ELSEVIER

Engineering Analysis with Boundary Elements 30 (2006) 371–381

ENGINEERING ANALYSIS *with* BOUNDARY ELEMENTS

www.elsevier.com/locate/enganabound

The fast multipole boundary element method for potential problems: A tutorial

Y.J. Liu [a,*], N. Nishimura [b,1]

[a] *Department of Mechanical, Industrial and Nuclear Engineering, University of Cincinnati, P.O. Box 210072, Cincinnati, OH 45221-0072, USA*
[b] *Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606-8501, Japan*

Received 6 May 2005; accepted 23 November 2005
Available online 15 March 2006

Abstract

The fast multipole method (FMM) has been regarded as one of the top 10 algorithms in scientific computing that were developed in the 20th century. Combined with the FMM, the boundary element method (BEM) can now solve large-scale problems with several million degrees of freedom on a desktop computer within hours. This opened up a wide range of applications for the BEM that has been hindered for many years by the lack of efficiencies in the solution process, although it has been regarded as superb in the modeling stage. However, understanding the fast multipole BEM is even more difficult as compared with the conventional BEM, because of the added complexities and different approaches in both FMM formulations and implementations. This paper is an introduction to the fast multipole BEM for potential problems, which is aimed to overcome this hurdle for people who are familiar with the conventional BEM and want to learn and adopt the fast multipole approach. The basic concept and main procedures in the FMM for solving boundary integral equations are described in detail using the 2D potential problem as an example. The structure of a fast multipole BEM program is presented and the source code is also made available that can help the development of fast multipole BEM codes for solving other problems. Numerical examples are presented to further demonstrate the efficiency, accuracy and potentials of the fast multipole BEM for solving large-scale problems.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Fast multipole method; Boundary element method; 2D potential problems

1. Introduction

The boundary integral equation (BIE) formulations and their numerical solutions using boundary element method (BEM) for mechanics problems originated about 40 years ago. The 2D potential problem was first formulated in terms of a direct BIE and solved by Jaswon [1]. This work was later extended by Rizzo to the vector case—2D elastostatic problem [2]. Following these early works, extensive research efforts have been made for the development of the BIE/BEM. Some of the important textbooks and research volumes in English can be found in Refs. [3–9].

Although the BEM has enjoyed the reputation of easy meshing in modeling for many problems, its efficiency in solutions has been a serious problem for analyzing large-scale models that have emerged in applications with the availability of more powerful computers. For example, while the finite element method (FEM) has been used routinely to solve models with several millions of degrees of freedom (DOF's), the BEM has been limited to solving problems with a few thousands DOF's for many years. This is because the conventional BEM in general produces dense and non-symmetric matrices that, although smaller in sizes, requires $O(N^2)$ operations to compute the coefficients and another $O(N^3)$ operations to solve the system using direct solvers (where N is the number of equations of the linear system).

In the mid of 1980s, Rokhlin and Greengard [10–12] pioneered the innovative fast multipole method (FMM) that can be used to accelerate the solutions of boundary integral equations by several folds, promising to reduce the CPU time in FMM accelerated BEM to $O(N)$. However, it took almost a decade for the mechanics community to realize the potential of the FMM for the BEM. Some of the early research on FMM–BEM in applied mechanics can be found in Refs. [13–17], which show great promises of the FMM–BEM for solving large-scale engineering problems. Most recently, composite material models containing tens of thousands of fibers [18,19] and models of electromagnetic wave scatterings from a full

* Corresponding author. Tel.: +1 513 556 4607; fax: +1 513 556 3390.
E-mail addresses: yijun.liu@uc.edu (Y.J. Liu), nchml@media.kyoto-u.ac.jp (N. Nishimura).
[1] Tel.: +81 (75) 753 7457; fax: +81 (75) 753 7450.

0955-7997/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enganabound.2005.11.006

aircraft at giga hertz frequencies [20] all have been solved successfully by using the FMM–BEM within hours and with moderate computing resources. A comprehensive review of the FMM BIE/BEM can be found in Ref. [21].

However, the use of the fast multipole method has increased the complexity in implementations of the BIE/BEM. Although there are many research papers dealing with the various subjects of the FMM–BEM, very few can be used as introductory materials to learn the new approach for researchers who are familiar with the conventional BEM or for new comers who are often frustrated with even the conventional BIE/BEM formulations and implementations. The research papers are often concise, without detailed information, which made the understanding of the FMM–BEM difficult. To fill the gap and hence promote the FMM–BEM in the BIE/BEM research community, we will provide in this paper an introduction to the FMM–BEM with complete information on formulations, discretizations and implementations using the 2D potential problem as an example.

This paper is organized as follows: in Section 2, we review the BIE formulation for potential problems and the conventional BEM in discretization of the BIE. In Section 3, we present the complete formulations and algorithms used in the FMM–BEM for 2D potential problem. In Section 4, we discuss the programming issues in FMM–BEM using a Fortran code that is available from the authors with this paper. In Section 5, we show some example problems solved by using the provided code to demonstrate the efficiencies of the FMM–BEM for large-scale problems. We conclude the article with some discussions in Section 6.

2. BIE formulation and the conventional BEM approach

The boundary integral equation formulation and its discretization using the conventional BEM for potential problems is summarized in this section. Consider the following Laplace equation governing potential problems (for example, a steady-state heat conduction problem) in a 2D domain V (Fig. 1)

$$\nabla^2\phi(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in V; \tag{1}$$

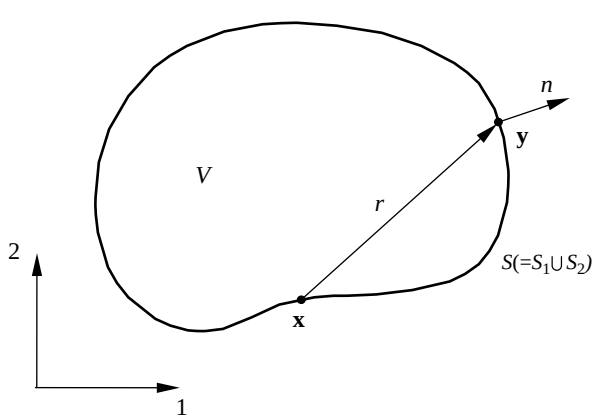

Fig. 1. Domain V and its boundary S.

under the boundary conditions

$$\phi(\mathbf{x}) = \bar{\phi}(\mathbf{x}), \quad \forall \mathbf{x} \in S_1; \tag{2}$$

$$q(\mathbf{x}) \equiv \frac{\partial\phi}{\partial n}(\mathbf{x}) = \bar{q}(\mathbf{x}), \quad \forall \mathbf{x} \in S_2; \tag{3}$$

where ϕ is the potential field in domain V, $S = S_1 \cup S_2$ the boundary of V, n the outward normal, and the barred quantities indicate given values on the boundary.

The solution to the boundary value problem described by Eqs. (1)–(3) can be written as the following representation integral (see, e.g. Refs. [7–9])

$$\phi(\mathbf{x}) = \int_S [G(\mathbf{x},\mathbf{y})q(\mathbf{y}) - F(\mathbf{x},\mathbf{y})\phi(\mathbf{y})]\mathrm{d}S(\mathbf{y}), \quad \forall \mathbf{x} \in V; \tag{4}$$

where $G(\mathbf{x},\mathbf{y})$ is the Green's function given by

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{2\pi}\ln\left(\frac{1}{r}\right), \tag{5}$$

for 2D problems, and

$$F(\mathbf{x},\mathbf{y}) = \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} = \frac{1}{2\pi r}\frac{\partial r}{\partial n} \tag{6}$$

with r being the distance between the collocation point $\mathbf{x}$ and field point $\mathbf{y}$ (Fig. 1).

Letting $\mathbf{x} \to S$ (Fig. 1), we obtain the following boundary integral equation [7–9]

$$C(\mathbf{x})\phi(\mathbf{x}) = \int_S [G(\mathbf{x},\mathbf{y})q(\mathbf{y}) - F(\mathbf{x},\mathbf{y})\phi(\mathbf{y})]\mathrm{d}S(\mathbf{y}), \quad \forall \mathbf{x} \in S; \tag{7}$$

in which the second-term on the right-hand side is a singular integral of the Cauchy-principal value (CPV) type, and the coefficient

$$C(\mathbf{x}) = -\int_S F(\mathbf{x},\mathbf{y})\mathrm{d}S(\mathbf{y}), \tag{8}$$

which is also a CPV integral. If the boundary S is smooth at the collocation point $\mathbf{x}$, we simply have $C(\mathbf{x}) = 1/2$. Note that by substituting (8) into (7), we can obtain the following weakly-singular form of the BIE (see, e.g. Refs. [22,23]) for potential problems

$$\int_S F(\mathbf{x},\mathbf{y})[\phi(\mathbf{y}) - \phi(\mathbf{x})]\mathrm{d}S(\mathbf{y}) = \int_S G(\mathbf{x},\mathbf{y})q(\mathbf{y})\mathrm{d}S(\mathbf{y}), \quad \forall \mathbf{x} \in S; \tag{9}$$

in which no singular integrals exist. One can choose either BIE (7) or (9) in the discretization using the BEM, depending on which form is more convenient.

As an example of discretization schemes, we use constant boundary elements, that is, dividing the boundary S into N line segments (elements) and placing one node on each element (Fig. 2). We obtain the following discretized equation of BIE (7) for node i [7–9]

$$\frac{1}{2}\phi_i = \sum_{j=1}^{N}[g_{ij}q_j - f_{ij}\phi_j], \quad \text{for } i = 1, 2, \ldots, N \tag{10}$$

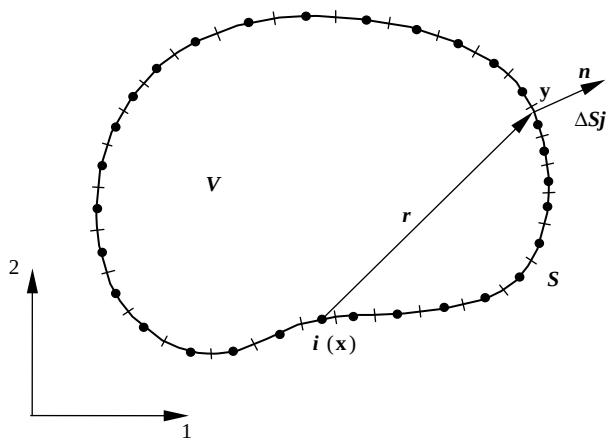

Fig. 2. Discretization of the boundary S using constant elements.

where ϕ_j and q_j ($j=1,2,\ldots,N$) are the nodal values of ϕ and q on element ΔS_j (Fig. 2), respectively, and the coefficients are given by

$$g_{ij} = \int_{\Delta S_j} G(\mathbf{x},\mathbf{y})\mathrm{d}S(\mathbf{y}), \qquad f_{ij} = \int_{\Delta S_j} F(\mathbf{x},\mathbf{y})\mathrm{d}S(\mathbf{y}), \tag{11}$$

for $i,j = 1,2,\ldots,N$

with the collocation point $\mathbf{x}$ being placed at node $\boldsymbol{i}$. In matrix form, Eq. (10) is written as

$$\frac{1}{2}\begin{Bmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_N\end{Bmatrix} + \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N}\\ f_{21} & f_{22} & \cdots & f_{2N}\\ \vdots & \vdots & \ddots & \vdots\\ f_{N1} & f_{N2} & \cdots & f_{NN}\end{bmatrix}\begin{Bmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_N\end{Bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N}\\ g_{21} & g_{22} & \cdots & g_{2N}\\ \vdots & \vdots & \ddots & \vdots\\ g_{N1} & g_{N2} & \cdots & g_{NN}\end{bmatrix}\begin{Bmatrix}q_1\\ q_2\\ \vdots\\ q_N\end{Bmatrix}. \tag{12}$$

In the conventional BEM approach, a standard linear system of equations is formed as follows by applying the boundary condition (either Eq. (2) or (3)) at each node and switching the columns in the two matrices in Eq. (12)

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N}\\ a_{21} & a_{22} & \cdots & a_{2N}\\ \vdots & \vdots & \ddots & \vdots\\ a_{N1} & a_{N2} & \cdots & a_{NN}\end{bmatrix}\begin{Bmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_N\end{Bmatrix} = \begin{Bmatrix}b_1\\ b_2\\ \vdots\\ b_N\end{Bmatrix}, \quad \text{or} \tag{13}$$

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{b},$$

where $\mathbf{A}$ is the coefficient matrix, $\boldsymbol{\lambda}$ the unknown vector and $\mathbf{b}$ the known right-hand side vector. Obviously, the construction of matrix $\mathbf{A}$ requires $O(N^2)$ operations using the two expressions in Eq. (11) and the size of the required memory for storing $\mathbf{A}$ is also $O(N^2)$ since $\mathbf{A}$ is in general a non-symmetric and dense matrix. The solution of system in Eq. (13) using direct solvers such as Gauss elimination is even worse, requiring $O(N^3)$ operations because of this general matrix. That is why the conventional BEM approach for solving the BIE's is in general slow and inefficient for large-scale problems, despite its robustness in the meshing stage as compared with other domain based methods.

3. The fast multipole method—formulation

The fast multipole method can be employed to accelerate the BEM for solving Eq. (7) or (9). The main idea of the FMM is to translate the node-to-node (or element-to-element) interactions to cell-to-cell interactions, where cells can have a hierarchical (tree) structure with the smallest cells (leaves) containing a specified number of elements. Iterative equation solvers (such as GMRES) are used in the FMM, where matrix–vector multiplications (as for the left-hand side of Eq. (13)) are calculated using fast multipole expansions. Using the FMM for the BEM, the solution time of a problem can be reduced to order $O(N)$. The memory requirement can also be reduced to $O(N)$ since iterative solvers do not need to store the entire matrix in the memory.

In this section, we first discuss the expansions that are used in the fast multipole method for 2D potential problems. The results of these expansions are scattered in the literature (see, e.g. Ref. [12]) and are summarized herein for completeness. Then, we describe the main procedures and algorithms in the fast multipole BEM.

Consider first the following integral with the G kernel in BIE (7) or (9)

$$\int_{S_0} G(\mathbf{x},\mathbf{y})q(\mathbf{y})\mathrm{d}S(\mathbf{y}), \tag{14}$$

in which S_0 is a subset of S and away from the collocation point $\mathbf{x}$ (Fig. 3). For convenience, we introduce the *complex notation*, that is, replace the collocation point $\mathbf{x}$ and field point $\mathbf{y}$ by $z_0 = x_1 + \mathrm{i}x_2$ and $z = y_1 + \mathrm{i}y_2$ (with $\mathrm{i} = \sqrt{-1}$ here), respectively (Fig. 3). We write

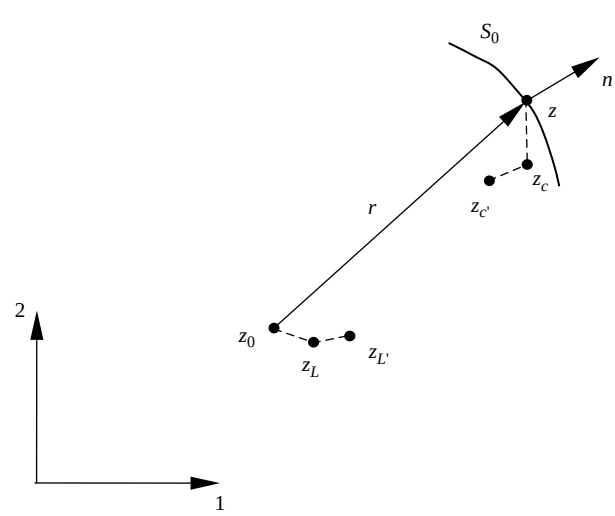

Fig. 3. Complex notation and the related points for fast multipole expansions.

$$G(\mathbf{x},\mathbf{y}) = \mathrm{Re}\{G(z_0,z)\}, \tag{15}$$

where

$$G(z_0,z) = -\frac{1}{2\pi}\ln(z_0-z). \tag{16}$$

Thus, the integral in (14) is equivalent to the real part of the following integral

$$\int_{S_0} G(z_0,z)q(z)\mathrm{d}S(z). \tag{17}$$

We now introduce several important concepts in the fast multipole method that are the building blocks for the FMM–BEM.

3.1. Multipole expansion (moments)

Assuming z_c is a point close to the field point z (Fig. 3), that is, $|z-z_c| \ll |z_0-z_c|$, we write

$$G(z_0,z) = -\frac{1}{2\pi}\ln(z_0-z) = -\frac{1}{2\pi}\left[\ln(z_0-z_c) + \ln\left(1-\frac{z-z_c}{z_0-z_c}\right)\right].$$

Applying the following Taylor series expansion

$$\ln(1-\xi) = -\sum_{k=1}^{\infty}\frac{\xi^k}{k}, \quad \text{for } |\xi|<1,$$

we obtain

$$G(z_0,z) = \frac{1}{2\pi}\sum_{k=0}^{\infty} O_k(z_0-z_c)I_k(z-z_c), \tag{18}$$

where, for convenience, we have introduced

$$I_k(z) = \frac{z^k}{k!}, \quad \text{for } k\geq 0; \qquad O_k(z) = \frac{(k-1)!}{z^k}, \quad \text{for } k\geq 1;$$

and $\quad O_0(z) = -\ln(z)$.

The derivatives of functions $I_k(z)$ and $O_k(z)$ satisfy:

$$I_k'(z) = I_{k-1}(z), \quad \text{for } k\geq 1; \quad \text{and} \quad O_k'(z) = -O_{k+1}(z),$$

for $k\geq 0$.

Note that in the G kernel given in Eq. (18), z_0 and z are separated now due to the introduction of the 'mid point' z_c, which is a key in the FMM. The integral in (17) is now evaluated as follows

$$\int_{S_0} G(z_0,z)q(z)\mathrm{d}S(z) = \frac{1}{2\pi}\int_{S_0}\left[\sum_{k=0}^{\infty} O_k(z_0-z_c)I_k(z-z_c)\right]q(z)\mathrm{d}S(z)$$

that is, the *multipole expansion*

$$\int_{S_0} G(z_0,z)q(z)\mathrm{d}S(z) = \frac{1}{2\pi}\sum_{k=0}^{\infty} O_k(z_0-z_c)M_k(z_c), \tag{19}$$

where

$$M_k(z_c) = \int_{S_0} I_k(z-z_c)q(z)\mathrm{d}S(z), \tag{20}$$

are called *moments* about z_c, which are independent of the collocation point z_0 and only need to be computed once. After these moments are obtained, the integral can be evaluated readily using Eq. (19) for any collocation point z_0 away from S_0 (which will be within a *cell* centered at z_c).

Detailed analysis of the errors in these multipole expansions and the subsequent translations can be found in Ref. [12].

3.2. Moment-to-moment translation (M2M)

If the point z_c is moved to a new location $z_{c'}$ (Fig. 3), the following translation holds for the moments

$$M_k(z_{c'}) = \int_{S_0} I_k(z-z_{c'})q(z)\mathrm{d}S(z) = \int_{S_0} I_k((z-z_c)+(z_c-z_{c'}))q(z)\mathrm{d}S(z).$$

Applying the binomial formula

$$(a+b)^n = \sum_{m=0}^{n}\binom{n}{m}a^m b^{n-m},$$

where $\displaystyle\binom{n}{m} = \frac{n!}{(n-m)!m!} = \binom{n}{n-m}$,

we obtain

$$M_k(z_{c'}) = \sum_{l=0}^{k} I_{k-l}(z_c-z_{c'})M_l(z_c). \tag{21}$$

This is the *M2M translation* for the moments when z_c is moved to $z_{c'}$. Note that there are only a finite number of terms in this translation.

3.3. Local expansion and moment-to-local translation (M2L)

Suppose z_L is a point close to the collocation point z_0 (Fig. 3), that is, $|z_0-z_L| \ll |z_c-z_L|$. From the multipole expansion (Eq. (19)), we have:

$$f(z_0) \equiv \int_{S_0} G(z_0,z)q(z)\mathrm{d}S(z) = \frac{1}{2\pi}\sum_{k=0}^{\infty} O_k(z_0-z_c)M_k(z_c). \tag{22}$$

We now expand $f(z_0)$ about point z_L, using the Taylor series expansion, that is

$$f(z_0) = \sum_{l=0}^{\infty} f^{(l)}(z_L) I_l(z_0 - z_L). \tag{23}$$

We evaluate

$$f^{(l)}(z_L) = \frac{(-1)^l}{2\pi} \sum_{k=0}^{\infty} O_{l+k}(z_L - z_c) M_k(z_c).$$

Substituting this result into (23), we obtain the following *local expansion*

$$\int_{S_0} G(z_0, z) q(z) \mathrm{d}S(z) = \sum_{l=0}^{\infty} L_l(z_L) I_l(z_0 - z_L), \tag{24}$$

where the coefficients are given by the following *M2L translation*

$$L_l(z_L) = \frac{(-1)^l}{2\pi} \sum_{k=0}^{\infty} O_{l+k}(z_L - z_c) M_k(z_c). \tag{25}$$

3.4. Local-to-local translation (L2L)

If the point for local expansion is moved from z_L to $z_{L'}$ (Fig. 3), we have the following expression using a local expansion from Eq. (24)

$$\int_{S_0} G(z_0,z) q(z) \mathrm{d}S(z) = \sum_{l=0}^{\infty} L_l(z_L) I_l(z_0 - z_L)$$

$$= \sum_{l=0}^{\infty} L_l(z_L) I_l((z_0 - z_{L'}) + (z_{L'} - z_L)).$$

Applying the formula $(a+b)^l = \sum_{m=0}^{l} \binom{l}{m} a^m b^{l-m}$, and the relation $\sum_{l=0}^{\infty} \sum_{m=0}^{l} = \sum_{m=0}^{\infty} \sum_{l=m}^{\infty}$, we obtain

$$\int_{S_0} G(z_0,z) q(z) \mathrm{d}S(z) = \sum_{l=0}^{\infty} L_l(z_{L'}) I_l(z_0 - z_{L'}), \tag{26}$$

where the new coefficients are given by the following *L2L translation*

$$L_l(z_{L'}) = \sum_{m=l}^{\infty} I_{m-l}(z_{L'} - z_L) L_m(z_L). \tag{27}$$

Replacing $m-1$ by k, we can also write (27) in an alternative form

$$L_l(z_{L'}) = \sum_{k=0}^{\infty} I_k(z_{L'} - z_L) L_{l+k}(z_L). \tag{28}$$

3.5. Expansions for integrals with the F kernel

We now consider the integral with the F kernel in BIE (7) or (9) in the complex notation

$$\int_{S_0} F(z_0, z) \phi(z) \mathrm{d}S(z), \tag{29}$$

where

$$F(z_0, z) = \frac{\partial G}{\partial n} = (n_1 + \mathrm{i} n_2) G' = n(z) G', \quad \text{with } G' \equiv \frac{\partial G}{\partial z}. \tag{30}$$

Thus,

$$F(\mathbf{x}, \mathbf{y}) = \mathrm{Re}\{F(z_0, z)\} = n_1 \,\mathrm{Re}\, G' - n_2 \,\mathrm{Im}\, G' \tag{31}$$

We have (Eq. (18))

$$G(z_0, z) = \frac{1}{2\pi} \sum_{k=0}^{\infty} O_k(z_0 - z_c) I_k(z - z_c).$$

Therefore

$$G' = \frac{1}{2\pi} \sum_{k=1}^{\infty} O_k(z_0 - z_c) I_{k-1}(z - z_c), \tag{32}$$

and the integral in (29) becomes

$$\int_{S_0} F(z_0, z) \phi(z) \mathrm{d}S(z) = \frac{1}{2\pi} \sum_{k=1}^{\infty} O_k(z_0 - z_c) N_k(z_c),$$

where

$$N_k(z_c) = \int_{S_0} n(z) I_{k-1}(z - z_c) \phi(z) \mathrm{d}S(z). \tag{33}$$

These are the *moments* for the F kernel integral, which are similar to those in Eq. (20) for the G kernel integral. It can also be shown that the M2M, M2L and L2L translations remain the same for the F kernel integral, except for the fact that $N_0 = 0$. That is, the translations used for M_k can be applied directly for N_k.

3.6. FMM algorithms and procedures

The algorithms and main procedures in the FMM–BEM are as follows:

Step 1. Discretization. For a given problem, discretize the boundary S as usual as in the conventional BEM (e.g. using constant elements as shown in Fig. 2).

Step 2. Determine the tree structure of the mesh. For a 2D problem, for example, consider a square that covers the entire boundary S and call this square the cell of level 0 (Fig. 4). Then, start dividing this *parent* cell into four equal *child* cells of level 1. Continue dividing in this way, that is, take a parent cell of level l and divide it into four child cells of level $l+1$. Stop dividing a cell if the number of elements in that cell is less than a specified number (this number is 1 in the example shown in Fig. 4). A cell having no child cells is called a *leaf* (e.g. the shaded cells in Fig. 4). Note that the edge length of a child cell is one-half of that of its parent cell. An element is considered within a cell if the center of the element is inside that cell. A quad-tree structure of the cells covering all the elements is thus formed using this procedure (Fig. 5).

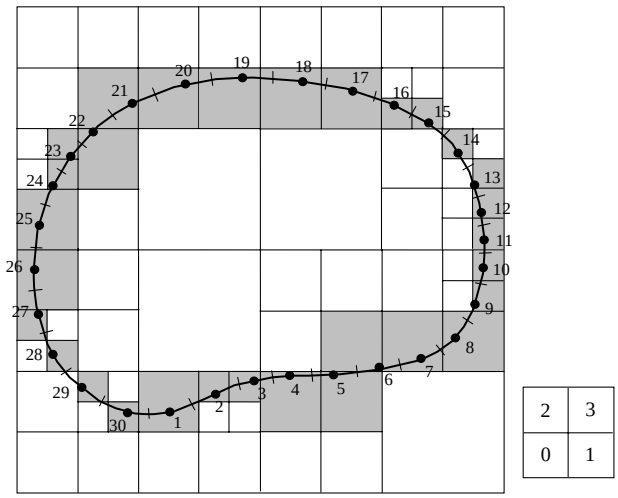

Fig. 4. A hierarchical cell structure covering all the boundary elements (the small square on the side shows the numbering scheme for the child cells of any given parent cell).

Step 3. Upward pass. Compute the moments on all cells, at all levels with $l \geq 2$, and for up to p terms, tracing the tree structure upward. For a leaf, Eq. (20) is applied directly (with S_0 being the elements contained in the leaf and z_c the centroid of the leaf). For a parent cell, the moment is calculated by summing the moments on its four child cells using the M2M translation, that is, Eq. (21), in which $z_{c'}$ is the centroid of the parent cell and z_c the centroid of a child cell (Fig. 6).

Step 4. Downward pass. To describe the downward pass, we first define a few terms. Two cells are said to be *adjacent cells at level l* if they have at least one common vertex. Two cells are said to be *well separated at level l* if they are not adjacent at level l but their parent cells are adjacent at level $l-1$. The list of all the well-separated cells from a level l cell C is called the *interaction list of C*. Cells are called to be *far cells of C* if their parent cells are not adjacent to the parent cell of C (Fig. 7). We now compute the local expansion (coefficients) on all cells starting from level 2 and tracing the tree structure downward to all the leaves. The local expansion associated with a cell C is the sum of the contributions from the cells in the interaction list of cell C and from all the far cells. The former is calculated by using the M2L translation, Eq. (25), with moments associated with cells in the interaction list, and the latter is calculated by using the L2L translation, Eq. (27) or (28), for the parent cell of C with the expansion point being shifted from the centroid of C's parent cell to that of C. For a cell C at level 2, we use only M2L translation to compute the coefficients of the local expansion.

Step 5. Evaluation of integrals in BIE (7). We use the first integral on the right-hand side of BIE (7) as an example. Suppose the density function is given and the collocation point z_0 is on an element in leaf C (Fig. 7). We compute the contributions from elements in leaf C and its adjacent cells directly as in the conventional BEM. Contributions from all other cells (cells in the interaction list of C and far cells) are computed by using the local expansion, that is, Eq. (24). This is done by using the local expansion coefficients for cell C, which have been computed in Step 4, and shifting the expansion point from the centroid of C to the collocation point z_0 (Fig. 6). The integral with the F kernel can be obtained in a similar way, assuming the density function is given.

Step 6. Iterations of the solution. We update the unknown vector in the system $\mathbf{A\lambda}=\mathbf{b}$ corresponding to BIE (7), and continue at Step 3 for the matrix and unknown vector multiplication until the solution converges within a given tolerance.

Utilizing a good pre-conditioner for the FMM–BEM is crucial for its convergence and computing efficiency. In this study, we use a block diagonal pre-conditioner which is formed on each leaf using direct evaluations of the fundamental solution on the elements within that leaf. This pre-conditioner may not be efficient if the number of elements in a leaf is large (for example, more than a few thousands). Using other forms of the pre-conditioners is possible and will be an important subject in improving the efficiency of the FMM–BEM in other applications. When the problem size is large, the estimated cost of the entire process described above is $O(N)$ with N being the

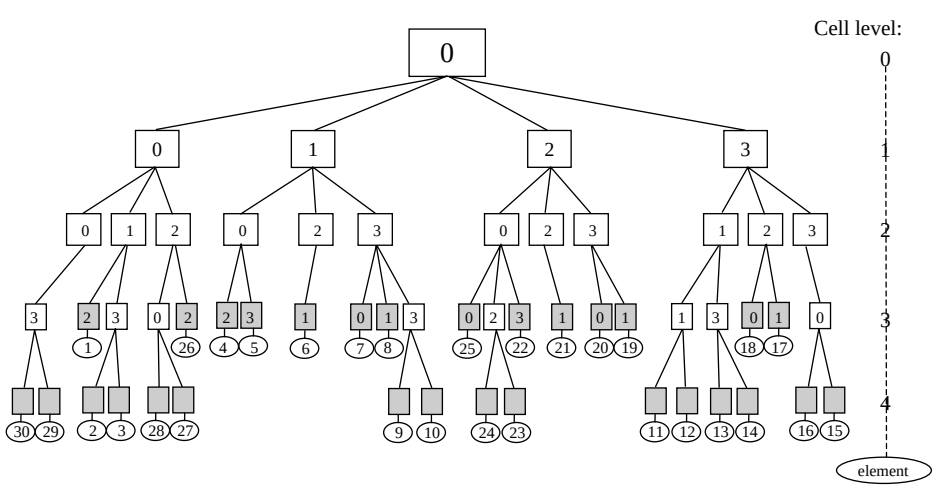

Fig. 5. The quad-tree structure.

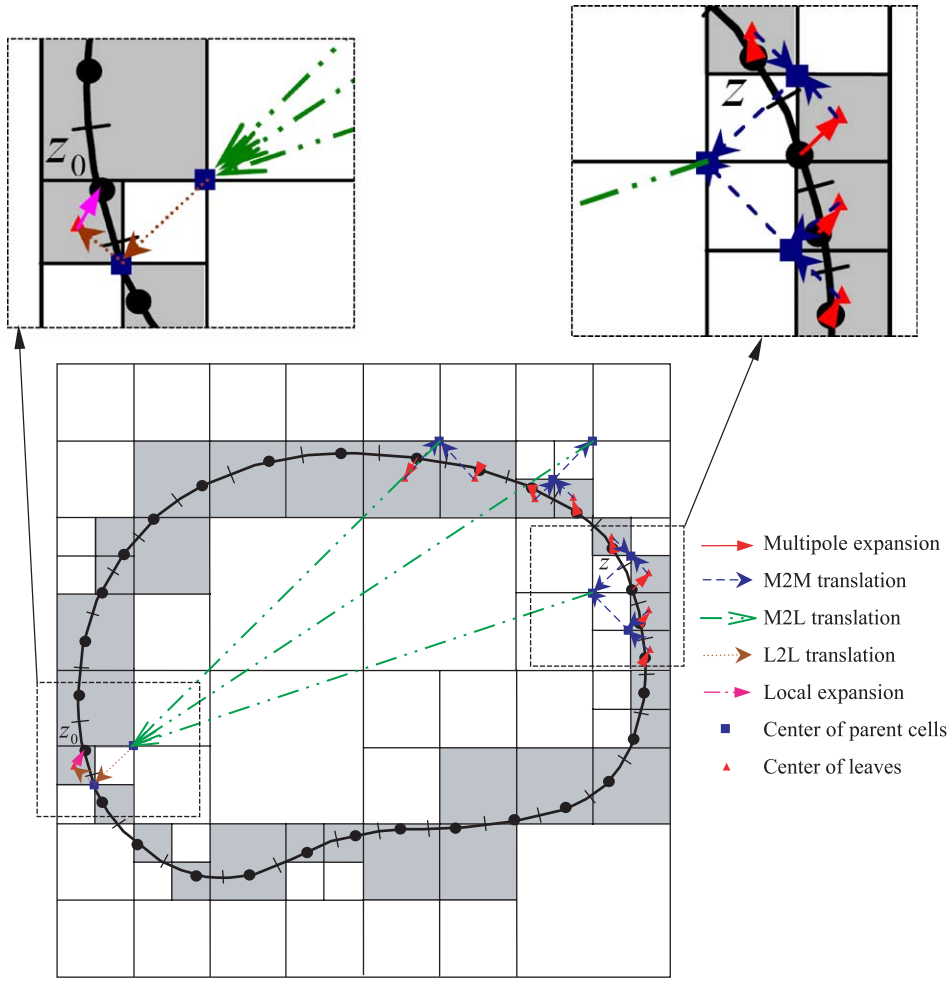

Fig. 6. Upward and downward passes using expansions and translations in the FMM.

number of elements or nodes, if the number of terms p in the multipole expansions and the number of elements in a leaf are kept constant (see Ref. [21]).

4. The fast multipole method—programming

We now discuss the main structure of an FMM–BEM code for solving general 2D potential problems. This Fortran code is based on an earlier code that was used for 2D crack and rigid-line inclusion problems governed by Laplace equation and formulated using a hypersingular BIE [18]. This general 2D FMM–BEM code for potential problems can be expanded to solve 3D potential problems, 2D and 3D elasticity problems, using constant or higher-order elements.

The flowchart of the 2D FMM–BEM code is given in Fig. 8. The chart shows the main tasks for the program and the related subroutines. The source code for the iterative solver GMRES can be downloaded from the website *netlib* (http://www.netlib.org/). The program starts with reading in the data for the boundary nodes and elements, and the parameters used in the fast multipole expansions and solver GMRES. Then, the quad-tree structure is created and the right-hand side vector is computed using the FMM algorithms (this need to be done only once). The main part of the program is in two subroutines prepared for GMRES, *msolve* which prepares a pre-conditioning matrix for the iterative solver, and *matvec* which provides the algorithm for the matrix-vector multiplication ($\mathbf{A}\boldsymbol{\lambda}$ in Eq. (13)) using the fast multipole algorithms (by calling the *upward* and *dwnwrd* subroutines).

The source code, input files (one for element data and one for parameters) and an output file are available from the authors [24]. The main variables in the subroutines are explained in the source code, and the format of the input and output files are self-explaining in these files.

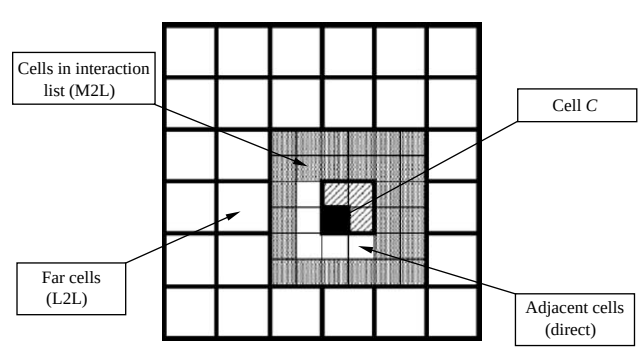

Fig. 7. Grouping of the cells for cell C (black one) at level l.

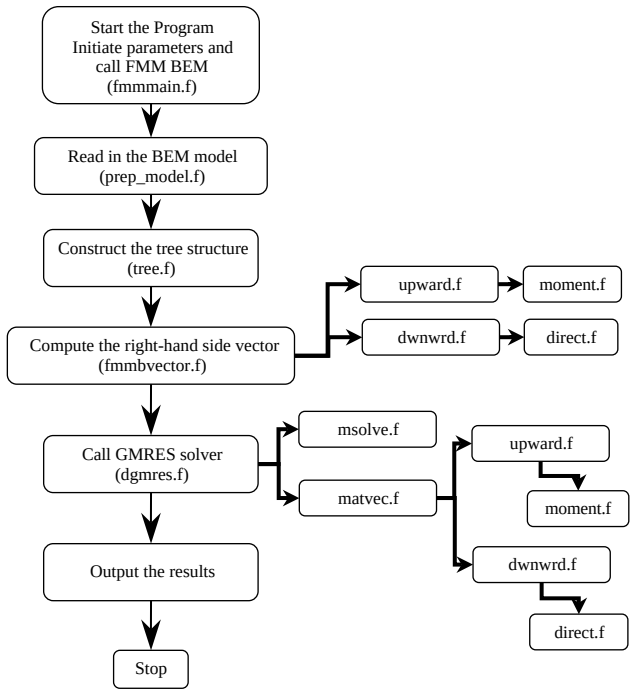

Fig. 8. Flow chart for the FMM–BEM code.

5. Numerical examples

We present two numerical examples to demonstrate the accuracy and efficiency of the FMM–BEM for 2D potential problems. All the computations were done on a Pentium IV laptop PC with a 2.4 GHz CPU, 1 GB RAM and 40 GB hard drive. The largest model with 1.4 million equations run for less than 11,000 CPU seconds on this laptop.

5.1. An annular region

We first consider a simple potential (e.g. heat conduction) problem in an annular region as shown in Fig. 9, for which the available analytical solution can be used to check the BEM results. The field ϕ is given on the inner boundary S_a, while the normal derivative is given on the outer boundary S_b. The analytical solution for this (axisymmetric) problem is given by

$$\phi(r) = \phi_a + q_b b \ln\left(\frac{r}{a}\right),$$

where ϕ_a and q_b are the given values of ϕ and q on the boundary S_a and S_b, respectively, and r is the radial coordinate in a polar-coordinate system centered at O. This gives

$$\phi_b = \phi(b) = \phi_a + q_b b \ln\left(\frac{b}{a}\right), \qquad q_a = \frac{\partial\phi}{\partial n}(a) = -q_b\frac{b}{a}.$$

For the test problem, we choose $a=1$, $b=2$, $\phi_a=100$, and $q_b=200$. This gives

$$\phi_b = 377.258872, \qquad q_a = -400.0.$$

We discretize the inner and outer boundaries with the same number of elements and run both the FMM–BEM code and a conventional BEM code using also constant elements. The conventional BEM code uses a direct solver (Gauss elimination) for solving the linear system. For the FMM–BEM, the numbers of terms for both moments and local expansions were set to 15, the maximum number of elements in a leaf to 20, and the tolerance for convergence of the solution to 10^{-8}. The FMM–BEM results converged in 11 iterations for the smallest model (with 36 elements) and in 43 iterations for the largest model (with 9600 elements). These numbers can be reduced to 9 and 28 iterations, respectively, if the tolerance is reduced to 10^{-6}.

Table 1 shows the results of ϕ_b and q_a for this problem using both the FMM–BEM and the conventional BEM as the total number of elements increase from 36 to 9600. As we can see, the results for both FMM–BEM and conventional BEM converge quickly to the exact solution for the mesh with only 72 constant elements with a relative error of 0.1%. The results continue to improve until the mesh with 4800 elements. For the two larger meshes (with 7200 and 9600 elements), the results for q_b deviate slightly from the exact solution for both FMM–BEM and conventional BEM, which may be caused by truncation errors. In general, the FMM–BEM is found to be

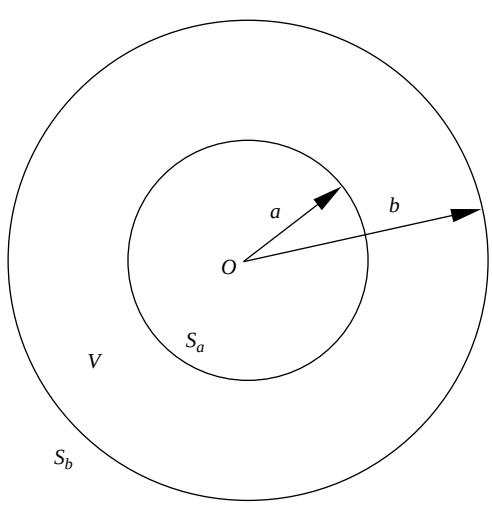

Fig. 9. A simple potential problem in an annular region V.

Table 1
Results of the potential and normal derivative for the annular region

N	q_a		ϕ_b	
	FMM–BEM	Conventional BEM	FMM–BEM	Conventional BEM
36	−401.771619	−401.771546	376.723694	376.723612
72	−400.400634	−400.400662	377.140967	377.140972
360	−400.014881	−400.014803	377.254774	377.254783
720	−400.003468	−400.003629	377.257857	377.257873
1440	−400.000695	−400.000533	377.258607	377.258622
2400	−400.001929	−400.000612	377.258795	377.258783
4800	−400.001557	−400.000561	377.258852	377.258856
7200	−399.997329	−399.998183	377.258848	377.258857
9600	−399.997657	−399.996874	377.258859	377.258871
Analytical solution	−400.0		377.258872	

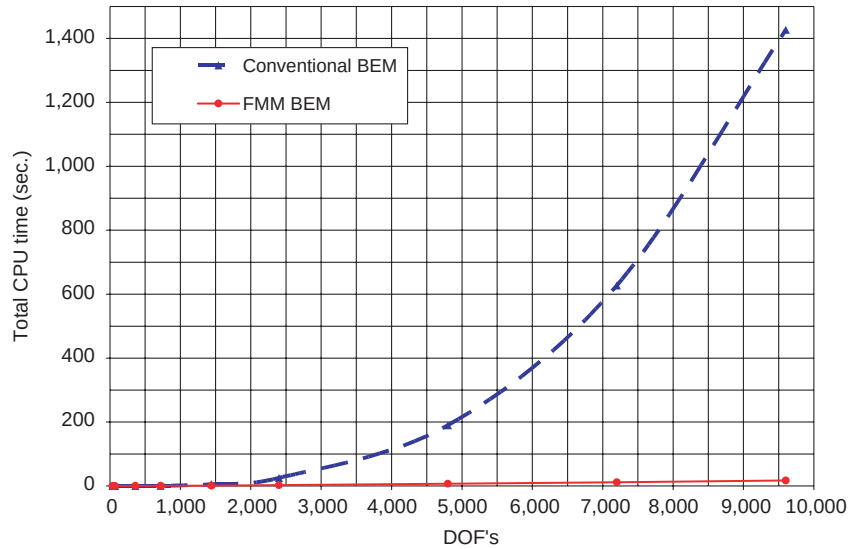

Fig. 10. Comparison of the CPU time used by the conventional BEM and FMM–BEM.

equally accurate as the conventional BEM with moderate values for the parameters in the FMM–BEM. The CPU times used for both approaches in these calculations are plotted in Fig. 10, which shows significant advantage of the FMM–BEM in the savings compared with the conventional BEM. For example, for the largest model with 9600 elements, the FMM–BEM used less than 17 s, while the conventional BEM used about 7500 s CPU time.

5.2. Square plates with many circular holes

We next study square plates with many randomly distributed circular holes, or perforated plates (Fig. 11). A total of $m \times m$ holes are placed randomly in a square plate of dimension 1×1, where $m = 20$, 30, 40,…,140, and 200. The radii of the holes are the same in each case and are determined in such a way that the total 'volume' fractions of the holes are kept to be the same for all the cases (equal to 12.47%). Each hole is discretized with 36 elements and the outer boundary of the plate is discretized with 400 elements. Thus, the largest model with 40,000 holes has the total DOF's equal to 1,440,400. For boundary conditions, $\phi = 0$ and 1 are imposed on the edges at $x = 0$ and 1, respectively, while $q = 0$ is imposed on both edges at $y = 0$ and 1, and on edges of all the holes.

For this example, the numbers of terms for both expansions were set to 10, the maximum number of elements in a leaf to 100, and the tolerance for convergence of the solution to 10^{-6}. Fig. 12 shows the CPU time for the FMM–BEM in solving these relatively large models. It is seen that the total CPU time increases almost linearly with the increase of the total DOF's (as indicated by the trendline), which clearly demonstrates the $O(N)$ efficiency of the FMM–BEM in solving large-scale problems. The largest model with 1,440,400 DOF's took less than 11,000 s on the Pentium IV laptop PC.

6. Discussions

A tutorial on the fast multipole boundary element method is presented in this paper, using the 2D potential problem as an example. Complete formulations and implementation details for the FMM–BEM are provided. A computer code in Fortran is also given that can be used to learn the structure of an FMM–BEM code and to expand to solve other problems. Numerical examples are presented to further demonstrate the efficiencies of the FMM–BEM for solving large-scale problems. The approach and code presented in this paper can be extended readily to 2D vector (elastostatic) problems, as well as 3D scalar and vector problems. Readers can refer to Ref. [21] for further information on research papers in these areas.

For 3D problems, the quad-tree structure should be replaced by an oct-tree structure for the cells covering all the boundary elements. Although Taylor series expansions have been used for the fundamental solutions in 3D problems, they have been found to be inefficient since that require large numbers of terms in the expansions for convergence. This difficulty lies in the fact that for 2D problems using the complex variables, each term in the Taylor series expansion is a harmonic function. Since the fundamental solution is also a harmonic function in nature, the Taylor series expansion in complex variable is a natural choice, as it was used in the earlier works by Rokhlin and Greengard [10–12]. However, this is not the case for 3D Taylor series expansion with real variables. A better choice in 3D should be to employ the solid harmonic functions for expansions of the fundamental solutions (see, e.g. Refs. [25,26]), which can be considered as a counter-part in 3D of the 2D case using complex variables. Expansions with about 20 terms are often found to be sufficient for the convergence with a tolerance of 10^{-6} in the solutions for 3D problems using expansions with solid harmonic functions [19].

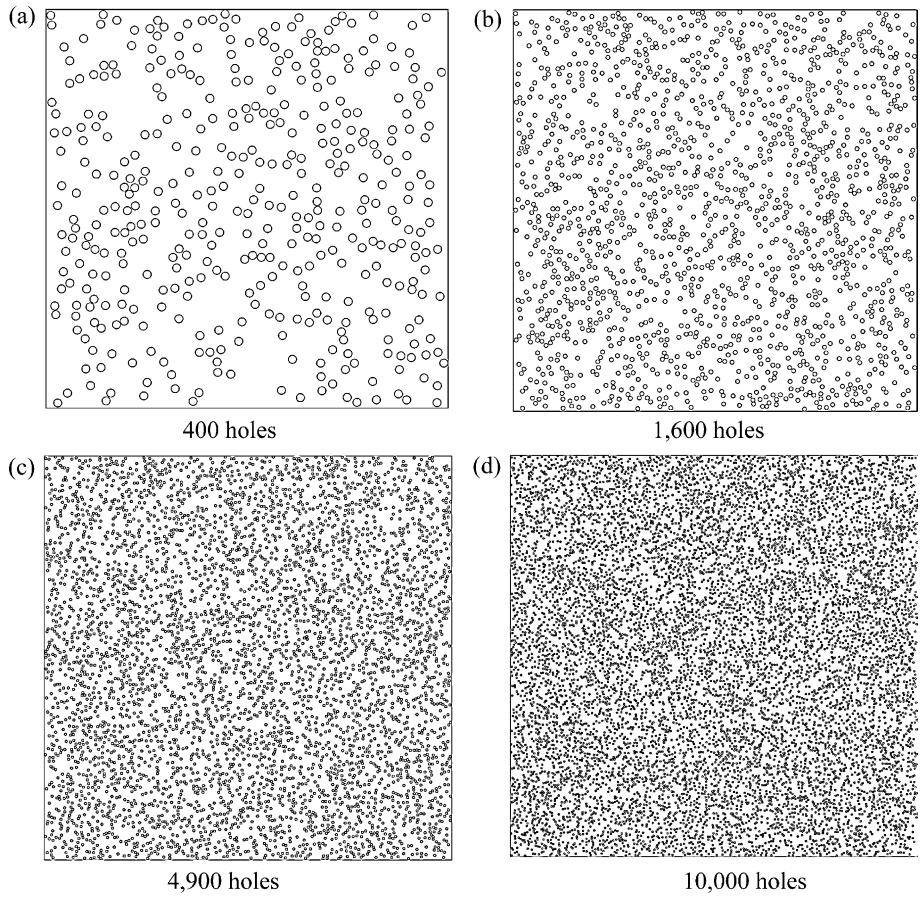

Fig. 11. Square plates with large numbers of randomly distributed circular holes.

Constant elements can certainly be replaced by higher-order elements to improve the accuracy of an FMM–BEM code. However, this may not be advantageous considering the efficiency of the code for large-scale problems. For constant elements, all the integrals are evaluated analytically, for all non-singular, nearly-singular and singular cases. There are no numerical integrations at all in the code. For higher-order elements, however, this may no longer be the case and one will have to introduce numerical integrations for the direct evaluations of the integrals that can involve singular and

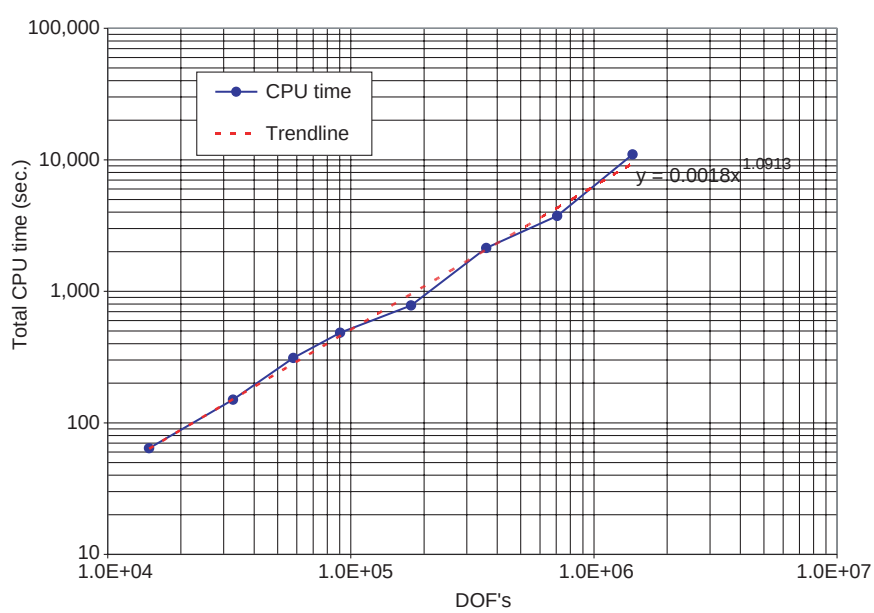

Fig. 12. Total CPU time used by the FMM–BEM for solving the plate-hole problems (log–log scale).

nearly-singular cases. This may complicate the code and reduce the efficiency in FMM–BEM solutions for large-scale problems.

More research need to be done to improve the FMM–BEM, regarding, for example, convergence of the solvers, optimization of the tree structures, and for dynamic and non-linear problems. Wide spread applications of the FMM–BEM for solving large-scale engineering problems may not be far away.

Acknowledgements

The first author (Y.J.L.) would like to thank the support by the fellowship of the Japan Society for the Promotion of Science (JSPS), the Academic Center for Computing and Media Studies of the Kyoto University, the Chunhui Fellowship of the Ministry of Education of China, and the Department of Engineering Mechanics of the Tsinghua University.

References

[1] Jaswon MA. Integral equation methods in potential theory I. Proc R Soc Lond A 1963;275:23–32.
[2] Rizzo FJ. An integral equation approach to boundary value problems of classical elastostatics. Q Appl Math 1967;25:83–95.
[3] Brebbia CA. The boundary element method for engineers. London: Pentech Press; 1978.
[4] Banerjee PK et al, editor. Developments in boundary element methods, vols. I–VII. London: Elsevier; 1979–1991.
[5] Mukherjee S. Boundary element methods in creep and fracture. New York: Applied Science Publishers; 1982.
[6] Cruse TA. Boundary element analysis in computational fracture mechanics. Dordrecht, The Netherlands: Kluwer; 1988.
[7] Brebbia CA, Dominguez J. Boundary elements—an introductory course. New York: McGraw-Hill; 1989.
[8] Banerjee PK. The boundary element methods in engineering. 2nd ed. New York: McGraw-Hill; 1994.
[9] Kane JH. Boundary element analysis in engineering continuum mechanics. Englewood Cliffs, NJ: Prentice-Hall; 1994.
[10] Rokhlin V. Rapid solution of integral equations of classical potential theory. J Comput Phys 1985;60:187–207.
[11] Greengard LF, Rokhlin V. A fast algorithm for particle simulations. J Comput Phys 1987;73(2):325–48.
[12] Greengard LF. The rapid evaluation of potential fields in particle systems. Cambridge: The MIT Press; 1988.
[13] Peirce AP, Napier JAL. A spectral multipole method for efficient solution of large-scale boundary element models in elastostatics. Int J Numer Methods Eng 1995;38:4009–34.
[14] Gomez JE, Power H. A multipole direct and indirect BEM for 2D cavity flow at low Reynolds number. Eng Anal Bound Elem 1997;19:17–31.
[15] Fu Y, Klimkowski KJ, Rodin GJ, Berger E, Browne JC, Singer JK, et al. A fast solution method for three-dimensional many-particle problems of linear elasticity. Int J Numer Methods Eng 1998;42:1215–29.
[16] Nishimura N, Yoshida K, Kobayashi S. A fast multipole boundary integral equation method for crack problems in 3D. Eng Anal Bound Elem 1999;23:97–105.
[17] Mammoli AA, Ingber MS. Stokes flow around cylinders in a bounded two-dimensional domain using multipole-accelerated boundary element methods. Int J Numer Meth Eng 1999;44:897–917.
[18] Nishimura N, Liu YJ. Thermal analysis of carbon-nanotube composites using a rigid-line inclusion model by the boundary integral equation method. Comput Mech 2004;35(1):1–10.
[19] Liu YJ, Nishimura N, Otani Y, Takahashi T, Chen XL, Munakata H. A fast boundary element method for the analysis of fiber-reinforced composites based on a rigid-inclusion model. J Appl Mech 2005;72(1): 115–28.
[20] Chew WC, Chao HY, Cui TJ, Lu CC, Ohnuki S, Pan YC, et al. Fast integral equation solvers in computational electromagnetics of complex structures. Eng Anal Bound Elem 2003;27(8):803–23.
[21] Nishimura N. Fast multipole accelerated boundary integral equation methods. Appl Mech Rev 2002;55(4 (July)):299–324.
[22] Liu YJ, Rudolphi TJ. Some identities for fundamental solutions and their applications to weakly-singular boundary element formulations. Eng Anal Bound Elem 1991;8(6):301–11.
[23] Liu YJ, Rudolphi TJ. New identities for fundamental solutions and their applications to non-singular boundary element formulations. Comput Mech 1999;24(4):286–92.
[24] Nishimura, N. Liu, YJ. A fast multipole boundary element code for solving 2-D potential problems (downloadable at http://gspsun1.gee.kyoto-u.ac.jp/nchml/software for Unix/Linux platform or http://urbana.mie.uc.edu/yliu/Software/ for Windows platform).
[25] Yoshida K, Nishimura N, Kobayashi S. Application of new fast multipole boundary integral equation method to crack problems in 3D. Eng Anal Bound Elem 2001;25:239–47.
[26] Yoshida K, Nishimura N, Kobayashi S. Application of fast multipole Galerkin boundary integral equation method to crack problems in 3D. Int J Numer Methods Eng 2001;50:525–47.