

Fast Multipole Accelerated Boundary Element Methods for the 3D Helmholtz Equation

Nail A. Gumerov and Ramani Duraiswami

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

(Dated: 31 December 2007)

Abstract

The development of a fast multipole method accelerated iterative solution of the boundary element equations for large problems involving hundreds of thousands elements for the Helmholtz equations in 3D is described. The BEM requires several approximate computations (numerical quadrature, approximations of the boundary shapes using elements) and the convergence criterion for iterative computation. When accelerated using the FMM, 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 used, choices of representations, and boundary element integration consistent with these approximations are described. A novel preconditioner for accelerating convergence, using a low accuracy FMM accelerated solver as a right preconditioner is also described. Results of the developed and tested solvers for boundary value problems for the Helmholtz equations using the solver are presented for the number of unknowns $N \lesssim 10^6$ and product of wavenumber k times domain size D , $kD \lesssim 200$ and show good performance close to theoretical expectations.

PACS numbers: 43.20.Rz, 43.55.Ka

Keywords: Fast Multipole Method, Boundary Element Method, Preconditioning, Low Frequency, High Frequency, Scattering

I. INTRODUCTION

Boundary element methods (BEM) 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 the integral equation techniques lead to linear systems with dense and possibly non-symmetric matrices. For a problem with N unknowns, direct solution requires $O(N^3)$ solution cost, and storage of $O(N^2)$ elements of these matrices. The computation of the individual matrix elements is also expensive requiring quadrature of nonsingular, weakly singular, or hypersingular functions. To reduce the singularity order and achieve symmetric matrices, many investigators employ Galerkin techniques, which lead to further $O(N^2)$ integral computations. Direct solution of the linear systems has an $O(N^3)$ cost. Use of iterative methods does not reduce the memory or integral computation costs, but can reduce the cost to $O(N_{iter}N^2)$ 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. Because of these reasons the BEM was not used for very large problems.

The development of the fast multipole method (FMM) [15] and use of preconditioned Krylov iterative methods presents a promising approach to improving the scalability of integral equation methods. 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 this fast matrix vector product in a quickly convergent iterative scheme allows the system of equations to be rapidly solved with $O(N_{iter}N)$ cost. The FMM was initially developed for gravity or electrostatic potential problems. Later this method was intensively studied and extended to solution of many other problems, including those arising from the Helmholtz, Maxwell, biharmonic and elasticity equations. While the literature and previous work on the FMM is extensive, reasons of space do not permit us to provide a complete discussion of the literature. We refer the reader to the comprehensive review [21].

Numerical solution of the Helmholtz equation is somewhat complicated since the equation con-

tains a length scale parameter (the wavenumber) at which order the solution must be resolved. The FMM for the Helmholtz equations also exploits factorized representations of the Green’s function. However, for the Helmholtz equation the size of the representations, and the cost of translations are all related to the wavenumber, and this makes the assessment of costs more complicated, and the development of FMM accelerated BEM more involved. We discuss these issues further below.

Despite several publications related to the FMM accelerated BEM for scattering [13, 14, 26–30] there are several issues which must be resolved. These include

- Choice of translation scheme which is stable and efficient over a range of wavenumbers
- Choice of appropriate preconditioned Krylov iteration method, and associated preconditioner
- Ability to resolve problems caused at wavenumbers at which the corresponding interior problem has resonances via the Burton-Miller technique
- Efficient implementations that achieve predicted scalings in CPU time and memory

We discuss these issues further in the context of the contributions of this paper, and distinguish it from previous published work.

A. Error and fast multipole accelerated boundary elements

Since the FMM achieves an approximate matrix vector product, we should emphasize that in practice this accuracy can be made close to machine precision. In any case an accuracy criterion is employed to stop the iterative process, and the accuracy of the FMM should be considered together with accuracy of the BEM technique. The latter is quite approximate in practice, since significant errors are introduced via surface approximation via discretization and less significant ones via approximate computation of the boundary integrals. In practice (see e.g., [3, 14, 26]) error tolerances are quite high (from a high of a few percent to at most 10^{-4}). Since these errors all influence the final solution error, it does not make sense to perform any particular part of the calculation, with an exceedingly high precision. In fact all errors must be balanced so that the specified error is achieved.

B. Resolving calculations at all 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 enough frequencies, since at lower frequencies the accuracy of shape representation is more restrictive. So two basic regimes are usually recognized: the low-frequency regime and the high-frequency regime. These regimes can be characterized by some threshold value $(kD)_*$ of parameter kD , where k is the wavenumber and D is the computational domain size. For each of these regimes the complexity of the FMM is different [18]. The goal of the FMM is to reduce this per iteration cost, while preconditioning attempts to reduce the number of iterations.

1. Low frequency regime

In this regime $kD < (kD)_*$ and the per iteration step cost of the FMM can be expressed as

$$\text{Cost per step} = O(N) \times \text{CT},$$

where CT is the cost of translations, where CT and N do not depend on kD and can be determined based on the required accuracy of computations. The choice of the translation scheme affects CT, and so the asymptotic constant, as the overall complexity will be in any case $O(N_{iter}N)$. For the low frequency regime the most efficient translation schemes are based on the RCR (rotation-coaxial translation-back rotation) decompositions [17, 18], which have $O(p^3)$ complexity, where p^2 is the number of terms in the multipole expansion (p in this regime can be constant), and based on the low-frequency exponential forms [10, 16], which have the same complexity, but with a different asymptotic constant. The method based on sampling of the far field signature function [23] are not stable in this region due to exponential growth of terms in the multipole-to-local translation kernel.

2. High frequency regime

In this regime $kD > (kD)_*$ and kD heavily affects the complexity of the FMM. In practice 5 to 10 points per wavelength are required. Since the wavenumber k is inversely proportional to wavelength, if a numerical method is a surface based method (such as the boundary element method), then

problem size N scales as $O((kD)^2)$, while for complex, space-filling surfaces (e.g. problem with many scatterers distributed in a volume) N scales as $O((kD)^3)$. Direct solution procedures require $O((kD)^4)$ memory and $O((kD)^6)$ operations for simple shapes and $O((kD)^6)$ memory and $O((kD)^9)$ operations for space-filling surfaces, while iterative solution can be achieved for $O((kD)^4)$ memory and $O((kD)^4)$ computation per iteration cost for simple shapes and $O((kD)^6)$ complexities for space-filling surfaces.

The high-frequency FMM is designed in this case to increase the size of wavefunction representation, $O(p^2)$, as levels go up in the hierarchical space subdivision, with p proportional to the size of the boxes at given level [18]. Because of this the complexity of the FMM is heavily affected by the complexity of a single translation. It is shown that $O(p^3)$ schemes in this case provide the 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 $O((kD)^4)$ and $O((kD)^5)$ for simple shapes and, in fact the use of such schemes has no complexity advantage compared to the direct matrix-vector multiplication. Despite this some authors used such high complexity translation schemes for the BEM (e.g. [28]), which therefore limits practical use of their software to the low-frequency regime.

In a sense of reaching the best scaling algorithm in this regime translation methods based on sampling of the far-field signature function [23] are appropriate. 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. However, it is noteworthy, that function representations via samples of the far-field signature function are at least twice as large as those for the multipole expansions. Also the multipole-to-local translation kernel should be sampled with double frequency compared to the translated function size, which increases the size of the transforms and complexity of filtering. These costs means that the value of p at which the asymptotically efficient algorithm performs better than the $O(p^3)$ algorithm can be high.

3. Switch in function representations

The use of the FMM for the BEM even at high kD requires use of different translation schemes for fine and coarse levels of space subdivision. Indeed at the fine levels parameter ka , where a is a representative box size is small enough and translations efficient for low-frequency regime should be used, while for coarse levels this parameter is large and the high-frequency regime should be treated efficiently. So a combined scheme in which the spherical wave function representation can be converted to signature function sample representation can be applied. Such a switch was suggested and tested recently in [5]. In the present paper we also use a switch, while our scheme is different and does not require interpolation/antepolation. We use $O(p^3)$ spherical transform, while this can be replaced by methods of lower asymptotic complexity (such as based on FFT or 1D-FMM, [5, 18]) for larger problems, though in our experience the costs for the range of kD considered in this paper, are comparable. The spherical wavefunction based representation is convenient for differentiation, and this allows easier implementation of the Burton-Miller method, as the differential operators there can be easily expressed in terms of the expansion coefficients, as is shown further below.

C. Iterative methods and preconditioning

The second issue with fast multipole acceleration is the choice of preconditioning strategy. Preconditioning for boundary element matrices is in general a lesser studied issue than for finite element and finite difference based discretization. From that theory, it is known that for high wave numbers preconditioning is difficult, and an area of active research. Many conventional pre-conditioning strategies rely on sparsity in the matrix, and applying them to these dense 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 [13, 14]. However such local preconditioning strategies work well only well for low wavenumbers. Instead we consider the use of a low accuracy FMM itself as a preconditioner by using a flexible GMRES procedure [24]. This novel preconditioner appears to work well at all wavenumbers considered, and stays within the required cost.

II. FORMULATION AND PRELIMINARIES

A. Boundary value problem

We consider numerical solution of the Helmholtz equation for the possibly complex valued potential ϕ

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

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

$$\alpha(\mathbf{x}) \phi(\mathbf{x}) + \beta(\mathbf{x}) q(\mathbf{x}) = \gamma(\mathbf{x}), \quad q(\mathbf{x}) = \frac{\partial \phi}{\partial n}(\mathbf{x}), \quad |\alpha| + |\beta| \neq 0, \quad \mathbf{x} \in S. \quad (2)$$

Here and below all normal derivatives are taken assuming that the normal to the surface is directed outward to V . For external problems we also assume that the field satisfies 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 (e.g. $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ cannot be simultaneously zero), and we rely on the typical physically meaningful conditions, which make the problem of determination of $\phi(\mathbf{x})$ well-posed. Particularly, if α and β are constant we have the Robin problem, which degenerates to the Dirichlet or Neumann problem, if $\beta = 0$ (sound-soft boundary) and $\alpha = 0$ (sound-hard boundary), respectively. Variation of α and β along the surface happens e.g., when we consider problems with variable impedance. Also this covers the case when on some part of the boundary $\phi(\mathbf{x})$ is known, while on another part of the boundary its normal derivative is provided. In this case $\alpha(\mathbf{x})$, $\beta(\mathbf{x})$, and $\gamma(\mathbf{x})$ are piecewise smooth.

B. Boundary integral equations

The boundary element method uses a formulation in terms of boundary integral equations whose solution with the boundary conditions provides the values of $\phi(\mathbf{x})$ and $q(\mathbf{x})$ on the boundary.

Knowledge of these quantities enables determination of $\phi(\mathbf{y})$ for any domain point \mathbf{y} . This can be done, e.g. using Greens 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:

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

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 we have for smooth S

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

The well-known deficiency of this formulation is related to possible degeneration of operators L and $(M - \frac{1}{2})$ at certain frequencies depending on S , which correspond to resonances of the internal problem for sound-soft and sound-hard boundaries [2, 4]. Even though the solution of the external problem is unique for these frequencies, Eq. (7) is deficient in these cases, since it provides a non-unique solution. Moreover, despite the spurious values of resonance frequencies, 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 non-uniqueness of the solution for the internal problem, seems to have physical meaning, as there, in fact, can be resonances.

In any case, boundary integral equation (7) can be modified to avoid the artifact of degeneracy of boundary operators when solving the correctly posed problem (1)-(3). This can be done using different techniques, including direct and indirect formulations, introduction of some additional field points, etc. We use direct formulation based on the integral equation combining Green's and Maue's identities, which is the same trick as proposed by Burton and Miller (1971) for sound-hard boundaries. The latter identity is

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

where

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

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

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

Burton & Miller [2] proved that it is sufficient to have $\text{Im}(\lambda) \neq 0$ to guarantee uniqueness of the solution for the external problem.

C. Combined equation

The system of equations (2) and (10) can be reduced to a single linear system for some vector of unknowns $[\psi]$

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

which is convenient for computations. Boundary operator A , and functions ψ and c can be constructed as follows (this is equivalent to elimination of one of the unknowns ψ or q on the part of the boundary, based on the magnitude of coefficients $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$).

Let $\sigma(\mathbf{x})$ be a logical function so that

$$\sigma(\mathbf{x}) = \begin{cases} 1, & |\alpha(\mathbf{x})| \leq |\beta(\mathbf{x})| \\ 0, & |\alpha(\mathbf{x})| > |\beta(\mathbf{x})| \end{cases}, \quad (12)$$

Then we define

$$\psi = \sigma\phi + (1 - \sigma)q. \quad (13)$$

Furthermore, we introduce new variables

$$\begin{aligned} u &= \left[\sigma - (1 - \sigma) \frac{\beta}{\alpha} \right] \psi, & u' &= \left[-\sigma \frac{\alpha}{\beta} + (1 - \sigma) \right] \psi, \\ b &= -(1 - \sigma) \frac{\gamma}{\alpha}, & b' &= -\sigma \frac{\gamma}{\beta}, \end{aligned} \quad (14)$$

with the remark that in case $\alpha = 0$ we have logical $\sigma = 1$ and $(1 - \sigma)/\alpha$ should be set to zero. The same relates to $\beta = 0$ where we have $\sigma/\beta = 0$. Also u and u' are proportional to a single unknown

ψ and can be found easily if ψ is specified. From these definitions and boundary conditions (2) we have

$$\begin{aligned}\phi &= \sigma\phi + (1-\sigma)\phi = \sigma\phi + (1-\sigma)\frac{\gamma-\beta q}{\alpha} = u-b, \\ q &= \sigma q + (1-\sigma)q = \sigma\frac{\gamma-\alpha\phi}{\beta} + (1-\sigma)q = u'-b'.\end{aligned}\tag{15}$$

Substituting these expressions into Eq. (10) we obtain Eq. (11), where

$$\begin{aligned}A[\psi] &= (L+\lambda L') [u'] - (M+\lambda M') [u] \mp \frac{1}{2} (u + \lambda u') \\ c &= (L+\lambda L') [b'] - (M+\lambda M') [b] \mp \frac{1}{2} (b + \lambda b').\end{aligned}\tag{16}$$

As soon as equation (11) is solved and ψ is found, we can determine u and u' from Eq. (14) and ϕ and q from Eq. (15).

D. Discretization

Boundary discretization leads to approximation of boundary functions via finite vectors of their surface samples and integral operators via matrices acting on that 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 can be computed, we obtain

$$\begin{aligned}L[q] \left(\mathbf{x}_l^{(c)} \right) &= \sum_{l'=1}^M \int_{S_{l'}} q(\mathbf{x}) G \left(\mathbf{x}, \mathbf{x}_l^{(c)} \right) dS(\mathbf{x}) \approx \sum_{l'=1}^M L_{ll'} q_{l'}, \quad l = 1, \dots, M, \\ q_{l'} &= q \left(\mathbf{x}_{l'}^{(c)} \right), \quad L_{ll'} = \int_{S_{l'}} G \left(\mathbf{x}, \mathbf{x}_l^{(c)} \right) dS(\mathbf{x}),\end{aligned}\tag{17}$$

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 [4, 20]. Similar formulae can be used for other operators. Note that to accurately capture the solution variation at the relevant length scales, the discretization should satisfy

$$kr_{\max} \ll 1,\tag{18}$$

where r_{\max} is the maximum size of the element. In practice, discretizations which provide several elements per wavelength are usually provide an accuracy consistent with the other errors of the BEM.

We have implemented and tested the above method of discretization with collocation at the panel centers. We also used another method while provides collocation at the vertices, which are located on the surface S , and sometimes more convenient as it leads to smaller systems. It can be shown that formally both have the same accuracy, despite the fact that vertex collocation has linear element functions. Formally, this is based on the following geometrical consideration.

We note that in the case when a triangular surface mesh is available, each triangle of area S_l with vertices \mathbf{x}_{l_1} , \mathbf{x}_{l_2} , and \mathbf{x}_{l_3} is subdivided into six triangles $S_{l''}$ of equal areas by its medians (the fact which is easy to prove using elementary geometry), which intersect at the triangle center $\mathbf{x}_l^{(c)} = (\mathbf{x}_{l_1} + \mathbf{x}_{l_2} + \mathbf{x}_{l_3})/3$. After that the smaller triangles having common vertex \mathbf{x}_j can be united into a non-flat patch with center \mathbf{x}_j which area s_j is $1/3$ of the sum of the areas of original triangles which include vertex \mathbf{x}_j . This is schematically shown in Fig. 1.

If not given, the normal to the surface at \mathbf{x}_j for smooth enough surfaces can be approximated by a sum of normals to those elements weighted by areas. So we have

$$s_j = \sum_{S_{l''} \ni \mathbf{x}_j} S_{l''} = \frac{1}{3} \sum_{S_l \ni \mathbf{x}_j} S_l, \quad \mathbf{n}_j = \frac{1}{3s_j} \sum_{S_l \ni \mathbf{x}_j} \mathbf{n}_l S_l. \quad (19)$$

Discretization of surface operator with collocation at the vertices can be then written as

$$\begin{aligned} L[q](\mathbf{x}_j) &= \sum_{l=1}^M \int_{S_l} q(\mathbf{x}) G(\mathbf{x}, \mathbf{x}_j) dS(\mathbf{x}) = \sum_{l''=1}^{6M} \int_{S_{l''}} q(\mathbf{x}) G(\mathbf{x}, \mathbf{x}_j) dS(\mathbf{x}) \\ &= \sum_{j'=1}^N \int_{s_{j'}} q(\mathbf{x}) G(\mathbf{x}, \mathbf{x}_j) dS(\mathbf{x}) \approx \sum_{j=1}^N L_{jj'} q_{j'} \\ L_{jj'} &= \int_{s_{j'}} q(\mathbf{x}) G(\mathbf{x}, \mathbf{x}_j) dS(\mathbf{x}) = \sum_{S_{l''} \ni \mathbf{x}_j} \int_{S_{l''}} G(\mathbf{x}, \mathbf{x}_j) dS(\mathbf{x}), \quad q_j = q(\mathbf{x}_j), \quad j = 1, \dots, N. \end{aligned} \quad (20)$$

In any case there exists a large literature and available subroutines for computation of the integrals over flat triangular elements including nearly singular, singular, weakly singular, and hypersingular cases [4]. Below we also propose a novel method for treatment of singular integrals. Discretization of the boundary operators reduces problem (11) to a system of linear equations which for small sized systems can be solved directly, while for larger problems iterative methods can be used to speed up computations.

E. Iterative Methods

Different iterative methods can be tried to solve equation (11) with a non-symmetric dense matrix A . Any iterative method requires matrix-vector multiplication or computation of $A[x]$, where $[x]$ is some input vector. For large systems we use the FMM (Fast Multipole Method) to achieve reduction in the memory and operational complexity in the way described below. The method of our choice is fGMRES (Flexible Generalized Minimal Residual Method) [24], which has an advantage of use of approximate right preconditioner, which in its turn can be computed by executing of the internal iteration loop using unpreconditioned GMRES [25]. Different preconditioners can be tried as soon as they provide grouping of matrix eigenvalues about 1 or approximate the matrix inverse. Choice of the preconditioning method must be achieved for a cost that is $O(N)$ or smaller. Also in some cases we employed the FMM for system preconditioning.

III. USE OF THE FAST MULTIPOLE METHOD

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

$$A = A_{sparse} + A_{dense}, \quad (21)$$

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, which selection can be based on some estimates or error bounds, while the dense part has nonzero entries A_{lj} for which $|\mathbf{x}_l - \mathbf{x}_j| \geq r_c$. Storage and multiplication of the sparse matrix has memory and computational complexity $O(N)$ as soon as $N \times N$ matrices are considered, while the corresponding complexities for the dense part are both $O(N^2)$, if direct methods are considered. The use of the FMM reduces the memory complexity 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 [18]. The decomposition (21) is at the heart of the FMM, with summation of the near-field interactions performed directly, while for the far-field interactions multipole and local expansions are used. The acceleration achieved is via memory and time efficient computation of the matrix-vector products involving A_{dense} .

A. FMM strategy

The use of the FMM for solution of boundary integral equations brings a substantial shift in the computational strategy. In the traditional BEM, since the full system matrix should 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 computation of the surface integrals. Even if one uses quadratures with 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, it is also possible to compute non-singular integrals very accurately in the FMM using expansions of Green's function, such as

$$G(\mathbf{x}, \mathbf{x}_l^{(c)}) = ik \sum_{n=0}^{\infty} \sum_{m=-n}^n R_n^{-m}(\mathbf{x} - \mathbf{x}_{l'}^{(c)}) S_n^m(\mathbf{x}_l^{(c)} - \mathbf{x}_{l'}^{(c)}), \quad (22)$$

where R_n^m and S_n^m are the spherical basis functions for the Helmholtz equation, we obtain from Eq. (17)

$$L_{ll'} = \sum_{n=0}^{\infty} \sum_{m=-n}^n C_n^m S_n^m(\mathbf{x}_l^{(c)} - \mathbf{x}_{l'}^{(c)}), \quad C_n^m = ik \int_{S_{l'}} R_n^{-m}(\mathbf{x} - \mathbf{x}_{l'}^{(c)}) dS(\mathbf{x}). \quad (23)$$

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

All methods, based on high order quadratures, computations of the expansion coefficients or use of some analytical formulae with compositionally complex functions impose substantial limits on the use of the advantages of the FMM, which otherwise is capable to handle million size problems on usual desktop PCs. 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 we face then with a well-known dilemma to compute integrals in the flat panel approximation with high number of nodes, or just increase the total number of nodes (discretization density) and use low order quadrature. In the case of use of the FMM with “on fly” integral computations the computational complexity will be almost the same for both of these ways, while the latter way seems preferable, as it allows the function 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 normal are accounted better). Of course these variations in any case should be small enough, for the method to converge. We also note that based on our experience for the same problem (geometry and wavelength) the number of iterations to achieve convergence, after some N (once the system is resolved) almost does not depend on N .

Therefore, in the case of the use of the FMM we can try to use the following approximation, at least in the far-field, for the non-singular integrals (in case of vertex collocation):

$$\begin{aligned} L_{lj} &= s_j G(\mathbf{x}_j, \mathbf{x}_l), \quad M_{lj} = s_j \frac{\partial G}{\partial n_j}(\mathbf{x}_j, \mathbf{x}_l), \\ L'_{lj} &= s_j \frac{\partial G}{\partial n_l}(\mathbf{x}_j, \mathbf{x}_l), \quad M'_{lj} = s_j \frac{\partial^2 G}{\partial n_l \partial n_j}(\mathbf{x}_j, \mathbf{x}_l), \quad l, j = 1, \dots, N, \quad \mathbf{x}_l \neq \mathbf{x}_j. \end{aligned} \quad (24)$$

Collocation at the panel center uses the same formulae with respective values of areas, nodes and normals. For the treatment of the singular integrals ($\mathbf{x}_l = \mathbf{x}_j$) we use a method described later. As we mentioned above we use these approximations for computations of the product involving the dense part of the matrix, or the far field. For near field computations these formulae could be used with a fine enough discretization for the non singular integrals, though one may prefer to use higher order quadrature. We performed several tests using for near field integral representation Gauss quadratures of varying order (in range 1-625 nodes per element) and found that approximation (24) used for near field provides fairly good results for good meshes (the element size and aspect ratios stay within some bounds), though some poor meshes provide not so good results, and we provide this choice as a switch in the code.

B. FMM algorithm

The Helmholtz FMM algorithm which we employed for matrix vector products is described in [18, 19], with modifications that allows use of different translation schemes for low and high frequencies. Particulars of our algorithm are that we use level-dependent truncation number p_l and rectangularly truncated translation operators for multipole-to-multipole and local-to-local translations, which are performed using rotation-coaxial translation-rotation (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 $ka_l < ka_*$, where a_l is the radius of the smallest sphere surrounding a box on level l , and ka_* is some critical value of the size parameter. For levels corresponding to

$ka_l \geq ka_*$ we convert the multipole expansions to samples of the signature function at a cost of $O(p^3)$, then employ diagonal forms of the translation operator $O(p^2)$, and in the downward pass at some appropriate level use conversion of the signature function to the local expansion of the required length at a cost of $O(p^3)$. This procedure automatically provides filtering. We also note that conversions from multipole and 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. So our algorithm, in this part, is close to that described in [5]. The difference is that we do not need any interpolation/antepolation procedures. Also for low-frequency translation we used RCR-decomposition for the multipole-to-local translation, which we found as efficient as the method based on conversion into exponential forms for moderate p . Particulars of our implementation include precomputation of all translation operators, particularly translation kernels, so during the run time of the procedure, which is performed many times for the iterative process only simple arithmetic operations (additions and multiplications) are executed. Below we briefly describe some details of the algorithm.

1. Data structure

The present version of the FMM employs a traditional 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 precompute lists of neighbors and children, which are stored and used as needed. So the adaptivity of the FMM used is in skipping of “empty” boxes at all levels (such boxes simply do not enter the data structure). It is perhaps because of our use of these data structures that even on a modest PC the times we report for the FMM matrix vector product are superior to those of several authors.

2. Level dependent truncation number

Each level is characterized by the size of the expansion domain, which is a sphere of radius a_l concentric with the box. Selection of the truncation number in the algorithm is automated based

on equation $p_l = p(ka_l, \epsilon, \delta)$, where ϵ is the prescribed accuracy of the FMM, and δ is the separation parameter (we used $\delta = 2$ (see the justification in [18])). Detailed discussion and theoretical error bounds for such dependences can be found elsewhere (e.g. [9, 18]). Particularly, the following approximation combining low and high-frequency asymptotics for monopole expansions can be utilized [18]:

$$p_{low} = \frac{1}{\ln \delta} \ln \frac{1}{\epsilon (1 - \delta^{-1})^{3/2}} + 1, \quad p_{high} = ka + \frac{1}{2} \left(3 \ln \frac{1}{\epsilon} \right)^{2/3} (ka)^{1/3}, \quad (25)$$

$$p = (p_{low}^4 + p_{high}^4)^{1/4}.$$

It is also shown in [18] 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. In our automatic settings we computed p_{low} and p_{high} and if it happened that $p - p_{high} > p_*(\epsilon)$, where p_* is some number dictated by the overall accuracy requirements (for the errors acceptable for this paper constant $p_* = 5$ was good enough for most cases tested; in fact, this function should be proportional to $\ln(1/\epsilon)$) we used $p = p_0 + p_{high}$, otherwise Eq. (25) was used.

As mentioned above, in the algorithm we implemented an automatic switch from the RCR-decomposition to the diagonal forms of the translation operators based on criterion $ka_l \geq ka_*$. Parameter ka_* was based on the error bounds (25) and was selected for the level at which $p - p_{high} \leq p_{**}$ (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 (33) starts to grow exponentially (see [18]).

Fig. 2 shows the dependences provided by Eq. (25). We note that FMM with very small 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 [5, 10] a method based on diagonalization of the translation operators, different from [23] were developed. This method, however requires some complication in data structure (decomposition to the x, y, z -directional lists) and 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 due to flat panel discretization). In this case efficiency of the matrix-based methods, such as the RCR-decomposition is comparable, or even better. 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 on operations on larger representing vectors.

3. Multipole expansions

Expansions over the singular (radiating) spherical basis functions $S_n^m(\mathbf{r})$ in the form (22)-(23) can be applied to represent the monopole source or respective integrals. In these formulae the singular and regular solutions of the Helmholtz equation are defined as

$$S_n^m(\mathbf{r}) = h_n(kr)Y_n^m(\theta, \varphi), \quad R_n^m(\mathbf{r}) = j_n(kr)Y_n^m(\theta, \varphi), \quad n = 0, 1, 2, \dots; \quad m = -n, \dots, n, \quad (26)$$

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

$$Y_n^m(\theta, \varphi) = (-1)^m \sqrt{\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta) e^{im\varphi}, \quad (27)$$

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

and $P_n^{|m|}(\mu)$ are the associated Legendre functions consistent with that in [1], or Rodrigues' formulae

$$P_n^m(\mu) = (-1)^m (1 - \mu^2)^{m/2} \frac{d^m}{d\mu^m} P_n(\mu), \quad n \geq 0, \quad m \geq 0, \quad (28)$$

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

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 type (22)-(23) for the monopoles by applying appropriately truncated differential operators in the space of the expansion coefficients [18], 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

$$\hat{C}_n^m = \frac{1}{2t} [(n_x + in_y) (b_n^m C_{n-1}^{m+1} - b_{n+1}^{m-1} C_{n+1}^{m+1}) + (n_x - in_y) (b_n^{-m} C_{n-1}^{m-1} - b_{n+1}^{m-1} C_{n+1}^{m-1})] \quad (29)$$

$$+ n_z (a_n^m C_{n+1}^m - a_{n-1}^m C_{n-1}^m), \quad m = 0, \pm 1, \pm 2, \dots, \quad n = |m|, |m| + 1, \dots$$

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

$$\begin{aligned}
a_n^m &= a_n^{-m} = \sqrt{\frac{(n+1+m)(n+1-m)}{(2n+1)(2n+3)}}, \quad \text{for } n \geq |m|, \\
a_n^m &= b_n^m = 0, \quad \text{for } n < |m|, \\
b_n^m &= \sqrt{\frac{(n-m-1)(n-m)}{(2n-1)(2n+1)}} \text{ for } 0 \leq m \leq n, \\
b_n^m &= -\sqrt{\frac{(n-m-1)(n-m)}{(2n-1)(2n+1)}} \text{ for } -n \leq m < 0.
\end{aligned} \tag{30}$$

4. Translations

Translations of the expansions can be also thought as application 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 $p_{l'}^2$ translated coefficients relate to p_l^2 original coefficient via $p_{l'}^2 \times p_l^2$ matrix. Even for precomputed and stored matrices this requires $O(p^4)$ operations, which is unallowable cost for the translation if using with boundary element methods [18]. Several methods to reduce this cost are well-known. Particularly we use the RCR-decomposition of the **(S|S) (t) = (R|R) (t)** matrices

$$(\mathbf{R}|\mathbf{R}) (\mathbf{t}) = \mathbf{Rot}^{-1}(\mathbf{t}/t) \underline{(\mathbf{R}|\mathbf{R})} (t) \mathbf{Rot}(\mathbf{t}/t), \tag{31}$$

where \mathbf{t} is the translation vector, $t = |\mathbf{t}|$, and $\mathbf{Rot}(\mathbf{t}/t)$ is the rotation matrix, which expresses coefficients in the rotated reference frame, which z -axis is collinear with \mathbf{t} , while $\underline{(\mathbf{R}|\mathbf{R})} (t)$ is the coaxial translation operator (along axis z). In the RCR-decomposition all operators cost $O(p^3)$ due to they act on different subspaces. As the geometry of the problem is specified all these matrices can be precomputed for the cost of $O(p^3)$ operations using recursions [17, 18] and stored. We note also that due to rectangular truncation operators $\mathbf{Rot}(\mathbf{t}/t)$ and $\mathbf{Rot}^{-1}(\mathbf{t}/t)$ act on the vectors of length p_l^2 and $p_{l'}^2$, respectively, produces the same size vectors, while $\underline{(\mathbf{R}|\mathbf{R})} (t)$ acts on vector of size p_l^2 and produces the vector of size $p_{l'}^2$. Therefore there is no need in any interpolation or filtering procedures, as this is embedded into the decomposition. We apply similar decomposition to the **(S|R) (t)** matrix for low frequencies, which provides numerically stable low-frequency procedure (for levels corresponding to $ka_l < ka_*$).

For higher frequencies (for levels $ka_l \geq ka_*$) we use the following decomposition of the translation matrix $(\mathbf{S}|\mathbf{R})(\mathbf{t})$:

$$(\mathbf{S}|\mathbf{R})(\mathbf{t}) = \mathbf{S}\mathbf{p}^{-1}\mathbf{\Lambda}_s(\mathbf{t})\mathbf{S}\mathbf{p}, \quad (32)$$

where $\mathbf{S}\mathbf{p}$ can be thought 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), $\mathbf{\Lambda}_s(\mathbf{t})$ is a diagonal translation matrix of size $N_l \times N_l$ and $\mathbf{S}\mathbf{p}^{-1}$ is the matrix of size $p_l^2 \times N_l$, which provides 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 $2p_l - 1$ Gauss abscissas with respect to $-1 \leq \mu = \cos \theta \leq 1$ and $4p_l - 3$ equispaced abscissas with respect to the 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 $\mathbf{\Lambda}_s(\mathbf{t})$ can be computed as

$$\Lambda_{jj}(\mathbf{t}) = \sum_{n=0}^{2p_l-2} i^n (2n+1) h_n(kt) P_n\left(\frac{\mathbf{s}_j \cdot \mathbf{t}}{t}\right), \quad j = 1, \dots, N_l, \quad (33)$$

which is a diagonal form of the translation operator [23]. The bandwidth of this function, $2p_l - 2$, provides that decomposition (32) of the $p_l^2 \times p_l^2$ translation matrix $(\mathbf{S}|\mathbf{R})(\mathbf{t})$ is exact [18]. Note that for a given grid (which is the same for all translations at level l) the cost of computation of $\mathbf{\Lambda}_s(\mathbf{t})$ for each translation vector \mathbf{t} is $O(p_l^3)$. In our implementation we precompute and store all these entries, so no computations of $\mathbf{\Lambda}_s(\mathbf{t})$ is needed during the run part of the algorithm. We also speed up the precomputation part by employing some data structure, which eliminates computations of $\Lambda_{jj}(\mathbf{t})$ for repeated entries $\frac{\mathbf{s}_j \cdot \mathbf{t}}{t}$ and kt for all translations, and, in fact allows substantially reduce the preset part of the algorithm.

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 [5, 11, 18]). If performed straightforward each of them requires $O(p^3)$ operations. Despite there exist algorithms for fast Legendre transform and the 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 still can be 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, 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 for computation of the forward transform.

As mentioned earlier, since the same transforms \mathbf{Sp} and \mathbf{Sp}^{-1} should be applied to all expansions at given level, the amortization of computations should be used to make (32) more efficient than the RCR-decomposition. In this case we apply first transform \mathbf{Sp} to all box expansions at given level, then perform all diagonal translations and consolidations, and, finally, apply transform \mathbf{Sp}^{-1} to all boxes.

5. Comparison of algorithms

Fig. 3 illustrates the present algorithm (on the right) and also compares it with the FMM for the Helmholtz equation proposed in [5]. These algorithms have in common 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 algorithm [5] for lower frequencies, while instead of conversion to the exponential form the spherical transform is used to convert the S-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 (for translation in each coordinate direction). Also this trick is efficient for large enough p , which is necessary for the high frequency region. However, despite the use of these efficient techniques, our 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 are used.

According to [18] both algorithms have the same complexity at low frequencies (with different asymptotic constants), while the asymptotic complexity of the algorithm [5] at large kD is $O((kD)^2 \log^\alpha(kD))$ for simple shapes (with relatively high asymptotic constant) and $O((kD)^3)$ for space-filling surfaces. The complexities of the present algorithm are $O((kD)^3)$ for simple shapes (with relatively low asymptotic constant) and $O((kD)^3 \log^\alpha(kD))$, respectively. The concept of “space-filling” surfaces (related to how many boxes at certain level are occupied by the surface) should be considered together with the value of kD . Indeed, the switch to the high-frequency regime occurs not at the finest level, but rather at some level, at which a substantial number of boxes in the computational domain may have expansions at their centers, i.e. are “occupied”. Thus, as far as the algorithm performance is concerned, the effective dimension of the manifold representing the surface of the object at this level will not appear as 2, but somewhere between 2 and 3 (see [18]). In this case the complexity of algorithm [5] will be between $O((kD)^2 \log^\alpha(kD))$ and $O((kD)^3)$.

Because of this, the practical efficiency of the two algorithms may be similar for many problems. In any case this issue is complicated and additional analysis is required to compare them (also one should take into account hardware limits and the efficiency of implementation). Our results below indicate that the present algorithm can be successfully used for efficient solution of many problems.

6. *Evaluation of expansions*

Finally we mention that for computation of 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 application of the differentiation operator in the coefficient space (see Eq. (29)).

7. *Simultaneous matrix-vector products*

As described above the FMM allows in one run to compute not one, but a sum of four matrix-vector products

$$\Sigma + \lambda \Sigma' = (L_{dense} + \lambda L'_{dense})[q] - (M_{dense} + \lambda M'_{dense})[\phi], \quad (34)$$

for input vectors q and ϕ , which is required for iterative solution of Eq. (10). Also if needed, results for the parts Σ and Σ' can be separated (e.g. for application of Green's identity alone for computation of the potential in the internal domain points). The dense parts of the matrices correspond to decomposition (21), and in the case of use of a simple scheme (24) are the matrices with eliminated diagonals.

IV. COMPUTATION OF SINGULAR ELEMENTS

Despite there exist techniques for 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 semi-analytical formulae), these methods can be costly, and below we propose a technique for approximation of such integrals, which is consistent with the use of the FMM. 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 [22], 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

$$\begin{aligned} L[\sigma] &= \int_{S_j^\epsilon} \sigma(\mathbf{x}) G(\mathbf{x}, \mathbf{y}) dS(\mathbf{x}) + \int_{S \setminus S_j^\epsilon(\mathbf{y}, \epsilon)} \sigma(\mathbf{x}) G(\mathbf{x}, \mathbf{y}) dS(\mathbf{x}) = L_j^\epsilon[\sigma] + \tilde{L}_j[\sigma] \\ M[\sigma] &= \int_{S_j^\epsilon} \sigma(\mathbf{x}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{x})} dS(\mathbf{x}) + \int_{S \setminus S_j^\epsilon(\mathbf{y}, \epsilon)} \sigma(\mathbf{x}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{x})} dS(\mathbf{x}) = M_j^\epsilon[\sigma] + \tilde{M}_j[\sigma]. \end{aligned} \quad (35)$$

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

$$L_j^\epsilon[\sigma] \approx \sigma_j l_j^\epsilon(\mathbf{y}), \quad M_j^\epsilon[\sigma] \approx \sigma_j m_j^\epsilon(\mathbf{y}), \quad (36)$$

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):

$$\gamma \psi = L \left[\frac{\partial \psi}{\partial n} \right] - M[\psi], \quad (37)$$

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

$$\psi(\mathbf{x}) = e^{iks \cdot \mathbf{x}}, \quad q(\mathbf{x}) = \frac{\partial \psi}{\partial n}(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot \nabla e^{iks \cdot \mathbf{x}} = ik \mathbf{n}(\mathbf{x}) \cdot \mathbf{s} e^{iks \cdot \mathbf{x}}, \quad |\mathbf{s}| = 1, \quad (38)$$

which represent plane waves propagating in direction \mathbf{s} . For these functions we have from Eqs. (35)-(37)

$$m_j^\epsilon(\mathbf{y}) - ik(\mathbf{n}_j \cdot \mathbf{s}) l_j^\epsilon(\mathbf{y}) = e^{-iks \cdot \mathbf{x}_j} \left\{ ik \tilde{L} \left[(\mathbf{n} \cdot \mathbf{s}) e^{iks \cdot \mathbf{x}} \right] - \tilde{M} \left[e^{iks \cdot \mathbf{x}} \right] - \gamma(\mathbf{y}) e^{iks \cdot \mathbf{y}} \right\}. \quad (39)$$

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

$$\omega_{j\alpha}(\mathbf{y}) = e^{-iks_\alpha \cdot \mathbf{x}_j} \left\{ ik \tilde{L} \left[(\mathbf{n} \cdot \mathbf{s}_\alpha) e^{iks_\alpha \cdot \mathbf{x}} \right] - \tilde{M} \left[e^{iks_\alpha \cdot \mathbf{x}} \right] - \gamma(\mathbf{y}) e^{iks_\alpha \cdot \mathbf{y}} \right\}, \quad n_{j\alpha} = \mathbf{n}_j \cdot \mathbf{s}_\alpha, \quad (40)$$

we obtain

$$m_j^\epsilon(\mathbf{y}) - ik n_{j\alpha} l_j^\epsilon(\mathbf{y}) = \omega_{j\alpha}(\mathbf{y}), \quad \alpha = 1, \dots, 4. \quad (41)$$

Note then that 4 vectors in the three dimensional space are linearly dependent, i.e. there exist 4 scalars λ_α , such that

$$\sum_{\alpha=1}^4 \lambda_\alpha \mathbf{s}_\alpha = \mathbf{0}, \quad \sum_{\alpha=1}^4 \lambda_\alpha n_{j\alpha} = \mathbf{n}_j \cdot \left(\sum_{\alpha=1}^4 \lambda_\alpha \mathbf{s}_\alpha \right) = 0. \quad (42)$$

Hence, multiplying Eq. (41) by λ_α and summing up them for all α we obtain

$$m_j^\epsilon(\mathbf{y}) = \frac{1}{\lambda} \sum_{\alpha=1}^4 \lambda_\alpha \omega_{j\alpha}(\mathbf{y}), \quad \lambda = \sum_{\alpha=1}^4 \lambda_\alpha. \quad (43)$$

This determines function $m_j^\epsilon(\mathbf{y})$. Let us select, for example, $\mathbf{s}_1, \dots, \mathbf{s}_3$ as coordinate unit vectors, and \mathbf{s}_4 as their normalized sum:

$$\mathbf{s}_1 = \mathbf{i}_x, \quad \mathbf{s}_2 = \mathbf{i}_y, \quad \mathbf{s}_3 = \mathbf{i}_z, \quad \mathbf{s}_4 = \frac{\mathbf{i}_x + \mathbf{i}_y + \mathbf{i}_z}{3^{1/2}}, \quad (44)$$

in this case we have

$$m_j^\epsilon(\mathbf{y}) = \frac{1}{3 - 3^{1/2}} \left[\sum_{\alpha=1}^3 \omega_{j\alpha}(\mathbf{y}) - 3^{1/2} \omega_{j4}(\mathbf{y}) \right]. \quad (45)$$

Furthermore, if we multiply Eq. (41) by $n_{j\alpha}$ and sum up all equations for $\alpha = 1, \dots, 3$, then, in the selected basis we obtain taking into account that \mathbf{n} is a unit vector, $n_{j1}^2 + n_{j2}^2 + n_{j3}^2 = 1$:

$$l_j^\epsilon(\mathbf{y}) = \frac{i}{k} \sum_{\alpha=1}^3 n_{j\alpha} [\omega_{j\alpha}(\mathbf{y}) - m_j^\epsilon(\mathbf{y})]. \quad (46)$$

Obviously, similar consideration can be given to decompose surface operators $L'[\sigma]$ and $M'[\sigma]$:

$$L'[\sigma] = L_j'^\epsilon[\sigma] + \tilde{L}'_j[\sigma], \quad M'[\sigma] = M_j'^\epsilon[\sigma] + \tilde{M}'_j[\sigma], \quad (47)$$

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

We note that these operators are employed only for points on the boundary, so we will use Maue's identity (8) for the internal problem. In this case using test functions (38) we modify Eqs. (39)-(41) as follows

$$\begin{aligned} m_j'^\epsilon(\mathbf{y}) - ik n_{j\alpha} l_j'^\epsilon(\mathbf{y}) &= \omega'_{j\alpha}(\mathbf{y}), \quad \alpha = 1, \dots, 4, \\ \omega'_{j\alpha}(\mathbf{y}) &= e^{-iks \cdot \mathbf{x}_j} \left\{ ik \tilde{L}' \left[(\mathbf{n} \cdot \mathbf{s}_\alpha) e^{iks_\alpha \cdot \mathbf{x}} \right] - \tilde{M}' \left[e^{iks_\alpha \cdot \mathbf{x}} \right] - \gamma_\alpha(\mathbf{y}) e^{iks_\alpha \cdot \mathbf{y}} \right\}, \\ \gamma_\alpha(\mathbf{y}) &= \frac{1}{2} ik (\mathbf{n}(\mathbf{y}) \cdot \mathbf{s}_\alpha), \quad \mathbf{y} \in S. \end{aligned} \quad (48)$$

Then for set of directions (44) solution will be provided by Eqs. (45) and (46), where one should replace $m_j^\epsilon(\mathbf{y})$, $l_j^\epsilon(\mathbf{y})$, and $\omega_{j\alpha}(\mathbf{y})$, with $m_j'^\epsilon(\mathbf{y})$, $l_j'^\epsilon(\mathbf{y})$, and $\omega'_{j\alpha}(\mathbf{y})$, respectively. As we noted above the FMM provides 4 simultaneous matrix-vector multiplications, and so 4 runs of the FMM ($\alpha = 1, \dots, 4$) is sufficient to get all diagonals.

8. Discretization

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

$$L_{jj} = l_j^\epsilon(\mathbf{x}_j), \quad M_{jj} = m_j^\epsilon(\mathbf{x}_j), \quad L'_{jj} = l_j^{\prime\epsilon}(\mathbf{x}_j), \quad M'_{jj} = m_j^{\prime\epsilon}(\mathbf{x}_j), \quad j = 1, \dots, N. \quad (49)$$

In fact, for solution of the BIE 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 of each diagonal entries using the same method as described above.

V. NUMERICAL EXPERIMENTS

The BEM/FMM was implemented in Fortran 95 and computations were performed for different geometries and wavenumbers with an appropriate discretization of the surface by triangular mesh. Some results of computations are reported below.

A. Scattering from a single sphere

Example of scattering of an incident plane wave from a single sphere is valuable for tests of the performance of the method, since analytical solution is available in this case. For the incident field $\phi^{in}(\mathbf{r}) = e^{i\mathbf{k}\mathbf{s}\cdot\mathbf{r}}$, where \mathbf{s} is the unit vector collinear with the wave vector, solution for the total (incident plus scattered field) for impedance boundary conditions can be found elsewhere [18]:

$$\phi|_S(\mathbf{s}') = \frac{i}{(ka)^2} \sum_{n=0}^{\infty} \frac{(2n+1)i^n P_n(\mathbf{s} \cdot \mathbf{s}')}{h'_n(ka) + (i\sigma/k) h_n(ka)}, \quad (50)$$

$$\frac{\partial \phi}{\partial n} + i\sigma \phi \Big|_S = 0,$$

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 we may have for the scattered field Neumann, Dirichlet, or Robin problem.

In the numerical solution for a unit sphere we varied k , discretization, parameter λ in Eq. (10), the boundary admittance, tolerance, and parameters controlling the FMM accuracy and performance. As iterative solver we tried unpreconditioned GMRES and fGMRES with different right preconditioners. A typical configuration and computational result is shown in Fig. 4.

1. Preconditioning

The best preconditioners we found were the preconditioners which compute solution of the system $A\psi_j = c_j$ on the j th step using unpreconditioned GMRES (inner loop), low accuracy FMM, and 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^{-5} , and the iterative process was terminated as the residual reaches 10^{-4} , for the inner preconditioning loop we used FMM with prescribed accuracy 0.05 and the iterative process was terminated at residual value 0.5. The process of approximate solution with such parameters is much faster than that for parameters of the outer loop, as it requires lesser iterations and the matrix-vector product is computed several times faster (lower truncation numbers). On the other hand the use of the preconditioner reduces by the order of magnitude the number of iterations in the outer loop. This is important from the point of view of memory management for large problems. Indeed, the GMRES or fGMRES requires storage of K vectors of length N , where K is the dimensionality of the Krylov subspace. The iterative process becomes much longer if restarts of GMRES are used, so it is preferable to achieve convergence in $N_{iter} \leq K$. In the case of use of GMRES-based preconditioner the storage memory will be of order $(K + K')N$, where K' is the dimensionality of the Krylov subspace for preconditioning. Since both numbers K and K' are much smaller than K required for unpreconditioned GMRES the required memory for solution reduces substantially.

Fig. 5 shows convergence of the unpreconditioned GMRES and preconditioned fGMRES with the FMM-based preconditioner as described above. The computations were made for a sound-hard sphere of radius a , which surface was discretized by 101,402 vertices and 202,808 triangular elements for the relative wavenumber $ka = 50$. As it is seen the number of iterations reduces dramatically with the use of the preconditioner, while due to execution of the inner loop for the preconditioned method, the relative cost of each outer iteration in the fGMRES becomes larger. Nevertheless fast convergence provides finally 2.5 times faster time (computational cost) to solve the same problem. We note also that in the case illustrated the matrix-vector product used low accuracy FMM in preconditioning was computed approximately 6 times faster than the matrix-vector product computed with higher accuracy.

2. Spurious modes

The test case for the sphere is also good to illustrate advantage of the Burton-Miller formulation compared to the boundary integral equation based on Green's identity. According to the theory [2, 7], the Green's identity formulation may result in convergence to a solution, which for sound-hard sphere is true solution plus a non-zero solution of the internal problem corresponding to zero boundary conditions at given wavenumber for the potential. Such solutions are not physical, since the solution of the external scattering problem is unique, and, therefore, they manifest deficiency of the numerical method based on the Green's identity.

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

$$\phi^{int}(\mathbf{r}) = \sum_{n=0}^{\infty} j_n(kr) \sum_{m=-n}^n B_n^m Y_n^m(\theta, \varphi), \quad (51)$$

where (r, θ, φ) are the spherical coordinates of \mathbf{r} , and B_n^m are arbitrary constants. The set of zeros of functions $j_n(ka)$ provides a discrete set of values of ka for which $\phi^{int}|_S = 0$, while ϕ^{int} is not identically zero inside the sphere. The minimum resonant value of ka is the first zero of function $j_0(ka)$, which is $ka = \pi$. So we conducted some numerical tests with Burton-Miller and Green's formulations for a range of ka ($0.01 \leq ka \leq 50$) to check what happens when ka takes a resonant value.

Fig. 6 provides an illustration for case $ka = 3\pi \approx 9.424778$, which is the third zero of function $j_0(ka)$. In case of using of Burton-Miller formulation with some λ , $\text{Im}(\lambda) \neq 0$ solution converges to the solution, which is consistent with the analytical solution (50). However if the same case is computed using the Green's formula solution converges to a function, which is different. We checked that in this case the converged solution can be well approximated ($\sigma = 0$) by

$$\phi|_S(\mathbf{s}') = B + \frac{i}{(ka)^2} \sum_{n=1}^{\infty} \frac{(2n+1)i^n P_n(\mathbf{s} \cdot \mathbf{s}')}{h'_n(ka)}, \quad (52)$$

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 eigen function failed to be determined correctly, which is an expected result. We also note that such type of solution appears if using iterative methods like GMRES, where degeneration of the matrix operator for some subspace does not affect convergence in other subspaces. In case if the problem is solved directly (say the linear system is solved using the LU-decomposition) the system matrix becomes degenerate (or very poorly conditioned), which should result in completely wrong solution.

We also note that in the non-resonant cases (ka differs from the resonant values more than some small $\epsilon(ka)$) the Green's formulation provided a good solution, while normally the number of iterations increased as ka became closer to resonance. For large ka (40 or more) computations using Green's identity become unstable, which is explainable by dense location of zeros of functions $j_n(ka)$. For the Burton-Miller formulation we found experimentally that parameter λ selected as

$$\lambda = \frac{i\eta}{k}, \quad (53)$$

where $\eta = 0.01 - 0.1$ (for the case illustrated above we used $\eta = 0.03$) provides good results for the range of parameters studied, while for $ka < 1$ the number of iterations increases compared to the Green's identity. So for such low ka , when there are certainly no resonances, Green's identity can be recommended. Increase of this parameter usually decreases the accuracy of computations, since more weight is put on the hypersingular part of the integral equation, while decrease of the parameter for large ka leads to the increase in the number of iterations, and for $\eta \ll 0.01$ the Burton-Miller integral equation shows the problems of spurious modes.

3. Performance

By performance we mean how the above algorithm scales with respect to the number of elements in the mesh and the relative size of the computational domain. For the FMM, which uses cubic boxing, the characteristic scale is usually based on the diagonal (maximum size) of the computational domain, D . So kD is an important dimensionless parameter. Further, we can compute the maximum size of the boundary element, which for triangular mesh is the maximum side of the triangle, d , which 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 $N \sim S/d^2 \sim D^2/d^2$. Formal constraint for discretizations used for accurate solution of the Helmholtz equation is $d/\lambda_a \ll 1$, where $\lambda_a = 2\pi/k$ is the acoustic wavelength. In practice we replace this condition with $kd < \chi$, where χ is some constant of order 1, so we have not less than $2\pi/\chi$ mesh elements per wavelength. This number usually varies in range 5-10. This shows that the total number of elements should be $N \sim D^2/d^2 \gtrsim (kD)^2/\chi^2 = O((kD)^2)$ (note that we dropped here factors like 4π which add an extra order of magnitude, also the estimates do not take into account the non-uniformity of real meshes).

Fig.7 shows results of numerical experiments for scattering from a sound-hard sphere, where

we fixed parameter $\chi \approx 0.94$ for $kD > 30$ which provided at least 6 elements per wavelength and increased the number of elements proportionally to $(kD)^2$ (the maximum case plotted corresponds to $kD = 173$ where the mesh contains 135,002 vertices and 270,000 elements). The total CPU time required for solution of the problem is scaled approximately as $O((kD)^3)$ which is consistent with the theory if the number of iterations is constant. In fact, for the cases computed the number of iterations both in the inner and outer loops of the preconditioned fGMRES was growing with kD , however this growth was not very substantial (e.g. for the outer loop 8 iterations for $kD = 10$, while 14 iterations for $kD = 173$ to converge to the value of the residual 10^{-4}). The total number of iterations in the inner loop grow also, but substantially because of the growth of the number of iterations in the outer loop, while the average number of inner iterations per one outer iteration was varying in range (1-8) (in any case the inner loop was limited by maximum 10 iterations). The overall accuracy was controlled by comparison with the analytical solution (50) and in most cases the maximum absolute error (L_∞ -norm) was in the range 1-2 % at $kD > 3$. In range $3 < kD < 30$ the error increases a couple of times if we keep constant $\chi \approx 0.94$. We decreased this parameter to maintain the same error. For $kD < 3$, which we characterize as very low frequency regime, we used constant mesh with 866 vertices and 1728 elements. This mesh provided errors in range $10^{-4} - 10^{-2}$ for all cases. Also for this range we used $\lambda = 0$. This provided faster convergence, despite selection based on Eq. (53) with $\eta = 0.03$ also provided accurate results. In fact, at low frequencies the discretization can be reduced further to have consistent 1% accuracy, but since we tried to test the FMM, which works efficiently only for problem sizes $N \gtrsim 10^3$ we fixed the discretization in this range. Another acceleration (several times) comes from precomputation and storage of the near-field integrals in the BEM (matrix A_{sparse} , Eq. (21)). However, based on the RAM (we used 3.5 GB) this works only for N below 10^5 . So to show scaling we did not use such storage and recomputed the sparse matrix entries each time as the respective matrix-product was needed.

B. More complex shapes

Many problems in acoustics require computations for substantially complex shapes, which includes bioacoustics, human hearing, sound propagation in dispersed media, engine acoustics, room acoustics, etc. We tested our algorithm by solving several problems like that and Fig. 8 provides an idea on the sizes and geometries we were working on. We note that modeling of complex shapes

requires surface discretizations which is 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 solution of the Laplace equation ($k = 0$) the boundary element methods can use thousands and millions of elements 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.

For the last geometry illustrated in Fig. 8 we conducted some study on the method performance and accuracy for the range of kD from 0.35 to 175 ($ka = 0.01 - 5$, where a is the largest axis of an ellipsoid). First, we should note that 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. Indeed, we checked that the convergence for the Neumann, Dirichlet, and Robin problems was very fast (just a few iterations) for small ka , where, as we discussed above the use of low-frequency FMM is important. For this problem both formulations, based on the Green's identity and Burton-Miller equation are acceptable, while instead of scaled value (53) of the regularization parameter in the latter formulation, some small constant value is more appropriate at small ka (in fact at $ka \ll 1$ there is no any internal resonance modes, so Green's identity works well). The convergence was not affected by the increase of the number of nodes, and, in principle, discretization with the number of elements of order 100 was in this sense acceptable. Nonetheless we should increase discretizations, since the accuracy of computations suffered from poor shape representation (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). The quality of the mesh is also important (our mesh was obtained by simple mapping of a regular mesh on a cube surface to the ellipsoid surface). For larger discretizations as we used we were able achieve $\sim 1\%$ relative errors in strong norm (L_∞) for the range of parameters we used.

Fig. 9 shows an absolute relative error at each vertex

$$\epsilon_i = \frac{|\phi_i^{(BEM)} - \phi_i^{(an)}|}{|\phi_i^{(an)}|}, \quad i = 1, \dots, N_{vert}, \quad (54)$$

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

VI. CONCLUSION

We presented here a version of the FMM accelerated BEM, 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 the FMM for efficient BEM requires different schemes for treatment of low and high frequency regions, and switching from multipole to signature function representation of solution of the Helmholtz equation is important for broadband BEM. The tests of the methods for simple and complex shapes show that it can be used for efficient solution of scattering and other acoustical problems encountered in practice for a wide range of frequencies.

VII. ACKNOWLEDGEMENTS

This work was partially supported by a Maryland Technology Development Corporation grant to the authors.

-
- [1] Abramowitz, M., and Stegun, I. A. (1964) *Handbook of Mathematical Functions* (National Bureau of Standards, Washington, D.C.).
 - [2] Burton, A. J. and Miller, G. F. (1971). "The application of the integral equation methods to the numerical solution of some exterior boundary-value problems," Proc. Roy. Soc. London, Series A, Math. Phys. Sci. **323**, 201–210.
 - [3] Chandrasekhar, B., and Rao, S. M. (2004). "Acoustic scattering from rigid bodies of arbitrary shape—double layer formulation," J. Acoust. Soc. Am., **115**, 1926-1933.
 - [4] Chen, L. H., and Zhou, J. (1992). *Boundary Element Methods* (Academic Press, New York).
 - [5] Cheng, H., Crutchfield, W. Y., Gimbutas, Z. , Greengard, L. F. , Ethridge, J. F., Huang, J., Rokhlin, V., Yarvin, N., and Zhao, J. (2006). "A wideband fast multipole method for the Helmholtz equation in three dimensions," J. Comput. Phys., **216**, 300-325.

- [6] Chew, W. C., Jin, J. M., Michelssen, E., and Song, J. (eds.) (2001). *Fast and Efficient Algorithms in Computational Electromagnetics* (Artech House, Boston).
- [7] Colton, D., and Kress, R. (1998). *Inverse Acoustic and Electromagnetic Scattering Theory* (Springer-Verlag, Berlin).
- [8] Darve, E. (2000). “The fast multipole method: numerical implementation, J. Comput. Phys., **160**, 195–240.
- [9] Darve, E. (2000). “The fast multipole method (I): error analysis and asymptotic complexity,” SIAM J. Numer. Anal., **38**, 98–128.
- [10] Darve, E., and Havé, P. (2004). “Efficient fast multipole method for low-frequency scattering,” J. Comput. Phys., **197**(1), 341–363.
- [11] Driscoll, J. R., and Healy, D. M., Jr. (1994) “Computing Fourier transforms and convolutions on the 2-sphere,” Advances in Appl. Math., **15**, 202-250.
- [12] Epton, M. A., and Dembart, B. (1995) “Multipole translation theory for the three-dimensional Laplace and Helmholtz equations,” SIAM J. Sci. Comput., **16**, 865-897.
- [13] Fischer, M. (2004) *The Fast Multipole Boundary Element Method and its Application to Structure-Acoustic Field Interaction* (Ph.D. dissertation: Institut A für Mechanik der Universität Stuttgart). (Available at <http://elib.uni-stuttgart.de/opus/volltexte/2004/1959/pdf/diss.pdf>; Accessed 12/30/07).
- [14] Fischer, M., Gauger, U., and Gaul, L. (2004) “A multipole Galerkin boundary element method for acoustics,” Engg. Anal. with Boundary Elements, **28**, 155–162.
- [15] Greengard, L. (1988). *The Rapid Evaluation of Potential Fields in Particle Sytems* (MIT Press, Cambridge, MA).
- [16] Greengard, L., Huang, J., Rokhlin, V., and Wandzura, S. (1998). “Accelerating fast multipole methods for the Helmholtz equation at low frequencies,” IEEE Comput. Sci. Engrg., **5**(3), 32-38.
- [17] Gumerov, N. A., and Duraiswami, R. (2003). “Recursions for the computation of multipole translation and rotation coefficients for the 3-D Helmholtz equation,” SIAM J. Sci. Stat. Comput., **25**(4), 1344-1381.
- [18] Gumerov, N. A. and Duraiswami, R. (2004). *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions* (Elsevier, Oxford, UK).
- [19] Gumerov, N. A. and Duraiswami, R. (2005). “Computation of scattering from clusters of spheres using the fast multipole method,” J. Acoust. Soc. Am., **117**(4), 1744-1761.
- [20] Kirkup, S. (2004). *The Boundary Element Method in Acoustics* (Available online at <http://www.boundary-element-method.com/acoustics/index.htm>; Accessed 12/30/07).
- [21] Nishimura, N. (2002). “Fast multipole accelerated boundary integral equation methods,” Appl Mech. **55**, 299-324.
- [22] Lutz, E, Ingrassia, A. R., and Gray, L. J. (1992). “Use of ‘simple solutions’ for boundary integral equation methods in elasticity and fracture analysis,” Int. J. Num. Meth. Engg., **35**, 1737-1751.

- [23] Rokhlin, V. (**1993**). “Diagonal forms of translation operators for the Helmholtz equation in three dimensions,” *Appl. and Comp. Harmonic Analysis*, **1**, 82-93.
- [24] Saad, Y. (**1993**). “A flexible inner-outer preconditioned GMRES algorithm,” *SIAM J. Sci. Comput.*, **14**(2), 461-469, 1993.
- [25] Saad, Y., and Schultz, M. H. (**1986**). “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM J. Sci. Comput.*, **7**, 856-869.
- [26] Sakuma, T., and Yasuda, Y. (**2002**). “Fast multipole boundary element method for large-scale steady-state sound field analysis, part I: Setup and validation”, *Acustica/Acta Acustica*, **88**, 513–525.
- [27] Schneider, S. (**2003**). “Application of fast methods for acoustic scattering and radiation problems,” *J. Comput. Acoustics*, **11**(3), 387–401.
- [28] Shen, L. and Liu, Y. J. (**2007**). “An adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton–Miller formulation,” *Comput. Mech.*, **40**, 461–472.
- [29] Yasuda, Y., and Sakuma, T. (**2003**). “Fast multipole boundary element method for large-scale steady-state sound field analysis, part II: Examination of numerical items”, *Acustica/Acta Acustica*, **89**, 28–38.
- [30] Yasuda, Y., and Sakuma, T. (**2005**). “An effective setting of hierarchical cell structure for the fast multipole boundary element method,” *J. Comput. Acoustics*, **13**(1), 47-70.

List of Figure Captions

Figure 1. Two schemes of collocation based on a triangular mesh: collocation nodes at the panel centers and at the vertices.

Figure 2. Dependences of the truncation number p on the dimensionless domain size ka for different prescribed accuracies of the FMM ϵ according to Eq. (25) ($\delta = 2$) (solid lines). The dashed lines show the high-frequency asymptotics $p_{high}(ka)$. 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.

Figure 3. Illustration comparing the wideband FMMs of Cheng et al (2006) and that presented in this paper for a problem in which the FMM octree has 4 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. The function in each box represented via multipole expansion (S), local expansion (R), far field signature function samples (F), or exponential form for each coordinate direction (E). The “glued” boxes mean that for a given box 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. S_p and S_p^{-1} denote forward and inverse spherical transform, S|E and E|R are respective conversion operators. F|F+i and F|F+f mean that the translation should be accompanied by use of an interpolation or filtering procedure.

Figure 4. Typical BEM computations of the scattering problem. The graph shows comparison between the analytical solution (50) and BEM solution for the vicinity of the rear point of the sphere for $ka = 30$.

Figure 5. 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, $\lambda = 6 \cdot 10^{-4}i$.

Figure 6. 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 $ka = 9.424778$ (triangular mesh 15,002 vertices and 30,000 elements). Analytical solution is shown by the circles.

Figure 7. The CPU time (Xeon 3.2 GHz) for the complete run of the BEM program (cicles), the number of iterations in the outer (triangles), and total number of iterations in the inner (squares) loops of the preconditioned fGMRES vs parameter kD . All cases for $kD > 30$ were computed with the same $\chi = kd \approx 0.94$, while for cases $kD < 3$ the mesh was fixed. The solid line shows dependence $y = ax^3$ in log-log coordinates.

Figure 8. Examples of test problems solved with the present version of the BEM: human head-torso, and bunny models (7.85 kHz and 25 kHz acoustic sources located inside the objects, $kD = 110$ and 96, respectively), and plane wave scattering by 512 randomly oriented ellipsoids ($kD = 29$).

Figure 9. Error in the boundary condition at each vertex for the case of the ellipsoids in Fig. 8.

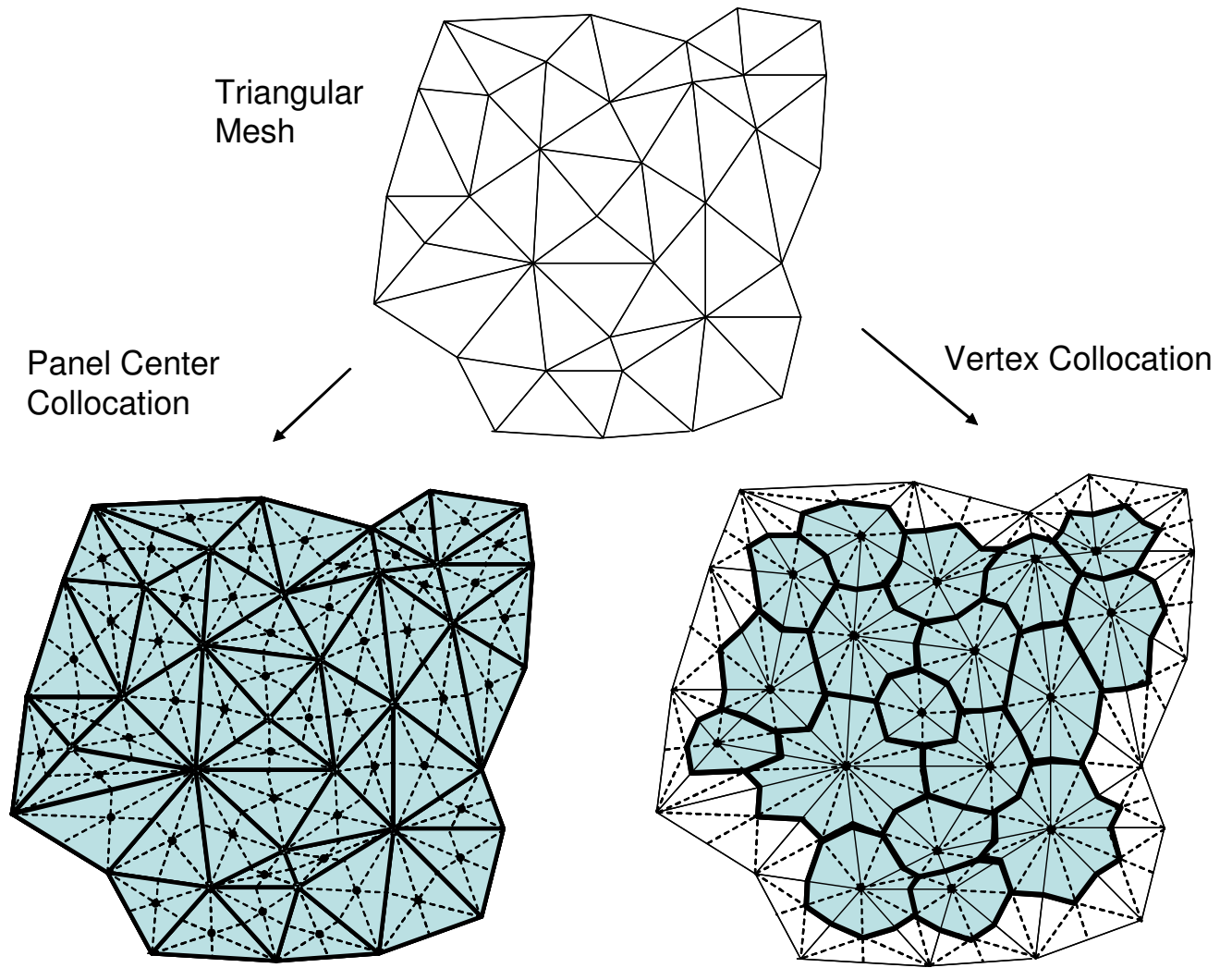


FIG. 1: Two schemes of collocation based on a triangular mesh: collocation nodes at the panel centers and at the vertices.

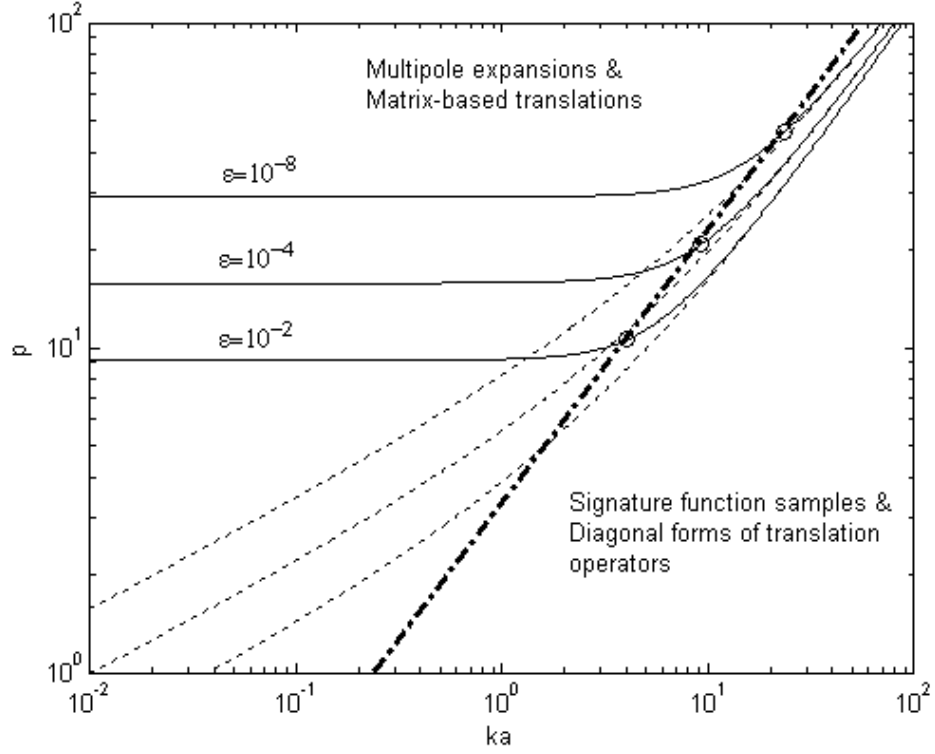


FIG. 2: Dependences of the truncation number p on the dimensionless domain size ka for different prescribed accuracies of the FMM ϵ according to Eq. (25) ($\delta = 2$) (solid lines). The dashed lines show the high-frequency asymptotics $p_{high}(ka)$. The circles mark the points of switch from function 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.

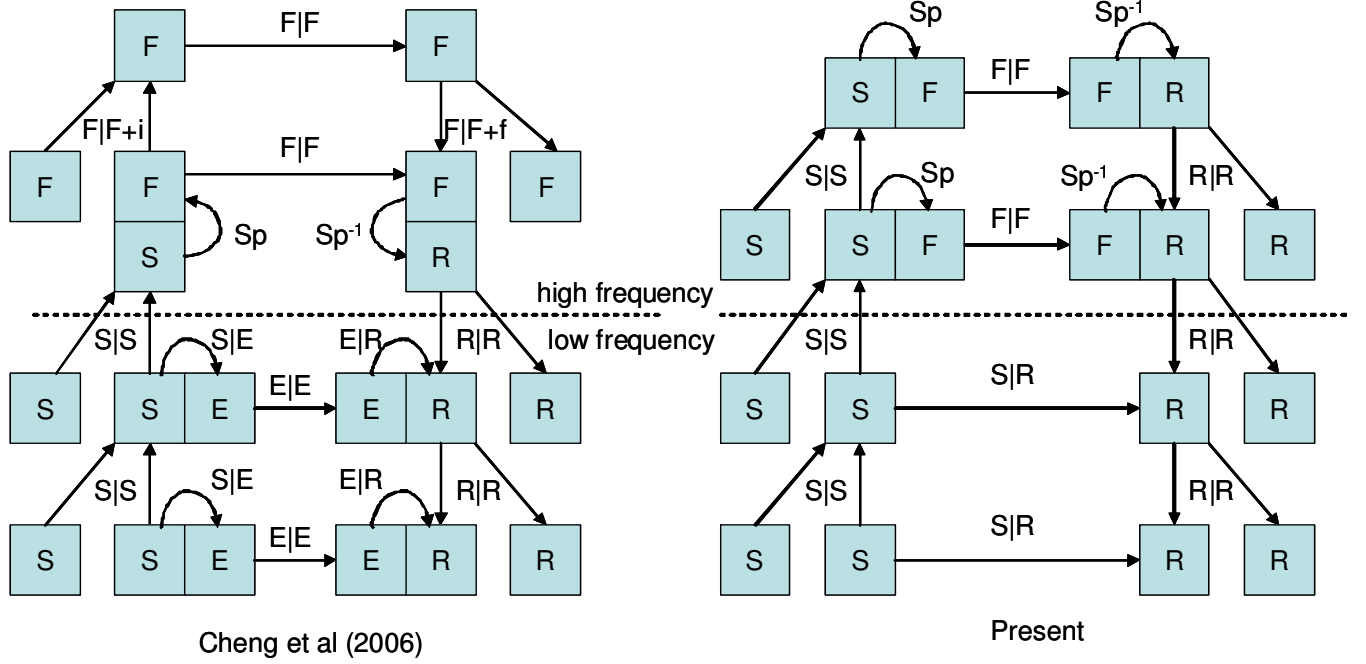


FIG. 3: Illustration comparing the wideband FMMs of Cheng et al (2006) and that presented in this paper for a problem in which the FMM octree has 4 levels, and in which the high-low frequency switch threshold occurs between levels 2 and 3. The left hand side for each algorithm shows the FMM upward pass, while the right hand side shows the FMM downward pass. The function in each box represented via multipole expansion (S), local expansion (R), far field signature function samples (F), or exponential form for each coordinate direction (E). The “glued” boxes mean that for a given box 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. S_p and S_p^{-1} denote forward and inverse spherical transform, $S|E$ and $E|R$ are respective conversion operators. $F|F+i$ and $F|F+f$ mean that the translation should be accompanied by use of an interpolation or filtering procedure.

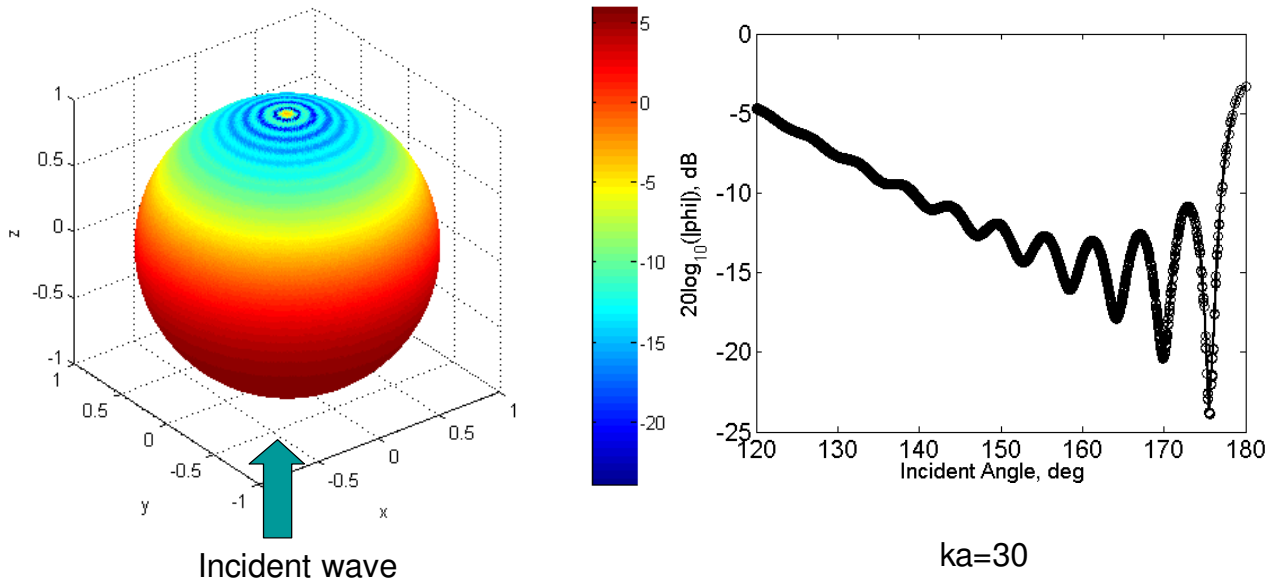


FIG. 4: Typical BEM computations of the scattering problem. The graph shows comparison between the analytical solution (50) and BEM solution for the vicinity of the rear point of the sphere for $ka = 30$.

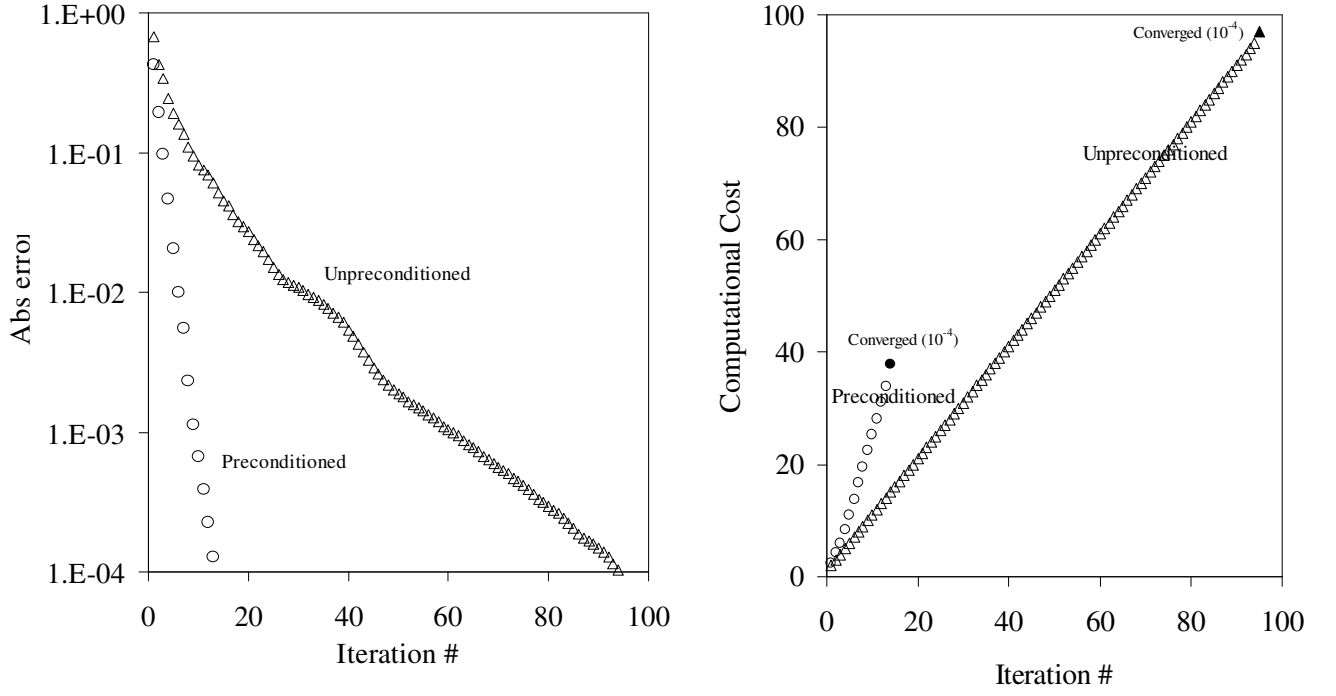


FIG. 5: Left: The absolute error in the residual in the unpreconditioned GMRES (triangles) and in the preconditioned fGMRES (circles) as a function of the number of iterations (outer loop for the fGMRES). Right: the relative computational cost to achieve the same error in the residual for these methods (1 cost unit = 1 iteration using the unpreconditioned method). Computations for sphere, $ka = 50$ for mesh with 101,402 vertices and 202,808 elements, $\lambda = 6 \cdot 10^{-4}i$.

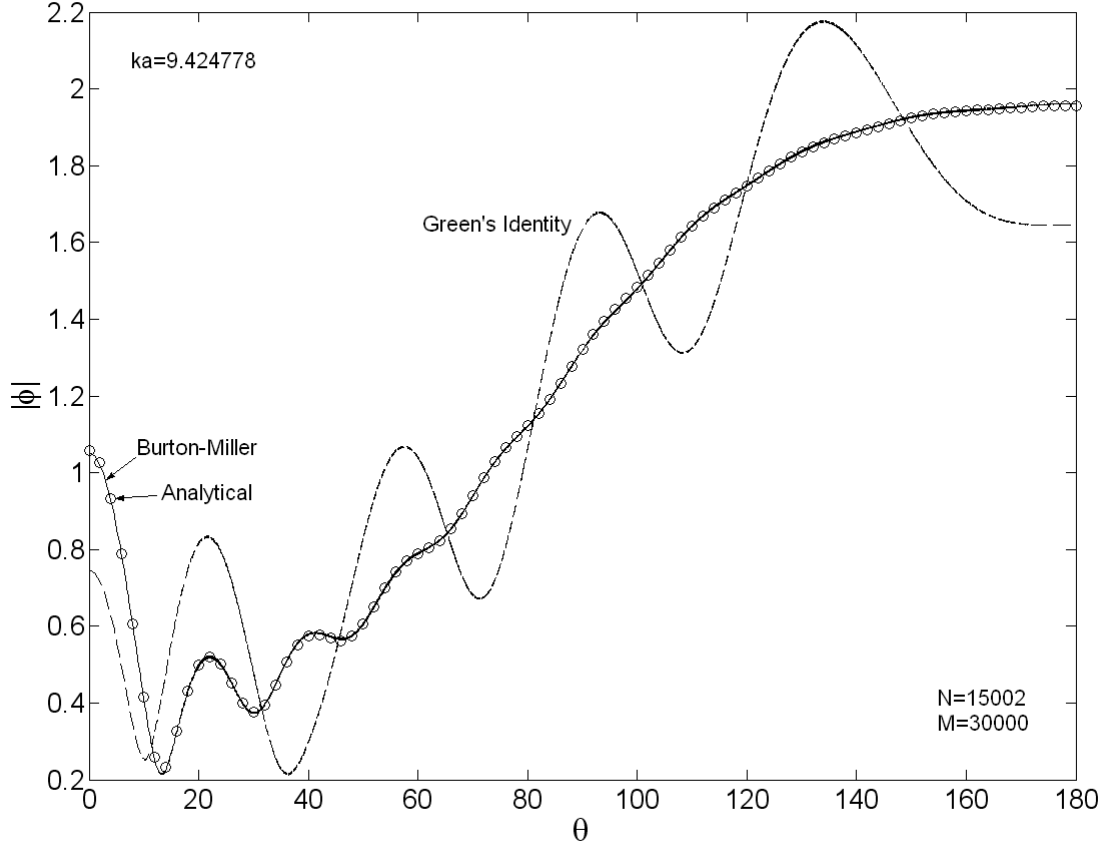


FIG. 6: Solutions of the plane wave scattering problem obtained using BEM with the Burton-Miller and Green's identity formulations and fGMRES iterator for a sphere at resonance $ka = 9.424778$ (triangular mesh 15,002 vertices and 30,000 elements). Analytical solution is shown by the circles.

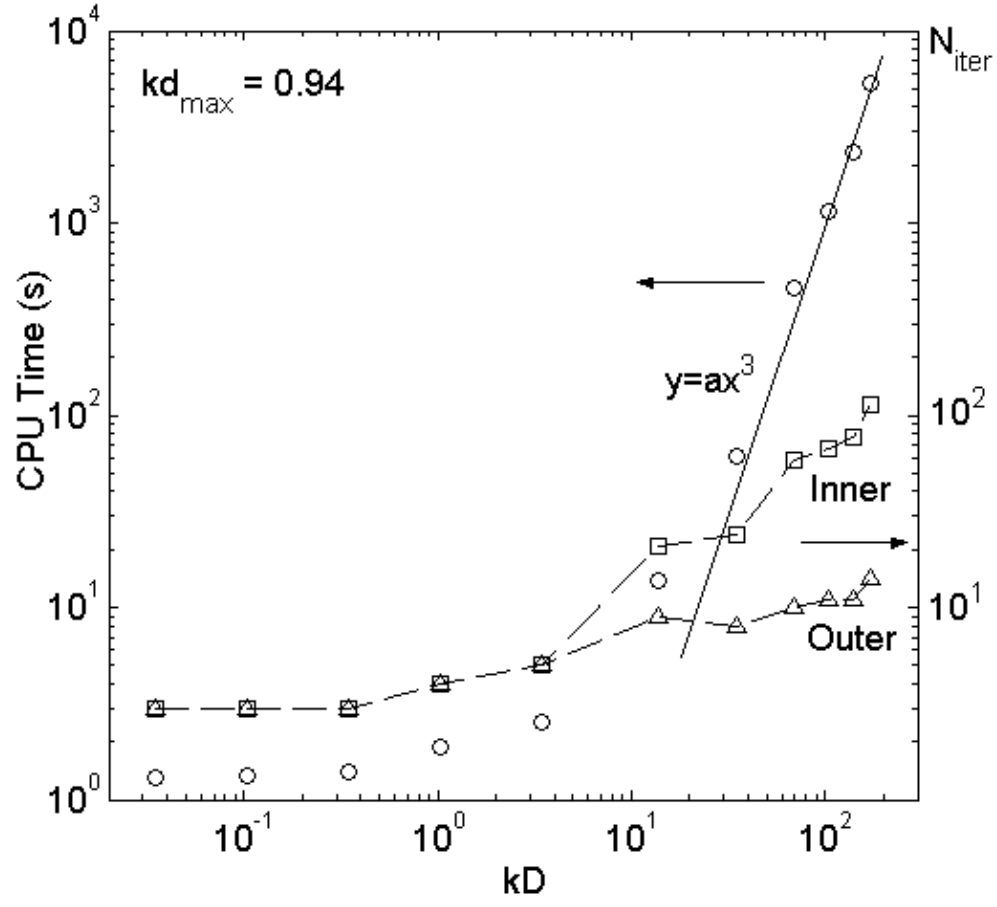


FIG. 7: The CPU time (Xeon 3.2 GHz) for the complete run of the BEM program (cicles), the number of iterations in the outer (triangles), and total number of iterations in the inner (squares) loops of the preconditioned fgmres vs parameter kD . All cases for $kD > 30$ were computed with the same $\chi = kd \approx 0.94$, while for cases $kD < 3$ the mesh was fixed. The solid line shows dependence $y = ax^3$ in log-log coordinates.

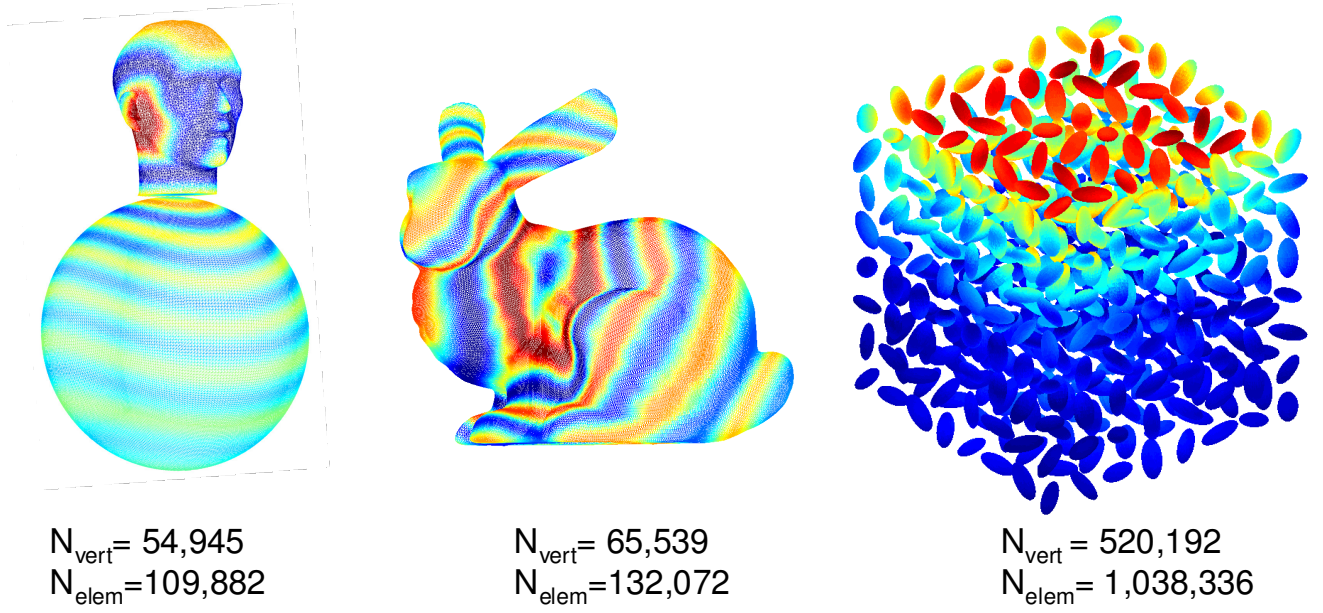


FIG. 8: Examples of test problems solved with the present version of the BEM: human head-torso, and bunny models (7.85 kHz and 25 kHz acoustic sources located inside the objects, $kD = 110$ and 96, respectively), and plane wave scattering by 512 randomly oriented ellipsoids ($kD = 29$).

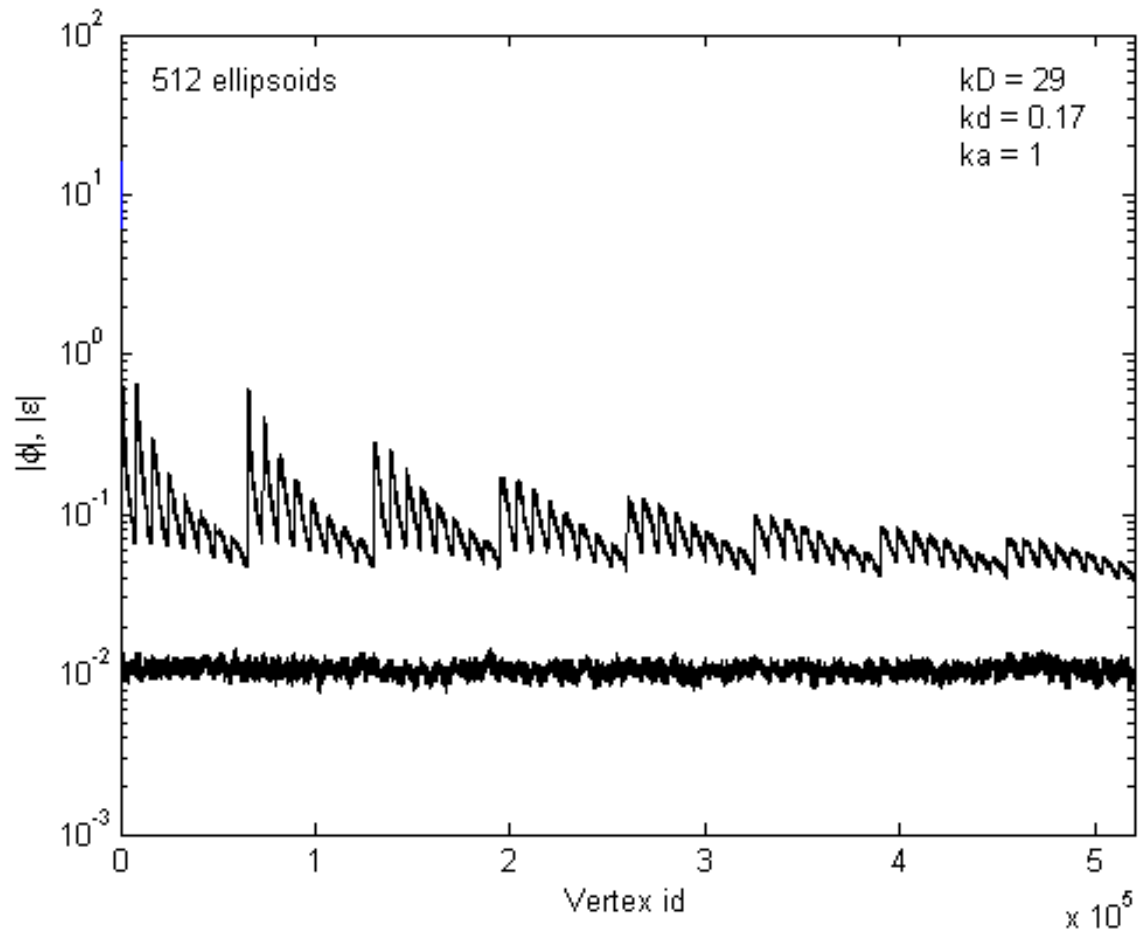


FIG. 9: Error in the boundary condition at each vertex for the case of the ellipsoids in Fig. 8.