

## A FAST DIRECTIONAL ALGORITHM FOR HIGH FREQUENCY ACOUSTIC SCATTERING IN TWO DIMENSIONS\*

BJÖRN ENGQUIST<sup>†</sup> AND LEXING YING<sup>‡</sup>

**Abstract.** This paper is concerned with fast solution of high frequency acoustic scattering problems in two dimensions. We introduce a directional multiscale algorithm for the  $N$ -body problem of the two dimensional Helmholtz kernel. The algorithm follows the approach developed in [Engquist and Ying, SIAM J. Sci. Comput., 29 (4), 2007], where the three dimensional case was studied. The main observation is that, for two regions that follow a directional parabolic geometric configuration, the interaction between these two regions through the 2D Helmholtz kernel is approximately low rank. We propose an improved randomized procedure for generating the low rank separated representation for the interaction between these regions. Based on this representation, the computation of the far field interaction is organized in a multidirectional and multiscale way to achieve maximum efficiency. The proposed algorithm is accurate and has the optimal  $O(N \log N)$  complexity for problems from two dimensional scattering applications. Finally, we combine this fast directional algorithm with standard boundary integral formulations to solve acoustic scattering problems that are of thousands of wavelengths in size.

**Key words.**  $N$ -body problems, Helmholtz equation, oscillatory kernels, fast multipole methods, multidirectional computation, multiscale methods.

**AMS subject classifications.** 65N38; 65R20.

### 1. Introduction

**1.1. Problem statement.** This paper is concerned with the rapid solution of time harmonic acoustic scattering problems [10] in two dimensions. Suppose that  $D$  is a smooth bounded object with a diameter of  $K$  wavelengths and  $u^{inc}(x)$  to be the incoming field. If  $D$  represents an impenetrable scatterer, the scattered field  $u(x)$  satisfies the Helmholtz equation with Dirichlet boundary condition:

$$\begin{aligned} \Delta u(x) + (2\pi)^2 u(x) &= 0 \quad \text{for } x \in \mathbb{R}^d \setminus \bar{D} \\ u(x) &= -u^{inc}(x) \quad \text{for } x \in \partial D \end{aligned} \quad (1.1)$$

$$\lim_{|x| \rightarrow \infty} |x|^{1/2} \left( \left( \frac{x}{|x|}, \nabla u(x) \right) - 2\pi i u(x) \right) = 0.$$

Here we set the wave number to be  $2\pi$  and the wavelength  $\lambda$  equal to 1. The last condition is the Sommerfeld radiation condition and guarantees that the scattered field  $u(x)$  propagates to infinity. In this setup, the solution becomes oscillatory when the scatterer  $D$  is large. One efficient way of solving this problem is to reformulate it with a boundary integral equation (BIE) [10] such as

$$\frac{1}{2} \phi(x) + \int_{\partial D} \left( \frac{\partial G(x,y)}{\partial n(y)} - \eta \cdot G(x,y) \right) \phi(y) dy = -u^{inc}(x), \quad \forall x \in \partial D \quad (1.2)$$

where  $\phi(x)$  for  $x \in \partial D$  is the unknown distribution on  $\partial D$ ,  $n(y)$  is the exterior normal of  $\partial D$  at  $y$ , and  $\eta$  is a constant  $\approx 2\pi$ . The kernel  $G(x,y) = \frac{i}{4} H_0^{(1)}(2\pi|x-y|)$  is the

---

\*Received: November 10, 2008; accepted: January 28, 2009.

<sup>†</sup>Department of Mathematics and Institute for Computational Engineering and Sciences, 1 University Station, C1200, University of Texas at Austin, Austin, TX 78712, USA (engquist@math.utexas.edu).

<sup>‡</sup>Department of Mathematics and Institute for Computational Engineering and Sciences, 1 University Station, C1200, University of Texas at Austin, Austin, TX 78712, USA (lexing@math.utexas.edu).

fundamental solution of the 2D Helmholtz equation. Here  $H_0^{(1)}$  denotes the first kind Hankel function of order zero and we use  $\iota$  to denote  $\sqrt{-1}$ . The main advantages of the BIE approach is that it reduces an unbounded 2D problem (1.1) to a bounded 1D problem (1.2). In addition, the condition number of (1.2) is independent of  $K$ . Once  $\phi$  is solved from (1.2), the scattered field  $u(x)$  can be simply computed by

$$u(x) = \int_{\partial D} \left( \frac{\partial G(x,y)}{\partial n(y)} - \eta \cdot G(x,y) \right) \phi(y) dy, \quad \forall x \in \mathbb{R}^2 \setminus \bar{D}.$$

One obvious disadvantage of the boundary formulation (1.2) is that it is a dense operator. Therefore, iterative algorithms such as GMRES [34] become the natural tool for solving the resulting discrete system. At each step of the iterative solver, one then needs to evaluate the following  $N$ -body problem of the high frequency Helmholtz kernel

$$u_i = \sum_{j=1}^N G(p_i, p_j) \cdot f_j \quad (1.3)$$

where  $\{p_i\}_{1 \leq i \leq N} \subset B(0, K/2)$  are the appropriate quadrature points,  $\{f_i\}_{1 \leq i \leq N}$  are the sources at  $\{p_i\}_{1 \leq i \leq N}$ , and  $\{u_i\}_{1 \leq i \leq N}$  are the potentials. A similar problem can be formulated for the kernel  $\frac{\partial G(x,y)}{\partial n(y)}$  as well.

It is well known that the complexity of a scattering problem often scales with the size of scatterer in terms of the wavelength. Since the wavelength is taken to be one in our setup, the complexity of (1.3) depends on the size of the scatterer  $K$ , which can be on the order of thousands or even tens of thousands for a typical large scale scattering problem. Since one often discretizes (1.2) with a constant number of points per wavelength, the number of points  $N$  is then proportional to  $K$ .

**1.2. Previous work.** The direct computation of (1.3) takes  $O(N^2) = O(K^2)$  operations. This can be quite time consuming when  $K$  is large. Various fast algorithms have been proposed to reduce this complexity in the past two decades. Among them, the most popular approach is the high frequency fast multipole method (HF-FMM) developed by Rokhlin et al. [8, 32]. In the HF-FMM, the whole computational domain is partitioned into a quadtree and one associates with each square of the quadtree two expansions: the far field expansion and the local field expansion [8]. These expansions accelerate the computation in the low frequency regime. In the high frequency regime, since the translations between them become diagonal operators under the Fourier basis, the Fourier transforms of these expansions are used instead to achieve optimal efficiency. The HF-FMM has an optimal  $O(N \log N)$  complexity and has been widely used. Other algorithms using related techniques can be found in [9, 11, 36].

Another approach for speeding up the computation of (1.3) is to exploit the translation-invariant property of the kernel and use the fast Fourier transform to perform the non-adjacent computation in the Fourier domain [2, 6]. Though quite efficient for many situations, the asymptotic complexity of this approach is not optimal.

A different way to accelerate the  $N$ -body computation is to discretize the integral Equ. (1.2) under the Galerkin framework with local Fourier bases or wavelet packets. The stiffness matrix becomes approximately sparse under these bases since most of the entries are close to zero and can be safely discarded. Early algorithms [1, 4, 7, 14, 15, 19] of this approach focus on finding the correct one dimensional basis, while

recent developments [13, 24] consider two dimensional tensor product wave packets and wave atoms, thus offering more flexibility and better compression rates.

In [30], Michielssen and Boag proposed a multilevel multiplication algorithm to bring the overall complexity down to  $O(N \log^2 N)$ . This algorithm is quite similar to the FFT and is later extended by [31] to more general problems.

In [18], we proposed an algorithm for the three dimensional  $N$ -body problem of the high frequency Helmholtz kernel. It relies on a low rank property of the 3D Helmholtz kernel for certain geometric configurations. The algorithm organizes the computation in a multidirectional and multilevel fashion and has an optimal  $O(N \log N)$  complexity.

**1.3. A multidirectional approach.** In this paper, we adapt the approach in [18] to the two dimensional  $N$ -body problem of the Helmholtz kernel. The main observation is a similar low rank property of the 2D Helmholtz kernel. We say that two sets  $Y$  and  $X$  satisfy the *directional parabolic separation condition* if  $Y$  is a disk of radius  $r$  and  $X$  is a conical region with spanning angle  $1/r$  and at least  $r^2$  away from  $Y$  (see figure 1.1).

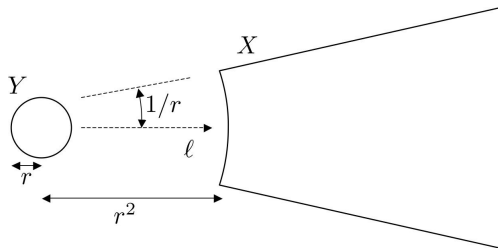


FIG. 1.1. Two sets  $Y$  and  $X$  that satisfy the directional parabolic separation condition.

Once  $Y$  and  $X$  satisfy the directional parabolic separation condition, one can show that for any fixed accuracy the interaction between  $X$  and  $Y$  via the Helmholtz kernel  $G(x, y)$  is approximately low rank and the rank is independent of the size of  $Y$ . More precisely, for any accuracy  $\varepsilon$  there exists a constant  $t_\varepsilon$  and two sets of functions  $\{\alpha_i(x)\}_{1 \leq i \leq t_\varepsilon}$  and  $\{\beta_i(y)\}_{1 \leq i \leq t_\varepsilon}$  such that

$$\left| G(x, y) - \sum_{i=1}^{t_\varepsilon} \alpha_i(x) \beta_i(y) \right| \leq \varepsilon, \quad \forall x \in X, \forall y \in Y.$$

Notice that  $\{\alpha_i(x)\}_{1 \leq i \leq t_\varepsilon}$  and  $\{\beta_i(y)\}_{1 \leq i \leq t_\varepsilon}$  are only functions of  $x$  and  $y$  respectively. We call such an approximation a *directional separated approximation*. One major component of our approach is to use this representation to build equivalent sources for well-separated interaction.

Similar to the 3D algorithm in [18], our 2D algorithm starts by generating a quadtree for the whole computational domain (see figure 1.2(a)). In the bottom part of the quadtree where the squares are of size less than 1, the interactions are accelerated using the kernel independent FMM algorithm in [38]. In the top part where the squares are of size greater than or equal to 1, the far field of each square is partitioned into wedges that follow the directional parabolic separation condition (see figure 1.2(b)). Between the square and each one of its wedges, the computation is then accelerated via the directional separated approximation.

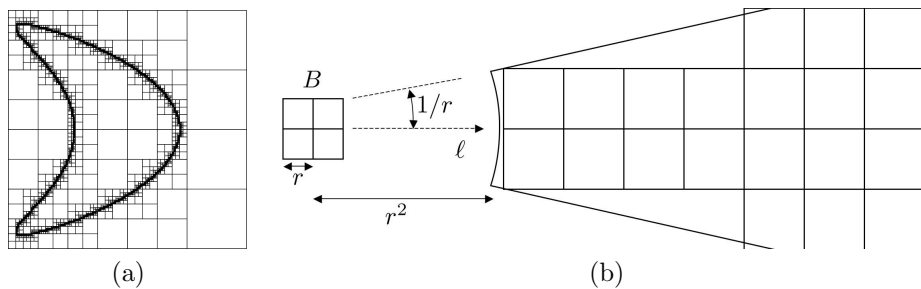


FIG. 1.2. (a) The quadtree constructed for a point distribution supported on a curve. (b) For each square  $B$  in the high frequency regime, its far field is partitioned into multiple wedges. We construct a separated approximation for the interaction between  $B$  and each of its wedges. This approximation is used to accelerate the computation between  $B$  and the wedge.

**1.4. Contributions and contents.** Apart from extending the multidirectional algorithm of [18] to the 2D Helmholtz kernel, this paper also contains two new contributions:

- We provide an improved randomized procedure for constructing the directional separated approximations. This new procedure is more efficient and it generates approximations with smaller ranks in practice.
- We apply our multidirectional algorithm to the solution of boundary integral formulations of high frequency scattering problems, which is not performed in [18]. This allows us to study large scatterers that are thousands of wavelengths wide.

We would like to mention that the idea of using directional separation for oscillatory kernels appeared earlier in the work of [5, 3]. However, the current algorithm is the first attempt to achieve an  $O(N \log N)$  algorithm for the  $N$ -body problem of the 2D Helmholtz kernel using the directional separation idea.

The rest of this paper is organized as follows. In section 2, we describe the improved procedure for constructing the separated approximations. Our multidirectional algorithm is described in detail in section 3 and in section 4 we present numerical results for several test examples. In section 5 we apply our algorithm to the boundary integral solution of high frequency acoustic scattering applications. Finally in section 6, we provide some comments on future research directions.

## 2. Directional separated approximations

**DEFINITION 2.1.** Let  $f(x, y)$  be a function for  $x \in X$  and  $y \in Y$ . We say that  $f(x, y)$  has a  $t$ -term  $\varepsilon$ -expansion for  $X$  and  $Y$  if there exist functions  $\{\alpha_s(x)\}_{1 \leq s \leq t}$  and  $\{\beta_s(y)\}_{1 \leq s \leq t}$  such that

$$\left| f(x, y) - \sum_{s=1}^t \alpha_s(x) \beta_s(y) \right| \leq \varepsilon, \quad \forall x \in X, \forall y \in Y.$$

Since the two sets of functions  $\{\alpha_s(x)\}_{1 \leq s \leq t}$  and  $\{\beta_s(y)\}_{1 \leq s \leq t}$  depend only on  $x$  and  $y$  respectively, the above expansion is called *separated*. Suppose  $r \geq \sqrt{2}$ . For our problem, we consider

$$Y = B(0, r) \quad \text{and} \quad X = \{x : \theta(x, \ell) \leq 1/r, |x| \geq r^2\} \quad (2.1)$$

where  $\ell$  is a given unit vector and  $\theta(x, \ell)$  is the spanning angle between vectors  $x$  and  $\ell$ . The geometric relationship between  $Y$  and  $X$  is illustrated in figure 1.1. The following theorem serves as the theoretical foundation of our approach.

**THEOREM 2.2.** *For any  $\varepsilon > 0$  there exists a number  $t_\varepsilon$  that is independent of  $r$  such that*

$$G(x, y) = \frac{i}{4} H_0^{(1)}(2\pi|x - y|)$$

has a  $t_\varepsilon$ -term  $\varepsilon$ -expansion for any  $X$  and  $Y$  given by (2.1).

One way to prove this theorem is to follow the proof of the 3D Helmholtz kernel in [18]. However, such an argument does not result an optimal estimate for the separation rank  $t_\varepsilon$ , since it only utilizes the smoothness of the kernel and ignores the analytical properties of the Hankel function. Instead, we quote here a sharp result by V. Rokhlin in [33], which gives the optimal estimate  $t_\varepsilon = O(\log(1/\varepsilon))$ . In fact, Thm. 1.1 in [33] states a general result of the sparse diagonal forms for the translation operators of the 2D Helmholtz kernel: for any two regions  $X$  and  $Y$ ,

$$t_\varepsilon = O(\log(1/\varepsilon)) + O\left(\frac{r^2}{\rho}\right),$$

where  $\rho$  is the distance between  $X$  and  $Y$ . Specializing this result to our case  $\rho = O(r^2)$  proves Thm. 2.2. We refer to [33] for the details of the proof.

**2.1. Construction of directional separated approximations.** In [18], we described a procedure based on random sampling for constructing the directional separated representation between  $X$  and  $Y$ . Here, we propose an improved approach that results in better accuracy and stability.

For a given pair  $Y$  and  $X$  that satisfy the directional parabolic separation condition, we sample  $Y$  randomly with about 2 to 3 points per wavelength and denote the resulting samples by  $\{y_j\}_{1 \leq j \leq n}$ . Similarly,  $X \cap B(0, K)$  is sampled with a set of samples  $\{x_i\}_{1 \leq i \leq m}$  at the same rate. Let  $A$  be the  $m \times n$  matrix with the  $(i, j)$ th element equal to

$$G(x_i, y_j) = \frac{i}{4} H_0^{(1)}(2\pi|x_i - y_j|),$$

for  $1 \leq i \leq m$  and  $1 \leq j \leq n$ . For a given set of rows  $S$  and a set of columns  $T$ ,  $A(S, T)$  denotes the submatrix of  $A$  that consists of the entries  $A_{ij}$  with  $i \in S$  and  $j \in T$ . Moreover,  $A(:, T)$  denotes the submatrix that consists of the columns in  $T$  and  $A(S, :)$  is the submatrix that consists of the rows in  $S$ .

Thm. 2.2 states that  $A$  can be factorized, within error  $O(\varepsilon)$ , into the product of two matrices, with the first one having  $t_\varepsilon$  columns and the second one having  $t_\varepsilon$  rows. The goal of the new procedure is to construct the pseudoskeleton decomposition [20, 21] of the matrix  $A$ :

$$A \approx A(:, \Pi_1) \cdot D \cdot A(\Pi_2, :),$$

where  $\Pi_1$  is a set of  $t_\varepsilon$  columns,  $\Pi_2$  is a set of  $t_\varepsilon$  rows, and  $D \approx (A(\Pi_2, \Pi_1))^{-1}$ . The column set  $\Pi_1$  is chosen so that the parallelepiped spanned by the columns in  $\Pi_1$  has the largest  $t_\varepsilon$ -dimensional volume among all parallelepipeds spanned by a column set of size  $t_\varepsilon$ . Similarly, the parallelepiped spanned by the rows in  $\Pi_2$  has the largest  $t_\varepsilon$ -dimensional volume among all parallelepipeds spanned by a row set of size  $t_\varepsilon$ .

Since the size of the matrix  $m \times n$  can be quite large, choosing the sets  $\Pi_1$  and  $\Pi_2$  is a non-trivial task. One attractive idea that is proposed in the recent literature is to project the columns (or rows) of  $A$  onto a  $p$ -dimensional subspace  $L$  randomly selected from a certain probability measure over all  $p$ -dimensional subspaces (see for example [26, 35]). As long as  $p$  is adequately larger than  $t_\varepsilon$ , the volume spanned by any set of  $t_\varepsilon$  columns (or rows) is preserved up to a good accuracy ([12, 28]). Therefore, one approach for finding the column set  $\Pi_1$  would take the following steps

1. Project the columns of  $A$  onto a random  $p$  dimensional subspace.
2. Find the set of the projected columns that have the largest  $t_\varepsilon$ -dimensional volume.
3. Pick the corresponding columns of  $A$  to be the answer.

The same procedure can be used to pick  $\Pi_2$  if we project the rows of  $A$  instead. However, the complexity of this approach is at least  $O(mn \log p)$  [26, 35], which can still be quite expensive in our setting since both  $m$  and  $n$  are very large. Here, we describe a new procedure that replaces the general random projection with a projection to a randomly selected set of coordinates and finds the sets  $\Pi_1$  and  $\Pi_2$  in only  $O(t_\varepsilon^2 \cdot (m+n))$  operations. This procedure contains the following steps:

1. *Compute the column set  $\Pi_1$ .* As we mentioned already, finding these columns directly is quite expensive since the size of the  $m \times n$  matrix  $A$  is quite large. In order to achieve that efficiently, we first randomly pick a set of rows  $\Omega_1$  with  $|\Omega_1| = O(t_\varepsilon)$  and seek a set of  $t_\varepsilon$  columns of the submatrix  $A(\Omega_1, :)$  whose parallelepiped has almost the largest possible  $t_\varepsilon$ -dimensional volume. To do this, we simply use the pivoted QR factorization to obtain the decomposition

$$A(\Omega_1, \Lambda_1) = Q_1 R_1.$$

Here,  $\Lambda_1$  is a permutation of the  $n$  columns,  $Q_1$  is a unitary matrix, and  $R_1$  is an upper triangular matrix with positive diagonal entries in decreasing order. The construction of this decomposition takes at most  $O(|\Omega_1|^2 \cdot n) = O(t_\varepsilon^2 \cdot n)$  steps. We then set  $\Pi_1$  to be the first  $t_\varepsilon$  members of  $\Lambda_1$ . In practice, we observe that the volume of the parallelepiped spanned by the vectors in  $A(:, \Pi_1)$  is quite close to the largest possible volume. We shall also use  $\{e_t\}_{1 \leq t \leq t_\varepsilon}$  to denote  $\{y_j\}_{j \in \Pi_1}$ , i.e. the locations associated with the columns in  $\Pi_1$  (see figure 2.1).

2. *Compute the row set  $\Pi_2$ .* Similar to the previous step, we first randomly pick a set of columns  $\Omega_2$  with  $|\Omega_2| = O(t_\varepsilon)$  and seek a set of  $t_\varepsilon$  rows of the submatrix  $A(:, \Omega_2)$  that has almost the largest possible  $t_\varepsilon$ -dimensional volume. Applying the pivoted QR factorization to  $(A(:, \Omega_2))^*$  gives the following decomposition

$$A(\Lambda_2, \Omega_2) = R_2^* Q_2^*$$

where  $\Lambda_2$  is a permutation of the  $m$  rows. Similar to the previous step, this decomposition takes at most  $O(|\Omega_2|^2 \cdot m) = O(t_\varepsilon^2 \cdot m)$  steps. We then set  $\Pi_2$  to be the first  $t_\varepsilon$  members of  $\Lambda_2$ . In practice, the volume of the parallelepiped spanned by the vectors in  $A(\Pi_2, :)$  is quite close to the largest possible volume. We shall use  $\{c_s\}_{1 \leq s \leq t_\varepsilon}$  to denote  $\{x_i\}_{i \in \Pi_2}$ , i.e., the locations associated with the columns in  $\Pi_2$  (see figure 2.1).

3. *Compute the matrix  $D$ .* Here we seek a matrix  $D$  that minimizes

$$\|A - A(:, \Pi_1) \cdot D \cdot A(\Pi_2, :)\|_F.$$

A direct formula for computing  $D$  is given by

$$D = (A(:, \Pi_1))^+ \cdot A \cdot (A(\Pi_2, :))^+$$

where  $(\cdot)^+$  stands for the pseudo-inverse. However, computing  $D$  directly using this formula can be quite costly since  $A$  is a large  $m \times n$  matrix. Instead, we restrict the problem again to a random subspace. We pick randomly a set of rows  $\Gamma_1$  and a set of columns  $\Gamma_2$ , both of size  $O(t_\varepsilon)$ , and choose  $D$  to be the minimizer of

$$\|A(\Gamma_1, \Gamma_2) - A(\Gamma_1, \Pi_1) \cdot D \cdot A(\Pi_2, \Gamma_2)\|_F$$

instead. The resulting  $D$  is given explicitly by

$$D = (A(\Gamma_1, \Pi_1))^+ \cdot A(\Gamma_1, \Gamma_2) \cdot (A(\Pi_2, \Gamma_2))^+.$$

Notice that the matrix  $D$  is of size  $t_\varepsilon \times t_\varepsilon$ . In practice, we observe that

$$|A_{ij} - (A(:, \Pi_1) \cdot D \cdot A(\Pi_2, :))_{ij}| = O(\varepsilon), \quad 1 \leq i \leq m, 1 \leq j \leq n. \quad (2.2)$$

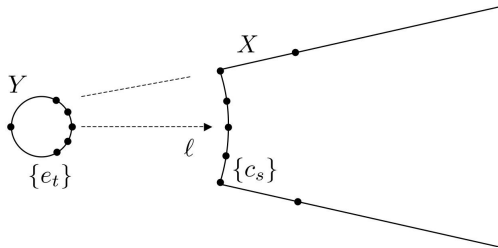


FIG. 2.1. The construction of the separated approximation between  $X$  and  $Y$ . The  $\{e_t\}$  are the samples associated with the columns in  $A(:, \Pi_1)$  (Step 1). The  $\{c_s\}$  are the samples associated with the columns in  $A(\Pi_2, :)$  (Step 2).

This randomized procedure performs quite well in practice as we will see from the numerical results in section 4. In the above description, the sizes of the random index sets  $\Omega_1$ ,  $\Omega_2$ ,  $\Gamma_1$ , and  $\Gamma_2$  are all equal to a constant multiple of  $t_\varepsilon$ . When one increases these numbers, the resulting pseudoskeleton approximation (2.2) becomes more and more accurate. In practice, we find that a factor of 4 to 6 is sufficient to provide good results. Compared with the approach described at the beginning of this section, the projection to a randomly selected coordinates is more restrictive compared to a general random projection. However, since both the columns and the rows of our matrix  $A$  are highly incoherent with the Dirac delta functions, a projection to a randomly selected set of coordinates preserves the  $t_\varepsilon$ -dimensional volume well with an overwhelming probability.

We would like to emphasize that there has been a lot of research devoted to the construction of low rank approximations from both the numerical linear algebra community and the theoretical computer science community (see [17, 29, 26, 35, 16, 23] for details). Many of these approaches have rigorous error estimates. However, for the best of our knowledge, in the situation where the desired accuracy  $\varepsilon$  is small all of these methods require an complexity that is at least  $O(mn)$ . On the other hand, the procedure proposed here takes only  $O(t_\varepsilon^2 \cdot (m+n))$  operations.

By denoting the entries of the matrix  $D$  by  $(d_{ts})_{1 \leq s, t \leq t_\varepsilon}$ , we can rewrite (2.2) as

$$\left| G(x_i, y_j) - \sum_{t=1}^{t_\varepsilon} G(x_i, e_t) \left( \sum_{s=1}^{t_\varepsilon} d_{ts} G(c_s, y_j) \right) \right| = O(\varepsilon)$$

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

$$\left| G(x, y) - \sum_{t=1}^{t_\varepsilon} G(x, e_t) \left( \sum_{s=1}^{t_\varepsilon} d_{ts} G(c_s, y) \right) \right| = O(\varepsilon) \quad (2.3)$$

for any  $x \in X \cap B(0, K)$  and  $y \in Y$ . In order to represent such a low rank approximation, one only needs to store  $\{c_s\}_{1 \leq s \leq t_\varepsilon}$ ,  $\{e_t\}_{1 \leq t \leq t_\varepsilon}$ , and the matrix  $D = (d_{ts})_{1 \leq s, t \leq t_\varepsilon}$ . Since all of them together takes at most  $O(t_\varepsilon^2)$  storage space, the storage requirement of this approximation is extremely efficient.

### 3. Algorithm description

The structure of this algorithm is similar but more complicated when compared to the one of the HF-FMM. Without loss of generality, we assume that the size of the domain is  $K = 2^{2L}$  for a positive integer  $L$ .

We start by constructing a quadtree that contains the points  $\{p_i\}_{1 \leq i \leq N}$ . We shall use  $B$  to denote a square in the quadtree and  $w$  for its width. A square  $B$  is said to be in the low frequency regime if  $w < 1$  and in the high frequency regime if  $w \geq 1$ . In the high frequency regime of the quadtree, no adaptivity is used, i.e., every non-empty square is further partitioned until the width of the square is less than 1. In the low frequency regime, a square  $B$  is partitioned as long as the number of points in  $B$  is greater than a fixed constant  $N_p$ . The value of  $N_p$  is chosen to optimize the computational complexity and, in practice, we pick  $N_p = 20$ .

#### 3.1. Compact representations.

**3.1.1. Low frequency regime.** For a square  $B$  in the low frequency regime, we follow the description of the kernel independent FMM in [38]. The near field  $N^B$  is the union of the squares that are on the same level as  $B$  and adjacent to  $B$ . The far field  $F^B$  is the complement of  $N^B$ . The interaction list  $I^B$  contains the squares in  $N^P \setminus N^B$  on  $B$ 's level, where  $P$  is the parent square of  $B$ . There are two important compact representations associated with each square  $B$ :

- $\{f_t^B\}_t$  are the *outgoing equivalent sources* located at a set of points  $\{e_t^B\}_t \subset B$ .  $\{f_t^B\}_t$  encode the potential in  $F^B$  generated by the points in  $B$  and play the role of the multipole expansion in the classical FMM.
- $\{u_t^B\}_t$  are the *incoming check potentials* located at the same points  $\{e_t^B\}_t \subset B$ .  $\{u_t^B\}_t$  encode the potential in  $B$  generated by the points in  $F^B$  and play the role of the local expansion in the classical FMM.

**3.1.2. High frequency regime.** For a square  $B$  in the high frequency regime, the near field  $N^B$  is the union of the squares  $\{A\}$  that satisfy  $\text{dist}(A, B) \leq w^2$ . The far field  $F^B$  is the complement of  $N^B$ . The interaction list  $I^B$  contains the squares in  $N^P \setminus N^B$  on  $B$ 's level, where  $P$  is  $B$ 's parent square. Notice that the far field of a square  $B$  in the high frequency regime is pushed away in order to be compatible with the directional parabolic separation condition. In order to take advantage of the directional separated approximations discussed in the previous section, the far field



$F^B$  is further partitioned into a group of directional wedges, each belonging to a cone with spanning angle  $O(1/w)$ . We denote the set of the wedges of  $B$  by  $\{W^{B,\ell}\}$ . figure 3.1 illustrates the angular decompositions for  $w = 1, 2, \text{ and } 4$ .

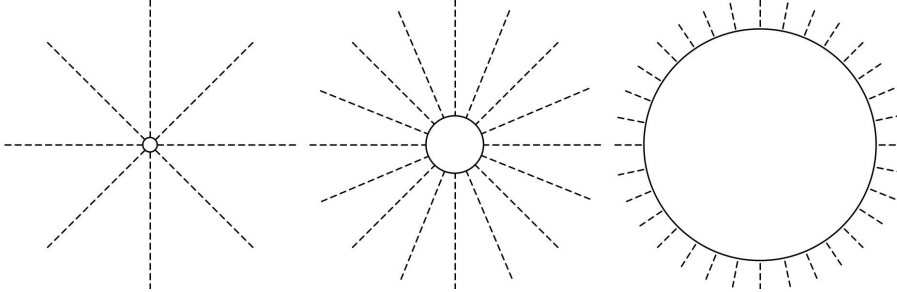


FIG. 3.1. The far field is partitioned into wedges. From left to right,  $w = 1, 2, 4$ . The radii are  $1, 4$ , and  $16$ , respectively.

Let us consider the interaction between a square  $B$  of width  $w$  and one of its wedges  $W^{B,\ell}$  using the separated approximations constructed in section 2 with  $Y = B$  and  $X = W^{B,\ell}$ . In the following discussion, we write the quantities  $\{e_t\}_{1 \leq t \leq t_\varepsilon}$ ,  $\{c_s\}_{1 \leq s \leq t_\varepsilon}$ , and  $D = (d_{ts})_{1 \leq t, s \leq t_\varepsilon}$  with superscript  $(\ )^{B,\ell}$  to denote their dependence on  $B$  and  $\ell$ . However, for a fixed width  $w$  and a fixed direction  $\ell$ , the relative positions of  $B$  and  $W^{B,\ell}$  remains the same. Therefore, the relative positions of  $\{e_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  and  $\{c_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  respect to  $B$ 's center are independent of the absolute position of  $B$ . Similarly, the matrix  $D^{B,\ell}$  only depends on  $w$  and  $\ell$ .

We first consider the potentials in  $W^{B,\ell}$  generated by the points in  $B$ . Applying (2.3) to  $y = p_i$  for each  $p_i \in B$  and summing them up with weight  $f_i$  gives the following estimate:

$$\left| \sum_{p_i \in B} G(x, p_i) f_i - \sum_{t=1}^{t_\varepsilon} G(x, e_t^{B,\ell}) \left( \sum_{s=1}^{t_\varepsilon} d_{ts}^{B,\ell} \left( \sum_{p_i \in B} G(c_s^{B,\ell}, p_i) f_i \right) \right) \right| \leq \left( \sum_{p_i \in B} |f_i| \right) \cdot \varepsilon.$$

This implies that we can place a set of sources

$$\left\{ f_t^{B,\ell} := \sum_{s=1}^{t_\varepsilon} d_{ts}^{B,\ell} \left( \sum_{p_i \in B} G(c_s^{B,\ell}, p_i) f_i \right) \right\}_{1 \leq t \leq t_\varepsilon} \tag{3.1}$$

at points  $\{e_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  to approximate the potential generated by the sources  $\{f_i\}$  located at points  $\{p_i \in B\}$ . Conceptually, the sources  $\{f_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  encode the potential in  $W^{B,\ell}$  generated by the points in  $B$  and we call them the *directional outgoing equivalent sources* of  $B$  in direction  $\ell$ . It is clear from (3.1) that  $\{f_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  can be computed simply from kernel evaluations and a matrix multiplication with  $D^{B,\ell} = (d_{ts}^{B,\ell})_{1 \leq t, s \leq t_\varepsilon}$ .

Let us now reverse the situation and consider the potential in  $B$  generated by the points  $\{p_i \in W^{B,\ell}\}$ . Since  $G(x, y) = G(y, x)$ , summing over (2.3) with  $x = p_i \in W^{B,\ell}$

and weights  $f_i$  leads us to the estimate

$$\begin{aligned} & \left| \sum_{p_i \in W^{B,\ell}} G(y, p_i) f_i - \sum_{s=1}^{t_\varepsilon} G(y, c_s^{B,\ell}) \left( \sum_{t=1}^{t_\varepsilon} d_{ts}^{B,\ell} \left( \sum_{p_i \in W^{B,\ell}} G(e_t^{B,\ell}, p_i) f_i \right) \right) \right| \\ & \leq \left( \sum_{p_i \in W^{B,\ell}} |f_i| \right) \cdot \varepsilon. \end{aligned}$$

This means that, once the potentials

$$\left\{ u_t^{B,\ell} := \sum_{p_i \in W^{B,\ell}} G(e_t^{B,\ell}, p_i) f_i \right\}_{1 \leq t \leq t_\varepsilon} \quad (3.2)$$

at points  $\{e_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  are given, we can then approximate at any  $y \in B$  the potential generated by the points in  $W^{B,\ell}$ , through a matrix multiplication with  $D^{B,\ell} = (d_{ts}^{B,\ell})$  and kernel evaluations. Conceptually, the potentials  $\{u_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  encode the potential in  $B$  generated by the points in  $W^{B,\ell}$  and we call them the *directional incoming check potentials* of  $B$  in direction  $\ell$ .

In the high frequency regime, the directional outgoing equivalent sources  $\{f_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  will play the role of the multipole expansion in classical FMM and the directional incoming check potentials  $\{u_t^{B,\ell}\}_{1 \leq t \leq t_\varepsilon}$  will play the role of the local expansion in classical FMM. However, it is important to notice that these quantities now depend on  $\ell$  and thus vary from one wedge to another.

**3.2. Translation operators.** An important component of our algorithm is the translation operators that transform the compact representations introduced above. Though no multipole or local expansions are involved in our algorithm, we follow the convention in [22, 32] and name these translation operators the M2M, L2L, and L2L translations. The translation operators for squares in the low frequency regime are detailed already in [38]. The translation operators in the high frequency regime are slightly more complicated and the main reason is that the computations are now directional.

**3.2.1. M2M translation.** For a square  $B$  and a direction  $\ell$ , the *M2M translation* constructs the outgoing directional equivalent sources  $\{f_t^{B,\ell}\}_t$  from the equivalent sources of  $B$ 's children. There are two cases to consider. In the first case,  $w=1$  and the children squares  $\{B_c\}_c$  have only non-directional equivalent sources. The M2M translation simply takes the equivalent sources  $\{f_t^{B_c}\}_{t,c}$  of these children squares as the true sources and performs the following step similar to (3.1):

$$f_t^{B,\ell} \Leftarrow \sum_s d_{ts}^{B,\ell} \left( \sum_c \sum_{t'} G(c_s^{B,\ell}, e_{t'}^{B_c}) f_{t'}^{B_c} \right). \quad (3.3)$$

In the second case,  $w > 1$ . Now the children squares have directional equivalent sources as well. Due to the nested structure of the wedges between  $B$  and  $B_c$ , for each  $\ell$  there exists a direction  $\ell'$  such that  $W^{B,\ell}$  is contained in  $W^{B_c,\ell'}$  for each child  $B_c$  of  $B$  (see figure 3.2). The M2M translation then takes the equivalent sources  $\{f_t^{B_c,\ell'}\}_{t,c}$  of the children squares as the true source and performs the following step similar to (3.1)

$$f_t^{B,\ell} \Leftarrow \sum_s d_{ts}^{B,\ell} \left( \sum_c \sum_{t'} G(c_s^{B,\ell}, e_{t'}^{B_c,\ell'}) f_{t'}^{B_c,\ell'} \right). \quad (3.4)$$

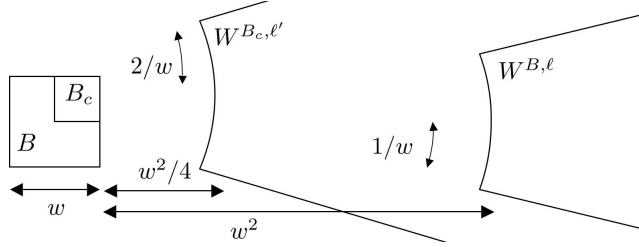


FIG. 3.2.  $B$  is a square with width  $w > 1$ . For any fixed  $\ell$ , there exists  $\ell'$  such that  $W^{B, \ell}$  is contained in  $W^{C, \ell'}$  where  $C$  is any one of  $B$ 's children.

**3.2.2. L2L translation.** For a square  $B$  and a direction  $\ell$ , the *L2L translation* constructs the incoming check potentials of  $B$ 's children from the incoming directional check potentials  $\{u_t^{B, \ell}\}_t$  of  $B$ . Again there are two cases to consider. In the first case  $w = 1$  and the children squares have only non-directional check potentials  $\{u_t^{B_c}\}_t$ . The L2L translation then regards  $\{u_t^{B, \ell}\}_t$  as the true potentials generated by the points in  $W^{B, \ell}$  and performs the following step:

$$u_t^{B_c} \leftarrow u_t^{B_c} + \sum_s G(e_t^{B_c}, c_s^{B, \ell}) \left( \sum_{t'} d_{t's}^{B, \ell} u_{t'}^{B, \ell} \right). \tag{3.5}$$

In the second case,  $w > 1$ . Now the children squares have directional check potentials. If  $W^{B_c, \ell'}$  is the wedge of  $B_c$  that contains  $W^{B, \ell}$ , the L2L translation then constructs the check potentials  $\{u_t^{B_c, \ell'}\}_t$  through the following step:

$$u_t^{B_c, \ell'} \leftarrow u_t^{B_c, \ell'} + \sum_s G(e_t^{B_c, \ell'}, c_s^{B, \ell}) \left( \sum_{t'} d_{t's}^{B, \ell} u_{t'}^{B, \ell} \right). \tag{3.6}$$

**3.2.3. M2L translation.** Finally, the *M2L translation* is applied to all pairs of squares  $A$  and  $B$  that are in each other's interaction list. Suppose that  $B$  is in the wedge  $W^{A, \ell'}$  of  $A$  while  $A$  is in the wedge  $W^{B, \ell}$  of  $B$ . The implementation of the M2L translation contains only one step:

$$u_t^{B, \ell} \leftarrow u_t^{B, \ell} + \sum_{t'} G(e_t^{B, \ell}, e_{t'}^{A, \ell'}) f_{t'}^{A, \ell'}. \tag{3.7}$$

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.

**3.3. Algorithm summary.** Now we are ready to give the overall structure of our new algorithm. An illustration of the various components of the algorithm is given in figure 3.3.

1. Construct the quadtree. In the high frequency regime, each non-empty square is partitioned dyadically. In the low frequency regime, the tree is constructed adaptively and each leaf square contains at most  $N_p$  points.

2. Travel up in the quadtree and visit the squares in the low frequency regime. These squares have width less than 1. For each square  $B$ , compute its outgoing non-directional equivalent sources  $\{f_t^B\}_t$  using the low frequency non-directional M2M translation.
3. Travel up in the quadtree and visit the squares in the high frequency regime. For every such square  $B$  and each  $\ell$ , use the high frequency directional M2M translation (3.3) or (3.4) to construct the directional equivalent sources  $\{f_t^{B,\ell}\}_t$ . We skip the squares with width greater than  $\sqrt{K}$  since their interaction lists are empty.
4. Travel down in the quadtree and visit the squares in the high frequency regime. For every such square  $B$  and for each direction  $\ell$ , perform the following two steps:
  - (a) For each square  $A$  that is in  $W^{B,\ell}$  and also in  $B$ 's interaction list, perform the high frequency directional M2L translation (3.7).
  - (b) Perform the high-frequency directional L2L translation (3.5) or (3.6) to transform  $\{u_t^{B,\ell}\}_t$  to the incoming check potentials for  $B$ 's children. Again, we skip the squares with width greater than  $\sqrt{K}$ .
5. Travel down in the quadtree. For every square  $B$  in the low frequency regime, we perform the following two steps:
  - (a) For each square  $A$  that is in  $B$ 's interaction list, use the low frequency non-directional M2L translation to update  $B$ 's incoming non-directional check potentials  $\{u_t^B\}_t$ .
  - (b) Perform the low frequency non-directional L2L translation. Depending on whether  $B$  is a leaf square or not, add the result either to the incoming check potentials of  $B$ 's children or to the potentials at the original points inside  $B$ .
6. Nearby interaction. For each leaf square  $B$  and for each  $p_i \in B$ , we add to  $u_i$  the nearby interaction from the points  $p_j \in N^B$ .

The following theorem summarizes the complexity of the proposed algorithm.

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

The proof of this theorem follows closely the steps of Thm. 4.1 of [18]. The main step of the proof is the observation that, for any fixed  $w > 1$ , there are at most  $O(K/w)$  squares of size  $w$  and, for each of them, there are at most  $O(w)$  squares to which we apply the M2L operator.

#### 4. Numerical results

In this section, we report some numerical results to illustrate the properties of our multidirectional algorithm. All of the computational results below are obtained on a desktop computer with a 2.8 GHz CPU.

Let us first study the performance of the randomized procedure presented in section 2. In Table 4.1, we list the number of terms in the separated approximation for two sets  $X$  and  $Y$  for different choices of accuracy  $\varepsilon$  and square width  $w$ . Here  $r$ , the radius of  $Y$ , is set to be  $\sqrt{2}w$  so that the square of width  $w$  is contained in  $Y$ .

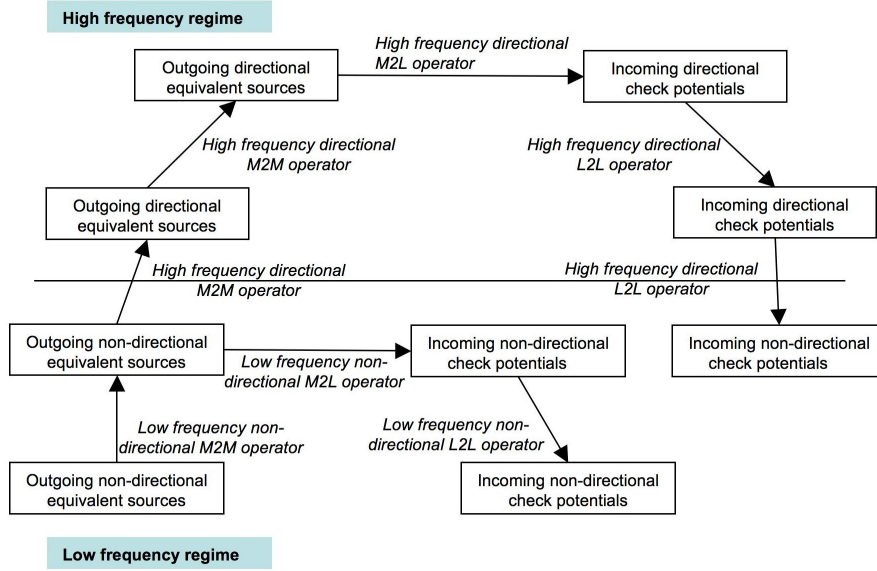


FIG. 3.3. A small part of the quadtree used in the computation. Each rectangular region stands for a square of the quadtree. The diagram shows how the outgoing non-directional equivalent sources from a leaf square have been transformed into incoming non-directional check potentials at other leaf squares. Far field interaction involves directional computation in the high frequency regime.

We can see from Table 4.1 that the separation rank is bounded by a constant which is independent of the values of  $w$ . This is consistent with Thm. 2.2. In fact it seems that as  $w$  grows the separation rank decays slightly.

	$w = 1$	$w = 2$	$w = 4$	$w = 8$	$w = 16$	$w = 32$	$w = 64$	$w = 128$
$\varepsilon = 1e-4$	14	11	11	10	9	9	9	9
$\varepsilon = 1e-6$	19	16	14	13	12	12	12	11
$\varepsilon = 1e-8$	27	20	16	15	15	15	14	14

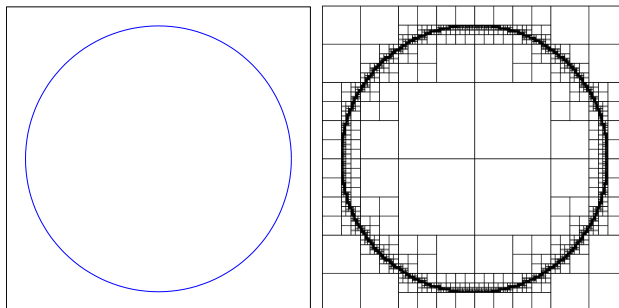
TABLE 4.1. The separation rank of the directional separated approximation for different choices of requested accuracy  $\varepsilon$  and square size  $w$ .

Next, we applied our algorithm to the  $N$ -body problems on several objects. In our experiments, the boundary of each object is represented by a piecewise smooth curve. For these tests, the points  $\{p_i\}_{1 \leq i \leq N}$  sample the boundary curve randomly with about 20 points per wavelength. The sources  $\{f_i\}_{1 \leq i \leq N}$  are generated from a random distribution with mean 0. Suppose that we use  $\{u_i\}_{1 \leq i \leq N}$  to denote the true discrete potentials and  $\{u_i^a\}_{1 \leq i \leq N}$  to denote the approximations obtained through our algorithm. In order to estimate the relative error, we pick a set  $S$  of 200 points from  $\{p_i\}_{1 \leq i \leq N}$  and compute the accuracy with

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

where the true potentials  $\{u_i\}_{i \in S}$  are computed by using direct evaluation.

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

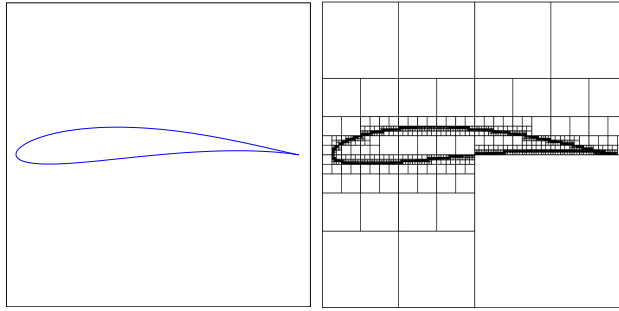


$(K, \varepsilon)$	$N$	$T_a(\text{sec})$	$T_d(\text{sec})$	$T_d/T_a$	$\varepsilon_a$
(2048,1e-4)	1.13e+5	3.40e+1	8.05e+3	2.37e+2	1.25e-4
(8192,1e-4)	4.50e+5	1.56e+2	1.28e+5	8.21e+2	1.31e-4
(32768,1e-4)	1.80e+6	7.07e+2	2.06e+6	2.91e+3	1.80e-4
(2048,1e-6)	1.13e+5	5.30e+1	8.00e+3	1.51e+2	7.88e-7
(8192,1e-6)	4.50e+5	2.39e+2	1.28e+5	5.37e+2	9.98e-7
(32768,1e-6)	1.80e+6	1.08e+3	2.06e+6	1.91e+3	1.00e-6
(2048,1e-8)	1.13e+5	8.20e+1	8.05e+3	9.82e+1	8.48e-9
(8192,1e-8)	4.50e+5	3.57e+2	1.29e+5	3.60e+2	1.18e-8
(32768,1e-8)	1.80e+6	1.58e+3	2.07e+6	1.31e+3	1.30e-8

TABLE 4.2. Results of a circle with the Helmholtz kernel.  $N$  is the number of points,  $K$  is the size of the problem in terms of the wavelength,  $\varepsilon$  is the prescribed error threshold such that the final error is to be bounded by a constant multiple of  $\varepsilon$ ,  $T_a$  is the running time of our algorithm in seconds,  $T_d$  is the estimated running time of the direct evaluation in seconds,  $T_d/T_a$  is the speedup factor, and  $\varepsilon_a$  is the estimated error of our algorithm.

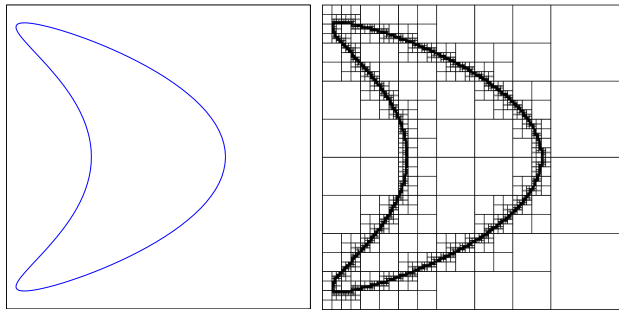
The first example is a circle and the results are summarized in Table 4.2. The second example is an airfoil and the results are shown in Table 4.3. The final example is a kite-shaped object and we report the numbers in Table 4.4. These numbers demonstrate clearly that our algorithm scales exactly like  $O(N \log N)$  in terms of the number of points. Furthermore, the error seems to grow only slightly as we increase the number of points, indicating that the separated approximations are stable.

Compared with the results presented in [8], our algorithm is slower by a factor of 7 to 8. The reason is that our multidirectional algorithm heavily uses the kernel evaluation formula. Since the 2D Helmholtz kernel is a Hankel function, its computational procedure is rather slow. On the other hand, all of the high frequency translations in [8] are precomputed and stored in the diagonal form and no special function evaluation is required during the computation.



$(K, \varepsilon)$	$N$	$T_a(\text{sec})$	$T_d(\text{sec})$	$T_d/T_a$	$\varepsilon_a$
(2048,1e-4)	7.82e+4	2.00e+1	3.87e+3	1.94e+2	1.15e-4
(8192,1e-4)	3.13e+5	8.80e+1	6.17e+4	7.02e+2	1.21e-4
(32768,1e-4)	1.25e+6	3.90e+2	9.90e+5	2.54e+3	1.07e-4
(2048,1e-6)	7.82e+4	3.20e+1	3.87e+3	1.21e+2	1.04e-6
(8192,1e-6)	3.13e+5	1.38e+2	6.20e+4	4.50e+2	9.65e-7
(32768,1e-6)	1.25e+6	6.05e+2	1.01e+6	1.67e+3	1.20e-6
(2048,1e-8)	7.82e+4	4.70e+1	3.87e+3	8.24e+1	8.58e-9
(8192,1e-8)	3.13e+5	2.03e+2	6.22e+4	3.06e+2	1.69e-8
(32768,1e-8)	1.25e+6	8.78e+2	9.95e+5	1.13e+3	1.33e-8

TABLE 4.3. Results of an airfoil with the Helmholtz kernel.



$(K, \varepsilon)$	$N$	$T_a(\text{sec})$	$T_d(\text{sec})$	$T_d/T_a$	$\varepsilon_a$
(2048,1e-4)	1.13e+5	4.00e+1	8.11e+3	2.03e+2	1.08e-4
(8192,1e-4)	4.53e+5	1.77e+2	1.30e+5	7.36e+2	1.33e-4
(32768,1e-4)	1.81e+6	8.04e+2	2.09e+6	2.60e+3	1.41e-4
(2048,1e-6)	1.13e+5	6.10e+1	8.11e+3	1.33e+2	9.35e-7
(8192,1e-6)	4.53e+5	2.72e+2	1.30e+5	4.78e+2	9.15e-7
(32768,1e-6)	1.81e+6	1.24e+3	2.10e+6	1.70e+3	8.80e-7
(2048,1e-8)	1.13e+5	9.20e+1	8.16e+3	8.87e+1	1.45e-8
(8192,1e-8)	4.53e+5	4.05e+2	1.30e+5	3.22e+2	1.31e-8
(32768,1e-8)	1.81e+6	1.80e+3	2.11e+6	1.17e+3	1.52e-8

TABLE 4.4. Results of a kite-shaped model with the Helmholtz kernel.

### 5. Acoustic scattering problems

In this section, we apply our algorithm to the solution of the BIE formulation

$$\frac{1}{2}\phi(x) + \int_{\partial D} \left( \frac{\partial G(x,y)}{\partial n(y)} - \eta \cdot G(x,y) \right) \phi(y) dy = -u^{inc}(x) \quad (5.1)$$

of the 2D scattering problem mentioned in section 1. Here, we report the numerical results for the smooth objects in Tables 4.2 and 4.4.

In our experiments, the incoming field  $u^{inc}(x)$  is equal to  $e^{2\pi i x \cdot d}$  with  $d=(1,0)$ . We set the coupling constant  $\eta=\pi$  and discretize (5.1) with the standard collocation method [10, 25] with piecewise linear basis functions. The boundary curve is discretized with about 15 points per wavelength. We use the standard Gaussian quadrature weights to integrate the interaction between two non-adjacent elements. The adjacent interaction has a logarithmic singularity due to the single layer part of the kernel and we integrate it with the generalized Gaussian quadrature rule described in [27]. The resulting discrete system is solved iteratively with the GMRES algorithm [34]. We choose a relative tolerance of 1e-3 and set the restarted number to be 80. Within each iteration of the GMRES solver, the application of the integral operator in (5.1) is accelerated using our multidirectional algorithm with a prescribed accuracy 1e-4.

Table 5.1 summarizes the numerical results for the circle with its size equal to 1024 to 8192 wavelengths. Here  $T_i$  is the averaged time of each iteration,  $N_i$  is the number of iterations, and  $T_t$  is the total time. Table 5.2 reports the results of the kite-shaped object shown in Table 4.4. From the tables, we observe that the number of iterations grows only slowly as we double the size of the problem each time.

$K$	$N$	$T_i(\text{sec})$	$N_i$	$T_t(\text{sec})$
1024	5.15e+04	3.10e+01	27	8.38e+02
2048	1.03e+05	6.40e+01	32	2.05e+03
4096	2.06e+05	1.42e+02	39	5.54e+03
8192	4.12e+05	3.82e+02	47	1.79e+04

TABLE 5.1. *Timings of computing the scattered field of the circle.  $K$  is the size of the problem in terms of the wavelength,  $N$  is the number of quadrature points,  $T_i$  is the averaged time of each iteration,  $N_i$  is the number of iterations, and  $T_t$  is the total time.*

$K$	$N$	$T_i(\text{sec})$	$N_i$	$T_t(\text{sec})$
1024	5.15e+04	3.10e+01	27	8.41e+02
2048	1.03e+05	6.60e+01	32	2.12e+03
4096	2.06e+05	1.47e+02	39	5.79e+03
8192	4.12e+05	4.45e+02	48	2.13e+04

TABLE 5.2. *Timings of computing the scattered field of the kite-shaped object.*

In figure 5.1, we display the scattered field of the kite-shaped object in a region that contains caustics. It clearly shows the geometric optics phenomenon, which is naturally expected when the wavelength is small compared to the size of the object. Here, the significance is that one can indeed bring the gap between the finite difference or finite element approach in the low frequency regime and the geometric



optics approach in the high frequency regime to address the scattering problems at all frequencies.

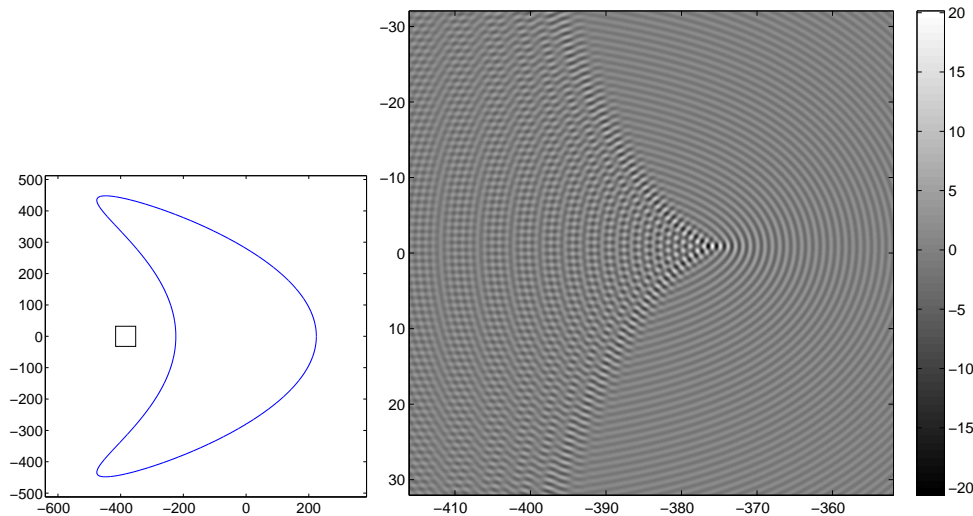


FIG. 5.1. Scattered field of the kite-shaped object with  $K=1024$ . Left: a square region that contains the caustics. Right: the real part of the scattered field inside the square. The field is sampled at 8 points per wavelength.

## 6. Conclusions

In this paper, we described a directional multiscale algorithm for computing the  $N$ -body problem for the high frequency Helmholtz kernel in two dimensions. The approach follows the framework described in [18]. By using the directional low rank representations for regions that follow the directional parabolic separation condition, our algorithm achieves the optimal  $O(N \log N)$  complexity. We have also introduced an improved randomized procedure for constructing the low rank separated approximations. The numerical results have shown that our algorithm is capable of addressing very large scale problems in high frequency scattering.

One direction of future research is to find a rigorous proof for the randomized procedure proposed in section 2. Another interesting direction for future research is to apply this kind of directional multiscale idea to other problems with oscillatory behavior, in both two and three dimensions. One typical example is the computation of the far field pattern of a scattered field [10, 37].

**Acknowledgement.** Engquist is partially supported by an NSF grant DMS 0714612 and a startup grant from the University of Texas at Austin. Ying is partially supported by an Alfred P. Sloan Research Fellowship and a startup grant from the University of Texas at Austin. The authors would like to thank P.G. Martinsson for helpful discussions.

## REFERENCES

- [1] 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(1), 19–53, 2000.

- [2] E. Bleszynski, M. Bleszynski and T. Jaroszewicz, *AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems*, Radio Science, 31, 1225–1252, 1996.
- [3] A. Boag, E. Michielssen and A. Brandt, *Nonuniform polar grid algorithm for fast field evaluation*, Antennas and Wireless Propagation Letters, IEEE, 1, 142–145, 2002.
- [4] B. Bradie, R. Coifman and A. Grossmann, *Fast numerical computations of oscillatory integrals related to acoustic scattering. I*, Appl. Comput. Harmon. Anal., 1(1), 94–99, 1993.
- [5] A. Brandt, *Multilevel computations of integral transforms and particle interactions with oscillatory kernels*, Comput. Phys. Comm., 65(1-3), 24–38, 1991.
- [6] 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(1), 80–110, 2001.
- [7] F.X. Canning, *Sparse approximation for solving integral equations with oscillatory kernels*, SIAM J. Sci. Statist. Comput., 13(1), 71–87, 1992.
- [8] H. Cheng, W. Crutchfield, Z. Gimbutas, L. Greengard, J. Huang, V. Rokhlin, N. Yarvin and J. Zhao, *Remarks on the implementation of the wideband FMM for the Helmholtz equation in two dimensions*, Inverse problems, multi-scale analysis and effective medium theory, Contemp. Math., Amer. Math. Soc., Providence, RI, 408, 99–110, 2006.
- [9] W. 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.
- [10] D. Colton and R. Kress, *Inverse acoustic and electromagnetic scattering theory*, Applied Mathematical Sciences, Springer-Verlag, Berlin, second edition, 93, 1998.
- [11] E. Darve, *The fast multipole method: numerical implementation*, J. Comput. Phys., 160(1), 195–240, 2000.
- [12] S. Dasgupta and A. Gupta, *An elementary proof of a theorem of Johnson and Lindenstrauss*, Random Struct. Algorithms, 22(1), 60–65, 2003.
- [13] L. Demanet and L. Ying, *Scattering in flatland: efficient representations via wave atoms*, Foundations of Computational Mathematics, to appear, 2009.
- [14] H. Deng and H. Ling, *Fast solution of electromagnetic integral equations using adaptive wavelet packet transform*, Antennas and Propagation, IEEE Transactions on, 47(4), 674–682, Apr 1999.
- [15] H. Deng and H. Ling, *On a class of predefined wavelet packet bases for efficient representation of electromagnetic integral equations*, Antennas and Propagation, IEEE Transactions on, 47(12), 1772–1779, Dec. 1999.
- [16] A. Deshpande and S. Vempala, *Adaptive sampling and fast low-rank matrix approximation*, Proc. 10th Internat. Workshop on Randomization and Computation (RANDOM '06), Springer, 292–303, 2006.
- [17] 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(1), 158–183, 2006.
- [18] B. Engquist and L. Ying, *Fast directional multilevel algorithms for oscillatory kernels*, SIAM J. Sci. Comput., 29(4), 1710–1737, 2007.
- [19] W. Golik, *Wavelet packets for fast solution of electromagnetic integral equations*, IEEE Transactions on Antennas and Propagation, 46(5), 618–624, May 1998.
- [20] S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261, 1–21, 1997.
- [21] S.A. Goreinov, N.L. Zamarashkin and E.E. Tyrtyshnikov, *Pseudoskeleton approximations by submatrices of greatest size*, XMat. Zametki, 62(4), 619–623, 1997.
- [22] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73(2), 325–348, 1987.
- [23] S. Har-Peled, *Low Rank Matrix Approximation in Linear Time*, Technical report, Manuscript, 2006.
- [24] D. Huybrechs and S. Vandewalle, *A two-dimensional wavelet-packet transform for matrix compression of integral equations with highly oscillatory kernel*, J. Comput. Appl. Math., 197(1), 218–232, 2006.
- [25] R. Kress, *Linear Integral Equations*, second edition, Applied Mathematical Sciences, Springer-Verlag, New York, 82, 1999.
- [26] E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin and M. Tygert, *Randomized algorithms for the low-rank approximation of matrices*, Proc. Natl. Acad. Sci. USA, 104(51), 20167–20172, 2007.
- [27] J. Ma, V. Rokhlin and S. Wandzura, *Generalized Gaussian quadrature rules for systems of arbitrary functions*, SIAM J. Numer. Anal., 33(3), 971–996, 1996.

- [28] A. Magen, *Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications*, J.D.P. Rolim and S.P. Vadhan (eds.), *RANDOM*, Lecture Notes in Computer Science, 2483, 239–253, Springer, 2002.
- [29] P.G. Martinsson, V. Rokhlin and M. Tygert, *A randomized algorithm for the approximation of matrices*, Technical report, Yale University, 2006.
- [30] E. Michielssen and A. Boag, *A multilevel matrix decomposition algorithm for analyzing scattering from large structures*, IEEE Transactions on Antennas and Propagation, 44(8), 1086–1093, 1996.
- [31] M. O’Neil and V. Rokhlin, *A new class of analysis-based fast transforms*, Technical report, Yale University, YALE/DCS/TR1384, 2007.
- [32] V. Rokhlin, *Rapid solution of integral equations of scattering theory in two dimensions*, J. Comput. Phys., 86(2), 414–439, 1990.
- [33] V. Rokhlin, *Sparse diagonal forms for translation operators for the Helmholtz equation in two dimensions*, Appl. Comput. Harmon. Anal., 5(1), 36–67, 1998.
- [34] Y. Saad and M.H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7(3), 856–869, 1986.
- [35] T. Sarlos, *Improved approximation algorithms for large matrices via random projections*, Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS), 143–152, 2006.
- [36] 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(1), 15–19, 1995.
- [37] L. Ying, *Sparse Fourier transform via butterfly algorithm*, SIAM J. Sci. Comput., to appear, 2009.
- [38] L. Ying, G. Biros and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys., 196(2), 591–626, 2004.