

FAST DIRECTIONAL MULTILEVEL ALGORITHMS FOR OSCILLATORY KERNELS*

BJÖRN ENGQUIST† AND LEXING YING†

Abstract. This paper introduces a new directional multilevel algorithm for solving N -body or N -point problems with highly oscillatory kernels. These systems often result from the boundary integral formulations of scattering problems and are difficult due to the oscillatory nature of the kernel and the non-uniformity of the particle distribution. 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. We then propose an efficient and accurate procedure which utilizes random sampling to generate such a separated, low rank representation. Based on the resulting representations, our new algorithm organizes the high frequency far field computation by a multidirectional and multiscale strategy to achieve maximum efficiency. The algorithm performs well on a large group of highly oscillatory kernels. Our algorithm is proved to have $O(N \log N)$ computational complexity for any given accuracy when the points are sampled from a two dimensional surface. We also provide numerical results to demonstrate these properties.

Key words. N -body problems, scattering problems, Helmholtz equation, oscillatory kernels, fast multipole methods, separated representations, random sampling, operator compression, multi-directional computation, multiscale methods

AMS subject classifications. 65C99, 65R99, 78A45

DOI. 10.1137/07068583X

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

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

where $G(x, y) = e^{2\pi i|x-y|}/|x-y|$ is the Green's function of the Helmholtz equation and is usually called the Helmholtz kernel. Throughout this paper, we use i to denote the complex number $\sqrt{-1}$. We have scaled the problem such that the wave length equals one 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 equation (PDE) formulations are transformed into boundary integral equation (BIE) formulations. The advantages of the BIE formulations are the convenience of handling the boundary conditions at infinity and the decrease in the dimensionality of the problem. The integral equation formulations are often discretized using appropriate quadrature methods. The resulting linear systems are always dense due to the fact that an integral formulation

*Received by the editors March 20, 2007; accepted for publication (in revised form) April 30, 2007; published electronically August 10, 2007.

<http://www.siam.org/journals/sisc/29-4/68583.html>

†Department of Mathematics, University of Texas at Austin, 1 University Station C1200, Austin, TX 78712 (engquist@math.utexas.edu, lexing@math.utexas.edu).

involves interaction between any two points or elements on the boundary. These dense linear systems are further solved using iterative methods, at each step of which the evaluation in (1) or closely related equations must be carried out.

Two observations are here in order. First, in 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 the value of K because the wavelength is equal to one in our setup. In many practical applications, K is usually equal to a few hundred or even thousand. 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 of order $O(K^2)$. Second, since the unknown field is supported only on the boundary in a BIE formulation, the points $\{p_i\}$ are samples of a two dimensional manifold. Consequently, the distribution of $\{p_i\}$ is highly nonuniform.

We would like to point out that, though the discussion in this paper mainly focuses on the Helmholtz kernel, our approach is quite general. Since many PDE kernels (e.g. the Green's functions of the time-harmonic Maxwell equations [21] and the linear elasticity equation [36]) are basically derivatives of the Helmholtz kernel, our algorithm can be readily extended to address these cases.

1.1. Previous work. Direct computation of (1) requires $O(N^2)$ operations, which can be intolerably slow for large values of N . During the last few decades, much attention has been devoted to the development of algorithms which evaluate (1) efficiently without compromising accuracy. An early class of algorithms (for example, [7, 8]) use the fast Fourier transform (FFT) by exploiting the fact that the kernel is translation invariant, and hence can be diagonalized through Fourier transform. Though quite efficient for uniformly distributed point sets, they require $O(N^{3/2} \log N)$ complexity in both computation time and storage space for a point set sampled from a two dimensional manifold. This poses a difficulty for large scattering problems. A recent improvement in [12] addresses this issue by using plane densities to reduce the sizes of the required FFTs.

The second class of algorithms (see for example [1, 3, 6, 10, 16]) discretize the boundary integral equations using the Galerkin approximation with either local Fourier bases or wavelet bases as the approximation space. The resulting discrete systems often become approximately sparse in the sense that a majority of the entries are close to zero with these bases. Such approaches have been shown to offer good theoretical estimates. However, constructing the remaining non-negligible entries both efficiently and accurately is non-trivial.

The third, and probably most popular, class of algorithms are the fast multipole methods (FMMs). The original FMM (see for example [17, 32, 34]) was developed for the kernel of the Laplace equation, and it evaluates the N -body problem in $O(N)$ operations for any fixed accuracy. Its success hinges on the observation that the interaction via the kernel between well-separated sets of points is approximately of low rank. Many other related developments, such as the panel clustering method [35], the H-matrix framework [9] and the interpolation techniques [11], are based on the same observation. In the low frequency regime where K is close to 1, this low rank property still holds for the Helmholtz kernel and the FMM for the Laplace equation can be applied to our problem with slight modifications (see [33]). In the high frequency regime where K is much larger than 1, the situation is however drastically different as the low rank assumption is not valid any more. In fact, the approximate rank of the interaction grows linearly with the size of the point sets (in terms of the wavelength). However, Rokhlin [42, 43] observed that the interaction between well-

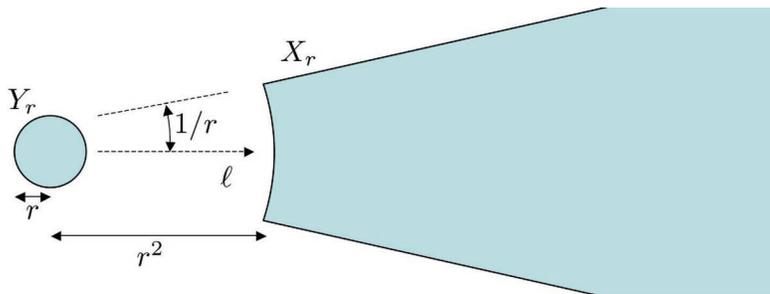


FIG. 1. Two sets Y_r and X_r which satisfy the directional parabolic separation condition.

separated point sets, though of large rank, can be applied efficiently by focusing on the far field signature of the potentials. In some sense, this is analogous to the fast Fourier transform. Even if the matrix is dense and of full rank, it can be applied efficiently in almost linear time. The resulting algorithm in the high frequency regime, often named HF-FMM, has $O(N \log N)$ complexity for a given accuracy. Though efficient and accurate, this algorithm is highly technical. It utilizes a large family of tools, such as partial wave expansion, far field signature, exponential expansion, filtering, and interpolation of spherical harmonics (see [37]). Many of these techniques have been developed only recently, and some of them solely for the purpose of the HF-FMM. Good descriptions of the HF-FMM include [18, 27, 41]. Other algorithms using similar techniques are given in [20, 23, 24, 45].

In [40], Michielssen and Boag proposed an interesting development called multi-level matrix decomposition. Their algorithm utilizes the idea of equivalent density, which also serves as the basis of the approaches in [2, 47], and it plays an important role in this paper. Even if their algorithm may not be as efficient as the HF-FMM, its three stage multiplication algorithm is ingenious and reminds one of the FFT.

1.2. Contribution and significance. In this paper, we propose a new directional multilevel algorithm that has $O(N \log N)$ complexity for a given accuracy. The starting point of our approach is a geometric consideration. Suppose Y_r is a ball of radius r centered at a point c , and X_r 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 Figure 1 for an illustration). Whenever two sets Y_r and X_r obey this geometric configuration, we say that Y_r and X_r 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_r and X_r via the Helmholtz kernel $G(x, y)$ is approximately of low rank for any fixed accuracy, and more importantly, the rank has an upper bound that is independent of r . To be more precise, the directional low rank property guarantees the following *directional separated representation* which is valid for any $x \in X_r$ and $y \in Y_r$:

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

where ε is the prescribed accuracy, $T(\varepsilon)$ is a constant which depends only on ε (not on r), and $\{\alpha_i(x)\}$ and $\{\beta_i(y)\}$ are functions of x and y respectively.

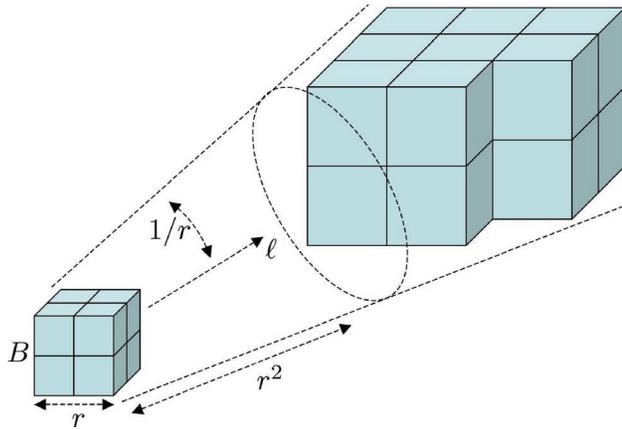


FIG. 2. For each box B , its far field is partitioned into multiple wedges. Our algorithm constructs one low rank separated representation for each wedge. Each resulting representation can be used to accelerate the interaction computation between B and all the boxes in that wedge.

The second contribution of this paper is to introduce a randomized procedure for generating the directional separated representation $\{\alpha_i(x)\}$ and $\{\beta_i(x)\}$ for the interaction between X_r and Y_r . This procedure only uses kernel evaluations and stable numerical routines such as the singular value decomposition (SVD) and the QR factorization. The resulting representation allows highly efficient computation of the interaction between sets like X_r and Y_r , and moreover, it can be stored in an economic way.

Our algorithm starts by partitioning the domain recursively using an octree, which is similar to the standard FMM. 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 [47], which is a variant of the standard FMM. In the high frequency region, the computation is organized in a multidirectional way. For a given box B , all the boxes well separated from it are partitioned into a group of wedges, where each wedge and the box itself follow the directional parabolic separation condition (see Figure 2). The randomized procedure mentioned above is then used repetitively to construct multiple low rank representations about B , one for each wedge. As a result, the calculation of the interactions between B and all 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 in the high frequency regime to achieve maximum efficiency.

Our directional multilevel algorithm has the following properties. First, it is highly efficient and accurate. A careful complexity analysis shows that, for a finite point set on a two dimensional manifold, our algorithm uses $O(N \log N)$ operations, which is the same as the complexity of the HF-FMM. Second, our algorithm is less technical than the HF-FMM as it uses no analytic expansions and translations. The computational steps of our algorithm only involve kernel evaluation and stable linear algebra routines. Finally, our algorithm works for more general oscillatory kernels as well (see the numerical results in section 5).

This paper is organized as follows. In section 2, we prove the directional low rank property. Following that, we describe the randomized procedure for generating

a directional low rank separated representations in section 3. Our main algorithm is presented in detail in section 4 where also its computational complexity is analyzed. After reporting the numerical results in section 5, we give conclusions and suggestions for future work in section 6.

2. Directional low rank property. In this section, we prove the main theoretical result of this paper: the directional low rank property. Suppose $r \geq \sqrt{3}$ and let

$$(2) \quad Y_r = B(0, r) \quad \text{and} \quad X_r = \{x : \theta(x, \ell) \leq 1/r, |x| \geq r^2\}$$

where ℓ is a given unit vector and $\theta(a, b)$ is the spanning angle between vectors a and b . The geometric relationship between Y_r and X_r is illustrated in Figure 1.

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 $\{\alpha_i(x), 1 \leq i \leq T\}$ and $\{\beta_i(y), 1 \leq i \leq T\}$ such that

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

for all $x \in X$ and $y \in Y$.

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. For any $\varepsilon > 0$, there exists a number $T(\varepsilon)$ which is independent of r such that $e^{2\pi i|x-y|}/|x-y|$ has a $T(\varepsilon)$ -term ε -expansion for any X_r and Y_r satisfying (2).

Results similar to Theorem 2.2 have been discovered before by researchers from applied physics and electric engineering [13, 29, 40]. Their proofs involve special functions such as spherical harmonics and Bessel functions, and give much sharper estimates on the number of terms in the expansion. We believe that our proof in the rest of this section is still of theoretical interest since one can easily adapt it to show similar results for other oscillatory kernels such as $e^{2\pi i|x-y|}$. The numerical results in section 5 validate this assertion.

Our proof of Theorem 2.2 is based on the following lemmas.

LEMMA 2.3. For any $\varepsilon > 0$, there exists a number $T(\varepsilon)$ which is independent of r such that $|x-y|$ has a $T(\varepsilon)$ -term ε -expansion for X_r and Y_r .

Proof. Let us write

$$|x-y| = |x| \cdot f(y/|x|)$$

where $f(p) = |\hat{x}-p|$ and \hat{x} is the unit direction $x/|x|$. Here we suppress the dependence of f on \hat{x} . Taylor expansion of f at $p=0$ gives

$$f(p) = \sum_{d=0}^{\infty} \left[\sum_{|\alpha|=d} \frac{f^{(\alpha)}(0)}{\alpha!} p^\alpha \right]$$

where α is a multiindex. For any $d \geq 0$, we use $f_d(p)$ to denote the quantity in the square bracket. The series $\sum_{d=0}^{\infty} f_d(p)$ is a convergent series with radius 1. This implies that for any η with $\eta \geq 1$ there exists an integer $D_1(\eta)$ such that $f_d(p) \leq (\eta|p|)^d$ for every $d \geq D_1(\eta)$. We assume, without loss of generality, that $\eta \leq 1.1$.

We argue that there exists a number $D(\varepsilon)$ such that

$$|x| \cdot f_{D(\varepsilon)}(y/|x|) \leq \varepsilon.$$

To show this, it is sufficient to find an integer d such that $|x| \cdot (\eta \cdot |y|/|x|)^d \leq \varepsilon$. Notice that for $d \geq 2$, $|x| \cdot (\eta \cdot |y|/|x|)^d$ is a decreasing function of $|x|$. Therefore, the function reaches its maximum value when $|x| = r^2$. Since $|y| \leq r$,

$$|x| \cdot \left(\eta \cdot \frac{|y|}{|x|} \right)^d \leq r^2 \cdot \left(\frac{\eta}{r} \right)^d = \frac{\eta^d}{r^{d-2}} \leq \frac{\eta^d}{\sqrt{3}^{d-2}}.$$

Since we have assumed that $\eta \leq 1.1$, it is obvious that there exists a $D^2(\eta, \varepsilon)$ such that for any $d \geq D^2(\eta, \varepsilon)$ we have $\eta^d/\sqrt{3}^{d-2} \leq \varepsilon$. Therefore, setting $D(\varepsilon) = \max(2, D^1(\eta), D^2(\eta, \varepsilon))$ guarantees that

$$|x| \cdot \left(\eta \cdot \frac{|y|}{|x|} \right)^{D(\varepsilon)} \leq \varepsilon.$$

Since the series $\{|x| \cdot (\eta \cdot |y|/|x|)^d : d \geq 0\}$ is geometric and $f_d(y/|x|) \leq (\eta \cdot |y|/|x|)^d$, we have the following estimate

$$\sum_{d > D(\varepsilon)} |x| \cdot f_d(y/|x|) \leq 4\varepsilon,$$

which is equivalent to

$$(3) \quad \left| |x - y| - \sum_{d=0}^{D(\varepsilon)} |x| \cdot f_d(y/|x|) \right| \leq 4\varepsilon.$$

Notice that the number of terms $D(\varepsilon)$ is independent of $|x|$. In fact, it is independent of \hat{x} as well since the kernel is rotation invariant.

Now, each term $f_d(y/|x|)$ can be written as

$$(4) \quad f_d(y/|x|) = \sum_{|\alpha|=d} \frac{f^{(\alpha)}(0)}{\alpha!} \cdot \frac{1}{|x|^{|\alpha|}} \cdot y^\alpha$$

where $f^{(\alpha)}(0)$ depends on \hat{x} implicitly. Since (4) is a separated representation,

$$(5) \quad \sum_{d > D(\varepsilon)} |x| \cdot f_d(y/|x|)$$

has a separated representation as well. Now suppose $T(\varepsilon)$ to be the total number of terms after the expansion of (5). Then (3) states that $|x - y|$ has a $T(\varepsilon)$ -term 4ε -expansion. \square

Similarly, we have the following lemma for the function $1/|x - y|$. In fact, the proof is simpler than the one of Lemma 2.3 because $1/|x - y|$ is bounded for $x \in X_r$ and $y \in Y_r$ for $r \geq \sqrt{3}$.

LEMMA 2.4. *For any $\varepsilon > 0$, there exists a number $T(\varepsilon)$ which is independent of r such that $1/|x - y|$ has a $T(\varepsilon)$ -term ε -expansion for X_r and Y_r .*

COROLLARY 2.5. *For any $\varepsilon > 0$, there exists a number $T(\varepsilon)$ which is independent of r such that $|x - y| - \hat{x} \cdot (x - y)$ has a $T(\varepsilon)$ -term ε -expansion for X_r and Y_r .*

Proof. Since $\hat{x} \cdot (x - y)$ has a 2-term 0-expansion

$$\hat{x} \cdot (x - y) = |x| - \hat{x} \cdot y,$$

the corollary follows from Lemma 2.3. \square

The following lemmas show that certain functions are bounded by constants.

LEMMA 2.6. *There exists a constant G such that*

$$|(\hat{x} - \ell) \cdot y| \leq G$$

and

$$\| |x - y| - \hat{x} \cdot (x - y) \| \leq G$$

for $x \in X_r$ and $y \in Y_r$.

Proof. For the first part, we observe

$$|(\hat{x} - \ell) \cdot y| \leq |(\hat{x} - \ell)| \cdot |y| \leq \theta(x, \ell) \cdot |y|.$$

From the definitions of X_r and Y_r , we have $|y| \leq r$ and $\theta(x, \ell) \leq 1/r$. Therefore,

$$|(\hat{x} - \ell) \cdot y| \leq r \cdot \frac{1}{r} = 1.$$

For the second part, suppose that θ is the spanning angle between x and $x - y$. From the definitions of X_r and Y_r we have $\theta \leq c_1 \cdot r / |x - y|$, $|x - y| \geq c_2 \cdot r^2$, $1 - \cos(\theta) \leq c_3 \cdot \theta^2$ for constants c_1, c_2 and c_3 . Therefore,

$$|x - y| - \hat{x} \cdot (x - y) = |x - y| \cdot (1 - \cos(\theta)) \leq c_3 \cdot |x - y| \cdot \theta^2 \leq c_3 c_1^2 \cdot \frac{r^2}{|x - y|} \leq \frac{c_3 c_1^2}{c_2}.$$

Setting $G = \max(1, c_3 c_1^2 / c_2)$ completes the proof. \square

The next lemma, which is quite elementary, is concerned with the accuracy of the Taylor expansion of the exponential function. We reproduce the proof for completeness (see [14]).

LEMMA 2.7. *For any $Z > 0$ and $\varepsilon > 0$, let $N = \lceil \max(4e\pi Z, \log_2(1/\varepsilon)) \rceil$. Then*

$$\left| e^{2\pi i x} - \sum_{k=0}^{N-1} \frac{(2\pi i x)^k}{k!} \right| \leq \varepsilon$$

for any x with $|x| \leq Z$.

Proof. Truncated Taylor expansion of $e^{2\pi i x}$ at the origin gives

$$e^{2\pi i x} = \sum_{k=0}^{N-1} \frac{(2\pi i x)^k}{k!} + \frac{(2\pi i \tilde{x})^N}{N!}$$

where \tilde{x} lies between 0 and x . Since $|\tilde{x}| \leq |x| \leq Z$,

$$\left| e^{2\pi i x} - \sum_{k=0}^{N-1} \frac{(2\pi i x)^k}{k!} \right| \leq \frac{(2\pi Z)^N}{N!}.$$

We use the Stirling formula to estimate the last quantity. Since $N! \geq (N/e)^N$,

$$\frac{(2\pi Z)^N}{N!} \leq \left(\frac{2\pi eZ}{N}\right)^N.$$

Now, as $N \geq \max(4e\pi Z, \log_2(1/\varepsilon))$,

$$\left(\frac{2\pi eZ}{N}\right)^N \leq \left(\frac{1}{2}\right)^{\log_2(1/\varepsilon)} = \varepsilon. \quad \square$$

LEMMA 2.8. *Let $\varepsilon > 0$ and $\eta > 0$. Suppose that a function $f(x, y)$ has a $T^f(\varepsilon)$ -term ε -expansion on X_r and Y_r , and $|f(x, y)| \leq B^f$. Suppose also that a function $g(x, y)$ has a $T^g(\eta)$ -term η -expansion on X_r and Y_r , and $|g(x, y)| \leq B^g$. Then, $f(x, y) \cdot g(x, y)$ has a $T^f(\varepsilon) \cdot T^g(\eta)$ -term $(B^g\varepsilon + B^f\eta + \varepsilon\eta)$ -expansion.*

Proof. By assumption, there exist functions $\{\alpha_i^f(x)\}$ and $\{\beta_i^f(x)\}$ for $f(x, y)$ such that

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

Similarly, there exist functions $\{\alpha_i^g(x)\}$ and $\{\beta_i^g(x)\}$ for $g(x, y)$ such that

$$\left| g(x, y) - \sum_{i=1}^{T^g(\eta)} \alpha_i^g(x)\beta_i^g(y) \right| \leq \eta.$$

Consider two sets of functions $\{\alpha_i^f(x)\alpha_j^g(x) : 1 \leq i \leq T^f(\varepsilon), 1 \leq j \leq T^g(\eta)\}$ and $\{\beta_i^f(y)\beta_j^g(y) : 1 \leq i \leq T^f(\varepsilon), 1 \leq j \leq T^g(\eta)\}$, each containing $T^f(\varepsilon) \cdot T^g(\eta)$ terms.

$$\begin{aligned} & \left| f(x, y)g(x, y) - \sum_{i,j} (\alpha_i^f(x)\alpha_j^g(x))(\beta_i^f(y)\beta_j^g(y)) \right| \\ &= \left| f(x, y)g(x, y) - \left(\sum_i \alpha_i^f(x)\beta_i^f(y)\right) \left(\sum_j \alpha_j^g(x)\beta_j^g(y)\right) \right| \\ &\leq \left| \left(f(x, y) - \left(\sum_i \alpha_i^f(x)\beta_i^f(y)\right)\right) g(x, y) \right| + \\ & \quad \left| \left(\sum_i \alpha_i^f(x)\beta_i^f(y)\right) \left(g(x, y) - \left(\sum_j \alpha_j^g(x)\beta_j^g(y)\right)\right) \right| \\ &\leq \varepsilon \cdot B^g + (B^f + \varepsilon) \cdot \eta \\ &= (B^g\varepsilon + B^f\eta + \varepsilon\eta). \quad \square \end{aligned}$$

We are now ready to prove Theorem 2.2.

Proof of Theorem 2.2. Without loss of generality, we make the assumption that $\varepsilon < 1$.

First, let us construct an expansion for $e^{2\pi i(|x-y|-\hat{x}\cdot(x-y))}$. From Corollary 2.5, $|x - y| - \hat{x} \cdot (x - y)$ has a $T^1(\varepsilon)$ -term ε -expansion, i.e., there exist functions $\{\alpha_i^1(x)\}$

and $\{\beta_i^1(y)\}$ such that

$$\left| (|x - y| - \hat{x} \cdot (x - y)) - \sum_i \alpha_i^1(x) \beta_i^1(y) \right| \leq C^1 \varepsilon$$

Lemma 2.6 says that there exists a constant G such that

$$||x - y| - \hat{x} \cdot (x - y)| \leq G,$$

which implies

$$\left| \sum_i \alpha_i^1(x) \beta_i^1(y) \right| \leq G + \varepsilon.$$

Applying Lemma 2.7 with $Z = G + \varepsilon$ and $N^1(\varepsilon) = \lceil \max(4e\pi(G + \varepsilon), \log_2(1/\varepsilon)) \rceil$ gives

$$\left| e^{2\pi i \sum_i \alpha_i^1(x) \beta_i^1(y)} - \sum_{k=0}^{N^1(\varepsilon)-1} \frac{(2\pi i)^k}{k!} \left(\sum_i \alpha_i^1(x) \beta_i^1(y) \right)^k \right| \leq \varepsilon.$$

Expanding the $(\sum_i \alpha_i^1(x) \beta_i^1(y))^k$ terms and absorbing the coefficients $\frac{(2\pi i)^k}{k!}$, we find that there exists a number $T^2(\varepsilon)$ and two sets of functions $\{\alpha_i^2(x)\}$ and $\{\beta_i^2(y)\}$ such that

$$\left| e^{2\pi i \sum_i \alpha_i^1(x) \beta_i^1(y)} - \sum_{i=0}^{T^2(\varepsilon)} \alpha_i^2(x) \beta_i^2(y) \right| \leq \varepsilon,$$

or equivalently, that $e^{2\pi i \sum_i \alpha_i^1(x) \beta_i^1(y)}$ has a $T^2(\varepsilon)$ -term ε -expansion. Since

$$\begin{aligned} \left| e^{2\pi i (|x-y| - \hat{x} \cdot (x-y))} - e^{2\pi i \sum_i \alpha_i^1(x) \beta_i^1(y)} \right| &\leq 2\pi \left| (|x - y| - \hat{x} \cdot (x - y)) - \sum_i \alpha_i^1(x) \beta_i^1(y) \right| \\ &\leq 2\pi \varepsilon, \end{aligned}$$

it is obvious that $e^{2\pi i (|x-y| - \hat{x} \cdot (x-y))}$ has a $T^2(\varepsilon)$ -term $(2\pi + 1)\varepsilon$ -expansion

$$(6) \quad \left| e^{2\pi i (|x-y| - \hat{x} \cdot (x-y))} - \sum_{i=0}^{T^2(\varepsilon)} \alpha_i^2(x) \beta_i^2(y) \right| \leq (2\pi + 1)\varepsilon.$$

Second, we consider the term $e^{2\pi i \hat{x} \cdot (x-y)}$. From Lemma 2.6, we know that there exists a constant G such that $|(\hat{x} - \ell) \cdot y| \leq G$. Now applying Lemma 2.7 again with $Z = G$ and $N^3(\varepsilon) = \lceil \max(4e\pi G, \log_2(1/\varepsilon)) \rceil$ gives

$$\left| e^{2\pi i (\hat{x} - \ell) \cdot y} - \sum_{k=0}^{N^3(\varepsilon)-1} \frac{(2\pi i)^k}{k!} ((\hat{x} - \ell) \cdot y)^k \right| \leq \varepsilon.$$

Expanding the $((\hat{x} - \ell) \cdot y)^k$ terms for all k and absorbing the coefficients $\frac{(2\pi i)^k}{k!}$, we have the expansion

$$\left| e^{2\pi i (\hat{x} - \ell) \cdot y} - \sum_{i=0}^{T^3(\varepsilon)} \alpha_i^3(x) \beta_i^3(y) \right| \leq \varepsilon,$$

where $T^3(\varepsilon)$ is an ε -dependent integer and $\{\alpha_i^3(x)\}$ and $\{\beta_i^3(y)\}$ are two sets of functions. That is to say, $e^{2\pi i(\hat{x}-\ell)\cdot y}$ has a $T^3(\varepsilon)$ -term ε -expansion. Now exponentiating the identity

$$\hat{x} \cdot (x - y) = |x| - (\hat{x} - \ell) \cdot y - \ell \cdot y$$

gives

$$e^{2\pi i\hat{x}\cdot(x-y)} = e^{2\pi i|x|} \cdot e^{-2\pi i(\hat{x}-\ell)\cdot y} \cdot e^{-2\pi i\ell\cdot y}.$$

Then it is obvious that $e^{2\pi i\hat{x}\cdot(x-y)}$ has a $T^4(\varepsilon)$ -term ε -expansion

$$(7) \quad \left| e^{2\pi i\hat{x}\cdot(x-y)} - \sum_{i=0}^{T^4(\varepsilon)} \alpha_i^4(x)\beta_i^4(y) \right| \leq \varepsilon,$$

with $T^4(\varepsilon) = T^3(\varepsilon)$ and the functions $\{\alpha_i^4(x)\}$ and $\{\beta_i^4(y)\}$ given by

$$\alpha_i^4(x) = e^{2\pi i|x|} \cdot \overline{\alpha_i^3(x)} \quad \text{and} \quad \beta_i^4(y) = e^{-2\pi i\ell\cdot y} \cdot \overline{\beta_i^3(y)}.$$

Next, Lemma 2.4 claims that $1/|x - y|$ has $T^5(\varepsilon)$ -term ε expansion

$$(8) \quad \left| \frac{1}{|x - y|} - \sum_{i=0}^{T^5(\varepsilon)} \alpha_i^5(x)\beta_i^5(y) \right| \leq \varepsilon$$

with functions $\{\alpha_i^5(x)\}$ and $\{\beta_i^5(y)\}$.

Finally, since

$$\frac{e^{2\pi i|x-y|}}{|x - y|} = e^{2\pi i(|x-y|-\hat{x}\cdot(x-y))} \cdot e^{2\pi i\hat{x}\cdot(x-y)} \cdot \frac{1}{|x - y|},$$

applying Lemma 2.8 twice to the product of (6), (7) and (8) shows that $\frac{e^{2\pi i|x-y|}}{|x-y|}$ has a $(T^2(\varepsilon) \cdot T^4(\varepsilon) \cdot T^5(\varepsilon))$ -term $(12\pi + 10)\varepsilon$ -expansion. Here we use the facts that

$$\left| e^{2\pi i(|x-y|-\hat{x}\cdot(x-y))} \right| < 1, \quad \left| e^{2\pi i\hat{x}\cdot(x-y)} \right| < 1 \quad \text{and} \quad \left| \frac{1}{|x - y|} \right| < 2$$

for $x \in X_r$ and $y \in Y_r$ along with the assumption that $\varepsilon < 1$. □

The message of Theorem 2.2 is that, for a fixed ε , the number of terms in an ε -expansion is independent of r , as long as X_r and Y_r satisfy the directional parabolic separation condition, i.e., X_r belongs to a cone with spanning angle $1/r$ and it is an order $O(r^2)$ distance away from Y_r .

The proof given above is not optimal in the sense that $(T^2(\varepsilon) \cdot T^4(\varepsilon) \cdot T^5(\varepsilon))$, the bound of the number of terms in the expansion, can be fairly large. In practice, numerical results show that the actual number of terms grows linearly with $\log(1/\varepsilon)$, see section 5.1.

So far in Theorem 2.2, we have assumed that Y_r is centered at the origin. However, as we have noticed in the proof, what is really important is the relative positions of the sets X_r and Y_r . Since the kernel $e^{2\pi i|x-y|}/|x - y|$ is translation invariant, shifting X_r and Y_r together by a constant vector would not change the result of Theorem 2.2. We have also assumed ℓ to be a fixed direction and suppressed the dependence of

X_r on ℓ in our notations. However, all the estimates are independent of the specific choice of ℓ as the kernel $e^{2\pi i|x-y|}/|x-y|$ is rotation invariant. Suppose we define $F_r = \{x : |x| \geq r^2\}$. There exists a set of unit vectors $\{\ell\}$ of size $O(r^2)$ such that the union of the cones centered at $\{\ell \in L\}$ with spanning angle $1/r$ cover the whole space. Now let us define

$$X_r^\ell = \{x : \theta(x, \ell) \leq 1/r, |x| \geq r^2\}$$

for each ℓ . Clearly their union is equal to F_r . Applying Theorem 2.2 to each X_r^ℓ results in a different $T(\varepsilon)$ -term ε -expansion for Y_r and X_r^ℓ . Even though the expansions are different from one X_r^ℓ to another, the bound $T(\varepsilon)$ on the number of terms is independent of ℓ .

3. Randomized construction of separated representation. The expansion for two sets X_r^ℓ and Y_r introduced in section 2 is a directional separated representation. It is directional since for a given direction ℓ the expansion is valid only for X_r^ℓ that belongs to a directional cone. It is separated since the two sets of functions $\{\alpha_i(x)\}$ and $\{\beta_i(y)\}$ depend only on x and y respectively. The proof of Theorem 2.2 is constructive in the sense that it provides a way to compute the functions $\{\alpha_i(x)\}$ and $\{\beta_i(y)\}$. However, the number of terms in the resulting expansion can be fairly large. In this section, we introduce a procedure which in practice gives expansions with very few terms. This procedure is accurate, efficient and based on random sampling.

In the rest of this section, r and ℓ are not explicitly included in the notation. Our randomized procedure consists of the following steps:

1. Sample Y randomly and densely to obtain a set of samples $\{y_i\}$. By densely, we mean a couple of samples per wavelength. Similarly, sample X to obtain a set of samples $\{x_i\}$. Suppose the numbers of samples in each set are respectively N_y and N_x . Since our problem is confined to a ball of radius $K/2$ where K is the number of wavelengths, there is no reason to sample the part of X that is outside of this ball. As we allow a constant number of samples per wavelength, $N_y = O(\text{vol}(Y)) = O(K^{1.5})$. Similarly, $N_x = O(\text{vol}(X \cap B(0, K))) = O(K^2)$. Let A be the N_x by N_y matrix defined by

$$A_{ij} = e^{2\pi i|x_i - y_j|}/|x_i - y_j|$$

for $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$. In the language of linear algebra, Theorem 2.2 states that A can be factorized, within error $O(\varepsilon)$, into the product of two matrices, the first of size N_x by $T(\varepsilon)$ and the second of size $T(\varepsilon)$ by N_y . In the next few steps, we construct such a factorization.

2. Let A_1 be the submatrix of A containing a set of N_1 randomly selected columns. Here N_1 is set to be a constant multiple of $T(\varepsilon)$. After obtaining the pivoted QR factorization of A_1 , we have the decomposition

$$A_1 P_1 = Q_1 R_1$$

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_c and R_c . Since A_1 , as a submatrix of A , has an approximate factorization of $T(\varepsilon)$, in practice the matrix Q_c has only $O(T(\varepsilon))$ columns after truncation. It is clear from the algorithm of the pivoted QR factorization

that the new matrices Q_c and R_c satisfy the relationship

$$Q_c R_c = A_c$$

where A_c is the submatrix containing the columns of A from which the matrix Q_c is generated. The samples (of Y) associated with these columns are denoted $\{y_i^c\}$. In practice, the columns of Q_c approximately span the column space of A (not just A_1), producing an $O(\varepsilon)$ error with overwhelming probability.

3. Let A_2 be a submatrix of A containing a set of N_2 randomly selected rows. Here N_2 is again set to be a constant multiple of $T(\varepsilon)$. Repeat the previous step on A_2^* . As a result, we have two matrices Q_r and R_r . Q_r is orthonormal and has $O(T(\varepsilon))$ columns again, while R_r is upper triangular. They satisfy the relationship

$$R_r^* Q_r^* = A_r$$

where A_r is a submatrix containing appropriate rows of A . We denote the samples (of X) associated with these rows by $\{x_i^r\}$. Similar to the previous step, the rows of Q_r^* approximately span the row space of A (not just A_2), producing an $O(\varepsilon)$ error with overwhelming probability.

4. Since Q_c and Q_r^* span, respectively, the column and the row spaces of A with error $O(\varepsilon)$, the matrix $M = Q_c^* A Q_r$ satisfies the relationship

$$(9) \quad |A - Q_c M Q_r^*| = O(\varepsilon).$$

Notice that $Q_c M$ and Q_r^* already provide us with a factorization of the matrix A with error $O(\varepsilon)$. However, since A is huge, computing $M = Q_c^* A Q_r$ can be very costly. To this end, we propose the following alternative. We randomly pick a set S of N_s rows and a set T of N_t columns. Set A_3 to be the minor containing the elements from rows in S and columns in T , $Q_{c,S}$ to be the submatrix of Q_c containing the rows in S , and $Q_{r,T}^*$ to be the submatrix of Q_r^* containing the columns in T . Now the constraint (9) restricted to S and T becomes

$$|A_3 - Q_{c,S} M Q_{r,T}^*| = O(\varepsilon).$$

To satisfy this constraint, we choose

$$(10) \quad M = (Q_{c,S})^+ A (Q_{r,T}^*)^+$$

where $()^+$ stands for pseudo-inverse. In practice, we pick N_s and N_t to be about ten times $T(\varepsilon)$.

5. We claim that

$$|A - Q_c M Q_r^*| = O(\varepsilon).$$

Moreover, since $Q_c R_c = A_c$ and $R_r^* Q_r^* = A_r$, we have

$$|A - A_c \cdot (R_c)^+ \cdot M \cdot (R_r^*)^+ \cdot A_r| = O(\varepsilon).$$

Let us define D to be the matrix $(R_c)^+ \cdot M \cdot (R_r^*)^+$ sandwiched between A_c and A_r . We then have the approximation

$$(11) \quad |A - A_c \cdot D \cdot (R_r^*)^+ \cdot A_r| = O(\varepsilon).$$

(11) is often called a pseudoskeleton approximation of A in the literature (see [30, 31]). Notice that the matrix D has only $O(T(\varepsilon))$ rows and columns. Denoting the entries of D by d_{pq} , we can rewrite the previous statement in the form

$$\left| \frac{e^{2\pi i|x_i - y_j|}}{|x_i - y_j|} - \sum_{p,q} \frac{e^{2\pi i|x_i - y_p^c|}}{|x_i - y_p^c|} \cdot d_{pq} \cdot \frac{e^{2\pi i|x_q^c - y_j|}}{|x_q^c - y_j|} \right| = O(\varepsilon)$$

for all x_i and y_j .

6. 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

$$(12) \quad \left| \frac{e^{2\pi i|x - y|}}{|x - y|} - \sum_{p,q} \frac{e^{2\pi i|x - y_p^c|}}{|x - y_p^c|} \cdot d_{pq} \cdot \frac{e^{2\pi i|x_q^c - y|}}{|x_q^c - y|} \right| = O(\varepsilon)$$

for any $x \in X \cap B(0, K)$ and $y \in Y$.

In (12), both $e^{2\pi i|x - y_p^c|}/|x - y_p^c|$ and $e^{2\pi i|x_q^c - y|}/|x_q^c - y|$ can be computed directly from the kernel formulae. Therefore, the only quantities that must be stored for (12) are the locations $\{x_q^c\}$ and $\{y_p^c\}$ and the matrix D . This costs only $O(1)$ in storage for a fixed error threshold ε . In fact, for the Helmholtz kernel, one only needs to sample the boundary of the domains X and Y densely in the first step. This improvement, which is due to the uniqueness and existence properties of the Dirichlet boundary value problem of the Helmholtz equation, significantly reduces the computational complexity of our randomized procedure.

The randomized procedure presented here benefits from the existing approaches for constructing low rank separated approximations. Kapur and Long [38] were among the first to use the idea of random sampling to construct low rank representation for electrostatic interaction. The adaptive cross approximation (ACA) by Bebendorf and Rjasanow [4, 5] constructs the approximation in an incremental way. Recently, it has been combined with the H-matrix framework [9] to address several problems in BIE formulations, including the scattering problems in the low frequency regime [22]. The idea of random sampling has been carefully studied in a series of papers by Drineas, Kannan, and Mahoney [25, 26] for general matrices, and our approach shares some similarities with the LinearTimeCUR algorithm [26] proposed by these authors. In [14], a similar algorithm is used to speed up the numerical computation of the Fourier integral operator. Finally, we would like to mention a recent algorithm proposed by Martinsson, Rokhlin, and Tygert [39] which exploits the spectral properties of the Gaussian random matrices and exhibits excellent numerical properties for matrices equipped with fast multiplication algorithms.

Our randomized procedure works well numerically. In section 5, we see that this procedure constructs low rank factorizations in a stable way for ε as small as 10^{-8} . The rank of the resulting factorization, i.e., the minimum of the cardinalities of the sets $\{y_p^c\}$ and $\{x_q^c\}$, is very close to the optimal rank obtained by factorizing A using an SVD.

In most of the cases, we do know the separation rank of A . Since the value of $T(\varepsilon)$ from Theorem 2.2 can be significantly larger than the true separation rank. Setting N_1 and N_2 to be a constant multiple of $T(\varepsilon)$ could be quite wasteful. In our implementation, we adopt the method proposed in [14]. The idea is to start with a relatively small value for N_1 . If the columns of Q_c constructed from the second step fail to span the column space of A , we then double N_1 and repeat the second step

until the columns of Q_c span the column space of A within the prescribed error ε . The same strategy is applied to N_2 in the construction of Q_r^* . The time spent on the unsuccessful attempts is bounded by that of the final successful attempt due to the geometric growth rate.

Though we are not able to provide a rigorous proof for the accuracy of this randomized procedure, we argue that its success is based on the following observations:

- For fixed y , the kernel $e^{2\pi i|x-y|}/|x-y|$ is a wave-like function of x . The same is true if we fix x and regard $e^{2\pi i|x-y|}/|x-y|$ as a function of y . In fact, as verified by numerical computation, the row and column bases of the matrix A contain oscillatory vectors, each of which consists of sufficiently dense samples of an oscillatory function in R^3 with wavelength close to 1. In this regard, one may assume these oscillatory vectors to be the Fourier modes around the frequency 2π . Now, if the row basis contains the Fourier modes, the column vectors of A are incoherent mixtures of the vectors of the column basis. This incoherence property comes from the uncertainty principle between the Fourier and the (canonical) delta bases [15, 44]. As a result, sampling the columns of A with a moderate oversampling factor ensures that the selected vectors contain significant contributions from all of the vectors of the column basis. This is why we are able to construct the column basis of A in a stable way by observing only $N_1 = O(T(\varepsilon))$ of its vectors.
- As the vectors of Q_r are similar to the Fourier basis, the same incoherence property [15, 44] suggests that, as long as the number of rows in S is significantly larger than the number of columns of Q_r , the matrix $Q_{r,S}$ is not too far from an orthogonal matrix and thus has a small condition number. This allows us to invert $Q_{r,S}$ robustly and the same is true for $Q_{r,T}^*$. This allows us to recover M from (10) in a stable way.

In order to prepare for the discussion of our main algorithm in section 4, it is useful to introduce the following definitions. Suppose we have a set of densities $\{f_i\}$ located at points $\{b_i\}$ in Y . Using the representation obtained from the randomized procedure, we have

$$\left| \sum_i \frac{e^{2\pi i|x-b_i|}}{|x-b_i|} f_i - \sum_p \frac{e^{2\pi i|x-y_p^c|}}{|x-y_p^c|} \sum_q d_{pq} \sum_i \frac{e^{2\pi i|x_q^c-b_i|}}{|x_q^c-b_i|} f_i \right| = O(\varepsilon).$$

This states that we can place a set of densities

$$(13) \quad \left\{ \sum_q d_{pq} \sum_i \frac{e^{2\pi i|x_q^c-b_i|}}{|x_q^c-b_i|} f_i \right\}$$

at points $\{y_p^c\}$ in order to reproduce the potential generated by the densities $\{f_i\}$ located at points $\{b_i\}$. To this end, the densities in (13) are called the *directional outgoing equivalent densities* of Y in direction ℓ , and the points $\{y_p^c\}$ are called the *directional outgoing equivalent points* of Y in direction ℓ . Here the word outgoing refers to the role of Y in the computation. In addition, we refer to the quantities

$$(14) \quad \left\{ \sum_i \frac{e^{2\pi i|x_q^c-b_i|}}{|x_q^c-b_i|} f_i \right\}$$

as the *directional outgoing check potentials* of Y in direction ℓ and the points $\{x_q^c\}$ as the *directional outgoing check points* of Y in direction ℓ .

Notice that in (14) the directional outgoing check potentials can be computed using only kernel evaluation. In addition, as long as one has precomputed the matrix $D = (d_{pq})$, the directional outgoing equivalent densities can be evaluated using a single matrix vector multiplication (i.e., (13)).

Now, let us reverse the situation. Suppose we have a set of densities $\{f_i\}$ located at points $\{a_i\}$ in X so that

$$\left| \sum_i \frac{e^{2\pi i|y-a_i|}}{|y-a_i|} f_i - \sum_q \frac{e^{2\pi i|y-x_q^c|}}{|y-x_q^c|} \sum_p d_{pq} \sum_i \frac{e^{2\pi i|y_p^c-a_i|}}{|y_p^c-a_i|} f_i \right| = O(\varepsilon).$$

This states that we can put a set of densities

$$(15) \quad \left\{ \sum_p d_{pq} \sum_i \frac{e^{2\pi i|y_p^c-a_i|}}{|y_p^c-a_i|} f_i \right\}$$

at points $\{x_q^c\}$ and they reproduce the potential generated by the densities $\{f_i\}$ located at points $\{a_i\}$. Therefore, we call the densities in (15) the *directional incoming equivalent densities* of Y in direction ℓ and the locations $\{x_q^c\}$ the *directional incoming equivalent points* of Y in direction ℓ . In analogy to the previous terminology,

$$(16) \quad \left\{ \sum_i \frac{e^{2\pi i|y_p^c-a_i|}}{|y_p^c-a_i|} f_i \right\}$$

are called the *directional incoming check potentials* of Y in direction ℓ and the location $\{y_p^c\}$ are called the *directional incoming check points* of Y in direction ℓ .

A couple of remarks are in order here. Let us fix the direction ℓ . Since the kernel $e^{2\pi i|x-y|}/|x-y|$ is translation invariant, the equivalent points and check points for a set centered at an arbitrary point can be obtained from those of the set centered at the origin by translation. However, the D matrix remains the same.

The kernel is also rotation invariant. For a fixed radius r , we only need to generate the equivalent points and check points for a fixed direction ℓ . These quantities for any other direction can be obtained by rotation.

4. Algorithm description. Based on the results in the previous section, we are ready to describe our new algorithm for the N -body problem with oscillatory kernels, i.e., the computation of

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

for all $1 \leq i \leq N$. Without loss of generality, we assume that $K = 2^{2L}$ for a positive integer L .

4.1. Data structure. Similar to the HF-FMM, our main data structure is an octree. The top level box of width K contains all the points $\{p_i\}$. In the rest of this paper, B denotes a box in the octree and w for 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 non-empty 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 [47]. We modify the notation slightly to accommodate the discussion of the current algorithm.

- The near field N^B is the union of the boxes A that satisfies $dist(A, B) = 0$, where $dist(A, B) = \inf_{x \in A, y \in B} |x - y|$.
- The far field F^B is the complement of N^B .
- The interaction list I^B contains all the boxes in $N^P \setminus N^B$ on B 's level, where P is the parent box of B .
- $\{y_k^{B,o}\}, \{f_k^{B,o}\}, \{x_k^{B,o}\}$ and $\{u_k^{B,o}\}$ are, respectively, the *outgoing* equivalent points, equivalent densities, 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 densities, check points, and check potentials.

To simplify the notation, we have omitted the range of the running index k .

Now let us consider a box B in the high frequency regime. The following definitions vary slightly from their low frequency counterparts:

- The near field N^B is the union of all the boxes $\{A\}$ that satisfy $dist(A, B) \leq w^2$.
- The far field F^B is the complement of N^B .
- The interaction list I^B contains all the boxes in $N^P \setminus N^B$ on B 's level, where P is B 's parent box.

In light of the preceding sections, we need to partition F^B into a group of directional regions, each belonging to a cone with spanning angle $O(1/w)$. To achieve this, we first cut F^B into six pyramids: $V_{1+}, V_{1-}, V_{2+}, V_{2-}, V_{3+}$ and V_{3-} . For example, V_{1+} contains all the points whose first coordinate is positive and greater than the absolute values of the second and third coordinates. The other pyramids are similarly defined. Let us define $C = 4w$. Each part is further partitioned into C^2 wedges. As an example, for each point $p = (p^1, p^2, p^3)$ in V_{1+} , we define

$$\theta(p) = \arctan(p^2/p^1) \quad \text{and} \quad \phi(p) = \arctan(p^3/p^1).$$

Notice $|\theta(p)| \leq \pi/4$ and $|\phi(p)| \leq \pi/4$ for $p \in V_{1+}$. The C^2 wedges of V_{1+} are

$$\left\{ p : -\frac{\pi}{4} + \frac{\pi}{2C}i \leq \theta(p) \leq -\frac{\pi}{4} + \frac{\pi}{2C}(i+1), -\frac{\pi}{4} + \frac{\pi}{2C}j \leq \phi(p) \leq -\frac{\pi}{4} + \frac{\pi}{2C}(j+1) \right\}$$

for $0 \leq i < C$ and $0 \leq j < C$. Clearly, the spanning angle for each of these wedges is $O(1/w)$. The wedges for other pyramids are generated in the same way. In total, for a box B of width w , its far field F^B is partitioned into $96 \cdot w^2$ wedges. We index these wedges using their center directions $\{\ell\}$. In Figure 3, 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. Suppose $w \geq 2$. For any directional index ℓ of B , one can always find an index ℓ' of the box with width $w/2$ such that the ℓ th wedge of B is contained in the ℓ' th wedge of each of B 's children. On the other hand, these wedges are not isometric anymore, i.e., one cannot obtain any other wedges by applying a rotation to an existing wedge. This implies that the point sets $\{x_c^q\}$ and $\{y_c^p\}$ and the matrix D can be very different for different wedges, even though the algorithm

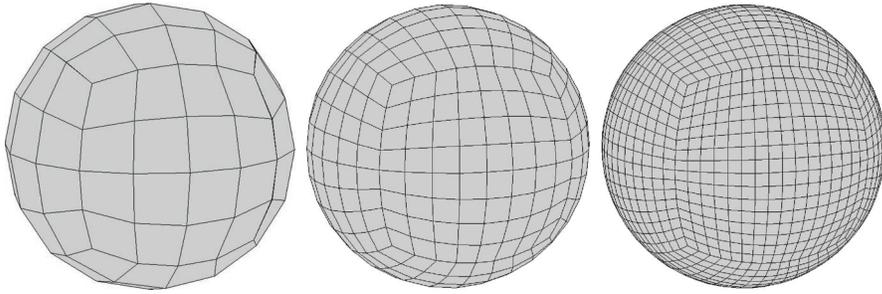


FIG. 3. The wedges cut the unit sphere into $96 \cdot w^2$ pieces. From left to right, $w = 1, 2, 4$.

used to construct them remains exactly the same. Fortunately, one does not need to construct the point sets $\{x_c^q\}$ and $\{y_c^p\}$ and the matrix D for each of these $96 \cdot w^2$ wedges due to a discrete isometric group acting on them. To be more precise, suppose one has a wedge with center direction $\ell = (\ell^1, \ell^2, \ell^3)$, then it is not difficult to see that we can obtain wedges centered along the following 48 directions through rotations:

$$\begin{aligned} &(\pm\ell^1, \pm\ell^2, \pm\ell^3)(\pm\ell^2, \pm\ell^3, \pm\ell^1)(\pm\ell^3, \pm\ell^1, \pm\ell^2) \\ &(\pm\ell^3, \pm\ell^2, \pm\ell^1)(\pm\ell^1, \pm\ell^3, \pm\ell^2)(\pm\ell^2, \pm\ell^1, \pm\ell^3). \end{aligned}$$

Therefore, one is only required to compute the point sets $\{x_c^q\}$ and $\{y_c^p\}$ and the matrix D for about $2w^2$ wedges.

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 densities, 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 densities, check points and check potentials respectively.

4.2. Translation operators. Similar to a standard FMM algorithm, our new algorithm utilizes several translation operators. The translation operators for boxes in the low frequency regime differs from those for boxes in the high frequency regime. Following tradition, we name these operators M2M translation, L2L translation, and L2L translation, though no multipole or local expansions are involved in our algorithm.

Low frequency regime. In the low frequency regime, the M2M, L2L, and M2L translations are exactly the same as the ones introduced in the kernel independent FMM. These operators are non-directional.

M2M operator. For a box B , the M2M operator constructs $\{f_k^{B,o}\}$, the box's outgoing non-directional equivalent densities. We start from a set of source densities, which are either the original densities in B if B is a leaf box or the union of the outgoing non-directional equivalent densities of B 's children if B is not. In this procedure, we first calculate B 's outgoing non-directional check potentials $\{u_k^{B,o}\}$ using kernel evaluation, and perform a matrix-vector multiplication to obtain $\{f_k^{B,o}\}$.

L2L operator. For a box B , the L2L operator starts with $\{u_k^{B,i}\}$, the box's incoming non-directional check potentials. It first constructs the incoming non-directional equivalent densities $\{f_k^{B,i}\}$ via a matrix-vector multiplication. The second step depends on whether B is a leaf box or not. If B is not a leaf, we evaluate the incoming

non-directional check potentials of B 's children. If B is a leaf, then the potentials at the original points inside B are calculated.

M2L operator. The M2L operator works between two boxes A and B on the same level. A and B need to be on each other's interaction lists. The M2L operator transforms the outgoing non-directional equivalent densities of A to the incoming non-directional check potentials of B through kernel evaluation. This operator can be accelerated by the FFT as the kernel is translation invariant.

High frequency regime. The operators in the high frequency regime are more complicated. The main reason is that the computations are now directional.

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

1. Use $\bigcup_C \{y_k^{C,o}\}$ as source points in B and $\bigcup_C \{f_k^{C,o}\}$ as source densities. Here the union is taken over all 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 densities as well. The M2M operator again iterates over all the directional indices $\{\ell\}$ of B . The steps for a fixed direction ℓ are:

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. Use $\bigcup_C \{y_k^{C,o,\ell'}\}$ as source points in B and $\bigcup_C \{f_k^{C,o,\ell'}\}$ as source densities. 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 .

L2L operator. 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 only have non-directional check potentials. The L2L operator iterates over all the directional indices $\{\ell\}$ of B and the steps for a fixed direction ℓ are:

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 densities at $\{y_k^{B,i,\ell}\}$.

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

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 children 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 densities at $\{y_k^{B,i,\ell}\}$.

M2L operator. 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 densities $\{f_k^{A,o,\ell}\}$ at points $\{y_k^{A,o,\ell'}\}$.

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

4.3. Algorithm. 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. These boxes have width less than 1. For each box B , compute its outgoing non-directional equivalent densities $\{f_k^{B,o}\}$. This is done using the low frequency non-directional 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 the outgoing directional equivalent densities $\{f_k^{B,o,\ell}\}$ 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 , perform the following two steps:
 - (a) Transform the outgoing directional equivalent densities $\{f_k^{A,o,\ell}\}$ of all the boxes $\{A\}$ in B 's interaction list and in direction ℓ via the high-frequency directional M2L operator. Next, add the result to the incoming directional check potentials $\{u_k^{B,i,\ell}\}$.
 - (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, we perform the following two steps:
 - (a) Transform the outgoing non-directional equivalent densities $\{f_k^{A,o}\}$ of all the boxes $\{A\}$ in B 's interaction list via the low frequency non-directional M2L operator. Next, add the result to the incoming non-directional check potentials $\{u_k^{B,i}\}$.
 - (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 various components of the algorithm is given in Figure 4.

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

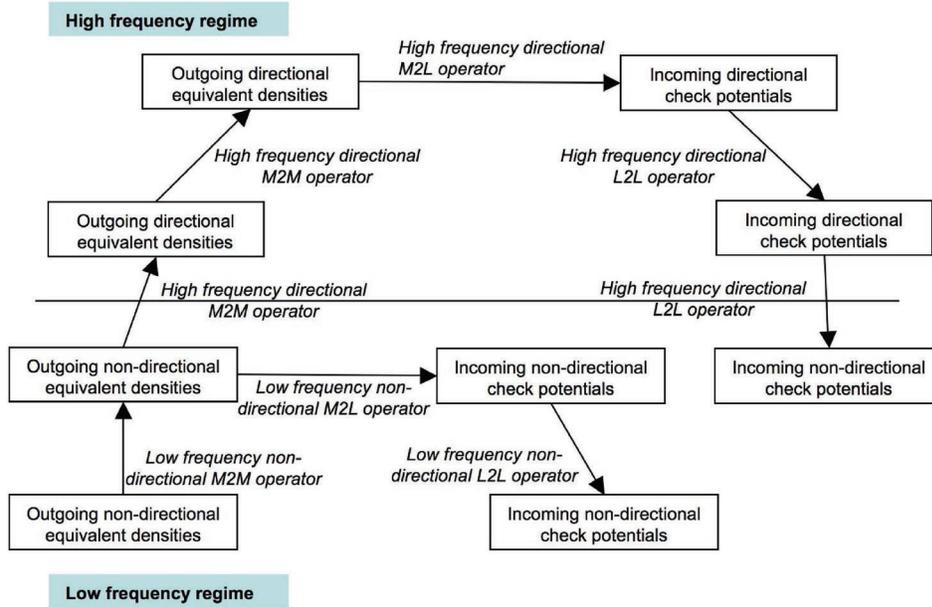


FIG. 4. A small part of the octree used in the computation. Each rectangular region stands for a box of the octree. The diagram shows how the outgoing non-directional equivalent densities from a leaf box have been transformed into incoming non-directional check potentials at other leaf boxes. Far field interaction involves directional computation in the high frequency regime. The text in each box denotes the quantities involved, while the translation operators are shown in italics.

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 [19, 34, 47].

4.4. Complexity. We will now discuss the computational complexity of our algorithm assuming that the N points are sampled from a two dimensional surface. This assumption implies that $N = O(K^2)$ and it is generally satisfied for almost all applications which involve boundary integral formulations. In the following analysis, we use K as the main parameter because all the points $\{p_i\}$ satisfy $|p_i| \leq K/2$ by assumption.

THEOREM 4.1. *Let \mathbb{S} be a surface in $B(0,1/2)$. Suppose that for a fixed K , the points $\{p_i, 1 \leq i \leq N\}$ 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 $O(K^2 \log K) = O(N \log N)$.*

Outline of the proof. We analyze the number of operations for each major step of the algorithm.

- The first step of the algorithm has computational complexity $O(N) = O(K^2)$, due to the efficiency of the octree construction.
- The second step takes $O(N) = O(K^2)$ operations, as there are at most $O(N)$ boxes in the low frequency regime and each one involves a constant number of operations.
- We claim that the third step takes $O(N \log N)$ operations. To verify this, we first look at a single box of a fixed width w ($1 \leq w \leq \sqrt{K}$). Since there are

$O(w^2)$ directions $\{\ell\}$, the M2M operator for this box takes $O(w^2)$ operations. Next count the number of boxes of size w . Noticing that our points are discrete samples from a two dimensional manifold inside $B(0, K/2)$, it is not difficult to see that there are only $O(K^2/w^2)$ of them. This means that the overall complexity of the M2M operator for all the boxes of size w is $O(w^2 \cdot K^2/w^2) = O(K^2)$. Since we have totally $O(\log K)$ levels with $w = 1, 2, 4, \dots, \sqrt{K}$, the complexity of this step is $O(K^2 \log K) = O(N \log N)$.

- We argue that the fourth step takes $O(N \log N)$ operations. The computation here contains two parts: the M2L operators and the L2L operators. The analysis for the L2L operators is exactly the same as the analysis of the third step, resulting an $O(N \log N)$ operation count. For the M2L operators, we know that, for a box B of width w , the boxes in its high frequency interaction list are approximately located in a ball centered at B with radius $(2w)^2$. Again, the fact that our points are samples from a two dimensional manifold implies that there are at most $O(w^4/w^2) = O(w^2)$ boxes in B 's interaction list. Noticing that each M2L operator takes $O(1)$ operations and there are at most $O(K^2/w^2)$ boxes with width w , we conclude that overall complexity of the M2L operators for all the boxes of size w is $O(w^2 \cdot 1 \cdot K^2/w^2) = O(K^2)$. This further implies that the total number of the operations in the M2L operators is $O(K^2 \log K) = O(N \log N)$.
- Similar to the second step, the fifth step takes $O(N) = O(K^2)$ operations by the same argument.

Summing these contributions, we reach the conclusion that the total complexity is $O(N \log N)$, which is the same as the HF-FMM developed by Rokhlin and his collaborators (see, for example, [18]). \square

5. Numerical results. In this section, we provide some numerical results to illustrate the properties of our new algorithm. Our implementation is written in C++ and all the computational results below are obtained on a desktop computer with a 3.0 GHz CPU. Due to the memory constraint, we restrict ourselves to problems which are at most 256 wavelengths (i.e., $K \leq 256$). The equivalent points, check points, and the related D matrices are precomputed for boxes with width $w = 1, 2, 4, 8$.

5.1. Separation rank. Let us first study the performance of the randomized procedure presented in section 3. In Table 1, we list the number of terms in the separated representation for two sets X_r and Y_r for different choices of accuracy ε and box width w . Here r is set to be $\sqrt{3}w$ so that the box of width w is contained in Y_r .

We can see from Table 1 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 $\log(1/\varepsilon)$. This is compatible with the results mentioned in [40].

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
$\varepsilon=1e-6$	85	82	81	81
$\varepsilon=1e-8$	121	114	113	111

The actual running time of the high frequency part of our algorithm depends on ε through the separation rank. In fact, the smaller ε , the larger the separation rank, and the longer the running time. As we pointed out already, the low frequency part of our algorithm is almost the same as the one described in [47]. For a prescribed value of ε , we need to choose a value for p , which is the size of the Cartesian grid used for the equivalent and check points. In our implementation, we pick $p = 4$ for $\varepsilon = 1e-4$, $p = 6$ for $\varepsilon = 1e-6$ and $p = 8$ for $\varepsilon = 1e-8$.

5.2. Applications to different geometries. We applied our algorithm, with G being the Helmholtz kernel, to several objects. 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 20 points per wavelength on average. This implies that the number of points is roughly 400 times the total surface area. Though these samples are far from optimal for a scattering application, they have the right distribution and are perfectly suitable for studying the performance of our algorithm.

In real applications, the actual sampling density often depends on the geometry of the surface. For example, one often places many more points in regions with large curvature. The result is to shift more computation towards the low frequency regime. Since the computation in the low frequency regime has lower complexity than the one for the high-frequency regime, our algorithm would generally perform better in these cases. 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, and ε is the prescribed error threshold such that the final error of is to be bounded by a constant multiple of ε .

In all experiments, the original densities $\{f_i\}$ are generated from a random distribution with mean 0. Use $\{u_i\}$ to denote the true potentials and $\{u_i^a\}$ to denote the approximations obtained through our algorithm. We estimate the relative error by picking a set S of 200 points from $\{p_i\}$. The true potentials $\{u_i, i \in S\}$ are computed by using direct evaluation. The error is then estimated to be

$$\sqrt{\frac{\sum_{i \in S} |u_i - u_i^a|^2}{\sum_{i \in S} |u_i|^2}}.$$

When reporting the numerical results, we use the following notations:

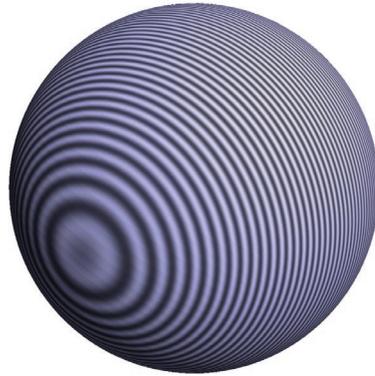
- T_a : the running time of our algorithm in seconds.
- T_d : the running time of the direct evaluation in seconds. This quantity is estimated by first calculating the time used for evaluating the true potentials at 200 points in S and then multiplying it by $N/200$.
- T_d/T_a : the speedup factor.
- ε_a : the error of our algorithm estimated using the method described above.

The first example is a sphere. The results are summarized in Table 2. The second example is a simplified model of an F16. The results are shown in Table 3. The final example is a submarine model. We summarize the computational results in Table 4.

From these tests, we can make the following observations about the performance of our algorithm:

- The running time scales roughly as $O(N \log N)$ in terms of the number of particles, matching well with the complexity analysis. In our implementation, most of the computation has been spent on computing $\{f_k^{B,o,\ell}\}$ and $\{u_k^{B,i,\ell}\}$ in steps 3 and 4(a) of the proposed algorithm (see section 4.3). The construction of one set of densities $\{f_k^{B,o,\ell}\}$ (or equivalently one set of potentials

TABLE 2
Results of the sphere with the Helmholtz kernel.

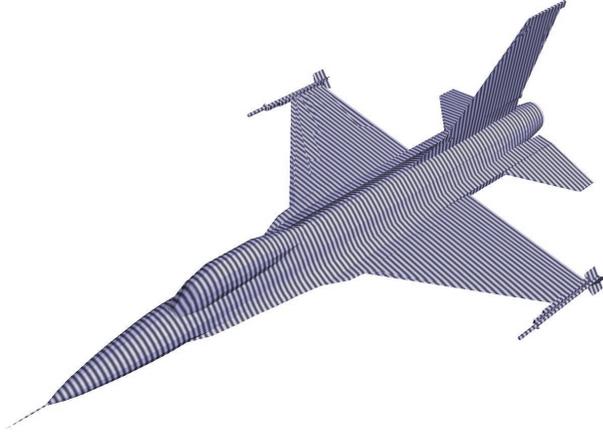


(K, ε)	N	T_a (sec)	T_d (sec)	T_d/T_a	ε_a
(16,1e-4)	3.22e+5	9.50e+1	1.21e+4	1.28e+2	5.08e-4
(32,1e-4)	1.29e+6	4.28e+2	1.95e+5	4.55e+2	5.91e-4
(64,1e-4)	5.15e+6	1.97e+3	3.04e+6	1.54e+3	6.30e-4
(16,1e-6)	3.22e+5	2.42e+2	1.18e+4	4.86e+1	2.92e-6
(32,1e-6)	1.29e+6	1.21e+3	1.87e+5	1.54e+2	2.12e-6
(64,1e-6)	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

$\{u_k^{B,i,\ell}\}$) requires a couple of matrix multiplications, where the dimensions of the matrices involved are approximately equal to the separation rank. Since the complexity of each matrix multiplication is quadratic in the dimension of the matrix, the overall complexity of our algorithm scales quadratically in terms of the separation rank (see Table 1).

- For a fixed threshold ε , the final error of our directional multilevel algorithm seems to depend linearly on $\log_2 \sqrt{K}$, i.e., the number of levels in the octree. This is easy to understand: every time we compute the equivalent densities in a higher level, we introduce some extra error proportional to ε . Since $\log_2 K$ is a quite small number, for example $K = 256$ gives $\log_2 K = 8$, we still have very good control of the overall error.
- For objects on a fixed scale, our algorithm runs faster if the shape of an object is elongated. For example, the results on the submarine are better than the ones for the plane, which are better than the ones for the sphere. In other words, the algorithm prefers slender objects. The reason is quite simple: for an elongated object, the number of directions $\{\ell\}$ a box needs to address is much smaller than the number of directions for an isotropic object. In fact, the sphere seems to be the most difficult surface to work with.
- A close look at the breakdown of the computation time shows that the algorithm spends most of its time in the high frequency regime. This implies that for real problems where certain parts need sub-wavelength sampling, our algorithm would result a larger speedup factor compared with direct evaluation.

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



(K, ε)	N	$T_a(\text{sec})$	$T_d(\text{sec})$	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

5.3. Generalization. As we have alluded, our algorithm works well for other oscillatory kernels as well. Here we provide some numerical results for the kernel,

$$e^{2\pi i|x-y|}.$$

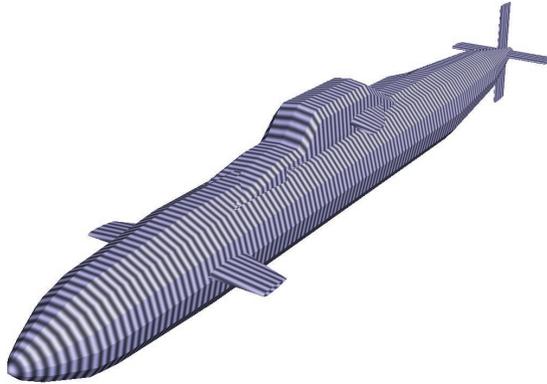
For this kernel, the proof in section 2 remains essentially the same, and the only change is to remove the estimate on the $1/|x-y|$ term. The randomized procedure in section 3 requires no modification as it only relies on the fact that the kernel has a low rank separated representation. Hence, our algorithm remains exactly the same in the high frequency regime. The only modification concerns the low frequency boxes. Our implementation follows the discussion in [46] and we refer to that paper for the details.

In Table 5, we report the results for the sphere with this new kernel. In Table 6, the results of the F16 model are presented.

These results suggest that the overall error of our algorithm for this new kernel with stronger far field oscillations is larger than the error for the Helmholtz kernel. However, the running time still follows the $O(N \log N)$ complexity very closely.

6. Conclusions and future work. In this paper, we introduced a new directional multilevel algorithm for computing the N -body problem with highly oscillatory kernels, and applied it to a boundary integral formulation of the Helmholtz equation. Our algorithm is based on the following three main components:

TABLE 4
Results of the submarine model with the Helmholtz kernel.



(K, ε)	N	$T_a(\text{sec})$	$T_d(\text{sec})$	T_d/T_a	ε_a
(32, 1e-4)	1.47e+5	3.90e+1	2.52e+3	6.46e+1	5.35e-4
(64, 1e-4)	5.85e+5	1.83e+2	4.00e+4	2.19e+2	6.37e-4
(128, 1e-4)	2.34e+6	8.26e+2	6.57e+5	7.95e+2	6.32e-4
(256, 1e-4)	9.36e+6	3.87e+3	1.01e+7	2.61e+3	8.32e-4
(32, 1e-6)	1.47e+5	9.10e+1	2.49e+3	2.74e+1	2.64e-6
(64, 1e-6)	5.85e+5	5.22e+2	4.07e+4	7.80e+1	2.17e-6
(128, 1e-6)	2.34e+6	2.55e+3	6.39e+5	2.51e+2	4.64e-6
(32, 1e-8)	1.47e+5	1.81e+2	2.50e+3	1.38e+1	6.37e-8
(64, 1e-8)	5.85e+5	1.08e+3	3.98e+4	3.68e+1	8.05e-8
(128, 1e-8)	2.34e+6	5.41e+3	6.63e+5	1.23e+2	8.51e-8

TABLE 5
Results of the sphere with the kernel $e^{2\pi i|x-y|}$.

(K, ε)	N	$T_a(\text{sec})$	$T_d(\text{sec})$	T_d/T_a	ε_a
(16, 1e-4)	3.22e+5	1.12e+2	1.14e+4	1.02e+2	5.09e-3
(32, 1e-4)	1.29e+6	5.10e+2	1.83e+5	3.58e+2	5.40e-3
(64, 1e-4)	5.15e+6	2.36e+3	2.85e+6	1.21e+3	5.74e-3
(16, 1e-6)	3.22e+5	3.82e+2	1.11e+4	2.90e+1	7.83e-6
(32, 1e-6)	1.29e+6	1.92e+3	1.77e+5	9.21e+1	7.23e-6
(64, 1e-6)	5.15e+6	9.36e+3	2.93e+6	3.13e+2	5.48e-6

TABLE 6
Results of the F16 model with the kernel $e^{2\pi i|x-y|}$.

(K, ε)	N	$T_a(\text{sec})$	$T_d(\text{sec})$	T_d/T_a	ε_a
(32, 1e-4)	1.87e+5	5.70e+1	3.79e+3	6.64e+1	6.31e-3
(64, 1e-4)	7.46e+5	2.64e+2	6.25e+4	2.37e+2	6.45e-3
(128, 1e-4)	2.98e+6	1.24e+3	9.81e+5	7.94e+2	6.75e-3
(32, 1e-6)	1.87e+5	1.81e+2	3.82e+3	2.11e+1	8.79e-6
(64, 1e-6)	7.46e+5	9.52e+2	6.12e+4	6.43e+1	9.00e-6
(128, 1e-6)	2.98e+6	4.76e+3	9.45e+5	1.99e+2	7.79e-6

- The directional low rank property, i.e., the interaction between two sets that follow the directional parabolic separation condition is approximately of low rank.
- A randomized procedure which efficiently and accurately constructs low rank, separated and stable representations.

- A multilevel and multidirectional strategy for organizing the overall computation. The fact that the high frequency part of our algorithm is not only multiscale but also multidirectional differs from the approach in the HF-FMM.

Our algorithm has been proved to have $O(N \log N)$ complexity and the numerical results show that it is also highly accurate. Furthermore, it can be adapted to handle kernels other than the Helmholtz kernel quite easily, which is not true for most other existing algorithms.

In future work, we plan to consider the following:

- It would be valuable to have a rigorous proof for the randomized procedure presented in section 3. The discussion presented here provides a possible outline of a proof.
- Currently, our implementation is capable of handling problems of up to about 256 wavelengths, however, many interesting scattering problems involve thousands of wavelengths. One solution is to design a parallel version of our algorithm. It is well-known that parallelizing standard FMMs is difficult since the top part of the octree is a bottleneck (see [48]). Fortunately, this bottleneck is alleviated by our algorithm, as we never visit the boxes of size greater than \sqrt{K} wavelengths. A trivial implementation of partitioning the work at the level with boxes of size \sqrt{K} would provide an appealing and practical solution.
- An FMM-type algorithm has been developed by Michielssen and his collaborators for time-domain scattering problems (see [20, 28]). It utilizes some of the techniques from the HF-FMM. It would be interesting to see whether our algorithm can be extended to handle time-domain problems.

Acknowledgments. The work presented in this paper is partially supported by startup funds of University of Texas at Austin. The authors thank R. Tsai and M. Tygert for stimulating discussions, L. Greengard, J. Qin, and R. Sharp for helping improve the manuscript, and the reviewers for their comments and suggestions.

REFERENCES

- [1] B. ALPERT, G. BEYLKIN, R. COIFMAN, AND V. ROKHLIN, *Wavelet-like bases for the fast solution of second-kind integral equations*, SIAM J. Sci. Comput., 14 (1993), pp. 159–184.
- [2] C. R. ANDERSON, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 923–947.
- [3] A. AVERBUCH, E. BRAVERMAN, R. COIFMAN, M. ISRAELI, AND A. SIDI, *Efficient computation of oscillatory integrals via adaptive multiscale local Fourier bases*, Appl. Comput. Harmon. Anal., 9 (2000), pp. 19–53.
- [4] M. BEBENDORF, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.
- [5] M. BEBENDORF AND S. RJASANOW, *Adaptive low-rank approximation of collocation matrices*, Computing, 70 (2003), pp. 1–24.
- [6] G. BEYLKIN, R. COIFMAN, AND V. ROKHLIN, *Fast wavelet transforms and numerical algorithms. I*, Comm. Pure Appl. Math., 44 (1991), pp. 141–183.
- [7] E. BLESZYNSKI, M. BLESZYNSKI, AND T. JAROSZEWICZ, *AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems*, Radio Science, 31 (1996), pp. 1225–1252.
- [8] N. BOJARSKI, *K-space formulation of the electromagnetic scattering problems*, Technical report, Air Force Avionic Lab. Technical Report AFAL-TR-71-75, 1971.
- [9] S. BÖRM, L. GRASEDYCK, AND W. HACKBUSCH, *Hierarchical matrices*, Technical Report 21, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2003.

- [10] B. BRADIE, R. COIFMAN, AND A. GROSSMANN, *Fast numerical computations of oscillatory integrals related to acoustic scattering*. I, *Appl. Comput. Harmon. Anal.*, 1 (1993), pp. 94–99.
- [11] A. BRANDT AND A. A. LUBRECHT, *Multilevel matrix multiplication and fast solution of integral equations*, *J. Comput. Phys.*, 90 (1990), pp. 348–370.
- [12] O. P. BRUNO AND L. A. KUNYANSKY, *A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications*, *J. Comput. Phys.*, 169 (2001), pp. 80–110.
- [13] O. M. BUCCI AND G. FRANCESCHETTI, *On the degrees of freedom of scattered fields*, *IEEE Trans. Antennas Propagat.*, 37 (1989), pp. 918–926.
- [14] E. J. CANDÈS, L. DEMANET, AND L. YING, *Fast computation of Fourier integral operators*, Technical report, California Institute of Technology, 2006.
- [15] E. J. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, *IEEE Trans. Inform. Theory*, 52 (2006), pp. 489–509.
- [16] F. X. CANNING, *Sparse approximation for solving integral equations with oscillatory kernels*, *SIAM J. Sci. Statist. Comput.*, 13 (1992), pp. 71–87.
- [17] J. CARRIER, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm for particle simulations*, *SIAM J. Sci. Statist. Comput.*, 9 (1988), pp. 669–686.
- [18] H. CHENG, W. Y. CRUTCHFIELD, Z. GIMBUTAS, L. F. GREENGARD, J. F. ETHRIDGE, J. HUANG, V. ROKHLIN, N. YARVIN, AND J. ZHAO, *A wideband fast multipole method for the Helmholtz equation in three dimensions*, *J. Comput. Phys.*, 216 (2006), pp. 300–325.
- [19] H. CHENG, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm in three dimensions*, *J. Comput. Phys.*, 155 (1999), pp. 468–498.
- [20] W. C. CHEW, E. MICHIELSSEN, J. M. SONG, AND J. M. JIN, EDS., *Fast and efficient algorithms in computational electromagnetics*, Artech House, Inc., Norwood, MA, USA, 2001.
- [21] D. L. COLTON AND R. KRESS, *Integral equation methods in scattering theory*, *Pure and Appl. Math.* (New York), John Wiley & Sons Inc., New York, 1983. A Wiley-Interscience Publication.
- [22] B. CRANGANU-CRETU, J. OSTROWSKI, AND Z. ANDJELIC, *Fast integral equation solution for electric field shielding*, *COMPEL: Internat. J. Comput. and Math.*, in *Electrical and Electronic Eng.*, 24 (2005), pp. 639–647.
- [23] E. DARVE, *The fast multipole method: Numerical implementation*, *J. Comput. Phys.*, 160 (2000), pp. 195–240.
- [24] E. DARVE AND P. HAVÉ, *A fast multipole method for Maxwell equations stable at all frequencies*, *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 362 (2004), pp. 603–628.
- [25] P. DRINEAS, R. KANNAN, AND M. W. MAHONEY, *Fast Monte Carlo algorithms for matrices. II. Computing a low-rank approximation to a matrix*, *SIAM J. Comput.*, 36 (2006), pp. 158–183 (electronic).
- [26] P. DRINEAS, R. KANNAN, AND M. W. MAHONEY, *Fast Monte Carlo algorithms for matrices. III. Computing a compressed approximate matrix decomposition*, *SIAM J. Comput.*, 36 (2006), pp. 184–206 (electronic).
- [27] M. A. EPTON AND B. DEMBART, *Multipole translation theory for the three-dimensional Laplace and Helmholtz equations*, *SIAM J. Sci. Comput.*, 16 (1995), pp. 865–897.
- [28] A. A. ERGIN, B. SHANKAR, AND E. MICHIELSSEN, *The plane-wave time-domain algorithm for the fast analysis of transient wave phenomena*, *IEEE Antennas and Propagation Magazine*, 41 (1999), pp. 39–52.
- [29] G. T. D. FRANZIA, *Degrees of freedom of an image*, *J. Opt. Soc. Am.*, 59 (1969), pp. 779–804.
- [30] S. A. GOREINOV, E. E. TYRTYSHNIKOV, AND N. L. ZAMARASHKIN, *A theory of pseudoskeleton approximations*, *Linear Algebra Appl.*, 261 (1997), pp. 1–21.
- [31] S. A. GOREINOV, N. L. ZAMARASHKIN, AND E. E. TYRTYSHNIKOV, *Pseudoskeleton approximations by submatrices of greatest size*, *Mat. Zametki*, 62 (1997), pp. 619–623.
- [32] L. GREENGARD, *The rapid evaluation of potential fields in particle systems*, *ACM Distinguished Dissertations*, MIT Press, Cambridge, MA, 1988.
- [33] L. GREENGARD, J. HUANG, V. ROKHLIN, AND S. WANDZURA, *Accelerating fast multipole methods for the Helmholtz equation at low frequencies*, *IEEE Comput. Sci. Eng.*, 5 (1998), pp. 32–38.
- [34] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, *J. Comput. Phys.*, 73 (1987), pp. 325–348.
- [35] W. HACKBUSCH AND Z. P. NOWAK, *On the fast matrix multiplication in the boundary element method by panel clustering*, *Numer. Math.*, 54 (1989), pp. 463–491.

- [36] J. G. HARRIS, *Linear elastic waves*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2001.
- [37] R. JAKOB-CHIEN AND B. K. ALPERT, *A fast spherical filter with uniform resolution*, J. Comput. Phys., 136 (1997), pp. 580–584.
- [38] S. KAPUR AND D. E. LONG, *IES3: A fast integral equation solver for efficient 3-dimensional extraction*, in ICCAD '97: Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design, Washington, DC, USA, 1997, IEEE Computer Society, pp. 448–455.
- [39] P.-G. MARTINSSON, V. ROKHLIN, AND M. TYGERT, *A randomized algorithm for the approximation of matrices*, Technical report, Yale University, 2006.
- [40] E. MICHIELSEN AND A. BOAG, *A multilevel matrix decomposition algorithm for analyzing scattering from large structures*, IEEE Transactions on Antennas and Propagation, 44 (1996), pp. 1086–1093.
- [41] N. NISHIMURA, *Fast multipole accelerated boundary integral equation methods*, Applied Mechanics Reviews, 55 (2002), pp. 299–324.
- [42] V. ROKHLIN, *Rapid solution of integral equations of scattering theory in two dimensions*, J. Comput. Phys., 86 (1990), pp. 414–439.
- [43] V. ROKHLIN, *Diagonal forms of translation operators for the Helmholtz equation in three dimensions*, Appl. Comput. Harmon. Anal., 1 (1993), pp. 82–93.
- [44] M. RUDELSON AND R. VERSHYNIN, *Sparse reconstruction by convex relaxation: Fourier and gaussian measurements*, 2006.
- [45] J. M. SONG AND W. C. CHEW, *Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering*, Microwave Opt. Tech. Lett., 10 (1995), pp. 15–19.
- [46] L. YING, *A kernel independent fast multipole algorithm for radial basis functions*, J. Comput. Phys., 213 (2006), pp. 451–457.
- [47] L. YING, G. BIROS, AND D. ZORIN, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys., 196 (2004), pp. 591–626.
- [48] L. YING, G. BIROS, D. ZORIN, AND H. LANGSTON, *A new parallel kernel-independent fast multipole method*, in SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing, Washington, DC, USA, 2003, IEEE Computer Society, p. 14.