 CMES, vol.58, no.2, pp.161-183, 2010

A New Adaptive Algorithm for the Fast Multipole Boundary Element Method

M. S. Bapat[1] **and Y. J. Liu**[1,2]

Abstract: A new definition of the interaction list in the fast multipole method (FMM) is introduced in this paper, which can reduce the moment-to-local (M2L) translations by about 30-40% and therefore improve the efficiency for the FMM. In addition, an adaptive tree structure is investigated, which is potentially more efficient than the oct-tree structure for thin and slender domains as in the case of micro-electro-mechanical systems (MEMS). A combination of the modified interaction list (termed L2 modification in the adaptive fast multipole BEM) and the adaptive tree structure in the fast multipole BEM has been implemented for both 3-D potential and 3-D acoustic wave problems. In the potential theory case, the code is based on the earlier adaptive algorithm proposed in (Shen, L. and Y. J. Liu (2007). "An adaptive fast multipole boundary element method for three-dimensional potential problems." Computational Mechanics 39(6): 681-691) with the so called "new FMM" where the M2L translations are replaced by the exponential (M2X, X2X, and X2L) translations. Suitable changes are proposed in the algorithm for the new adaptive fast multipole BEM. Finally, a new adaptive algorithm which encompasses all these modifications and the established algorithms is presented (that is, combining the original adaptive fast multipole BEM, L2 modification and adaptive tree for slender structures). Numerical results are presented to demonstrate the efficiencies of the new adaptive fast multipole BEM for solving both potential and acoustic wave problems. About 30-40% improvements in the computational efficiency are achieved with the L2 modification for all cases, and additional improvements are observed with the adaptive tree for some large-scale thin structures (MEMS models), without the lost of accuracy.

Keywords: Fast multipole method, boundary element method, interaction list, adaptive tree, MEMS

[1] CAE Research Laboratory, Department of Mechanical Engineering, University of Cincinnati, P.O. Box 210072, Cincinnati, Ohio 45221-0072, U.S.A.

[2] Corresponding author (*E-mail*: Yijun.Liu@uc.edu)

1 Introduction

The fast multipole method (FMM) pioneered by Rokhlin and Greengard (Rokhlin 1985; Greengard and Rokhlin 1987; Greengard 1988) has dramatically improved the computational efficiencies of the boundary element method (BEM). Both the computing time and memory storage requirement for the FMM accelerated BEM, or fast multipole BEM, are reduced to or close to $O(N)$ complexities (with N being the number of unknowns). Some of the early research on fast multipole BEM can be found in Refs. (Peirce and Napier 1995; Gomez and Power 1997; Fu, Klimkowski et al. 1998; Mammoli and Ingber 1999; Nishimura, Yoshida et al. 1999) and recent work in Refs. (Aoki, Amaya et al. 2004; Liu 2005; Liu, Nishimura et al. 2005; Liu, Nishimura et al. 2005; Wang and Yao 2005; Liu and Nishimura 2006; Liu and Shen 2007; Shen and Liu 2007; Liu 2008; Liu, Nishimura et al. 2008; Wang, Hall et al. 2008) on potential and elasticity problems. BEM models with the unknowns above one million can now be solved routinely on desktop PCs using the fast multipole BEM (More details and other examples of the fast multipole BEM can be found in the review paper by Nishimura (Nishimura 2002) and the textbook by Liu (Liu 2009)). However, challenges remain in improving the efficiencies of the fast multipole BEM in solving large-scale BEM models with complicated geometries, frequency- and time-dependent problems, and coupled fields. Optimization of the existing algorithms in the FMM and development of new ones are still needed to further improve the efficiencies of the fast multipole BEM in solving real engineering problems.

The work by Rokhlin and Greengard (Rokhlin 1985; Greengard and Rokhlin 1987; Greengard 1988); formed the foundation of the original FMM. In the original FMM, two lists are created for calculating the contributions from other cells at any given level in a tree structure, i.e., the list of adjacent cells and the interaction list. The adaptive algorithms (Greengard 1988; Cheng, Greengard et al. 1999; Liu and Shen 2007; Shen and Liu 2007) accelerate the calculations by optimizing the list of adjacent cells by introducing two additional lists. This four list method has known to have a few issues as shown by Cheng, Greengard et al. (Cheng, Greengard et al. 1999) and Shen and Liu (Shen and Liu 2007) .

The oct-tree concept for the elements/nodes was first introduced for the FMM by Rokhlin and Greengard (Rokhlin 1985; Greengard and Rokhlin 1987; Greengard 1988), which is good for domains that have a bulky shape like a cube. A binary tree was proposed by Urago, et al. (Urago, Koyama et al. 2003) and utilized in solving electromagnetic problems, achieving a reduction of about 50% in CPU time. Zhang and Tanaka (Zhang and Tanaka 2007) proposed an adaptive tree structure for solving problems with domains not in a bulky shape. They have demonstrated the utility of the adaptive tree structure together with an adaptive selection of the expansion order for slender domains and observed improved efficiencies. This adaptive tree approach in the FMM can provide potential advantages for solving thin film or thin shape problems using the BEM (Luo, Liu et al. 1998; Chen and Liu 2001; Liu and Fan 2002).

In this paper, we first briefly review the various tree structures in the FMM, and then extend the adaptive tree structure in conjunction with a modification to the interaction list (termed L2 modification) to the adaptive fast multipole BEM. The combination of the adaptive tree and the L2 modification are implemented in the fast multipole BEM for 3-D potential problems where the adaptive algorithm based on the new FMM is employed (Liu and Shen 2007; Shen and Liu 2007); and for the 3-D acoustic wave problems where an adaptive algorithm is applied (Shen and Liu 2007; Bapat, Shen et al. 2009); The L2 modification can accelerate the fast multipole BEM codes up to 30-40% in all the cases. Additional improvements can be achieved with the adaptive tree for large models of slender structures.

2 Data Structures in the FMM

Use of efficient data structures to accelerate the fast multipole codes has been an important aspect of the FMM algorithms. Various tree data structures have been proposed for the FMM that demonstrate different utilities. Below we discuss the various data structures.

2.1 Quad-tree or oct-tree

The simplest tree data structures introduced in the FMM are the quad-tree (for 2-D) and the oct-tree (for 3-D) data structures. These were first introduced by Rokhlin and Greengard (Rokhlin 1985; Greengard and Rokhlin 1987; Greengard 1988). In this approach, a parent square or cube is divided into 4 equal child squares or 8 equal cubes at each level for a 2-D or 3-D domain, respectively. The details of this approach can be found in Refs. (Rokhlin 1985; Greengard and Rokhlin 1987; Greengard 1988; Liu and Nishimura 2006; Liu 2009).

2.2 Binary tree

The motivation of the binary tree (Anderson 1999) is to reduce the number of M2L translations in the FMM algorithm. Urago, et al. (Urago, Koyama et al. 2003) have shown that the number of M2L translations decrease by about 25% in these cases. However, the number of M2M and L2L translations increases. Despite this increase, the total CPU time reduces because the total number of M2M translations and L2L translations are much fewer than M2L translations in the traditional FMM algorithm. The usage of minimum volume bounding boxes (tightening the cells) has shown further improvement in efficiency in their approach (Urago, Koyama et al. 2003).

2.3 Adaptive tree

The adaptive tree structure has been created with the motive to develop an efficient method for slender and thin domains. In these cases, starting with a cube for the original domain is not efficient (Figure 1). Zhang and Tanaka .(Zhang and Tanaka 2007) proposed a method of creating a minimum volume bounding box around the region. This box is further subdivided into approximately cubic boxes at level 1. The boxes are then further subdivided into approximate cubes in the ensuing levels (Figure 2). This method was shown to be very useful for slender and plate-like objects for models with moderate number of DOFs (up to 100,000) by Zhang and Tanaka .(Zhang and Tanaka 2007).

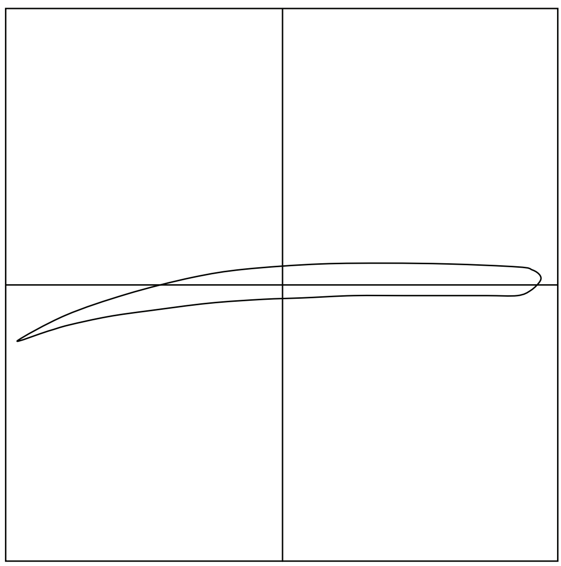

Figure 1: An oct-tree for a slender domain.

3 L2 Modification

It is well known that the M2L translations in the FMM are the most expensive ones to compute, besides the direct evaluations. The L2 modification or the modification of the interaction list (L2 list) is introduced in this work in order to reduce the number of M2L translations in the quad-tree or oct-tree (and also adaptive tree)

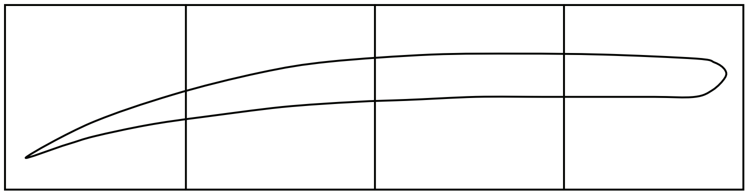

Figure 2: An adaptive tree for the slender domain.

data structure. In the traditional FMM algorithm, the L2 list or the interaction list of cell *b* contains all cells on the same level that are not neighbors of cell *b*, but their parent cells are neighbors of the parent of cell *b*. In this definition, the interaction list of any cell *b* only contains cells at the same level of cell *b*. In contrast, if the cells at level *b* are substituted with cells at level (*b*-1) wherever possible, the number of M2L translations can be reduced significantly.

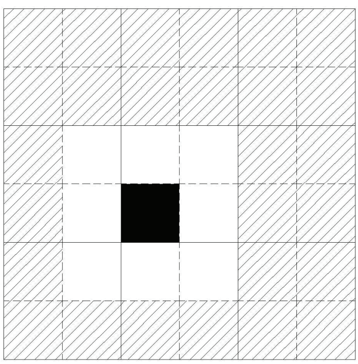

Figure 3: Traditional L2 list (27 cells).

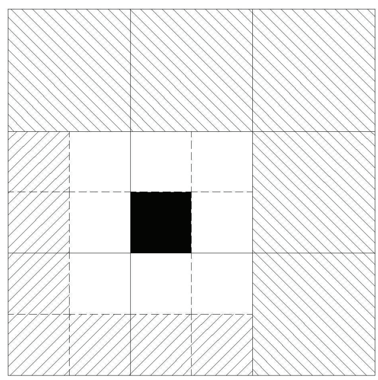

Figure 4: L2 modification for the interaction list (12 cells).

In Figure 3 and Figure 4 for the 2-D FMM algorithm, the black box represents cell *b*. In Figure 3, the maximum possible number of cells (27 cells) of the interaction list are shown (shaded squares). In Figure 4 when the parent cells replace the four child cells in the interaction list, the total number of the interaction list cells (shaded squares) is reduced to 12, a reduction of 56%. Greengard ((Greengard 1988), Lemma 2.2.2) has shown that the local expansions form a convergent series when the circles/spheres enclosing the target cell for the local expansion and source cell for the multipole expansion are well separated. This is referred to as the *distance* *criterion* in this paper. It can be clearly seen that the distance criterion for the M2L translations is still satisfied for the parent cells (That is, they are still separated from the target cell). In the case of 3-D domains, the maximum number of cells in the interaction list changes from 189 to 52, a reduction of 72%. Quite evidently, this L2 modification will yield more CPU time savings in the FMM, especially for models with densely packed nodes or elements.

4 The New Adaptive Algorithm

The adaptive algorithms in Refs. (Greengard 1988; Cheng, Greengard et al. 1999) are intended to optimize the list of adjacent cells in the traditional FMM. Cheng et al. (Cheng, Greengard et al. 1999) have modified the approach by Greengard. Shen and Liu (Shen and Liu 2007) have also briefly commented the problems on the same. In this paper, we prefer to modify the original algorithm by Greengard (Greengard 1988) in a separate fashion. The traditional scheme of creating the list of adjacent cells is shown in Figure 5. The black box *b* in Figure 5 depicts a leaf cell for which the adjacent list is being created. The hashed boxes, which are at the same level as box *b*, form the adjacent list. The adjacent list boxes can be at very high level as seen in Figure 5 (the smaller hashed boxes). Direct computation for these high level cells uses a lot of computation time. In Greengard's algorithm (Greengard 1988) to optimize the adjacent list, the adjacent list is further divided into three lists to reduce the number of the direct evaluations. He describes these lists as L1, L3 and L4. Greengard (Greengard 1988) proposed the definition of neighboring cells as the cells which share a boundary or vertex with the box *b*. By this definition, the distance criterion for M2L translation is observed not to be obeyed in some cases. When more leaf cells are spread across more tree levels, there are more cells which do not obey the M2L distance criterion. This was first pointed out by Shen and Liu (Shen and Liu 2007).

The new definition for *neighboring cells* used in this paper includes the cells for which the M2L distance criterion does not work. More precisely, for any two cells, if their circumscribing spheres (for 3-D) or circles (for 2-D) intersect or touch each other, then the two cells are said to be neighbors.

The lists L1, L2, L3 and L4 for a cell *b* are defined as follows:

- List 1 of a leaf cell *b* consists of *b* itself and all other leaf cells which are neighbors of *b* (see new definition of neighbors above). List 1 is empty if *b* is not a leaf cell.

- List 2 of any cell *b* contains the following:

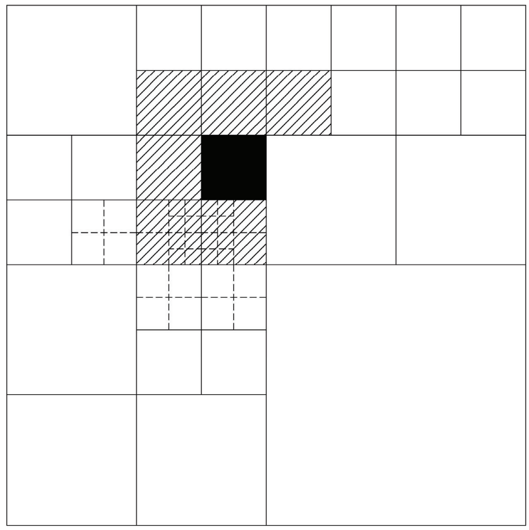

Figure 5: Traditional list of adjacent cells.

1. Neighbors of the parent cell of *b*, which are at the same level as the parent cell of *b* and are not in contact with *b* itself;
2. Cells *c* at the same level as *b* whose parents are neighbors of cell *b* but they are not neighbors of *b*.

- List 3 of a leaf cell *b* contains all descendents *c* of the neighboring cells of *b*, where parents of *c* are neighbors of box *b*, but *c* is not a neighbor of *b*. List 3 is empty if *b* is not a leaf cell. (A descendent cell is a child cell, grandchild cell, and so on.)

- List 4 of any cell *b* contains all cells *c* such that *b* belongs to list 3 of cell *c*, and such that *c* is not included in the list 2 of cell *b*.

The differences between Greengard's work (Greengard 1988) and the above definitions are the definitions of the neighboring cells and the L2 list.

The utilization of the adaptive tree structure with the adaptive algorithm requires the distance criteria for M2L to be obeyed. If the new definition for "neighboring cells" and the above definitions of lists are used, the adaptive algorithm can be implemented in conjunction with the adaptive tree.

Using both improvements, that is, adaptive tree with adaptive algorithm and the L2 modification, all FMM codes can be accelerated. Zhang and Tanaka (Zhang and Tanaka 2007) proposed the adaptive tree structure in which the cells are near cubic or square. It is observed that for the L2 modification, fewer M2L translations will be required if the cells are exactly cubic (3-D) or square (2-D). In this adaptive algorithm, for an adaptive tree discretizations of the space, square/cubic cells are formed from level 1 onwards. The level 0 bounding box is parallelepiped/ rectangular in shape.

The "new FMM" developed for the 3-D potential problems involves applying the M2X, X2X and X2L translation as compared to a single M2L translation. Detailed description of the "new FMM" can be found in Yoshida's dissertation (Yoshida 2001). Shen and Liu (Shen and Liu 2007) combined the new FMM with the adaptive algorithm to achieve better results for 3-D potential problems.

The main difference between the formulation for the potential and acoustic problems is the cells for which the L1, L3 and L4 lists are defined. In the case of the "new FMM" the lists are defined for the source cell (containing the collocation points) and for the discussion so far the lists were defined for the target cells (containing the integration points).

Another point to note regarding the "new FMM" is that the cells in which the M2L translation occurs are of the same size. The original formulas for the M2X, X2X and X2L translation can be written as follows:

M2X Translation:

$$X(k,j;O) = X|M_{n,m}^{k,j}(d)M_{n,m}(O), \tag{1}$$

where $X(k,j;O)$ is the coefficient of the exponential expansion centered at O, $M_{n,m}(O)$ is the moment centered at O and $X|M_{m,n}^{k,j}(d)$ is the M2X operator which is a function of d, the length of the side of the cell.

X2X Translation:

$$X(k,j;O_1) = X|X_{k,j}^{k',j'}(OO_1)X(k',j';O), \tag{2}$$

where $X|X_{m,n}^{k,j}(OO_1)$ is the X2X operator which is a function of the distance OO_1 between the center of the two cells.

X2L Translation:

$$L_{n,m}(O) = L|X_{k,j}^{n,m}(d)X(k,j;O), \tag{3}$$

where the X2L operator $L|X_{k,j}^{m,n}(d)$ is also a function of d, the edge length of the cell.

In the L2 modification for the new FMM, the four lists are defined for the source cell. The modified L2 list is formed as per the previous definition. In the downward pass though, there is a change in the algorithm. For every cell b, we do M2X, X2X and X2L translations for all cells c in whose L2 list the original cell b lies. Cells c lies at the same level as b or one level lower. If d_1 is the side length of the source cell b and d_2 the side length of the target cell c ($d_1 \geq d_2$), then the new M2X, X2X and X2L translations are given as follows:

M2X Translation:

$$X(k,j;O) = X|M_{n,m}^{k,j}(d_2)M_{n,m}(O), \tag{4}$$

where we treat the M2X operator as the same function of d_2 as earlier (replacing d).

X2X Translation:

$$X(k,j;O_1) = X|X_{k,j}^{k',j'}(OO_1)X(k',j';O), \tag{5}$$

which is identical to the earlier case.

X2L Translation:

$$L_{n,m}(O) = L|X_{k,j}^{n,m}(d_2)X(k,j;O), \tag{6}$$

where similar to M2X translation, X2L operator is the same function with d_2 replacing d. The above formulae are based on generalized Gaussian quadrature by Yarvin and Rokhlin (Yarvin and Rokhlin 1998).

Based on all the above new definitions and modifications, a new adaptive algorithm for utilizing the L2 modification can be summarized as follows:

For a 3-D domain, form a tree structure – either an oct-tree or an adaptive tree. Form L1, L2, L3 and L4 lists according to the new definitions discussed earlier for every cell of the tree.

Upward pass

Starting from the lowest level, the multipole moments are calculated for each cell and translated to the cell's parent's center using M2M. Continue the M2M until tree level 1 is reached. In case of oct-tree, even though the downward pass starts at level 2, M2M needs to be done till level 1. After the upward pass, every box down from level 1 should have a multipole moment set.

Downward pass

In the downward pass, the tree is traversed twice in the downward direction – once for performing the M2L translations and/or direct evaluations, and next for L2L and/or L2C (local expansion at the leaf cells) operations. The traversal starts at level 2 (or level 1 in the adaptive tree case).

At each level l, perform the following steps:

Step A.

1. For each cell b on level l, use M2L to translate the multipole moments from cells c in lists 2, 3 and 4 of cell b to the local coefficients of b. If the numbers of elements in both cells b and c are very small (e.g., less than 5), use the direct computation to replace M2L and the local expansion.

   When doing M2L translation using the "new FMM" (see Refs. (Yoshida 2001) and (Shen and Liu 2007)), the algorithm is made more efficient by doing M2X, X2X and X2L translations instead of a single M2L translation. In this case, the algorithm for doing these three translations is:

   - Do the M2X translations for all cells b at level l.
   - Do the X2X translations for all cells c in the lists 2, 3 and 4 of cell b.
   - Do the X2L translations for all those cells c.

2. For each box b on level l, use direct computation for all cells c in list 1.

Step B.

Translate local coefficients of b to the local coefficients of b's children using L2L translations. This step needs to be done only after the calculation of the local expansion coefficient for cell b is complete. Since the algorithm becomes too complex for the "new FMM", the L2L calculation is done in a separate downward traversal of the tree (after the M2L/direct calculation traversal).

After the downward pass, the calculation of local coefficients for each leaf is complete. In the above discussions, it is noted that the cells in lists 2, 3 and 4 are treated in the same way in the downward pass. Therefore, the three lists can be combined into one list in the new adaptive algorithm. However, in order to show the differences from the previous adaptive algorithms in Refs. (Greengard 1988; Cheng, Greengard et al. 1999) and (Shen and Liu 2007), these three lists are kept separately in this paper.

4.0.1 Evaluation of the integrals

For each leaf b (can be done during the second traversal of the downward pass), calculate the integral for each collocation point from local expansion of b. Add direct evaluations that have been calculated earlier.

More discussions on the basic steps in the fast multipole BEM can be found in Refs. (Greengard 1988; Yoshida 2001; Nishimura 2002; Liu 2009).

5 Numerical Examples

Several numerical examples are presented in this section to demonstrate the advantages of the new adaptive fast multipole algorithm, as compared with the earlier adaptive fast multipole BEM. Examples for both 3-D potential and acoustic wave problems are given, with the first three examples on the potential problems (including electrostatic analysis of MEMS) and the last example on an acoustic wave problem. Constant triangular elements are used in the codes.

A Dell® Vostro laptop PC with an Intel® Core2 Duo 2.00 GHz CPU, 4 GB RAM and Windows Vista 32-bit OS is used for solving the first two examples. A Dell® Precision desktop PC with an Intel® Core2 Duo 2.40 GHz CPU, 8 GB RAM and Windows Vista 64-bit OS is used for solving the last two examples.

5.1 A Block Model

First, we use a block model (Figure 6) with different aspect ratios to study a heat conduction problem with the developed new adaptive fast multipole BEM. The front and back surfaces (with the normal in the $+x$ or $-x$ direction) are applied with a temperature with its value equal to that of the x-coordinate of the surface. On all the other surfaces, a zero flux condition is imposed. The maximum heat flux on the front surface is computed using the previous fast multipole BEM code (termed Old FMBEM). The Old FMBEM includes the four list adaptive algorithm. The code has been first described by Shen and Liu (Liu and Shen 2007; Shen and Liu 2007); the new adaptive fast multipole BEM code with L2 modification only (New FMBEM I), and the new adaptive fast multipole BEM code with both L2 modification and the adaptive tree (New FMBEM II). In all the cases, only the conventional BIE is applied, 15 terms are used in the expansions, the number of elements allowed in a leaf is set to 100, and the tolerance for convergence is set at 10^{-6}.

Table 1 shows the results for the block model with four different sizes, changing from a bulky shape (a cube), to flat shapes, and finally to a slender shape. The computed maximum values of the heat flux (normal derivative of the temperature in this case) on the front surface using the three versions of the fast multipole BEM are reported in the table. From these results, it is seen that the accuracy of the new adaptive fast multipole BEM codes, with either the L2 modification only or both the L2 modification and the adaptive tree, is equivalent to that of the previous fast multipole BEM code. The error of about 1% in all the results can be further reduced with tightened parameters in the solutions.

The CPU time comparison of the three versions of the fast multipole BEM is also reported in Table 1. The effect of the L2 modification for the fast multipole BEM (New FMBEM I) is most evident in all the models studied, with the reductions

Table 1: Comparison of the results for the block model.

Model		*Max. Heat Flux*			*CPU Time*				
Size	Total DOFs	Old FMBEM	New FMBEM I	New FMBEM II	Old FMBEM	New FMBEM I	Savings	New FMBEM II	Savings
1X1X1	4,800	1.013	1.013	1.013	66.7	41.3	38.0%	49.2	26.2%
	10,800	1.012	1.012	1.012	214.9	123.1	42.7%	144.9	32.6%
	30,000	1.012	1.012	1.012	651.1	391.1	39.9%	463.5	28.8%
1X1X0.2	5,040	1.011	1.011	1.011	78.9	50.3	36.2%	73.3	7.2%
	14,000	1.011	1.011	1.011	328.7	209.3	36.3%	133.8	59.3%
	35,840	1.011	1.011	1.012	588.2	366.4	37.7%	536.0	8.9%
1X0.5X0.1	6,500	1.009	1.009	1.009	122.2	77.3	36.8%	87.7	28.3%
	16,640	1.010	1.010	1.010	206.8	134.3	35.1%	147.6	28.7%
	37,440	1.011	1.011	1.011	1,065.3	590.7	44.6%	715.1	32.9%
1X0.1X0.1	8,400	1.010	1.011	1.012	264.3	138.9	47.4%	162.2	38.6%
	16,464	1.010	1.010	1.016	334.4	192.1	42.5%	259.3	22.5%
	37,044	1.011	1.011	1.014	438.6	302.0	31.1%	695.8	-58.7%
		Exact Value: 1.000							

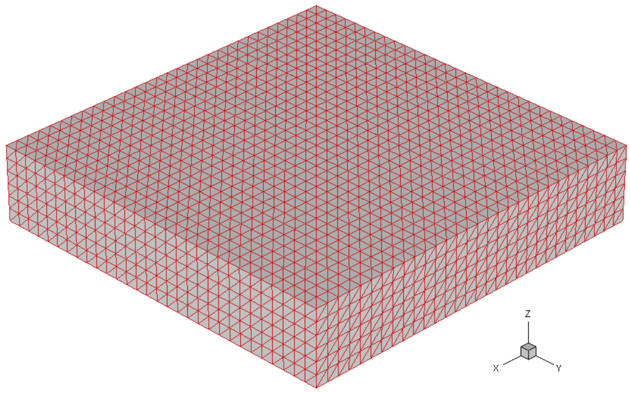

Figure 6: A block model for a heat conduction problem.

in the CPU time ranging from 31-47% as compared with the previous version (Old FMBEM). On the other hand, the effect of the fast multipole BEM with the adaptive tree in addition to the L2 modification (New FMBEM II) is mixed for all the models studied in this case. In fact, the reductions of the CPU time in general are less than those with the L2 modification only version of the code, except for one case when it delivered a 59% reduction for the flat domain. In another case, however, a -59% reduction for the slender domain is registered. These mixed results for the new adaptive fast multipole BEM code with both L2 modification and adaptive tree may stem from the overhead and variations associated with the adaptive tree and the relatively small model sizes used. The advantage of the adaptive tree will emerge in the following two examples, in which larger models will be studied. In addition, the drop in the effectiveness of the adaptive tree may be caused by the less effective preconditioners associated with the adaptive tree. With the adaptive tree, elements in a slender BEM model are more uniformly distributed among all the leaves, leading to smaller numbers of elements per leaf. This is good for the memory usage because fewer coefficients will need to be computed and stored for the preconditioner, but is less effective in preconditioning the system.

5.2 A Comb-Drive Model

Next, we study the same comb-drive model as used in Ref. (Liu and Shen 2007) to demonstrate the efficiency of the new fast multipole BEM. In this simplified comb-drive model, conducting beams of identical sizes are placed parallel to each other and with a small offset in the direction along the length of the beams. The beams are applied with a constant positive or negative voltage alternately and the electrical charge density is computed on the surfaces of the beams in this exterior domain problem. With the increase of the number of the beams in the model, the entire model becomes either a 1-D slender structure or a 2-D flat structure, both of which can potentially benefit from the adaptive tree structure implemented in the new fast multipole BEM. The dimensions and spacing of the beams are given in Ref. (Liu and Shen 2007) and each beam is discretized with 3260 elements. In this example, the dual BIE (Liu and Shen 2007) is applied, 10 terms are used in the expansions, the number of elements allowed in a leaf is set at 100 and the tolerance for convergence is 10^{-4}.

BEM models with 3, 7, 11, ..., and up to 101 beams are solved using the previous fast multipole BEM code (Old FMBEM) (Liu and Shen 2007; Shen and Liu 2007); the new adaptive fast multipole BEM code with L2 modification only (New FMBEM I), and the new adaptive fast multipole BEM code with both L2 modification and adaptive tree (New FMBEM II). The BEM models and the CPU time comparison results are given in Table 2. Again, the New FMBEM I code consistently delivered reductions of the CPU time ranging from 21-29%, whereas the New FMBEM II code delivered reductions of the CPU time ranging from 2-30%, for BEM models with DOFs below 200,000. The large variations of the effects of the New FMBEM II are caused by the variations of the tree structure when the problem size changes. For the last two BEM models (with DOFs above 200,000) for which the new adaptive tree code is more efficient in the usage of the memories and thus more coefficients in the direct evaluation can be saved for re-use (Shen and Liu 2007), the savings in the CPU time are significant as compared with the Old FMBEM and the New FMBEM I codes, both of which require larger memory for larger models.

A contour plot of the computed charge density for the 101-beam model is given in Figure 7. The differences in the BEM results with the old version and the new versions of the adaptive fast multipole BEM are less than 1% which is indistinguishable on these contour plots.

5.3 A Torsional MEMS Model

Next, we study a torsional accelerometer MEMS model (Figure 8) which is again an exterior electrostatic problem used to study the charge density distributions on the MEMS device. The model is discretized with 696,486 boundary elements and the mesh pattern is shown in Figure 9. For the boundary conditions, the substrate (the stationary part, shown in blue in Figure 8) is applied with a negative voltage and the rotor (the moving part, shown in red in Figure 8) is applied with a positive voltage. The new adaptive fast multipole BEM code is used to calculate the charge densities on this MEMS device. In this case, the dual BIE formulation is applied, 6

Table 2: Comparison of the CPU times for the comb-drive models.

Models		*CPU Times*				
Number of Beams	Total DOFs	Old FMBEM	New FMBEM I	Savings	New FMBEM II	Savings
3	9,780	44.3	33.1	25.3%	35.0	21.0%
7	22,820	119.5	86.7	27.5%	117.0	2.0%
11	35,860	212.6	151.2	28.9%	161.2	24.2%
15	48,900	294.0	216.1	26.5%	206.7	29.7%
23	74,980	474.6	354.5	25.3%	398.8	16.0%
31	101,060	668.2	523.3	21.7%	513.1	23.2%
43	140,180	969.2	752.7	22.3%	786.6	18.8%
55	179,300	1,424.9	1,128.7	20.8%	1,060.6	25.6%
65	211,900	5,299.4	4,170.2	21.3%	1,320.8	75.1%
101	329,260	8,852.6	8,070.0	8.8%	3,615.9	59.2%

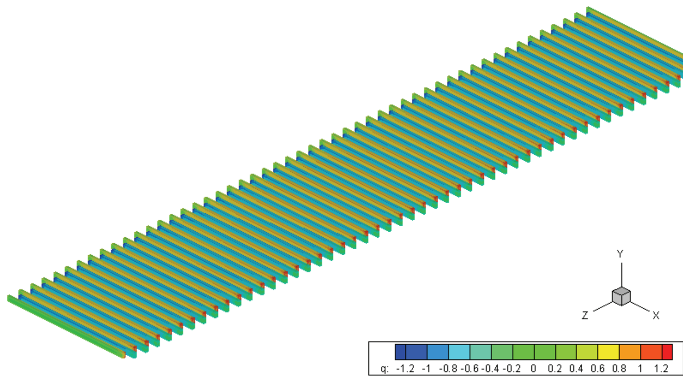

Figure 7: Computed charge densities on a comb-drive model with 101 beams that are discretized with 329,260 boundary elements.

terms are used in the expansions, the maximum number of elements in a leaf is set to 100 and the tolerance for convergence is set to 10^{-4}.

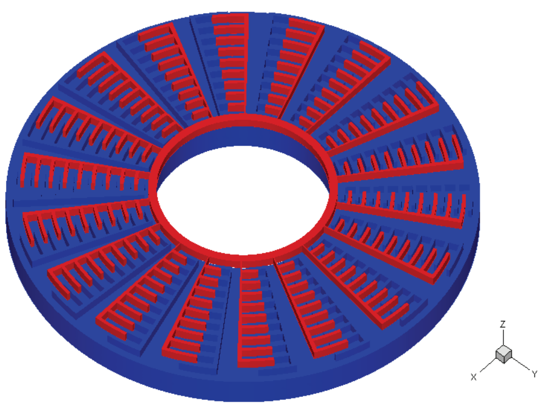

Figure 8: A torsional accelerometer MEMS model.

Figure 10 is a plot of the computed charge density on the device, which shows the converged result as compared with those using BEM models with smaller numbers of elements. The dual BIE (Liu and Shen 2007; Shen and Liu 2007) is used in solving this large BEM model with 696,486 DOFs on the Dell® Precision desktop PC. The CPU time is 4686 sec with the new fast multipole BEM with both the L2 modification and the adaptive tree, and is 7024 sec with the new fast multipole

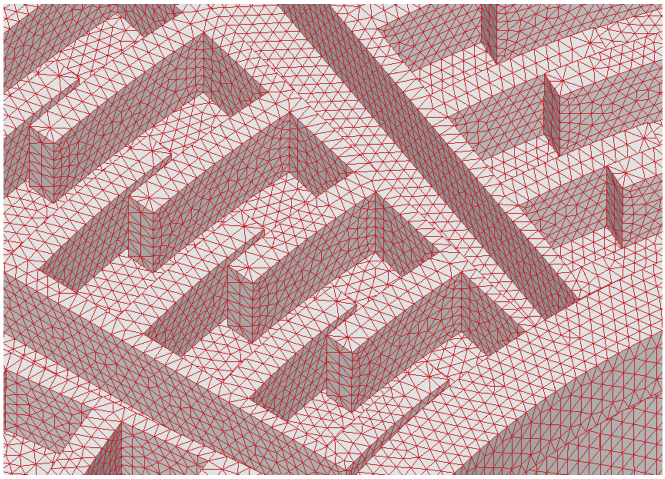

Figure 9: Mesh pattern for the torsional accelerometer MEMS model discretized with 696,486 boundary elements.

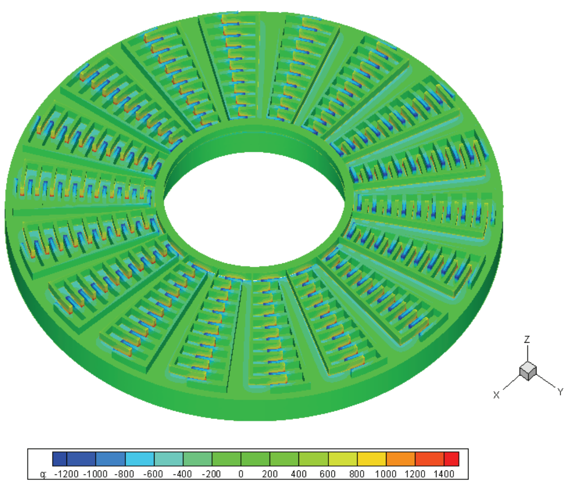

Figure 10: Computed charge density on the torsional accelerometer MEMS model.

BEM with the L2 modification only. The advantages of using the adaptive tree emerged for this large BEM model of a flat structure. Further investigation reveals that the number of the tree levels is 5 with the adaptive tree and is 7 with the L2 modification only version for this model. Therefore the numbers of the M2M, M2L and L2L translations are fewer with the adaptive tree version than those with the

L2 modification only version.

5.4 An Acoustic Radiation Model

As the last example, we use a 3-D acoustic wave problem to show that the new adaptive fast multipole algorithm can also be applied to improve the fast multipole BEM for solving other types of problems. In this case the baseline code used for comparison is the one that uses the $O(p^3)$ transformations introduced by Gumerov (Gumerov and Duraiswami 2003) in the fast multipole method for solving Helmholtz equations, where p is the expansion order. This code has been described in the Ref. (Bapat, Shen et al. 2009). See also Ref. (Brancati, Aliabadi et al. 2009) for a recent paper on comparison of the fast multipole BEM with the adaptive cross approximation method for solving 3-D acoustic problems.

The problem considered is the acoustic radiation from a slender box vibrating with a uniform boundary velocity. The dimensions of the box are 1x1x10. The value of the nondimensional wavenumber is 10 (wavenumber k = 1). The model with a BEM mesh is shown on the left in Figure 11. The model has an increasing number of elements and is solved on the Dell® Precision Desktop PC. In this case, 16 terms are used in the expansions, the maximum number of elements in a leaf is set to 50 and the tolerance for convergence is set to 10^{-5}.

The computed sound pressure on the boundary is shown on the right in Figure 11. The CPU time is shown in Figure 12. As predicted, the CPU time is observed to be the least when we use L2 modification in conjunction with the use of the adaptive tree. The L2 modification alone is seen to improve the efficiency of the code by 25-50% depending on the particular discretization. In addition, the adaptive tree structure is seen to have a reduction in the CPU time by about 15-20% depending on the model size.

6 Discussions

A new adaptive fast multipole algorithm based on a modified L2 list (or the interaction list), the four list adaptive algorithm and an adaptive tree structure is presented in this paper. This algorithm uses many of the existing efficient algorithms. Appropriate changes necessary to combine all the existing techniques have been discussed. This combination is faster than all the individual algorithms or modifications. The L2 modification, introduced in this paper, can consistently save a significant amount of CPU time in computing the M2L translations and the adaptive tree can potentially reduce the total CPU time for large models of slender and flat thin structures. This new adaptive fast multipole algorithm has been implemented in the fast multipole BEM code based on the new version of the FMM (using M2X,

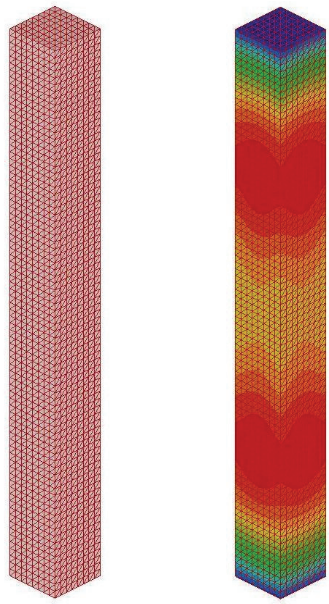

Figure 11: Acoustic radiation from a slender structure (the mesh and computed sound pressure).

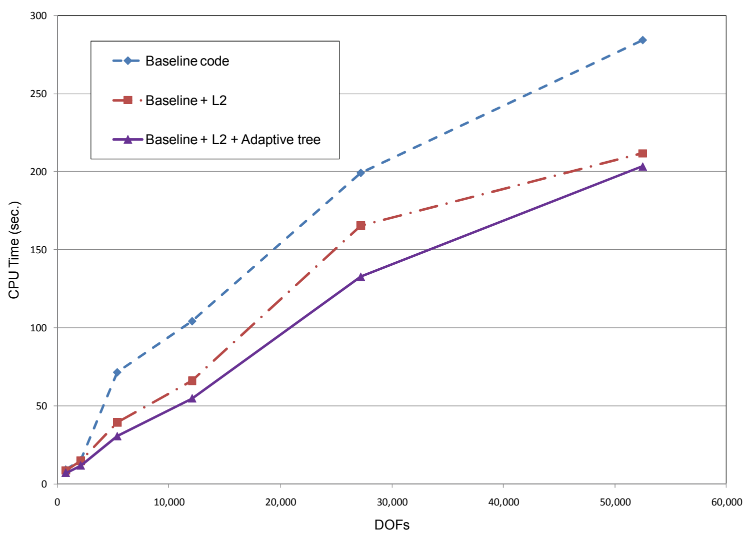

Figure 12: CPU time for the acoustic radiation model.

X2X, and X2L translations) for 3-D potential problems and an $O(p^3)$ code for 3-D acoustic wave problems.

Several numerical examples are presented to demonstrate the usefulness and efficiency of the proposed new adaptive fast multipole algorithm. About 30-40% improvements in the computational efficiency are achieved with the L2 modification. The adaptive tree, a promising data structure for the FMM, has also been investigated in this paper. Results with the adaptive tree structure are mixed. While the adaptive tree does not provide noticeable positive effects on the efficiency for smaller BEM models, it does show some advantages for larger BEM models, especially for MEMS models which are thin structures, and for acoustic problems. The implementation of L2 modification in existing codes is straightforward and it can be used to accelerate many other existing fast multipole BEM codes.

Acknowledgement: The support from US National Science Foundation through the grant CMS-0508232 is acknowledged. The authors also thank Mr. Bo Bi for his help in building the torsional accelerometer MEMS CAD model used in this work.

References

Anderson, R. J. (1999): Tree data structures for N-body simulation. *SIAM J. Comput.* 28: 1923–1940.

Aoki, S., K. Amaya, M. Urago, A. Nakayama (2004): Fast multipole boundary element analysis of corrosion problems. *CMES: Computer Modeling in Engineering and Sciences* 6(2): 123-132.

Bapat, M. S., L. Shen, Y. J. Liu (2009): Adaptive fast multipole boundary element method for three-dimensional half-space acoustic wave problems. *Engineering Analysis with Boundary Elements* 33(8-9): 1113-1123.

Brancati, A., M. H. Aliabadi, I. Benedetti (2009): Hierarchical adaptive cross approximation GMRES technique for solution of acoustic problems using the boundary element method. *CMES: Computer Modeling in Engineering & Sciences* 43(2): 149-172.

Chen, X. L., Y. J. Liu (2001): Thermal stress analysis of multi-layer thin films and coatings by an advanced boundary element method. *CMES: Computer Modeling in Engineering & Sciences* 2(3): 337-350.

Cheng, H., L. Greengard, V. Rokhlin (1999): A fast adaptive multipole algorithm in three dimensions. *Journal of Computational Physics* 155: 468-498.

Fu, Y., K. J. Klimkowski, G. J. Rodin, E. Berger, J. C. Browne, J. K., Singer,

R. A. V. D. Geijn, K. S. Vemaganti (1998): A fast solution method for three-dimensional many-particle problems of linear elasticity. *International Journal for Numerical Methods in Engineering* 42: 1215-1229.

Gomez, J. E., H. Power (1997): A multipole direct and indirect BEM for 2D cavity flow at low Reynolds number. *Engineering Analysis with Boundary Elements* 19: 17-31.

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

Greengard, L. F., V. Rokhlin (1987): A fast algorithm for particle simulations. *Journal of Computational Physics* 73(2): 325-348.

Gumerov, N. A., R. Duraiswami (2003): Recursions for the computation of multipole translation and rotation coefficients for the 3-D Helmholtz equation. *SIAM J. Sci. Comput.* 25(4): 1344-1381.

Liu, Y. J. (2005): A new fast multipole boundary element method for solving large-scale two-dimensional elastostatic problems. *International Journal for Numerical Methods in Engineering* 65(6): 863-881.

Liu, Y. J. (2008): A fast multipole boundary element method for 2-D multi-domain elastostatic problems based on a dual BIE formulation. *Computational Mechanics* 42(5): 761-773.

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

Liu, Y. J., H. Fan (2002): Analysis of thin piezoelectric solids by the boundary element method. *Computer Methods in Applied Mechanics and Engineering* 191(21-22): 2297-2315.

Liu, Y. J., N. Nishimura (2006): The fast multipole boundary element method for potential problems: a tutorial. *Engineering Analysis with Boundary Elements* 30(5): 371-381.

Liu, Y. J., N. Nishimura, Y. Otani (2005): Large-scale modeling of carbon-nanotube composites by the boundary element method based on a rigid-inclusion model. *Computational Materials Science* 34(2): 173-187

Liu, Y. J., N. Nishimura, Y. Otani, T. Takahashi, X. L. Chen, H. Munakata (2005): A fast boundary element method for the analysis of fiber-reinforced composites based on a rigid-inclusion model. *Journal of Applied Mechanics* 72(1): 115-128.

Liu, Y. J., N. Nishimura, D. Qian, N. Adachi, Y. Otani, V. Mokashi (2008): A boundary element method for the analysis of CNT/polymer composites with a cohesive interface model based on molecular dynamics. *Engineering Analysis with*

Boundary Elements 32(4): 299–308.

Liu, Y. J., L. Shen (2007): A dual BIE approach for large-scale modeling of 3-D electrostatic problems with the fast multipole boundary element method. *International Journal for Numerical Methods in Engineering* 71(7): 837–855.

Luo, J. F., Y. J. Liu, E. J. Berger (1998): Analysis of two-dimensional thin structures (from micro- to nano-scales) using the boundary element method. *Computational Mechanics* 22: 404-412.

Mammoli, A. A., M. S. Ingber (1999): Stokes flow around cylinders in a bounded two-dimensional domain using multipole-accelerated boundary element methods. *International Journal for Numerical Methods in Engineering* 44: 897-917.

Nishimura, N. (2002): Fast multipole accelerated boundary integral equation methods. *Applied Mechanics Reviews* 55(4 (July)): 299-324.

Nishimura, N., K. Yoshida, S. Kobayashi (1999): A fast multipole boundary integral equation method for crack problems in 3D. *Engineering Analysis with Boundary Elements* 23: 97-105.

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

Rokhlin, V. (1985): Rapid solution of integral equations of classical potential theory. *J. Comp. Phys.* 60: 187-207.

Shen, L., Y. J. Liu (2007): An adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton-Miller formulation. *Computational Mechanics* 40(3): 461-472.

Shen, L., Y. J. Liu (2007): An adaptive fast multipole boundary element method for three-dimensional potential problems. *Computational Mechanics* 39(6): 681-691.

Urago, M., T. Koyama, K. Takahashi, S. Saito and Y. Mochimaru (2003): Fast multipole boundary element method using the binary tree structure with tight bounds: application to a calculation of an electrostatic force for the manipulation of a metal micro particle. *Engineering Analysis with Boundary Elements* 27(8): 835-844.

Wang, H. T., G. Hall, S. Y. Yu, Z. H. Yao (2008): Numerical simulation of graphite properties using X-ray tomography and fast multipole boundary element method. *CMES: Computer Modeling in Engineering and Sciences* 37(2): 153-174.

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

Yarvin, N., V. Rokhlin (1998): Generalized Gaussian quadratures and singular value decompositions of integral operators. *SIAM J. Sci. Computing* 20: 699-718.

Yoshida, K. (2001): *Applications of Fast Multipole Method to Boundary Integral Equation Method.* Department of Global Environment Engineering. Kyoto, Japan, Kyoto University: 137.

Zhang, J., M. Tanaka (2007): Adaptive spatial decomposition in fast multipole method. *Journal of Computational Physics* 226(1): 17-28.