MULTISCALE MODEL. SIMUL.
Vol. 13, No. 3, pp. 829–846

DIRECTIONAL PRECONDITIONER FOR 2D HIGH FREQUENCY OBSTACLE SCATTERING*

LEXING YING†

Abstract. The boundary integral method is an efficient approach for solving time-harmonic obstacle scattering problems from bounded scatterers. This paper presents the directional preconditioner for the linear systems of the boundary integral method in two dimensions. This new preconditioner builds a data-sparse approximation of the integral operator, transforms it into a sparse linear system, and computes an approximate inverse with efficient sparse linear algebra algorithms. This preconditioner is efficient and results in small and almost frequency-independent iteration counts for nonresonant scatterers when combined with standard iterative solvers. Numerical results are provided to demonstrate the effectiveness of the new preconditioner.

Key words. boundary integral method, scattering, high-frequency waves, preconditioner, low-rank approximation, sparse linear algebra

AMS subject classifications. 65N38, 65R20, 78A45

DOI. 10.1137/140985135

1. Introduction.

This paper is concerned with solving the time-harmonic acoustic obstacle scattering problems in two dimensions. Suppose that $\Omega \subset \mathrm{R}^2$ is a bounded scatterer with smooth boundary $\partial\Omega$. In order to normalize the problem size, we assume that both the diameter and the boundary length of Ω are of order 1. Let ω be the time frequency and $e^{i\omega t}u_I(x)$ be the time-harmonic incident wave. In the sound-soft scattering problem, the scattered field $u(x)$ satisfies the Helmholtz equation with the Dirichlet boundary condition

$$\begin{aligned} \Delta u(x) + \omega^2 u(x) &= 0, && x \in \mathrm{R}^2 \setminus \Omega, \\ u(x) &= -u_I(x), && x \in \partial\Omega. \end{aligned}$$

In the sound-hard scattering problem, the scattered field $u(x)$ satisfies the Helmholtz equation with the Neumann boundary condition

$$\begin{aligned} \Delta u(x) + \omega^2 u(x) &= 0, && x \in \mathrm{R}^2 \setminus \Omega, \\ \frac{\partial u(x)}{\partial n(x)} &= -\frac{\partial u_I(x)}{\partial n(x)}, && x \in \partial\Omega. \end{aligned}$$

In both cases, $u(x)$ satisfies the Sommerfeld radiation condition

$$\lim_{r\to\infty} r^{1/2}\left(\frac{\partial u}{\partial r} - i\omega u\right) = 0.$$

An effective way to solve these problems is the boundary integral method, which relies on the free space Green's function

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

*Received by the editors September 3, 2014; accepted for publication (in revised form) May 15, 2015; published electronically July 28, 2015. This work was partially supported by the National Science Foundation under award DMS-1328230 and by the U.S. Department of Energy's Advanced Scientific Computing Research program under award DE-FC02-13ER26134/DE-SC0009409.
http://www.siam.org/journals/mms/13-3/98513.html
†Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (lexing@stanford.edu).

of the Helmholtz equation. For the sound-soft scattering, the combined field integral equation (CFIE) [13, 28] looks for a surface density $q(x)$ on the boundary such that for each $x \in \partial\Omega$,

$$\frac{1}{2}q(x) + \int_{\partial\Omega} \frac{\partial G(x,y)}{\partial n(y)} q(y)dy - i\eta \int_{\partial\Omega} G(x,y)q(y)dy = -u_I(x). \tag{1}$$

Once $q(x)$ is computed, the scattered field $u(x)$ can be evaluated through a boundary integral over $\partial\Omega$. For the sound-hard scattering, one again looks for $q(x)$ on the boundary such that for each $x \in \partial\Omega$,

$$\frac{1}{2}q(x) - \int_{\partial\Omega} \frac{\partial G(x,y)}{\partial n(x)} q(y)dy + \frac{1}{i\eta} \oint_{\partial\Omega} \frac{\partial^2 G(x,y)}{\partial n(x)\partial n(y)} q(y)dy = -\frac{\partial u_I(x)}{\partial n(x)}, \tag{2}$$

with $\oint$ being the principal value integral. In both cases, η is typically chosen to be of order $O(\omega)$ (see [23], for example). We refer the reader to [13, 28] for derivations and discussions of these integral equations.

By introducing the operators

$$\begin{aligned}
(Sq)(x) &= \int_{\partial\Omega} G(x,y)q(y)dy, \\
(Dq)(x) &= \int_{\partial\Omega} \frac{\partial G(x,y)}{\partial n(y)} q(y)dy, \\
(D'q)(x) &= \int_{\partial\Omega} \frac{\partial G(x,y)}{\partial n(x)} q(y)dy, \\
(Nq)(x) &= \oint_{\partial\Omega} \frac{\partial^2 G(x,y)}{\partial n(x)\partial n(y)} q(y)dy,
\end{aligned}$$

we can rewrite (1) and (2) in operator forms:

$$\left(\frac{1}{2}I + D - i\eta S\right) q = -u_I, \tag{3}$$

$$\left(\frac{1}{2}I + D' - \frac{1}{i\eta}N\right) q = -\frac{\partial u_I}{\partial n}. \tag{4}$$

Standard approaches for discretizing these boundary integral equations include the Nyström method, the Galerkin method, and the collocation method [13, 24]. To simplify the presentation, we assume that the Nyström method is used. The discussion remains similar for the other approaches as long as the basis functions employed are local. A typical discretization of these integral equations requires at least a few quadrature points per wavelength. Since both the diameter and the boundary length of Ω are of order 1, this implies that the boundary is discretized with a set P of $n = \Theta(\omega)$ points. Here the notation $f = \Theta(g)$ means that $cf \le g \le c'f$ for some uniform constants c and c'. For the resulting linear systems, we continue to use S, D, D', and N to denote the discrete matrices associated with these operators. Similarly, q, u_I, and $\partial u_I/\partial n$ are reused to denote the discrete versions of $q(x)$, $u_I(x)$, and $\partial u_I(x)/\partial n(x)$ sampled at the quadrature points. With this slight abuse of notation, the discrete linear systems take the same form as (3) and (4).

There has been a lot of work devoted to the fast solution of these linear systems. Since the system is dense, the standard direct solvers such as the LU factorization

take $O(n^3) = O(\omega^3)$ steps, which is prohibitively expensive when ω is large. Recently, several linear-complexity approaches based on recursive interpolative decomposition have been proposed by [14, 26, 21] for boundary integral equations with nonoscillatory kernels. However, for high-frequency scattering, the kernel is oscillatory (i.e., $\omega = \Theta(n)$), and the computational complexity of these approaches is still cubic in n. The only exception is the algorithm in [27] for quasi-1D domains, where the boundary integral equation essentially reduces to the 1D case and the complexity scales linearly in ω.

For this reason, iterative methods such as GMRES and TFQMR [19, 30, 29] are the main tools for solving these problems. However, though the CFIEs have much improved conditioning properties compared to other integral formulations, the number of iterations can still grow quickly with ω. Therefore, there is a need for preconditioning the CFIEs for high-frequency problems.

Over the past twenty years, there has been a significant amount of research devoted to this task. A few proposed algorithms suggest improving the conditioning property via modifying the standard CFIE formulation. For example, one idea is to replace the $i\eta$ term in (3) with better approximations of the Dirichlet-to-Neumann (DtN) operator, and the $1/(i\eta)$ term in (4) with better approximations of the Neumann-to-Dirichlet (NtD) operator [1, 4, 5, 9]. Typically, these new approximations are derived from leading order terms of the pseudodifferential symbols of the DtN and NtD operators.

A second approach is to precondition the integral equation analytically. Most methods guided by this approach consider the electric field integral equation for electromagnetic scattering and follow the Calderon relationship [31, 11, 12, 2, 3]. The resulting integral equations are of Fredholm second kind with good conditioning properties. However, the number of matrix vector multiplications per iteration is doubled.

There has also been a lot of work on sparsifying the integral operators using special basis functions, such as local cosine bases [6, 8], optimized wavelet packets [17, 18, 20, 22], etc. The resulting sparse representations typically have $O(n^{4/3})$ nonzero entries. Recently in [16], an approach using the wave atom transform [15] generated a sparse representation with $O(n \log n)$ nonzero entries. In [10], Canning constructed a sparse approximation with $O(n)$ nonzero entries via locally mollified exponential functions. However, when good accuracy is required, most of these methods access all entries of the integral operator, thus requiring an $O(n^2)$ precomputation cost for assembling the whole matrix.

In this paper, we propose a new method for preconditioning the CFIEs by incorporating the ideas from sparse representation. This approach builds a data-sparse representation of the boundary integral operator, transforms it into a sparse linear system, and computes an approximate inverse with efficient sparse linear algebra algorithms. This preconditioner is highly efficient to construct and to apply. It results in small and almost frequency-independent iteration counts for nonresonant scatterers when combined with standard iterative solvers. The rest of the paper is organized as follows. Section 2 describes the algorithm. In section 3, we present the results. Future work and open questions are discussed in section 4.

2. Algorithm. For a fixed frequency ω, the wavelength λ is equal to $2\pi/\omega$. We assume that the scatterer boundary $\partial\Omega$ is C^2 with uniformly bounded curvature and that both the diameter and the boundary length of Ω are $\Theta(1)$. To simplify the discussion, the length L of $\partial\Omega$ is assumed to be equal to $4^q\lambda$, where q is a positive integer. The actual number 4^q is not essential but it makes the presentation simple.

Combining this with $L = \Theta(1)$ implies that $\omega = O(4^q)$.

Suppose that $\rho : \partial\Omega \to [0, L]$ is the arclength parameterization of the boundary and that the boundary is sampled uniformly with $n = 4^q p$ discretization points for some $p = \Theta(1)$, i.e., p points per wavelength λ.

2.1. Data-sparse approximation. We start by decomposing the boundary into sufficiently planar segments. Let κ be the maximum absolute value of the curvature on $\partial\Omega$, and ℓ be the largest integer such that $2^\ell \le 2^q/\sqrt{\kappa}$. The boundary is partitioned into $2^{2q-\ell}$ segments, each of which is of length $h := 2^\ell\lambda$ and has $2^\ell p$ points. We denote this set of segments by $\mathcal{G} = \{P_1, \ldots, P_m\}$, where $m = 2^{2q-\ell}$ and the segments P_i are ordered according to their positions on the boundary.

In this paper, we focus on the setting of large ω and fixed geometry Ω. As $\omega = O(4^q)$ grows, the segment length h scales as $O(2^q\lambda) = O(1/\sqrt{\omega})$, and the number of segments m grows like $O(2^q) = O(\sqrt{\omega})$. Notice that since the boundary is assumed to be C^2, all segments in $\mathcal{G}$ are almost-planar for sufficiently large ω.

The discussion here shall treat the sound-soft case (3) and the sound-hard case (4) in the same way, since their kernels share the same oscillatory pattern. Therefore, it is convenient to use the general form

$$Mq = f \tag{5}$$

for both (3) and (4). Based on how the segments are generated, each P_j is of length $2^\ell\lambda$ and contains $2^\ell p$ equally spaced discretization points. After ordering the unknowns according to the ordering of P_j, the matrix M can be written in the following block form:

$$M = \begin{bmatrix} M_{11} & \dots & M_{1m} \\ \vdots & \ddots & \vdots \\ M_{m1} & \dots & M_{mm} \end{bmatrix},$$

where each M_{ij} is of size $2^\ell p \times 2^\ell p$. The next step is to find data-sparse approximations for the blocks M_{ij}.

2.1.1. Diagonal blocks. Let us first consider a diagonal block M_{jj}, which represents the interaction between P_j and itself. Since P_j is of length $2^\ell\lambda$, the difference between the geometry of P_j and its linear approximation is bounded by

$$\frac{1}{2}(2^\ell\lambda)^2\kappa \lesssim \left(\frac{2^q\lambda}{\sqrt{\kappa}}\right)^2 \kappa = O(\lambda).$$

Therefore, one can treat P_j approximately as planar. Based on this observation, let us define B_j to be the matrix obtained by restricting the integral operator to a straight segment of length $2^\ell\lambda$ with $2^\ell p$ equally spaced quadrature points. The fact that P_j is almost planar implies that

$$M_{jj} \approx B_j.$$

Noticing that B_j depends only on the integer ℓ and that there are only a few choices for ℓ, we can clearly precompute these matrices. Going through all P_i gives the following approximation B to the block-diagonal part of M:

$$B = \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_m \end{bmatrix}.$$

2.1.2. Off-diagonal blocks. Next we consider the off-diagonal blocks M_{ij} with $i \neq j$. We define

- c_i and c_j to be the centers of segments P_i and P_j,
- t_i and t_j to be the unit tangent directions of $\partial\Omega$ at the centers of P_i and P_j, and
- a_{ij} to be the unit vector from c_j to c_i, i.e., $(c_i - c_j)/|c_i - c_j|$ (see Figure 1 for an illustration).

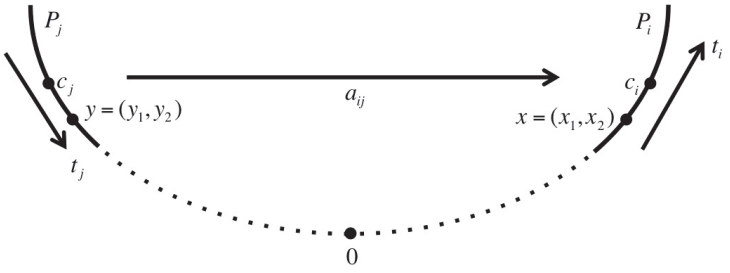

FIG. 1. *The geometric setup of two segments P_i and P_j.*

The main challenge here is that the high-frequency Helmholtz kernel $G(x, y)$ is highly oscillatory. Since the oscillation of $G(x, y)$ and its derivatives come from the term $\exp(i\omega|x - y|)$, it is instructive to focus on $\exp(i\omega|x - y|)$ for the moment. In the following discussion, the symbol $\sim$ is used to denote an approximation up to a nonoscillatory multiplicative term.

The first observation is that

$$\exp(i\omega|x - y|) \sim \exp(i\omega a_{ij} \cdot (x - y)) \tag{6}$$

for $x \in P_i$ and $y \in P_j$. To see this, we assume without loss of generality that $a_{ij} = (1, 0)^t$ by rotating the coordinates accordingly and placing the origin on $\partial\Omega$ between P_i and P_j (see Figure 1). In the new coordinate system, we have

$$\begin{aligned}\exp(i\omega|x - y|) &= \exp(i\omega(x_1 - y_1)) \exp(i\omega(|x - y| - (x_1 - y_1))) \\ &= \exp(i\omega(x_1 - y_1)) \exp\left(i\omega(x_1 - y_1)\left(\sqrt{1 + \frac{|x_2 - y_2|^2}{|x_1 - y_1|^2}} - 1\right)\right),\end{aligned}$$

where $x = (x_1, x_2)^t$ and $y = (y_1, y_2)^t$. Up to a constant factor, the phase of the last term can be bounded by $\omega\frac{|x_2 - y_2|^2}{|x_1 - y_1|}$, which can be estimated as follows:

- When the distance between the centers of P_i and P_j is $\Theta(1)$, we estimate this by

$$\omega\frac{|x_2 - y_2|^2}{|x_1 - y_1|} \lesssim \omega(2^q\lambda)^2 = \Theta(1)$$

  since $|x_1 - y_1| = \Theta(1)$ and $|x_2 - y_2|$ is bounded by the segment length $2^q\lambda$.
- When the distance between the centers of P_i and P_j is $o(1)$, we use the second order Taylor approximation based at the origin in the rotated frame (see Figure 1). Let P_i be the kth segment to the left of the origin, and P_j be the kth segment to the right of the origin. Then both x_1 and y_1 are of order

kh. Up to higher order errors, both x_2 and y_2 are between $\frac{1}{2}\kappa_0((k-1)h)^2$ and $\frac{1}{2}\kappa_0(kh)^2$, where κ_0 is the curvature at the origin. Therefore, we have the bounds

$$|x_2 - y_2| \lesssim ckh^2, \qquad |x_1 - y_1| = \Theta(kh).$$

These imply that

$$\omega \frac{|x_2 - y_2|^2}{|x_1 - y_1|} \lesssim \omega ckh^3 \lesssim \omega ch^2 \lesssim \Theta(1),$$

since $k \le m$, $kh < mh = L = \Theta(1)$, and $h = 2^\ell \lambda \le 2^q \lambda/\sqrt{\kappa}$.

In both cases, $\omega \frac{|x_2-y_2|^2}{|x_1-y_1|}$ is bounded by a $\Theta(1)$ constant, and hence (6) is valid up to a nonoscillatory multiplicative term. Next, we rewrite

$$\begin{aligned}
&\exp(i\omega|x-y|) \sim \exp(i\omega a_{ij} \cdot (x-y)) \\
&= \exp(i\omega a_{ij} \cdot ((x-c_i) + (c_i - c_j) + (c_j - y))) \\
&= \exp(i\omega a_{ij} \cdot (x-c_i)) \cdot \exp(i\omega a_{ij} \cdot (c_i - c_j)) \cdot \exp(-i\omega a_{ij} \cdot (y - c_j)). \qquad (7)
\end{aligned}$$

To approximate the first term in (7), we perform a Taylor expansion for $\rho^{-1}(t)$ near $t = \rho(c_i)$ and evaluate it at $\rho(x)$:

$$\begin{aligned}
&\left|\rho^{-1}(\rho(x)) - \left(\rho^{-1}(\rho(c_i)) + t_i(\rho(x) - \rho(c_i))\right)\right| \\
&\lesssim \frac{1}{2}|\rho(x) - \rho(c_i)|^2 \kappa \le \frac{1}{2}(2^\ell \lambda)^2 \kappa \le \frac{1}{2}\left(\frac{2^q \lambda}{\sqrt{\kappa}}\right)^2 \kappa = O(\lambda),
\end{aligned}$$

where κ is the maximum absolute value of the curvature in $\partial\Omega$. This is equivalent to

$$(x - c_i) = (\rho(x) - \rho(c_i)) \cdot t_i + O(\lambda).$$

Multiplying this with $i\omega a_{ij}$ and taking the exponential gives the approximation

$$\exp(i\omega a_{ij} \cdot (x - c_i)) \sim \exp(i(\omega a_{ij} \cdot t_i)(\rho(x) - \rho(c_i))). \qquad (8)$$

The same argument works for $(y - c_j)$ and gives

$$(y - c_j) = (\rho(y) - \rho(c_j)) \cdot t_j + O(\lambda)$$

and

$$\exp(-i\omega a_{ij} \cdot (y - c_j)) \sim \exp(i(-\omega a_{ij} \cdot t_j)(\rho(y) - \rho(c_j))). \qquad (9)$$

We can now further approximate the phase function of the complex exponentials in (8) and (9) as follows. Noticing that $\omega a_{ij} \cdot t_i \in [-\omega, \omega]$, we partition the interval $[-\omega, \omega]$ into $2^{\ell+1}$ equally spaced subintervals with a set K of $2^{\ell+1}+1$ gridpoints. We define $[k]$ to be the value of rounding k to the nearest gridpoint in K. Then for each $x \in P_i$,

$$(\omega a_{ij} \cdot t_i - [\omega a_{ij} \cdot t_i])(\rho(x) - \rho(c_i)) \le \frac{2\omega}{2^{\ell+1}} \cdot \frac{1}{2} \cdot \frac{2^\ell \lambda}{2} = \frac{2\pi}{4} = O(1).$$

Thus, replacing the phase $\omega a_{ij} \cdot t_i$ with $[\omega a_{ij} \cdot t_i]$ in (8) introduces an extra nonoscillatory term,

$$\exp(i\omega a_{ij} \cdot (x - c_i)) \sim \exp(i[\omega a_{ij} \cdot t_i](\rho(x) - \rho(c_i))). \qquad (10)$$

Similarly, we have for $y \in P_j$ that

$$(-\omega a_{ij} \cdot t_j + [\omega a_{ij} \cdot t_j])(\rho(y) - \rho(c_j)) \leq \frac{2\omega}{2^{\ell+1}} \cdot \frac{1}{2} \cdot \frac{2^\ell \lambda}{2} = \frac{2\pi}{4} = O(1).$$

This change of the phase function also introduces an extra nonoscillatory term,

$$\exp(-i\omega a_{ij} \cdot (y - c_j)) \sim \exp(-i[\omega a_{ij} \cdot t_j](\rho(y) - \rho(c_j))). \tag{11}$$

By introducing

$$\begin{aligned} k_{ij}^i &= [\omega a_{ij} t_i], \\ k_{ij}^j &= [\omega a_{ij} t_j], \\ U_i(x,k) &= \exp(ik(\rho(x) - \rho(c_i))), \\ U_j(y,k) &= \exp(ik(\rho(y) - \rho(c_j))), \end{aligned} \tag{12}$$

and putting (10) and (11) into (7), we have the following approximation:

$$\exp(i\omega|x-y|) \sim U_i(x, k_{ij}^i) \cdot \exp(i\omega(c_i - c_j)a_{ij}) \cdot \overline{U_j(y, k_{ij}^j)}$$

for all $x \in P_i$ and $y \in P_j$.

Since the kernel $G(x,y)$ and its derivatives have the same oscillation pattern as $\exp(i\omega|x-y|)$, $U_i(x, k_{ij}^i)$ and $\overline{U_j(y, k_{ij}^j)}$ also capture the oscillations of $G(x,y)$ for $x \in P_i$ and $y \in P_j$. Therefore, repeating the same argument allows us to write M_{ij} as

$$M_{ij}(x,y) = U_i(x, k_{ij}^i) \cdot \widetilde{M}_{ij}(x,y) \cdot \overline{U_j(y, k_{ij}^j)} \tag{13}$$

for $x \in P_i$ and $y \in P_j$, where the term $\widetilde{M}_{ij}(x,y)$ defined through this representation is nonoscillatory. Being nonoscillatory, $\widetilde{M}_{ij}(x,y)$ can be approximated with Chebyshev interpolation. To make this precise, we define

- R_i to be the Chebyshev grid of a constant size m_c in P_i,
- P_j to be the Chebyshev grid of a constant size m_c in P_j,
- I_i to be the Chebyshev interpolation operator, with entries given by $I_i(x,b)$ for $x \in P_i$ and $b \in R_i$, and
- I_j to be the Chebyshev interpolation operator, with entries given by $I_j(y,b)$ for $y \in P_j$ and $b \in R_j$.

This resulting Chebyshev approximation then reads

$$\widetilde{M}_{ij} \approx I_i \ \widetilde{M}_{ij}(R_i, R_j) \ I_j^t.$$

Putting this together with (13) and using matrix form gives

$$M_{ij} \approx \operatorname{diag}(U_i(:, k_{ij}^i)) \ I_i \ \widetilde{M}_{ij}(R_i, R_j) \ I_j^t \ \operatorname{diag}(\overline{U_j(:, k_{ij}^j)}). \tag{14}$$

For the data-sparse representation and the preconditioner, we need an aggressive rank-1 approximation for M_{ij} of the form

$$M_{ij} \approx U_i(:, k_{ij}^i)) \ e_{ij} \ U_j(:, k_{ij}^j)^*. \tag{15}$$

To do that, we use the approximation (14) and rewrite the last formula as

$$\operatorname{diag}(U_i(:, k_{ij}^i)) \ w \ e_{ij} \ w^t \ \operatorname{diag}(\overline{U_j(:, k_{ij}^j)}),$$

where w is defined to be the all-ones vector of length $2^\ell p$ and e_{ij} is a constant to be determined. To determine e_{ij}, we can solve a least squares problem,

$$e_{ij} = \text{argmin}_e \| I_i \ \widetilde{M}_{ij}(R_i, R_j) \ I_j^t - w \ e \ w^t \|^2. \tag{16}$$

The solution is

$$e_{ij} = w^\dagger \ I_i \ \widetilde{M}_{ij}(R_i, R_j) \ I_j^t \ (w^t)^\dagger,$$

where $(\cdot)^\dagger$ stands for the pseudoinverse. Notice that $w^\dagger I_i$ and $I_j^t (w^t)^\dagger$ depend only on ℓ. Therefore, they can be precomputed, and the remaining cost of computing e_{ij} is equal to $O(m_c^2)$.

Computing (15) for all pairs (i, j) with $i \neq j$ yields the following approximation for the off-diagonal part of M:

$$UEU^*.$$

Here

$$U = \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_m \end{bmatrix},$$

where each U_i is a matrix of size $2^\ell p \times (2^{\ell+1} + 1)$ given by (12). The E matrix also has an $m \times m$ block form,

$$E = \begin{bmatrix} E_{11} & \dots & E_{1m} \\ \vdots & \ddots & \vdots \\ E_{m1} & \dots & E_{mm} \end{bmatrix},$$

where E_{ij} is a matrix of size $(2^{\ell+1} + 1) \times (2^{\ell+1} + 1)$ with rows and columns indexed by K_i and K_j. E_{ij} is a matrix with value e_{ij} at entry (k_{ij}^i, k_{ij}^j) and zero everywhere else. Here we emphasize that

- U_i is a partial Fourier matrix, and
- E is extremely sparse.

These observations turn out to be essential in the construction of the preconditioner.

Summarizing the discussion for both the diagonal and off-diagonal blocks, we arrive at the data-sparse approximation

$$M \approx B + UEU^*. \tag{17}$$

2.2. Directional preconditioner. To precondition (5), we use the approximation (17) and consider the solution q of

$$(B + UEU^*)q = f.$$

First, introducing new vectors $r = -U^*q$ and $p = -Er$ gives an equivalent augmented system,

$$\begin{bmatrix} B & U & 0 \\ U^* & 0 & I \\ 0 & I & E \end{bmatrix} \begin{bmatrix} q \\ p \\ r \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}. \tag{18}$$

Factorizing the matrix in (18) gives

$$\begin{bmatrix} I & & \\ U^*B^{-1} & I & \\ & & I \end{bmatrix} \begin{bmatrix} I & & \\ & I & \\ & -T & I \end{bmatrix} \begin{bmatrix} B & & \\ & -S & \\ & & W \end{bmatrix} \begin{bmatrix} I & & \\ & I & -T \\ & & I \end{bmatrix} \begin{bmatrix} I & B^{-1}U & \\ & I & \\ & & I \end{bmatrix}, \tag{19}$$

with

$$S = U^*B^{-1}U, \quad T = S^{-1}, \quad W = E + T.$$

Since both U and B are block-diagonal, S and T are also block-diagonal:

$$S = \begin{bmatrix} S_1 & & \\ & \ddots & \\ & & S_m \end{bmatrix}, \qquad T = \begin{bmatrix} T_1 & & \\ & \ddots & \\ & & T_m \end{bmatrix},$$

with $S_i = U_i^* B_i^{-1} U_i$ and $T_i = S_i^{-1}$. Inverting the factorization (19) gives

$$\begin{bmatrix} q \\ p \\ r \end{bmatrix} = \begin{bmatrix} I & -B^{-1}U & \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} I & & \\ & I & T \\ & & I \end{bmatrix} \begin{bmatrix} B^{-1} & & \\ & -T & \\ & & W^{-1} \end{bmatrix} \begin{bmatrix} I & & \\ & I & \\ & T & I \end{bmatrix} \begin{bmatrix} I & & \\ -U^*B^{-1} & I & \\ & & I \end{bmatrix} \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}. \tag{20}$$

Applying (20) exactly can be quite costly. In order to construct an efficient preconditioner, it is essential to approximate (20) aggressively without sacrificing too much accuracy. In what follows, we shall use the notation $\lfloor A \rfloor$ to stand for an approximation to the matrix A, although the actual approximation scheme can be different for different matrices.

- First, the $2^\ell p \times 2^\ell p$ matrix B_i is the restriction of the integral operator to a straight segment of length $2^\ell \lambda$. Since the geometry is fixed, all B_i and B_i^{-1} can be precomputed. Since this is also a 1D problem (i.e., restriction to a straight segment), we can use the hierarchical matrix algebra [7] or the hierarchical semiseparable (HSS) matrices [32] to compress and apply B_i^{-1} efficiently. We denote by $\lfloor B_i^{-1} \rfloor$ the approximation for B_i^{-1} and accordingly by $\lfloor B^{-1} \rfloor$ the one for B^{-1}.
- Second, since each U_i is a partial Fourier matrix, applying U and U^* reduces to a number of FFTs, which is highly efficient.
- Third, S_i and T_i can be precomputed, as they depend only on U_i and B_i, both of which have already been precomputed. An important observation is that T_i is numerically sparse (see Figure 2). Therefore, for the sake of efficiency, we approximate T_i with $\lfloor T_i \rfloor$, which is obtained by thresholding the entries in absolute value. In the numerical results, the number of nonzero entries in $\lfloor T_i \rfloor$ is kept proportional to the dimension of T_i. This approximation of T is denoted by $\lfloor T \rfloor$.
- The final task is to build an approximate inverse of $W = E + T$. Here, the essential observation is the following:
  - T concentrates on its diagonal (see Figure 2).

  Based on this, we define $\lfloor W \rfloor$ to be the sum of E and the diagonal of T (i.e., thresholding the rest of the entries of T to zero). $\lfloor W \rfloor$ is extremely sparse, as

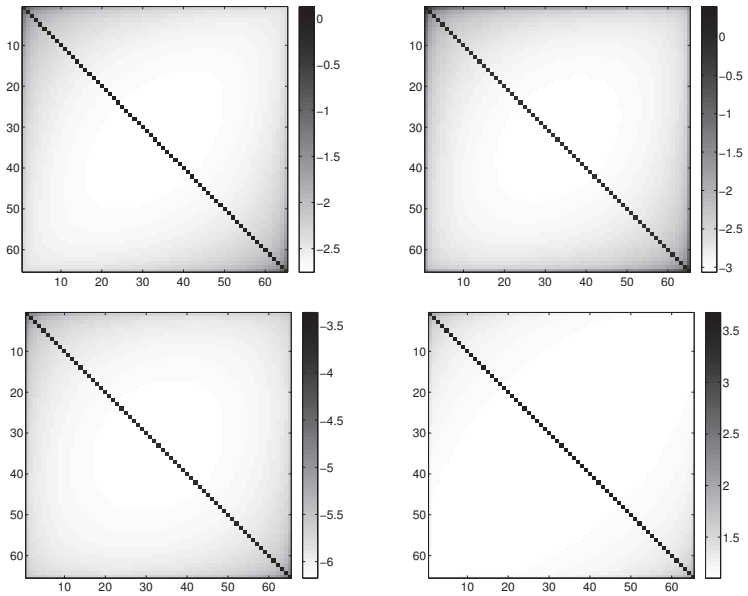

FIG. 2. *The absolute value of the entries of S_i (left) and T_i (right) in logarithmic scale (base 10) for the sound-soft case (top row) and the sound-hard case (bottom row). For this specific segment, $\ell_i = 5$ and $p = 8$. Hence, there are 256 equally spaced quadrature points in P_i, and the cardinality of K is 65. Both S_i and T_i are numerically sparse and diagonally dominant.*

the number of nonzeroes is about 3/2 times the dimension of the matrix in most cases. To approximate W^{-1}, we choose to perform an incomplete LU decomposition,

$$\lfloor W \rfloor \approx L_{\lfloor W \rfloor} R_{\lfloor W \rfloor},$$

where $L_{\lfloor W \rfloor}$ and $R_{\lfloor W \rfloor}$ are highly sparse upper- and lower triangular. Figure 3 shows the typical sparsity pattern of the matrix $L_{\lfloor W \rfloor}$ and $R_{\lfloor W \rfloor}$. Once these two matrices are available, applying W^{-1} approximately reduces to applying backward and forward substitutions with matrices $L_{\lfloor W \rfloor}$ and $R_{\lfloor W \rfloor}$.

Once these approximations have been formed, we approximate (20) with

$$\begin{bmatrix} q \\ p \\ r \end{bmatrix} \Leftarrow \begin{bmatrix} I & -\lfloor B^{-1} \rfloor U & \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} I & & \\ & I & \lfloor T \rfloor \\ & & I \end{bmatrix} \begin{bmatrix} \lfloor B^{-1} \rfloor & & \\ & -\lfloor T \rfloor & \\ & & R^{-1}_{\lfloor W \rfloor} L^{-1}_{\lfloor W \rfloor} \end{bmatrix} \begin{bmatrix} I & & \\ & I & \\ & \lfloor T \rfloor & I \end{bmatrix} \begin{bmatrix} I & & \\ -U^* \lfloor B^{-1} \rfloor & I & \\ & & I \end{bmatrix} \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}. \tag{21}$$

We emphasize again that the following approximations are used for computing (21):

- replacing B^{-1} with $\lfloor B^{-1} \rfloor$ via hierarchical matrix or HSS approximation for each B_i^{-1},
- applying U rapidly via FFT for each U_i,
- replacing T with $\lfloor T \rfloor$ via sparse approximation for each T_i, and
- replacing W^{-1} with $R^{-1}_{\lfloor W \rfloor} L^{-1}_{\lfloor W \rfloor}$ via sparse backward and forward substitutions for $L_{\lfloor W \rfloor}$ and $R_{\lfloor W \rfloor}$.

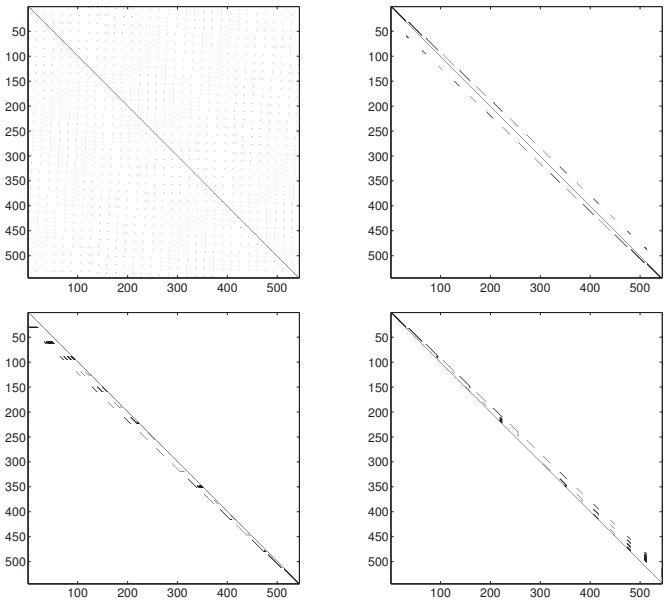

FIG. 3. *Top-left: the sparsity pattern of* $\lfloor W \rfloor$. *Top-right: the sparsity pattern of* $\lfloor W \rfloor$ *in a new ordering. In this ordering, the indices with the same frequency parameter* k *are grouped together. As a result, the matrix can be viewed as a block matrix with* $(2^{\ell+1}+1) \times (2^{\ell+1}+1)$ *blocks, each of size* $m \times m$. *Bottom-left: the sparsity pattern of* $L_{\lfloor W \rfloor}$ *in the new ordering. Bottom-right: the sparsity pattern of* $U_{\lfloor W \rfloor}$ *in the new ordering.*

Based on (21), our preconditioner is defined as follows. For a given f, it

- forms vector $(f^*, 0, 0)^*$,
- carries out the computation of (21), and
- extracts the first component q of the resulting vector.

Since the key step of constructing a data-sparse representation of the operator relies on the directional nature of the kernel $G(x, y)$, we name it the *directional preconditioner*.

2.3. Complexity analysis. We first consider the setup cost for the preconditioner, i.e., the approximate factorization in (21). Since the matrices B_i^{-1} and T_i for a segment P_i depend only on the integer length parameter ℓ of P_i, the possible choices for these matrices are fixed and independent of the scatterer. All these possible choices can be precomputed once and stored for future use.

As a result, the setup algorithm consists of only two parts: the evaluation of E and $\lfloor W \rfloor$, and the incomplete LU factorization $\lfloor W \rfloor \approx L_{\lfloor W \rfloor} R_{\lfloor W \rfloor}$.

- For the first part, since E has only ω nonzero entries and computing each entry takes $O(1)$ steps (16), the overall cost for this step is $O(\omega) = O(n)$. Once E is formed, computing $\lfloor W \rfloor$ also takes at most $O(n)$ steps.
- In our implementation, the incomplete LU factorization $\lfloor W \rfloor \approx L_{\lfloor W \rfloor} R_{\lfloor W \rfloor}$ is computed with the Crout version of ILU factorization, known as ILUC [25], with the drop tolerance set to be a small value, such as 10^{-6}. The resulting factors $L_{\lfloor W \rfloor}$ and $R_{\lfloor W \rfloor}$ contain on average 5 or 6 nonzero entries per row/column. Therefore, they are extremely sparse. In all our numerical experiments, the computational time for generating the ILUC factorization scales roughly linearly with the dimension of the matrix $\lfloor W \rfloor$.

Let us now consider the application cost of the preconditioner (21). For the major

steps of applying (21), we have the following estimates:

- The application of $\lfloor B^{-1} \rfloor$ is linear time due to the hierarchical matrix algebra approximation for B_i^{-1}.
- The application of U is $O(n \log n)$ since each U_i is a partial Fourier matrix and the FFT can be used.
- The application of $\lfloor T \rfloor$ is $O(n)$ since the number of nonzeroes in $\lfloor T \rfloor$ is proportional to $O(n)$ after we threshold each T_i.
- Applying $R_{\lfloor W \rfloor}^{-1} L_{\lfloor W \rfloor}^{-1}$ reduces to the sparse backward and forward substitution algorithm for matrices $L_{\lfloor W \rfloor}$ and $R_{\lfloor W \rfloor}$. Since the number of nonzero entries in both matrices scales linearly with respect to the matrix dimension, the cost of this step is also $O(n)$.

Putting these together shows that the application cost of the preconditioner scales like $O(n \log n)$.

3. Numerical results. The proposed preconditioner is implemented in MATLAB. The numerical results in this section are obtained on a desktop computer with a 3.60 GHz CPU.

In these experiments, the domain boundary is discretized by the Nyström method with $p = 8$ points per wavelength. The Chebyshev grid size m_c used to construct e_{ij} for the matrix E is set to be 10. The boundary conditions for both the sound-soft and the sound-hard problems correspond to an incoming plane wave pointing towards the positive x_1 direction. The fast algorithm described in [33] with accuracy 10^{-6} is used to accelerate the matrix-vector multiplication of the iterative solve. The coupling constant η is chosen to be $\frac{\omega}{2}$ for the sound-soft case (1) and to be $\frac{3\omega}{4}$ for the sound-hard case (2).

3.1. Example 1. The first example is an ellipse, shown in Figure 4, with major axis equal to $r_1 = 1$ and minor axis equal to $r_2 = 0.45$.

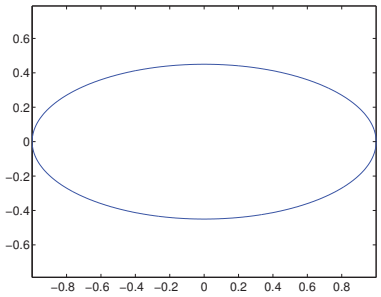

FIG. 4. *The scatterer of Example* 1.

The first test of this example concerns the Dirichlet problem of sound-soft scattering for different values of ω. For the GMRES solver, the relative tolerance is set to 10^{-6} to match the accuracy of the matrix-vector multiplication algorithm, and the restart number is equal to 50. The results are reported in Table 1, where

- T_s is the setup time of the preconditioner in seconds,
- T_a is the application time of the preconditioner in seconds,
- T_m is the matrix-vector multiplication time using the algorithm from [33],
- n_p is the iteration number of the iterative solver when the preconditioner is used, and finally,
- n_n is the iteration number without any preconditioning.

The ratio T_a/T_m is a good indicator of the computation cost of applying the preconditioner, and n_p/n_n shows the effectiveness of the preconditioner.

TABLE 1
Results of the sound-soft scattering for Example 1.

ω	n	T_s	T_a	T_m	n_p	n_n
1.4e+03	8.2e+03	1.2e+00	3.2e-03	3.8e-01	12	31
5.3e+03	3.3e+04	5.3e+00	2.4e-02	1.5e+00	14	47
2.1e+04	1.3e+05	2.5e+01	7.4e-02	6.3e+00	16	71

The second test considers the Neumann problem of sound-hard scattering, again for different values of ω. The results are summarized in Table 2.

TABLE 2
Results of the sound-hard scattering for Example 1.

ω	n	T_s	T_a	T_m	n_p	n_n
1.4e+03	8.2e+03	1.7e+00	2.7e-03	3.8e-01	12	24
5.5e+03	3.3e+04	7.4e+00	9.0e-03	1.6e+00	15	36
2.2e+04	1.3e+05	3.9e+01	5.1e-02	6.7e+00	17	53

The results of these first two tests show that the setup time of the preconditioner is typically equivalent to a couple of fast matrix-vector multiplications. On the other hand, the application time of the preconditioner is much lower, and, as a result, the cost of applying the preconditioner is almost negligible during the iterative solution. The iteration number of the preconditioned system is significantly lower than that of the unpreconditioned system and in fact scales like $O(\log \omega)$.

The third test examines the dependence of the convergence rate on the curvature of the domain boundary. In this case, ω is set to be 1.4e+03, and the minor axis r_2 is set to be 0.05, 0.15, and 0.45. From the results reported in Table 3, it seems that the convergence rate is quite insensitive to the curvature variation.

TABLE 3
Results of different r_2 values for Example 1.

r_2	n	n_p	n_n
0.05	8.2e+03	13	29
0.15	8.2e+03	13	24
0.45	8.2e+03	12	31

The fourth test checks the convergence behavior of different iterative algorithms. For multiple values of ω, the problem is solved with the restarted GMRES, the nonrestarted GMRES, and the TFQMR algorithm. For each algorithm, both the preconditioned and unpreconditioned equations are solved. The results, summarized in Table 4, show that the performance for the restarted and the nonrestarted GMRES algorithms is comparable both for preconditioned and unpreconditioned systems. On the other hand, for the TFQMR algorithm, the preconditioned solve exhibits a similar behavior while the unpreconditioned solve takes many more iterations.

The final test studies the spectra of the preconditioned and unpreconditioned systems. Here we let $r_2 = 0.45$, $\omega = 3.4\text{e+}02$, and $n = 2.0\text{e+}03$. The spectrum plots in Figure 5 show that the eigenvalue distribution of the preconditioned system is much more clustered, and this explains why the preconditioned system converges faster.

TABLE 4
Results of different iterative algorithms for Example 1. *GMRES-R stands for the restarted GMRES; GMRES-NR stands for the nonrestarted GMRES.*

		GMRES-R		GMRES-NR		TFQMR	
ω	n	n_p	n_n	n_p	n_n	n_p	n_n
1.4e+03	8.2e+03	12	31	12	31	15	62
5.5e+03	3.3e+04	14	46	14	46	16	141
2.2e+04	1.3e+05	16	74	16	69	19	240

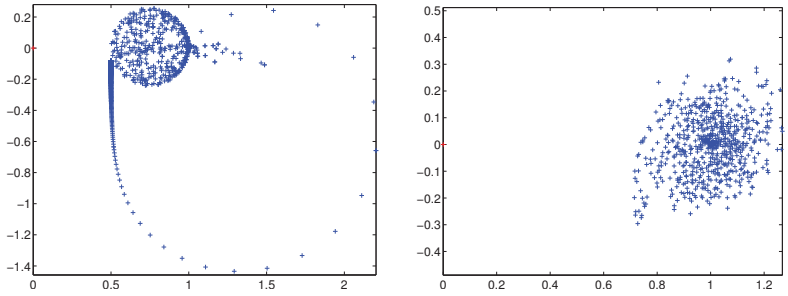

FIG. 5. *Spectra of unpreconditioned and preconditioned systems for Example* 1.

3.2. Example 2. The second example is a bean-shaped scatter, as shown in Figure 6.

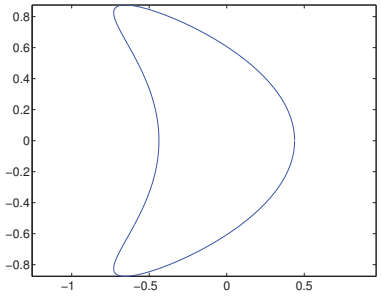

FIG. 6. *The scatterer of Example* 2.

The first test considers the Dirichlet problem of sound-soft scattering for different values of ω. The parameters of the GMRES solve are set to be the same as those for Example 1, i.e., the relative tolerance equal to 10^{-6} and the restart number equal to 50. The results are reported in Table 5.

TABLE 5
Results of the sound-soft scattering for Example 2.

ω	n	T_s	T_a	T_m	n_p	n_n
1.3e+03	8.2e+03	1.2e+00	2.3e-03	4.0e-01	13	35
5.2e+03	3.3e+04	5.5e+00	1.3e-02	1.8e+00	14	50
2.1e+04	1.3e+05	3.2e+01	2.4e-02	7.5e+00	16	79

The second test studies the Neumann problem of sound-hard scattering, again for different values of ω. The results are given in Table 6. The results in these two tests are qualitatively similar to the analogous tests for Example 1.

TABLE 6
Results of the sound-hard scattering for Example 2.

ω	n	T_s	T_a	T_m	n_p	n_n
1.3e+03	8.2e+03	1.7e+00	2.2e-03	4.1e-01	13	24
5.2e+03	3.3e+04	7.4e+00	8.5e-03	1.8e+00	15	33
2.1e+04	1.3e+05	3.9e+01	4.0e-02	7.4e+00	17	47

The third test checks the convergence behavior of different iterative algorithms. For different values ω, the problem is again solved with the restarted GMRES, the nonrestarted GMRES, and the TFQMR. The results, summarized in Table 7, show that the performance for the restarted and the nonrestarted GMRES algorithms is comparable and that the preconditioned system converges much faster for the TFQMR algorithm.

TABLE 7
Results of different iterative algorithms for Example 2.

		GMRES-R		GMRES-NR		TFQMR	
ω	n	n_p	n_n	n_p	n_n	n_p	n_n
1.3e+03	8.2e+03	13	35	13	35	16	62
5.2e+03	3.3e+04	14	50	14	50	17	91
2.1e+04	1.3e+05	16	79	16	74	22	152

The final test computes the spectra of the preconditioned and unpreconditioned systems. Here $\omega = 3.4\text{e+}02$ and $n = 2.0\text{e+}03$. The spectrum plots, given in Figure 7, show that the eigenvalue distribution of the preconditioned system is again much more clustered.

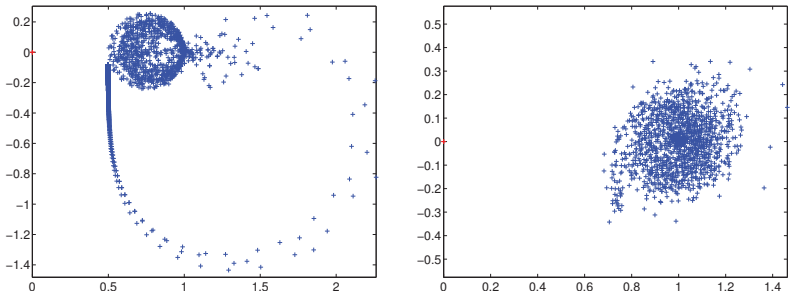

FIG. 7. *Spectra of unpreconditioned and preconditioned systems for Example* 2.

3.3. Example 3. The third example is a resonator, shown in Figure 8, with opening angle equal to $\frac{\pi}{2}$.

The first test considers the Dirichlet problem of sound-soft scattering for different values of ω, again with the GMRES relative tolerance equal to 10^{-6} and the restart number equal to 50. The results of this test are reported in Table 8.

The second test considers the Neumann problem of sound-hard scattering for different values of ω. The results are summarized in Table 9.

The third test examines the dependence of the convergence behavior on the opening angle of the resonator. In this case, the ω is set to be 1.4e+03, and the opening angle θ is equal to $3\pi/4$, $\pi/2$, and $\pi/4$. From the results in Table 10, we notice that the iteration count depends highly on the opening angle. When the opening

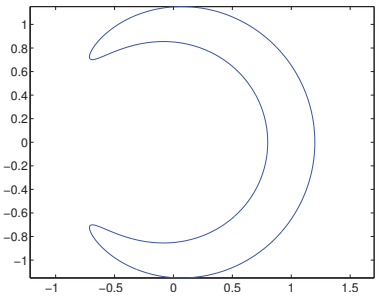

FIG. 8. *The scatterer of Example* 3.

TABLE 8
Results of the sound-soft scattering for Example 3.

ω	n	T_s	T_a	T_m	n_p	n_n
8.1e+02	8.2e+03	1.2e+00	3.6e-03	4.1e-01	23	53
3.2e+03	3.3e+04	5.4e+00	1.1e-02	1.8e+00	26	79
1.3e+04	1.3e+05	3.2e+01	3.8e-02	7.7e+00	30	122

TABLE 9
Results of the sound-hard scattering for Example 3.

ω	n	T_s	T_a	T_m	n_p	n_n
8.1e+02	8.2e+03	1.7e+00	2.7e-03	4.1e-01	20	39
3.2e+03	3.3e+04	7.3e+00	8.1e-03	1.8e+00	24	53
1.3e+04	1.3e+05	3.9e+01	3.8e-02	7.7e+00	27	85

TABLE 10
Results of different opening angle θ *values for Example* 3.

θ	n	n_p	n_n
$3\pi/4$	8.2e+03	19	44
$\pi/2$	8.2e+03	23	53
$\pi/4$	8.2e+03	76	178

angle is small, the scatterer is a strong resonator, and both the preconditioned and unpreconditioned iteration counts grow rapidly.

The fourth test again checks the convergence behavior of different iterative algorithms. For different values ω, the problem is still solved with the restarted GMRES, the nonrestarted GMRES, and the TFQMR algorithm. The results in Table 11 again show that the preconditioned system offers a huge speedup for the TFQMR algorithm.

TABLE 11
Results of different iterative algorithms for Example 3.

		GMRES-R		GMRES-NR		TFQMR	
ω	n	n_p	n_n	n_p	n_n	n_p	n_n
8.1e+02	8.2e+03	23	53	23	52	28	101
3.2e+03	3.3e+04	26	79	26	76	31	149
1.3e+04	1.3e+05	30	122	30	111	39	232

The final test computes the spectra of the preconditioned and unpreconditioned systems. Here, the opening angle θ is $\pi/2$, ω = 3.4e+02, and n = 2.0e+03. The spectrum plots in Figure 9 show that the eigenvalue distribution of the precondi-

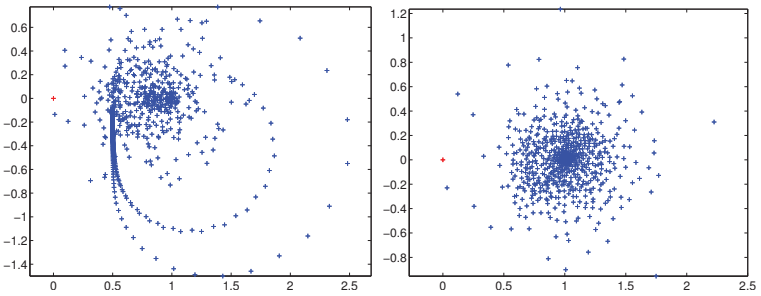

FIG. 9. *Spectra of unpreconditioned and preconditioned systems for Example* 3.

tioned system is still more clustered, though not as significant as the distributions in Examples 1 and 2.

4. Conclusion. This paper presents the directional preconditioner for the combined field integral equations (CFIEs) of high-frequency acoustic obstacle scattering in two dimensions. The main idea is to construct a data-sparse approximation of the linear operator, transform it into an approximate sparse linear system, and form an approximate inverse using efficient sparse and hierarchical linear algebra algorithms.

In this paper, we have assumed that the boundary is discretized with an equally spaced set of discretization points. For nonequally spaced points, the construction goes through as well, except that the FFT has to be replaced with nonuniform FFTs. As a result, some of the scatterer-independent precomputation can become dependent on the discretization pattern.

We would like to extend this approach to 3D scatterers. While the main idea should work, the lack of equally spaced discretization for general surfaces poses a clear challenge for this approach.

The long-term goal is to construct a direct solver for the boundary integral equations of the obstacle scattering problem. It is not clear at this point whether such a direct solver even exists. However, this paper can be viewed a first step toward exploring in this direction.

One important ingredient of our approach is to transform a dense oscillatory matrix into a sparse one. Once it is in a sparse form, we are able to leverage the power of sparse linear algebra algorithms. While traditionally there is relatively little overlap between work in integral equations and that in sparse linear algebra, this work hints at a fruitful exchange of ideas between these two fields.

Acknowledgments. The author thanks Anil Damle and the anonymous referees for comments and suggestions.

REFERENCES

[1] F. Alouges, S. Borel, and D. P. Levadoux, *A stable well-conditioned integral equation for electromagnetism scattering*, J. Comput. Appl. Math., 204 (2007), pp. 440–451.

[2] X. Antoine, A. Bendali, and M. Darbas, *Analytic preconditioners for the electric field integral equation*, Internat. J. Numer. Methods Engrg., 61 (2004), pp. 1310–1331.

[3] X. Antoine and Y. Boubendir, *An integral preconditioner for solving the two-dimensional scattering transmission problem using integral equations*, Int. J. Comput. Math., 85 (2008), pp. 1473–1490.

[4] X. Antoine and M. Darbas, *Alternative integral equations for the iterative solution of acoustic scattering problems*, Quart. J. Mech. Appl. Math., 58 (2005), pp. 107–128.

[5] X. Antoine and M. Darbas, *Generalized combined field integral equations for the iterative solution of the three-dimensional Helmholtz equation*, M2AN Math. Model. Numer. Anal., 41 (2007), pp. 147–167.
[6] 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.
[7] S. Börm, L. Grasedyck, and W. Hackbusch, *Hierarchical Matrices*, Max-Planck-Institute Lecture Notes, 2006.
[8] 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.
[9] O. Bruno, T. Elling, and C. Turc, *Regularized integral equations and fast high-order solvers for sound-hard acoustic scattering problems*, Internat. J. Numer. Methods Engrg., 91 (2012), pp. 1045–1072.
[10] F. X. Canning, *Sparse approximation for solving integral equations with oscillatory kernels*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 71–87.
[11] S. H. Christiansen and J.-C. Nédélec, *Des préconditionneurs pour la résolution numérique des équations intégrales de frontière de l'acoustique*, C. R. Acad. Sci. Paris Sér. I Math., 330 (2000), pp. 617–622.
[12] S. H. Christiansen and J.-C. Nédélec, *A preconditioner for the electric field integral equation based on Calderon formulas*, SIAM J. Numer. Anal., 40 (2002), pp. 1100–1135.
[13] D. Colton and R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory*, 3rd ed., Appl. Math. Sci. 93, Springer, New York, 2013.
[14] E. Corona, P.-G. Martinsson, and D. Zorin, *An O(N) direct solver for integral equations on the plane*, Appl. Comput. Harmon. Anal., 38 (2015), pp. 284–317.
[15] L. Demanet and L. Ying, *Wave atoms and sparsity of oscillatory patterns*, Appl. Comput. Harmon. Anal., 23 (2007), pp. 368–387.
[16] L. Demanet and L. Ying, *Scattering in flatland: Efficient representations via wave atoms*, Found. Comput. Math., 10 (2010), pp. 569–613.
[17] H. Deng and H. Ling, *Fast solution of electromagnetic integral equations using adaptive wavelet packet transform*, IEEE Trans. Antennas Propagation, 47 (1999), pp. 674–682.
[18] H. Deng and H. Ling, *On a class of predefined wavelet packet bases for efficient representation of electromagnetic integral equations*, IEEE Trans. Antennas Propagation, 47 (1999), pp. 1772–1779.
[19] R. W. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.
[20] W. L. Golik, *Wavelet packets for fast solution of electromagnetic integral equations*, IEEE Trans. Antennas Propagation, 46 (1998), pp. 618–624.
[21] K. L. Ho and L. Ying, *Hierarchical interpolative factorization for elliptic operators: Integral equations*, Comm. Pure Appl. Math., (2015), to appear.
[22] 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 (2006), pp. 218–232.
[23] R. Kress, *Minimizing the condition number of boundary integral operators in acoustic and electromagnetic scattering*, Quart. J. Mech. Appl. Math., 38 (1985), pp. 323–341.
[24] R. Kress, *Linear Integral Equations*, 3rd ed., Appl. Math. Sci. 82, Springer, New York, 2014.
[25] N. Li, Y. Saad, and E. Chow, *Crout versions of ILU for general sparse matrices*, SIAM J. Sci. Comput., 25 (2003), pp. 716–728.
[26] P. G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23.
[27] P. G. Martinsson and V. Rokhlin, *A fast direct solver for scattering problems involving elongated structures*, J. Comput. Phys., 221 (2007), pp. 288–302.
[28] J.-C. Nédélec, *Acoustic and Electromagnetic Equations, Integral Representations for Harmonic Problems*, Appl. Math. Sci. 144, Springer-Verlag, New York, 2001.
[29] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[30] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[31] O. Steinbach and W. L. Wendland, *The construction of some efficient preconditioners in the boundary element method. Numerical treatment of boundary integral equations*, Adv. Comput. Math., 9 (1998), pp. 191–216. .
[32] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li, *Fast algorithms for hierarchically semiseparable matrices*, Numer. Linear Algebra Appl., 17 (2010), pp. 953–976.
[33] L. Ying, *Fast directional computation of high frequency boundary integrals via local FFTs*, Multiscale Model. Simul., 13 (2015), pp. 423–439.