MULTISCALE MODEL. SIMUL.
Vol. 13, No. 2, pp. 644–660

SPARSIFYING PRECONDITIONER FOR THE LIPPMANN–SCHWINGER EQUATION*

LEXING YING†

Abstract. The Lippmann–Schwinger equation is an integral equation formulation for acoustic and electromagnetic scattering from an inhomogeneous medium and quantum scattering from a localized potential. We present the sparsifying preconditioner for accelerating the iterative solution of the Lippmann–Schwinger equation. This new preconditioner transforms the discretized Lippmann–Schwinger equation into sparse form and leverages the efficient sparse linear algebra algorithms for computing an approximate inverse. This preconditioner is efficient and easy to implement. When combined with standard iterative methods, it results in almost frequency-independent iteration counts. We provide two- and three-dimensional numerical results to demonstrate the effectiveness of this new preconditioner.

Key words. Lippmann–Schwinger equation, acoustic and electromagnetic scattering, quantum scattering, preconditioner, sparse linear algebra

AMS subject classifications. 65F08, 65F50, 65N22, 65R20, 78A45

DOI. 10.1137/140985147

1. Introduction.

This paper is concerned with the efficient solution of the Lippmann–Schwinger equation, which describes time-harmonic scattering from inhomogeneous media in acoustics and electromagnetics as well as time-harmonic scattering from localized potentials in quantum mechanics. The simplest form of this equation comes from inhomogeneous acoustic scattering. Let ω be the frequency of the time-harmonic wave, and denote the index of refraction by $1 - m(x)$. The inhomogeneity $m(x)$ is a function supported in a compact domain $\Omega \subset \mathrm{R}^d$ of size $O(1)$, so the index of refraction is 1 outside Ω. Given an incoming wave $u_I(x)$ that satisfies the free space Helmholtz equation

$$(\Delta + \omega^2) u_I(x) = 0, \quad x \in \mathrm{R}^d,$$

the problem of scattering from inhomogeneous media is to find the scattered field $u(x)$ such that the total field $u(x) + u_I(x)$ satisfies

$$(\Delta + \omega^2(1 - m(x)))(u(x) + u_I(x)) = 0, \quad x \in \mathrm{R}^d, \tag{1}$$

and $u(x)$ obeys the Sommerfeld radiation condition:

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

For the free space Helmholtz operator $-(\Delta + \omega^2)$, the Green's function is given by

$$G(x) = \begin{cases} \frac{1}{2i\omega} \exp(i\omega|x|), & d = 1, \\ \frac{i}{4} H_0^1(\omega|x|), & d = 2, \\ \frac{1}{4\pi|x|} \exp(i\omega|x|), & d = 3. \end{cases}$$

*Received by the editors September 3, 2014; accepted for publication (in revised form) April 27, 2015; published electronically June 24, 2015. This work was partially supported by the National Science Foundation under award DMS-1328230 and 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/13-2/98514.html

†Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (lexing@stanford.edu).

Rewriting (1) as

$$(-\Delta - \omega^2 + \omega^2 m(x))u(x) = -\omega^2 m(x) u_I(x), \quad x \in \mathrm{R}^d,$$

and convolving it with $G(x)$ gives

$$u + G * (\omega^2 m u) = G * (-\omega^2 m u_I), \tag{3}$$

which is the Lippmann–Schwinger equation written in terms of the scattered field $u(x)$. One can also write the Lippmann–Schwinger equation in terms of the total field $u(x)+u_I(x)$, but we shall stick to (3) since its unknown $u(x)$ satisfies the Sommerfeld radiation condition. For the inhomogeneous electromagnetic scattering and quantum scattering, the same derivation results in integral equations similar to (3).

For high-frequency wave fields (i.e., ω large), computing the numerical solution of scattering in inhomogeneous media remains a challenging computational problem especially in three dimensions. For practical and numerical purposes, working with (3) rather than (1) offers several advantages.

- First, (3) is written in terms of $u(x)$ only for $x \in \Omega$ as $m(x) = 0$ for $x \notin \Omega$. Therefore, the unknown function in (3) is defined on the compact set Ω versus the whole space R^d as in (1).
- Second, once $u(y)$ for $y \in \Omega$ is computed, the whole scattered field in R^d defined via
$$u(x) = -\int_\Omega G(x-y)\omega^2 m(y)(u(y)+u_I(y))dy \quad \forall x \in \mathrm{R}^d$$
satisfies the Sommerfeld radiation condition (2) automatically. However, for (1) a special treatment such as perfectly matched layer (PML) [5, 10] or absorbing boundary condition (ABC) [15] is required to approximate (2) at a finite distance.
- Third, the convolution structure of (3) allows for rapid application of the integral operator via the fast Fourier transform (FFT) [11].
- Finally, most treatments of (1) for high-frequency problems suffer from the pollution effect [3], due to the error from the local finite difference or finite element stencils. This problem does not show up in (3) because of the explicit introduction of the Green's function.

However, working with (3) also brings a couple of numerical issues. Discretizing the integral equation (3) results in a dense linear system, which renders typical direct solvers impractical except for one-dimensional (1D) problems. For this reason, most numerical solutions of the Lippmann–Schwinger equation use iterative solvers. Since the Lippmann–Schwinger equation is numerically ill-conditioned, standard iterative methods without preconditioning require a huge number of iterations for high-frequency problems. This is especially true when the inhomogeneity $m(x)$ has sharp transitions (see [13], for example). Therefore, there is a clear need for developing good preconditioners for the Lippmann–Schwinger equation.

Recently, there has been a substantial amount of work on developing efficient solvers and preconditioners for the integral formulations of second order partial differential equations [1, 4, 19, 20, 25]. The methods proposed in [2, 8, 9, 22, 23, 24] are specifically concerned with computing numerical solutions of the Lippmann–Schwinger equation in the recent literature.

Several of these methods focused on the design of efficient preconditioners. In [8], Bruno and Hyde constructed a preconditioner by replacing the inhomogeneity $m(x)$

with a piecewise constant and radially symmetric approximation and then inverting the associated approximate integral operator with semianalytic methods. Such a preconditioner is effective when there exists a good piecewise constant and radially symmetric approximation, and naturally deteriorates when there is no such approximation.

In [9], Chen proposed a method for solving the two-dimensional (2D) Lippmann–Schwinger equation using a technique that is now often referred to as recursive interpolative decomposition or recursive skeletonization. This method generates in $O(N^{3/2})$ steps an approximate inverse which can be used either as a direct solver if the accuracy is sufficient or as a preconditioner. More advanced and efficient methods for general integral equations have been proposed in [12] for 2D problems and in [21] for both 2D and three-dimensional (3D) problems. These recent methods achieve quasi-linear complexity for integral equations with nonoscillatory kernels, but fall back to the same $O(N^{3/2})$ cost for 2D high-frequency Lippmann–Schwinger equations. For all of these methods, the prefactor of the complexity depends on the required accuracy and can be very large when they are used as direct solvers.

There have also been attempts to apply two-grid and multigrid methods to precondition the Lippmann–Schwinger equation (see [23], for example). However, due to the highly oscillatory nature of the solution field, such methods tend to improve the iteration number by at most a constant factor.

In this paper, we introduce a new preconditioner called the sparsifying preconditioner for the Lippmann–Schwinger equation. The main idea is to transform the discretized Lippmann–Schwinger equation approximately into a sparse linear system. This is possible since the scattered field indeed satisfies the Helmholtz equation in Ω and approximately the absorbing-type boundary conditions on $\partial\Omega$. Once the sparse linear system is ready, we invert it by leveraging the efficiency of sparse linear algebra algorithms.

When combined with a standard iterative solver such as GMRES, the sparsifying preconditioner typically converges in a small number of iterations. The actual running time of the preconditioned iteration is often only a few times more compared to that of the nearest-neighbor finite difference method for (1). When compared with the recently proposed integral equation methods [9, 12, 21], the current approach offers much faster solution time even though all of the methods share the same asymptotic complexity. In a nutshell, the sparsifying preconditioner seamlessly combines the accuracy of the integral equation approach with the efficiency of the finite difference method.

The rest of this paper is organized as follows. In section 2 we introduce the discretization scheme and describe the main idea. Sections 3 and 4 explain the details of the sparsifying preconditioner for rectangular domains and general domains, respectively. Section 5 extends our approach to the Laplace equation with potential perturbation. Finally, discussions and directions for future work are given in section 6.

2. Discretization and main idea.

The sparsifying preconditioner to be presented is mostly independent of the discretization scheme and the dimension. We assume that a Nyström method is used in order to keep the presentation simple. Our discussion focuses on the 2D and 3D cases, since the 1D problem is not computationally challenging.

At any point $x \in \Omega$, the local wave velocity is given by $c(x) = (1 - m(x))^{-1/2}$. At frequency ω, the local wavelength at x is equal to $\lambda(x) = (2\pi/\omega)(1 - m(x))^{-1/2}$.

2.1. Discretization.

Let us first consider the 2D case. The domain Ω is discretized with a uniform grid of step size h, where h is chosen such that there are at least a few grid points per wavelength across the domain. Such a uniform grid is convenient when the variation in $c(x)$ is not too large. For problems with large variations in $c(x)$, an adaptively refined grid should be used instead. The grid points are indexed by integer pairs $i = (i_1, i_2)$ with location given by $x_i = ih = (i_1 h, i_2 h)$.

We assume that $m(x)$ is zero within two layers of grid points from the boundary $\partial\Omega$, which can be easily satisfied by slightly enlarging Ω if necessary. The discretization uses $m_i = m(x_i)$; let u_i be the numerical approximation to $u(x_i)$. Writing the integral on the left-hand side of (3) explicitly at the point x_i gives

$$(G * (\omega^2 mu))(x_i) = \int_\Omega G(x_i - y)(\omega^2 m(y)u(y))dy, \tag{4}$$

and this integral is approximated with numerical quadrature using the grid points in Ω.

When $m(x)$ is discontinuous, discretizing (4) efficiently and accurately is a nontrivial task, and we refer the reader to [6, 7] for recent developments. For simplicity, we assume here that $m(x)$ is smooth (but still with possible sharp transitions) and that the grid size h is chosen to sufficiently resolve the transitions in $m(x)$. For such $m(x)$, the discretization of (4) is more straightforward and the only difficulty is the singularity of $G(x)$ at the origin. A simple quadrature rule is

$$\int_\Omega G(x_i - y)(\omega^2 m(y)u(y))dy \approx \sum_{j:x_j\in\Omega} k_{i-j}\omega^2 m_j u_j,$$

where

$$k_t = G(ht)h^2, \quad t \neq (0,0),$$

and $k_{(0,0)}$ is given by a quadrature correction near the origin. This is a one-point correction of the trapezoidal rule and gives an accuracy of $\mathrm{O}(h^4\log(1/h))$ (see [13], for example). There are also higher order versions that correct the grid points in the neighborhood of the origin, but they do not affect the following discussion of the sparsifying preconditioner significantly.

After quadrature approximation, the discrete form of the Lippmann–Schwinger equation (3) takes the form, for i with $x_i \in \Omega$,

$$u_i + \sum_{j:x_j\in\Omega} k_{i-j}\omega^2 m_j u_j = g_i, \tag{5}$$

where g_i is the discretized value of the right-hand side of (3). In a slight abuse of notation, we define u to be the vector with entries given by u_i, and similarly for g and m. Then, (5) can be written as

$$(I + K\omega^2 m)u = g, \tag{6}$$

where K is the matrix with entries defined via $K(i,j) = k_{i-j}$ and m is understood as operator of multiplying with the vector m entrywise.

The situation for the 3D case is almost the same except that i is now an integer triple (i_1, i_2, i_3) and $k_t = G(th)h^3$ for $t \neq (0,0,0)$.

2.2. Main idea. The main idea of our approach is to find a sparse and local operator Q such that in

$$Q(I + K\omega^2 m)u = Qg$$

the operator $Q(I + K\omega^2 m)$ is also numerically sparse and localized. This is possible since the scattered field u satisfies

- the Helmholtz equation (1) in the interior of Ω, and
- approximately the absorbing-type boundary conditions on the boundary of Ω.

For a grid point indexed by i, we define its neighborhood $\mu(i)$ to be

$$\mu(i) = \{j : \|i - j\|_\infty \le 1, x_j \in \Omega\}.$$

A grid point i is called an *interior* point if all the grid points in $\mu(i)$ are also in Ω; otherwise it is a *boundary* point. The set $\mu(i)$ contains 9 points in two dimensions and 27 points in three dimensions for an interior point, but has fewer points for a boundary point. Let N_I and N_B denote the numbers of interior and boundary grid points, respectively. The total number of grid points N is equal to $N_I + N_B$.

For the interior points, we shall design a matrix A of size $N_I \times N$, with the rows indexed by the interior grid points. For a fixed interior index i, the row $A(i, :)$ satisfies two conditions:

- $A(i, :)$ has support in $\mu(i)$, and
- $(AK)(i, \mu(i)^c) = A(i, \mu(i))K(\mu(i), \mu(i)^c) \approx 0$.

Because the function K is the discretization of the Green's function, we expect $A(i, :)$ to resemble a finite difference stencil of the Helmholtz operator. Figure 1 gives a schematic illustration of the structure of the matrices A, K, and AK. Once A has been calculated, we can define a sparse matrix C of the same size such that

- $C(i, :)$ has support in $\mu(i)$, and
- $C(i, \mu(i)) := A(i, \mu(i))K(\mu(i), \mu(i))$.

The above conditions imply that

- C and A are sparse operators that represent a local stencil, and
- $AK \approx C$.

Therefore,

$$A(I + K\omega^2 m) \approx (A + C\omega^2 m). \tag{7}$$

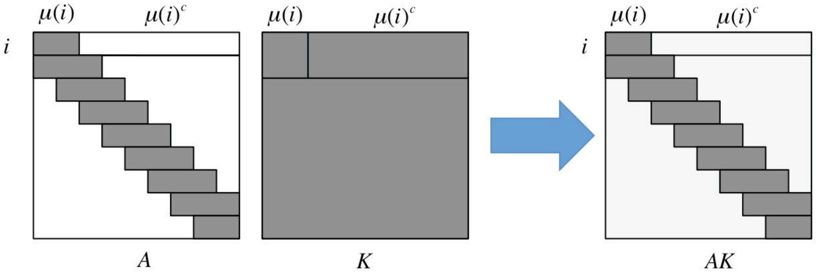

FIG. 1. *A schematic illustration of the sparsity pattern of A and AK. For a fixed i, $A(i, \mu(i)^c) = 0$ (shown in white) and $(AK)(i, \mu(i)^c) \approx 0$ (shown in light gray). The same pattern applies to B and BK as well.*

For the boundary points, we shall design a matrix B of size $N_B \times N$, with the rows indexed by the boundary grid points. For a fixed boundary index i, the row $B(i, :)$ satisfies two conditions:

- $B(i,:)$ has support in $\mu(i)$, and
- $(BK)(i,\mu(i)^c) = B(i,\mu(i))K(\mu(i),\mu(i)^c) \approx 0$.

We expect such B to exist since there exist reasonably accurate local boundary conditions for approximating the Sommerfeld radiation condition. Well-known examples include the ABC [15] and the PML [5]. In fact, B can be viewed as a numerically generated ABC. Since $m_j = 0$ for any grid point j within two layers from the boundary and i is a boundary point, $m_j = 0$ for $j \in \mu(i)$. Therefore, the second condition implies that

$$B(i,:)(K\omega^2 m)(:,:) \approx 0.$$

Hence,

$$B(I + K\omega^2 m) \approx B. \tag{8}$$

Applying matrices A and B to both sides of (6) yields

$$\begin{bmatrix} A \\ B \end{bmatrix} (I + K\omega^2 m)u = \begin{bmatrix} A \\ B \end{bmatrix} g. \tag{9}$$

Combining (7) and (8) with (9) gives

$$\begin{bmatrix} A + C\omega^2 m \\ B \end{bmatrix} u \approx \begin{bmatrix} A \\ B \end{bmatrix} g. \tag{10}$$

This suggests defining the preconditioner by

$$\tilde{u} = \begin{bmatrix} A + C\omega^2 m \\ B \end{bmatrix}^{-1} \begin{bmatrix} A \\ B \end{bmatrix} g, \tag{11}$$

where $\tilde{u}$ is clearly an approximate inverse. Since A and B are sparse, applying these operators on the right-hand side is fast. In addition, since A, B, and C are all designed to be local operators, the linear system solve in (11) can be done efficiently with, for example, the nested dissection method and the multifrontal method [14, 18].

In the following sections, we shall apply this procedure to several different settings. In each setting, we explicitly give the construction for A and B, analyze the complexity of constructing and applying the preconditioner, and provide numerical results.

3. Rectangular domains.

This section considers the case of a rectangular domain Ω.

3.1. Algorithm.

Let us consider the 2D case first. For simplicity, let $\Omega = (0,1)^2$ and define $n = 1/h$ to be the number of points in each dimension. Clearly, $N = n^2$, $N_B = O(n)$, and $N_I = O(n^2)$. In order for the operator B to serve as a sufficiently accurate approximation to the Sommerfeld radiation condition, we assume that $m(x)$ vanishes in a buffer region of a constant, but small, width near the boundary and the constant b denotes the ratio between the width and the step size.

For each interior point i, recall that we require

$$(AK)(i,j) = A(i,\mu(i))K(\mu(i),j) \approx 0 \quad \forall j \notin \mu(i).$$

Since K is translational invariant, it is convenient to require A be so as well. By translating the point i to the origin, it is sufficient to consider the problem of finding a vector α such that

$$\alpha \cdot K(\mu(0),j) \approx 0 \quad \forall j \in I_n := \{j : -n < j_1, j_2 < n, j \notin \mu(0)\}.$$

To solve for α, we formulate the following optimization problem:

$$\min_{\alpha:\|\alpha\|=1} \|\alpha \cdot K(\mu(0), I_n)\|^2. \tag{12}$$

Through the singular value decomposition $K(\mu(0), I_n) = USV^*$, the solution is given by

$$\alpha = U(:, |\mu(0)|)^*,$$

and we set $A(i, \mu(i)) = \alpha$ for all interior i.

For the boundary, there are two separate cases: edge points and corner points. For a boundary point i on an edge (suppose the right one), we require

$$(BK)(i,j) = B(i, \mu(i))K(\mu(i), j) \approx 0 \quad \forall j \notin \mu(i).$$

Translating the point i to the origin gives the sufficient problem of finding α such that

$$\alpha \cdot K(\mu_E(0), j) \approx 0 \quad \forall j \in E_n := \{j : -n < j_1 < -b, -n < j_2 < n\},$$

where $\mu_E(0)$ is the translated copy of $\mu(i)$. To solve for α, we consider

$$\min_{\alpha:\|\alpha\|=1} \|\alpha \cdot K(\mu_E(0), E_n)\|^2. \tag{13}$$

Once again, the singular value decomposition $K(\mu_E(0), E_n) = USV^*$ gives the solution

$$\alpha = U(:, |\mu_E(0)|)^*,$$

and we set $B(i, \mu(i)) = \alpha$ for all boundary i on the edges.

For a boundary point i at a corner (suppose the top right one), we require

$$B(i, \mu(i))K(\mu(i), j) \approx 0 \quad \forall j \notin \mu(i).$$

Translating the point i to the origin gives the problem of finding α such that

$$\alpha \cdot K(\mu_C(0), j) \approx 0 \quad \forall j \in C_n := \{j : -n < j_1, j_2 < -b\},$$

where $\mu_C(0)$ is the translated copy of $\mu(i)$. To solve for α, we consider

$$\min_{\alpha:\|\alpha\|=1} \|\alpha \cdot K(\mu_C(0), C_n)\|^2. \tag{14}$$

The singular value decomposition $K(\mu_C(0), C_n) = USV^*$ gives the solution

$$\alpha = U(:, |\mu_C(0)|)^*,$$

and we set $B(i, \mu(i)) = \alpha$ for all corner points i.

Once A and B are known, we compute C and form

$$\begin{bmatrix} A + C\omega^2 m \\ B \end{bmatrix}, \tag{15}$$

which is built from local compact stencils on a uniform rectangular grid. As we mentioned earlier, the nested dissection algorithm is used to compute a factorization of this operator.

The extension to three dimensions is quite straightforward, except that there are three types of boundary points: face points, edge points, and corner points. For the translated optimization problem, the point sets are defined via

- interior point

$$I_n := \{j : -n < j_1, j_2, j_3 < n, j \notin \mu(0)\},$$

- surface point (suppose at $x_1 = 1$)

$$F_n := \{j : -n < j_1 < -b, -n < j_2, j_3 < n\},$$

- edge point (suppose at $x_1 = 1, x_2 = 1$)

$$E_n := \{j : -n < j_1, j_2 < -b, -n < j_3 < n\},$$

- corner point (suppose at $x_1 = 2, x_2 = 1, x_3 = 1$)

$$C_n := \{j : -n < j_1, j_2, j_3 < -b\}.$$

3.2. Complexity. Here we analyze the cost of constructing and applying the sparsifying preconditioner.

In two dimensions, the setup algorithm consists of two parts: computing the singular value decompositions and factorizing (15) with the nested dissection algorithm. The former takes $O(n^2) = O(N)$ steps, while the latter takes $O(n^3) = O(N^{3/2})$. Therefore, the total time complexity of the setup algorithm is $O(N^{3/2})$. Applying the preconditioner is essentially a solve with the nested dissection algorithm, which has $O(n^2 \log n) = O(N \log N)$ steps. The memory cost is $O(N \log N)$ due to the contribution from storing the nested dissection factorization [18].

In three dimensions, factorizing (15) with the nested dissection algorithm has an $O(n^6) = O(N^2)$ cost, and this dominates the setup cost. The cost of applying the preconditioner is equal to $O(n^4) = O(N^{4/3})$, which is the cost of a nested dissection solve. The memory cost is also $O(N^{4/3})$ from the memory usage of the nested dissection factorization.

3.3. Numerical results. This preconditioner and the necessary nested dissection algorithm are implemented in MATLAB. The numerical results below are obtained on a desktop computer with CPU speed at 2.0Hz. For the iterative solver the GMRES algorithm is used with relative tolerance equal to 10^{-6}. The inhomogeneity $m(x)$ is confined in the unit cube, and the velocity field $c(x)$ for each example is between 2/3 and 1. The grid step size h is chosen such that there are six points per wavelength for the homogeneous region. This ensures a minimum of four points per wavelength across the whole domain.

The 2D case is tested with two examples: a Gaussian bump and a smoothed square cavity. The incoming wave is a plane wave pointing downward at frequency ω, and the results are summarized in Figures 2 and 3. The columns of the tables are the following:

- ω is the frequency,
- N is the number of unknowns,
- T_s is the setup time of the preconditioner in seconds,
- T_a is the application time of the preconditioner in seconds,
- n_p is the iteration number of the preconditioned iteration,
- T_p is the solution time of the preconditioned iteration in seconds, and
- n_u is the iteration number of the unpreconditioned iteration used for comparison. The symbol "-" means that the iterative solve fails to converge after 1000 iterations.

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
1.0e+02	9.0e+03	1.3e+00	6.6e-02	5	4.2e-01	3.4e+01
2.0e+02	3.6e+04	6.4e+00	2.6e-01	5	1.4e+00	1.0e+02
4.0e+02	1.5e+05	3.9e+01	1.0e+00	6	7.8e+00	4.5e+02

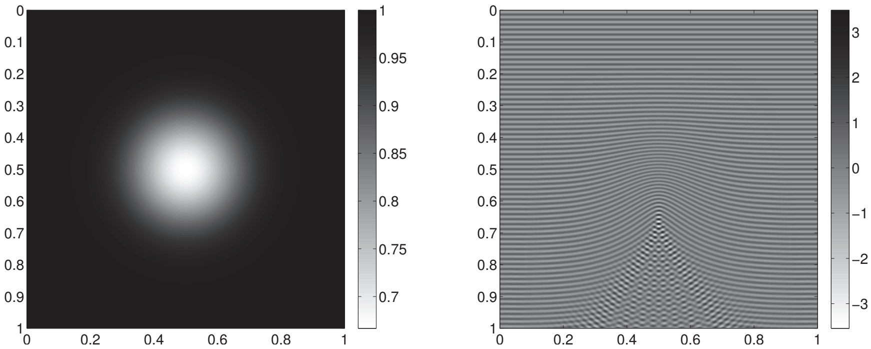

FIG. 2. *Example* 1 *of* 2D *rectangular case. Top: Numerical results. Bottom:* $c(x)$ *(left) and* $u(x) + u_I(x)$ *(right) for the largest* ω *value.*

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
1.0e+02	9.0e+03	1.3e+00	7.7e-02	6	4.3e-01	-
2.0e+02	3.6e+04	7.0e+00	4.1e-01	8	4.1e+00	-
4.0e+02	1.5e+05	4.2e+01	1.3e+00	8	1.3e+01	-

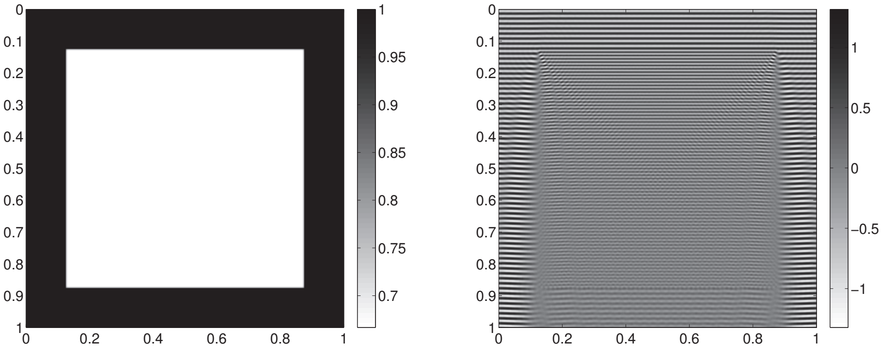

FIG. 3. *Example* 2 *of* 2D *rectangular case. Top: Numerical results. Bottom:* $c(x)$ *(left) and* $u(x) + u_I(x)$ *(right) for the largest* ω *value. The symbol "-" means that the iterative solve fails to converge after* 1000 *iterations.*

The results clearly show that, while the iterative solve with the sparsifying preconditioner typically converges in a small number of iterations, the unpreconditioned solve takes many more iterations and often fails to converge for large scale high-frequency problems.

The 3D case is also tested with two examples: a Gaussian bump and a smoothed cubic cavity. The incoming wave is again a plane wave pointing downward at frequency ω, and the results are given in Figures 4 and 5. The results show that the setup and application costs of the preconditioner scale with ω and N according to the complexity analysis given above. The preconditioner reduces the iteration number dramatically, and in fact it becomes essentially frequency-independent.

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
2.5e+01	1.2e+04	5.2e+00	6.3e-02	4	2.8e-01	1.1e+01
5.0e+01	1.0e+05	1.0e+02	6.4e-01	4	2.9e+00	1.7e+01
1.0e+02	8.6e+05	3.2e+03	7.8e+00	4	3.4e+01	3.2e+01

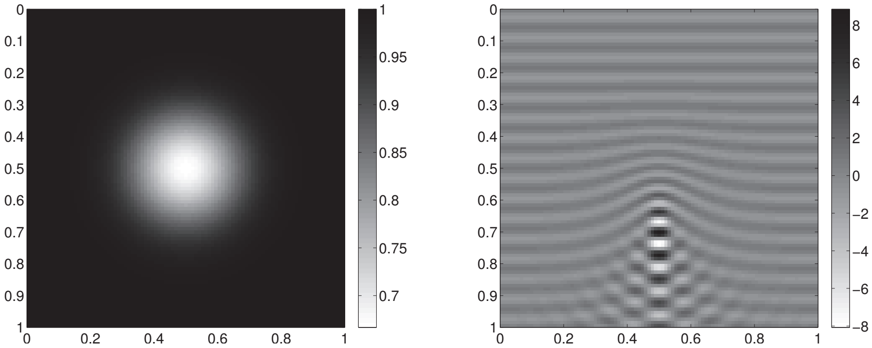

FIG. 4. *Example* 1 *of* 3D *rectangular case. Top: Numerical results. Bottom: Cross-sections of* $c(x)$ *(left) and* $u(x) + u_I(x)$ *(right) in the middle of the domain for the largest* ω *value.*

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
2.5e+01	1.2e+04	5.6e+00	7.3e-02	6	4.7e-01	1.0e+02
5.0e+01	1.0e+05	1.0e+02	5.9e-01	8	5.7e+00	-
1.0e+02	8.6e+05	2.9e+03	7.7e+00	9	7.5e+01	-

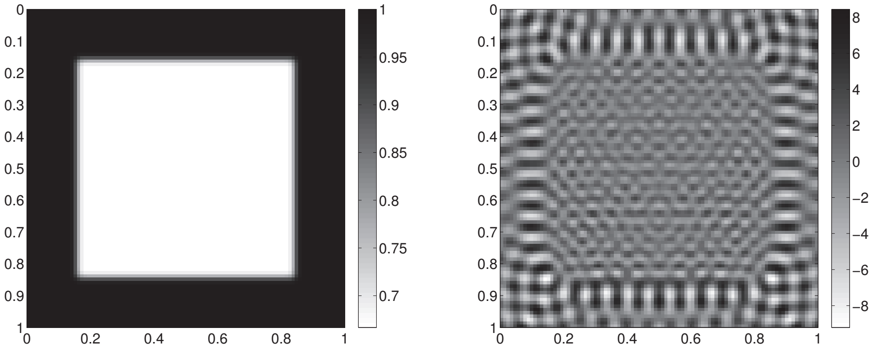

FIG. 5. *Example* 2 *of* 3D *rectangular case. Top: Numerical results. Bottom: Cross-sections of* $c(x)$ *(left) and* $u(x) + u_I(x)$ *(right) in the middle of the domain for the largest* ω *value.*

Most of the solution time T_p is spent on applying the preconditioner, since the application of the operator (6) is accelerated with the FFT. Therefore, the solution time of the current approach is roughly equal to the iteration number n_p times the solution time of the standard nearest-neighbor finite difference solver for the original PDE (1). However, as we pointed out already, the current approach has several clear advantages over the standard finite difference method: it avoids the pollution effect, treats the Sommerfeld radiation condition exactly, and provides much more accurate results.

4. General domains. For most problems, the support Ω of $m(x)$ is not necessarily rectangular. We can apply the approach in section 3 by embedding Ω into a rectangular domain. However, this approach increases the number of unknowns significantly, especially in three dimensions. This section proposes a different approach that does not suffer from this.

4.1. Algorithm. The construction of the operator A for the interior grid points is exactly the same as the one in section 3. The main difference is in the construction of B. For a boundary point i, we require

$$B(i,\mu(i))(K\omega^2 m)(\mu(i),:) \approx 0.$$

Since the local geometry at i can be quite different from point to point, it is impossible to find a uniform stencil $B(i,\mu(i))$ as was done in section 3. Therefore, one needs to consider i point by point. However, applying the approach of section 3 to each i individually is too costly since each singular value decomposition costs $O(N)$ steps. To keep the complexity under control, we propose the following randomized approach.

Let R be a Gaussian random matrix of size $N \times r$, where $r = O(1)$ is a constant multiple of the maximum stencil size (i.e., 9 in two dimensions and 27 in three dimensions), and define the $N \times r$ matrix T via

$$T = (K\omega^2 m)R.$$

For each boundary i, we look for $B(i,\mu(i))$ such that

$$B(i,\mu(i))T(\mu(i),:) \approx 0.$$

This is done by solving the optimization problem

$$\min_{\alpha:\|\alpha\|=1} \|\alpha \cdot T(\mu(i),:)\|^2.$$

With the singular value decomposition $T(\mu(i),:) = USV^*$, the optimal α is given by

$$\alpha = U(:,|\mu(i)|)^*.$$

Finally, we set $B(i,\mu(i)) = \alpha$. This process is repeated for each boundary point i. Since $T(\mu(i),:)$ is of size $O(1) \times O(1)$, the computational cost of the singular value decomposition setup for each i is also of order $O(1)$.

The effectiveness of this preconditioner depends on the assumption that there exists a good local stencil in $\mu(i)$ for a boundary point i. When the domain is convex or nearly convex, such a stencil is guaranteed due to the absorbing-type boundary conditions. However, when the domain is highly nonconvex, the preconditioner is less effective.

4.2. Complexity. To analyze the cost of this approach, we assume that the size of the support of $m(x)$ is more than a constant fraction of the size of its bounding box, which implies that $N = O(n^d)$.

In two dimensions, the setup algorithm consists of three parts: (i) forming $T = (K\omega^2 m)R$, (ii) computing a singular value decomposition for each boundary point i, and (iii) factorizing (15) with the nested dissection algorithm.

- The first part can be accomplished in $O(N \log N)$ steps by using the FFT.
- The second part takes $O(\sqrt{N})$ steps since there are at most $O(\sqrt{N})$ boundary points and computing a singular value decomposition for each one takes $O(1)$ steps.

- Finally, factorizing the sparse system with the nested dissection algorithm takes $O(N^{3/2})$ steps.

Adding these together shows that the construction cost is $O(N^{3/2})$. The application cost of the preconditioner is $O(N \log N)$ since it is essentially a solve with an existing 2D nested dissection factorization. Similar to the rectangular case, the memory cost in the 2D case is still $O(N \log N)$.

In three dimensions, among the three parts of the setup algorithm, only the cost of computing a nested dissection factorization increases to $O(N^2)$. This now becomes the dominant part of the setup cost. The application algorithm is again a solve with the existing nested dissection factorization with a cost of $O(N^{4/3})$. For a typical 3D problem, the memory cost scales as $O(N^{4/3})$ due to the storage of the nested dissection factorization.

Though the asymptotic complexity obtained for this case is similar to that of the rectangular case, the actual running time is often lower because the size of the supernodes in the nested dissection algorithm is often much smaller.

4.3. Numerical results. The 2D case is tested with two examples: a smoothed cavity of an ℓ_2 ball and a smoothed cavity of an ℓ_1 ball. The incoming wave is a plane wave pointing downward at frequency ω, and the numerical results are given in Figures 6 and 7.

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
1.0e+02	7.6e+03	9.8e-01	4.1e-02	6	2.4e-01	-
2.0e+02	3.0e+04	3.7e+00	1.3e-01	7	1.2e+00	-
4.0e+02	1.2e+05	2.1e+01	5.1e-01	8	6.5e+00	-

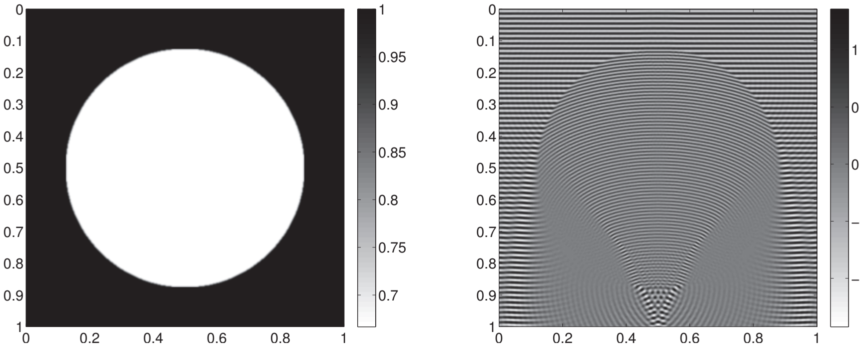

FIG. 6. *Example 1 of 2D general case. Top: Numerical results. Bottom: $c(x)$ (left) and $u(x) + u_I(x)$ (right) for the largest ω value.*

Two examples are tested for the 3D case: a smoothed cavity of an ℓ_2 ball and a smoothed cavity of an ℓ_1 ball. The incoming wave is a plane wave pointing downward at frequency ω, and the numerical results are given in Figures 8 and 9.

The results show that the setup and application costs of the preconditioner are significantly lower compared to those of the rectangular case since the number of unknowns is smaller. The number of iterations remains essentially independent of the frequency, indicating that the boundary stencils computed randomly provide a sufficiently accurate boundary condition for the solution.

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
1.0e+02	5.7e+03	7.4e-01	3.3e-02	5	2.0e-01	9.2e+02
2.0e+02	2.1e+04	2.8e+00	7.6e-02	8	1.1e+00	-
4.0e+02	7.9e+04	1.8e+01	2.9e-01	8	5.3e+00	-

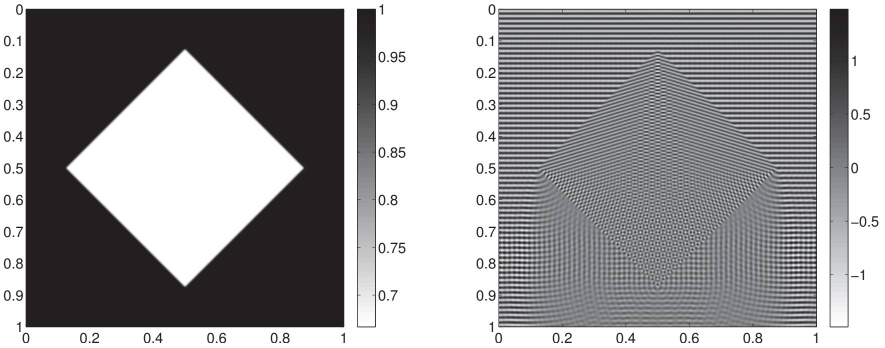

FIG. 7. *Example 2 of 2D general case. Top: Numerical results. Bottom: $c(x)$ (left) and $u(x) + u_I(x)$ (right) for the largest ω value.*

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
2.5e+01	6.5e+03	7.1e+00	2.7e-02	4	1.6e-01	2.4e+01
5.0e+01	5.5e+04	7.9e+01	3.4e-01	5	2.1e+00	-
1.0e+02	4.5e+05	1.4e+03	4.1e+00	5	2.3e+01	-

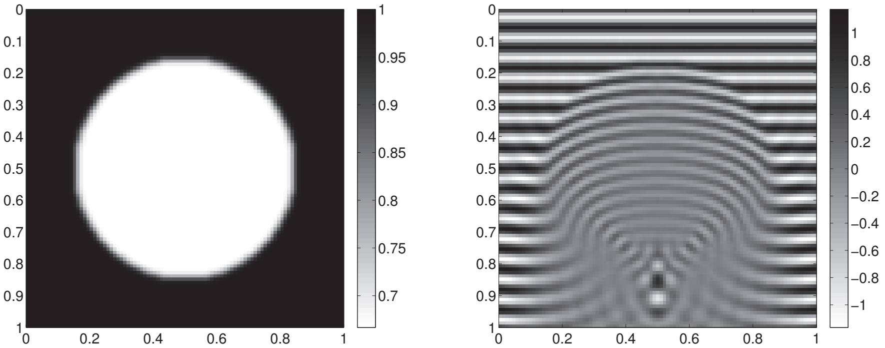

FIG. 8. *Example 1 of 3D general case. Top: Numerical results. Bottom: Cross-sections of $c(x)$ (left) and $u(x) + u_I(x)$ (right) in the middle of the domain for the largest ω value.*

5. Laplace equation.

In this section, we consider the 3D Laplace equation with compact potential perturbation, i.e.,

$$(-\Delta + V(x))u(x) = f(x), \quad x \in \mathrm{R}^3, \tag{16}$$

where $V(x)$ is supported in $\Omega = (0, 1)^3$ and $u(x)$ decays like $1/|x|$ as x goes to infinity. This can be regarded as a limiting case as ω goes to zero. The Green's function is given by

$$G(x) = \frac{1}{4\pi|x|}.$$

ω	N	T_s(sec)	T_a(sec)	n_p	T_p(sec)	n_u
2.5e+01	3.7e+03	4.9e+00	3.0e-02	4	1.0e-01	1.4e+01
5.0e+01	3.1e+04	4.5e+01	1.8e-01	6	1.4e+00	9.4e+01
1.0e+02	2.3e+05	7.5e+02	2.1e+00	9	2.3e+01	1.7e+02

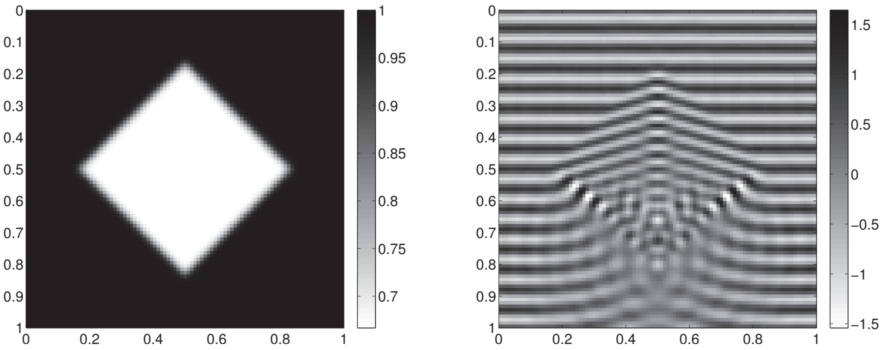

FIG. 9. *Example* 2 *of* 3*D general case. Top: Numerical results. Bottom: Cross-sections of* $c(x)$ *(left) and* $u(x) + u_I(x)$ *(right) in the middle of the domain for the largest* ω *value.*

Convolving $G(x)$ with (16) gives the Lippmann–Schwinger equation

$$u + G * (Vu) = G * f. \tag{17}$$

For a potential V that is not too negative, this problem remains elliptic, and standard iteration methods such as GMRES converge within a small number of iterations. However, when $V(x)$ is sufficiently negative, the problem loses ellipticity, and the iteration number increases dramatically.

As we mentioned, when ω is finite, PML and ABC offer reasonable approximations to the Sommerfeld radiation condition. When ω goes to zero, it is not clear how to define analytically a similar local boundary condition at a finite distance for the $1/|x|$ decaying condition. However, the sparsifying preconditioner proposed in sections 3 and 4 can be applied to (17) without any modification, and the resulting operator B offers a good numerical approximation to the $1/|x|$ decaying condition.

5.1. Numerical results. The tests are performed for two examples of $V(x)$: a negative Gaussian function and a smoothed characteristic function of an ℓ_2 ball. In each example, the right-hand side f is chosen to be a delta source in Ω located near the top left corner of the middle cross-section. The potential $V(x)$ is chosen to be negative and proportional to $1/h^2$ in order to make sure that the problem loses ellipticity. The numerical results are summarized in Figures 10 and 11.

The results demonstrate that the number of iterations grows at most logarithmically with the problem size. This implies that the numerically computed boundary condition operator B is a good approximation to the decaying condition of u at infinity.

6. Conclusion.

This paper introduces the sparsifying preconditioners for the numerical solution of the Lippmann–Schwinger equation. The main idea is to numerically transform this integral equation into a sparse and localized linear system and then leverage existing efficient sparse linear algebra algorithms. This approach combines the appealing features of the integral equation formulation with the efficiency

$\max(\lvert V\rvert)$	N	$T_s(\mathrm{sec})$	$T_a(\mathrm{sec})$	n_p	$T_p(\mathrm{sec})$	n_u
5.8e+02	1.2e+04	5.0e+00	4.9e-02	5	2.1e-01	1.5e+01
2.3e+03	1.0e+05	9.8e+01	2.8e-01	5	1.8e+00	5.6e+01
9.2e+03	8.6e+05	2.1e+03	3.0e+00	6	2.1e+01	-

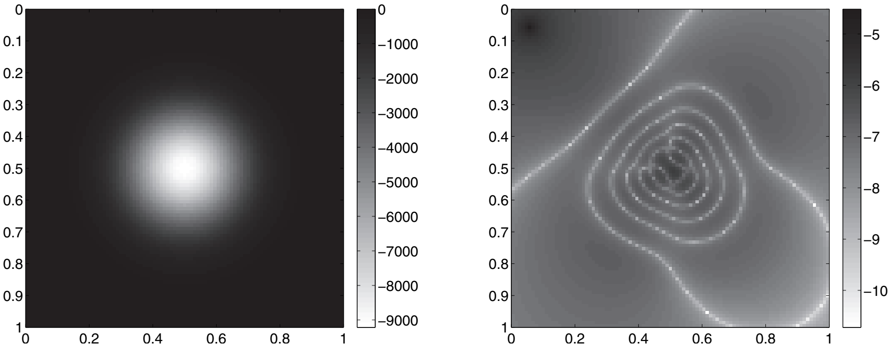

FIG. 10. *Example* 1 *of* 3D *Laplace equation with potential perturbation. Top: Numerical results. Bottom: Cross-sections of* $V(x)$ *(left) and* $\log_{10}|u(x)|$ *(right) in the middle of the domain for the highest resolution test.*

$\max(\lvert V\rvert)$	N	$T_s(\mathrm{sec})$	$T_a(\mathrm{sec})$	n_p	$T_p(\mathrm{sec})$	n_u
5.8e+02	1.2e+04	5.1e+00	3.8e-02	5	2.9e-01	2.6e+01
2.3e+03	1.0e+05	9.8e+01	2.6e-01	7	2.3e+00	-
9.2e+03	8.6e+05	2.0e+03	2.8e+00	7	2.4e+01	-

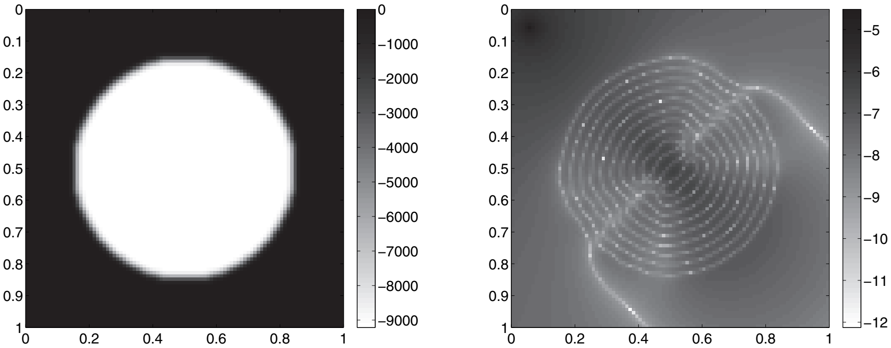

FIG. 11. *Example* 2 *of* 3D *Laplace equation with potential perturbation. Top: Numerical results. Bottom: Cross-sections of* $V(x)$ *(left) and* $\log_{10}|u(x)|$ *(right) in the middle of the domain for the highest resolution test.*

of the PDE formulations. We discuss the algorithmic details for rectangular and general domains and extend this approach to the 3D Laplace equation with a potential perturbation.

From the numerical results, we observe that most of the construction time of the preconditioners is spent on the construction of the nested dissection factorization for the operator in (11). It is possible to replace this step with other solvers such as the sweeping preconditioners [16, 17], which are asymptotically more efficient.

Most of the discussion here is in the setting of inhomogeneous acoustic scattering. An immediate task is to use this preconditioner to study problems from electromagnetic and quantum scattering.

This approach is not restricted only to scattering in free space. A future direction is to extend it to fully periodic systems or systems that are periodic in certain directions. This should have direct applications in the study of photonic crystals.

For most numerical methods, the stencil is defined analytically based on the PDE. In the current approach, however, the stencils in the preconditioning operators A and B are defined using the typical behavior of the solution, either deterministically as in section 3 or randomly as in section 4. In general, such a practice seems to be fairly unexplored in the field of numerical analysis.

Acknowledgments. The author thanks Lenya Ryzhik for providing computing resources and Anil Damle for providing feedback on an early draft. The author also thanks the anonymous reviewers for constructive comments.

REFERENCES

[1] S. Ambikasaran and E. Darve, *An $O(N \log N)$ fast direct solver for partial hierarchically semi-separable matrices: With application to radial basis function interpolation*, J. Sci. Comput., 57 (2013), pp. 477–501.

[2] F. Andersson and A. Holst, *A fast, bandlimited solver for scattering problems in inhomogeneous media*, J. Fourier Anal. Appl., 11 (2005), pp. 471–487.

[3] I. M. Babuška and S. A. Sauter, *Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?*, SIAM J. Numer. Anal., 34 (1997), pp. 2392–2423.

[4] M. Bebendorf, *Hierarchical Matrices. A Means to Efficiently Solve Elliptic Boundary Value Problems*, Lect. Notes Comput. Sci. Eng. 63, Springer-Verlag, Berlin, 2008.

[5] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200.

[6] G. Beylkin, C. Kurcz, and L. Monzón, *Fast algorithms for Helmholtz Green's functions*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 464 (2008), pp. 3301–3326.

[7] G. Beylkin, C. Kurcz, and L. Monzón, *Fast convolution with the free space Helmholtz Green's function*, J. Comput. Phys., 228 (2009), pp. 2770–2791.

[8] O. P. Bruno and E. M. Hyde, *An efficient, preconditioned, high-order solver for scattering by two-dimensional inhomogeneous media*, J. Comput. Phys., 200 (2004), pp. 670–694.

[9] Y. Chen, *A fast, direct algorithm for the Lippmann-Schwinger integral equation in two dimensions. Modeling and computation in optics and electromagnetics*, Adv. Comput. Math., 16 (2002), pp. 175–190.

[10] W. C. Chew and W. H. Weedon, *A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates*, Microwave Opt. Technol. Lett., 7 (1994), pp. 599–604.

[11] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp., 19 (1965), pp. 297–301.

[12] 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.

[13] R. Duan and V. Rokhlin, *High-order quadratures for the solution of scattering problems in two dimensions*, J. Comput. Phys., 228 (2009), pp. 2152–2174.

[14] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[15] B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp., 31 (1977), pp. 629–651.

[16] B. Engquist and L. Ying, *Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation*, Comm. Pure Appl. Math., 64 (2011), pp. 697–735.

[17] B. Engquist and L. Ying, *Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers*, Multiscale Model. Simul., 9 (2011), pp. 686–710.

[18] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.

[19] A. Gillman, A. H. Barnett, and P.-G. Martinsson, *A spectrally accurate direct solution technique for frequency-domain scattering problems with variable media*, BIT, 55 (2015), pp. 141–170.

[20] K. L. Ho and L. Greengard, *A fast direct solver for structured linear systems by recursive skeletonization*, SIAM J. Sci. Comput., 34 (2012), pp. A2507–A2532.

[21] K. L. Ho and L. Ying, *Hierarchical Interpolative Factorization for Elliptic Operators: Integral Equations*, preprint, arXiv:1307.2666v3 [math.NA], 2015.

[22] F. Lanzara, V. Maz'ya, and G. Schmidt, *Numerical solution of the Lippmann-Schwinger equation by approximate approximations*, J. Fourier Anal. Appl., 10 (2004), pp. 645–660.

[23] J. Sifuentes, *Preconditioned Iterative Methods for Inhomogeneous Acoustic Scattering Applications*, Ph.D. thesis, Rice University, Houston, TX, 2010.

[24] G. Vainikko, *Fast solvers of the Lippmann-Schwinger equation*, in Direct and Inverse Problems of Mathematical Physics (Newark, DE, 1997), Int. Soc. Anal. Appl. Comput. 5, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 423–440.

[25] 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.