Journal of Computational and Applied Mathematics 234 (2010) 1851–1859

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Fast directional algorithms for the Helmholtz kernel

Björn Engquist, Lexing Ying *

Department of Mathematics, University of Texas, Austin, TX 78712, USA

ARTICLE INFO

Article history:
Received 3 December 2007
Received in revised form 21 May 2008

Keywords:
N-body problems
Scattering problems
Helmholtz equation
Oscillatory kernels
Fast multipole methods
Separated representations
Random sampling
Operator compression
Multidirectional computation
Multiscale methods

ABSTRACT

This paper presents a new directional multilevel algorithm for solving N-body or N-point problems with highly oscillatory kernels. We address the problem by first proving that the interaction between a ball of radius r and a well-separated region has an approximate low rank representation, as long as the well-separated region belongs to a cone with a spanning angle of $O(1/r)$ and is at a distance which is at least $O(r^2)$ away from the ball. Based on this representation, our algorithm organizes the high frequency computation using a multidirectional and multiscale strategy. Our algorithm is proved to have an optimal $O(N \log N)$ computational complexity for any given accuracy when the points are sampled from a two-dimensional surface.

1. Introduction

This paper is concerned with the rapid solutions to a class of N-body problems. Let $\{f_i\}_{1\le i\le N}$ be a set of N charges located at points $\{p_i\}_{1\le i\le N}$ in $\mathbb{R}^3$, with $|p_i| \le K/2$, where $|\cdot|$ is the Euclidean norm and K is a fixed constant. Our goal is to compute the potentials $\{u_i\}_{1\le i\le N}$ defined by

$$u_i = \sum_{j=1}^{N} G(p_i, p_j) \cdot f_j, \tag{1}$$

where $G(x, y) = e^{2\pi\imath|x-y|}/|x-y|$ is Green's function of the Helmholtz equation and is usually called the Helmholtz kernel. Throughout this paper, we use $\imath$ to denote the complex number $\sqrt{-1}$. We have scaled the problem such that the wavelength is equal to one ($\lambda = 1$) and thus high frequencies correspond to problems with large computational domains.

Such a computation comes mainly from applications in acoustic and electromagnetic scattering, where the usual partial differential equations (PDEs) are transformed into boundary integral equations (BIEs) that involve only quantities on the domain boundary. The advantages of the integral formulations are better condition numbers, convenience of handling the boundary conditions at infinity, and reduction in the dimensionality of the unknown field. The discrete versions of the BIEs are, however, dense linear systems which require iterative solution methods. At each step of the iteration, we need to perform the summation in (1).

In most scattering applications, the complexity of a problem scales with the size of its boundary in terms of the wavelength. For a prescribed accuracy, the complexity of (1) depends on K because the wavelength $\lambda = 1$ in our setup. Assuming that the boundary surface in $\mathbb{R}^3$ is discretized with a fixed number of points per wavelength, the number of samples N is then of order $O(K^2)$. Moreover, since the unknown field in a BIE formulation is supported only on the boundary, the quadrature points $\{p_i\}$ on the boundary have highly non-uniform distribution.

* Corresponding author.
E-mail addresses: engquist@math.utexas.edu (B. Engquist), lexing@math.utexas.edu (L. Ying).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.08.036

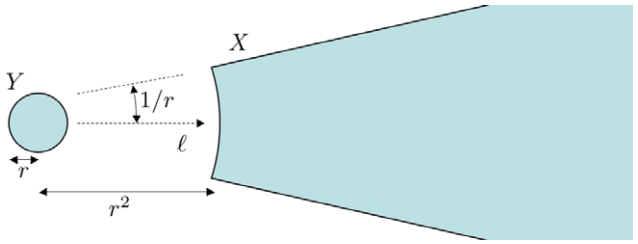

Fig. 1. Two sets Y and X that satisfy the directional parabolic separation condition.

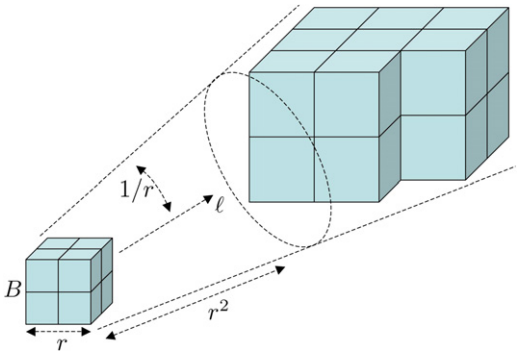

Fig. 2. For each box B, its far field is partitioned into multiple wedges. Our algorithm constructs one low rank separated representation for each wedge. Each resulting representation can be used to accelerate the interaction computation between B and all of the boxes in that wedge.

Direct computation of (1) requires $O(N^2)$ operations, which can be very slow for large values of N. During the past few decades, much attention has been devoted to the development of algorithms which evaluate (1) efficiently without compromising accuracy (see, for example, [1–3] for methods using fast Fourier transforms, [4,5] for methods based on local Fourier bases, and [6–8] for FMM-type methods). In this paper, we describe a new directional multilevel algorithm that has $O(N \log N)$ complexity.

The starting point of our approach is a geometric consideration. Suppose Y is a ball of radius r centered at a point c, and X is the set of all points which are at a distance r^2 or greater from the origin and belong to a cone centered at c with spanning angle $1/r$ (see Fig. 1 for an illustration). Whenever two sets Y and X obey this geometric configuration, we say that Y and X satisfy the *directional parabolic separation condition*. At the heart of our algorithm is a *directional low rank property*, which states that the interaction between Y and X via the Helmholtz kernel $G(x, y)$ is approximately of low rank for any fixed accuracy ε. More precisely, there exist functions $\{\alpha_i(x)\}$ and $\{\beta_i(y)\}$ of x and y such that for any $x \in X$ and $y \in Y$:

$$\left| G(x, y) - \sum_{i=1}^{T(\varepsilon)} \alpha_i(x)\beta_i(y) \right| < \varepsilon$$

where the rank $T(\varepsilon)$ has an upper bound that is independent of r.

Our algorithm starts by partitioning the domain recursively using an octree, which is similar to the standard FMM [9]. The top part of the octree that contains the boxes with widths greater than or equal to 1 is called the high frequency regime, while the bottom part that contains the boxes with widths less than 1 is called the low frequency regime. In the low frequency regime, the interactions are computed using the kernel-independent FMM [10]. In the high frequency regime, the computation is organized in a multidirectional way. For a box B of size r, its far field, defined to be the region at least r^2 away from B, is partitioned into a group of wedges that satisfy the directional parabolic separation condition (see Fig. 2). The calculation of the interactions between B and all of the boxes in a specific wedge can be accelerated using the low rank representation associated with this wedge. This framework is repeated recursively at all levels to achieve the optimal $O(N \log N)$ complexity.

The presentation of this paper follows [11]. The new contribution, which is presented in Section 2, is an improved algorithm for the construction of low rank separated representation, which gives better efficiency and stability.

2. Directional low rank property

The main theoretical result of this paper is the directional low rank property. Suppose $r \geq \sqrt{3}$, and let $Y = B(0, r)$ and $X = \{x : \theta(x, \ell) \leq 1/r, |x| \geq r^2\}$. where ℓ is a given unit vector and $\theta(x, \ell)$ is the angle between vectors x and ℓ. The geometric relationship between Y and X is illustrated in Fig. 1.

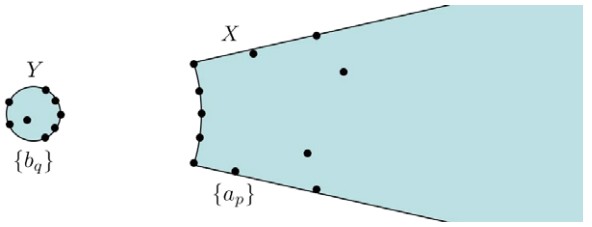

Fig. 3. Constructions of the separated representation between X and Y. $\{b_q\}$ are the samples associated with the columns in A_c (Step 2). $\{a_p\}$ are the samples associated with the columns in A_r (Step 3).

Definition 2.1. Let $f(x, y)$ be a function for $x \in X$ and $y \in Y$. We say $f(x, y)$ has a T-term ε-expansion for X and Y if there exist functions $\{\alpha_i(x), 1 \le i \le T\}$ and $\{\beta_i(y), 1 \le i \le T\}$ such that, for all $x \in X$ and $y \in Y$,

$$\left| f(x, y) - \sum_{i=1}^{T} \alpha_i(x)\beta_i(y) \right| \le \varepsilon.$$

The importance of this definition is that $\{\alpha_i(x)\}$ and $\{\beta_i(y)\}$ depend only on x and y, respectively. Expansions of this type are called *separated*. The following theorem is a precise statement of the directional low rank property.

Theorem 2.2. *Suppose $\varepsilon > 0$. For X and Y defined above, there exists a number $T(\varepsilon)$, which is independent of r, such that $e^{2\pi\imath|x-y|}/|x-y|$ has a $T(\varepsilon)$-term ε-expansion.*

The proof of Theorem 2.2 can be found in [11]. The message of this theorem is that, for a fixed ε, the number of terms in an ε-expansion is independent of r, as long as X and Y satisfy the directional parabolic separation condition; i.e., X belongs to a cone with spanning angle $1/r$ and it is an order $O(r^2)$ distance away from Y.

The ε-expansion for X and Y introduced here is a directional separated representation. It is directional since for a given direction ℓ the expansion is valid only for X centered along ℓ. It is separated since the two sets of functions $\{\alpha_i(x)\}$ and $\{\beta_i(y)\}$ depend only on x and y, respectively. In [11], we described a randomized algorithm for constructing these functions. In the rest of this section, we propose an improved version which gives better results in practice. This new version, which is again based on random sampling, consists of the following steps:

(1) Sample Y randomly by a set of samples $\{y_j\}_{1\le j\le N_Y}$ with 2 to 3 points per wavelength. Similarly, sample X to obtain a set of samples $\{x_i\}_{1\le i\le N_X}$. Let A be the matrix defined by $A_{ij} = G(x_i, y_j)$, for $1 \le i \le N_X$ and $1 \le j \le N_Y$. In fact, Theorem 2.2 states that A can be factorized, within error $O(\varepsilon)$, into the product of two matrices, the first containing $T(\varepsilon)$ columns and the second containing $T(\varepsilon)$ rows.

(2) We look for a set of $T(\varepsilon)$ columns of A that has large $T(\varepsilon)$-dimensional volume. To do this efficiently, we use a version of the random projection techniques that are known to preserve the volume [12,13]. We define A_1 be a submatrix of A containing a set of N_1 randomly selected rows with $N_1 \approx 3 \cdot T(\varepsilon)$. The *pivoted* QR factorization of A_1 gives us a decomposition $A_1P_1 = Q_1R_1$, where P_1 is a permutation matrix, Q_1 is orthonormal and R_1 is upper triangular. Now identify the diagonal elements of R_1 which are less than ε and truncate the associated columns of Q_1 and rows of R_1. Denote the resulting matrices by $Q_{1,c}$ and $R_{1,c}$. It is clear that $Q_{1,c}R_{1,c} = A_{1,c}$, where $A_{1,c}$ is the submatrix of A_1 containing the columns from which the matrix $Q_{1,c}$ is generated. We denote by A_c the submatrix of A that consists of the same columns. The samples (of Y) associated with these columns are denoted $\{b_q\}$ (see Fig. 3).

(3) We look for a set of $T(\varepsilon)$ rows of A that has large $T(\varepsilon)$-dimensional volume. Let A_2 be a submatrix of A containing a set of N_2 randomly selected columns with $N_2 \approx 3 \cdot T(\varepsilon)$. After repeating the previous step on A_2^*, we have the factorization $R_{2,r}^*Q_{2,r}^* = A_{2,r}$ where $A_{2,r}$ is a submatrix of A_2, $Q_{2,r}$ is orthonormal, and $R_{2,r}$ is upper triangular. We denote by A_r the submatrix of A that consists of the rows that appeared in $A_{2,r}$ and by $\{a_p\}$ the samples (of X) associated with these rows (see Fig. 3).

(4) We randomly pick a set of rows S and a set of columns T. Set A_3 to be the minor containing the elements in these rows and columns, $A_{c,S}$ to be the submatrix of A_c containing the rows in S, and $A_{r,T}$ to be the submatrix of A_r containing the columns in T. Next, we choose $D = (A_{c,S})^+A_3(A_{r,T})^+$, where $(\ )^+$ stands for pseudoinverse. We claim that $|A - A_cDA_r| = O(\varepsilon)$. Such an approximate factorization is often called a pseudoskeleton approximation of A in the literature (see [14]). Notice that the matrix D has only $O(T(\varepsilon))$ rows and columns. Denoting the entries of D by d_{qp}, we can rewrite the previous statement as for all x_i and y_j

$$\left| G(x_i, y_j) - \sum_{p,q} G(x_i, b_q) \cdot d_{qp} \cdot G(a_p, y_j) \right| = O(\varepsilon).$$

(5) Finally, since $\{x_i\}$ and $\{y_j\}$ sample the sets X and Y with a constant number of points per wavelength, it is reasonable to expect for any $x \in X \cap B(0, K)$ and $y \in Y$,

$$\left| G(x, y) - \sum_{p,q} G(x, b_q) \cdot d_{qp} \cdot G(a_p, y) \right| = O(\varepsilon). \tag{2}$$

This random procedure works quite well as we will see in Section 4. The random projections reduce the skeleton selection problem in a high-dimensional space (N_X or N_Y) to a much smaller dimensional space ($O(T(\varepsilon))$), while at the same time preserving the volume approximately. Although the pivoted QR factorization is a greedy algorithm, it works well in practice and selects columns with almost the maximal volume efficiently.

As we can see from Fig. 3, the samples $\{b_q\}$ and $\{a_p\}$ generated above are highly non-uniform. In fact, they have a strong directional preference and cluster at the parts of Y and X that are close to each other. This is a clear illustration of the advantage of the directionality: we only need to sample the part close to X (or Y) densely, and everywhere else, we can use very sparse grids.

In order to prepare for the discussion of our main algorithm in Section 3, it is useful to introduce several definitions here. Suppose we have a set of charges $\{f_i\}$ located at points $\{y_i\}$ in Y. Using the representation obtained from the randomized procedure, we have

$$\left| \sum_i G(x, y_i) f_i - \sum_q G(x, b_q) \left(\sum_p d_{qp} \sum_i G(a_p, y_i) f_i \right) \right| = O(\varepsilon).$$

This states that we can place a set of charges $\{\sum_p d_{qp} \sum_i G(a_p, y_i) f_i\}$ at points $\{b_q\}$ in order to reproduce the potential generated by the charges $\{f_i\}$ located at points $\{y_i\}$. To this end, these charges are called the *directional outgoing equivalent charges* of Y in direction ℓ, and the points $\{b_q\}$ are called the *directional outgoing equivalent points* of Y in direction ℓ. In addition, we refer to the quantities $\{\sum_i G(a_p, y_i) f_i\}$ as the *directional outgoing check potentials* of Y in direction ℓ and the points $\{a_p\}$ as the *directional outgoing check points* of Y in direction ℓ.

Now, let us reverse the roles of X and Y. Suppose we have a set of charges $\{f_i\}$ located at points $\{x_i\}$ in X. For the potential at y inside Y, we have

$$\left| \sum_i G(y, x_i) f_i - \sum_p G(y, a_p) \left(\sum_q d_{qp} \sum_i G(b_q, x_i) f_i \right) \right| = O(\varepsilon).$$

This states that we can put a set of charges $\{\sum_q d_{qp} \sum_i G(b_q, x_i) f_i\}$ at points $\{a_p\}$, and they reproduce the potential generated by the charges $\{f_i\}$ located at points $\{x_i\}$. Therefore, we call these charges the *directional incoming equivalent charges* of Y in direction ℓ and the locations $\{x_q^c\}$ the *directional incoming equivalent points* of Y in direction ℓ. Similarly, $\{\sum_i G(b_q, x_i) f_i\}$ are called the *directional incoming check potentials* of Y in direction ℓ, and the locations $\{b_q\}$ are called the *directional incoming check points* of Y in direction ℓ.

So far, we have assumed that $Y = B(0, r)$. However, if Y is centered at a different location, the locations $\{a_p\}$ and $\{b_q\}$ can be obtained through a simple translation since the kernel $G(x, y)$ is translation invariant. Moreover, the matrix D remains the same. We note that none of these quantities depends on the positions $\{p_i\}$ and $\{f_i\}$ and, therefore, all of them can be precomputed and used over and over for different evaluations of (1). The representation in (2) is highly storage efficient. For any fixed r and ℓ, we only need to store $\{a_p\}$, $\{b_q\}$, and D, all of which have size $O(1)$ for a fixed accuracy ε.

3. Algorithm description

Without loss of generality, we assume that $K = 2^{2L}$ for a positive integer L. Similar to the HF–FMM [7], our main data structure is an octree. The top level box of width K contains all of the points $\{p_i\}$. In the rest of this paper, B denotes a box in the octree and w its width. We say a box B is in the low frequency regime if $w < 1$, and B is in the high frequency regime if $w \geq 1$.

In the high frequency regime of the octree, no adaptivity is used; i.e., every nonempty box is further partitioned until the width of the box is less than 1. In the low frequency regime, a box B is partitioned as long as the number of points in B is greater than a fixed constant N_p. In practice, the value of N_p is chosen to optimize the computational complexity.

As we have mentioned already, an FMM algorithm for the Laplace kernel can be easily modified to handle the low frequency case. For a box B in the low frequency regime, its data structure follows the description of the kernel-independent FMM in [10]. We modify the notation slightly to accommodate the discussion of the current algorithm.

- $\{y_k^{B,o}\}$, $\{f_k^{B,o}\}$, $\{x_k^{B,o}\}$, and $\{u_k^{B,o}\}$ are, respectively, the *outgoing* equivalent points, equivalent charges, check points, and check potentials.
- $\{y_k^{B,i}\}$, $\{f_k^{B,i}\}$, $\{x_k^{B,i}\}$, and $\{u_k^{B,i}\}$ are, respectively, the *incoming* equivalent points, equivalent charges, check points, and check potentials.

Now let us consider a box B in the high frequency regime. The near field N^B is the union of all of the boxes $\{A\}$ that satisfy $\text{dist}(A, B) \leq w^2$. The far field F^B is the complement of N^B. In light of the preceding discussion, we need to partition F^B into

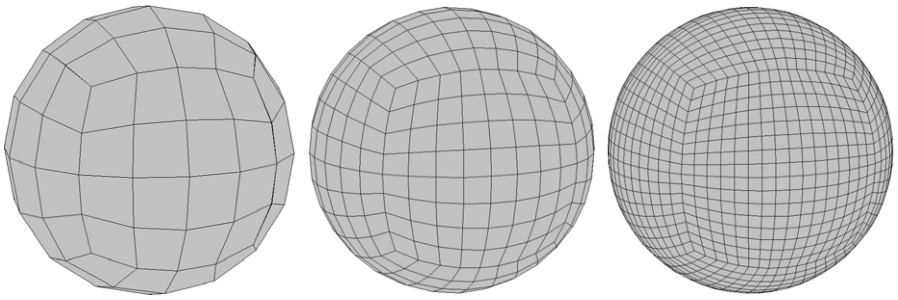

Fig. 4. The wedges cut the unit sphere into $96 \cdot w^2$ pieces. From left to right, $w = 1, 2, 4$.

a group of directional regions, each belonging to a cone with spanning angle $O(1/w)$. We index these wedges using their center directions $\{\ell\}$. In Fig. 4, we illustrate how these wedges cut the unit sphere into $96 \cdot w^2$ pieces for $w = 1, 2, 4$.

This construction has the advantage of ensuring a hierarchical structure of the wedges across adjacent levels. For any directional index ℓ of B, one can always find an index ℓ' of the box with width $w/2$ such that the ℓth wedge of B is contained in the ℓ'th wedge of each of B's children.

For each box B and each direction ℓ, we summarize the relevant quantities as follows:

- $\{y_k^{B,o,\ell}\}$, $\{f_k^{B,o,\ell}\}$, $\{x_k^{B,o,\ell}\}$, and $\{u_k^{B,o,\ell}\}$ are the *outgoing directional* equivalent points, equivalent charges, check points, and check potentials respectively.
- $\{y_k^{B,i,\ell}\}$, $\{f_k^{B,i,\ell}\}$, $\{x_k^{B,i,\ell}\}$, and $\{u_k^{B,i,\ell}\}$ are the *incoming directional* equivalent points, equivalent charges, check points, and check potentials respectively.

Similar to a standard FMM algorithm, our new algorithm utilizes several translation operators. Following tradition, we name these operators M2M, M2L, and L2L translations. The translation operators for boxes in the low frequency regime are detailed already in [10]. The operators in the high frequency regime are more complicated. The main reason is that the computations are now directional.

For a box B in the high frequency regime, the *M2M operator* constructs the outgoing directional equivalent charges of B from the outgoing equivalent charges of B's children. There are two cases to consider. In the first case, $w = 1$. The children boxes have only nondirectional equivalent charges. The M2M operator iterates over all of the directional indices $\{\ell\}$ of B, and the steps for a fixed direction ℓ are as follows:

(1) Use $\bigcup_C \{y_k^{C,o}\}$ as source points in B and $\bigcup_C \{f_k^{C,o}\}$ as source charges. Here the union is taken over all of the children boxes of B.
(2) Compute $\{u_k^{B,o,\ell}\}$ at points $\{x_k^{B,o,\ell}\}$ with kernel evaluation, and then obtain $\{f_k^{B,o,\ell}\}$ by multiplying $\{u_k^{B,o,\ell}\}$ with the matrix D associated with B and direction ℓ.

In the second case, $w > 1$. Now the children boxes have directional equivalent charges as well. The M2M operator again iterates over all of the directional indices $\{\ell\}$ of B. The steps for a fixed direction ℓ are as follows:

(1) Pick ℓ', which is a direction associated with the boxes of width $w/2$, such that the wedge of B indexed by ℓ is contained in the wedge indexed by ℓ' of each of B's children. The existence of ℓ' is ensured by the way we partition F^B.
(2) Use $\bigcup_C \{y_k^{C,o,\ell'}\}$ as source points in B and $\bigcup_C \{f_k^{C,o,\ell'}\}$ as source charges. Here the union is taken over all the children boxes of B.
(3) Compute $\{u_k^{B,o,\ell}\}$ at $\{x_k^{B,o,\ell}\}$ with kernel evaluation and then obtain $\{f_k^{B,o,\ell}\}$ by multiplying $\{u_k^{B,o,\ell}\}$ with the matrix D associated with B and direction ℓ.

The *L2L operator* constructs the incoming check potentials of B's children from the incoming directional check potentials of B. Again there are two cases to consider. In the first case $w = 1$. The children boxes have only nondirectional check potentials. The L2L operator iterates over all of the directional indices $\{\ell\}$ of B, and the steps for a fixed direction ℓ are as follows:

(1) Compute $\{f_k^{B,i,\ell}\}$ from $\{u_k^{B,i,\ell}\}$ by multiplying it with the appropriate D matrix.
(2) For each child C of the box B, add to $\{u_k^{C,i}\}$ the potentials evaluated at $\{x_k^{C,i}\}$ using $\{f_k^{B,i,\ell}\}$ as the source charges at $\{y_k^{B,i,\ell}\}$.

In the second case, $w > 1$. Now the children boxes have directional equivalent charges. The L2L operator iterates over all of the directional indices $\{\ell\}$ of B. The steps for a fixed direction ℓ are as follows:

(1) Pick ℓ', which is a direction associated with the boxes of width $w/2$, such that the wedge of B indexed by ℓ is contained in the wedge indexed by ℓ' of each of B's children. The existence of ℓ' is ensured by the way we partition F^r.
(2) Compute $\{f_k^{B,i,\ell}\}$ from $\{u_k^{B,i,\ell}\}$ by multiplying it with the appropriate D matrix.
(3) For each child C of the box B, add to $\{u_k^{C,i,\ell'}\}$ the potentials evaluated at $\{x_k^{C,i,\ell'}\}$ using $\{f_k^{B,i,\ell}\}$ as the source charges at $\{y_k^{B,i,\ell}\}$.

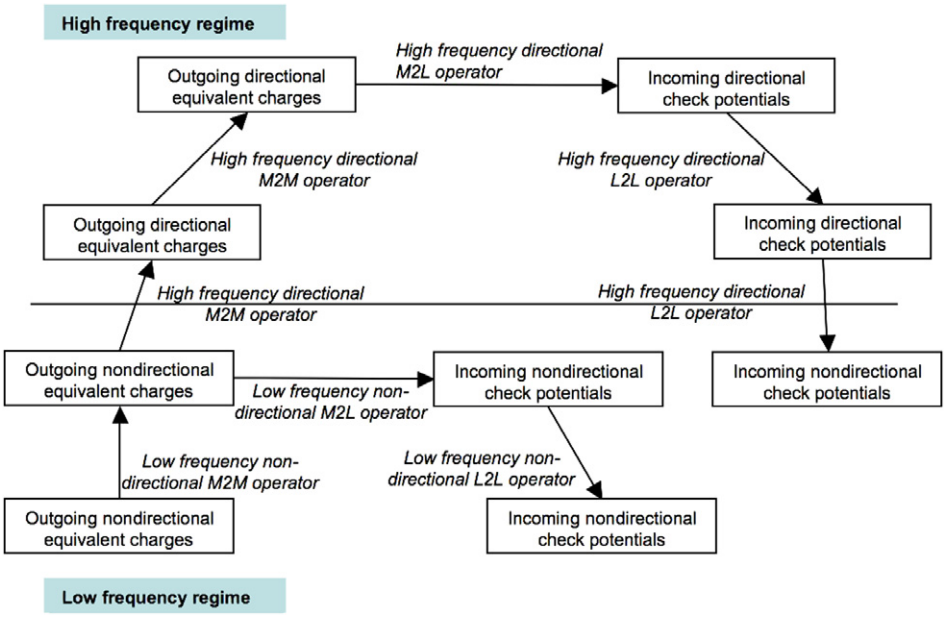

Fig. 5. A small part of the octree used in the computation. Each rectangular region stands for a box of the octree. The diagram shows how the outgoing nondirectional equivalent charges from a leaf box have been transformed into incoming nondirectional check potentials at other leaf boxes. Far field interaction involves directional computation in the high frequency regime.

Finally, the *M2L operator* is applied to pairs of boxes A and B on the same level of the octree. They need to be on each other's interaction lists. Consider ℓ and ℓ' such that B falls into the wedge of A indexed by ℓ while A falls into the wedge of B indexed by ℓ'. The implementation of the M2L operator contains only one step:

(1) Add to $\{u_k^{B,i,\ell'}\}$ the potentials evaluated at $\{x_k^{B,i,\ell'}\}$ using the charges $\{f_k^{A,o,\ell}\}$ at points $\{y_k^{A,o,\ell'}\}$.

To summarize the discussion on the transition operators, we would like to emphasize that all of these operators involve only kernel evaluation and matrix–vector multiplication with precomputed matrices. Therefore, they are simple to implement and highly efficient.

Now we are ready to give the overall structure of our new algorithm. It contains the following steps:

(1) Construct the octree. In the high frequency regime, the boxes are partitioned uniformly. In the low frequency regime, a leaf box contains at most N_p points.
(2) Travel up in the octree and visit the boxes in the low frequency regime. For each box B, compute $\{f_k^{B,o}\}$. This is done using the low frequency nondirectional M2M operator.
(3) Travel up in the octree and visit the boxes in the high frequency regime. For every such box B, use the high frequency directional M2M operator to compute $\{f_k^{B,o,\ell}\}$ for each outgoing direction ℓ. We skip the boxes with width greater than $\sqrt{K}$ since their interaction lists are empty.
(4) Travel down in the octree and visit the boxes in the high frequency regime. For every such box B and for each direction ℓ,
   (a) Use the high frequency directional M2L operator to transform $\{f_k^{A,o,\ell}\}$ of all of the boxes $\{A\}$ in B's interaction list and in direction ℓ. Next, add the result to $\{u_k^{B,i,\ell}\}$.
   (b) Perform the high frequency directional L2L operator to transform $\{u_k^{B,i,\ell}\}$ to the incoming check potentials for B's children.

   Again, we skip the boxes with width greater than $\sqrt{K}$.
(5) Travel down in the octree. For every box B in the low frequency regime,
   (a) Transform $\{f_k^{A,o}\}$ of all of the boxes $\{A\}$ in B's interaction list via the low frequency nondirectional M2L operator. Next, add the result to $\{u_k^{B,i}\}$.
   (b) Perform the low frequency directional L2L operator. Depending on whether B is a leaf box or not, add the result to the incoming check potentials of B's children or to the potentials at the original points inside B.

An illustration of the structure of the algorithm is given in Fig. 5. In the description of the algorithm, we have assumed that the octree is full. This ensures that the M2L operator itself is sufficient to transform outgoing data to incoming data. When the octree is constructed adaptively, the situation is much more complicated as one needs to keep the so-called U, V, W, and X lists for each box B in the low frequency regime. The necessary modifications to include this can be found in [15,9,10].

Table 1
The separation rank of the directional separated representation for different choices of requested accuracy ε and box size w.

	$w = 1$	$w = 2$	$w = 4$	$w = 8$
$\varepsilon =$ 1e−4	45	45	45	45
$\varepsilon =$ 1e−6	84	82	80	80
$\varepsilon =$ 1e−8	122	113	111	111

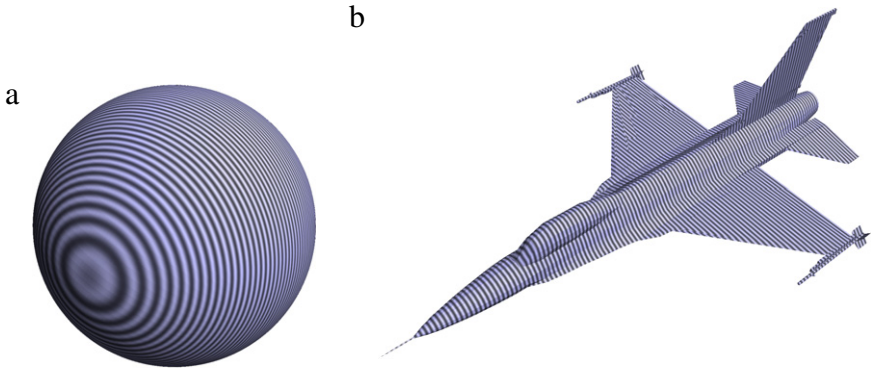

Fig. 6. (a) The sphere. (b) The F16 model.

The following theorem from [11] shows that our algorithm has optimal complexity.

Theorem 3.1. *Let $\mathbb{S}$ be a surface in $B(0, 1/2)$. Suppose that for a fixed K the points $\{p_i, 1 \le i \le N\}$ are samples of $K\mathbb{S}$, where $N = O(K^2)$ and $K\mathbb{S} = \{K \cdot p, p \in \mathbb{S}\}$ (the surface obtained by magnifying $\mathbb{S}$ by a factor of K). Then, for any prescribed accuracy, the proposed algorithm has a computational complexity $O(K^2 \log K) = O(N \log N)$.*

4. Numerical results

Our algorithm is implemented in C++ and all of the computational results in this sections are obtained on a desktop with a 3.0 GHz CPU. Due to memory constraint, we restrict ourselves to problems which are at most 256 wavelengths.

As we have mentioned earlier, the equivalent positions, along with the matrix D, can be computed for all sizes w and directions ℓ. This precomputation step takes about 30 min on our desktop even for the highest accuracy $\varepsilon =$ 1e−8, and the resulting data is less than 100 MB.

Let us first study the performance of the randomized procedure presented in Section 2. In Table 1, we list the number of terms in the separated representation for a box B of width w and one of its wedges for different accuracy ε. We can see from the results that the separation rank is bounded by a constant which is independent of the values of w. This is consistent with our theoretical estimate in Theorem 2.2. In fact, as w grows, it seems that the separation rank decays slightly. The results also show that the separation rank seems to increase linearly with respect to $\log(1/\varepsilon)$.

We use two objects to test the proposed algorithm: a sphere in Fig. 6(a) and an F16 model in Fig. 6(b). In our experiments, the surface of each object is represented by a triangular mesh. The point set $\{p_i\}$ is generated by sampling the triangular mesh randomly with several points per wavelength.

Before reporting the results, let us summarize some relevant notations: N is the number of points, K is the size of the problem in terms of the wavelength, ε is the prescribed error threshold such that the final error is to be bounded by a constant multiple of ε, T_a is the running time of our algorithm in seconds, T_d is the running time of the direct evaluation in seconds, T_d/T_a is the speedup factor, and ε_a is the estimated error of our algorithm.

Next, we study the performance of the proposed algorithm for increasing values of K with the sampling rate fixed at 20 points per wavelength. The results of the sphere (see Fig. 6(a)) are summarized in Table 2, and the ones of the F16 model are given in Table 3. It is clear from these results that the complexity of our algorithm grows almost linearly with respect to the number of points. Our algorithm is also quite accurate and stable. The error grows very slowly with K, which indicates that our low rank separated representation does not introduce numerical stability. This is to a large extent attributed to the pivoted QR factorization. A rigorous proof of the stability of the randomized construction process is still missing and we are currently looking at this issue.

The running times in these two tables show that for the same K our algorithm performs much better on the airplane model than the sphere. The reason is rooted in the directional nature of our algorithm: Since the plane is relatively fat and has a preferred direction, there are much fewer directions ℓ to visit in our algorithm.

Finally, we report in Table 4 the performance of the proposed algorithm for different sampling densities. In this test, we use only the sphere model, since low sampling rates such as 5 point per wavelength cannot be achieved for complicated

Table 2
Results of the sphere with the Helmholtz kernel.

(K, ε)	N	T_a (s)	T_d (s)	T_d/T_a	ε_a
(16, 1e−4)	3.22e+5	9.50e+1	1.21e+4	1.28e+2	5.08e−4
(32, 1e−4)	1.29e+6	4.28e+2	1.95e+5	4.55e+2	5.91e−4
(64, 1e−4)	5.15e+6	1.97e+3	3.04e+6	1.54e+3	6.30e−4
(16, 1e−6)	3.22e+5	2.42e+2	1.18e+4	4.86e+1	2.92e−6
(32, 1e−6)	1.29e+6	1.21e+3	1.87e+5	1.54e+2	2.12e−6
(64, 1e−6)	5.15e+6	5.95e+3	3.13e+6	5.27e+2	3.70e−6
(16, 1e−8)	3.22e+5	5.11e+2	1.22e+4	2.39e+1	7.16e−8
(32, 1e−8)	1.29e+6	2.62e+3	1.96e+5	7.51e+1	9.19e−8
(64, 1e−8)	5.15e+6	1.25e+4	3.15e+6	2.52e+2	9.14e−8

Table 3
Results of the F16 model with the Helmholtz kernel.

(K, ε)	N	T_a (s)	T_d (s)	T_d/T_a	ε_a
(32, 1e−4)	1.87e+5	5.00e+1	4.17e+3	8.34e+1	6.13e−4
(64, 1e−4)	7.46e+5	2.27e+2	6.58e+4	2.90e+2	6.69e−4
(128, 1e−4)	2.98e+6	1.04e+3	1.03e+6	9.87e+2	6.89e−4
(256, 1e−4)	1.19e+7	5.04e+3	1.64e+7	3.25e+3	7.63e−4
(32, 1e−6)	1.87e+5	1.18e+2	4.06e+3	3.44e+1	2.72e−6
(64, 1e−6)	7.46e+5	6.12e+2	6.56e+4	1.07e+2	3.30e−6
(128, 1e−6)	2.98e+6	3.07e+3	1.06e+6	3.45e+2	4.16e−6
(32, 1e−8)	1.87e+5	2.38e+2	4.07e+3	1.71e+1	6.34e−8
(64, 1e−8)	7.46e+5	1.29e+3	6.64e+4	5.14e+1	8.10e−8
(128, 1e−8)	2.98e+6	6.42e+3	1.06e+6	1.64e+2	6.55e−8

Table 4
Results of different sampling rates of the sphere with $K = 32$.

Points per λ	N	T_a (s)	T_d(s)	T_d/T_a	ε_a
5	8.06e+4	3.56e+2	7.62e+2	2.14e+0	4.98e−4
10	3.22e+5	3.88e+2	1.21e+4	3.12e+1	5.34e−4
20	1.29e+6	4.28e+2	1.95e+5	4.55e+2	5.91e−4

geometries such as the airplane model in general. The results show that at low sampling rates the running time of the direct evaluation is greatly reduced, while the time of our algorithm is mostly unaffected since its complexity mostly depends on the size of the object in wavelengths. At low sampling rates, one needs to work with larger values of K in order to reproduce the speedup factors reported in Tables 2 and 3.

Though we sample the object with a uniform distribution in these examples, our algorithm handles adaptive sampling. In fact, the randomly generated points automatically test the adaptivity of our algorithm since these points will cluster at certainly locations. A closer look at different parts of the running time shows that our algorithm spends most of the time in the high frequency regime. Therefore, for problems that require adaptive sampling to capture sub-wavelength features, our algorithm would have virtually the same running time. On the other hand, the running time of direct evaluation will increase significantly.

Compared with the results reported in [7], our algorithm has a relatively small constant, though both algorithms have the optimal $O(N \log N)$ complexity. Due to its directional nature, our algorithm performs significantly better for elongated objects such as the airplane model in Table 3. Finally, we would like to point out that our algorithm is more general in the sense that it can handle other oscillatory kernels such as $e^{2\pi i|x-y|}$ without any modification (see [11]).

References

[1] E. Bleszynski, M. Bleszynski, T. Jaroszewicz, AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems, Radio Sci. 31 (1996) 1225–1252.
[2] N. Bojarski, K-space formulation of the electromagnetic scattering problems, Tech. rep., Air Force Avionic Lab. Technical Report AFAL-TR-71-75, 1971.
[3] O.P. Bruno, L.A. Kunyansky, A fast, high-order algorithm for the solution of surface scattering problems: Basic implementation, tests, and applications, J. Comput. Phys. 169 (1) (2001) 80–110.
[4] B. Bradie, R. Coifman, A. Grossmann, Fast numerical computations of oscillatory integrals related to acoustic scattering. I, Appl. Comput. Harmon. Anal. 1 (1) (1993) 94–99.
[5] F.X. Canning, Sparse approximation for solving integral equations with oscillatory kernels, SIAM J. Sci. Stat. Comput. 13 (1) (1992) 71–87.
[6] V. Rokhlin, Diagonal forms of translation operators for the Helmholtz equation in three dimensions, Appl. Comput. Harmon. Anal. 1 (1) (1993) 82–93.
[7] H. Cheng, W.Y. Crutchfield, Z. Gimbutas, L.F. Greengard, J.F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, J. Zhao, A wideband fast multipole method for the Helmholtz equation in three dimensions, J. Comput. Phys. 216 (1) (2006) 300–325.
[8] W. Chew, E. Michielssen, J.M. Song, J.M. Jin (Eds.), Fast and Efficient Algorithms in Computational Electromagnetics, Artech House, Inc., Norwood, MA, USA, 2001.
[9] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.

[10] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, J. Comput. Phys. 196 (2) (2004) 591–626.
[11] B. Engquist, L. Ying, Fast directional multilevel algorithms for oscillatory kernels, SIAM J. Sci. Comput. 29 (4) (2007) 1710–1737.
[12] S. Dasgupta, A. Gupta, An elementary proof of a theorem of Johnson and Lindenstrauss, Random Struct. Algorithms 22 (1) (2003) 60–65.
[13] A. Magen, Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications, in: J.D.P. Rolim, S.P. Vadhan (Eds.), RANDOM, in: Lecture Notes in Computer Science, vol. 2483, Springer, 2002.
[14] S.A. Goreinov, E.E. Tyrtyshnikov, N.L. Zamarashkin, A theory of pseudoskeleton approximations, Linear Algebra Appl. 261 (1997) 1–21.
[15] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys. 155 (2) (1999) 468–498.