

A fast directional BEM for large-scale acoustic problems based on the Burton–Miller formulation



Yanchuang Cao^a, Lihua Wen^{a,b,*}, Jinyou Xiao^{a,b}, Yijun Liu^{b,c}

^a School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, PR China

^b Institute for Computational Mechanics and Its Applications, Northwestern Polytechnical University, Xi'an 710072, PR China

^c Department of Mechanical Engineering, University of Cincinnati, Cincinnati, OH 45221-0072, USA

ARTICLE INFO

Article history:

Received 21 February 2014

Received in revised form

8 June 2014

Accepted 23 July 2014

Keywords:

Fast directional algorithm

Boundary element method

Burton–Miller formulation

Nyström method

ABSTRACT

In this paper, a highly efficient fast boundary element method (BEM) for solving large-scale engineering acoustic problems in a broad frequency range is developed and implemented. The acoustic problems are modeled by the Burton–Miller boundary integral equation (BIE), thus the fictitious frequency issue is completely avoided. The BIE is discretized by using the Nyström method based on the curved quadratic elements, leading to simple numerical implementation (no edge or corner problems) and high accuracy in the BEM analysis. The linear systems are solved iteratively and accelerated by using a newly developed kernel-independent wideband fast directional algorithm (FDA) for fast summation of oscillatory kernels. In addition, the computational efficiency of the FDA is further promoted by exploiting the low-rank features of the translation matrices, resulting in two- to three-fold reduction in the computational time of the multipole-to-local translations. The high accuracy and nearly linear computational complexity of the present method are clearly demonstrated by typical examples. An acoustic scattering problem with dimensionless wave number kD (where k is the wave number and D is the typical length of the obstacle) up to 1000 and the degrees of freedom up to 4 million is successfully solved within 10 h on a computer with one core and the memory usage is 24 GB.

© 2014 Published by Elsevier Ltd.

1. Introduction

The boundary element method (BEM), besides its wide applications in other branches of science and engineering, has been an important numerical method in acoustics. This is mainly due to its unique advantages in dimension reduction, solution accuracy and treating infinite and semi-infinite domain problems where the radiation condition at infinity is automatically satisfied. The capability of the acoustic BEM has been further improved when it is combined with the famous Burton–Miller formulation [1] or the combined Helmholtz integral equation formulation (CHIEF) [2], in which the annoying non-uniqueness problem of the conventional Helmholtz boundary integral equation (BIE) can be successfully circumvented. Among them the Burton–Miller formulation is more attractive since it does not need to choose interior points by users as the CHIEF method [3].

The traditional BEM, however, cannot be used for large-scale numerical simulation, because of the densely populated system matrices and thus the square scaling of computational cost with

respect to N , the degrees-of-freedom (DOF). Fortunately, this limitation has now been removed to a large extent by various BEM acceleration techniques emerged in the past three decades. The representative examples are the fast multipole method (FMM) [4–6], \mathcal{H} -matrix [7], wavelet compression [8,9], pre-corrected FFT [10,11], ACA [12,13], etc. Of those acceleration techniques, it is the FMM that has found the most substantial applications in many areas [14], such as acoustics [15,16], electromagnetics and elastodynamics [17].

The original FMM was first proposed by Greengard and Rokhlin [4] to accelerate the evaluation of interactions of large ensembles of particles governed by Laplace equation. Its success hinges on the observation that the interaction via the kernel between well-separated sets of points is approximately of low rank. In the low frequency regime, the low rank property still holds for the Helmholtz kernels, thus the FMM for the Laplace equation is applicable to the Helmholtz equation with slight modifications [14]. The complexity of the resulting low frequency FMM is of order $O(N)$ for a given accuracy. In the high frequency regime, however, the situation is drastically different as the low rank assumption is not valid any more. In fact, the approximate rank of the translation matrix grows linearly with the size of the point sets (in terms of the wavelength). However, Rokhlin [18] observed that the translation matrix between well-separated point sets, though

* Corresponding author at: School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, PR China.

E-mail addresses: caoyanch@126.com (Y. Cao), lhwen@nwpu.edu.cn (L. Wen), xiaojy@nwpu.edu.cn (J. Xiao), Yijun.Liu@uc.edu (Y. Liu).

of large rank, can be expressed in “diagonal form” in Fourier space. This observation leads to a high frequency FMM with $O(N \log N)$ complexity. Nevertheless, this algorithm is only suitable for relatively high frequency; it becomes numerically unstable in the low frequency regime. The desirability for a FMM that is accurate and efficient in a broad frequency range, from low frequency to high frequency, gives rise to the wideband FMM in [19,20], which are essentially hybrid schemes of the FMMs in low and high frequency regimes. However, besides the success in dealing with both non-oscillatory and oscillatory kernels, one should note that the implementation of the original FMM is highly technical, mathematically involved and kernel-dependent. This poses a severe limitation on its engineering application and its generalization to other problems.

With this background, the kernel-independent FMM (KIFMM) has gain its popularity in recent years. The main advantage of the KIFMM lies in that it requires no complicated analytic expansions for the kernels, but only uses kernel evaluations in its implementation; see e.g. [21–23] and the references therein. For low frequency problems, we would like to mention two KIFMM algorithms proposed by Ying et al. [21] and Fong and Darve [22], both of which have attracted much attention in the past several years. In [21] the analytic expansions of the original FMM are replaced with equivalent density representations, while in [22] they are replaced with Chebyshev polynomial interpolations. For applications of the algorithm in [21], see e.g., [24,25]; recent developments of the algorithm in [22] can be found in [26,27].

The KIFMM for high frequency problems are built upon the directional low rank property of the oscillatory kernels [28]. Two representative examples along the this line are the fast directional algorithm (FDA) proposed by Engquist and Ying [29] which uses equivalent density representations (as in [21]), as well as the one proposed by Messner et al. [27] which uses Chebyshev polynomial interpolations (as in [22]). The readers are referred to [27] for the restrictions and advantages of the FDA, and to [26] for its efficiency enhancement.

This paper will concentrate on the FDA in [29]. The good accuracy and efficiency ($O(N \log N)$ complexity) of the algorithm have been proved theoretically and numerically in [29]. Furthermore, the algorithm can be adapted to handle kernels other than the Helmholtz kernel quite easily, which is not true for most other existing algorithms. The algorithm was used to accelerate the BEM for electromagnetic scattering problems by Tsuji and Ying in [30]. However, to the best of the authors' knowledge, no reported work is devoted to the application of the algorithm to accelerating BEM for the Burton–Miller BIE, which is essential for solving real-world acoustic problems.

There are two contributions in this paper. The first one is to adapt the FDA in [29] to accelerate the BEM (the resulting method will be called fast directional BEM, or FDBEM for short, from now on) for the Burton–Miller BIE; see Section 3. The advantages of the FDBEM for Burton–Miller BIE are the following: (1) it does not require complicated analytical expressions for the far-field approximations, but only uses kernel evaluations, which results in a simpler implementation and ease of use; (2) the algorithm can handle the kernels of the Burton–Miller formulation in a similar manner. Therefore, it is more user-friendly than the other FMM algorithms.

The second contribution is achieved by further acceleration for the translations in the FDA. It is based on the observation that generally the ranks of most of the M2L translation matrices are much lower than their dimensions. The improvement of the algorithm is implemented by the following two steps: (1) compressing all the translation matrices into more compact forms, and (2) approximating the compressed M2L matrices by low rank representations, see Section 4. Numerical experiments show that

the computational time for M2L translations can be reduced by a factor of 2–3, and that for the upward and downward passes can be reduced by about 40%, the memory consumption is reduced by about 30%, see Section 5.

2. Burton–Miller formulation and its Nyström discretization

2.1. Acoustic Burton–Miller formulation

The time harmonic acoustic waves in a homogenous and isotropic acoustic medium Ω is described by the following Helmholtz equation:

$$\nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \Omega, \quad (1)$$

where ∇^2 is the Laplace operator, $u(\mathbf{x})$ is the velocity potential at the point $\mathbf{x} = (x_1, x_2, x_3)$ in the physical coordinate system, $k = \omega/c$ is the wave number, with ω being the angular frequency and c being the sound speed. The boundary conditions can be any combinations of the Dirichlet, Neumann or Robin boundary conditions. For exterior problems, the Sommerfeld condition at infinity has to be satisfied as well.

By using Green's second theorem, the solution of Eq. (1) can be expressed by integral representation

$$u(\mathbf{x}) + \int_{\Gamma} \frac{\partial G}{\partial \mathbf{n}_{\mathbf{y}}}(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) \, d\mathbf{y} = \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) \, d\mathbf{y} + u^{\text{inc}}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma, \quad (2)$$

where \mathbf{x} denotes the field point and \mathbf{y} denotes the source point on the boundary Γ ; $\mathbf{n}_{\mathbf{y}}$ denotes the unit normal vector at the source point \mathbf{y} ; $q(\mathbf{y}) = \partial u(\mathbf{y}) / \partial \mathbf{n}_{\mathbf{y}}$ is the normal gradient of velocity potential, i.e., the normal vibrating velocity on \mathbf{y} . The incident wave $u^{\text{inc}}(\mathbf{x})$ will not be presented for radiation problems. The three-dimensional (3D) fundamental solution G is given as

$$G(\mathbf{x}, \mathbf{y}) = \frac{e^{ikr}}{4\pi r}, \quad (3)$$

with $r = |\mathbf{x} - \mathbf{y}|$ being the Euclidean distance between the source and the field points, and $i = \sqrt{-1}$ being the imaginary unit.

Before presenting the BIEs it is convenient to introduce the associated single, double, adjoint and hypersingular layer operators which are denoted by S , \mathcal{D} , \mathcal{M} and \mathcal{H} , respectively, that is,

$$S q(\mathbf{y}) = \int_{\Gamma} G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) \, d\mathbf{y}, \quad (4a)$$

$$\mathcal{D} u(\mathbf{y}) = \int_{\Gamma} \frac{\partial G}{\partial \mathbf{n}_{\mathbf{y}}}(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) \, d\mathbf{y}, \quad (4b)$$

$$\mathcal{M} q(\mathbf{y}) = \int_{\Gamma} \frac{\partial G}{\partial \mathbf{n}_{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) \, d\mathbf{y}, \quad (4c)$$

$$\mathcal{H} u(\mathbf{y}) = \int_{\Gamma} \frac{\partial^2 G}{\partial \mathbf{n}_{\mathbf{x}} \partial \mathbf{n}_{\mathbf{y}}}(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) \, d\mathbf{y}. \quad (4d)$$

The operator S is weakly singular and the integral is well-defined, while the operators \mathcal{D} and \mathcal{M} are defined in Cauchy principal value sense (CPV). The operator \mathcal{H} , on the other hand, is hypersingular and unbounded as a map from the space of smooth functions on Γ to itself. It should be interpreted in the Hadamard finite part sense (HFP). Denoting a vanishing neighbourhood surrounding \mathbf{x} by Γ_ϵ , the CPV and HFP integrals are those after extracting free terms from a limiting process to make Γ_ϵ tends to zero in deriving BIEs [31].

If the field point \mathbf{x} approaches the boundary Γ , Eq. (2) becomes the conventional BIE (CBIE)

$$c(\mathbf{x}) u(\mathbf{x}) + \mathcal{D} u(\mathbf{x}) = S q(\mathbf{x}) + u^{\text{inc}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \quad (5)$$

where $c(\mathbf{x})$ is the free term coefficient which equals to 1/2 on smooth boundary. By taking the normal derivative of Eq. (2) and

letting the field point \mathbf{x} going to the boundary Γ , one obtains the hypersingular BIE (HBIE)

$$c(\mathbf{x})q(\mathbf{x}) + \mathcal{H}u(\mathbf{x}) = \mathcal{M}q(\mathbf{x}) + q^{\text{inc}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma. \quad (6)$$

Both CBIE and HBIE can be applied to solve the unknown boundary values of interior problems. For exterior problems, they have different set of fictitious frequencies at which a unique solution cannot be obtained. However, Eqs. (5) and (6) will always have only one solution in common. Given this fact, the Burton–Miller formulation which is a linear combination of Eqs. (5) and (6) (CHBIE) should yield unique solutions for all frequencies

$$c(\mathbf{x})u(\mathbf{x}) + (\mathcal{D} + \alpha\mathcal{H})u(\mathbf{y}) - u^{\text{inc}}(\mathbf{x}) = (S + \alpha\mathcal{M})q(\mathbf{y}) - \alpha[c(\mathbf{x})q(\mathbf{x}) - q^{\text{inc}}(\mathbf{x})], \quad \mathbf{x} \in \Gamma, \quad (7)$$

where α is a coupling constant that can be chosen as i/k [1].

2.2. Nyström boundary element discretization

In this paper, the Nyström method is used to discretize the Burton–Miller equation. We choose to use the Nyström method because the resultant linear systems are more like a summation, namely, the far-field part of the system matrix is just the values of the integral kernels. As such, many fast methods, like the FMM and the FDA, which are primarily proposed for accelerating summations, can be applied to BEM acceleration with minor modifications.

The implementation of the Nyström BEM follows the work in [32]. The boundary Γ is partitioned into N_e curved triangular quadratic elements. The 6-point Gauss quadrature rule on triangle is used in evaluating regular element integrals, and thus the quadrature points of the Nyström method on each element are those of the 6-point Gauss rule. As a result, the total number of DOFs is $N = 6 \cdot N_e$.

For each quadrature point \mathbf{x}_i , $i = 1, \dots, N$, a local region, denoted by D_i , should be assigned. In this paper D_i consists of all the elements whose distance to \mathbf{x}_i is not larger than 2 times of their largest side length. To show the discretizing procedures, let \mathcal{K} be one of the integral operators in (4) and K be the associated kernel function. For any field point \mathbf{x}_i the boundary integral $\int_{\Gamma} K(\mathbf{x}_i, \mathbf{y})u(\mathbf{y}) \, d\mathbf{y}$ can be divided into two parts by the local region D_i ; i.e.,

$$\int_{\Gamma} K(\mathbf{x}_i, \mathbf{y})u(\mathbf{y}) \, d\mathbf{y} \approx \sum_{\Delta \notin D_i} \int_{\Delta} K(\mathbf{x}_i, \mathbf{y})u(\mathbf{y}) \, d\mathbf{y} + \sum_{\Delta \in D_i} \int_{\Delta} K(\mathbf{x}_i, \mathbf{y})u(\mathbf{y}) \, d\mathbf{y} \quad (8)$$

where Δ denotes the boundary elements. For elements outside the local region D_i the integral is regular and thus is accurately evaluated by using the Gauss quadrature,

$$\int_{\Delta} K(\mathbf{x}_i, \mathbf{y})u(\mathbf{y}) \, d\mathbf{y} \approx \sum_j \omega_j K(\mathbf{x}_i, \mathbf{y}_j)u(\mathbf{y}_j), \quad \Delta \notin D_i, \quad (9)$$

where \mathbf{y}_j and ω_j are the j -th quadrature point and weight over element Δ , respectively.

For elements inside the local region D_i , however, the kernels exhibit various types of singularity. As a result, conventional quadratures fail to give correct results. In order to maintain high-order properties, the quadrature weights are adjusted by a local correction procedure [32]. Consequently one has

$$\int_{\Delta} K(\mathbf{x}_i, \mathbf{y})u(\mathbf{y}) \, d\mathbf{y} \approx \sum_j \bar{\omega}_j^{\Delta}(K; \mathbf{x}_i)u(\mathbf{y}_j), \quad \Delta \in D_i, \quad (10)$$

where $\bar{\omega}_j^{\Delta}(K; \mathbf{x}_i)$ represents the locally corrected quadrature weights associated with element Δ , kernel K and field point \mathbf{x}_i . The local corrected procedure is performed by approximating the unknown quantities using linear combination of polynomial basis functions which are defined on intrinsic coordinates of the element. The locally corrected quadrature weights are obtained

by solving the linear system

$$\sum_j \bar{\omega}_j^{\Delta}(K; \mathbf{x}_i)\phi^{(m)}(\mathbf{y}_j) = \int_{\Delta} K(\mathbf{x}_i, \mathbf{y})\phi^{(m)}(\mathbf{y}) \, d\mathbf{y}, \quad \Delta \in D_i, \quad (11)$$

where $\phi^{(m)}$ are polynomial basis functions. For the Nyström method based on quadratic elements as used in this paper, $\phi^{(m)}$ are given by

$$\phi^{(m)}(\xi_1, \xi_2) = \xi_1^p \xi_2^q, \quad p + q \leq 2, \quad (12)$$

where p and q are integers, ξ_1 and ξ_2 denote local intrinsic coordinates.

The integrals in Eq. (11) become nearly singular when \mathbf{x}_i is close to the element, and these integrals are computed by using a recursive subdivision quadrature in this paper. When the field point \mathbf{x}_i locates on the element, singularity appears in the integrals. For the first three operators in Eq. (4), the integrals have weak singularity of order r^{-1} , while for the hyper-singular operator \mathcal{H} the integral has singularity of order r^{-3} , as $r \rightarrow 0$. The accurate evaluation of those singular integrals is crucial to ensuring the accuracy of the BEM. Here, an efficient numerical method recently proposed in [33] is used. The method is capable of treating weakly, strongly and hyper-singular integrals in a similar manner with high accuracy, and the code is open.

Consequently, the boundary integral (8) can be transformed into a summation as

$$\int_{\Gamma} K(\mathbf{x}_i, \mathbf{y})u(\mathbf{y}) \, d\mathbf{y} \approx \sum_{j=1}^N \bar{K}(\mathbf{x}_i, \mathbf{y}_j)u_j, \quad (13)$$

where $u_j = u(\mathbf{y}_j)$ and

$$\bar{K}(\mathbf{x}_i, \mathbf{y}_j) = \begin{cases} K(\mathbf{x}_i, \mathbf{y}_j)\omega_j, & \mathbf{y}_j \text{ is on a element } \Delta \notin D_i, \\ \bar{\omega}_j^{\Delta}(K; \mathbf{x}_i), & \mathbf{y}_j \text{ is on a element } \Delta \in D_i. \end{cases} \quad (14)$$

Performing the Nyström discretization to the boundary integrals in the Burton–Miller equation (7) leads to the BEM linear system of form

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q} + \mathbf{f}, \quad (15)$$

where \mathbf{H} and \mathbf{G} are $N \times N$ matrices, \mathbf{u} and \mathbf{q} are N -vectors of the boundary values of $u(\mathbf{x})$ and $q(\mathbf{x})$, respectively, and \mathbf{f} consists of the values of the incident wave terms in (7). It is worth noting that when \mathbf{y}_j is on an element outside the local region D_i of \mathbf{x}_i the kernel values of the matrices \mathbf{H} and \mathbf{G} are just the quadrature weight ω_j of \mathbf{y}_j time the function values

$$H_{ij} = \frac{\partial G}{\partial \mathbf{n}_y}(\mathbf{x}_i, \mathbf{y}_j) + \alpha \frac{\partial^2 G}{\partial \mathbf{n}_x \partial \mathbf{n}_y}(\mathbf{x}_i, \mathbf{y}_j) \quad \text{and} \quad (16a)$$

$$G_{ij} = G(\mathbf{x}_i, \mathbf{y}_j) + \alpha \frac{\partial G}{\partial \mathbf{n}_x}(\mathbf{x}_i, \mathbf{y}_j). \quad (16b)$$

Therefore, the matrix–vector product with matrices \mathbf{H} and \mathbf{G} is in fact summations with the respective kernels.

By considering the boundary conditions, Eq. (15) is recast as the following linear system of equations to be solved:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (17)$$

where the $N \times N$ matrix \mathbf{A} consists of columns of matrices \mathbf{H} and \mathbf{G} , depending on the specific boundary conditions, \mathbf{b} and \mathbf{x} are known and unknown N -dimensional vectors, respectively. For large-scale problems, linear system (17) is often solved by using the iterative solvers; the generalized minimal residual method (GMRES) will be used in this paper. The main computational work of the iterative solver is in the evaluation of matrix–vector product $\mathbf{A}\mathbf{x}$. Since matrix \mathbf{A} is always fully populated in BEM, the computational cost for a naive evaluation should be $O(N^2)$, which suddenly becomes prohibitive with the increase of N . In the next section, the fast

directional algorithm is employed to reduce the square scaled computational cost to almost linear.

3. Fast directional algorithm for Burton–Miller formulation

In this section, the FDA is adapted to accelerate the Nyström BEM for Burton–Miller BIE, resulting in a fast BEM solver (denoted by FDBEM) for acoustic problems. The central is to develop a FDA for the fast summation with the kernels in (16).

3.1. Multilevel FDA

The multilevel FDA recently developed for the Helmholtz kernel is briefly reviewed here, detailed description can be found in [29,34].

3.1.1. Octree structure

The implementation of the FDA relies on an adaptive octree structure by which the source and target points are grouped [21]. The octree is constructed by the following steps. First, find a level-0 cube containing all the points. Then, subdivide each cube C at level- l of the octree into eight equally sized level- $(l+1)$ cubes if it contains more than N_p points. The subdivision process is performed recursively until each leaf cube contains no more than N_p points. In this paper, N_p is determined by

$$N_p = \max(30, 50(\log_{10} \varepsilon - 3)), \quad (18)$$

where ε is the target accuracy of the FDA. The finest level of the octree is indexed by L .

In order to developing a FDA that is efficient in both low- and high-frequencies, the octree is often divided into two regimes, namely the low frequency regime and high frequency regime, according the side length of the cubes and the wavelength. The low frequency regime consists of cubes whose side length w is less than the wavelength, and the high frequency regime consists of the rest cubes. Obviously, when the non-dimensional wave number kD (where D is the typical size of the boundary Γ) is sufficiently small, all the levels would lie in the low frequency regime, and the FDA would degenerate to the kernel independent FMM [21].

With the aid of the octree, the basic notations of the FDA can be defined. Consider a cube C with width w in the octree.

- *Near field* N^C : When C is in the low frequency regime, its near field N^C is defined as the union of the cubes that are adjacent with C ; when it is in the high frequency regime, N^C is defined as the union of the cubes B that are separated from C by no more than n_s cubes in the same level, with n_s being the rounded number of $(c_d kw)$. In other words, for each cube $B \in N^C$,

$$\text{dist}(B, C) \leq c_d (kw)^2,$$

where $\text{dist}(B, C)$ denotes the distance between B and C ; Let w_B and w_C be the widths and \mathbf{c}_B and \mathbf{c}_C be the centroids of cubes B and C , respectively, then

$$\text{dist}(B, C) = \max_{i=1,2,3} (|\mathbf{c}_{B,i} - \mathbf{c}_{C,i}| - w_B/2 - w_C/2).$$

The constant $c_d = 1/\pi$ is used in this paper.

- *Far field* F^C : Consists of all the cubes not in N^C .
- *Interaction field* I^C : $I^C = N^p \setminus N^C$.
- *Directional wedges*: When C is in the high frequency regime, I^C can be further divided into multiple directional wedges, each of which has the spanning angle no greater than $O(1/(kw))$. The directional wedges would be indexed by their center directions \mathbf{u} , as shown in Fig. 1. The area of the boundary of N^C is $6[(2n_s + 1)w]^2$, and it should be cutted by the directional

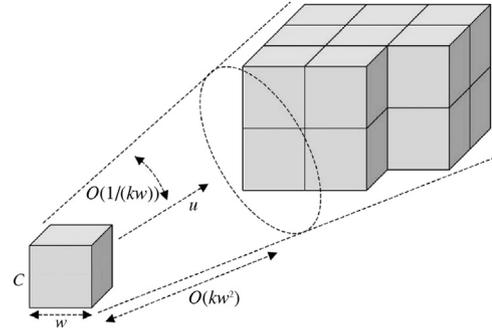


Fig. 1. The interaction field of a cube C in the high frequency regime is partitioned into multiple directional wedges.

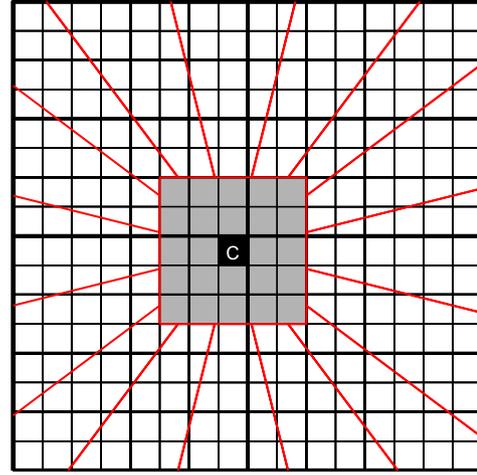


Fig. 2. Near field (the gray domain) and directional wedges for a cube C in high frequency regime, with $n_s=2$. When B lies in the gray domain, $B \in N^C$; when B is one of the white cubes, $B \in I^C$, and it lies in one of directional wedges.

wedges into pieces with size $O(1/(kw)) \cdot O(kw^2) = O(w)$. Thus, the number of directional wedges should be $6[(2n_s + 1)w]^2 / O(w^2) = O(n_s^2)$. It is noticed that the Cartesian grid with n_s points along each direction contains $O(n_s^2)$ points on the surface, which inspired us to define the directional wedges with these surface grid points. In other words, the directional wedges are defined by center directions $\mathbf{u} = (i, j, k)$, with $i, j, k = -n_s, -n_s + 1, \dots, n_s$ and $\max(i, j, k) = n_s$. For each cube $B \in I^C$, the directional wedge in which it lies is defined to be the one with center direction \mathbf{u} closest to $\mathbf{c}_B - \mathbf{c}_C$. The near field and directional wedges for a cube in high frequency regime are illustrated in Fig. 2. The low frequency regime can be considered as a special case of the high frequency regime which consists of only one directional wedge indexed by $\boldsymbol{\zeta} = (0, 0, 0)$.

It is proved that the numerical rank of interaction matrices is low for the points in C and each directional wedge of I^C [29]. Therefore, the far field summation can be accelerated by using the low rank approximation.

3.1.2. Translations in FDA

The FDA recently developed in [29] is a fast algorithm for computing summations with the Helmholtz kernel, i.e.,

$$p_i = \sum_j G(\mathbf{x}_i, \mathbf{y}_j) q_j. \quad (19)$$

where G is given by (3). The basic procedure consists of five translations, namely, the source-to-multipole (S2M), multipole-to-multipole (M2M), multipole-to-local (M2L), local-to-local (L2L) and local-to-target (L2T) translations, which is similar to the well-known FMM. As a kernel independent algorithm, FDA constructs the *outgoing equivalent densities* to take the role of *multipole expansions* in analytic FMM, and the *incoming check potentials* to take the role of *local expansions*. The main procedure of FDA is illustrated in Fig. 3. To evaluate the contribution of the sources (marked with “+”) to the potentials at the target points (marked with “×”) in a far-away cube, the five translations are carried out serially. First translate the original sources into *outgoing equivalent densities* (the blue points marked with “•”) by S2M; then compute the *outgoing equivalent densities* on coarser levels by M2M; evaluate the *incoming check potentials* (the red points marked with “○”) produced by the *outgoing equivalent densities* by M2L; translate the *incoming check potentials* on the coarse levels to finer levels by L2L; finally the contribution to the potentials on the target points is computed using the *incoming check potentials* by L2T. For the cubes in low frequency regime (smaller cubes in Fig. 3), the *outgoing equivalent densities* and *incoming check potentials* are non-directional, as denoted by the equally spaced points on squares in Fig. 3. For cubes in high frequency regime (larger cubes in Fig. 3), they are directional, and are denoted by the non-equally spaced points on circles, concentrating near a direction. The translations are described in more detail as follows.

For a leaf cube C , the S2M operation, represented by matrix S , translates the original sources inside C to the outgoing equivalent densities of C . The definition of the outgoing equivalent points and the associated outgoing check points is illustrated in Fig. 4, where

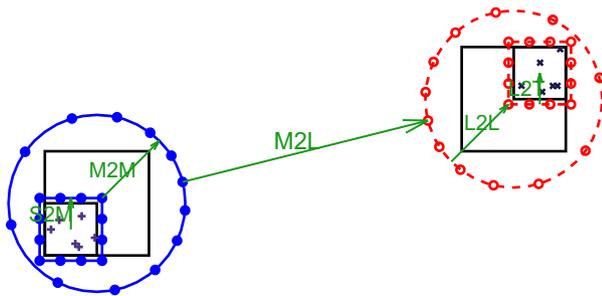


Fig. 3. The main procedure of FDA. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

all the definitions are demonstrated using the 2D cases, but they can be generalized to 3D readily. The S2M translation can be further divided into two steps: (1) the evaluation of the outgoing check potentials at the outgoing check points produced by the original sources, the matrix of which is denoted by E_{up} , and (2) the inversion to construct the outgoing equivalent densities at the outgoing equivalent points that can reproduce the outgoing check potentials, the matrix of this linear translation is denoted by R_{up}^+ .

When C is in the low frequency regime, the outgoing equivalent points and the outgoing check points are non-directional, and can be sampled straightforwardly at Cartesian grid points, as described in [21]. The matrix R_{up}^+ is defined as the pseudo-inverse of $[R_{up}]_{ij} = G(\mathbf{a}_i, \mathbf{b}_j)$, where \mathbf{a}_i is the i -th outgoing check point, and \mathbf{b}_j is the j -th outgoing equivalent point. When C is in the high frequency regime, since its interaction field is partitioned into multiple directional wedges, a group of the outgoing equivalent and check points have to be defined for each directional wedge. In this paper, these points and R_{up}^+ in direction $(1, 0, 0)$ are computed by the algorithm in [34]. For the other directional wedges, the points can be obtained by rotation and the matrices R_{up}^+ remain the same. In both low and high frequency regime, the numbers of equivalent points and check points are controlled by the required accuracy ε of the FDA; see [29,34].

For each non-leaf cube at level- l , the M2M operation, associated with matrix M_l , translates the outgoing equivalent densities of its children to the outgoing equivalent densities of itself. When C is in the low frequency regime, only one group of outgoing equivalent densities of C are to be translated, otherwise one has to loop over all the directional wedges to compute the directional outgoing equivalent densities for each wedge.

The translations in the downward pass of FDA are performed based on another two groups of points, namely the incoming check points and incoming equivalent points whose definitions are similar to those of the outgoing points with the roles reversed. For example, in the downward pass the outgoing equivalent points are used as the incoming check points, and the outgoing check points are used as the incoming equivalent points, as illustrated in Fig. 5.

The M2L operation, represented by matrix K , translates the outgoing equivalent points to incoming check potentials. Assume B is a cube in C 's interaction list and in direction ζ , then C must lie in B 's interaction list and in direction $-\zeta$. The M2L translation evaluates B 's incoming check potentials in direction $-\zeta$ using C 's outgoing equivalent densities in direction ζ .

For each non-leaf cube C at the l -th level, the L2L matrix L_l translates the incoming check potentials of itself to the incoming

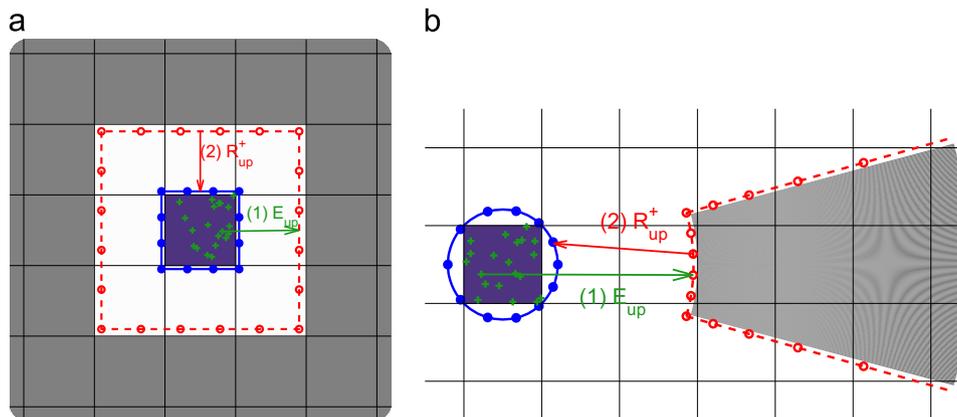


Fig. 4. The outgoing equivalent points (the blue points marked with “•”), outgoing check points (the red points marked with “○”) and S2M translations in the low and high frequency regime. The computation of the outgoing equivalent densities for the original sources inside a cube (located at the points marked with “+”) includes two steps shown by arrows: (1) the evaluation of the check potentials at the outgoing check points using the original sources and (2) the inversion to construct the outgoing equivalent densities. (a) Low frequency regime and (b) high frequency regime. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

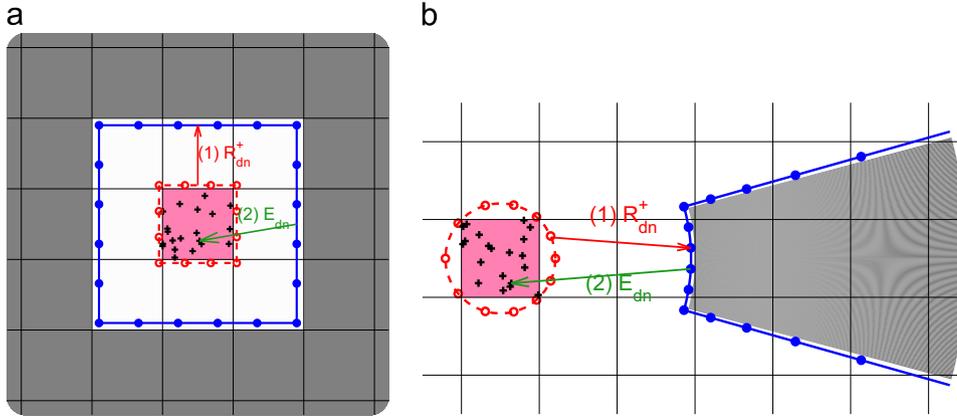


Fig. 5. The incoming equivalent points (the blue points marked with “•”), incoming check points (the red points marked with “◦”) and L2T translations in the low and high frequency regime. The potentials at the points inside the cube (marked with “+”) are computed by two steps shown by arrows: (1) the inversion to construct the incoming equivalent densities that can reproduce the incoming check potentials at the check points, and (2) the evaluation of the potentials at the target points inside the cube using the equivalent densities. (a) Low frequency regime and (b) high frequency regime. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

check potentials of its children. Similar to the translations in the upward pass, it also consists of two steps: (1) the inversion to construct C 's incoming equivalent densities at the incoming equivalent points that can reproduce its incoming check potentials, the matrix of which is denoted by \mathbf{R}_{dn}^+ , and (2) the evaluation of the incoming check potentials of C 's child cubes produced by C 's incoming equivalent densities, the matrix of which is denoted by \mathbf{E}_{dn} .

The L2T matrices \mathbf{T} translate the incoming check potentials of each leaf cube C to the potentials at the target points inside C . Similar to L2L, it also consists of the inversion and evaluation steps, as illustrated in Fig. 5.

3.2. FDA for Burton–Miller formulation

The FDA outlined above is a multilevel algorithm for the evaluation of potentials defined by (19) where G is the Helmholtz kernel. Although the algorithm is kernel-independent, its application to the evaluation of the potentials associated with the kernels (16) of the Burton–Miller BIE is not straightforward, because the kernels involve the normal vector \mathbf{n}_x at the field point \mathbf{x} which cannot be taken into consideration in the construction of equivalent densities. Below, strategies to cope with this issue and their numerical implementations are described.

Consider the potential summation with kernel G_{ij} in (16b),

$$\begin{aligned} p_i &= \sum_{j=1}^N \left[G(\mathbf{x}_i, \mathbf{y}_j) + \alpha \frac{\partial G}{\partial \mathbf{n}_x}(\mathbf{x}_i, \mathbf{y}_j) \right] q_j \\ &= \sum_{j=1}^N \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x} \right) G(\mathbf{x}_i, \mathbf{y}_j) q_j \\ &= \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x} \right) \sum_{j=1}^N G(\mathbf{x}_i, \mathbf{y}_j) q_j. \end{aligned} \quad (20)$$

It follows that the potential p_i can be computed by two steps. The first step is the evaluation of the single layer potentials, denoted by $\hat{p}(\mathbf{x})$, generated by the sources q_j , i.e.,

$$\hat{p}(\mathbf{x}) = \sum_{j=1}^N G(\mathbf{x}, \mathbf{y}_j) q_j. \quad (21)$$

This can be accelerated by using the FDA in Section 3.1. The second step is the computation of p_i by

$$p_i = \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x} \right) \hat{p}(\mathbf{x}_i). \quad (22)$$

The numerical implementation of this step requires a change of the matrix \mathbf{E}_{dn} that is used in the computation of the L2T translation matrix $\mathbf{L} = \mathbf{E}_{dn} \mathbf{R}^+$. More specifically, for a cube C , suppose that a group of incoming equivalent densities \bar{q}_j at the incoming equivalent points \mathbf{b}_j are obtained in the first step of the L2T translation, which can produce $\hat{p}(\mathbf{x})$ inside C , i.e.,

$$\hat{p}(\mathbf{x}) = \sum_j G(\mathbf{x}, \mathbf{b}_j) \bar{q}_j.$$

Then the potential p_i in (22) can be evaluated by

$$\begin{aligned} p_i &= \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x} \right) \sum_j G(\mathbf{x}_i, \mathbf{b}_j) \bar{q}_j \\ &= \sum_j \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x} \right) G(\mathbf{x}_i, \mathbf{b}_j) \bar{q}_j \\ &= \sum_j \left[G(\mathbf{x}_i, \mathbf{b}_j) + \alpha \frac{\partial G}{\partial \mathbf{n}_x}(\mathbf{x}_i, \mathbf{b}_j) \right] \bar{q}_j \end{aligned} \quad (23)$$

This suggests that p_i at \mathbf{x}_i can be computed by using

$$[\mathbf{E}_{dn}]_{ij} = G(\mathbf{x}_i, \mathbf{b}_j) + \alpha \frac{\partial G}{\partial \mathbf{n}_x}(\mathbf{x}_i, \mathbf{b}_j) \quad (24)$$

as the translation matrix for the evaluation step in L2T.

The FDA for the potential summation with kernel H_{ij} in (16a) can be attained analogously, since

$$\begin{aligned} p_i &= \sum_{j=1}^N \left[\frac{\partial G}{\partial \mathbf{n}_y}(\mathbf{x}_i, \mathbf{y}_j) + \alpha \frac{\partial^2 G}{\partial \mathbf{n}_x \partial \mathbf{n}_y}(\mathbf{x}_i, \mathbf{y}_j) \right] q_j \\ &= \sum_{j=1}^N \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x} \right) \frac{\partial G}{\partial \mathbf{n}_y}(\mathbf{x}_i, \mathbf{y}_j) q_j \\ &= \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x} \right) \sum_{j=1}^N \frac{\partial G}{\partial \mathbf{n}_y}(\mathbf{x}_i, \mathbf{y}_j) q_j. \end{aligned} \quad (25)$$

The potential p_i can be computed with two steps. The first step is the evaluation of the double layer potentials, denoted by $\tilde{p}(\mathbf{x})$, generated by the sources q_j ,

$$\tilde{p}(\mathbf{x}) = \sum_{j=1}^N \frac{\partial G}{\partial \mathbf{n}_y}(\mathbf{x}, \mathbf{y}_j) q_j. \quad (26)$$

Summation (26) can be accelerated by using the FDA in Section 3.1 with minor modification. The rationale is that the source densities q_j are essentially dipoles in the \mathbf{n}_y direction. Since the field produced by dipoles can be approximated by a group of monopoles in its vicinity, the potentials produced by the sources inside a cube C can be approximated by a group of equivalent densities, when observing at the far field of C .

Therefore, in the FDA for single layer kernel G in Section 3.1, the S2M matrix \mathbf{S} is computed as $\mathbf{S} = \mathbf{R}_{\text{up}}^+ \mathbf{E}_{\text{up}}$. The entries of matrix \mathbf{E}_{up} are given by the values of the single layer kernel,

$$[\mathbf{E}_{\text{up}}]_{ij} = G(\mathbf{a}_i, \mathbf{y}_j),$$

because the source densities are monopoles. When FDA is adopted to the double layer potential in (26), the entries of \mathbf{E}_{up} have to be the values of the double layer kernel,

$$[\mathbf{E}_{\text{up}}]_{ij} = \frac{\partial G}{\partial \mathbf{n}_y}(\mathbf{a}_i, \mathbf{y}_j). \quad (27)$$

The second step is the computation of p_i by

$$p_i = \left(1 + \alpha \frac{\partial}{\partial \mathbf{n}_x}\right) \tilde{p}(\mathbf{x}_i), \quad (28)$$

which can be realized in the same way as (22).

From the above discussion, it is clear that there are two common points in the FDA for the Helmholtz kernel (19) and those for the kernels of the Burton–Miller BIE. First, equivalent sources of monopoles are always used even though the Burton–Miller kernels (16) are the combinations of the Helmholtz kernel and its derivatives. Second, since the M2M, M2L and L2L translations only involve the equivalent densities, no modification is needed at all for these translations.

Besides the common points, three modifications are described for the application of FDA to the Burton–Miller formulation:

- (1) For the summation with H_{ij} , the outgoing check potentials for S2M should be computed by using the kernel $(\partial G / \partial \mathbf{n}_y)(\mathbf{x}, \mathbf{y})$.
- (2) The matrix \mathbf{E}_{dn} in the computation of the L2T matrix should be evaluated by (24), for both summations with kernels H_{ij} and G_{ij} .
- (3) The contribution to the potentials by the sources in the near field should be computed by using the kernels (16a) and (16b), as well as their local corrections.

In the implementation of the FDA, the matrices of the M2M, M2L and L2L translations are computed only once and then stored in memory for later use. In addition, from the definition of the equivalent points and check points, it is known that the inverse operator in the downward pass is the transpose of that in the upward pass, and the L2L translation matrix is just the transpose of the M2M translation matrix for the same directional wedge; that is, $\mathbf{R}_{\text{dn}}^+ = (\mathbf{R}_{\text{up}}^+)^T$, $\mathbf{E}_{\text{dn}} = \mathbf{E}_{\text{up}}^T$, and

$$\mathbf{L} = \mathbf{E}_{\text{dn}} \mathbf{R}_{\text{dn}}^+ = \mathbf{E}_{\text{up}}^T (\mathbf{R}_{\text{up}}^+)^T = (\mathbf{R}_{\text{up}}^+ \mathbf{E}_{\text{up}})^T = \mathbf{M}^T. \quad (29)$$

Therefore only one set of matrix is needed to be computed and saved for both the L2L and M2M translations in each directional wedge.

4. Further improvement to FDA

In this section, further improvement to the FDA is proposed based on the observation that generally the translation matrices are rank deficient. Therefore, the computational cost of the FDA can be reduced by compressing these translation matrices into more compact forms. Furthermore, our numerical experiments show that, after the matrix reduction, most of the M2L matrices are still of ranks much lower than their dimensions, as shown in Fig. 6. Therefore these M2L translations can be further accelerated by using the low rank approximations for the translation matrices. In the present work, due to the small sizes of the matrices, the low-rank decomposition is computed by using singular value decomposition (SVD) to get the optimal ranks.

		13	14	15	15	15	15	15	15	15	14	13		
	13	14	15	15	15	15	15	15	15	15	15	14	13	
13	14	15	15	15	16	17	17	17	15	15	15	15	14	13
14	15	15	15	17	21	21	21	21	21	17	15	15	15	14
15	15	15	17	21	24	27	28	27	25	21	17	15	15	15
15	15	15	21	25						24	21	16	15	15
15	15	17	21	27						27	21	17	15	15
15	15	17	21	28			C			28	21	17	15	15
15	15	17	21	27						27	21	17	15	15
15	15	16	21	24						25	21	15	15	15
15	15	15	17	21	25	27	28	27	24	21	17	15	15	15
14	15	15	15	17	21	21	21	21	21	17	15	15	15	14
13	14	15	15	15	15	17	17	17	16	15	15	15	14	13
	13	14	15	15	15	15	15	15	15	15	14	13		
		13	14	15	15	15	15	15	15	15	14	13		

Fig. 6. The ranks of M2L matrices in the finest level of the high frequency regime, i.e., $w = \lambda$, with $\epsilon = 1e-4$. The dimension of the compressed M2L matrices is $s=38$. The purple number denotes the M2L matrices whose rank is less than $s/2$, that is, these translations can be accelerated by using the low rank approximations. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

4.1. Matrix reduction

Consider two leaf cubes B and C at the l -th level such that B is in the interaction field of C 's parent cube. Then, the potentials \mathbf{p}_l in C produced by the source densities \mathbf{q}_l in B are evaluated as

$$\mathbf{p}_l = \mathbf{T}_l \mathbf{L}_{l-1} \mathbf{K}_{l-1} \mathbf{M}_{l-1} \mathbf{S}_l \mathbf{q}_l, \quad (30)$$

where \mathbf{S}_l and \mathbf{T}_l are the S2M and L2T translation matrices, respectively, at the l -th level; \mathbf{M}_{l-1} , \mathbf{K}_{l-1} and \mathbf{L}_{l-1} are the M2M, M2L and L2L translation matrices at the $(l-1)$ -th level, respectively.

Below, a scheme for reducing the sizes of aforementioned matrices is proposed. Consider the computation of the S2M and L2T matrices,

$$\mathbf{S}_l = \mathbf{R}_{\text{up},l}^+ \mathbf{E}_{\text{up},l}, \quad (31a)$$

$$\mathbf{T}_l = \mathbf{E}_{\text{dn},l} \mathbf{R}_{\text{dn},l}^+, \quad (31b)$$

where \mathbf{R}^+ is the pseudo-inverse of \mathbf{R} , which is computed by the SVD,

$$\mathbf{R} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^H, \quad (32)$$

where \mathbf{U} and \mathbf{V} are unity matrices, \mathbf{V}^H is the Hermitian matrix of \mathbf{V} , and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, with $\sigma_1 > \sigma_2 > \dots > \sigma_n$, is the diagonal matrix of singular values. For a given error tolerance ϵ , identify the singular values $\sigma_i < \epsilon \sigma_1$ and truncate the associated columns of matrices \mathbf{U} and \mathbf{V} to obtain two matrices $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ which consist of the columns of \mathbf{U} and \mathbf{V} corresponding to the singular values $\sigma_i \geq \epsilon \sigma_1$. Then \mathbf{R}^+ is given by

$$\mathbf{R}^+ \approx \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}^H, \quad (33)$$

where $\tilde{\mathbf{\Sigma}}$ is the diagonal matrix of the remaining singular values. In this paper, the error tolerance ϵ is chosen to be the controlling accuracy in the sampling process of the directional equivalent points and check points.

The matrices \mathbf{M}_{l-1} and \mathbf{L}_{l-1} have the similar decompositions with the matrices \mathbf{S}_l and \mathbf{T}_l in (31). By plugging those

decompositions into (30) and using (33), one gets

$$\begin{aligned} \tilde{\mathbf{p}}_l &= \mathbf{T}_l \mathbf{L}_{l-1} \mathbf{K}_{l-1} \mathbf{M}_{l-1} \mathbf{S}_l \mathbf{q}_l \\ &= (\mathbf{E}_{\text{dn},l} \mathbf{R}_{\text{dn},l}^+) (\mathbf{E}_{\text{dn},l-1} \mathbf{R}_{\text{dn},l-1}^+) \mathbf{K}_{l-1} (\mathbf{R}_{\text{up},l-1}^+ \mathbf{E}_{\text{up},l-1}) (\mathbf{R}_{\text{up},l}^+ \mathbf{E}_{\text{up},l}) \mathbf{q}_l \\ &\approx \mathbf{E}_{\text{dn},l} (\tilde{\mathbf{V}}_{\text{dn},l} \tilde{\Sigma}_{\text{dn},l}^{-1} \tilde{\mathbf{U}}_{\text{dn},l}^H) \mathbf{E}_{\text{dn},l-1} (\tilde{\mathbf{V}}_{\text{dn},l-1} \tilde{\Sigma}_{\text{dn},l-1}^{-1} \tilde{\mathbf{U}}_{\text{dn},l-1}^H) \\ \mathbf{K}_{l-1} (\tilde{\mathbf{V}}_{\text{up},l-1} \tilde{\Sigma}_{\text{up},l-1}^{-1} \tilde{\mathbf{U}}_{\text{up},l-1}^H) \mathbf{E}_{\text{up},l-1} (\tilde{\mathbf{V}}_{\text{up},l} \tilde{\Sigma}_{\text{up},l}^{-1} \tilde{\mathbf{U}}_{\text{up},l}^H) \mathbf{E}_{\text{up},l} \mathbf{q}_l \\ &= \tilde{\mathbf{T}}_l \tilde{\mathbf{L}}_{l-1} \tilde{\mathbf{K}} \tilde{\mathbf{M}}_{l-1} \tilde{\mathbf{S}}_l \mathbf{q}_l, \end{aligned} \quad (34)$$

where all the new translation matrices have smaller sizes,

$$\tilde{\mathbf{S}}_l = \tilde{\Sigma}_{\text{up},l}^{-1} \tilde{\mathbf{U}}_{\text{up},l}^H \mathbf{E}_{\text{up},l}, \quad (35)$$

$$\tilde{\mathbf{M}}_l = \tilde{\Sigma}_{\text{up},l}^{-1} \tilde{\mathbf{U}}_{\text{up},l}^H \mathbf{E}_{\text{up},l}, \tilde{\mathbf{V}}_{\text{up},l+1} \quad (36)$$

$$\tilde{\mathbf{K}}_l = \tilde{\mathbf{U}}_{\text{dn},l}^H \mathbf{K}_l \tilde{\mathbf{V}}_{\text{up},l}, \quad (37)$$

$$\tilde{\mathbf{L}}_l = \tilde{\mathbf{U}}_{\text{dn},l+1}^H \mathbf{E}_{\text{dn},l} \tilde{\mathbf{V}}_{\text{dn},l} \tilde{\Sigma}_{\text{dn},l}^{-1} = \tilde{\mathbf{M}}_l^T, \quad (38)$$

$$\tilde{\mathbf{T}}_l = \mathbf{E}_{\text{dn},l} \tilde{\mathbf{V}}_{\text{dn},l} \tilde{\Sigma}_{\text{dn},l}^{-1}. \quad (39)$$

Thus the computational cost of all the translations can be reduced by using these new matrices (34). It is shown that the reduction of computational time can be up to 30% for each FDA summation. Below, a method for further and much more considerable reduction of the computational time of the FDA is proposed.

4.2. Low rank approximations for the compressed M2L matrices

Here the M2L translation is considered in detail because it consumes the most computational time of the FDA. Let $\tilde{\mathbf{K}}$ be a compressed M2L matrix in (37) with dimension s . It is found that, to the accuracy of ϵ , there are always many M2L matrices $\tilde{\mathbf{K}}$ whose numerical ranks are much less than $s/2$. This indicates that the M2L translation would be further accelerated by performing matrix–vector production using the low rank decomposition of $\tilde{\mathbf{K}}$ instead of $\tilde{\mathbf{K}}$ itself.

The low rank decomposition for each compressed M2L matrix $\tilde{\mathbf{K}}$ is computed by the truncated SVD with the singular values $\sigma_i < \epsilon \sigma_1$ being dropped; that is,

$$\tilde{\mathbf{K}} \approx \hat{\mathbf{U}} \hat{\Sigma} \hat{\mathbf{Q}}^H = \hat{\mathbf{U}} \hat{\mathbf{V}}, \quad (40)$$

where $\hat{\mathbf{U}}$ and $\hat{\mathbf{Q}}$ are matrices with orthogonal columns, $\hat{\Sigma}$ are the truncated singular value matrix and $\hat{\mathbf{V}} = \hat{\Sigma} \hat{\mathbf{Q}}^H$. Moreover, since there are at most $O(\sqrt{N})$ M2L matrices in FDA [29], the

computational time of the decomposition of the M2L matrices is of order $O(s^3 \sqrt{N})$ which is negligible in the total CPU time of the FDA.

For each M2L matrix $\tilde{\mathbf{K}}$ with rank $r < s/2$, the low-rank decomposition (40) is used in the translation. Otherwise, the matrix $\tilde{\mathbf{K}}$ itself is used.

The objective of the compression and decomposition to the translation matrices in this section is similar with the SARcmp M2L optimizer in [26]. However, two distinctions of our method should be noticed. First, the compressed matrices in Section 4.1 are computed during the definition of the equivalent and check points, thus one do not need to compute the low rank decomposition of the collected M2L matrices as in SARcmp. Second, all the translation matrices are compressed in our work, while only the M2L matrices are compressed in [26].

5. Numerical studies

In this section, representative numerical examples are provided to demonstrate the performance of the improved FDA and its application for the acceleration of the BEM for Burton–Miller formulation, denoted by FDBEM. Our codes are implemented serially in C++. The computing platform is a workstation with a Xeon 5450 (2.66 GHz) CPU and 32 GB RAM.

5.1. Performance of the improvements in Section 4

First, the performance of our acceleration technique in Section 4 is tested by evaluating the potential summation

$$p_i = \sum_{j=1}^N \left[G(\mathbf{x}_i, \mathbf{y}_j) + \alpha \frac{\partial G}{\partial \mathbf{n}_y}(\mathbf{x}_i, \mathbf{y}_j) \right] q_j, \quad (41)$$

with G being the Helmholtz kernel in (3). We set $\alpha = i/k$ as this will be used in the Burton–Miller formulation. The points $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_j\}$ are sampled on the surface of a unit sphere with about 20 points per wavelength. The densities $\{q_j\}$ on $\{\mathbf{y}_j\}$ are randomly defined with mean 0. Eq. (41) is evaluated by the original FDA and the improved FDA in Section 4. The error ϵ_a of the potential p_i ,

$$\epsilon_a = \sqrt{\frac{\sum_{i=1}^{N_t} |p_i^{(a)} - p_i^{(d)}|^2}{\sum_{i=1}^{N_t} |p_i^{(d)}|^2}}, \quad (42)$$

is computed and compared, where $p_i^{(a)}$ is the potential on a N_t randomly selected points \mathbf{x}_i computed by the fast algorithm, while

Table 1
Results of the example in Section 5.1.

N	k	T_{M2L} (s)		T_{up} (s)		ϵ_a		M (MB)	
		Original	Improved	Original	Improved	Original	Improved	Original	Improved
$\epsilon = 1e-4$									
73,728	8π	10	4	5	3	$4.86e-5$	$5.88e-5$	310	158
294,912	16π	44	17	17	10	$7.30e-5$	$1.31e-4$	735	441
1,179,648	32π	218	94	103	58	$1.15e-4$	$2.33e-4$	1993	1338
4,718,592	64π	966	383	458	214	$2.00e-4$	$3.70e-4$	6818	5053
$\epsilon = 1e-6$									
73,728	8π	33	11	12	7	$2.97e-7$	$9.33e-7$	1010	471
294,912	16π	154	55	59	36	$7.75e-7$	$2.72e-6$	2186	1189
1,179,648	32π	631	235	218	145	$1.30e-6$	$3.83e-6$	4744	2941
4,718,592	64π	2937	1145	1153	742	$3.26e-6$	$1.06e-5$	13,938	9764
$\epsilon = 1e-8$									
73,728	8π	93	29	24	16	$2.39e-8$	$2.79e-8$	2644	1156
294,912	16π	421	137	119	84	$4.93e-8$	$1.11e-7$	5154	2684
1,179,648	32π	1806	602	563	384	$7.12e-8$	$7.27e-8$	10,147	5944
4,718,592	64π	7721	2527	2478	1586	$1.71e-7$	$2.55e-7$	26,860	18,121

$p_i^{(d)}$ is the potentials evaluated by straightforwardly summation. In this paper, N_t is chosen as 200. For particle summation problems, the S2M and L2T matrices are used only once, thus they are not pre-computed but only computed when they are to be used.

The CPU times and errors are given in Table 1. T_{M2L} and T_{up} are the CPU time of the M2L translation and upward pass, respectively; M is the total memory usage. The summations are evaluated with different controlling accuracies ϵ and wave number k . It can be seen that for arbitrary frequencies the M2L translation of the improved FDA is always 2–3 times faster than that of the original FDA. The upward passes are accelerated by a factor of about 40%. Since the translation matrices in the downward pass have the same size with those in the upward pass, the same improvement can be expected for the downward pass. The improved FDA can reduce the memory consumption by about 30%. The resulting errors are increased slightly, since additional errors are introduced in the acceleration technique.

5.2. Efficiency and accuracy of the FDBEM

Here the efficiency and accuracy of the FDBEM developed in this paper are demonstrated by solving a benchmark problem: the sound radiation of a unit sphere pulsating with radial velocity $q = \partial u / \partial \mathbf{n} = 1$. This is a Neumann boundary value problem. The velocity potential u on the spherical surface has analytical solution $u = 1/(1 - ik)$.

Radiation problems with wave numbers $k = 2\pi, 4\pi, 8\pi, 16\pi$ and 32π are solved. For each problem, the spherical surface is discretized by curved triangular quadratic elements with element size $h = \lambda/5$, where $\lambda = 2\pi/k$ is the wave length. In the problem with $k = 2\pi$, the wave length is one half of the diameter of the sphere. The mesh consists of $N_e = 720$ elements (see Fig. 7) and the DOF is $N = 6N_e = 4320$. In the problem with the highest wave number $k = 32\pi$, the wave length is 1/32 of the diameter. The corresponding mesh consists of $N_e = 188,796$ elements, $N = 1132776$ DOFs.

One should notice that this series of wave numbers are exactly the characteristic frequencies of the unit sphere, at which the BEM based on the Helmholtz BIE fails to get the correct solution. However, as it will be shown that our FDBEM can still achieve highly accurate results. In the FDBEM, the S2M and L2T matrices are precomputed and saved in memory. The resulting linear system is solved by the GMRES solver with convergence tolerance being equal to ϵ .

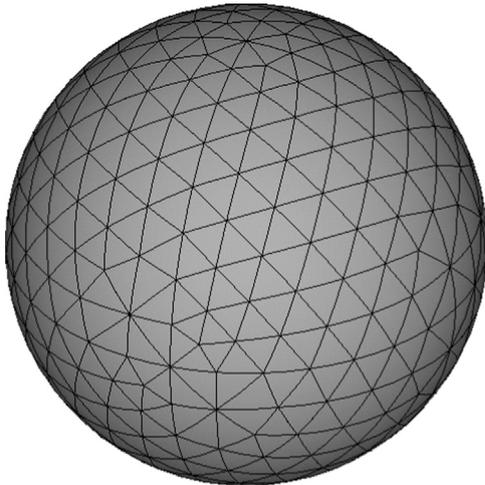


Fig. 7. The spherical surface is discretized with 720 triangular quadratic elements for $k = 2\pi$.

The models are solved with different precision, determined by the controlling accuracy ϵ , and the results are illustrated in Fig. 8. Fig. 8(a) illustrates the L_2 error behavior of the solution u with the refinement of mesh and the change of the controlling accuracy ϵ . It is seen that the FDBEM can attain the given precision $O(\epsilon)$ for $\epsilon \geq 10^{-6}$. When $\epsilon = 10^{-7}$, the error of the solution keeps almost the same with that of $\epsilon = 10^{-6}$, because in this case the error in the numerical evaluation of the singular and near-singular boundary element integrations becomes dominate. Meanwhile, the error curves first go down to $O(\epsilon)$ with the refinement of the meshes, and then tend to rise slightly with the further refinement of the mesh. This is because, in the first stage the BEM discretization error is dominate, and in the second stage the error induced by the translations in the FDA which is of order $O(\epsilon \log N)$ [21] becomes dominate.

The memory consumption, overall CPU time and the CPU time for each iteration are illustrated in Fig. 8(b), (c) and (d), respectively. For each precision ϵ , the CPU time for each iteration and the total memory usage of the FDBEM are of order $O(N \log N)$. The overall CPU time in 8(c) grows almost linearly with N . This is because a large part of the total CPU time is spent on the numerical evaluation of the various singular integrals in the near-field matrix which scales linearly with N . For example, in the case with $\epsilon = 1e-3$, $k = 32\pi$ and $N = 1,132,776$, the overall CPU time is 8181 s, while evaluating the near field part of the matrices \mathbf{H} and \mathbf{G} takes 7053 s, which is about 86% of the overall CPU time.

5.3. Large-scale simulations

Finally, the performance of the FDBEM for large-scale simulations is demonstrated by three numerical examples, two with Neumann boundary conditions and the third with mixed boundary conditions. All of these problems are solved with the controlling accuracy and the GMRES converging tolerance being $\epsilon = 10^{-3}$.

5.3.1. Unit sphere scattering problem

The first example is a scattering problem with a sound-hard unit sphere, thus the boundary condition is of Neumann type, with $q = \partial u / \partial \mathbf{n} = 0$. The velocity potential of the incident plane wave is defined as $u^{\text{inc}}(\mathbf{x}) = e^{i\mathbf{k}\cdot\mathbf{x}\cdot\mathbf{d}}$, with the propagating direction being $\mathbf{d} = (1, 0, 0)$, see Fig. 9(a) for the coordinate system. The wave number is set to be $k = 157.1$, so that the diameter D of the sphere is of 50 wavelengths long, and the non-dimensional wave number is $kD = 314.2$. The analytical solution of the scattered acoustic velocity potential on the spherical surface is given by

$$u^s(\theta) = \sum_{m=0}^{\infty} \left[-i^m (2m+1) \frac{j_m'(k)}{h_m'(k)} \right] h_m(k) \cdot P_m(\cos \theta), \quad (43)$$

where θ is the angle between $\vec{O\mathbf{x}}$ and \mathbf{d} , j_m is the spherical Bessel function of the first kind, h_m is the spherical Hankel function of the second kind, and P_m is the Legendre polynomial of order m . Then the analytical solution can be achieved by $u = u^{\text{inc}} + u^s$.

The spherical surface is discretized into 668,352 curved triangular quadratic elements, and the DOFs $N = 4,040,112$. It takes about 8.1 h to solve this problem. The CPU time for each iteration is about 243.7 s. The memory usage is 27.7 GB. The GMRES solver converged within $N_{it} = 14$ iterations. The acoustic velocity potential of the total sound field u on the surface is illustrated in Fig. 9 (a), and the comparison between the analytical solution and our numerical results of the scattered field u^s on the surface is illustrated in Fig. 9(b). It is shown that our numerical results agree very well with the analytical solution.

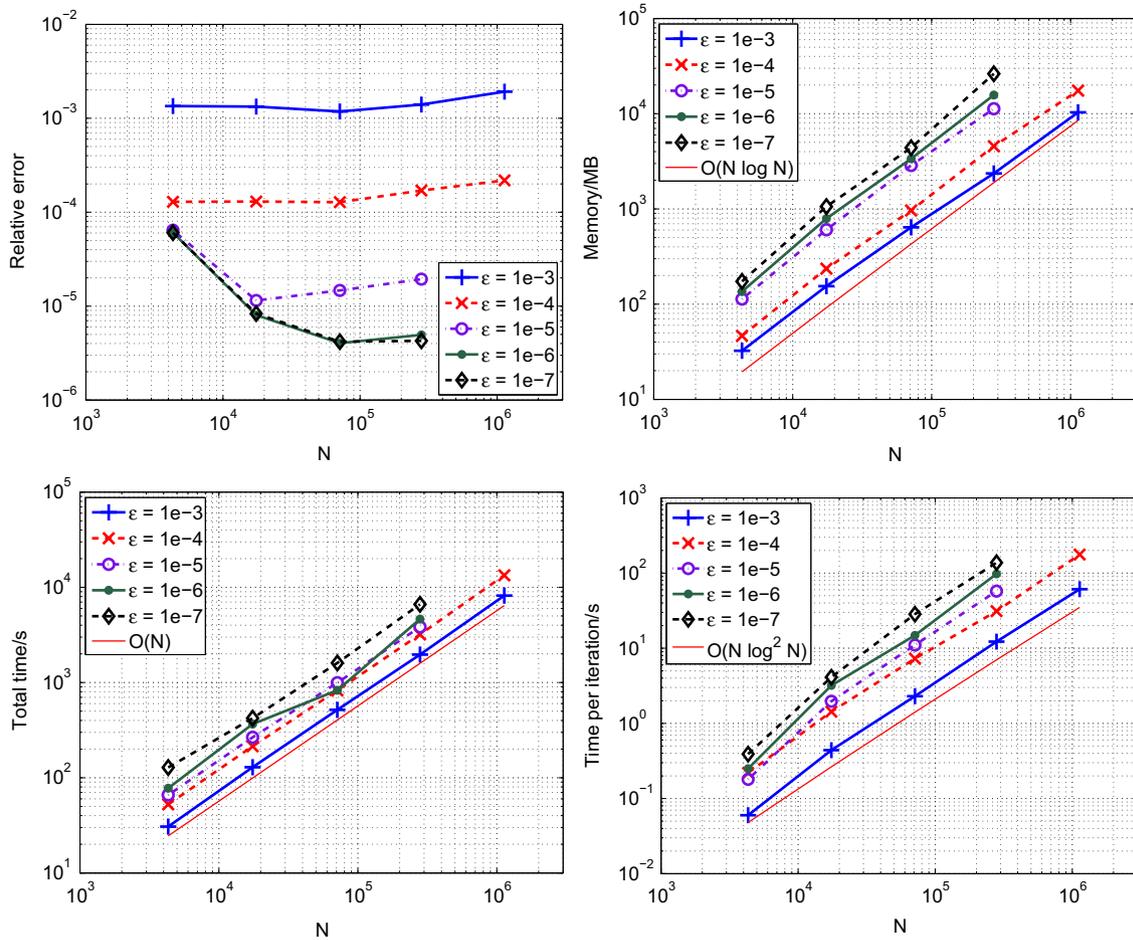


Fig. 8. Numerical results of the unit sphere radiating problem. (a) Relative error versus the DOF, (b) memory consumption versus the DOF, (c) computational time versus the DOF and (d) iteration time versus the DOF.

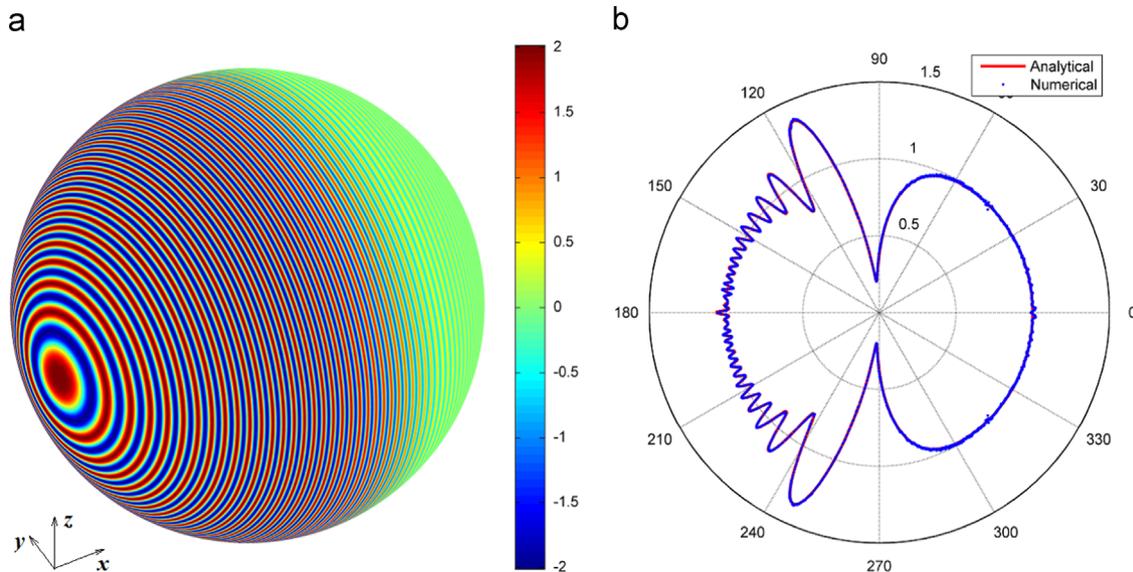


Fig. 9. Results of the unit sphere scattering problem. (a) The real part of total acoustic field and (b) the amplitude of the scattering field.

5.3.2. Submarine scattering problem

The second example is also a scattering problem with the same incident plane wave as that in Section 5.3.1. The scatter is a sound hard submarine with length $D=80$ m. The wave number is set to be $k=12.5$, thus the non-dimensional wave number $kD=1000$ and the length of the submarine is 159 wavelengths. The surface of the

submarine is discretized into 670,564 curved triangular quadratic elements, so the DOFs $N=4,023,384$. It takes 9.3 h to solve this problem. The CPU time for each iteration is 250 s. The GMRES solver converges within 42 iterations. The total memory usage is 24 GB. The acoustic velocity potential of the total field u on the surface is illustrated in Fig. 10.

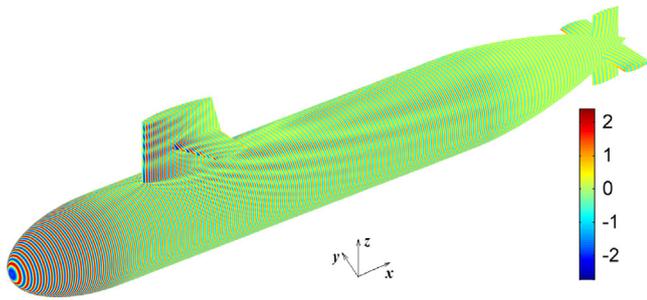


Fig. 10. The real part of the total acoustic field of the submarine scattering problem.

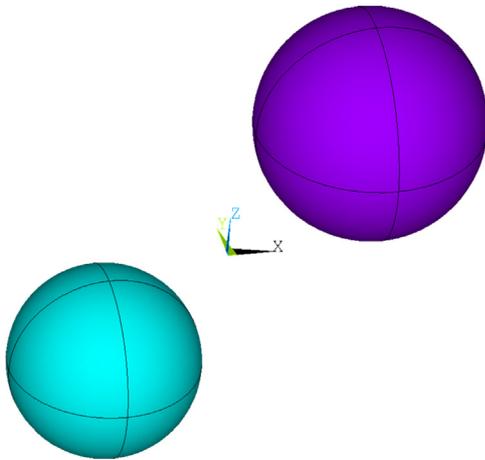


Fig. 11. Geometric model and coordinate system for the large-scale problem with two spheres and mixed boundary conditions. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

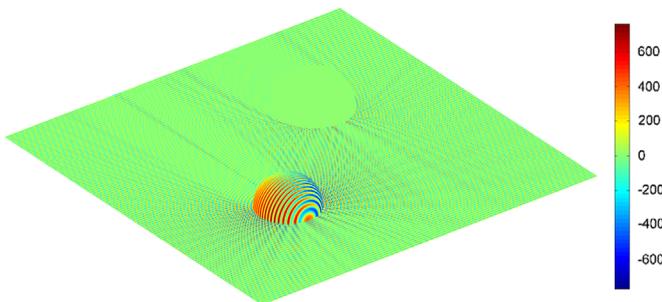


Fig. 12. The sound pressure of the case with two spheres and mixed boundary conditions.

5.3.3. A two-sphere problem with mixed boundary conditions

The model of the third example consists of two spheres, as shown in Fig. 11. The blue one is a unit sphere lies at $(-2, -2, 0)$, oscillating in x direction with velocity $\mathbf{v} = (1, 0, 0)$, thus the boundary condition on its surface is of Neumann type with $q = \mathbf{v} \cdot \mathbf{n}$. The purple sphere lies at $(2, 2, 0)$ with radius $R = 1.2$. It is sound soft, thus the boundary condition on its surface is of Dirichlet type with $u = 0$. The sound pressure of the incident plane wave is defined by $p^{\text{inc}}(\mathbf{x}) = 200e^{i\mathbf{k}\mathbf{x} \cdot \mathbf{d}}$, with the propagating direction being $\mathbf{d} = (0, 1, 0)$. The density of the acoustic medium is $\rho = 1.225$, and the sound speed is $c = 340$. The maximum length of the model is $D \approx 7.86$. The wave number is set to be 20π , thus the non-dimensional wave number $kD \approx 494$. The surfaces are meshed with 266,544 curved triangular quadratic elements, so the DOFs is $N = 1,599,264$. It takes 4.8 h to solve this problem. The CPU time for each iteration is about 188 s. The GMRES solver converged with 26 iterations. The total memory usage is 9.9 GB. Then the acoustic

pressure distribution on the boundary and on a plane in the acoustic field is evaluated, and are illustrated in Fig. 12.

6. Conclusion

The fast directional algorithm (FDA) recently developed in [29] is a highly efficient method for the evaluation of potential summations with Helmholtz kernels. In this paper, a FDA accelerated BEM, denoted by FDBEM, for solving Burton–Miller BIE in a broad frequency range is developed and implemented. The Nyström method based on curved quadratic elements is used to discretize the BIE, by which (1) the edge and corner problems are completely avoided, (2) high accuracy can be achieved and (3) the resulting linear system is more like a summation that is suitable for FDA acceleration. The FDA for the potential summations with the kernels of the Burton–Miller formulation is proposed. The computational efficiency of the FDA is further elevated by exploiting the low-rank property of the translation matrices.

By using the FDBEM, large-scale wideband acoustic problems can be solved with controllable accuracy up to 10^{-6} . The computational time and memory requirements are of order $O(N \log N)$. A representative acoustic scattering problem with dimensionless wave number kD being up to 1000 and the DOF being up to 4 million has been successfully solved within 10 h on a computer with one core and the memory usage is 24 GB.

Acknowledgments

This work is supported by the National Science Foundation of China under Grants 11074201 and 11102154, the Funds for Doctor Research Programs from the Chinese Ministry of Education under Grants 20106102120009 and 20116102110006, and the Doctorate Foundation of Northwestern Polytechnical University under Grant no. CX201220.

References

- [1] Burton AJ, Miller GF. The application of integral equation methods to the numerical solution of some exterior boundary-value problems. Proc R Soc Lond Ser A 1971;323:201–10.
- [2] Schenck HA. Improved integral formulation for acoustic radiation problems. J Acoust Soc Am 1968;44(1):41–58.
- [3] Chen IL, Chen JT, Liang MT. Analytical study and numerical experiments for radiation and scattering problems using the CHIEF method. J Sound Vib 2001;248(5):809–28.
- [4] Greengard L, Rokhlin V. A fast algorithm for particle simulation. J Comput Phys 1987;73:325–48.
- [5] Greengard L, Rokhlin V. A new version of the fast multipole method for the Laplace equation in three dimensions. Acta Numer 1997;6(1):229–69.
- [6] Darve Eric. The fast multipole method: numerical implementation. J Comput Phys 2000;160:195–240.
- [7] Hackbusch W. A sparse matrix arithmetic based on H-matrices. Part I. Introduction to H-matrices. Computing 1999;62:89–108.
- [8] Beylkin G, Coifman R, Rokhlin V. Fast wavelet transforms and numerical algorithms. Pure Appl Math 1991;37:141–83.
- [9] Xiao Jinyou, Wen Lihua, Tausch Johannes. On fast matrix–vector multiplication in wavelet Galerkin BEM. Eng Anal Bound Elem 2009;33(2):159–67.
- [10] Phillips Joel R, White Jacob K. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. IEEE Trans Comput-Aided Des Integr Circuits Syst 1997;16(10):1059–72.
- [11] Xiao Jinyou, Ye Wenjing, Cai Yaxiong, Zhang Jun. Precorrected FFT accelerated BEM for large-scale transient elastodynamic analysis using frequency-domain approach. Int J Numer Methods Eng 2012;90(1):116–34.
- [12] Bebendorf Mario. Approximation of boundary element matrices. Numer Math 2000;86:565–89.
- [13] Bebendorf Mario, Rjasanow Sergej. Adaptive low-rank approximation of collocation matrices. Computing 2003;70:1–24.
- [14] Nishimura N. Fast multipole accelerated boundary integral equation methods. Appl Mech Rev 2002;55(4):299–324.
- [15] Shen L, Liu YJ. An adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton–Miller formulation. Comput Mech 2007;40(3):461–72.

- [16] Chen JT, Chen KH. Applications of the dual integral formulation in conjunction fast multipole method in large-scale problems for 2D exterior acoustics. *Eng Anal Bound Elem* 2004;28:685–709.
- [17] Chaillat Stéphanie, Bonnet Marc, Semblat Jean-François. A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain. *Comput Methods Appl Mech Eng* 2008;197:4233–49.
- [18] Rokhlin V. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Appl Comput Harmon Anal* 1993;1:82–93.
- [19] Cheng Hongwei, Crutchfield William Y, Gimbutas Zydrunas, Greengard Leslie F, Frank Ethridge J, Huang Jingfang, et al. A wideband fast multipole method for the Helmholtz equation in three dimensions. *J Comput Phys* 2006;216:300–25.
- [20] Gumerov Nail A, Duraiswami Ramani. A broadband fast multipole accelerated boundary element method for the three dimensional Helmholtz equation. *J Acoust Soc Am* 2009;125(1):191–205.
- [21] Ying Lexing, Biros George, Zorin Denis. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J Comput Phys* 2004;196:591–626.
- [22] Fong William, Darve Eric. The black-box fast multipole method. *J Comput Phys* 2009;228(23):8712–25.
- [23] Zhang Bo, Huang Jingfang, Pitsianis Nikos P, Sun Xiaobai. A Fourier-series-based kernel-independent fast multipole method. *J Comput Phys* 2011;230(15):5807–21.
- [24] Bajaj Chandrajit, Chen Shun-Chuan, Rand Alexander. An efficient higher-order fast multipole boundary element solution for Poisson–Boltzmann-based molecular electrostatics. *SIAM J Sci Comput* 2011;33(2):826–48.
- [25] Ying Lexing, Biros George, Zorin Denis. A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains. *J Comput Phys* 2006;219(1):247–75.
- [26] Messner Matthias, Bramas Berenger, Coulaud Olivier, Darve Eric. Optimized M2L kernels for the Chebyshev interpolation based fast multipole method, arXiv preprint, arxiv:1210.7292; 2012. p. 1–23.
- [27] Messner Matthias, Schanz Martin, Darve Eric. Fast directional multilevel summation for oscillatory kernels based on Chebyshev interpolation. *J Comput Phys* 2012;231(4):1175–96.
- [28] Brandt Achi. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. *Comput Phys Commun* 1991;65(1):24–38.
- [29] Engquist Björn, Ying Lexing. Fast directional multilevel algorithms for oscillatory kernels. *SIAM J Sci Comput* 2007;29(4):1710–37.
- [30] Tsuji Paul, Ying Lexing. A fast directional algorithm for high-frequency electromagnetic scattering. *J Comput Phys* 2011;230(14):5471–87.
- [31] Guiggiani M, Krishnasamy G, Rudolphi TJ, Rizzo FJ. A general algorithm for the numerical solution of hypersingular boundary integral equations. *J Appl Mech* 1992;59(September):604–14.
- [32] Strain J. Locally corrected multidimensional quadrature rules for singular functions. *SIAM J Sci Comput* 1995;16(4):992–1017.
- [33] Rong Junjie, Wen Lihua, Xiao Jinyou. Efficient improvement of the polar coordinate transformation for evaluating BEM singular integrals on curved elements. *Eng Anal Bound Elem* 2014;38(January):83–93.
- [34] Engquist Björn, Ying Lexing. Fast directional algorithms for the Helmholtz kernel. *J Comput Appl Math* 2010;234:1851–9.