptive algorithm corrects some errors in some special cases in the original algorithm given by Cheng et al. [19] for particle-interaction problems. Differences between the adaptive algorithm and non-adaptive algorithm will be addressed.

3.1 Initialization

Choose the number of expansion terms based on the required precision. Select the maximum number of elements allowed in a childless box (to be called a *leaf*).

3.2 Tree structure

In the adaptive algorithm, an adaptive hierarchical tree of boxes is constructed first by dividing the problem domain into smaller and smaller sub-domains. On the 0th level, we have a single box containing the entire domain. Boxes on level $l + 1$ are obtained by subdividing each box on level l into eight equal sized child boxes, then trim off empty child boxes. On the same level, two boxes are said to be *colleagues* if they share at least a boundary point (a box is considered a colleague of itself), otherwise, they are said to be well separated. Every box b starting from level 2 has an *interaction list*, consisting of the children of colleagues of b 's parent box, which are well separated from b .

To implement the adaptive algorithm, the following four lists associated with box b on level l are defined (as illustrated in Fig. 2):

- *List 1* consists all childless boxes adjacent to childless box b . If b is a parent box, its list 1 is empty.
- *List 2* is the interaction list of box b .
- *List 3* includes all boxes c on level $l + 1$ that are not adjacent to childless box b but separated from b with a box the same size as c . List 3 also includes all childless boxes on level $l - 1$ that are not adjacent to b (b does not have to be childless) but are separated from b with a box the same size as b .
- *List 4* consists of all boxes c above level $l + 1$ that are not adjacent to childless box b but separated from b with a box the same size as c . List 4 also includes all childless boxes c under level $l - 1$ that are not adjacent to b (b does not have to be childless) but are separated from b with a box the same size as b .

In the non-adaptive FM-BEM algorithm, only two lists are defined. If b is a leaf, list 1 contains all same level boxes adjacent to b . If b is not a leaf, list 1 contains all the same level childless boxes adjacent to b . List 2 is the interaction list of box b .

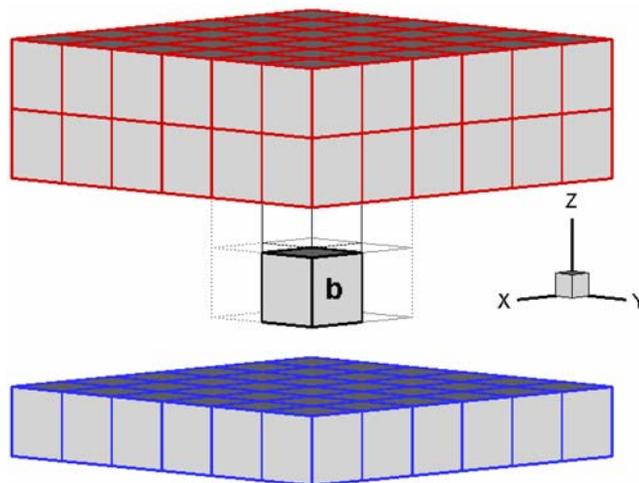


Fig. 3 The Uplist (blue) and Downlist (green) of box b (red)

To perform M2X and X2L translations in the new FMM, list 2 needs to be further divided into six sub-lists associated with six coordinate directions:

- *Uplist* contains all boxes located above b ($z > z_b$).
- *Downlist* contains all boxes lie below b ($z < z_b$).
- *Northlist* contains boxes c with $y_c > y_b$ except boxes in the up and down lists.
- *Southlist* contains boxes c with $y_c < y_b$ except boxes in the up and down lists.
- *Eastlist* contains boxes c with $x_c > x_b$ except boxes in the up, down, north, and south lists.
- *Westlist* contains boxes c with $x_c < x_b$ except boxes in the up, down, north and south lists.

A general case of the Uplist and Downlist of box b is shown in Fig. 3.

3.3 Upward pass

For each childless box b , calculate multipole moments at its center from all elements in b using Eqs. (12) and (13). Use M2M (14) to translate moments from b 's center to its parent's center. After this step, a p th-order multipole expansion is formed for each box b at its center, representing the contribution from all the elements in box b to elements far away from b . Non-adaptive algorithm has the same upward pass as the adaptive algorithm.

3.4 Downward pass

Starting from level 2 to the lowest level:

Step A

- (1) For each box b on level l , use direct M2L (Eq. (15)) to translate the multipole moments of list 3 boxes to the local coefficients of b .

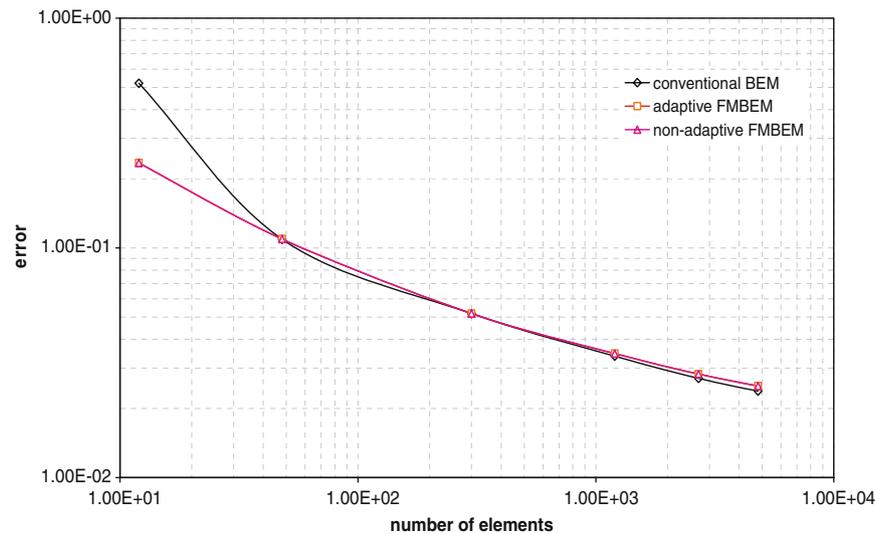


Fig. 4 Comparison of L_2 -Norm error of conventional BEM and adaptive FMBEM for the cube model

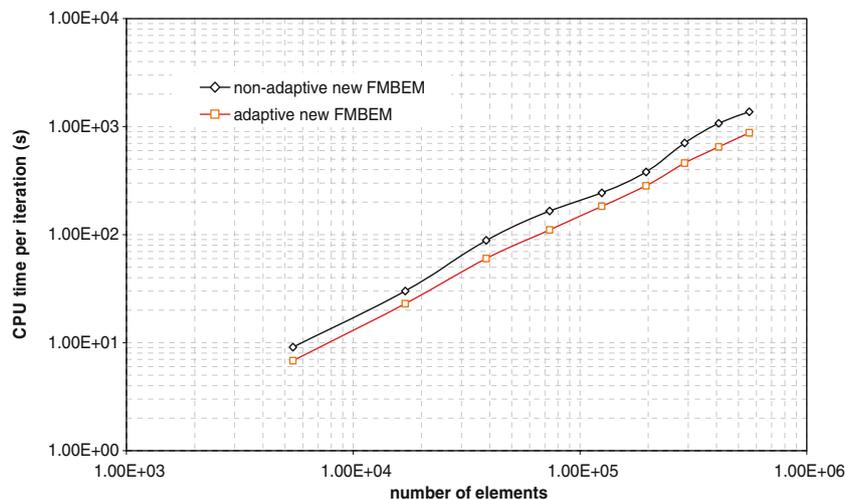


Fig. 5 Comparison of time/iteration between adaptive and non-adaptive FMBEM for the porous models

- (2) For each box c on level l , convert the multipole moments of c to the local coefficients of box b in list 2 of c using M2X, X2X, X2L (Eqs. (17), (18), and (19)).

The adaptive algorithm given in Ref. [19] groups same set of boxes into different lists 3 and 4. It selects M2L (Eq. (15)) or direct evaluation (Eqs. (5) and (6)) for these boxes based on the number of charges (elements) in box b . However, when dealing with list 4 boxes defined in this paper, the distance between collocation point \mathbf{x} and b 's center \mathbf{x}_1 can be either very close to or larger than the distance between list 4 box's center \mathbf{y}_c and b 's center \mathbf{x}_1 , which can make the M2L translation (Eq. (15)) inaccurate or invalid. The result will not converge in the invalid cases using the original adaptive algorithm.

In the direct evaluation (Eqs. (5) and (6)), expensive analytical or numerical integrations are involved, which means M2L (Eq. (15)) is often faster than the direct evaluation.

Therefore, the new version of the adaptive algorithm in this current paper compares the location of boxes and chooses the M2L translation whenever possible.

Step B Translate local coefficients of b_i to the local coefficients of b_i 's children using L2L translation (Eq. (20)).

In the non-adaptive algorithm, only Step A (2) and Step B are performed.

After downward pass, local coefficients for each leaf are calculated.

3.5 Evaluation of $g_{ij}q_j$ or $f_{ij}\phi_j$

For each leaf b_i , both non-adaptive and adaptive algorithm calculate $g_{ij}q_j$ and $f_{ij}\phi_j$ for each collocation point \mathbf{x}_i from local expansion of b_i using Eq. (21). Add direct evaluation

(Eqs. (5) and (6)) of $g_{ij}q_j$ and $f_{ij}\phi_j$ for j elements in b_i 's list 1 boxes. The adaptive algorithm also performs direct evaluation (Eqs. (5) and (6)) of $g_{ij}q_j$ and $f_{ij}\phi_j$ for j elements in b_i 's list 4 boxes.

3.6 Solution

Use iterative solver GMRES (<http://www.netlib.org>) to solve the Eq. (7), where \mathbf{b} and $\mathbf{A}\lambda$ are calculated in upward and downward pass. Initially, λ is chosen to be a zero vector. Calculate $\mathbf{A}\lambda$ iteratively with updated λ vector until the residue reaches an acceptable tolerance.

A block diagonal matrix is used as the preconditioner according to Nishida and Hayami [23] to achieve faster convergences. The entries in the preconditioner are the coefficients directly evaluated by Eqs. (5) and (6). The preconditioner will be decomposed into a lower triangular matrix and an upper triangular matrix (LU decomposition) at the beginning of the solution process. To avoid direct evaluations of these coefficients again in the iterations, one can retrieve the coefficients from the stored LU decomposition using the following relations:

$$\begin{aligned} A_{ij} &= L_{i1}U_{1j} + L_{i2}U_{2j} + \dots + L_{ii}U_{ij}, & i < j, \\ A_{ij} &= L_{i1}U_{1j} + L_{i2}U_{2j} + \dots + L_{ii}U_{jj}, & i = j, \\ A_{ij} &= L_{i1}U_{1j} + L_{i2}U_{2j} + \dots + L_{ij}U_{jj}, & i > j, \end{aligned} \quad (22)$$

where L and U are the lower triangular matrix and upper triangular matrix, respectively.

4 Numerical results

The new adaptive FMBEM algorithm has been implemented in Fortran 90, and tested using several examples. All models are meshed with triangular constant elements. Analytical integrations (Eqs. (5) and (6)) are used, so that singular and nearly singular difficulties are calculated exactly. 3×3 quadrature points are used in the calculations of the multipole moments in Eqs. (12) and (13). In the FMM implementation, we choose eight terms in the multipole, local, and exponential expansions, and a maximum of 100 elements in a leaf. Tolerance for the iterative solver is 10^{-6} . The first two models are run on a laptop computer with a 1.6GHz CPU and 512MB memory.

4.1 A cube with linear temperature distribution

A unit cube ($-0.5 \leq x, y, z \leq 0.5$) is used as the first example to verify the accuracy of the new FMBEM with the adaptive algorithm. The heat conduction in the cube is studied with the following boundary conditions:

$$\begin{aligned} \phi &= 0 \text{ at } x = -0.5, & \phi &= 1 \text{ at } x = 0.5, \\ & \text{and } q &= 0 \text{ on all other surfaces.} \end{aligned}$$

The problem is solved by the conventional BEM, adaptive FMBEM, and non-adaptive FMBEM (both the adaptive and

non-adaptive FMBEM's use the new formulas with diagonal translations, Eq. (18)). Figure 4 shows the L_2 -norm errors of these three methods. All three methods converge quickly. The L_2 -norm error differences between FMBEM and BEM are small and negligible. The adaptive FMBEM has the same accuracy as the non-adaptive FMBEM, and both are slightly less accurate than the conventional BEM for larger models because of the truncation error introduced in the multipole moment expansion.

4.2 Porous material models

Porous material models are used in the second example. One major advantage of the BEM over the finite element method (FEM) is that it can handle complex geometries easily due to its boundary discretization features. Porous materials can have very complicated geometries that require large numbers of elements to model using either the BEM or FEM. Thus, the FMBEM is very attractive in the modeling of porous materials regarding both modeling and solution efficiencies. The main purpose of this example is to compare the performance of the adaptive FMBEM with the non-adaptive version.

The porous model has a dimension $L \times L \times L$, containing $L \times L \times L$ randomly distributed spherical voids of the same size (radius = 0.3), where L has the value from 1 to 10 in different models. The number of elements ranges from 828 (for a one-void block) to 558,000 (for a 1,000-void block) accordingly. All the models are subject to the same boundary conditions:

$$\begin{aligned} \phi &= \pm L/2 \text{ at } x = \pm L/2 \\ & \text{and } q = 0 \text{ on all other surfaces,} \end{aligned}$$

where origin of the coordinate system is located at the center of the block.

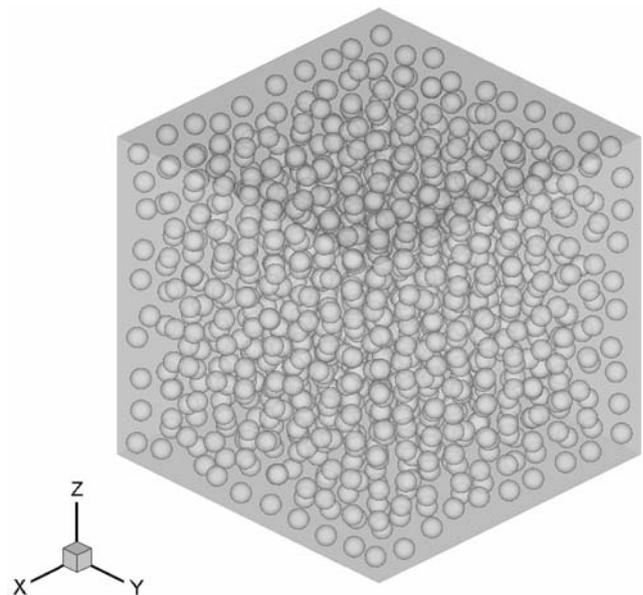


Fig. 6 A porous block with 1,000 spherical voids

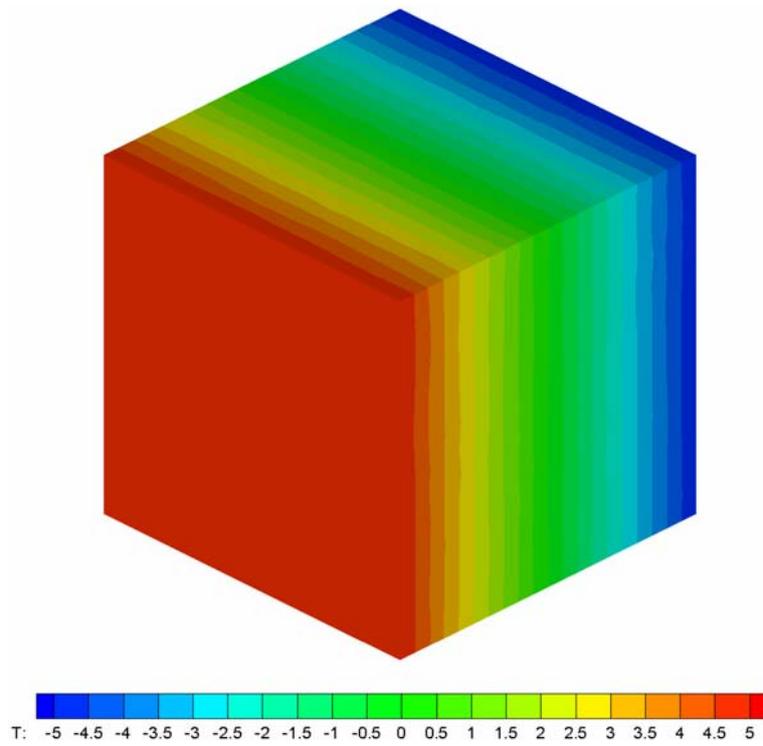


Fig. 7 Contour plot of the computed potential (temperature) for the porous block model

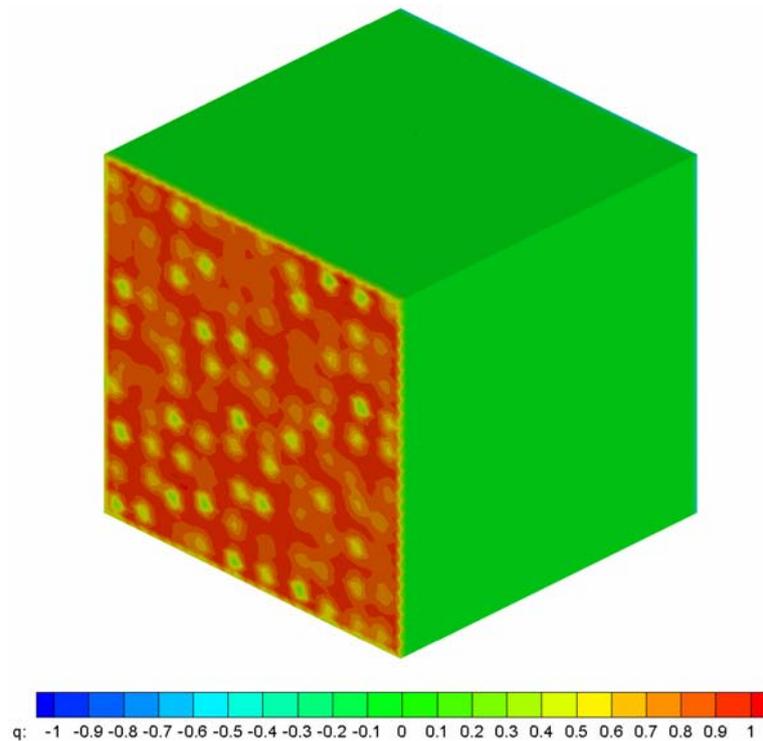


Fig. 8 Contour plot of the computed normal derivative of the potential (heat flux) for the porous block model

Figure 5 compares the performance between the non-adaptive FMBEM and the adaptive FMBEM (both use the new formulas with diagonal translations, Eqs. (17), (18) and (19)). It is shown that the adaptive FMBEM runs much faster than the non-adaptive FMBEM. For the largest model with 558,000 elements, the adaptive FMBEM takes 878 s per

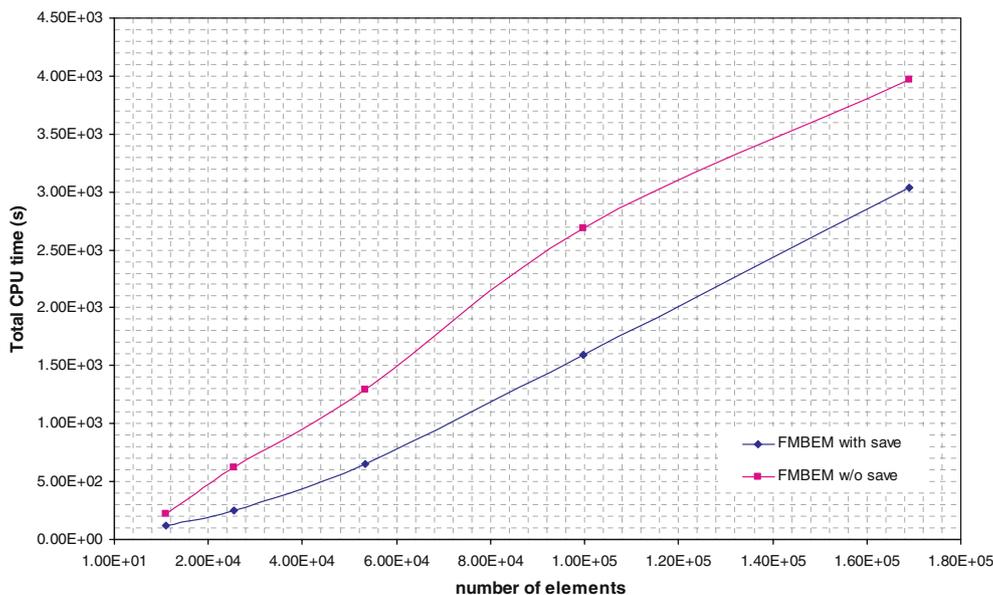


Fig. 9 Total CPU time of the adaptive FMBEM with or without saving the coefficients

iteration, while the non-adaptive version takes 1,375 s per iteration. The adaptive FMBEM is about 56% faster than the non-adaptive FMBEM. If the element number increases and the geometry become more complex, the adaptive algorithm is expected to be even more efficient than the non-adaptive version.

Figure 6 shows the BEM model containing 1,000 voids. Figures 7 and 8 show the contour plots of the results for the same porous block. The analytical solution for the same block without the voids is $\phi(x, y, z) = x$ and thus the range of ϕ should be: $-5 \leq \phi \leq 5$. In Fig. 7, which shows the potential field for the block with the voids, the overall range of the potential is slightly outside of $-5 \leq \phi \leq 5$, while in Fig. 8, the normal derivative of the potential (flux) has values slightly outside the range $-1 \leq q \leq 1$. These results show subtle effects of the voids on the thermal property of the porous material.

The FMBEM can run even faster in the expenses of more memory space. The adaptive FMBEM can store coefficients calculated in the direct evaluation, and retrieve the coefficients in the iterative solution process, thus direct evaluation times are reduced. Figure 9 compares the CPU time of adaptive FMBEM with and without storing the coefficients. The improvement in the CPU time is roughly 40% when the coefficients are saved. However, such improvements largely depend on the available memory size of the computer.

4.3 A heat-sink model

A heat-sink is modeled next using both the FEM (ANSYS) and the developed adaptive FMBEM to further verify the efficiency of the FMBEM code. This example is intended to investigate both the modeling and solution efficiencies of the

FEM and FMBEM for solving practical engineering problems. The heat-sink model is taken from Ref. [24]. A given temperature field ($\phi = 120$) is specified on the bottom surface and a heat flux condition ($\partial\phi/\partial n = 0.2$) is specified on all the other surfaces. The ANSYS mesh is shown in Fig. 10a, where 127,149 nodes were used with the 20-node brick elements. The mesh was obtained by extruding the cross-section so that a mapped mesh was obtained. Another larger ANSYS model with 10-node tetrahedron elements were used in a free mesh, where more than 300,000 nodes were used. The BEM mesh used the same element pattern on the surfaces as shown in Fig. 10a, where each rectangular surface area was divided into two triangular area elements. There are 34,616 nodes in this BEM model.

Figure 10b shows the contour plot of the temperature determined by using ANSYS, while Fig. 10c shows that by the FMBEM. The two results are very close (with the difference in the calculated minimum temperatures equal to 0.67%) and deemed reasonable based on the applied boundary conditions. On a 2.4 GHz laptop PC, the FMBEM code used 8 min with a tolerance of 10^{-3} to calculate the unknown boundary values, while the ANSYS used 1 min for the 20-node brick element model and more than 20 min for the 10-node tetrahedron element model (the two FEM results are the same within the first three significant digits). The larger FEM model may be needed when the heat-sink has more geometric features, such as cooling holes. In addition, meshing the volume using either mapped or free meshes is significantly more time consuming than meshing the surfaces using the BEM. Thus, combining the advantages in the modeling stage for more complicated geometries, the developed FMBEM approach seems to be a very attractive alternative. However, for models with relatively simple geometries where mapped FEM meshes can be obtained readily, the FEM seems to be still

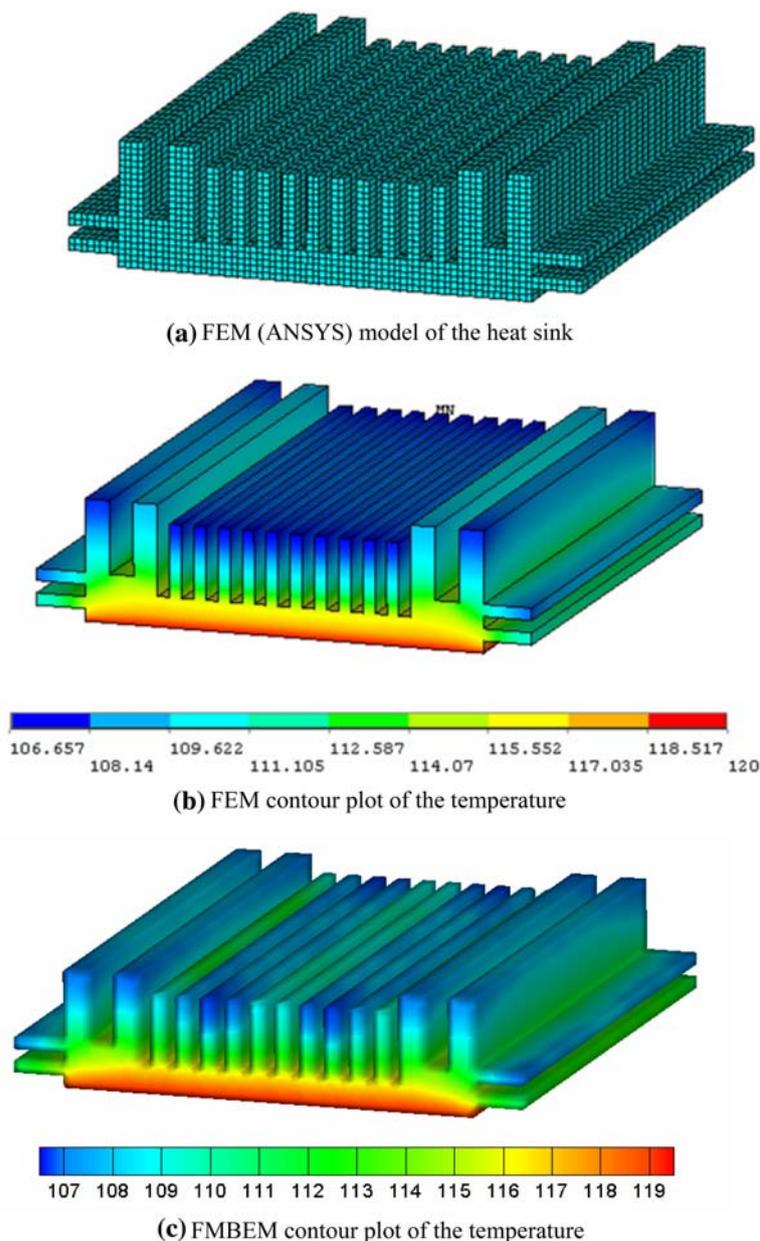


Fig. 10 Comparison of the FMBEM and FEM for analyzing a heat-sink model

faster than the FMBEM with the same accuracy. Further investigations of the FEM and FMBEM for analyzing large-scale realistic engineering problems need to be conducted.

5 Discussions

An adaptive FMBEM has been developed for 3-D potential problems based on the adaptive algorithm originally proposed by Cheng et al. [19] for the particle-interaction simulations. Some corrections in the original algorithm have been made. The overall acceleration with the new adaptive FMBEM is about 50% over the non-adaptive FMBEM when

both the FMBEMs use the diagonal translations (the new FMBEM).

The FMM and GMRES are integrated more closely in this adaptive FMBEM. The coefficients initially stored in the preconditioner for GMRES are reused to calculate $\mathbf{A}\lambda$ in the direct evaluation of the coefficients in the downward pass, so that there is no need to directly evaluate those coefficients repeatedly in each iteration.

In the numerical examples, constant triangular elements are chosen to discretize the BEM models. The singular and nearly singular integrals do not pose any problems because all the integrals can be evaluated analytically with the constant elements. When dealing with higher-order elements

where analytical integrations cannot be performed, the adaptive algorithm can further improve the performance of the FMBEM, because it can avoid the many expensive numerical integrations in the direct evaluation by using the M2L translations.

The present adaptive algorithm can be readily extended to other FMBEM applications, including 3-D elastostatic, acoustic, and elastodynamic problems. The adaptive algorithm can also be applied to other fast methods, such as the fast mesh-free boundary node method [25] and the fast multipole accelerated method of fundamental solutions [26].

References

1. Saad Y, Schultz M (1986) A generalized minimal residual algorithm for solving nonsymmetric linear system. *SIAM J Sci Stat Comput* 7:856–869
2. Rokhlin V (1985) Rapid solution of integral equations of classical potential theory. *J Comput Phys* 60:187–207
3. Greengard LF, Rokhlin V (1987) A fast algorithm for particle simulations. *J Comput Phys* 73:325–348
4. Greengard LF (1988) The rapid evaluation of potential fields in particle systems. The MIT Press, Cambridge, MA
5. Nishimura N, Liu YJ (2004) Thermal analysis of carbon-nanotube composites using a rigid-line inclusion model by the boundary integral equation method. *Comput Mech* 35:1–10
6. Liu YJ, Nishimura N (2005) The fast multipole boundary element method for potential problems – a tutorial. *Eng Anal Bound Elem* (in press)
7. Nishimura N (2002) Fast multipole accelerated boundary integral equation methods. *Appl Mech Rev* 55:299–324
8. Yoshida K, Nishimura N, Kobayashi S (2001) Application of new fast multipole boundary integral equation method to crack problems in 3D. *Eng Anal Bound Elem* 25:239–247
9. Yoshida K, Nishimura N, Kobayashi S (2001) Application of fast multipole Galerkin boundary integral equation method to crack problems in 3D. *Int J Num Meth Eng* 50:525–547
10. Yoshida K, Nishimura N, Kobayashi S (2001) Application of new fast multipole boundary integral equation method to elastostatic crack problems in three dimensions. *JSCE J Struct Eng* 47A:169–179
11. Nishimura N, Yoshida K, Kobayashi S (1999) A fast multipole boundary integral equation method for crack problems in 3D. *Eng Anal Bound Elem* 23:97–105
12. Chew WC et al (2003) Fast integral equation solvers in computational electromagnetics of complex structures. *Eng Anal Bound Elem* 27:803–823
13. Liu YJ, Nishimura N, Otani Y (2005) Large-scale modeling of carbon-nanotube composites by the boundary element method based on a rigid-inclusion model. *Comput Mater Sci* 34:173–187
14. Liu YJ, Nishimura N, Otani Y et al (2005) A fast boundary element method for the analysis of fiber-reinforced composites based on a rigid-inclusion model. *J Appl Mech* 72:115–128
15. Beatson R, Greengard L (1996) A short course on fast multipole methods. In: Ainsworth M et al (eds) *Wavelets, multilevel methods and elliptic PDEs*. Oxford University Press, Oxford, pp 1–37
16. White CA, Head-Gordon M (1996) Rotating around the quartic angular momentum barrier in fast multipole method calculations. *J Chem Phys* 105:5061–5067
17. Greengard L, Rokhlin V (1997) A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Num* 6:229–269
18. Wang H, Yao ZH (2005) A new fast multipole boundary element method for large scale analysis of mechanical properties in 3D particle-reinforced composites. *Comput Model Eng Sci* 7:85
19. Cheng H, Greengard L, Rokhlin V (1999) A fast adaptive multipole algorithm in three dimensions. *J Comput Phys* 155:468–498
20. Brebbia CA, Dominguez J (1989) *Boundary elements – an introductory course*. McGraw-Hill, New York
21. Banerjee PK (1994) *The boundary element methods in engineering*. McGraw-Hill, New York
22. Yoshida K (2001) Applications of fast multipole method to boundary integral equation method. PhD Thesis, Department of Global Environment Engineering, Kyoto University
23. Nishida N, Hayami K (1997) Application of the fast multipole method to the 3-D BEM analysis of electron guns. In: *Boundary elements XIX, Proceedings of 19th International Conference on the boundary element method*. Computational Mechanics Publications
24. Moaveni S (2002) *Finite element analysis – theory and application with ANSYS*. Prentice-Hall, Englewood Cliffs, NJ
25. Zhang J, Tanaka M, Endo M (2005) The hybrid boundary node method accelerated by fast multipole expansion technique for 3D potential problems. *Int J Num Meth Eng* 63:660–680
26. Liu YJ, Nishimura N, Yao ZH (2005) A fast multipole accelerated method of fundamental solutions for potential problems. *Eng Anal Bound Elem* 29:1016–1024