

A broadband fast multipole accelerated boundary element method for the three dimensional Helmholtz equation

Nail A. Gumerov^{a),b)} and Ramani Duraiswami^{b),c)}

*Perceptual Interfaces and Reality Laboratory, Institute for Advanced Computer Studies,
University of Maryland, College Park, Maryland 20742*

(Received 2 January 2008; revised 3 October 2008; accepted 10 October 2008)

The development of a fast multipole method (FMM) accelerated iterative solution of the boundary element method (BEM) for the Helmholtz equations in three dimensions is described. The FMM for the Helmholtz equation is significantly different for problems with low and high kD (where k is the wavenumber and D the domain size), and for large problems the method must be switched between levels of the hierarchy. The BEM requires several approximate computations (numerical quadrature, approximations of the boundary shapes using elements), and these errors must be balanced against approximations introduced by the FMM and the convergence criterion for iterative solution. These different errors must all be chosen in a way that, on the one hand, excess work is not done and, on the other, that the error achieved by the overall computation is acceptable. Details of translation operators for low and high kD , choice of representations, and BEM quadrature schemes, all consistent with these approximations, are described. A novel preconditioner using a low accuracy FMM accelerated solver as a right preconditioner is also described. Results of the developed solvers for large boundary value problems with $0.0001 \leq kD \leq 500$ are presented and shown to perform close to theoretical expectations. © 2009 Acoustical Society of America. [DOI: 10.1121/1.3021297]

PACS number(s): 43.55.Ka, 43.20.Fn, 43.28.Js [NX]

Pages: 191–205

I. INTRODUCTION

Boundary element methods (BEMs) have long been considered as a very promising technique for the solution of many problems in computational acoustics governed by the Helmholtz equation. They can handle complex shapes, lead to problems in boundary variables alone, and lead to simpler meshes where the boundary alone must be discretized rather than the entire domain. Despite these advantages, one issue that has impeded their widespread adoption is that they lead to linear systems with dense and possibly nonsymmetric matrices. As the domain size increases to many wavelengths, the number of variables in the discretized problem, N , should increase correspondingly to satisfy the Nyquist sampling criterion. For a problem with N unknowns, a direct solution requires $O(N^3)$ solution cost and storage of $O(N^2)$. Use of iterative methods does not reduce the memory but can reduce the cost to $O(N_{\text{iter}}N^2)$ operations, where N_{iter} is the number of iterations required, and the $O(N^2)$ per iteration cost arises from the dense matrix-vector product. In practice this is still quite large. An iteration strategy that minimizes N_{iter} is also needed. Other steps in the BEM are also expensive, such as the computation of the individual matrix elements, which require quadrature of nonsingular, weakly singular, or hyper-singular functions. To reduce the singularity order and achieve symmetric matrices, many investigators employ Galerkin techniques, which lead to further $O(N^2)$ integral computations. Because of these reasons, the BEM was not used for very large problems. In contrast finite-difference and

finite element methods, despite requiring larger volumetric discretizations, have well established iterative solvers and are more widely used.

The combination of the fast multipole method¹ (FMM) and the preconditioned Krylov iterative methods presents a promising approach to improving the scalability of BEMs and is an active area of research. The FMM for potential problems allows the matrix-vector product to be performed to a given precision ϵ in $O(N)$ operations and further does not require the computation or storage of all N^2 elements of the matrices, reducing the storage costs to $O(N)$ as well. Incorporating the fast matrix-vector product in a quickly convergent iterative scheme allows the system of equations to be rapidly solved with $O(N_{\text{iter}}N)$ cost. The FMM was initially developed for gravity or electrostatic potential problems. Later this method was intensively studied and extended to the solution of problems arising from the Helmholtz, Maxwell, biharmonic, elasticity, and other equations. While the literature and previous work on the FMM is extensive, reasons of space do not permit a complete discussion of the literature. The reader is referred to Ref. 2 for a comprehensive review.

II. FMM AND FMM ACCELERATED BEM FOR THE HELMHOLTZ EQUATION

The FMM and FMM accelerated BEM for the Helmholtz equation have seen significant work, and several authors have recently published on various aspects of the problem,^{3–9} and new articles have appeared when this article was in review.¹⁰ This article presents a FMM accelerated boundary element solver for the Helmholtz equation that has

^{a)}Electronic mail: gumerov@umiacs.umd.edu

^{b)}Also at Fantalgo, LLC, Elkridge, MD 20751.

^{c)}Electronic mail: ramani@umiacs.umd.edu

the following contributions that distinguish it from previous work:

- achieves good performance at both low and high frequencies by changing the representation used,
- has a quickly convergent iterative scheme via a novel preconditioner, which is described here,
- achieves further efficiency by considering the error of the quadrature in the BEM in the overall error analysis of the FMM, and
- uses the Burton–Miller (combined) boundary integral formulation for external problems and avoids the problems with spurious resonances.

A. Error and fast multipole accelerated boundary elements

During the approximate solution of the Helmholtz equation via the FMM accelerated BEM, the following different sources of error are encountered:

- geometric error due to the discretization of the surface with meshes,
- quadrature error in computation of boundary integrals,
- matrix-vector product error due to the FMM, and
- residue error used as a termination criterion during the iterative solution process.

One of the contributions of this paper is to consider all the errors together and design an algorithm that provides the required accuracy and avoids wasteful computations. It should be noted that the error achieved in practice in many reported simulation error tolerances is quite high, from a high of a few percent to at most 10^{-4} (see, e.g., Refs. 4, 5, and 11). Many previous FMM/BEM simulations contain computations that are wasteful of CPU time and memory when considering the final error desired or achieved.

B. Resolving calculations over a large range of wavenumbers

To be accurate, any calculation must resolve the smallest wavelengths of interest, and to satisfy the Nyquist criterion, the discretization must involve at least two points per wavelength. The restriction imposed by this requirement manifests itself at large frequencies since at lower frequencies the discretization is controlled by the necessity to accurately represent the boundary. Thus, two basic regimes for the FMM BEM are usually recognized in acoustic simulations: the low-frequency regime and the high-frequency regime. These regimes can be characterized by some threshold value $(kD)_*$ of the parameter kD , where k is the wavenumber and D is the computational domain size. For each of these regimes the computational complexity of the FMM exhibits a different behavior.¹²

1. Low-frequency regime

In the low-frequency regime, $kD < (kD)_*$, the per iteration step cost of the FMM is proportional to N and not very much affected by the value of kD . Here, the most efficient representation is in terms of spherical multipole wavefunc-

tions, and the translation schemes are based on the rotation-coaxial translation-backrotation (RCR) decompositions,^{12,13} which have $O(p^3)$ complexity. Here p^2 is the number of terms in the multipole expansion (p in this regime can be constant). Alternatively one may use the low-frequency exponential forms,^{14,15} which have the same complexity, but with a different asymptotic constant. The method of function representation based on sampling of the far-field signature function¹⁶ is not stable in this region due to exponential growth of terms in the multipole-to-local translation kernel.

2. High-frequency regime

In the high frequency regime, $kD > (kD)_*$, and the value of kD heavily affects the cost. Since the wavenumber k is inversely proportional to wavelength and, in practice, five to ten points per wavelength are required for accuracy, for a surface-based numerical method (such as the BEM) the problem size N scales as $O(kD)^2$, while for volumetric problems (e.g., for many scatterers distributed in a volume) N scales as $O(kD)^3$. For this regime, the size of the wavefunction representation, which is $O(p^2)$, must increase as the levels go up in the hierarchical space subdivision, with p proportional to the size of the boxes at a given level.¹² Because of this, the complexity of the FMM is heavily affected by the complexity of a single translation. It was shown¹² that $O(p^3)$ schemes (such as the RCR scheme) result in an overall complexity of the FMM $O(kD)^3$ for simple shapes and $O((kD)^3 \log(kD))$ for space-filling surfaces. The use of translation schemes of $O(p^4)$ and $O(p^5)$ complexities in this case provides the overall complexity of the FMM that is slower than the direct matrix-vector product. Where FMM with translation schemes of such complexity have been used with the BEM (e.g., see Ref. 7), one must recognize that the software is only usable in the low-frequency regime.

To reach the best scaling algorithm in the high-frequency regime, translation methods based on representations that sample the far-field signature function¹⁶ are necessary. The translation cost in this case scales as $O(p^2)$, while at least $O(p^2 \log p)$ additional operations are needed for the spherical filtering necessary for numerical stabilization of the procedure. In this case the overall FMM complexity will be $O((kD)^2 \log^\alpha(kD))$ ($\alpha \geq 1$) for simple shapes and $O((kD)^3)$ for space-filling shapes.

3. Switch in function representations

As discussed above, different representations are appropriate for low and high kD . However, even for high kD problems, since the FMM employs a hierarchical decomposition, at the fine levels, the problems behave as a low kD problem. Indeed at the fine levels, parameter ka , where a is a representative box size, is smaller than $(kD)_*$, and translations appropriate to the low-frequency regime should be used. For coarser levels, ka is large and the high-frequency regime should be used, and a combined scheme in which the spherical wavefunction representation can be converted to signature function sample representation is needed. Such a switch was also suggested and tested recently in Ref. 17. The

present scheme, however, is different from that of Ref. 17 and does not require interpolation/interpolation.

4. Comparison with volumetric methods

Similarly volumetric methods, such as the finite-difference time domain (FDTD) method for the wave equation, have the requirement that to resolve a simulation they need several points per wavelength. Thus for a problem on a domain of size kD , in three dimensions we have $N \sim (kD)^3$, with this restriction being controlling at higher frequencies. Fast iterative methods, e.g., based on multigrid solve these systems in constant number of iterations and for a cost of $O(N^{4/3})$ per time step. So for a problem with M time steps, we have $O(MN^{4/3}) = O(M(kD)^4)$ complexity for the volumetric methods (this is based on the discussion in Ref. 18). In contrast a fast multipole accelerated BEM, which is preconditioned with a preconditioner requiring a constant number of steps and is solved at M frequencies, can achieve solution in $O(M(kD)^2 \log(kD))$ or $O(M(kD)^3)$ steps. However, in terms of programming ease the FDTD and finite element TD methods are much easier to implement, and the preconditioners developed for them work better. Further they are easier to generalize to nonisotropic media. Their difficulty in handling infinite domains has been largely solved via perfectly matched layer methods. Accordingly, despite the advantages provided by integral equation approaches for the Helmholtz equation, volumetric approaches are popular.

C. Iterative methods and preconditioning

Preconditioning can be very beneficial for fast convergence of Krylov subspace iterative methods. Preconditioning for boundary element matrices is, in general, a lesser studied issue than for finite-element- and finite-difference-based discretizations. From that theory, it is known that for high wavenumbers preconditioning is difficult and an area of active research. Many conventional preconditioning strategies rely on sparsity in the matrix, and applying them to dense BEM matrices requires computations that have a formal time or memory complexity of $O(N^2)$, which negates the advantage of the FMM.

One strategy that has been applied with the FMBEM is the construction of approximate inverses for each row based on a local neighborhood of the row. If K neighboring elements are considered, then constructing this matrix has a cost of $O(NK^3)$, and there is a similar cost to applying the preconditioner at each step.^{3,4} However such local preconditioning strategies appear to work well only for low wavenumbers. Instead in this paper the use of a low accuracy FMM itself as a preconditioner by using a flexible generalized minimal residual (fGMRES) procedure¹⁹ is considered. This novel preconditioner appears to work reasonably at all wavenumbers considered and stays within the required cost.

III. FORMULATION AND PRELIMINARIES

A. Boundary value problem

Consider the Helmholtz equation for the complex valued potential ϕ ,

$$\nabla^2 \phi + k^2 \phi = 0, \quad (1)$$

with real wavenumber k inside or outside finite three dimensional (3D) domain V bounded by closed surface S , subject to mixed boundary conditions

$$\alpha(\mathbf{x}) \phi(\mathbf{x}) + \beta(\mathbf{x}) q(\mathbf{x}) = \gamma(\mathbf{x}), \quad q(\mathbf{x}) = \frac{\partial \phi}{\partial n}(\mathbf{x}), \quad (2)$$

$$|\alpha| + |\beta| \neq 0, \quad \mathbf{x} \in S.$$

Here and below all normal derivatives are taken, assuming that the normal to the surface is directed outward to V . For external problems ϕ is assumed to satisfy the Sommerfeld radiation condition

$$\lim_{r \rightarrow \infty} \left[r \left(\frac{\partial \phi}{\partial r} - ik \phi \right) \right] = 0, \quad r = |\mathbf{x}|. \quad (3)$$

This means that for scattering problems ϕ is treated as the scattered potential.

Note then that there should be some constraints on surface functions $\alpha(\mathbf{x})$, $\beta(\mathbf{x})$, and $\gamma(\mathbf{x})$ for existence and uniqueness of the solution. Particularly, if α and β are constant, this leads to the Robin problem, which degenerates to the Dirichlet or Neumann problem. For $\beta=0$ the case of a “sound-soft” boundary is obtained, and for $\alpha=0$ the “sound-hard” boundary case is obtained.

B. Boundary integral equations

The BEM uses a formulation in terms of boundary integral equations whose solution with the boundary conditions provides $\phi(\mathbf{x})$ and $q(\mathbf{x})$ on the boundary and subsequently determines $\phi(\mathbf{y})$ for any domain point \mathbf{y} . This can be done, e.g., using Green’s identity

$$\pm \phi(\mathbf{y}) = L[q] - M[\phi], \quad \mathbf{y} \notin S. \quad (4)$$

Here the upper sign in the left hand side should be taken for the internal domain, while the lower sign is for the external domain (this convention is used everywhere below), and L and M denote the following boundary operators:

$$L[q] = \int_S q(\mathbf{x}) G(\mathbf{x}, \mathbf{y}) dS(\mathbf{x}), \quad (5)$$

$$M[\phi] = \int_S \phi(\mathbf{x}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{x})} dS(\mathbf{x}),$$

where G is the free-space Green’s function for the Helmholtz equation

$$G(\mathbf{x}, \mathbf{y}) = \frac{e^{ikr}}{4\pi r}, \quad r = |\mathbf{x} - \mathbf{y}|. \quad (6)$$

In principle, Green’s identity can be also used to provide necessary equations for determination of the boundary values of $\phi(\mathbf{x})$ and $q(\mathbf{x})$, as in this case for smooth S one obtains

$$\pm \frac{1}{2} \phi(\mathbf{y}) = L[q] - M[\phi], \quad \mathbf{y} \in S, \quad (7)$$

The well-known deficiency of this formulation is related to possible degeneration of the operators L and $(M - \frac{1}{2})$ at cer-

tain frequencies depending on S , which correspond to resonances of the internal problem for sound-soft and sound-hard boundaries.^{20,21} Even though the solution of the external problem is unique for these frequencies, Eq. (7) is deficient in these cases. Moreover, for frequencies in the vicinity of the resonances, the system becomes poorly conditioned numerically. On the other hand, when solving internal problems (e.g., in room acoustics), the nonuniqueness of the solution for the internal problem has a physical meaning, as there are resonances.

In any case, Eq. (7) can be modified to avoid the artifact of degeneracy of boundary operators when solving the correctly posed problems (1)–(3). This can be done using different techniques, including direct and indirect formulations, introduction of some additional field points, etc. A direct formulation based on the integral equation combining Green's and Maue's identities, which is the method proposed by Burton and Miller²⁰ for sound-hard boundaries, is used. The Maue identity is

$$\pm \frac{1}{2}q(\mathbf{y}) = L'[q] - M'[\phi], \quad \mathbf{y} \in S, \quad (8)$$

where

$$L'[q] = \int_S q(\mathbf{x}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{y})} dS(\mathbf{x}), \quad (9)$$

$$M'[\phi] = \frac{\partial}{\partial n(\mathbf{y})} \int_S \phi(\mathbf{x}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{x})} dS(\mathbf{x}).$$

Multiplying Eq. (8) by some complex constant λ and summing with Eq. (7), one obtains

$$\pm \frac{1}{2}[\phi(\mathbf{y}) + \lambda q(\mathbf{y})] = (L + \lambda L')[q] - (M + \lambda M')[\phi]. \quad (10)$$

Burton and Miller²⁰ proved that it is sufficient to have $\text{Im}(\lambda) \neq 0$ to guarantee the uniqueness of the solution for the external problem.

C. Combined equation

Using the boundary conditions, the system of equations (2) and (10) can be reduced to a single linear system for some vector of elemental or nodal unknowns $[\psi]$,

$$A[\psi] = c, \quad (11)$$

which is convenient for computations. The boundary operator A and functions ψ and c can be constructed following standard BEM procedures. They can be expressed as

$$A[\psi] = (L + \lambda L')[u'] - (M + \lambda M')[u] \mp \frac{1}{2}(u + \lambda u'), \quad (12)$$

$$c = (L + \lambda L')[b'] - (M + \lambda M')[b] \mp \frac{1}{2}(b + \lambda b'),$$

where u and u' are related to the unknown ψ , and b and b' are related to the knowns. For example, for the Neumann problem $u = \phi$, $u' = 0$, while $b' = q$ and $b = 0$.

D. Discretization

Boundary discretization leads to approximation of boundary functions via finite vectors of their surface samples

and integral operators via matrices acting on these vectors. For example, if the surface is discretized by a mesh with M panels (elements), $S_{l'}$, and N vertices, \mathbf{x}_j , and integrals over the boundary elements are computed, one obtains

$$\begin{aligned} L[q](\mathbf{x}_l^{(c)}) &= \sum_{l'=1}^M \int_{S_{l'}} q(\mathbf{x}) G(\mathbf{x}, \mathbf{x}_l^{(c)}) dS(\mathbf{x}) \\ &\approx \sum_{l'=1}^M L_{ll'} q_{l'}, \quad l = 1, \dots, M, \end{aligned} \quad (13)$$

$$q_{l'} = q(\mathbf{x}_{l'}^{(c)}), \quad L_{ll'} = \int_{S_{l'}} G(\mathbf{x}, \mathbf{x}_l^{(c)}) dS(\mathbf{x}),$$

where $\mathbf{x}_{l'}^{(c)}$ is the center of the l' th element, and for computations of matrix entries $L_{ll'}$ one can use well-known quadratures, including those for singular integrals.^{21,22} The above equation is for the case of panel collocation, while analogous equations can be derived for vertex collocation. Similar formulas can be used for other operators. Note that to accurately capture the solution variation at the relevant length scales, the discretization should satisfy $kr_{\max} \ll 1$, where r_{\max} is the maximum size of the element. In practice, discretizations that provide several elements per wavelength usually achieve an accuracy consistent with the other errors of the BEM. The above method of discretization with collocation either at the panel centers or vertices was implemented and tested. Discretization of the boundary operators reduces problem (11) to a system of linear equations.

E. Iterative methods

Different iterative methods can be tried to solve Eq. (11), which has a nonsymmetric dense complex valued matrix A . Any iterative method requires computation of the matrix-vector product $A[x]$, where $[x]$ is some input vector. The method used in the present algorithm is the fGMRES method,¹⁹ which has the advantage that it allows use of approximate right preconditioner, which in its turn can be computed by executing of the internal iteration loop using unpreconditioned GMRES.²³ Choice of the preconditioning method must be achieved for a cost that is $O(N)$ or smaller. This flexibility is exploited in the present algorithm.

IV. USE OF THE FAST MULTIPOLE METHOD

The main idea of the use of the FMM for the solution of the discretized boundary integral equation is based on the decomposition of operator A ,

$$A = A_{\text{sparse}} + A_{\text{dense}}, \quad (14)$$

where the sparse part of the matrix has only nonzero entries A_{lj} corresponding to the vertices \mathbf{x}_l and \mathbf{x}_j , such that $|\mathbf{x}_l - \mathbf{x}_j| < r_c$, where r_c is some distance usually of the order of the distance between the vertices, whose selection can be based on some estimates or error bounds, while the dense part has nonzero entries A_{lj} for which $|\mathbf{x}_l - \mathbf{x}_j| \geq r_c$. The use of the FMM reduces the memory complexity of the overall product to $O(N)$ and the computational complexity to $o(N^2)$,

which can be $O(N)$, $O(N \log^\beta N)$, $\beta \geq 1$, or $O(N^\alpha)$, $\alpha < 2$, depending on the wavenumber, domain size, effective dimensionality of the boundary, and translation methods used.¹²

A. FMM strategy

The use of the FMM for the solution of boundary integral equations brings a substantial shift in the computational strategy. In the traditional BEM the full system matrix must be computed to solve the resulting linear system either directly or iteratively. The memory needed to store this matrix is fixed and is not affected by the accuracy imposed on the computation of the surface integrals. Even if one uses quadratures with a relatively high number of abscissas and weights to compute integrals over the flat panels in a constant panel approximation, the memory cost is the same, and the relative increase in the total cost is small, as that cost is dominated by the linear system solution.

If one chooses, as done by previous authors using the FMM accelerated BEM (see, e.g., Refs. 5 and 7), to compute nonsingular integrals very accurately in the FMM using expansions of Green's function, such as

$$G(\mathbf{x}, \mathbf{x}_l^{(c)}) = ik \sum_{n=0}^{\infty} \sum_{m=-n}^n R_n^{-m}(\mathbf{x} - \mathbf{x}_l^{(c)}) S_n^m(\mathbf{x}_l^{(c)} - \mathbf{x}_l^{(c)}), \quad (15)$$

where R_n^m and S_n^m are the spherical basis functions for the Helmholtz equation, then from Eq. (13) the expressions for the matrix entries may be obtained as

$$L_{ll'} = \sum_{n=0}^{\infty} \sum_{m=-n}^n C_n^m S_n^m(\mathbf{x}_l^{(c)} - \mathbf{x}_l^{(c)}), \quad (16)$$

$$C_n^m = ik \int_{S_{l'}} R_n^{-m}(\mathbf{x} - \mathbf{x}_l^{(c)}) dS(\mathbf{x}).$$

As the sum is truncated for maximum $n = p - 1$, then there are p^2 complex expansion coefficients per element. If this p is the same as the truncation number for the FMM, this requires substantial memory to store Mp^2 complex values.

A different strategy is proposed here. To reduce the memory consumption, one should use schemes where the integrals are computed at the time of the matrix-vector product and only at the necessary accuracy. In the case of the use of higher-order quadratures, one is faced then with a well-known dilemma to either compute integrals in the flat panel approximation with higher-order formulas or just increase the total number of nodes (discretization density) and use lower-order quadrature. In the case of use of the FMM with "on the fly" integral computations, the computational complexity will be almost the same for both ways, while the latter way seems preferable, as it allows the function to vary from point to point and employs better approximation for the boundary (as the vertices are located on the actual surface and variations of the surface normal are accounted for better).

Therefore, in the case of the use of the FMM, one can try to use the following approximation, at least in the far-field (the dense part), for the nonsingular integrals:

$$L_{lj} = s_j G(\mathbf{x}_j, \mathbf{x}_l), \quad M_{lj} = s_j \frac{\partial G}{\partial n_j}(\mathbf{x}_j, \mathbf{x}_l),$$

$$L'_{lj} = s_j \frac{\partial G}{\partial n_l}(\mathbf{x}_j, \mathbf{x}_l), \quad M'_{lj} = s_j \frac{\partial^2 G}{\partial n_l \partial n_j}(\mathbf{x}_j, \mathbf{x}_l), \quad (17)$$

$$l, j = 1, \dots, N, \quad \mathbf{x}_l \neq \mathbf{x}_j,$$

where in the case of panel collocation, s_j are the panel areas, \mathbf{x}_j are the centers of panels, and n_j is the normal to the panels. In the case of vertex collocation, these quantities are appropriately modified. For the treatment of the singular integrals ($\mathbf{x}_l = \mathbf{x}_j$), a method described later is used.

For near-field computations, these formulas could be used with a fine enough discretization for the nonsingular integrals, although one may prefer to use higher-order quadrature. Several tests, using for near-field integral representation Gauss quadratures of varying order (in the range of 1–625 nodes per element), showed that approximation (17) used for near field provides fairly good results for good meshes.

B. FMM algorithm

The Helmholtz FMM algorithm employed for matrix-vector products is described in Refs. 12 and 24, with modifications that allow use of different translation schemes for low and high frequencies. Particulars of the algorithm are that a level-dependent truncation number p_l is used and that rectangularly truncated translation operators are employed for multipole-to-multipole and local-to-local translations. These are performed using the RCR-decomposition and result in $O(p^3)$ single translation complexity. The RCR-decomposition is also used for the multipole-to-local translations for levels with $ka_l < (kD)_*$, where a_l is the radius of the circumsphere of a box on level l . For levels corresponding to $ka_l \geq (kD)_*$, the multipole expansions are converted to samples of the signature function at a cost of $O(p^3)$, and then diagonal forms of the translation operator $O(p^2)$ are used, and in the downward pass at some appropriate level conversion of the signature function to the local expansion of the required length at a cost of $O(p^3)$ is used. This procedure automatically provides filtering to ensure that the representation has the correct bandwidth. It must be noted that conversions from multipole to local expansions are required only once per box since consolidation of the translated functions is performed in terms of signature functions. This amortizes the $O(p^3)$ conversion cost and makes the scheme faster than the one based on the RCR-decomposition for the same accuracy. The algorithm, in this part, is thus close to the one described in Ref. 17. The difference is that interpolation/interpolation procedures are unnecessary here. Also, for low-frequency translation the RCR-decomposition for the multipole-to-local translation is used and found to be as efficient as the method based on conversion into exponential forms for moderate p . Particulars of the present implementation include a precomputation of all translation operators, particularly translation kernels, so during the run time of the

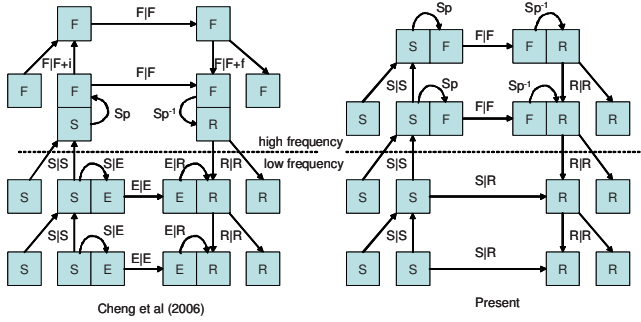


FIG. 1. (Color online) Illustration comparing the wideband FMMs of Cheng *et al.* (Ref. 17) and that presented in this paper for a problem in which the FMM octree has four levels and in which the high-low frequency switch threshold occurs between levels 2 and 3. The left hand side for each algorithm shows the FMM upward pass, while the right hand side shows the FMM downward pass. Each box represents various steps for that level, such as multipole expansion (S), local expansion (R), far-field signature function samples (F), and exponential form for each coordinate direction (E). The “glued” boxes mean that for a given box at that level the two types of expansions are constructed. $S|S$, $R|R$, $S|R$, $E|E$, and $F|F$ denote translation operators acting on the respective representations. \mathbf{Sp} and \mathbf{Sp}^{-1} denote forward and inverse spherical transform, $S|E$ and $E|R$ are the respective conversion operators. $F|F+i$ and $F|F+f$ mean that the translation is accompanied by use of an interpolation or filtering procedure.

procedure, which is performed many times for the iterative process, only simple arithmetic operations (additions and multiplications) are executed.

1. Comparison of algorithms

Figure 1 illustrates the present 3D Helmholtz FMM algorithm (on the right) and also compares it with that proposed in Ref. 17. These algorithms have in common the separation of the high- and low-frequency regions where different translation methods are used. It is seen that the present algorithm at high frequencies implements the idea used in the algorithm¹⁷ for lower frequencies, while instead of conversion to the exponential form the spherical transform is used to convert the multipole expansion to the signature function representation and back. The signature function representation is omnidirectional and, in contrast to the exponential forms, does not require additional data structures and multiple representations (since translations for this representation are different in each coordinate direction). Also, this approach is valid for the values of p are necessary for the high-frequency region. However, despite the use of these efficient techniques, the present algorithm has a formal translational scaling $O(p^3)$ since in the high-frequency region for the multipole-to-multipole $S|S$ and local-to-local $R|R$ operators the spherical function representations are used.

2. Data structure

The version of the FMM used in this paper employs an octree-based data structure, when the computational domain is enclosed into a cube of size $D \times D \times D$, which is assigned to level 0, and further the space is subdivided by the octree to the level l_{\max} . The algorithm works with cubes from level 2 to l_{\max} . For generation of the data structure, we use hierarchical box ordering based on the bit interleaving and pre-

compute lists of neighbors and children, which are stored and used as needed. The FMM used skips “empty” boxes at all levels.

3. Level-dependent truncation number

Each level is characterized by the size of the expansion domain, which is the radius a_l of the circumsphere of the boxes at level l . Selection of the truncation number in the algorithm is automated based on an expression of the form $p_l = p(ka_l, \epsilon, \delta)$, where ϵ is the prescribed accuracy and δ the separation parameter [we used $\delta=2$ (see the justification in Ref. 12)]. A detailed discussion and theoretical error bounds can be found elsewhere (see e.g., Refs. 12 and 25). Particularly, the following approximation combining low- and high-frequency asymptotics for monopole expansions can be utilized:¹²

$$p_{lo} = 1 - \frac{\log \epsilon (1 - \delta^{-1})^{3/2}}{\log \delta},$$

$$p_{hi} = ka + \frac{(3 \log 1/\epsilon)^{2/3}}{2} (ka)^{1/3}, \quad (18)$$

$$p = (p_{lo}^4 + p_{hi}^4)^{1/4}.$$

It is also shown in Ref. 12 that for the use of the rectangularly truncated translation operators the principal term of the error can be evaluated based on this dependence. The numerical experiments show that the theoretical bound frequently overestimates the actual errors, so some corrections can be also applied. The software developed here, in its automatic setting, computes p_{lo} and p_{hi} and—if it happens that $p - p_{hi} > p_*(\epsilon)$, where p_* is some number dictated by the overall accuracy requirements—uses $p = p_{hi} + p_*(\epsilon)$; otherwise, Eq. (18) is used. In fact, we also had some bound for $p_*(\epsilon)$ depending on ka to avoid blowout in computation of functions at extremely low ka .

As previously mentioned, an automatic switch was implemented from the RCR-decomposition to the diagonal forms of the translation operators based on criterion $ka_l \geq (kD)_*$. The parameter $(kD)_*$ was based on the error bounds (18) and was selected for the level at which $p - p_{hi} \leq p_{**}$ (we used $p_{**}=2$). This is dictated by the estimation of the threshold at which the magnitude of the smallest truncated term in the translation kernel (26) starts to grow exponentially (see Ref. 12).

Figure 2 illustrates the dependence provided by Eq. (18). We note that FMM with coarse accuracy like $\epsilon=10^{-2}$ can be used for efficient preconditioning. We also can remark that function representation via the multipole expansions and use of the matrix-based translations (such as RCR-decomposition) is not the only choice, and in Refs. 14 and 17 a method based on diagonalization of the translation operators, different from Ref. 16, was developed. This method, however, requires some complication in data structure (decomposition to the x -, y -, and z -directional lists) and is efficient for moderate to large truncation numbers. As we mentioned, the truncation numbers in the low-frequency region can be reduced (plus the BEM itself has a limited accuracy

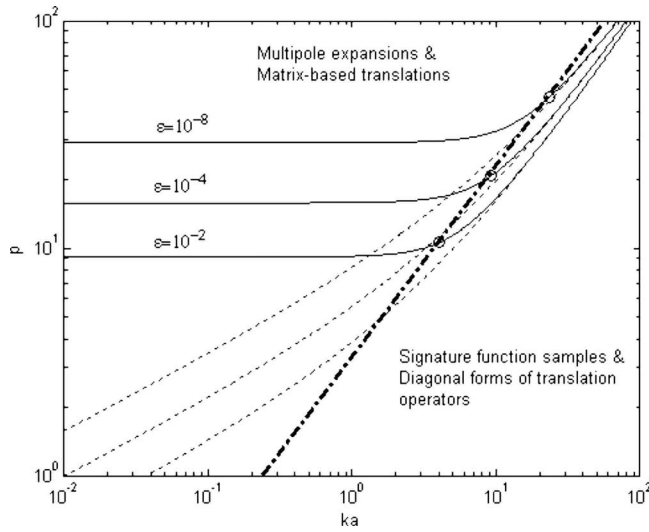


FIG. 2. Dependences of the truncation number p on the dimensionless domain size ka for different prescribed accuracies of the FMM ϵ according to Eq. (18) ($\delta=2$) (solid lines). The dashed lines show the high-frequency asymptotics $p_{hi}(ka)$. The circles mark the points of switch from function representation via multipole expansions to samples of the far-field signature function and, respectively, the translation method used. The dash-dotted line separates the (ka, p) region into the domains where different function representations are used.

due to flat panel discretization). In this case the efficiency of the matrix-based methods, such as the RCR-decomposition, has comparable or better efficiency. Indeed, function representations via the samples of the far-field signature functions are at least two times larger, which results in larger memory consumption and reduction of efficiency of operations on larger representing vectors.

4. Multipole expansions

Expansions over the singular (radiating) spherical basis functions $S_n^m(\mathbf{r})$ in forms (15) and (16) can be applied to represent the monopole source or respective integrals. In these formulas the singular and regular solutions of the Helmholtz equation are defined as

$$S_n^m(\mathbf{r}) = h_n(kr)Y_n^m(\theta, \varphi), \quad R_n^m(\mathbf{r}) = j_n(kr)Y_n^m(\theta, \varphi), \quad (19)$$

$$n = 0, 1, 2, \dots, \quad m = -n, \dots, n,$$

where in spherical coordinates $\mathbf{r} = r(\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$ symbols $h_n(kr)$ and $j_n(kr)$ denote spherical Hankel (first kind) and Bessel functions, and $Y_n^m(\theta, \varphi)$ the spherical harmonics,

$$Y_n^m(\theta, \varphi) = (-1)^m \sqrt{\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta) e^{im\varphi}, \quad (20)$$

$$n = 0, 1, 2, \dots, \quad m = -n, \dots, n,$$

and $P_n^{|m|}(\mu)$ are the associated Legendre functions consistent with that in Ref. 26, or Rodrigues' formulas

$$P_n^m(\mu) = (-1)^m (1 - \mu^2)^{m/2} \frac{d^m}{d\mu^m} P_n(\mu), \quad n \geq 0, \quad m \geq 0,$$

$$P_n(\mu) = \frac{1}{2^n n!} \frac{d^n}{d\mu^n} (\mu^2 - 1)^n, \quad n \geq 0, \quad (21)$$

where $P_n(\mu)$ are the Legendre polynomials.

In the boundary integral formulation also normal derivatives of Green's function should be expanded (or integrals of these functions over the boundary elements). These expansions can be obtained from expansions of the type of Eqs. (15) and (16) for the monopoles by applying appropriately truncated differential operators in the space of the expansion coefficients,¹² which are sparse matrices and so the cost of differentiation is $O(p^2)$. Indeed if $\{C_n^m\}$ are the expansion coefficients of some function $F(\mathbf{r})$ over basis $S_n^m(\mathbf{r})$, while $\{\hat{C}_n^m\}$ are the expansion coefficients over the same basis of function $\mathbf{n} \cdot \nabla F(\mathbf{r})$ for unit normal $\mathbf{n} = (n_x, n_y, n_z)$, then

$$\hat{C}_n^m = \frac{1}{2} [(n_x + in_y)(b_n^m C_{n-1}^{m+1} - b_{n+1}^{m-1} C_{n+1}^{m+1}) + (n_x - in_y)(b_n^{-m} C_{n-1}^{m-1} - b_{n+1}^{m-1} C_{n+1}^{m-1})] + n_z (a_n^m C_{n+1}^m - a_{n-1}^m C_{n-1}^m),$$

$$m = 0, \pm 1, \pm 2, \dots, \quad n = |m|, |m| + 1, \dots, \quad (22)$$

where a_n^m and b_n^m are the differentiation coefficients,

$$a_n^m = a_n^{-m} = \sqrt{\frac{(n+1+m)(n+1-m)}{(2n+1)(2n+3)}} \quad \text{for } n \geq |m|,$$

$$a_n^m = b_n^m = 0 \quad \text{for } n < |m|, \quad (23)$$

$$b_n^m = \sqrt{\frac{(n-m-1)(n-m)}{(2n-1)(2n+1)}} \quad \text{for } 0 \leq m \leq n,$$

$$b_n^m = -\sqrt{\frac{(n-m-1)(n-m)}{(2n-1)(2n+1)}} \quad \text{for } -n \leq m < 0.$$

5. Translations

Translations of the expansions can be also thought of as applications of matrices to the vectors of coefficients. If translation occurs from level l to l' ($l' = l - 1$ for the multipole to multipole, or $S|S$ -translation, $l' = l$ for the multipole to local, or $S|R$ -translation, and $l' = l + 1$ for the local to local, or $R|R$ -translation), then $p_{l'}$ translated coefficients relate to the p_l^2 original coefficient via the $p_{l'}^2 \times p_l^2$ matrix. Even for pre-computed and stored matrices, this requires $O(p^4)$ operations, which is unallowable cost for the translation if using with BEMs.¹² Several methods to reduce this cost are well known. Particularly use of the RCR-decomposition of the $(S|S)(\mathbf{t}) = (R|R)(\mathbf{t})$ matrices

$$(R|R)(\mathbf{t}) = \text{Rot}^{-1}(\mathbf{t}/t) (R|R)(t) \text{Rot}(\mathbf{t}/t), \quad (24)$$

where \mathbf{t} is the translation vector, $t = |\mathbf{t}|$, and $\text{Rot}(\mathbf{t}/t)$ is the rotation matrix—which expresses coefficients in the rotated reference frame, whose z -axis is collinear with \mathbf{t} , while $(R|R)(t)$ is the coaxial translation operator (along axis z)—reduces the cost of application of all operators to $O(p^3)$. As the geometry of the problem is specified, all these matrices can be precomputed for a cost of $O(p^3)$ operations using recursions^{13,12} and can be stored. We note also that the rect-

angular truncation operators $\text{Rot}(\mathbf{t}/t)$ and $\text{Rot}^{-1}(\mathbf{t}/t)$ act on the vectors of length p_l^2 and $p_l'^2$, respectively, and produce vectors of the same size, while $(\mathbf{R}|\mathbf{R})(t)$ acts on a vector of size p_l^2 and produces a vector of size $p_l'^2$. Therefore, there is no need for any interpolation or filtering as this is embedded into the decomposition. A similar decomposition is applied to the $(\mathbf{S}|\mathbf{R})(t)$ matrix for low frequencies, which provides a numerically stable low-frequency procedure [for levels corresponding to $ka_l < (kD)_*$].

For levels with $ka_l \geq (kD)_*$, we use the following decomposition of the translation matrix $(\mathbf{S}|\mathbf{R})(\mathbf{t})$:

$$(\mathbf{S}|\mathbf{R})(\mathbf{t}) = \mathbf{S}\mathbf{p}^{-1}\Lambda_s(\mathbf{t})\mathbf{S}\mathbf{p}, \quad (25)$$

where $\mathbf{S}\mathbf{p}$ can be thought of as a matrix of size $N_l \times p_l^2$, which performs transform of the expansion coefficients to N_l samples of the far-field signature function (spherical transform), $\Lambda_s(\mathbf{t})$ is a diagonal translation matrix of size $N_l \times N_l$, and $\mathbf{S}\mathbf{p}^{-1}$ is a matrix of size $p_l^2 \times N_l$, which provides a transform back to the space of the coefficients. The number of samples depends on the truncation number, and it is sufficient to use $N_l = (2p_l - 1)(4p_l - 3)$, where the grid is a Cartesian product of the $2p_l - 1$ Gauss quadrature abscissas with respect to the elevation angle $-1 \leq \mu = \cos \theta \leq 1$ and $4p_l - 3$ equispaced abscissas with respect to the azimuthal angle $0 \leq \varphi < 2\pi$. This grid also can be interpreted as a set of points on the unit sphere $\{\mathbf{s}_j\}$. The entries of the diagonal matrix $\Lambda_s(\mathbf{t})$ are

$$\Lambda_{jj}(\mathbf{t}) = \sum_{n=0}^{2p_l-2} i^n (2n+1) h_n(kt) P_n\left(\frac{\mathbf{s}_j \cdot \mathbf{t}}{t}\right), \quad j = 1, \dots, N_l, \quad (26)$$

which is a diagonal form of the translation operator.¹⁶ The bandwidth of this function, $2p_l - 2$, provides that decomposition (25) of the $p_l^2 \times p_l^2$ translation matrix $(\mathbf{S}|\mathbf{R})(\mathbf{t})$ is exact.¹² Note that for a given grid (which is the same for all translations at level l), the cost of computation of $\Lambda_s(\mathbf{t})$ for each translation vector \mathbf{t} is $O(p_l^3)$. In the present implementation, all these entries are precomputed and stored, so no computations of $\Lambda_s(\mathbf{t})$ are needed during the run part of the algorithm. The precomputation part may be sped up by employing a data structure, which eliminates computations of $\Lambda_{jj}(\mathbf{t})$ for repeated entries $\mathbf{s}_j \cdot \mathbf{t}/t$ and kt for all translations and, in fact, allows a substantial reduction in the cost of the present part of the algorithm.

The operator $\mathbf{S}\mathbf{p}$ can be decomposed into the Legendre transform with respect to $\mu = \cos \theta$ followed by the Fourier transform with respect to φ (e.g., see Refs. 12, 17, and 27). If performed in a straightforward way, each of them requires $O(p^3)$ operations. Despite the fact that there exist algorithms for fast Legendre transform and the fast Fourier transform (FFT) can be employed, which reduces the cost of application of operator $\mathbf{S}\mathbf{p}$ to $O(p^2 \log p)$ or so, for moderate p straightforward methods are much more efficient. Note that the major cost (about 90%) comes from the Fourier transform, so if the FFT is applied efficiently, this speeds up the

procedure. Furthermore, the operator $\mathbf{S}\mathbf{p}^{-1}$ can be decomposed into the inverse Fourier transform, diagonal matrix of the Legendre weights, and inverse Legendre transform. The cost of this procedure is the same as that for the computation of the forward transform.

As mentioned earlier, since the same transforms $\mathbf{S}\mathbf{p}$ and $\mathbf{S}\mathbf{p}^{-1}$ should be applied to all expansions at a given level, we can make Eq. (25) more efficient than the RCR-decomposition by first applying the transform $\mathbf{S}\mathbf{p}$ to all box expansions at a given level, then performing all diagonal translations and consolidations, and, finally, applying transform $\mathbf{S}\mathbf{p}^{-1}$ to all boxes.

6. Evaluation of expansions

Finally we mention that for the computation of the BEM operators L' and M' , the normal derivative of computed sums at the evaluation point should be taken. As the expansions are available for the sources outside the neighborhood of the evaluation points, this can be performed by the application of the differentiation operator in the coefficient space [see Eq. (22)].

7. Simultaneous matrix-vector products

For an efficient iterative solution of Eq. (10), the FMM can be used to compute in one run the sum of four matrix-vector products together,

$$\Sigma + \lambda \Sigma' = (L_{\text{dense}} + \lambda L'_{\text{dense}})[q] - (M_{\text{dense}} + \lambda M'_{\text{dense}})[\phi], \quad (27)$$

for input vectors q and ϕ . Also, if needed, results for the parts Σ and Σ' can be separated (e.g., for the application of Green's identity alone for the computation of the potential at internal domain points). The dense parts of the matrices correspond to decomposition (14) and, in the case of use of a simple scheme [Eq. (17)] are the matrices with eliminated diagonals.

V. COMPUTATION OF SINGULAR ELEMENTS

Despite the fact that there exist techniques for the computation of the integrals over the singular or nearly singular elements (e.g., with increasing number of nodes and element partitioning or using analytical or semianalytical formulas), these methods can be costly, and below we propose a technique for the approximation of such integrals, which is consistent with the use of the FMM and BEM. This technique is similar to the "simple solution" technique used by some authors to compute the diagonal elements for the BEM for potential problems and for elasticity,²⁸ except that it is updated with the use of the FMM, and to the case of the Helmholtz equation.

Let $\{\mathbf{x}_j\}$ be a set of points sampling the surface, and U_j^ϵ be a sphere of radius ϵ centered at \mathbf{x}_j and $S_j^\epsilon = S \cap U_j^\epsilon$. The surface operators can be decomposed as

$$\begin{aligned}
L[\sigma] &= \int_{S_j^\epsilon} \sigma(\mathbf{x}) G(\mathbf{x}, \mathbf{y}) dS(\mathbf{x}) \\
&+ \int_{S \setminus S_j^\epsilon(\mathbf{y}, \epsilon)} \sigma(\mathbf{x}) G(\mathbf{x}, \mathbf{y}) dS(\mathbf{x}) = L_j^\epsilon[\sigma] + \tilde{L}_j[\sigma], \\
M[\sigma] &= \int_{S_j^\epsilon} \sigma(\mathbf{x}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{x})} dS(\mathbf{x}) \\
&+ \int_{S \setminus S_j^\epsilon(\mathbf{y}, \epsilon)} \sigma(\mathbf{x}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{x})} dS(\mathbf{x}) = M_j^\epsilon[\sigma] + \tilde{M}_j[\sigma].
\end{aligned} \tag{28}$$

Note that for small enough ϵ we have the following approximations of the integrals:

$$L_j^\epsilon[\sigma] \approx \sigma_j l_j^\epsilon(\mathbf{y}), \quad M_j^\epsilon[\sigma] \approx \sigma_j m_j^\epsilon(\mathbf{y}), \tag{29}$$

where functions $l_j^\epsilon(\mathbf{y})$ and $m_j^\epsilon(\mathbf{y})$ are regular inside the domain. Thus, they can be approximated by a set of some basis functions, which satisfy the same Helmholtz equation. To construct such a set and approximation, consider Green's identity for a function, which is regular inside the finite domain (internal problem),

$$\gamma \psi = L \left[\frac{\partial \psi}{\partial n} \right] - M[\psi], \tag{30}$$

where $\gamma=1$ for points inside the domain, $\gamma=1/2$ for points on the boundary, and $\gamma=0$ for points outside the domain. Consider the following test functions:

$$\psi(\mathbf{x}) = e^{iks \cdot \mathbf{x}}, \tag{31}$$

$$q(\mathbf{x}) = \frac{\partial \psi}{\partial n}(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot \nabla e^{iks \cdot \mathbf{x}} = ik \mathbf{n}(\mathbf{x}) \cdot \mathbf{s} e^{iks \cdot \mathbf{x}}, \quad |\mathbf{s}| = 1,$$

which represent plane waves propagating in any direction \mathbf{s} . For these functions from Eqs. (28)–(30), one obtains

$$\begin{aligned}
m_j^\epsilon(\mathbf{y}) - ik(\mathbf{n}_j \cdot \mathbf{s}) l_j^\epsilon(\mathbf{y}) &= e^{-iks \cdot \mathbf{x}_j} \{ ik \tilde{L}'[(\mathbf{n} \cdot \mathbf{s}) e^{iks \cdot \mathbf{x}}] \\
&- \tilde{M}[e^{iks \cdot \mathbf{y}}] - \gamma(\mathbf{y}) e^{iks \cdot \mathbf{y}} \}.
\end{aligned} \tag{32}$$

Let $\mathbf{s}_1, \dots, \mathbf{s}_4$ be four different unit vectors provided that functions $e^{iks \cdot \mathbf{x}}$, $\alpha=1, \dots, 4$, are linearly independent. Then denoting

$$\begin{aligned}
\omega_{j\alpha}(\mathbf{y}) &= e^{iks \cdot \mathbf{x}_j} \{ ik \tilde{L}'[(\mathbf{n} \cdot \mathbf{s}_\alpha) e^{iks \cdot \mathbf{x}}] - \tilde{M}[e^{iks \cdot \mathbf{y}}] \\
&- \gamma(\mathbf{y}) e^{iks \cdot \mathbf{y}} \}, \quad n_{j\alpha} = \mathbf{n}_j \cdot \mathbf{s}_\alpha,
\end{aligned} \tag{33}$$

one obtains

$$m_j^\epsilon(\mathbf{y}) - ik n_{j\alpha} l_j^\epsilon(\mathbf{y}) = \omega_{j\alpha}(\mathbf{y}), \quad \alpha = 1, \dots, 4. \tag{34}$$

Obviously the surface operators $L'[\sigma]$ and $M'[\sigma]$ can be similarly decomposed as

$$L'[\sigma] = L_j'^\epsilon[\sigma] + \tilde{L}'_j[\sigma], \quad M'[\sigma] = M_j'^\epsilon[\sigma] + \tilde{M}'_j[\sigma], \tag{35}$$

$$L_j'^\epsilon[\sigma] \approx \sigma_j l_j'^\epsilon(\mathbf{y}), \quad M_j'^\epsilon[\sigma] \approx \sigma_j m_j'^\epsilon(\mathbf{y}).$$

Note that these operators are employed only for points on the boundary, and the forms for internal and external problems are either the same or related via a sign change. Thus one can use Maue's identity (8) for the internal problem. In this case using test functions (31), Eqs. (32)–(34) can be modified as follows:

$$m_j'^\epsilon(\mathbf{y}) - ik n_{j\alpha} l_j'^\epsilon(\mathbf{y}) = \omega_{j\alpha}'(\mathbf{y}), \quad \alpha = 1, \dots, 4, \tag{36}$$

$$\begin{aligned}
\omega_{j\alpha}'(\mathbf{y}) &= e^{-iks \cdot \mathbf{x}_j} \{ ik \tilde{L}'[(\mathbf{n} \cdot \mathbf{s}_\alpha) e^{iks \cdot \mathbf{x}}] - \tilde{M}'[e^{iks \cdot \mathbf{x}}] \\
&- \gamma_\alpha(\mathbf{y}) e^{iks \cdot \mathbf{y}} \},
\end{aligned}$$

$$\gamma_\alpha(\mathbf{y}) = \frac{1}{2} ik(\mathbf{n}(\mathbf{y}) \cdot \mathbf{s}_\alpha), \quad \mathbf{y} \in S.$$

Then for the chosen set of directions, solution will be provided by the previous equations with $m_j^\epsilon(\mathbf{y})$, $l_j^\epsilon(\mathbf{y})$, and $\omega_{j\alpha}(\mathbf{y})$, replaced with $m_j'^\epsilon(\mathbf{y})$, $l_j'^\epsilon(\mathbf{y})$, and $\omega_{j\alpha}'(\mathbf{y})$, respectively.

The small 4×2 linear systems for each [Eqs. (34) and (36)] can be solved via least squares, and as noted above the FMM provides four simultaneous matrix-vector multiplications, and so four matrix-vector products via the FMM ($\alpha = 1, \dots, 4$) are sufficient to get all diagonals.

A. Discretization

The above formulas obviously provide expressions for the diagonal elements of matrices

$$L_{jj} = l_j^\epsilon(\mathbf{x}_j), \quad M_{jj} = m_j^\epsilon(\mathbf{x}_j), \quad L_{jj}' = l_j'^\epsilon(\mathbf{x}_j), \tag{37}$$

$$M_{jj}' = m_j'^\epsilon(\mathbf{x}_j), \quad j = 1, \dots, N.$$

In fact, for the solution of the boundary integral equation, only quantities $L_{jj} + \lambda L_{jj}'$ and $M_{jj} + \lambda M_{jj}'$ are needed. So for given λ the storage can be reduced twice. Also combinations $L_{jj} + \lambda L_{jj}'$ and $M_{jj} + \lambda M_{jj}'$ can be computed instead using the same method as described above.

VI. NUMERICAL EXPERIMENTS

The BEM/FMM was implemented in FORTRAN 95 and was parallelized for symmetric multiprocessing architectures, such as modern multicore PCs, using OpenMP. The results reported below were obtained on a four core PC [Intel Core 2 Extreme QX6700 2.66 GHz processor with 2×4 Mbytes L2 Cache and 8 Gbytes random access memory (RAM)] running Windows XP-64. Parallelization requires the replication of data among threads and is controlled by the size of the cache. In this implementation, to reduce the size of the stack only the sparse matrix computations and the $S|R$ -translations using the RCR-decomposition (i.e., operators employed for levels finer than the high-low-frequency switch level) were parallelized. Such a parallelization strategy was found to be efficient enough (overall parallelization efficiency was about 80%–95%) and enabled computations for kD 's up to 500. For $kD < 200$ more data can be placed on the available stack, and a complete parallelization with effi-

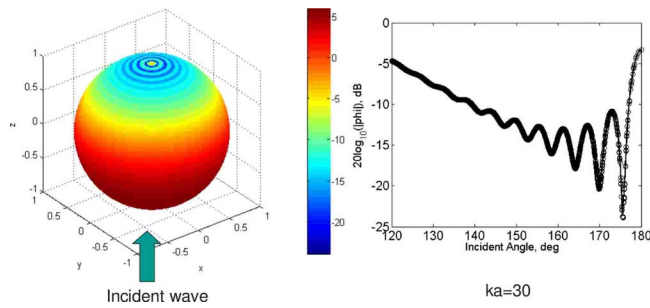


FIG. 3. (Color online) Typical BEM computations of the scattering problem. The graph shows comparison between the analytical solution (38) and BEM solution for the vicinity of the rear point of the sphere for $ka=30$.

ciency close to 100% was obtained. However, the results reported here use the partial parallelization to enable a uniform comparison of the results obtained over a wide range of kD 's.

A. Scattering from a single sphere

The example of an incident plane wave off a single sphere is valuable for tests of the performance of the method since an analytical solution is available in this case. For the incident field $\phi^{\text{in}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}}$, where \mathbf{s} is the unit vector collinear with the wave vector, the solution for the total (incident plus scattered field) for impedance boundary conditions can be found elsewhere, e.g., in Ref. 12,

$$\phi|_S(\mathbf{s}') = \frac{i}{(ka)^2} \sum_{n=0}^{\infty} \frac{(2n+1)i^n P_n(\mathbf{s} \cdot \mathbf{s}')}{h'_n(ka) + (i\sigma/k)h_n(ka)}, \quad \left. \frac{\partial \phi}{\partial n} + i\sigma\phi \right|_S = 0, \quad (38)$$

where a is the sphere radius, \mathbf{s}' is a unit vector pointing to the location of the evaluation surface point, and σ is the boundary admittance, which is zero for sound-hard surfaces and infinity for sound-soft surfaces. Depending on this, one may have for the scattered field a Neumann, Dirichlet, or Robin problem.

The following were varied in the simulations: k , discretization, parameter λ in Eq. (10), the boundary admittance, tolerance, and parameters controlling the FMM accuracy and performance. A typical computational result is shown in Fig. 3.

1. Preconditioning

The best preconditioners found were right preconditioners that compute the solution of the system $A\psi_j = c_j$ at the j th step using unpreconditioned GMRES (inner loop) and low accuracy FMM with lower bounds for convergence of iterations. For example, if in the outer loop of the fGMRES the prescribed accuracy for the FMM solution was 10^{-4} (actual achieved accuracy was $\sim 10^{-6}$) and the iterative process was terminated when the residual reaches 10^{-4} , for the inner preconditioning loop we used FMM with prescribed accuracy of 0.2 (actual accuracy was $\sim 5 \times 10^{-3}$) and the iterative process was terminated at a residual value of 0.45. The inner loop solution was much faster than the outer loop, as it is stopped after fewer iterations, and the matrix-vector product was sev-

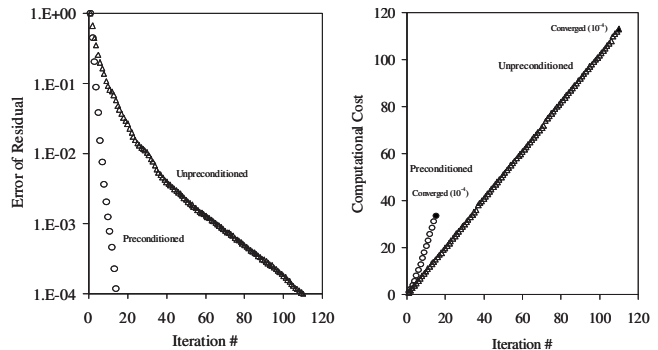


FIG. 4. Left: The absolute error in the residual in the unpreconditioned GMRES (triangles) and in the preconditioned fGMRES (circles) as a function of the number of iterations (outer loop for the fGMRES). Right: the relative computational cost to achieve the same error in the residual for these methods (1 cost unit=1 iteration using the unpreconditioned method). Computations for sphere; $ka=50$ for mesh with 101 402 vertices and 202 808 elements, and $\lambda=6 \times 10^{-4}i$.

eral times faster (lower truncation numbers). The preconditioner reduced by an order of magnitude the number of iterations in the outer loop, more than compensating for the increased cost of inner iterations. Beyond the benefit of improved time, this also has the benefit of reduced memory for large problems. This is because in the GMRES or fGMRES K vectors of length N must be stored, where K is the dimensionality of the Krylov subspace. The iterative process becomes slower if K is restricted, and GMRES is restarted when K vectors have been stored. Hence it is preferable to achieve convergence in $N_{\text{iter}} \leq K$ iterations. In the case of the GMRES-based preconditioner, the storage memory is of order $(K+K')N$, where K' is the dimensionality of the Krylov subspace for preconditioning. Since both K and K' are much smaller than the K required for unpreconditioned GMRES, the required memory for solution is reduced substantially.

Figure 4 shows the convergence of the unpreconditioned GMRES and the preconditioned fGMRES with the FMM-based preconditioner as described above. The computations were made for a sound-hard sphere of radius a , whose surface was discretized by 101 402 vertices and 202 808 triangular elements for the relative wavenumber $ka=50$. The preconditioned iteration converges three times faster in terms of the time required. The matrix-vector product via the low accuracy FMM in preconditioning was approximately seven times faster than that computed with higher accuracy in the outer loop (1.4 s versus 9.75 s). The overall error of the obtained solution was slightly below 5×10^{-4} in the relative L_2 -norm.

2. Spurious modes

The test case for the sphere is also good to illustrate the advantage of the Burton–Miller formulation over the formulation based on Green's identity. According to theory,^{20,29} the Green's identity formulation may result in convergence to a solution, which is the true solution plus a nonzero solution of the internal problem corresponding to zero boundary conditions at the given wavenumber. Such solutions are not physical since the solution of the external scattering problem is

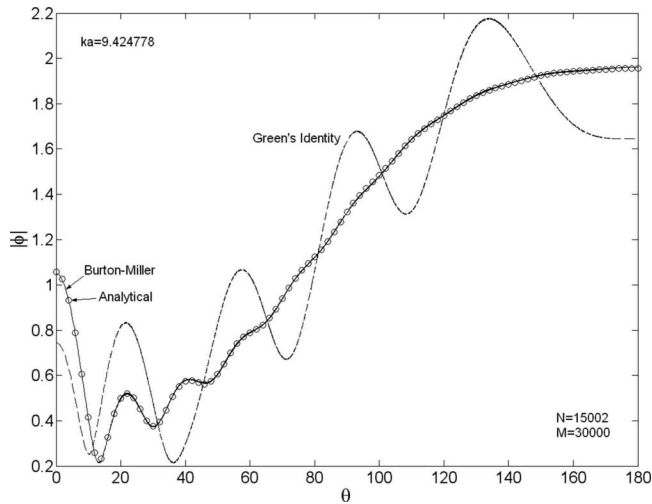


FIG. 5. Solutions of the plane wave scattering problem obtained using BEM with the Burton–Miller and Green’s identity formulations and fGMRES iterator for a sphere at resonance $ka=9.424\,778$ (triangular mesh of 15 002 vertices and 30 000 elements). Analytical solution is shown by the circles.

unique, and, therefore, they manifest a deficiency of the numerical method based on the Green’s identity.

For a sphere any internal solution can be written in the form

$$\phi^{\text{int}}(\mathbf{r}) = \sum_{n=0}^{\infty} j_n(kr) \sum_{m=-n}^n B_n^m Y_n^m(\theta, \varphi), \quad (39)$$

where (r, θ, φ) are the spherical coordinates of \mathbf{r} and B_n^m are arbitrary constants. The set of zeros of the functions $j_n(ka)$ provides a discrete set of values of ka for which $\phi^{\text{int}}|_S=0$, even though ϕ^{int} is not identically zero inside the sphere. The minimum resonant k can be obtained from the first zero of $j_0(ka)$, which is $ka=\pi$. We conducted some numerical tests with Burton–Miller and Green’s formulations for a range of ka ($0.01 \leq ka \leq 50$) for different resonant values of ka .

Figure 5 provides an illustration for case $ka=3\pi \approx 9.424\,778$, which is the third zero of $j_0(ka)$. When the Burton–Miller formulation is used with some λ with $\text{Im}(\lambda) \neq 0$, a solution consistent with Eq. (38) is obtained, while when computations are performed using Green’s formula a wrong solution with the additional spurious modes is achieved. We checked that in this case the converged solution can be well approximated ($\sigma=0$) by

$$\phi|_S(\mathbf{s}') = B + \frac{i}{(ka)^2} \sum_{n=1}^{\infty} \frac{(2n+1)i^n P_n(\mathbf{s} \cdot \mathbf{s}')}{h'_n(ka)}, \quad (40)$$

where B is some complex constant depending on the initial guess in the iterative process. This shows that, in fact, the zero-order harmonic of the solution, corresponding to the resonating eigenfunction, failed to be determined correctly, which is the expected result. Note that such solutions appear when iterative methods are used, where degeneration of the matrix operator for some subspace does not affect convergence in the other subspaces. If the problem is solved directly, the degeneracy or poor conditioning of the system matrix would result in a completely wrong solution.

In the nonresonant cases, the Green’s formulation provided a good solution, while normally the number of iterations was observed to increase as ka comes closer to a resonance. For large ka (40 or more), computations using Green’s identity become unstable, which is explainable by density of the zeros of $j_n(ka)$. For the Burton–Miller formulation, the empirically found value of the parameter λ was

$$\lambda = \frac{i\eta}{k}, \quad \eta \in [0.01, 1]. \quad (41)$$

For the case above $\eta=0.03$ provides good results, while for $ka < 1$ the number of iterations increases compared to the Green’s identity, and when there are certainly no resonances, Green’s identity can be recommended. Increase in this parameter usually decreases the accuracy of computations since more weight is put on the hypersingular part of the integral equation, while decrease in the parameter for large ka leads to an increase in the number of iterations. For $\eta \ll 0.01$ the Burton–Miller integral equation shows the problems of spurious modes.

3. Performance

For the FMM, the characteristic scale is usually based on the diagonal (maximum size) of the computational domain, D , and kD is an important dimensionless parameter. Further, one can compute the maximum size of the boundary element, which for a triangular mesh is the maximum side of the triangle, d . This produces another dimensionless parameter, kd . For a fixed body of surface area $S \sim D^2$, the number of elements in the mesh is of the order $N \sim S/d^2 \sim D^2/d^2$. A formal constraint for discretizations used for an accurate solution of the Helmholtz equation is $d/\lambda_a \ll 1$, where $\lambda_a = 2\pi/k$ is the acoustic wavelength. In practice this condition can be replaced with $kd < \chi$, where χ is some constant of order 1, so there are not less than $2\pi/\chi$ mesh elements per wavelength. This number usually varies in range of 5–10. This shows that the total number of elements should be at least $N \sim D^2/d^2 \gtrsim (kD)^2/\chi^2 = O((kD)^2)$.

Figure 6 shows the results of numerical experiments for scattering from a sound-hard sphere, where we fixed parameter $\chi \approx 0.94$ for $kD \geq 20$, which provided at least six elements per wavelength and increased the number of elements proportionally to $(kD)^2$ (the maximum case plotted corresponds to $kD=500$ where the mesh contains 1 500 002 vertices and 3 000 000 elements). Additional data on some tests showing the relative error in the L_2 -norm, ϵ_2 , and peak memory required for the problem solution for the current implementation are given in Table I. Note that in all cases the relative error in the L_∞ -norm was less or about 1%. For $kD < 3$, which we characterize as very low-frequency regime, we used a constant mesh with 866 vertices and 1728 elements. Also for this range λ was set to zero, while for other cases λ was computed using Eq. (41) with $\eta=0.03$. The time and memory required for computations substantially depend on the accuracy of the FMM, tolerance for convergence, and settings for the preconditioner. The prescribed FMM error for the outer loop was 10^{-4} , which, in fact, provides accuracy about 10^{-6} in the L_2 -norm. This was found from the tests,

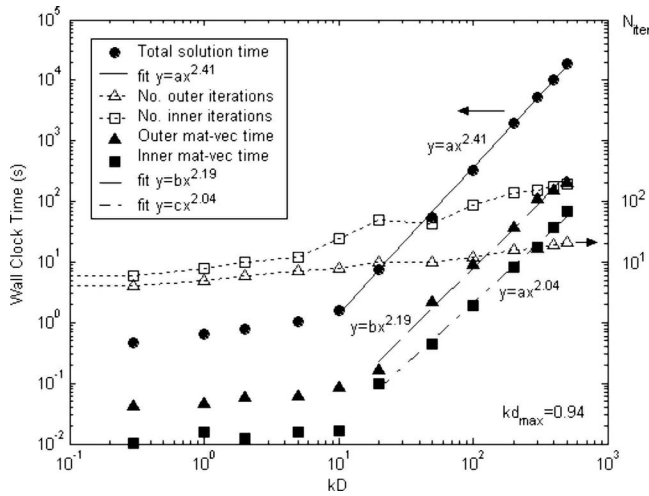


FIG. 6. Wall clock time and the number of iterations for solution of scattering problem from a sound-hard sphere in range $0.1 \leq kd \leq 500$. The curves that fit the time to the data at high frequencies were obtained using least squares.

where the actual FMM error was computed by comparisons with the straightforward (“exact”) method at 100 check points. The prescribed accuracy for the preconditioner was 4.5×10^{-2} , which also substantially overestimates the actual accuracy. Tolerance for convergence for the outer loop was 10^{-4} , while that for the inner loop was 0.45, where the number of iterations was limited by 11. The memory required depends on the dimensionalities of the Krylov subspaces for the outer and inner loops, which were set to 35 and 11, respectively, while the convergence for all cases was within 20 outer iterations, so the memory, in fact, can be reduced. Additional acceleration can be obtained almost for all cases by the storage of the near-field data [matrix A_{sparse} , Eq. (14)]. However, based on the RAM (8 Gbytes) this works only for $N < 5 \times 10^5$. To show a scaling without jumps, such a storage was not used, and the sparse matrix entries were recomputed each time as the respective matrix-product was needed.

a. Computational complexity scaling. The total time required for the solution of the problem is scaled approximately as $O((kd)^{2.4})$, while the times for the matrix-vector multiplication in the outer and in the inner iteration loops of the preconditioned fGMRES scale, respectively, as $O((kd)^{2.2})$ and $O((kd)^{2.04})$. A standard least squares fit to the data at $kd > 10$ was used.

These results require some analysis and explanation since the theoretical expectation of the matrix-vector complexity for the current version of the FMM is $O((kd)^3)$. There are several different reasons that the algorithm is scaled better. First, we note that the complexity of the sparse matrix-vector product, which in a well-balanced algorithm

takes a substantial part of the computational time, is scaled as $O(N)$, i.e., as $O((kd)^2)$. Second, for high-frequency computations a switch between the high-frequency and low-frequency representations always happens. As soon as the low-frequency part is limited by condition $ka_l < (kd)_*$, where $(kd)_*$ depends on the prescribed accuracy only, the computational cost for this part becomes proportional to the number of occupied boxes in the low-frequency regime. This number is $O(B_{l_{\text{max}}})$, where $B_{l_{\text{max}}}$ is the number of occupied boxes at the finest level l_{max} , which for simple shapes is $O(N) = O((kd)^2)$. Therefore, the low-frequency part for the fixed prescribed accuracy can be executed in $O((kd)^2)$ time. Therefore, all the $O((kd)^3)$ scaling comes only from the high-frequency part. However, here the most expensive part related to the $S|R$ -translations is performed using the diagonal forms with complexity $O((kd)^2)$, while the asymptotic constant related to the $S|S$ - and $R|R$ -translations and spherical transform is relatively small. Profiling of the algorithm shows that sparse matrix-vector multiplication and $S|R$ -translations in the low-frequency part contribute up to 90% to the complexity in the range of kd 's studied. Adding all these facts together, we can see that the FMM for problems solved is rather the $O((kd)^2)$ algorithm, and small addition to power 2 is due to $O((kd)^3)$ operations with much smaller asymptotic constant.

According to the above discussion, the complexity of the FMM can be estimated as

$$\text{Time} = A(kd)^3 + B(kd)^2, \quad (42)$$

where A and B are some constants. The least squares fit of high-frequency data on the matrix-vector multiplication in the outer loop with Eq. (42) is as good as fit $O((kd)^{2.19})$, shown in Fig. 6, and provides $B/A = 1.39 \times 10^3$. This number, indeed, is large, and it actually shows that only at $kd \gg 10^3$ the high-frequency asymptotical behavior can be reached for the prescribed accuracy of 10^{-4} . For the low accuracy preconditioner used, this ratio becomes even smaller ($B/A = 1.75 \times 10^3$), which justifies its applicability even for higher kd 's.

The overall algorithm scaling as $O((kd)^{2.4})$ now is not surprising since the factor $O((kd)^{0.2})$ can be ascribed to the (slowly) increasing number of iterations. Indeed, Fig. 6 shows that the number of iterations in the outer loop grows slowly.

b. Memory complexity scaling. Theoretically, the memory required to solve the problem should be scaled as $O(N) = O((kd)^2)$. Due to different possibilities for memory management and optimizations, plus realization of trade-off option memory versus speed and limited resources, the actual scaling can deviate from the theoretical one. Table I

TABLE I. Error, memory, and FMM-BEM solution wall clock time for some test cases with a spherical scatterer with known analytical solution.

kd	0.0001	0.01	1	20	50	100	200	300	400	500
N	866	866	866	2402	1.5×10^4	6×10^4	2.4×10^5	5.4×10^5	9.6×10^5	1.5×10^6
ϵ_2	3.3×10^{-5}	3.3×10^{-5}	1.1×10^{-4}	5.4×10^{-3}	2.25×10^{-4}	4.1×10^{-4}	5.34×10^{-4}	3.6×10^{-4}	4.92×10^{-4}	3.2×10^{-3}
Memory, Mbyte	20	20	32	63	268	461	1144	2861	3755	4748
Time, s	0.438	0.406	0.656	7.59	53.6	327	1990	5150	10 400	19 100

TABLE II. Results from some recent FMM and FMM accelerated BEM publications operating in the high-frequency regime. The performance of the FMM and the FMM-BEM reported in this paper is comparable or superior (see Tables I and IV). Data not reported are indicated with a “–”; for the FMM only (Ref. 17) the values reported can be considered as those applying to one iteration of the solution procedure.

Ref.	kD	N	$N/(kD)^2$	Memory	Error	Time	Iteration	Remarks
10	435	1 046 528	5.53	4800	$>10^{-5}$	54 267	–	Scattering
9	126.64	98 304	6.13	1487	0.03	24 325	59	Internal
9	126.64	49 152	3.06	799.6	0.03	11 848	58	Internal
17	544	646 143	2.09	549	10^{-3}	672	n/a	FMM only
17	544	619 520	2.09	1111	10^{-6}	1832	n/a	FMM only

shows that in the implementation used for this paper, the memory required grows slower than $O((kD)^2)$ in the range studied. Certainly, it should be scaled not less than $O((kD)^2)$ at larger kD since at least several vectors of size N should be stored. The table provides some data, which show that computation of a million size problem with $kD \sim 500$ is reachable on desktop PCs or workstations.

c. Comparison with recent publications. As discussed previously two regimes can be recognized in FMM-BEM simulations, the low and high-frequency regimes. Tables II and III indicate some results from papers published in the past few years. The data here should be compared with those presented in Table I where data for the present code are given. Looking first at the high-frequency regime, there appear to be three relevant papers (Refs. 9, 10, and 17). Reference 17 only has a FMM matrix-vector product, and a detailed comparison of the present algorithm and the one of that paper was provided in Sec. IV. As mentioned previously, the size of the discretization N in a BEM simulation should be at least $O(kD)^2$, although more points may be necessary if the geometry is complex. In each case the comparison is quite favorable for the present code.

The results of the best performing FMM BEM solver¹⁰ which appeared during the time when this paper was under review, is compared with the present work. In this paper the authors provide data on test cases for scattering from a sphere, so a comparison was possible. In this case a Robin problem (2) with $\alpha=1$ and $\beta=-1/(10-10i)$ was solved for a sphere of radius $a=20\lambda$ ($kD=435$). The incident field was generated by a source located at $(0,0,10a)$. Unfortunately there were no data on the iteration process used, and the only notice was that the process converged to the residual 10^{-5} . So the same parameters were used for the present test case. Further, a mesh containing 1 046 528 panels was used. Since this mesh was not available, a mesh of similar size that could

be generated was used. This mesh contained 1 130 988 triangles and 565 496 vertices. Another mesh tried contained 2 096 688 panels and 1 048 346 vertices. In the cited paper Green’s identity was used, while in the present study the combined Eq. (12) with λ from Eq. (41) is used. An increase in η from the value of 0.03 used for the solution of the Neumann problem to a value of 1.2 improved convergence for the Robin problem. The results are shown in Table IV.

It is seen that the present algorithm performed one order of magnitude faster, while the memory consumption was of the same order. Of course, the present algorithm was parallelized and executed on a four core machine, compared to the one core serial implementation of the Multilevel Fast Multipole Algorithm. However, even assigning a perfect parallelization factor of 4, we see that the same problem can be solved several times faster. The speed-up of the present algorithm can be related to several issues, one of which is the preconditioning and use of the combined equation instead of Green’s identity. For the largest case reported in Table IV, the present iteration took 23 outer loop iterations, for which matrix-vector multiplication took about 61 s/iteration, while for the inner loop, which was in any case limited by 11 iterations, the matrix-vector product took about 27 s/iteration.

As far as low-frequency computation comparisons are concerned, comparisons were performed against Refs. 4, 7, and 9. Here again Table I shows that the present code has performance that is comparable or better. Here it is seen that some of the simulations reported in the literature have discretizations that are much finer than what may have been required by the problem.

B. More complex shapes

Many problems in acoustics require computations for substantially complex shapes, which include bioacoustics,

TABLE III. Results from some recent FMM and FMM accelerated BEM publications operating in the low-frequency regime. The performance of the FMM and the FMM-BEM reported in this paper is comparable or superior (see Table I). The parameter $N/(kD)^2$ represents how overd discretized the problem is vis-à-vis the restrictions imposed by the frequency.

Ref.	kD	N	$N/(kD)^2$	Mem	Time	Iteration	Time/Iteration	Remarks
9	16	1536	6.13	4800	28	5	5.6	Internal room
9	16	98 304	6.13	1487	14	5	2.8	Internal room
14	0.9375	45 056	51 263	...	41 962	452	92.2	Internal L-shape
14	0.9375	704	801	...	25.1	21	1.19	Internal L-shape
7	3.464	200 000	16 668	8000	...	External scattering

TABLE IV. Comparison of solution times and memory at $kD=435$ reported in Ref. 10 and the present study.

	No. of elements	N (unknowns)	T (s)	M (Gbyte)
10	1 046 528	1 046 528	54 267	4.8
Present (case 1)	1 130 988	565 496	3 820	2.7
Present (case 2)	2 096 688	1 048 346	7 762	2.8

human hearing, sound propagation in dispersed media, engine acoustics, and room acoustics. Many such cases are reported in the literature: e.g., scattering off aircraft,¹⁷ off animals,¹⁷ off engine blocks,⁷ and off many scattering ellipsoids.⁷ The present algorithm was tested on many such cases to show that the iteration process is convergent for these cases and that solutions can be obtained on bodies with thin and narrow appendages. Figure 7 provides an idea on the sizes and geometries that were tested. Note that modeling of complex shapes requires surface discretizations, which are determined not only by the wavelength-based conditions $\chi \ll 2\pi$ but also by the requirements that the topology and some shape features should be properly represented. Indeed even for a solution of the Laplace equation ($k=0$), the boundary element methods can use thousands and millions of elements that just properly represent the geometry. Usually the same mesh is used for multifrequency analysis, in which case the number of elements is fixed and selected to satisfy criteria for the largest k required. In this case the number of elements per wavelength for small k can be large. Also, of course, discretization plays an important role in the accuracy of computations. So if some problem with complex geometry should be solved with high accuracy, then the number of elements per wavelength can be again large enough.

The last geometry illustrated in Fig. 7 is similar to the case studied in Ref. 7 for $kD=3.464$. Here the performance and accuracy were studied for a range of kD from 0.35 to 175 ($ka=0.01-5$, where a is the largest axis of an ellipsoid). The surface of each ellipsoid was discretized with more than 1000 vertices and 2000 elements to provide an acceptable accuracy of the method even for low frequencies. The convergence for the Neumann, Dirichlet, and Robin problems was very fast (just a few iterations) for small ka , where, as discussed above, the use of low-frequency FMM is necessary. The convergence was not affected by the increase in the number of nodes, although for accuracy better discretizations

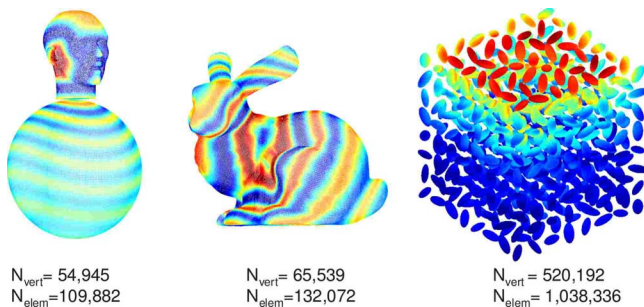


FIG. 7. (Color online) Examples of test problems solved with the present version of the BEM: human head-torso, bunny models (7.85 and 25 kHz acoustic sources located inside the objects, $kD=110$ and 96, respectively), and plane wave scattering by 512 randomly oriented ellipsoids ($kD=29$).

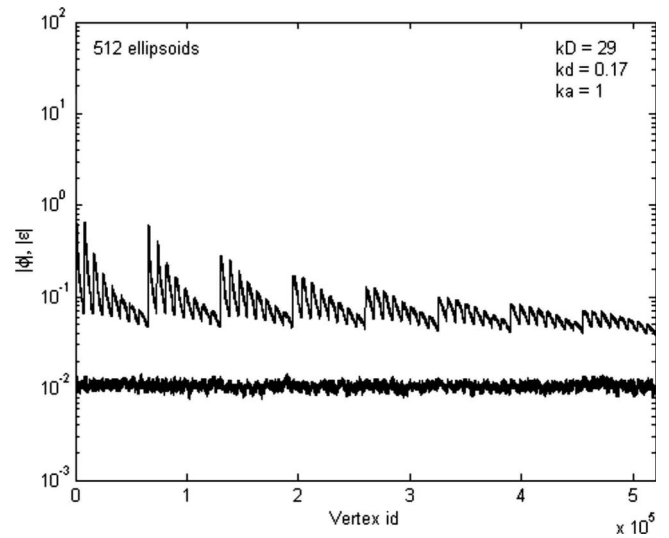


FIG. 8. Solution (the upper curve) and error (the lower curve) in the boundary condition at each vertex for the case of the ellipsoids in Fig. 7.

are preferred. As a test solution we used an analytical solution when a source was placed inside one of the ellipsoids, and surface values and normal derivatives were computed at each vertex location analytically. For larger discretizations that were used, we were able to achieve $\sim 1\%$ relative errors in strong norm (L_∞) for the range of parameters used.

Figure 8 shows an absolute relative error at each vertex,

$$\epsilon_i = \frac{|\phi_i^{(\text{BEM})} - \phi_i^{(\text{an})}|}{|\phi_i^{(\text{an})}|}, \quad i = 1, \dots, N_{\text{vert}}, \quad (43)$$

where $\phi_i^{(\text{an})}$ and $\phi_i^{(\text{BEM})}$ are the analytical and BEM solutions, and $|\phi_i^{(\text{an})}|$ is the modulus of the solution. The maximum error here was $\max(\epsilon_i) = 1.58\%$, which is usually acceptable for physics-based problems and engineering computations.

C. Computational challenges and drawbacks

Despite the fact that the present algorithm can run cases in a wide range of kD up to 500, the numerical stability for larger kD is still a question. This mostly relates to the computation of matrix $S|S$ - and $R|R$ -translation operators at higher frequencies via the RCR-decomposition. Particularly we should use different recursions for the computation of entries of the rotation operators from those that were acceptable for lower truncation numbers than those described in our earlier paper¹³. Such recursions are available (see p. 336 in Ref. 12) and stable if recursive backpropagation is used. An analysis shows that recursive computation of coaxial translation operators can be also unstable at larger frequencies, and this subject requires additional consideration, which is beyond the present paper.

In terms of practical use of the algorithm, it is still a research question: how to improve preconditioning. For example, the current preconditioner worked well for the external Neumann problem in automatic settings, while the solution of Robin or mixed problems showed a slow-down of the convergence, and there was a need for intervention and tuning of the η parameter and number of inner loop iterations

and accuracy to improve convergence. Further research on the convergence of the iterative method is necessary.

VII. CONCLUSION

A version of the FMM accelerated BEM is presented, where a scalable FMM is used both for dense matrix-vector multiplication and preconditioning. The equations solved are based on the Burton–Miller formulation. The numerical results show scaling consistent with the theory, which far outperforms conventional BEM in terms of memory and computational speed. Realization of a broadband FMM for efficient BEM requires different schemes for the treatment of low- and high-frequency regions and switching from multipole to signature function representation. The tests of the methods for simple and complex shapes show that it can be used for an efficient solution of scattering and other acoustical problems encountered in practice for a wide range of frequencies.

ACKNOWLEDGMENTS

This work was partially supported by a Maryland Technology Development Corporation grant in 2006 (at the University of Maryland) and partially by Fantalgo, LLC. We thank Dr. Vincent Cottoni and Dr. Phil Shorter of the ESI Group U.S. R&D for a careful testing of the software on different meshes before its inclusion in their VA-One product.

- ¹L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (MIT, Cambridge, MA, 1988).
- ²N. Nishimura, “Fast multipole accelerated boundary integral equation methods,” *Appl. Mech. Rev.* **55**, 299–324 (2002).
- ³M. Fischer, “The fast multipole boundary element method and its application to structure-acoustic field interaction,” Ph.D. dissertation, Institut A für Mechanik der Universität Stuttgart (2004).
- ⁴M. Fischer, U. Gauger, and L. Gaul, “A multipole Galerkin boundary element method for acoustics,” *Eng. Anal. Boundary Elem.* **28**, 155–162 (2004).
- ⁵T. Sakuma and Y. Yasuda, “Fast multipole boundary element method for large-scale steady-state sound field analysis, Part I: Setup and validation,” *Acust. Acta Acust.* **88**, 513–525 (2002).
- ⁶S. Schneider, “Application of fast methods for acoustic scattering and radiation problems,” *J. Comput. Acoust.* **11**, 387–401 (2003).
- ⁷L. Shen and Y. J. Liu, “An adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton–Miller formulation,” *Comput. Mech.* **40**, 461–472 (2007).
- ⁸Y. Yasuda and T. Sakuma, “Fast multipole boundary element method for large-scale steady-state sound field analysis, Part II: Examination of numerical items,” *Acust. Acta Acust.* **89**, 28–38 (2003).

- ⁹Y. Yasuda and T. Sakuma, “An effective setting of hierarchical cell structure for the fast multipole boundary element method,” *J. Comput. Acoust.* **13**, 47–70 (2005).
- ¹⁰M. S. Tong, W. C. Chew, and M. J. White, “Multilevel fast multipole algorithm for acoustic wave scattering by truncated ground with trenches,” *J. Acoust. Soc. Am.* **123**, 2513–2521 (2008).
- ¹¹B. Chandrasekhar and S. M. Rao, “Acoustic scattering from rigid bodies of arbitrary shape—double layer formulation,” *J. Acoust. Soc. Am.* **115**, 1926–1933 (2004).
- ¹²N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions* (Elsevier, Oxford, UK, 2004).
- ¹³N. A. Gumerov and R. Duraiswami, “Recursions for the computation of multipole translation and rotation coefficients for the 3-D Helmholtz equation,” *SIAM J. Sci. Comput. (USA)* **25**, 1344–1381 (2003).
- ¹⁴E. Darve and P. Havé, “Efficient fast multipole method for low-frequency scattering,” *J. Comput. Phys.* **197**, 341–363 (2004).
- ¹⁵L. Greengard, J. Huang, V. Rokhlin, and S. Wandzura, “Accelerating fast multipole methods for the Helmholtz equation at low frequencies,” *IEEE Comput. Sci. Eng.* **5**, 32–38 (1998).
- ¹⁶V. Rokhlin, “Diagonal forms of translation operators for the Helmholtz equation in three dimensions,” *Appl. Comput. Harmon. Anal.* **1**, 82–93 (1993).
- ¹⁷H. Cheng, W. Y. Crutchfield, Z. Gimbutas, L. F. Greengard, J. F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao, “A wideband fast multipole method for the Helmholtz equation in three dimensions,” *J. Comput. Phys.* **216**, 300–325 (2006).
- ¹⁸*Fast and Efficient Algorithms in Computational Electromagnetics*, edited by W. C. Chew, J. M. Jin, E. Michelsen, and J. Song (Artech House, Boston, 2001).
- ¹⁹Y. Saad, “A flexible inner-outer preconditioned GMRES algorithm,” *SIAM J. Sci. Comput. (USA)* **14**, 461–469 (1993).
- ²⁰A. J. Burton and G. F. Miller, “The application of the integral equation methods to the numerical solution of some exterior boundary-value problems,” *Proc. R. Soc. London, Ser. A* **323**, 201–210 (1971).
- ²¹L. H. Chen and J. Zhou, *Boundary Element Methods* (Academic, New York, 1992).
- ²²S. Kirkup, “The boundary element method in acoustics,” <http://www.boundary-element-method.com/acoustics/index.htm> (Last viewed December 30, 2007).
- ²³Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput.* **7**, 856–869 (1986).
- ²⁴N. A. Gumerov and R. Duraiswami, “Computation of scattering from clusters of spheres using the fast multiple method,” *J. Acoust. Soc. Am.* **117**, 1744–1761 (2005).
- ²⁵E. Darve, “The fast multipole method (I): Error analysis and asymptotic complexity,” *SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal.* **38**, 98–128 (2000).
- ²⁶M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions* (National Bureau of Standards, Washington, DC, 1964).
- ²⁷J. R. Driscoll and D. M. Healy, Jr., “Computing Fourier transforms and convolutions on the 2-sphere,” *Adv. Appl. Math.* **15**, 202–250 (1994).
- ²⁸E. Lutz, A. R. Ingraffea, and L. J. Gray, “Use of ‘simple solutions’ for boundary integral equation methods in elasticity and fracture analysis,” *Int. J. Numer. Methods Eng.* **35**, 1737–1751 (1992).
- ²⁹D. Colton and R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory* (Springer-Verlag, Berlin, 1998).