A Fast Multipole Accelerated Singular Boundary Method for Potential Problems

W. Chen[1,2]**, C. J. Liu**[1] **and Y. Gu**[2,3]

Abstract: The singular boundary method (SBM) is a recently-developed mesh-less boundary collocation method. This method overcomes the well-known fictitious boundary issue associated with the method of fundamental solutions (MFS) while remaining the merits of the later of being truly meshless, integral-free, and easy-to-program. Similar to the MFS, this method, however, produces dense and unsymmetrical coefficient matrix, which although much smaller in size compared with domain discretization methods, requires $O(N^2)$ operations in the iterative solution of the resulting algebraic system of equations. To remedy this bottleneck problem for its application to large-scale problems, this paper makes the first attempt to develop a fast multipole SBM (FM-SBM) formulation for two-dimensional (2D) potential problems. The proposed strategy can solve large-scale problems with several millions boundary discretization nodes on a desktop computer.

Keywords: Singular boundary method, method of fundamental solutions, fast multipole method, meshless boundary collocation method, potential problem.

1   Introduction

The singular boundary method (SBM) [Chen and Wang (2010)] is a recently-developed meshless boundary collocation method [Atluri et al. (2006); Dong and Atluri (2011), (2012); Zhang et al. (2002); Zhang et al. (2003)] and has since been successfully applied to a variety of physical problems, such as potential [Chen and Gu (2012); Gu and Chen (2013); Gu et al. (2012a)] and elastostatic problems [Gu et al. (2014); Gu et al. (2012b); Gu et al. (2011)]. This method inherits the merits of the method of fundamental solutions (MFS) [Chen (1995); Fairweather and

---

[1] State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering, International Center for Simulation Software in Engineering and Sciences, College of Mechanics and Materials, Hohai University, Nanjing 210098, China.

[2] Corresponding author. E-mail: chenwen@hhu.edu.cn; guyan1913@163.com

[3] College of Mathematics, Qingdao University, Qingdao 266071, PR China.

Karageorghis (1998); Sarler (2009); Sarler and Liu (2013); Young et al. (2007)] and is truly free of mesh and integration. The SBM avoids the perplexing fictitious boundary issue associated with the traditional MFS. However, the application of this method has so far been limited to model relatively small scale problems. This is mainly because the SBM produces dense interpolation matrix that, although smaller in size compared with the domain discretization methods such as the FEM, requires $O(N^2)$ operations in the iterative solution of its resulting algebraic equations. The SBM solution of large-scale problems is computationally expensive.

Inspired by the successful application of the fast multipole method (FMM) [Greengard and Rokhlin (1987); Rokhlin (1985)] in significantly reducing computing costs of the desnse matrix equations from boundary integraltion methods (BIEs) [Liu and Nishimura (2006); Liu et al. (2005); Nishimura (2002); Wang and Yao (2004)], this study makes the first attempt to apply the FMM to accelerating the solutions of the SBM discretization equations. Applying the FMM to the SBM can potentially reduce both the operation count and memory requirement to $O(N)$. This will be a significant improvement to the SBM in extending its applications.

This paper demonstrates that the proposed FM-SBM strategy is capable of solving large-scale problems. The FM-SBM formulations and its numerical implementation will be discussed in details. Numerical examples with up to 4,000,000 collocation points are successfully solved on a Core II PC using the developed FM-SBM code.

A brief outline of the rest of this paper is as follows: Section 2 reviews the conventional SBM formulations for potential problems. Section 3 provides detailed formulations and algorithms used in the FM-SBM, followed in Section 4 two benchmark numerical examples are well studied to illustrate the accuracy and efficiency of the proposed strategy. Finally, the conclusions and remarks are provided in Section 5.

2   Formulations of the conventional SBM

Without loss of generality, we consider the following Laplace equation governing potential problems in a 2D domain Ω (see Fig.1)

$$\nabla^2 u(\mathbf{x}) = 0, \mathbf{x} \in \Omega, \tag{1}$$

subject to the following boundary conditions

$$u(\mathbf{x}) = \bar{u}(\mathbf{x}), \mathbf{x} \in S_1 \text{ (Dirichlet boundary condition)}, \tag{2}$$

$$q(\mathbf{x}) = \frac{\partial u(\mathbf{x})}{\partial \mathbf{n}} = \bar{q}(\mathbf{x}), \mathbf{x} \in S_2 \text{ (Neumann boundary condition)}, \tag{3}$$
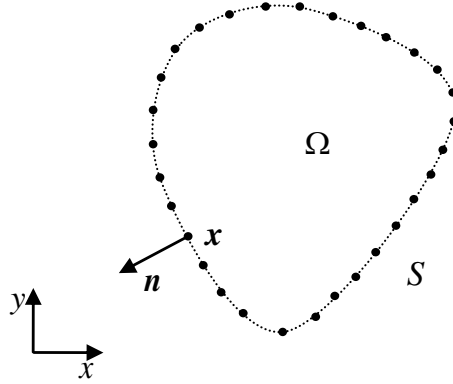
Figure 1: Domain Ω and its boundary S.

where u denotes the potential field, Ω is a bounded domain in $\mathbf{R}^2$ with boundary $S = S_1 + S_2$ which is assumed to be piecewise smooth, $\mathbf{n}$ presents the outward normal, and the barred quantities indicate given values along the boundary.

In the SBM, the collocation and source points are coincident and are both placed on the physical boundary, without requiring a fictitious boundary as compared with the traditional MFS. The SBM interpolation formula for 2D potential problems is given by [Chen and Gu (2012); Gu et al. (2012a)]

$$u(\mathbf{x}_i) = \sum_{j=1,i\neq j}^{N} \alpha_j G(\mathbf{x}_i, \mathbf{s}_j) + \alpha_i u_{ii}, \quad \mathbf{x}_i \in S_1, \ \mathbf{s}_j \in S, \tag{4}$$

$$q(\mathbf{x}_i) = \frac{\partial u(\mathbf{x}_i)}{\partial \mathbf{n}_{x_i}} = \sum_{j=1,i\neq j}^{N} \alpha_j A(\mathbf{x}_i, \mathbf{s}_j) + \alpha_i q_{ii}, \quad \mathbf{x}_i \in S_2, \ \mathbf{s}_j \in S, \tag{5}$$

where $\mathbf{x}_i$ is the ith collocation point, $\mathbf{s}_j$ the jth source point, α_j the jth unknown intensity of the distributed source at $\mathbf{s}_j$, N the number of the collocation points, and

$$G(\mathbf{x}_i, \mathbf{s}_j) = -\frac{1}{2\pi} \ln \|\mathbf{x}_i - \mathbf{s}_j\|_2, \tag{6}$$

$$A(\mathbf{x}_i, \mathbf{s}_j) = \frac{\partial G(\mathbf{x}_i, \mathbf{s}_j)}{\partial \mathbf{n}_{\mathbf{x}_i}}, \tag{7}$$

the fundamental solutions for 2D potential problems.

In the above Eqs. (4) and (5), u_{ii} and q_{ii} are defined as origin intensity factors, i.e., the diagonal elements of the SBM interpolation matrix. The fundamental assumption of the SBM is the existence of the origin intensity factor upon the singularity

of the coincident source-collocation point for mathematically well-posed problems. Our numerical experiments illustrate that the origin intensity factors do exist and have finite values, only depending on the distribution of boundary nodes and their respective boundary conditions. We refer readers to Refs. [Gu et al. (2014); Gu et al. (2011)] for further detailed derivation of the aforementioned origin intensity factors.

3   The fast multipole method for SBM

The key idea of the FMM is to convert the point-to-point interactions to cell-to-cell interactions, in which the procedure is efficiently carried out with tree structures. An iterative solver, such as GMRES, is employed to reduce the computational complexity of the matrix-vector multiplication from $O(N^2)$ to $O(N)$. Using the FMM, both the solution time and the memory requirements can be reduced to order of $O(N)$ [Rokhlin (1985)].

This section will present a FMM formulation for the SBM, which is similar to that for the BEM, except for the calculation of the kernel expansions where integrals are replaced by summations. For completeness, all the required formulas are provided. The detailed derivation of some of these formulas can be found in Refs. [Liu and Nishimura (2006); Liu et al. (2005)].

For convenience, complex notations are used to represent the collocation point **x** and the source point **s** by $z_0 = x_0 + iy_0$ and $z = x_1 + iy_1$, respectively (see Fig.2). The fundamental solution (6) can then be rewritten as

$$G(z_0, z) = -\frac{1}{2\pi} \ln(z_0 - z), \tag{8}$$

in which the real part gives the values of fundamental solution. And the kernels A and B shown blow related to the G kernel, can be rewritten as

$$A(z_0, z) = n(z_0)\frac{\partial G(z_0, z)}{\partial z_0}, \tag{9}$$

$$B(z_0, z) = n(z)\frac{\partial G(z_0, z)}{\partial z}, \tag{10}$$

with $n = n_1 + in_2$, in complex notation.

3.1   Multipole moments

Suppose that z_c is a point close to source point z, that is, inequality $|z - z_c| \leq |z_0 - z_c|$ is valid (see Fig. 2). Using the subtracting and adding-back technique,
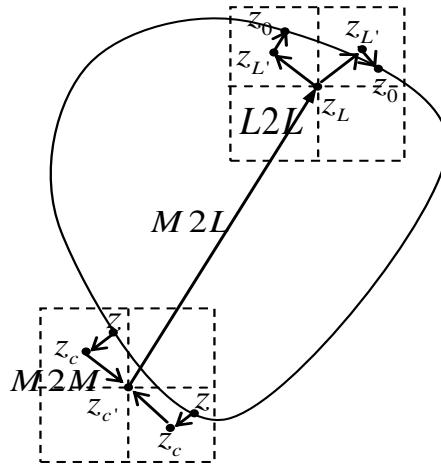
Figure 2: Main steps of FMM.

Eq. (8) can be expressed as

$$G(z_0, z) = -\frac{1}{2\pi} \ln(z_0 - z_c + z_c - z) = -\frac{1}{2\pi} \left[\ln(z_0 - z_c) + \ln\left(1 - \frac{z - z_c}{z_0 - z_c}\right) \right]. \quad (11)$$

Applying the Taylor series expansion to the second log term of Eq. (11), we get the following equation

$$G(z_0, z) = -\frac{1}{2\pi} \left[\ln(z_0 - z_c) - \sum_{k=1}^{\infty} \frac{1}{k} \frac{(z - z_c)^k}{(z_0 - z_c)^k} \right]. \quad (12)$$

Considering a group of source points, when the collocation point z_0 is sufficiently far away from this group, the sum $\sum_j G(z_0, z_j)\alpha_j$ on the right-hand side of Eq. (4) can be obtained by the following multipole expansion

$$\sum_j G(z_0, z_j)\alpha_j = -\frac{1}{2\pi} \left[\ln(z_0 - z_c)M_0 + \sum_{k=1}^{\infty} \frac{M_k(z_c)}{(z_0 - z_c)^k} \right], \quad (13)$$

where $M_k(z_c)$ are multiple moments centered at z_c defined by

$$M_k(z_c) = \begin{cases} \sum_j \alpha_j, & k = 0, \\ -\frac{1}{k}\sum_j (z_j - z_c)^k \alpha_j, & k \geq 1, \end{cases} \quad (14)$$

and only need to be computed once.

In addition, for the sum on the right-hand side of Eq. (5), we can also derive the following multipole expansion

$$\sum_j A(z_0, z_j)\alpha_j = -\frac{n(z_0)}{2\pi}\left[\frac{M_0}{z_0 - z} - \sum_{k=1}^{\infty}\frac{kM_k(z_c)}{(z_0 - z)^{k+1}}\right], \tag{15}$$

by using the same moments given in Eq. (14).

In order to calculate the diagonal coefficients of Neumann boundary Eq. (5), the sum $\sum_j l_j B(z_0, z_j)/l_i$ should be calculated. By using Eqs. (10) and (12), we can obtain the following multipole equation

$$\sum_j \frac{l_j}{l_i} B(z_0, z_j)\alpha_j = -\frac{1}{2\pi l_i}\sum_{k=1}^{\infty}\frac{M_k'(z_c)}{(z_0 - z_c)^k}, \tag{16}$$

where

$$M_k'(z_c) = \begin{cases} 0, & k=0, \\ -\sum_j l_j n(z_j)(z_j - z_c)^{k-1}\alpha_j, & k \geq 1, \end{cases} \tag{17}$$

are another group of moments about z_c.

3.2 M2M translation

Suppose that the point z_c is moved to a new location $z_{c'}$ (see Fig. 2), we obtain the following equations

$$M_k(z_{c'}) = -\frac{1}{k}\sum_j (z_j - z_{c'})^k \alpha_j = -\frac{1}{k}\sum_j [(z_j - z_c) + (z_c - z_{c'})]^k \alpha_j, \tag{18}$$

$$M_k'(z_{c'}) = -\sum_j l_j n(z_j)(z_j - z_{c'})^{k-1}\alpha_j = -\sum_j l_j n(z_j)[(z_j - z_c) + (z_c - z_{c'})]^{k-1}\alpha_j. \tag{19}$$

When applying the binomial formula and rearranging the terms to Eqs. (18) and (19), respectively, we can further get

$$M_k(z_{c'}) = \begin{cases} M_0(z_c), & k=0, \\ -\frac{1}{k}(z_c - z_{c'})^k M_0(z_c) + \sum_{l=1}^{k}\binom{k-1}{l-1} & \\ (z_c - z_{c'})^{k-1}M_l(z_c), & k \geq 1, \end{cases} \tag{20}$$

$$
M'_k(z_{c'}) = \begin{cases} 0, & k=0, \\ \sum\limits_{l=1}^{k} \begin{pmatrix} k-1 \\ l-1 \end{pmatrix} (z_c - z_{c'})^{k-1} M'_l(z_c), & k \geq 1. \end{cases} \tag{21}
$$

Note that Eqs. (20) and (21) translate the moments centered at z_c to the moments centered at $z_{c'}$, which are called M2M translation.

3.3   Local expansions and M2L translation

Suppose z_L is a point close to the collocation point z_0, that is, the inequality $|z_0 - z_L| <<$ $|z_c - z_L|$ is valid (see Fig.2). Using the multipole expansions Eqs. (13) and (16), we can get

$$
f(z_0) = -\frac{1}{2\pi} \left[\ln(z_0 - z_c) M_0 + \sum_{k=1}^{\infty} \frac{M_k(z_c)}{(z_0 - z_c)^k} \right], \tag{22}
$$

and

$$
g(z_0) = -\frac{1}{2\pi l_i} \sum_{k=1}^{\infty} \frac{M'_k(z_c)}{(z_0 - z_c)}, \tag{23}
$$

respectively.

Expanding $f(z_0)$ and $g(z_0)$ about z_L using the Taylor series expansion, we can derive the following local expansions

$$
\sum_j G(z_0, z_j) \alpha_j = \sum_{l=0}^{\infty} L_l(z_L)(z_0 - z_L)^l, \tag{24}
$$

$$
\sum_j B(z_0, z_j) \alpha_j = \sum_{l=0}^{\infty} L'_l(z_L)(z_0 - z_L)^l, \tag{25}
$$

where the coefficients are given by the following equations

$$
L_l(z_L) = \begin{cases} -\frac{1}{2\pi} \left[\ln(z_L - z_c) M_0 + \sum\limits_{k=1}^{\infty} \frac{M_k(z_c)}{(z_L - z_c)^k} \right], & l=0, \\ -\frac{1}{2\pi} \left[\frac{(-1)^{l+1}}{l} \frac{M_0}{(z_L - z_c)^l} + \frac{(-1)^l}{(z_L - z_c)^l} \right. \\ \left. \sum\limits_{k=1}^{\infty} \begin{pmatrix} l+k-1 \\ k-1 \end{pmatrix} \frac{M_k(z_c)}{(z_L - z_c)^k} \right], & l \geq 1, \end{cases} \tag{26}
$$

$$
L'_l(z_L) = \begin{cases} -\frac{1}{2\pi l_i} \sum\limits_{k=1}^{\infty} \frac{M'_k(z_c)}{(z_L - z_c)^k}, & l=0, \\ -\frac{1}{2\pi l_i} \left[\frac{(-1)^{l+1}}{l} \frac{M'_0}{(z_L - z_c)^l} + \frac{(-1)^l}{(z_L - z_c)^l} \right. \\ \left. \sum\limits_{k=1}^{\infty} \begin{pmatrix} l+k-1 \\ k-1 \end{pmatrix} \frac{M'_k(z_c)}{(z_L - z_c)^k} \right], & l \geq 1, \end{cases} \tag{27}
$$

which are called M2L translation.

Using Eq. (26), the local expansion for the *A* kernel sum can be naturally obtained by

$$\sum_j A(z_0, z_j)\alpha_j = n(z_0) \sum_{l=1}^{\infty} L_l(z_L) l(z_0 - z_L)^{l-1}. \tag{28}$$

3.4 L2L translation

Suppose the point for local expansion is moved from z_L to $z_{L'}$ (see Fig. 2), using Eq. (24), we obtain the following local equation

$$\sum_j G(z_0, z_j)\alpha_j = \sum_{l=0}^{n} L_l(z_L) \left[(z_0 - z_{L'}) + (z_{L'} - z_L)\right]^l. \tag{29}$$

Applying the binomial formula and the relation $\sum_{l=0}^{n} \sum_{m=0}^{l} = \sum_{m=0}^{n} \sum_{l=m}^{n}$ to Eq. (29), we can derive the following equation

$$\sum_j G(z_0, z_j)\alpha_j = \sum_{l=0}^{n} L_l(z_{L'})(z_0 - z_{L'})^l, \tag{30}$$

where the new coefficients are given by the following L2L translation

$$L_l(z_{L'}) = \sum_{m=l}^{n} \binom{m}{l} (z_{L'} - z_L)^{m-l} L_m(z_L), \text{ for } l \geq 0. \tag{31}$$

Note that Eqs. (24) and (25) have the same form, and the L2L translations for *B* kernel sum can be easily obtained by replacing $L_l(z_{L'})$ and $L_m(z_L)$ with $L'_l(z_{L'})$ and $L'_m(z_L)$, respectively.

It is noted that the M2M and L2L translations for *A* kernel sum and *G* kernel sum remain the same.

3.5 FMM formulation for the SBM

The algorithms or procedures for the FM-SBM are similar to those for the FM-BEM [Liu et al. (2005); Nishimura (2002)]. The main steps in the FM-SBM can be described as follows:

Step 1. *Discretization.* Discretize the boundary *S* in the same manner as in the conventional SBM, that is, place a certain number of collocation points, which are also the source points at the same time, on *S* as shown in Fig. 1.

Step 2. *Construction of quad-tree structure.* Consider a square that covers all the collocation points. Call this square the cell of level 0 (see Fig. 3). Now take a cell (a parent cell) of level $l(l \geq 0)$ and divide it into four equal sub squares whose edge length is half of that of the parent cell and call any of them a cell (a child cell) of level $l + 1$ if some collocation points belong to this square (see Fig. 3). Stop dividing a cell if the number of points in that cell is less than a given number as one in the example is shown in Fig. 3. A childless cell is called a leaf.
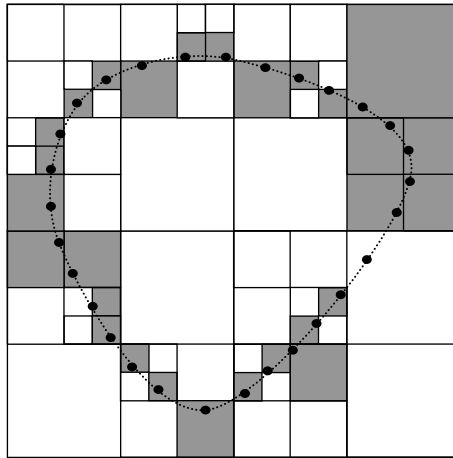

Figure 3: Construction of a tree structure.

Step 3. *Upward pass.* First we compute the multipole moments associated with leaves using Eqs. (14) and (17) for up to p terms. For a parent cell, we compute the multipole moments by adding all the moments of its children using the M2M translations Eqs. (20) and (21), in which $z_{c'}$ is the centroid of the parent cell and z_c the centroid of a child cell (see Fig. 2). This procedure is repeated for $l \geq 2$ tracing the tree structure upward.

Step 4. *Downward pass.* Two cells are called "adjacent cells at level l" if these cells are both of the level l and share at least one common vertex. Two cells are "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. Now we compute the local expansion associated with a cell C. The local expansion associated with C is the sum of the contributions due to the collocation points in cells of the interaction list of C and the contributions due to collocations points in cells which are not adjacent to the parent of cell C. The former is calculated by using M2L translations Eqs. (26) and (27), with moments

associated with cells in the interaction list, and the latter by using L2L translations which shifts the local expansions from the centroid of C's parent to the centroid of cell C (see Fig. 2). We repeat these procedures starting from level 2 and tracing the tree structure downward to all leaves. For a cell at level 2, the coefficients of the local expansions are only involved M2L translation.

Step 5. *Evaluation of the sum in Eq. (4) or (5).* Let C be a leaf to which the collocation point z_0 belongs, and the values of α_j be given. We calculate the contributions from source points in leaf C and its adjacent cells directly as in conventional SBM. We also compute contributions from all other cells by using the coefficients of the local expansions associated with C which have been computed in Step 4.

Step 6. First, we let α_j equals 1 to compute the diagonal coefficients of the coefficient matrix in Neumann boundary equation (5) using Step 5. Then, we update the unknown vector in the system produced by using Eqs. (4) or (5), and go back to Step 3 for the matrix and unknown vector multiplication until the solution converges within a given residue using generalized minimal residual (GMRES) solver.

3.6   Preconditioners for the FM-SBM

Similar to FM-BEM, a good pre-conditioner for FM-SBM is crucial for its convergence and computing efficiency. Since the conditioning number of the SBM is almost the same as that of the BEM, the present FM-SBM directly utilizes the block diagonal pre-conditioner proposed for FM-BEM [Wang and Yao (2004)]. The pre-conditioner is based on leaves of the hierarchical tree, which is formed on each leaf using direct evaluation of the fundamental solution with the collocation points within that leaf and their corresponding source points. This pre-conditioner will be inefficient if the number of collocation points in a leaf is large.

4   Numerical examples

To verify the feasibility and efficiency of the proposed FM-SBM, two numerical examples are examined in which the results are compared with the corresponding exact solutions and those obtained using conventional SBM. To assess the accuracy of the FM-SBM, the numerical results will be compared with exact solutions in terms of the relative error defined by

$$\text{Relative Error} = \left[\frac{1}{M} \sum_{k=1}^{M} \left(\frac{I_{\text{numerical}}^k - I_{exact}^k}{I_{exact}^k} \right)^2 \right]^{1/2}, \tag{32}$$

where $I_{\text{numerical}}^k$ and I_{exact}^k stand for the numerical and analytical solutions at k^{th} calculated point, respectively. Numerical results are presented below. All the com-

putations were implemented on a Core(TM) II laptop PC with a 2.13 GHz CPU and 4GB RAM.
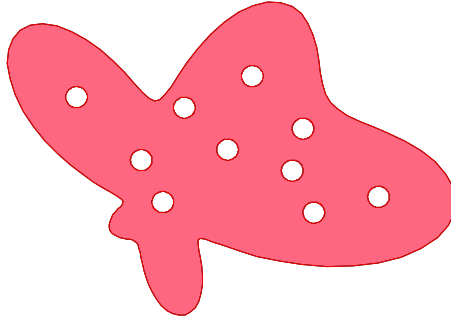

Figure 4: An amoeba-like domain with ten holes.

4.1   An amoeba-like domain with ten holes

First, we consider a simple heat condition problem in an amoeba-like domain with ten inner holes (see Fig. 4), whose outer boundary is defined as

$$\Gamma = \left\{ (r\cos\theta, r\sin\theta) : r = 2(e^{\sin\theta}\sin^2(2\theta) + e^{\cos\theta}\cos^2(2\theta)), 0 \le \theta \le 2\pi \right\}, \quad (33)$$

and the ten inner holes are randomly distributed, whose radius are all equal to 0.2. In this case, the analytical solution is given by

$$u(x,y) = e^y(\cos x + \sin x). \tag{34}$$

The Dirichlet boundary conditions are specified on the outer boundary while the Numann boundary conditions are specified on the rest of the boundaries.

We discretize the inner and outer boundaries with the same number of boundary points, and every inner holes boundary has the same points. The conventional SBM code uses direct solver Gauss elimination for solving the linear system equations. For FM-SBM, the numbers of the terms for both moments and local expansions were set to 20, and the maximum number of collocation points allowed in a leaf of the tree structure was prescribed to 400. The residue for iteration is taken as 1×10^{-6}, and the initial guess is zero.

With the number of collocation points varying from 2,000 to 360,000, the relative error of u and q by either the FM-SBM or the conventional SBM are listed in Table 1. It can be observed from this table that the FM-SBM performs as well as

Table 1: Relative errors of computed potential and normal derivative.

N	u		q	
	Conventional SBM	FM-SBM	Conventional SBM	FM-SBM
2,000	1.56080e-003	1.56076e-003	3.37403e-003	3.37364e-003
4,000	5.51119e-004	5.51099e-004	2.09459e-003	2.09315e-003
8,000	2.75707e-004	2.76151e-004	1.04402e-003	1.04840e-003
20,000	-	1.10151e-004	-	4.10237e-004
60,000	-	3.72716e-005	-	1.33734e-004
200,000	-	1.17167e-005	-	3.40158e-005
360,000	-	8.19122e-006	-	1.71258e-005

the conventional SBM when the number of collocation points less than 8,000. As the number of boundary collocation point further increases, the conventional SBM becomes infeasible due to explosive increase of memory demands. On the other hand, the proposed FM-SBM can continue to provide accurate results even as *N* reaches 360,000 without any difficulty.

The comparisons of CPU time and memory requirement between FM-SBM and conventional SBM are illustrated in Figs. 5 and 6, respectively. To show the order of complexity of FM-SBM more clearly, logarithmic scale was used in Figs. 5 and 6. The vertical axis of Fig. 5 denotes the 10-base logarithmic of CPU time, simplified as *log(t)*, and the horizontal axis represents the 10-base logarithmic of number of collocation points, simplified as *log(N)*. Similarly, the vertical axis of Fig. 6 denotes the 10-base logarithmic of memory requirement, simplified as *log(m)*, and the horizontal axis represents the 10-base logarithmic of number of collocation points, simplified as *log(N)*. These figures show a significant edge of the present FM-SBM in the savings of CPU time and memory, which are observed both approximately proportional to the number of collocation points. The results verify that both multiplication operations and storage costs of FM-SBM are $O(N)$. This shows that the FM-SBM has great promise to handle large-scale problems.

Fig. 7 show the iteration number and maximum number of boundary points in a leaf node vary with the number of collocation points, respectively, as the residue for iteration and the maximum of points allowed in a leaf of the tree structure were fixed. It can be concluded that, in general, the iteration number and the maximum number of boundary increase with the number of collocation points.

The relationship between the iteration number and residue of iteration is given in
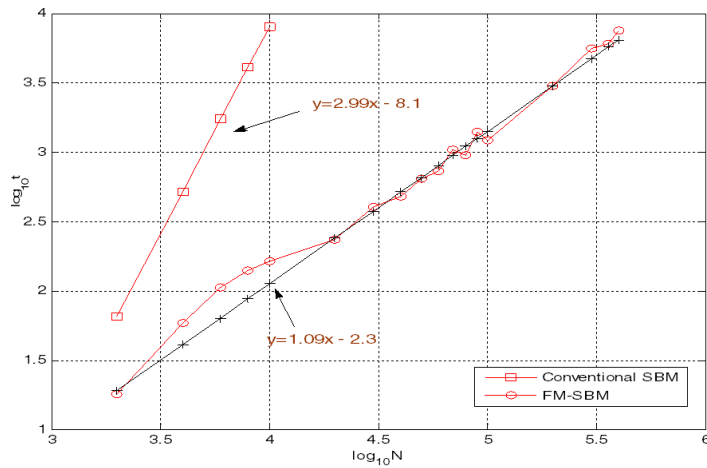
Figure 5: Comparison of the CPU time for the problem of the amoeba-like domain with ten holes.
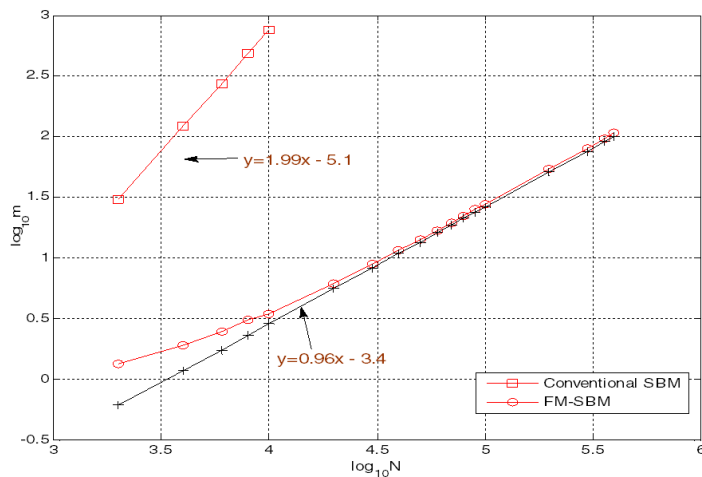

Figure 6: Comparison of the memory requirement for the problem of the amoeba-like domain with ten holes.
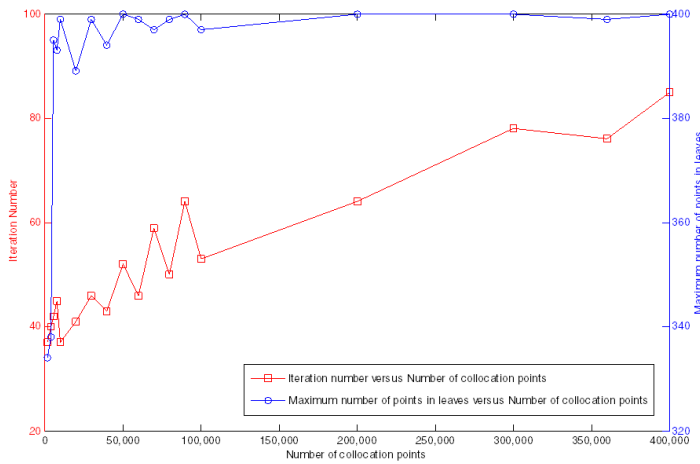
Figure 7: Relationships of iteration number and maximum number of points in leaves with the number of collocation points.
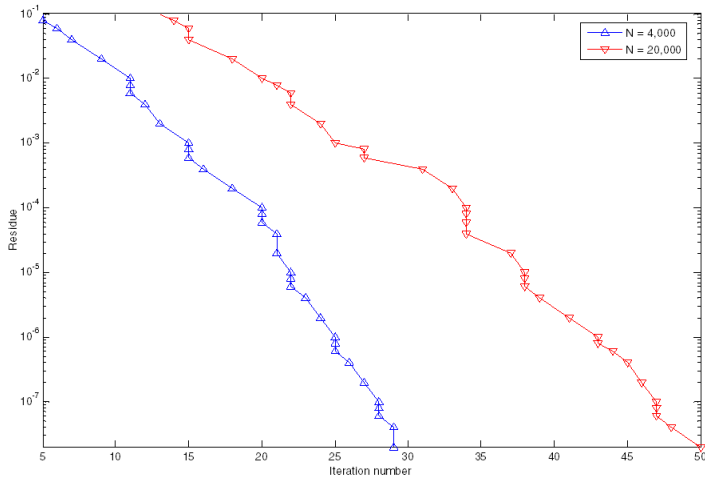
Figure 8: Iteration number versus residue of iteration under different scale of problems.

Fig. 8. From this figure, we can obtain that, when the residue of iteration gets less, FM-SBM needs more iteration steps to get the resolution. And, this tendency doesn't matter with the scale of the problem. We can also get that, for the same residue of iteration, when the scale becomes larger, FM-SBM will need more iteration steps.

Table 2: Relative errors of computed potential and normal derivative.

$N = 40,000$			$N = 200,000$		
N_{leaf}	N_{level}	t	N_{leaf}	N_{level}	t
20	6	630.32	20	7	4000.97
40	5	490.72	40	7	4080.08
80	5	502.31	80	6	3975.51
100	5	513.24	100	6	4000.73
200	4	475.23	200	5	3000.42
400	4	480.63	400	5	3005.87
600	3	500.74	600	5	3900.21
800	3	618.15	800	4	4100.59

In Table 2, N_{leaf} and N_{level} denote the number of collocation points in a leaf and the number of levels of the tree structure, respectively. Table 2 shows that, when the number of collocation points increases, the number of levels decreases. As we also know, when the number of collocation points increases, it will need more CPU time to calculate the interaction of nodes in a leaf. So, as illustrated in Table 2, there will be a better value of the number of collocation points which is 400 in this example to get less CPU time.

4.2 A square plate with numerous circular holes

Here, the thermal conductivity of a square plate with uniformly distributed circular holes, or a perforated plate, is studied using the FM-SBM (see Fig. 9). Several models of the plate, with increasing number of holes, are considered. Each model of the plate contains a total of $2m \times 2m$ holes, with $m = 2, 3, ..., 28, 30$. The radii of the holes are the same and equal to $4.0/(2m + 2)$. Each hole is discretized with 100 collocation points and the outer boundary of the plate is discretized with $400m$ collocation points.

In this example, the analytical solution is given by

$$u(x, y) = \sin(x) \cosh(y). \tag{35}$$

The Dirichlet boundary conditions are specified on the outer boundary, and the Numann boundary condition are specified on the remaining boundaries. For FM-SBM, the number of terms equals 20 for multipole and local expansions, and the maximum of collocation points allowed in a leaf is set to 400. The residue for iteration is prescribed to 1×10^{-6}, and the initial guess is zero.
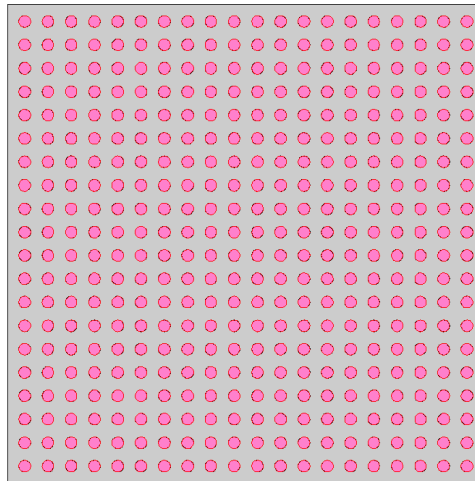

Figure 9: A model of the square plate with 400 uniformly distributed circular holes

Table 3: Relative error of computed potential and normal derivative for the square plate with circular holes.

Model (Number of holes)	u		q	
	Conventional SBM	FM-SBM	Conventional SBM	FM-SBM
4×4	4.78940e-004	4.79008e-004	7.73679e-004	7.73763e-004
6×6	2.94266e-004	2.94242e-004	4.66750e-004	4.66878e-004
8×8	2.43501e-004	2.43278e-004	3.70455e-004	3.70560e-004
10×10	1.83726e-004	1.83860e-004	3.75355e-004	3.75282e-004
16×16	-	1.10719e-004	-	1.22663e-004
20×20	-	8.71265e-005	-	9.33177e-005
36×36	-	4.78870e-005	-	7.46717e-005
40×40	-	4.32436e-005	-	6.82993e-005
56×56	-	3.07946e-005	-	3.19743e-005
60×60	-	2.85264e-005	-	2.97691e-005

Table 3 illustrates the relative errors of u and q by both the FM-SBM and conventional SBM. As shown in this table, the conventional SBM can only provide accurate numerical solution when the number of circular holes less than 100, after which the method becomes infeasible due to memory limitations. In sharp contrast, the proposed FM-SBM is still workable when the number of circular holes is as large as 3,600, with the largest relative error less than 3E-5.
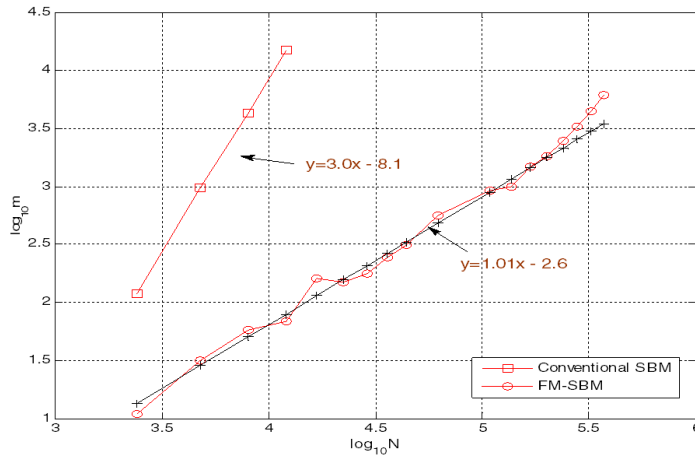

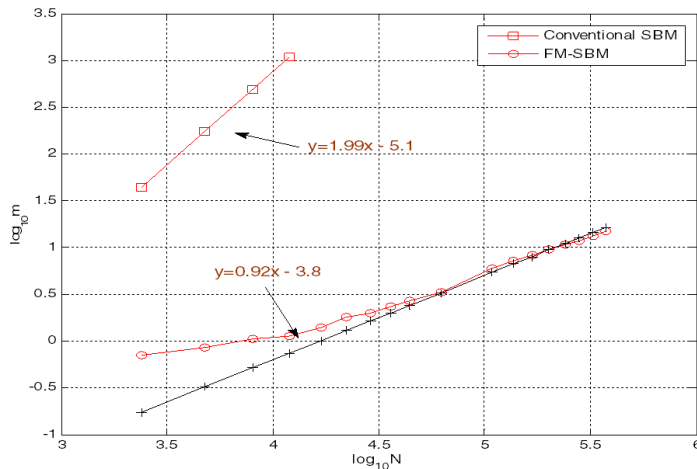Figure 10: Comparison of the CPU time for the square plate with circular holes.

Figure 11: Comparison of the memory for the square plate with circular holes.

The relationships of iteration number and maximum number of points in leaves with the number of collocation points, is shown in Fig. 12. From this figure, we can conclude that the iteration number and the maximum number of boundary increase with the number of collocation points, generally.
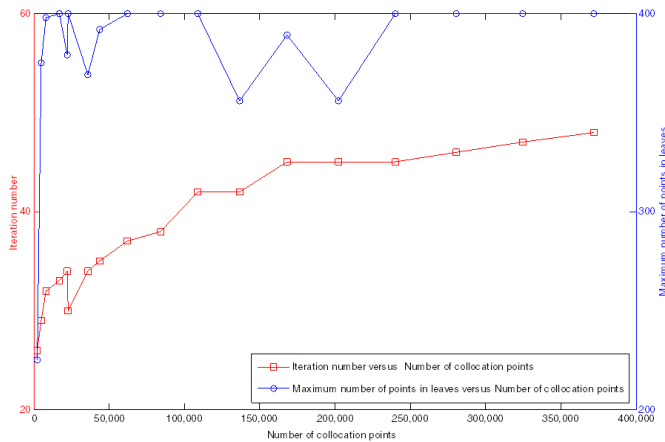

Figure 12: Relationships of iteration number and maximum number of points in leaves with the number of collocation points.

5   Conclusions

In this paper, a fast multipole accelerated singular boundary method is presented for 2D potential problems. Complete formulations and implementation details for FM-SBM are provided. Two numerical examples demonstrate the efficiencies of the present strategy for solving large-scale problems.

The convergence issue in the FM-SBM requires further investigations, and a better pre-conditioner is required to devise. Furthermore, the algorithms of the FM-SBM can also be improved. In this work, the calculation of the potential and its derivatives were done by direct calculation using SBM interpolation formula, which may further be speeded up by applying FMM, when the number of data points is comparable to that of collocation points. All these issues merit further investigations based on this study.

References

Atluri, S. N.; Liu, H. T.; Han, Z. D. (2006): Meshless local Petrov-Galerkin (MLPG) mixed collocation method for elasticity problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, no. 3, pp. 141-152.

Chen, C. S. (1995): The method of fundamental solutions for non-linear thermal explosions. *Commun. Numer. Methods. Eng.*, vol. 11, no. 8, pp. 675-681.

Chen, W.; Gu, Y. (2012): An improved formulation of singular boundary method. *Adv. Appl. Math. Mech.*, vol. 4, no. 5, pp. 543-558.

Chen, W.; Wang, F. Z. (2010): A method of fundamental solutions without fictitious boundary. *Eng. Anal. Bound. Elem.*, vol. 34, no. 5, pp. 530-532.

Dong, L.; Atluri, S. N. (2011): Development of T-Trefftz four-node quadrilateral and Voronoi Cell Finite Elements for macro- & micromechanical modeling of solids. *CMES: Computer Modeling in Engineering & Sciences*, vol. 81, no. 1, pp. 69-118.

Dong, L.; Atluri, S. N. (2012): A simple multi-source-point Trefftz method for solving direct/Inverse SHM problems of plane elasticity in arbitrary multiply-connected domains. *CMES: Computer Modeling in Engineering & Sciences*, vol. 85, no. 1, pp. 1-44.

Fairweather, G.; Karageorghis, A. (1998): The method of fundamental solutions for elliptic boundary value problems. *Adv. Comput. Math.*, vol. 9, no. 1, pp. 69-95.

Greengard, L.; Rokhlin, V. (1987): A fast algorithm for particle simulations. *J. Comput. Phys.*, vol. 73, no. 2, pp. 325-348.

Gu, Y.; Chen, W. (2013): Infinite domain potential problems by a new formulation of singular boundary method. *Appl. Math. Model.*, vol. 37, no. 4, pp. 1638-1651.

Gu, Y.; Chen, W.; He, X.-Q. (2012a): Singular boundary method for steady-state heat conduction in three dimensional general anisotropic media. *Int. J. Heat. Mass. Transf.*, vol. 55, no. 17–18, pp. 4837-4848.

Gu, Y.; Chen, W.; He, X. (2014): Improved singular boundary method for elasticity problems. *Comput. Struct.*, vol. 135, no. 0, pp. 73-82.

Gu, Y.; Chen, W.; He, X. Q. (2012b): Domain-Decomposition Singular Boundary Method for Stress Analysis in Multi-Layered Elastic Materials. *CMC: Computers Materials & Continua*, vol. 29, no. 2, pp. 129-154.

Gu, Y.; Chen, W.; Zhang, C.-Z. (2011): Singular boundary method for solving

plane strain elastostatic problems. *Int. J. Solids. Struct.*, vol. 48, no. 18, pp. 2549-2556.

Liu, Y. J.; Nishimura, N. (2006): The fast multipole boundary element method for potential problems: A tutorial. *Engineering Analysis with Boundary Elements*, vol. 30, no. 5, pp. 371-381.

Liu, Y. J.; Nishimura, N.; Yao, Z. H. (2005): A fast multipole accelerated method of fundamental solutions for potential problems. *Engineering Analysis with Boundary Elements*, vol. 29, no. 11, pp. 1016-1024.

Nishimura, N. (2002): Fast multipole accelerated boundary integral equation methods. *Applied Mechanics Reviews*, vol. 55, no. 4, pp. 299-324.

Rokhlin, V. (1985): Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, vol. 60, no. 2, pp. 187-207.

Sarler, B. (2009): Solution of potential flow problems by the modified method of fundamental solutions: Formulations with the single layer and the double layer fundamental solutions. *Eng. Anal. Bound. Elem.*, vol. 33, no. 12, pp. 1374-1382.

Sarler, B.; Liu, Q. G. (2013): Non-singular method of fundamental solutions for two-dimensional isotropic elasticity problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 91, no. 4, pp. 235-266.

Wang, H. T.; Yao, Z. H. (2004): Application of a new fast multipole BEM for simulation of 2D elastic solid with large number of inclusions. *Acta Mechanica Sinica*, vol. 20, no. 6, pp. 613-622.

Young, D. L.; Chen, K. H.; Chen, J. T.; Kao, J. H. (2007): A modified method of fundamental solutions with source on the boundary for solving Laplace equations with circular and arbitrary domains. *CMES: Computer Modeling in Engineering & Sciences*, vol. 19, no. 3, pp. 197-221.

Zhang, J. M.; Yao, Z. H.; Li, H. (2002): A hybrid boundary node method. *Int. J. Numer. Methods. Eng.*, vol. 53, no. 4, pp. 751-763.

Zhang, J. M.; Yao, Z. H.; Tanaka, M. (2003): The meshless regular hybrid boundary node method for 2D linear elasticity. *Eng. Anal. Bound. Elem.*, vol. 27, no. 3, pp. 259-268.