MULTISCALE MODEL. SIMUL.
Vol. 14, No. 2, pp. 799–822
© 2016 Society for Industrial and Applied Mathematics

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ADDITIVE SWEEPING PRECONDITIONER FOR THE HELMHOLTZ EQUATION*

FEI LIU† AND LEXING YING‡

Abstract. We introduce a new additive sweeping preconditioner for the Helmholtz equation based on the perfectly matched layer (PML). This method divides the domain of interest into thin layers and proposes a new transmission condition between the subdomains where the emphasis is on the boundary values of the intermediate waves. This approach can be viewed as an effective approximation of an additive decomposition of the solution operator. When combined with the standard GMRES solver, the iteration number is essentially independent of the frequency. Several numerical examples are tested to show the efficiency of this new approach.

Key words. Helmholtz equation, perfectly matched layers, preconditioners, high frequency waves

AMS subject classifications. 65F08, 65N22, 65N80

DOI. 10.1137/15M1017144

1. Introduction.

Let the domain of interest be $D = (0,1)^d$, where $d = 2, 3$. The Helmholtz equation is

$$\Delta u(x) + \frac{\omega^2}{c^2(x)} u(x) = f(x) \quad \forall x \in D, \tag{1}$$

where $u(x)$ is the time-independent wave field generated by the time-independent force $f(x)$, ω is the angular frequency, and $c(x)$ is the velocity field. Commonly used boundary conditions are the approximations of the Sommerfeld radiation condition. By rescaling the system, we assume $c_{\min} \le c(x) \le c_{\max}$, where $c_{\min}$ and $c_{\max}$ are of $\Theta(1)$. Then $\omega/(2\pi)$ is the typical wave number and $\lambda = 2\pi/\omega$ is the typical wavelength.

Solving the equation numerically is challenging in high frequency settings for two reasons. First, in most applications, the equation is discretized with at least a constant number of points per wavelength, which makes the number of points in each direction $n = \Omega(\omega)$ and the total degree of freedom $N = n^d = \Omega(\omega^d)$ very large. Second, the system is highly indefinite and has a very oscillatory Green's function, which makes most of the classical iterative methods no longer effective.

There has been a sequence of papers on developing iterative methods for solving (1). The AILU method by Gander and Nataf [10] is the first to use the incomplete LU factorization to precondition the equation. Engquist and Ying [6, 7] developed a series of sweeping preconditioners based on approximating the inverse of the Schur complements in the LDU factorization and obtained essentially ω-independent iteration numbers. In [15], Stolk proposed a domain decomposition method based on

*Received by the editors April 16, 2015; accepted for publication (in revised form) March 2, 2016; published electronically May 12, 2016. This work was partially supported by the National Science Foundation under awards DMS-1328230 and DMS-1521830 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/14-2/M101714.html

†Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (liufei@stanford.edu).

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

the perfectly matched layer (PML) which constructs delicate transmission conditions between the subdomains by considering the "pulses" generated by the intermediate waves. In [19], Vion and Geuzaine proposed a double sweep preconditioner based on the Dirichlet-to-Neumann (DtN) map, and several numerical simulations of the DtN map were compared. In [2, 3], Chen and Xiang introduced a source transfer domain decomposition method which emphasizes transferring the sources between the subdomains. In [20], Zepeda-Núñez and Demanet developed a novel domain decomposition method for the two-dimensional (2D) case by pairing up the waves and their normal derivatives at the boundary of the subdomains and splitting the transmission of the waves into two directions. Most recently, in [13], Liu and Ying proposed a recursive sweeping preconditioner for three-dimensional (3D) Helmholtz problems. Other progresses include [14, 18, 16, 17], and we refer the reader to Erlangga [8] and Ernst and Gander [9] for a complete discussion.

Inspired by [15] and these previous approaches, we propose a new domain decomposition method in this paper which shares some similarities with [7, 15]. The novelty of this new approach is that the transmission conditions are built with the boundary values of the intermediate waves directly. For each wave field on the subdomains, we divide it into three parts—the waves generated by the force to the left of the subdomain, to the right of the subdomain, and within the subdomain itself. This corresponds to an $L + D + U$ decomposition of the Green's matrix G as the sum of its lower triangular part, upper triangular part, and diagonal part. This is why we call this new preconditioner the additive sweeping preconditioner.

The rest of this paper is organized as follows. First in section 2 we use the one-dimensional (1D) case to illustrate the idea of the method. Then in section 3 we introduce the preconditioner in two dimensions and present the 2D numerical results. Section 4 discusses the 3D case. Conclusions and some future directions are provided in section 5.

2. 1D illustration. We use the PML [1, 4, 12] to simulate the Sommerfeld condition. The PML introduces the auxiliary functions

$$\sigma(x) := \begin{cases} \dfrac{C}{\eta}\left(\dfrac{x-\eta}{\eta}\right)^2, & x \in [0,\eta), \\ 0, & x \in [\eta, 1-\eta], \\ \dfrac{C}{\eta}\left(\dfrac{x-1+\eta}{\eta}\right)^2, & x \in (1-\eta, 1], \end{cases}$$

$$s(x) := \left(1 + \mathrm{i}\frac{\sigma(x)}{\omega}\right)^{-1},$$

where C is an appropriate positive constant independent of ω, and η is the PML width which is typically around one wavelength.

The Helmholtz equation with PML in one dimension is

$$\begin{cases} \left(\left(s(x)\dfrac{\mathrm{d}}{\mathrm{d}x}\right)^2 + \dfrac{\omega^2}{c^2(x)}\right) u(x) = f(x) \quad \forall x \in (0,1), \\ u(0) = 0, \\ u(1) = 0. \end{cases}$$

We discretize the system with step size $h = 1/(n+1)$; then n is the degree of freedom.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

With the standard central difference numerical scheme the discretized equation is

$$\frac{s_i}{h}\left(\frac{s_{i+1/2}}{h}(u_{i+1}-u_i)-\frac{s_{i-1/2}}{h}(u_i-u_{i-1})\right)+\frac{\omega^2}{c_i^2}u_i=f_i \quad \forall 1\le i\le n, \tag{2}$$

where the subscript i means that the corresponding function is evaluated at $x=ih$.

We denote (2) as $A\boldsymbol{u}=\boldsymbol{f}$, where $\boldsymbol{u}$ and $\boldsymbol{f}$ are the discrete array of the wave field and the force

$$\boldsymbol{u}:=[u_1,\dots,u_n]^T, \quad \boldsymbol{f}:=[f_1,\dots,f_n]^T.$$

In one dimension, A is tridiagonal and (2) can be solved without any difficulty. However, here we are aiming at an approach which can be generalized to higher dimensions, so the rest of this section takes another point of view to solve (2) instead of exploiting the sparsity structure of A directly.

With the Green's matrix $G=A^{-1}$, $\boldsymbol{u}$ can be written as $\boldsymbol{u}=G\boldsymbol{f}$. Now let us divide the discrete grid into m parts. We assume that $\eta=\gamma h$ and $n=2\gamma+mb-2$, where γ and b are some small constants and m is comparable to n, and we define

$$\begin{aligned}
X_1&:=\{ih: 1\le i\le \gamma+b-1\},\\
X_p&:=\{ih: \gamma+(p-1)b\le i\le \gamma+pb-1\}, \quad p=2,\dots,m-1,\\
X_m&:=\{ih: \gamma+(m-1)b\le i\le 2\gamma+mb-2\},
\end{aligned}$$

which means that X_1 is the leftmost part containing the left PML of the original problem and a small piece of grid with b points, X_m is the rightmost part containing the right PML and a grid of b points, and X_p, $p=2,\dots,m-1$, are the middle parts, each of which contains b points. $\boldsymbol{u}_p$ and $\boldsymbol{f}_p$ are defined as the restrictions of $\boldsymbol{u}$ and $\boldsymbol{f}$ on X_p for $p=1,\dots,m$, respectively,

$$\begin{aligned}
\boldsymbol{u}_1&:=[u_1,\dots,u_{\gamma+b-1}]^T,\\
\boldsymbol{u}_p&:=[u_{\gamma+(p-1)b},\dots,u_{\gamma+pb-1}]^T, \quad p=2,\dots,m-1,\\
\boldsymbol{u}_m&:=[u_{\gamma+(m-1)b},\dots,u_{2\gamma+mb-2}]^T,\\
\boldsymbol{f}_1&:=[f_1,\dots,f_{\gamma+b-1}]^T,\\
\boldsymbol{f}_p&:=[f_{\gamma+(p-1)b},\dots,f_{\gamma+pb-1}]^T, \quad p=2,\dots,m-1,\\
\boldsymbol{f}_m&:=[f_{\gamma+(m-1)b},\dots,f_{2\gamma+mb-2}]^T.
\end{aligned}$$

Then $u=Gf$ can be written as

$$\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\\ \vdots\\ \boldsymbol{u}_m\end{bmatrix}=\begin{bmatrix}G_{1,1} & G_{1,2} & \dots & G_{1,m}\\ G_{2,1} & G_{2,2} & \dots & G_{2,m}\\ \vdots & \vdots & & \vdots\\ G_{m,1} & G_{m,2} & \dots & G_{m,m}\end{bmatrix}\begin{bmatrix}\boldsymbol{f}_1\\ \boldsymbol{f}_2\\ \vdots\\ \boldsymbol{f}_m\end{bmatrix}.$$

By introducing $\boldsymbol{u}_{p,q}:=G_{p,q}\boldsymbol{f}_q$ for $1\le p,q\le m$, one can write $\boldsymbol{u}_p=\sum_{q=1}^m \boldsymbol{u}_{p,q}$. The physical meaning of $\boldsymbol{u}_{p,q}$ is the contribution of the force $\boldsymbol{f}_q$ defined on the grid X_q acting upon the grid X_p. If we know the matrix G, the computation of $\boldsymbol{u}_{p,q}$ can be carried out directly. However, computing G, or even applying G to the vector $\boldsymbol{f}$, is computationally expensive. The additive sweeping method circumvents this difficulty by approximating the blocks of G sequentially, and the idea works in higher dimensions. In what follows, we shall use $\tilde{\boldsymbol{u}}_{p,q}$ to denote the approximations of $\boldsymbol{u}_{p,q}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.1. Approximating $\boldsymbol{u}_{p,q}$ with auxiliary PMLs.

2.1.1. Wave generated by $\boldsymbol{f}_1$. The components $\boldsymbol{u}_{p,1}$ for $p = 1, \ldots, m$ can be regarded as a sequence of right-going waves generated by $\boldsymbol{f}_1$. Note that the boundary condition of the system is the approximated Sommerfeld condition. If we assume that the reflection during the transmission of the wave is negligible, then, to approximate $\boldsymbol{u}_{1,1}$, we can simply put an artificial PML on the right of the grid X_1 to solve a much smaller problem, since the domain of interest here is only X_1 (see Figure 2(b)). To be precise, we define

$$\sigma_1^M(x) := \begin{cases} \dfrac{C}{\eta}\left(\dfrac{x-\eta}{\eta}\right)^2, & x \in [0,\eta), \\ 0, & x \in [\eta, \eta + (b-1)h], \\ \dfrac{C}{\eta}\left(\dfrac{x-(\eta+(b-1)h)}{\eta}\right)^2, & x \in (\eta+(b-1)h, 2\eta+(b-1)h], \end{cases}$$

$$s_1^M(x) := \left(1 + \mathrm{i}\frac{\sigma_1^M(x)}{\omega}\right)^{-1}.$$

We consider a subproblem on the auxiliary domain $D_1^M := (0, 2\eta + (b-1)h)$:

$$\begin{cases} \left(\left(s_1^M(x)\dfrac{\mathrm{d}}{\mathrm{d}x}\right)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = g(x) & \forall x \in D_1^M, \\ v(x) = 0 & \forall x \in \partial D_1^M. \end{cases}$$

With the same discrete numerical scheme and step size h, we have the corresponding discrete system $H_1^M \boldsymbol{v} = \boldsymbol{g}$ on the extended grid

$$X_1^M := \{ih : 1 \le i \le 2\gamma + b - 2\}.$$

Figure 1 shows a graphical view of X_1^M, as well as other extended grids which we will see later.

With the discrete system $H_1^M \boldsymbol{v} = \boldsymbol{g}$, we can define an operator $\tilde{G}_1^M : \boldsymbol{y} \to \boldsymbol{z}$, which is an approximation of $G_{1,1}$, by the following:

1. Introduce a vector $\boldsymbol{g}$ defined on X_1^M by setting $\boldsymbol{y}$ to X_1 and zero everywhere else.
2. Solve $H_1^M \boldsymbol{v} = \boldsymbol{g}$ on X_1^M.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_1.

Then $\tilde{\boldsymbol{u}}_{1,1}$ can be set as

$$\tilde{\boldsymbol{u}}_{1,1} := \tilde{G}_1^M \boldsymbol{f}_1.$$

Once we have computed $\tilde{\boldsymbol{u}}_{1,1}$, we can use the right boundary value of $\tilde{\boldsymbol{u}}_{1,1}$ to compute $\tilde{\boldsymbol{u}}_{2,1}$ by introducing an auxiliary PML on the right of X_2 and solving the boundary value problem with the left boundary value at $x = (\gamma + b - 1)h$ equal to the right boundary value of $\tilde{\boldsymbol{u}}_{1,1}$. The same process can be repeated to compute $\tilde{\boldsymbol{u}}_{p+1,1}$ by exploiting the right boundary value of $\tilde{\boldsymbol{u}}_{p,1}$ recursively for $p = 2, \ldots, m-1$ (see Figure 2(c)). In the following context of this section, we introduce notations g^L, g^R for a vector array $\boldsymbol{g} = [g_1, \ldots, g_s]^T$ by

$$g^L := g_1, \quad g^R := g_s,$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

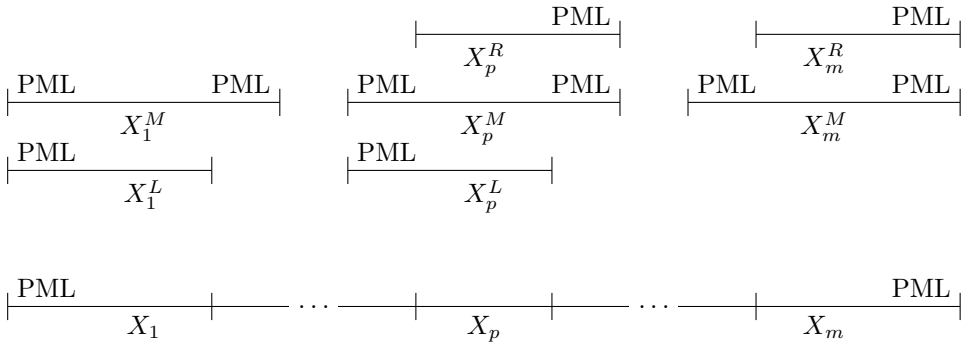


FIG. 1. *An illustration of how the grids X_p are extended with auxiliary PMLs.*

where g^L and g^R should be interpreted as the leftmost and the rightmost element of the array $\boldsymbol{g}$.

To formalize the definition of $\tilde{\boldsymbol{u}}_{p,1}$ for each $p = 2, \ldots, m$, we introduce the auxiliary domain D_p^R, which will be defined below, to simulate the right-transmission of the waves. The superscript R means that the auxiliary domain is intended for approximating the right-going waves. The left boundary of D_p^R will be denoted as $\partial^L D_p^R$, on which the boundary value will be used to approximate the wave transmission, as we shall see. We also extend X_p with an auxiliary PML on the right to form an extended grid X_p^R (see Figure 1), which corresponds the discretization of D_p^R. To be specific, we define

$$\begin{aligned} D_p^R &:= (\eta + ((p-1)b-1)h, 2\eta + (pb-1)h), \\ \partial^L D_p^R &:= \{\eta + ((p-1)b-1)h\}, \\ X_p^R &:= \{ih : \gamma + (p-1)b \le i \le 2\gamma + pb - 2\}. \end{aligned}$$

Note that the grid X_m^R is X_m itself, since X_m already contains the original right PML region. The purpose of using the notation X_m^R is to simplify the description of the algorithm.

For the PML on D_p^R, we define

$$\sigma_p^R(x) := \begin{cases} 0, & x \in [\eta + ((p-1)b-1)h, \eta + (pb-1)h], \\ \dfrac{C}{\eta}\left(\dfrac{x - (\eta + (pb-1)h)}{\eta}\right)^2, & x \in (\eta + (pb-1)h, 2\eta + (pb-1)h], \end{cases}$$

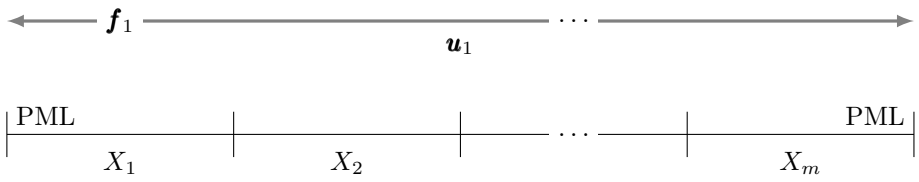
$$s_p^R := \left(1 + \mathrm{i}\frac{\sigma_p^R(x)}{\omega}\right)^{-1}$$

and consider the following subproblem:

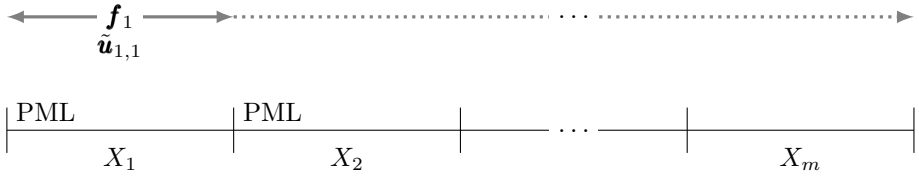
$$\begin{cases} \left((s_p^R(x)\dfrac{\mathrm{d}}{\mathrm{d}x})^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = 0 & \forall x \in D_p^R, \\ v(x) = w & \forall x \in \partial^L D_p^R, \\ v(x) = 0 & \forall x \in \partial D_p^R \setminus \partial^L D_p^R, \end{cases}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

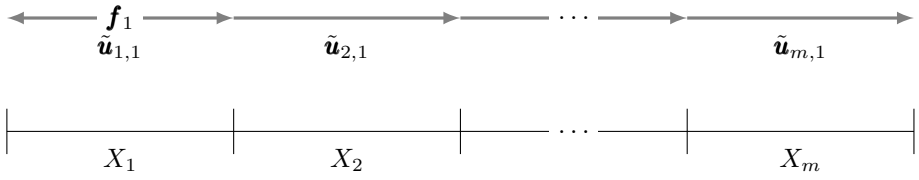
Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



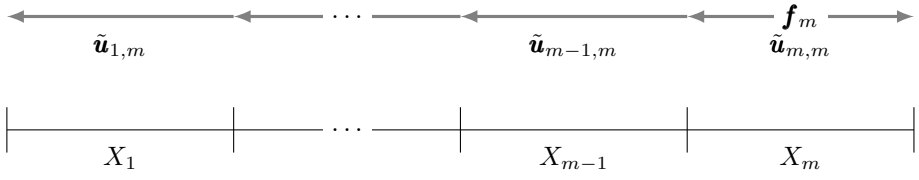
(a) The wave $\boldsymbol{u}_1$ (shown as the gray arrow) generated by $\boldsymbol{f}_1$.



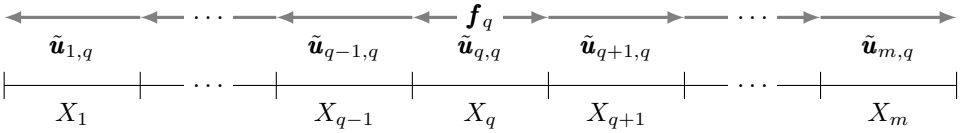
(b) $\tilde{\boldsymbol{u}}_{1,1}$ is computed by introducing an auxiliary PML on the right of X_1. The dotted gray arrow stands for the restriction of $\boldsymbol{u}_1$ on $X_2 \cup \cdots \cup X_m$, which is to be approximated.



(c) $\tilde{\boldsymbol{u}}_{p,1}$ for $p = 2, \ldots, m$ are computed sequentially.



(d) $\tilde{\boldsymbol{u}}_{p,m}$ for $p = m, \ldots, 1$ are computed sequentially.



(e) $\tilde{\boldsymbol{u}}_{p,q}$ are computed for $p = q$ first, and then for $p = q+1, \ldots, m$ and for $p = q-1, \ldots, 1$ sequentially.

FIG. 2. *An illustration of how $\tilde{\boldsymbol{u}}_{p,q}$ are generated. The direction of the arrows indicates the computing orders of the approximating waves.*

where w is the left boundary value of the unknown $v(x)$. We define $H_p^R \boldsymbol{v} = \boldsymbol{g}$ as the discretization of this problem on X_p^R where the right-hand side $\boldsymbol{g}$ is given by $\boldsymbol{g} := (-1/h^2)[w, 0, \ldots, 0]^T$ as a result of the central discretization. The subproblem $H_p^R \boldsymbol{v} = \boldsymbol{g}$ for each $p = 2, \ldots, m$ induces the approximation operator $\tilde{G}_p^R : w \to \boldsymbol{z}$ by the following procedure:

1. Set $\boldsymbol{g} = (-1/h^2)[w, 0, \ldots, 0]^T$.
2. Solve $H_p^R \boldsymbol{v} = \boldsymbol{g}$ on X_p^R.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_p.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Then $\tilde{\boldsymbol{u}}_{p,1}$ can be defined recursively for $p = 2, \ldots, m$ by

$$\tilde{\boldsymbol{u}}_{p,1} := \tilde{G}_p^R \tilde{\boldsymbol{u}}_{p-1,1}^R.$$

Note that the operator $\tilde{G}_p^R$ is not an approximation of the matrix block $G_{p,1}$, since $\tilde{G}_p^R$ maps the right boundary value of $\tilde{\boldsymbol{u}}_{p-1,1}$ to $\tilde{\boldsymbol{u}}_{p,1}$ while $G_{p,1}$ maps $\boldsymbol{f}_1$ to $\boldsymbol{u}_{p,1}$.

2.1.2. Wave generated by $\boldsymbol{f}_m$. The components $\boldsymbol{u}_{p,m}$ for $p = 1, \ldots, m$ can be regarded as a sequence of left-going waves generated by $\boldsymbol{f}_m$. The method for approximating them is similar to what was done for $\boldsymbol{f}_1$ (see Figure 2(d)). More specifically, for $\tilde{\boldsymbol{u}}_{m,m}$, we define

$$\begin{aligned}
D_m^M &:= (1 - 2\eta - (b-1)h, 1),\\
X_m^M &:= \{ih : (m-1)b + 1 \le i \le 2\gamma + mb - 2\},\\
\sigma_m^M(x) &:= \begin{cases} \dfrac{C}{\eta}\left(\dfrac{x - (1-\eta-(b-1)h)}{\eta}\right)^2, & x \in [1-2\eta-(b-1)h, 1-\eta-(b-1)h),\\ 0, & x \in [1-\eta-(b-1)h, 1-\eta],\\ \dfrac{C}{\eta}\left(\dfrac{x-(1-\eta)}{\eta}\right)^2, & x \in (1-\eta, 1], \end{cases}\\
s_m^M(x) &:= \left(1 + \mathrm{i}\frac{\sigma_m^M(x)}{\omega}\right)^{-1}.
\end{aligned}$$

We consider the continuous problem

$$\begin{cases} \left(\left(s_m^M(x)\dfrac{\mathrm{d}}{\mathrm{d}x}\right)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = g(x) & \forall x \in D_m^M,\\ v(x) = 0 & \forall x \in \partial D_m^M \end{cases}$$

and define $H_m^M \boldsymbol{v} = \boldsymbol{g}$ as its discretization on X_m^M. The operator $\tilde{G}_m^M : \boldsymbol{y} \to \boldsymbol{z}$ can be defined as follows:

1. Introduce a vector $\boldsymbol{g}$ defined on X_m^M by setting $\boldsymbol{y}$ to X_m and zero everywhere else.
2. Solve $H_m^M \boldsymbol{v} = \boldsymbol{g}$ on X_m^M.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_m.

Then

$$\tilde{\boldsymbol{u}}_{m,m} := \tilde{G}_m^M \boldsymbol{f}_m.$$

For each $\tilde{\boldsymbol{u}}_{p,m}$, $p = 1, \ldots, m-1$, we introduce the auxiliary domain D_p^L, the right boundary $\partial^R D_p^L$, the extended grid X_p^L, and the corresponding PML functions $\sigma_p^L(x), s_p^L(x)$ as follows:

$$\begin{aligned}
D_p^L &:= ((p-1)bh, \eta + pbh),\\
\partial^R D_p^L &:= \{\eta + pbh\},\\
X_p^L &:= \{x_i : (p-1)b + 1 \le i \le \gamma + pb - 1\},\\
\sigma_p^L(x) &:= \begin{cases} \dfrac{C}{\eta}\left(\dfrac{x-(\eta+(p-1)bh)}{\eta}\right)^2, & x \in [(p-1)bh, \eta + (p-1)bh),\\ 0, & x \in [\eta + (p-1)bh, \eta + pbh], \end{cases}\\
s_p^L(x) &:= \left(1 + \mathrm{i}\frac{\sigma_p^L(x)}{\omega}\right)^{-1},
\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

and we consider the continuous problem

$$\begin{cases} \left(\left(s_p^L(x) \dfrac{\mathrm{d}}{\mathrm{d}x} \right)^2 + \dfrac{\omega^2}{c^2(x)} \right) v(x) = 0 & \forall x \in D_p^L, \\ v(x) = w & \forall x \in \partial^R D_p^L, \\ v(x) = 0 & \forall x \in \partial D_p^L \setminus \partial^R D_p^L, \end{cases}$$

where y is the right boundary value of $v(x)$. Let $H_p^L \boldsymbol{v} = \boldsymbol{g}$ be its discretization on X_p^L with $\boldsymbol{g} := (-1/h^2)[0, \ldots, 0, w]^T$. We introduce the operator $\tilde{G}_p^L : w \mapsto \boldsymbol{z}$ by the following:

1. Set $\boldsymbol{g} = (-1/h^2)[0, \ldots, 0, w]^T$.
2. Solve $H_p^L \boldsymbol{v} = \boldsymbol{g}$ on X_p^L.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_p.

Then $\tilde{\boldsymbol{u}}_{p,m}$ can be defined recursively for $p = m-1, \ldots, 1$ by

$$\tilde{\boldsymbol{u}}_{p,m} := \tilde{G}_p^L \tilde{\boldsymbol{u}}_{p+1,m}^L.$$

2.1.3. Wave generated by $\boldsymbol{f}_q$ for $q = 2, \ldots, m-1$. For each q, the components $\boldsymbol{u}_{p,q}$ for $p = 1, \ldots, m$ can be regarded as a sequence of left- and right-going waves generated by $\boldsymbol{f}_q$ (see Figure 2(e)). For $\tilde{\boldsymbol{u}}_{q,q}$, we introduce

$$\begin{aligned} D_q^M &:= ((q-1)bh, 2\eta + (qb-1)h), \\ X_q^M &:= \{x_i : (q-1)b + 1 \le i \le 2\gamma + qb - 2\}, \\ \sigma_q^M(x) &:= \begin{cases} \dfrac{C}{\eta} \left(\dfrac{x - (\eta + (q-1)bh)}{\eta} \right)^2, & x \in [(q-1)bh, \eta + (q-1)bh), \\ 0, & x \in [\eta + (q-1)bh, \eta + (qb-1)h], \\ \dfrac{C}{\eta} \left(\dfrac{x - (\eta + (qb-1)h)}{\eta} \right)^2, & x \in (\eta + (qb-1)h, 2\eta + (qb-1)h], \end{cases} \\ s_q^M(x) &:= \left(1 + \mathrm{i} \frac{\sigma_q^M(x)}{\omega} \right)^{-1} \end{aligned}$$

and define $H_q^M \boldsymbol{v} = \boldsymbol{g}$ as the discrete problem of the continuous problem

$$\begin{cases} \left(\left(s_q^M(x) \dfrac{\mathrm{d}}{\mathrm{d}x} \right)^2 + \dfrac{\omega^2}{c^2(x)} \right) v(x) = g(x) & \forall x \in D_q^M, \\ v(x) = 0 & \forall x \in \partial D_q^M. \end{cases}$$

We introduce the operator $\tilde{G}_q^M : \boldsymbol{y} \to \boldsymbol{z}$ as follows:

1. Introduce a vector $\boldsymbol{g}$ defined on X_q^M by setting $\boldsymbol{y}$ to X_q and zero everywhere else.
2. Solve $H_q^M \boldsymbol{v} = \boldsymbol{g}$ on X_q^M.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_q.

Then

$$\tilde{\boldsymbol{u}}_{q,q} := \tilde{G}_q^M \boldsymbol{f}_q.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

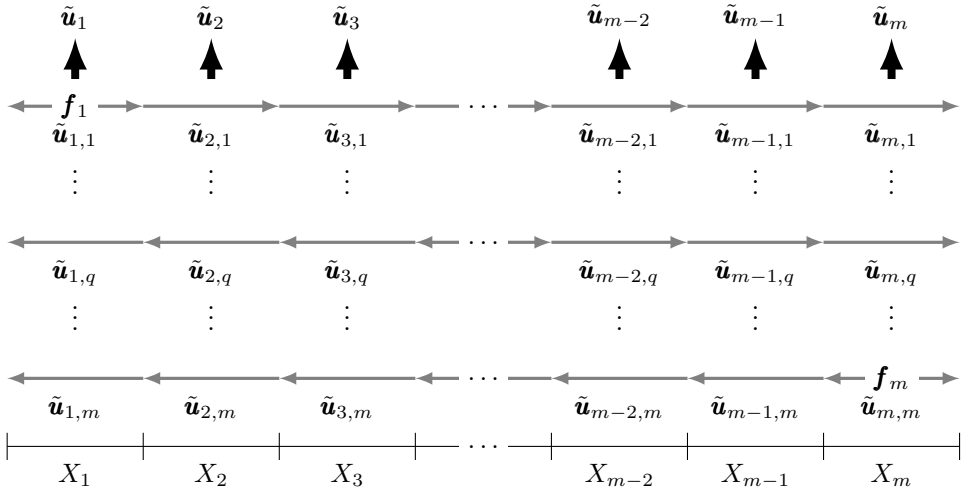
Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Following the above discussion, the remaining components $\tilde{\boldsymbol{u}}_{p,q}$ are defined recursively as

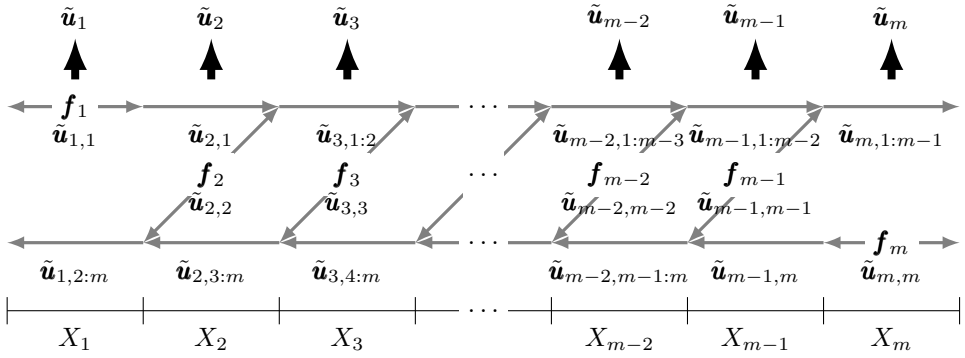
$$\tilde{\boldsymbol{u}}_{p,q} := \tilde{G}_p^R \tilde{\boldsymbol{u}}_{p-1,q}^R \quad \text{for } p = q+1, \dots, m,$$
$$\tilde{\boldsymbol{u}}_{p,q} := \tilde{G}_p^L \tilde{\boldsymbol{u}}_{p+1,q}^L \quad \text{for } p = q-1, \dots, 1.$$

2.2. Accumulating the boundary values. After all the above are done, an approximation of $\boldsymbol{u}_p$ is given by (see Figure 3(a))

$$\tilde{\boldsymbol{u}}_p := \sum_{q=1}^{m} \tilde{\boldsymbol{u}}_{p,q}, \quad p = 1, \dots, m.$$



(a) $\tilde{\boldsymbol{u}}_p$ is a superposition of $\tilde{\boldsymbol{u}}_{p,q}, q = 1, \dots, m$.



(b) $\tilde{\boldsymbol{u}}_p$ is a superposition of $\tilde{\boldsymbol{u}}_{p,1:p-1}$, $\tilde{\boldsymbol{u}}_{p,p}$, and $\tilde{\boldsymbol{u}}_{p,p+1:m}$.

FIG. 3. *An illustration of how the boundary values are accumulated after each step. The thin arrows indicate the transmission directions of the waves. The bold, up-pointing arrows symbolize that summing up the corresponding waves on X_p gives the superposition wave $\tilde{\boldsymbol{u}}_p$.*

In the algorithm described above, the computation of each component $\tilde{\boldsymbol{u}}_{p,q}$ requires a separate solution of a problem of form $H_p^R \boldsymbol{v} = \boldsymbol{g}$ or $H_p^L \boldsymbol{v} = \boldsymbol{g}$. Since there are

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$O(m^2)$ such components, the algorithm is computationally expensive. A key observation is that the computation associated with each p can be combined in one single shot by accumulating the boundary values of the waves. More precisely, we define

$$\tilde{\boldsymbol{u}}_{p,q_1:q_2} := \sum_{t=q_1}^{q_2} \tilde{\boldsymbol{u}}_{p,t},$$

which is the total contribution of the waves generated by $\boldsymbol{f}_{q_1}, \ldots, \boldsymbol{f}_{q_2}$ restricted to the grid X_p. The quantity $\tilde{\boldsymbol{u}}_{p,1:p-1}$, which is the total right-going wave generated by $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{p-1}$ upon X_p, can be computed sequentially for $p = 2, \ldots, m$ without computing each component and then adding them together, as we described above, as long as we accumulate the boundary values after each intermediate step. Specifically, we first compute $\tilde{\boldsymbol{u}}_{q,q} = \tilde{G}_q^M \boldsymbol{f}_q$ for $q = 1, \ldots, m$. This step is similar to what we did above. Then to compute $\tilde{\boldsymbol{u}}_{p,1:p-1}$ we carry out the following steps:

$$\tilde{\boldsymbol{u}}_{p,1:p-1} = \tilde{G}_p^R \tilde{\boldsymbol{u}}^R_{p-1,1:p-1}, \quad \tilde{\boldsymbol{u}}^R_{p,1:p} = \tilde{\boldsymbol{u}}^R_{p,1:p-1} + \tilde{\boldsymbol{u}}^R_{p,p} \quad \text{for } p = 2, \ldots, m.$$

This means that before computing the total right-going wave $\tilde{\boldsymbol{u}}_{p+1,1:p}$ on subdomain X_{p+1}, the boundary values of the previous right-going waves, $\tilde{\boldsymbol{u}}^R_{p,1:p-1}$ and $\tilde{\boldsymbol{u}}^R_{p,p}$, are added together, so that the current right-going wave $\tilde{\boldsymbol{u}}_{p+1,1:p}$ can be computed in one shot, eliminating the trouble of solving the subproblems for many times and adding the results together (see Figure 3(b)).

For the left-going waves $\tilde{\boldsymbol{u}}_{p,p+1:m}$, a similar process gives rise to the recursive formula

$$\tilde{\boldsymbol{u}}_{p,p+1:m} = \tilde{G}_p^L \tilde{\boldsymbol{u}}^L_{p+1,p+1:m}, \quad \tilde{\boldsymbol{u}}^L_{p,p:m} = \tilde{\boldsymbol{u}}^L_{p,p} + \tilde{\boldsymbol{u}}^L_{p,p+1:m} \quad \text{for } p = m-1, \ldots, 1.$$

Finally, each $\tilde{\boldsymbol{u}}_p$ can be computed by summing $\tilde{\boldsymbol{u}}_{p,1:p-1}$, $\tilde{\boldsymbol{u}}_{p,p}$, and $\tilde{\boldsymbol{u}}_{p,p+1:m}$ together (for the leftmost and the rightmost ones, $\tilde{\boldsymbol{u}}_1$ and $\tilde{\boldsymbol{u}}_m$, only two terms need to be summed), i.e.,

$$\begin{aligned}
\tilde{\boldsymbol{u}}_1 &= \tilde{\boldsymbol{u}}_{1,1} + \tilde{\boldsymbol{u}}_{1,2:m}, \\
\tilde{\boldsymbol{u}}_p &= \tilde{\boldsymbol{u}}_{p,1:p-1} + \tilde{\boldsymbol{u}}_{p,p} + \tilde{\boldsymbol{u}}_{p,p+1:m}, \quad p = 2, \ldots, m-1, \\
\tilde{\boldsymbol{u}}_m &= \tilde{\boldsymbol{u}}_{m,1:m-1} + \tilde{\boldsymbol{u}}_{m,m}.
\end{aligned}$$

We see that, by accumulating the boundary values after each intermediate step, we only need to solve $O(m)$ subproblems instead of $O(m^2)$.

In this algorithm, the approximation $\tilde{\boldsymbol{u}}_p$ on each small subdomain is divided into three parts. From a matrix point of view, this is analogous to splitting the block matrix G into its lower triangular part, diagonal part, and upper triangular part, and then approximating each part as an operator to get the intermediate waves and then summing the intermediate results together. This is why we call it the additive sweeping method.

Equation (3) shows an analogy of this procedure, where the matrix G is split into

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$3m-2$ blocks, each of which corresponds to a subproblem solving process:

$$\tilde{\boldsymbol{u}}_{q,q} \approx \boldsymbol{u}_{q,q} = G_{q,q}\boldsymbol{f}_q, \quad q = 1, \dots, m,$$

$$\tilde{\boldsymbol{u}}_{p,1:p-1} \approx \boldsymbol{u}_{p,1:p-1} = \sum_{q=1}^{p-1} G_{p,q}\boldsymbol{f}_q, \quad p = 2, \dots, m,$$

$$\tilde{\boldsymbol{u}}_{p,p+1:m} \approx \boldsymbol{u}_{p,p+1:m} = \sum_{q=p+1}^{m} G_{p,q}\boldsymbol{f}_q, \quad p = 1, \dots, m-1,$$

(3)
$$\begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \dots \\ \boldsymbol{u}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_{1,1} + \boldsymbol{u}_{1,2:m} \\ \boldsymbol{u}_{2,1} + \boldsymbol{u}_{2,2} + \boldsymbol{u}_{2,3:m} \\ \dots \\ \boldsymbol{u}_{m,1:m-1} + \boldsymbol{u}_{m,m} \end{bmatrix} = \left[\begin{array}{c|c|ccc|c} G_{1,1} & G_{1,2} & \dots & & G_{1,m} \\ \hline G_{2,1} & G_{2,2} & G_{2,3} & \dots & G_{2,m} \\ \hline & & \dots & & \\ \hline G_{m,1} & \dots & & G_{m,m-1} & G_{m,m} \end{array}\right] \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \dots \\ \boldsymbol{f}_m \end{bmatrix}.$$

When combined with standard iterative solvers, the approximation algorithm serves as a preconditioner for (2), and it can be easily generalized to higher dimensions. In the following sections, we will discuss the details of the algorithm in two and three dimensions. To be structurally consistent, we will keep the notations for two and three dimensions the same with the 1D case without causing ambiguity. Some of the key notations and concepts are listed below as a reminder to the reader:

- $\{X_p\}_{p=1}^m$ The sliced partition of the discrete grid.
- $\{D_q^M\}_{q=1}^m$ The auxiliary domains with two-sided PML padding.
- $\{D_p^R\}_{p=2}^m$ The auxiliary domains with right-side PML padding.
- $\{D_p^L\}_{p=1}^{m-1}$ The auxiliary domains with left-side PML padding.
- $\{X_q^M\}_{q=1}^m$ X_q with two-sided PML padding, the discretization of D_q^M.
- $\{X_p^R\}_{p=2}^m$ X_p with right-side PML padding, the discretization of D_p^R.
- $\{X_p^L\}_{p=1}^{m-1}$ X_p with left-side PML padding, the discretization of D_p^L.
- $\{\tilde{G}_q^M\}_{q=1}^m$ The auxiliary Green's operators, each of which maps the force on X_q to the approximation of the wave field restricted to X_q.
- $\{\tilde{G}_p^R\}_{p=2}^m$ The auxiliary Green's operators, each of which maps the left boundary value to the approximated wave field restricted to X_p, which simulates the right-transmission of the waves.
- $\{\tilde{G}_p^L\}_{p=1}^{m-1}$ The auxiliary Green's operators, each of which maps the right boundary value to the approximated wave field restricted to X_p, which simulates the left-transmission of the waves.

3. Preconditioner in two dimensions.

3.1. Algorithm. The domain of interest is $D = (0,1)^2$. We put the PML on the two opposite sides of the boundary, $x_2 = 0$ and $x_2 = 1$, to illustrate the idea. The resulting equation is

$$\begin{cases} \left(\partial_1^2 + (s(x_2)\partial_2)^2 + \dfrac{\omega^2}{c^2(x)}\right) u(x) = f(x) & \forall x = (x_1, x_2) \in D, \\ u(x) = 0 & \forall x \in \partial D. \end{cases}$$

We discretize D with step size $h = 1/(n+1)$ in each direction, which results in the Cartesian grid

$$X := \{(i_1 h, i_2 h) : 1 \le i_1, i_2 \le n\}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

and the discrete equation

$$\begin{aligned}&\frac{s_{i_2}}{h}\left(\frac{s_{i_2+1/2}}{h}(u_{i_1,i_2+1}-u_{i_1,i_2})-\frac{s_{i_2-1/2}}{h}(u_{i_1,i_2}-u_{i_1,i_2-1})\right)\\&+\frac{u_{i_1+1,i_2}-2u_{i_1,i_2}+u_{i_1-1,i_2}}{h^2}+\frac{\omega^2}{c^2_{i_1,i_2}}u_{i_1,i_2}=f_{i_1,i_2}\quad\forall 1\le i_1,i_2\le n,\end{aligned}\tag{4}$$

where the subscript (i_1, i_2) means that the corresponding function is evaluated at (i_1h, i_2h), and since $s(x_2)$ is a function of x_2 only, we omit the i_1 subscript. $\boldsymbol{u}$ and $\boldsymbol{f}$ are defined to be the column-major ordering of the discrete array u and f on the grid X:

$$\boldsymbol{u} := [u_{1,1},\dots,u_{n,1},\dots,u_{n,n}]^T,\quad \boldsymbol{f} := [f_{1,1},\dots,f_{n,1},\dots,f_{n,n}]^T.$$

Now (4) can be written as $A\boldsymbol{u} = \boldsymbol{f}$.

We divide the grid into m parts along the x_2 direction:

$$\begin{aligned}X_1 &:= \{(i_1h,i_2h) : 1\le i_1\le n, 1\le i_2\le \gamma+b-1\},\\ X_p &:= \{(i_1h,i_2h) : 1\le i_1\le n, \gamma+(p-1)b\le i_2\le \gamma+pb-1\},\quad p=2,\dots,m-1,\\ X_m &:= \{(i_1h,i_2h) : 1\le i_1\le n, \gamma+(m-1)b\le i_2\le 2\gamma+mb-2\},\end{aligned}$$

and we define $\boldsymbol{u}_p$ and $\boldsymbol{f}_p$ as the column-major ordering restriction of u and f on X_p:

$$\begin{aligned}\boldsymbol{u}_1 &:= [u_{1,1},\dots,u_{n,1},\dots,u_{n,\gamma+b-1}]^T,\\ \boldsymbol{u}_p &:= [u_{1,\gamma+(p-1)b},\dots,u_{n,\gamma+(p-1)b},\dots,u_{n,\gamma+pb-1}]^T,\quad p=2,\dots,m-1,\\ \boldsymbol{u}_m &:= [u_{1,\gamma+(m-1)b},\dots,u_{n,\gamma+(m-1)b},\dots,u_{n,2\gamma+mb-2}]^T,\\ \boldsymbol{f}_1 &:= [f_{1,1},\dots,f_{n,1},\dots,f_{n,\gamma+b-1}]^T,\\ \boldsymbol{f}_p &:= [f_{1,\gamma+(p-1)b},\dots,f_{n,\gamma+(p-1)b},\dots,f_{n,\gamma+pb-1}]^T,\quad p=2,\dots,m-1,\\ \boldsymbol{f}_m &:= [f_{1,\gamma+(m-1)b},\dots,f_{n,\gamma+(m-1)b},\dots,f_{n,2\gamma+mb-2}]^T;\end{aligned}$$

then $\boldsymbol{u} = G\boldsymbol{f}$ for $G = A^{-1}$ can be written as

$$\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\\ \vdots\\ \boldsymbol{u}_m\end{bmatrix}=\begin{bmatrix}G_{1,1} & G_{1,2} & \dots & G_{1,m}\\ G_{2,1} & G_{2,2} & \dots & G_{2,m}\\ \vdots & \vdots & & \vdots\\ G_{m,1} & G_{m,2} & \dots & G_{m,m}\end{bmatrix}\begin{bmatrix}\boldsymbol{f}_1\\ \boldsymbol{f}_2\\ \vdots\\ \boldsymbol{f}_m\end{bmatrix}.$$

3.1.1. Auxiliary domains. Following to the 1D case, the extended subdomains and the corresponding left and right boundaries are defined by

$$\begin{aligned}D_q^M &= (0,1)\times((q-1)bh, 2\eta+(qb-1)h),\quad q=1,\dots,m,\\ D_p^R &= (0,1)\times(\eta+((p-1)b-1)h, 2\eta+(pb-1)h),\quad p=2,\dots,m,\\ D_p^L &= (0,1)\times((p-1)bh, \eta+pbh),\quad p=1,\dots,m-1,\\ \partial^L D_p^R &= (0,1)\times\{\eta+((p-1)b-1)h\},\quad p=2,\dots,m,\\ \partial^R D_p^L &= (0,1)\times\{\eta+pbh\},\quad p=1,\dots,m-1.\end{aligned}$$

The extended grids for these domains are

$$\begin{aligned}X_q^M &:= \{(i_1h,i_2h) : 1\le i_1\le n, (q-1)b+1\le i_2\le 2\gamma+qb-1\},\quad q=1,\dots,m,\\ X_p^R &:= \{(i_1h,i_2h) : 1\le i_1\le n, \gamma+(p-1)b\le i_2\le 2\gamma+pb-2\},\quad p=2,\dots,m,\\ X_p^L &:= \{(i_1h,i_2h) : 1\le i_1\le n, (p-1)b+1\le i_2\le \gamma+pb-1\},\quad p=1,\dots,m-1.\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3.1.2. Auxiliary problems. For $q = 1, \ldots, m$, we define $H_q^M \boldsymbol{v} = \boldsymbol{g}$ to be the discretization on X_q^M of the problem

$$\begin{cases} \left(\partial_1^2 + (s_q^M(x_2)\partial_2)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = g(x) & \forall x \in D_q^M, \\ v(x) = 0 & \forall x \in \partial D_q^M. \end{cases}$$

For $p = 2, \ldots, m$, $H_p^R \boldsymbol{v} = \boldsymbol{g}$ is the discretization on X_p^R of the problem

$$\begin{cases} \left(\partial_1^2 + (s_p^R(x_2)\partial_2)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = 0 & \forall x \in D_p^R, \\ v(x) = w(x_1) & \forall x \in \partial^L D_p^R, \\ v(x) = 0 & \forall x \in \partial D_p^R \setminus \partial^L D_p^R, \end{cases}$$

where $\boldsymbol{g} := (-1/h^2)[\boldsymbol{w}^T, 0, \ldots, 0]^T$ and $\boldsymbol{w} := [w_1, \ldots, w_n]^T$ is the discrete value of $w(x_1)$. Finally, for $p = 1, \ldots, m-1$, $H_p^L \boldsymbol{v} = \boldsymbol{g}$ is the discretization on X_p^L of the problem

$$\begin{cases} \left(\partial_1^2 + (s_p^L(x_2)\partial_2)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = 0 & \forall x \in D_p^L, \\ v(x) = w(x_1) & \forall x \in \partial^R D_p^L, \\ v(x) = 0 & \forall x \in \partial D_p^L \setminus \partial^R D_p^L, \end{cases}$$

where $\boldsymbol{g} := (-1/h^2)[0, \ldots, 0, \boldsymbol{w}^T]^T$ and $\boldsymbol{w} := [w_1, \ldots, w_n]^T$.

3.1.3. Auxiliary Green's operators. For $q = 1, \ldots, m$, we define $\tilde{G}_q^M : \boldsymbol{y} \mapsto \boldsymbol{z}$ to be the operator defined by the following operations:

1. Introduce a vector $\boldsymbol{g}$ defined on X_q^M by setting $\boldsymbol{y}$ to X_q and zero otherwise.
2. Solve $H_q^M \boldsymbol{v} = \boldsymbol{g}$ on X_q^M.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_q.

For $p = 2, \ldots, m$, the operator $\tilde{G}_p^R : \boldsymbol{w} \mapsto \boldsymbol{z}$ is given by the following:

1. Set $\boldsymbol{g} = (-1/h^2)[\boldsymbol{w}^T, 0, \ldots, 0]^T$.
2. Solve $H_p^R \boldsymbol{v} = \boldsymbol{g}$ on X_p^R.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_p.

Finally, for $p = 1, \ldots, m-1$, $\tilde{G}_p^L : \boldsymbol{w} \mapsto \boldsymbol{z}$ is defined as follows:

1. Set $\boldsymbol{g} = (-1/h^2)[0, \ldots, 0, \boldsymbol{w}^T]^T$.
2. Solve $H_p^L \boldsymbol{v} = \boldsymbol{g}$ on X_p^L.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_p.

3.1.4. Putting together. Similar to the previous section, we introduce the left boundary value $\boldsymbol{g}^L$ and the right boundary value $\boldsymbol{g}^R$ for a column-major ordering array $\boldsymbol{g} = [g_{1,1}, \ldots, g_{s_1,1}, \ldots, g_{s_1,s_2}]^T$ induced from some grid with size $s_1 \times s_2$ by

$$\boldsymbol{g}^L := [g_{1,1}, \ldots, g_{s_1,1}]^T, \quad \boldsymbol{g}^R := [g_{1,s_2}, \ldots, g_{s_1,s_2}]^T.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Then the approximations for $\boldsymbol{u}_p$, $p = 1, \ldots, m$, can be defined step by step as

$$\begin{aligned}
\tilde{\boldsymbol{u}}_{q,q} &:= \tilde{G}_q^M \boldsymbol{f}_q, \quad q = 1, \ldots, m, \\
\tilde{\boldsymbol{u}}_{p,1:p-1} &:= \tilde{G}_p^R \tilde{\boldsymbol{u}}_{p-1,1:p-1}^R, \quad \tilde{\boldsymbol{u}}_{p,1:p}^R := \tilde{\boldsymbol{u}}_{p,1:p-1}^R + \tilde{\boldsymbol{u}}_{p,p}^R \quad \text{for } p = 2, \ldots, m, \\
\tilde{\boldsymbol{u}}_{p,p+1:m} &:= \tilde{G}_p^L \tilde{\boldsymbol{u}}_{p+1,p+1:m}^L, \quad \tilde{\boldsymbol{u}}_{p,p:m}^L := \tilde{\boldsymbol{u}}_{p,p}^L + \tilde{\boldsymbol{u}}_{p,p+1:m}^L \quad \text{for } p = m-1, \ldots, 1, \\
\tilde{\boldsymbol{u}}_1 &:= \tilde{\boldsymbol{u}}_{1,1} + \tilde{\boldsymbol{u}}_{1,2:m}, \\
\tilde{\boldsymbol{u}}_p &:= \tilde{\boldsymbol{u}}_{p,1:p-1} + \tilde{\boldsymbol{u}}_{p,p} + \tilde{\boldsymbol{u}}_{p,p+1:m}, \quad p = 2, \ldots, m-1, \\
\tilde{\boldsymbol{u}}_m &:= \tilde{\boldsymbol{u}}_{m,1:m-1} + \tilde{\boldsymbol{u}}_{m,m}.
\end{aligned}$$

To solve the subproblems $H_q^M \boldsymbol{v} = \boldsymbol{g}$, $H_p^R \boldsymbol{v} = \boldsymbol{g}$, and $H_p^L \boldsymbol{v} = \boldsymbol{g}$, we notice that they are indeed quasi-1D problems, since γ and b are some small constants. Therefore, for each one of them, we can reorder the system by grouping the elements along dimension 2 first and then dimension 1, which results in a banded linear system that can be solved by the LU factorization efficiently. These factorization processes induce the factorizations for the operators $\tilde{G}_q^M$, $\tilde{G}_p^R$, and $\tilde{G}_p^L$ symbolically, which leads to our setup algorithm of the preconditioner in two dimensions as described in Algorithm 1 and the application algorithm as described in Algorithm 2.

Algorithm 1 Construction of the 2D additive sweeping preconditioner of (4). Complexity $= O(n^2(b+\gamma)^3/b) = O(N(b+\gamma)^3/b)$.

for $q = 1, \ldots, m$ **do**
  Construct the LU factorization of H_q^M, which defines $\tilde{G}_q^M$.
end for
for $p = 2, \ldots, m$ **do**
  Construct the LU factorization of H_p^R, which defines $\tilde{G}_p^R$.
end for
for $p = 1, \ldots, m-1$ **do**
  Construct the LU factorization of H_p^L, which defines $\tilde{G}_p^L$.
end for

To analyze the complexity, we note that, in the setup process, there are $O(n/b)$ subproblems, each of which is a quasi-1D problem with $O(\gamma+b)$ layers along the second dimension. Therefore, the setup cost of each subproblem by the LU factorization is $O(n(\gamma+b)^3)$ and the application cost is $O(n(\gamma+b)^2)$. So the total setup cost is $O(n^2(\gamma+b)^3/b)$. Besides, one needs to solve each subproblem once during the application process, so the total application cost is $O(n^2(\gamma+b)^2/b)$.

There are some differences when implementing the method practically:

1. In the above setting, PMLs are put only on two opposite sides of the unit square for illustration purposes. In reality, PMLs can be put on other sides of the domain if needed. As long as there are two opposite sides with a PML boundary condition, the method can be implemented.
2. The thickness of the auxiliary PMLs introduced in the interior part of the domain need not be the same as the thickness of the PML at the boundary. In fact, the thickness of the auxiliary PML is typically thinner in order to improve efficiency.
3. The widths of the subdomains are completely arbitrary, and they need not be the same. Practically, the widths can be chosen to be larger for subdomains where the velocity field varies heavily.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

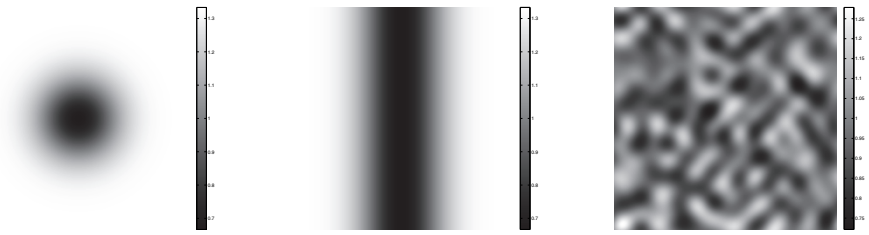
Algorithm 2 Computation of $\tilde{\boldsymbol{u}} \approx G\boldsymbol{f}$ using the preconditioner from Algorithm 1. Complexity $= O(n^2(b+\gamma)^2/b) = O(N(b+\gamma)^2/b)$.

for $q = 1, \ldots, m$ **do**
  $\tilde{\boldsymbol{u}}_{q,q} = \tilde{G}_q^M \boldsymbol{f}_q$
end for
for $p = 2, \ldots, m$ **do**
  $\tilde{\boldsymbol{u}}_{p,1:p-1} = \tilde{G}_p^R \tilde{\boldsymbol{u}}^R_{p-1,1:p-1}$
  $\tilde{\boldsymbol{u}}^R_{p,1:p} = \tilde{\boldsymbol{u}}^R_{p,1:p-1} + \tilde{\boldsymbol{u}}^R_{p,p}$
end for
for $p = m-1, \ldots, 1$ **do**
  $\tilde{\boldsymbol{u}}_{p,p+1:m} = \tilde{G}_p^L \tilde{\boldsymbol{u}}^L_{p+1,p+1:m}$
  $\tilde{\boldsymbol{u}}^L_{p,p:m} = \tilde{\boldsymbol{u}}^L_{p,p} + \tilde{\boldsymbol{u}}^L_{p,p+1:m}$
end for
$\tilde{\boldsymbol{u}}_1 = \tilde{\boldsymbol{u}}_{1,1} + \tilde{\boldsymbol{u}}_{1,2:m}$
for $p = 2, \ldots, m-1$ **do**
  $\tilde{\boldsymbol{u}}_p = \tilde{\boldsymbol{u}}_{p,1:p-1} + \tilde{\boldsymbol{u}}_{p,p} + \tilde{\boldsymbol{u}}_{p,p+1:m}$
end for
$\tilde{\boldsymbol{u}}_m = \tilde{\boldsymbol{u}}_{m,1:m-1} + \tilde{\boldsymbol{u}}_{m,m}$

4. The symmetric version of the equation can be adopted to save memory and computational cost.

3.2. Numerical results. Here, we present some numerical results in two dimensions to illustrate the efficiency of the algorithm. The proposed method is implemented in MATLAB, and the tests are performed on a 2.0 GHz computer with 256 GB memory. GMRES is used as the iterative solver with relative residual equal to 10^{-3} and restart value equal to 40. PMLs are put on all sides of the unit square. The velocity fields tested are given in Figure 4:

(a) A converging lens with a Gaussian profile at the center of the domain.
(b) A vertical waveguide with a Gaussian cross-section.
(c) A random velocity field.



(a) (b) (c)

FIG. 4. *The three velocity fields tested in two dimensions.*

For each velocity field, two external forces are tested:

(a) A Gaussian point source centered at $(1/2, 1/8)$.
(b) A Gaussian wave packet with wavelength comparable to the typical wavelength of the domain. The packet centers at $(1/8, 1/8)$ and points to the direction

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

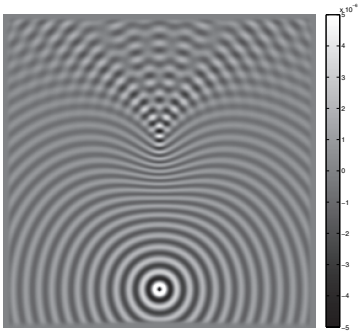
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$(1/\sqrt{2}, 1/\sqrt{2})$.

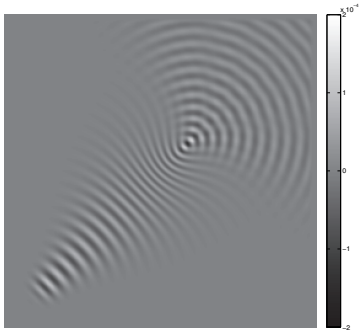
In these tests, each typical wavelength is discretized with eight points. The width of the PML at the boundary and that of the PMLs introduced in the interior parts of the domain are both $9h$, i.e., $\gamma = 9$. The number of layers in each interior subdomain is $b = 8$, the number of layers in the leftmost subdomain is $b + \gamma - 1 = 16$, and that in the rightmost is $b + \gamma - 2 = 15$.

We vary the typical wave number $\omega/(2\pi)$ and test the behavior of the algorithm. The test results are presented in Tables 1 to 3. T_{setup} is the setup time of the algorithm in seconds. T_{solve} is the total solve time in seconds, and N_{iter} is the iteration number. From these tests we see that the setup time scales like $O(N)$ as well as the solve time per iteration, which is consistent with the algorithm complexity analysis. The iteration number remains constant or grows at most logarithmically, which shows the efficiency of the preconditioner.

TABLE 1
Results for velocity field (a) *in two dimensions. Solutions with* $\omega/(2\pi) = 32$ *are presented.*



Force (a)



Force (b)

Velocity field (a)			Force (a)		Force (b)	
$\omega/(2\pi)$	N	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{solve}
16	127^2	8.1669e−01	4	5.3199e−01	4	2.5647e−01
32	255^2	3.4570e+00	4	7.3428e−01	4	7.2807e−01
64	511^2	1.5150e+01	5	3.6698e+00	4	3.7239e+00
128	1023^2	6.2713e+01	5	1.6812e+01	4	1.6430e+01
256	2047^2	2.6504e+02	6	7.8148e+01	4	5.6936e+01

4. Preconditioner in three dimensions.

4.1. Algorithm. In this section we briefly state the preconditioner in the 3D case. The domain of interest is $D = (0,1)^3$. PMLs are put on two opposite faces of the unit cube, $x_3 = 0$ and $x_3 = 1$, which results in the equation

$$\begin{cases} \left(\partial_1^2 + \partial_2^2 + (s(x_3)\partial_3)^2 + \dfrac{\omega^2}{c^2(x)}\right) u(x) = f(x) & \forall x = (x_1, x_2, x_3) \in D, \\ u(x) = 0 & \forall x \in \partial D. \end{cases}$$

Discretizing D with step size $h = 1/(n+1)$ gives the grid

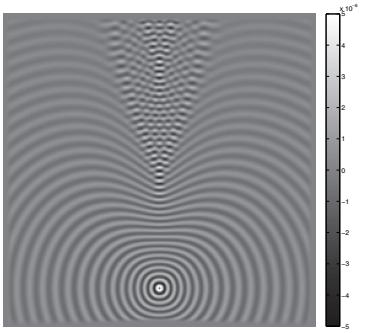
$$X := \{(i_1 h, i_2 h, i_3 h) : 1 \le i_1, i_2, i_3 \le n\}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

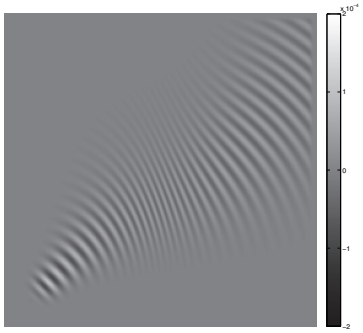
Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 2
Results for velocity field (b) *in two dimensions. Solutions with* $\omega/(2\pi) = 32$ *are presented.*



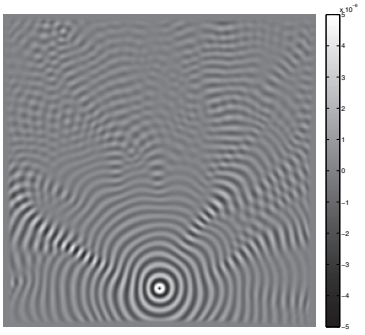
Force (a)



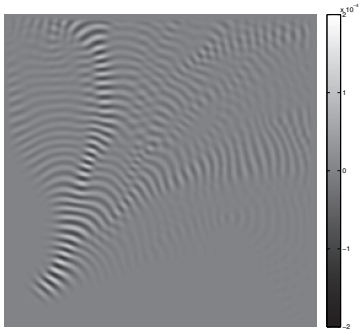
Force (b)

Velocity field (b)			Force (a)		Force (b)	
$\omega/(2\pi)$	N	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{solve}
16	127^2	7.0834e−01	6	2.9189e−01	4	1.9408e−01
32	255^2	3.2047e+00	8	1.6147e+00	4	7.9303e−01
64	511^2	1.4079e+01	8	6.3057e+00	4	3.9008e+00
128	1023^2	6.0951e+01	8	2.9097e+01	4	1.5287e+01
256	2047^2	2.6025e+02	8	1.1105e+02	5	7.2544e+01

TABLE 3
Results for velocity field (c) *in two dimensions. Solutions with* $\omega/(2\pi) = 32$ *are presented.*



Force (a)



Force (b)

Velocity field (c)			Force (a)		Force (b)	
$\omega/(2\pi)$	N	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{solve}
16	127^2	7.0495e−01	5	2.4058e−01	6	2.8347e−01
32	255^2	3.1760e+00	5	1.0506e+00	5	9.9551e−01
64	511^2	1.4041e+01	6	4.7083e+00	7	6.7852e+00
128	1023^2	6.1217e+01	6	1.8652e+01	6	1.9792e+01
256	2047^2	2.5762e+02	8	1.1214e+02	6	8.6936e+01

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

and the discrete equation

$$\begin{aligned}(5)\qquad &\frac{s_{i_3}}{h}\left(\frac{s_{i_3+1/2}}{h}(u_{i_1,i_2,i_3+1}-u_{i_1,i_2,i_3})-\frac{s_{i_3-1/2}}{h}(u_{i_1,i_2,i_3}-u_{i_1,i_2,i_3-1})\right)\\ &+\frac{u_{i_1+1,i_2,i_3}-2u_{i_1,i_2,i_3}+u_{i_1-1,i_2,i_3}}{h^2}+\frac{u_{i_1,i_2+1,i_3}-2u_{i_1,i_2,i_3}+u_{i_1,i_2-1,i_3}}{h^2}\\ &+\frac{\omega^2}{c^2_{i_1,i_2,i_3}}u_{i_1,i_2,i_3}=f_{i_1,i_2,i_3}\quad\forall 1\le i_1,i_2\le n.\end{aligned}$$

$\boldsymbol{u}$ and $\boldsymbol{f}$ are defined as the column-major ordering of u and f on the grid X:

$$\begin{aligned}\boldsymbol{u}&:=[u_{1,1,1},\dots,u_{n,1,1},\dots,u_{n,n,1},\dots,u_{n,n,n}]^T,\\ \boldsymbol{f}&:=[f_{1,1,1},\dots,f_{n,1,1},\dots,f_{n,n,1},\dots,f_{n,n,n}].\end{aligned}$$

X is divided into m parts along the x_3 direction:

$$\begin{aligned}X_1&:=\{(i_1h,i_2h,i_3h):1\le i_1\le n,1\le i_2\le n,1\le i_3\le \gamma+b-1\},\\ X_p&:=\{(i_1h,i_2h,i_3h):1\le i_1\le n,1\le i_2\le n,\gamma+(p-1)b\le i_3\le \gamma+pb-1\},\\ &\quad p=2,\dots,m-1,\\ X_m&:=\{(i_1h,i_2h,i_3h):1\le i_1\le n,1\le i_2\le n,\gamma+(m-1)b\le i_3\le 2\gamma+mb-2\}.\end{aligned}$$

$\boldsymbol{u}_p$ and $\boldsymbol{f}_p$ are the column-major ordering restrictions of u and f on X_p:

$$\begin{aligned}\boldsymbol{u}_1&:=[u_{1,1,1},\dots,u_{n,1,1},\dots,u_{n,n,1},\dots,u_{n,n,\gamma+b-1}]^T,\\ \boldsymbol{u}_p&:=[u_{1,1,\gamma+(p-1)b},\dots,u_{n,1,\gamma+(p-1)b},\dots,u_{n,n,\gamma+(p-1)b},\dots,u_{n,n,\gamma+pb-1}]^T,\\ &\quad p=2,\dots,m-1,\\ \boldsymbol{u}_m&:=[u_{1,1,\gamma+(m-1)b},\dots,u_{n,1,\gamma+(m-1)b},\dots,u_{n,n,\gamma+(m-1)b},\dots,u_{n,n,2\gamma+mb-2}]^T,\\ \boldsymbol{f}_1&:=[f_{1,1,1},\dots,f_{n,1,1},\dots,f_{n,n,1},\dots,f_{n,n,\gamma+b-1}]^T,\\ \boldsymbol{f}_p&:=[f_{1,1,\gamma+(p-1)b},\dots,f_{n,1,\gamma+(p-1)b},\dots,f_{n,n,\gamma+(p-1)b},\dots,f_{n,n,\gamma+pb-1}]^T,\\ &\quad p=2,\dots,m-1,\\ \boldsymbol{f}_m&:=[f_{1,1,\gamma+(m-1)b},\dots,f_{n,1,\gamma+(m-1)b},\dots,f_{n,n,\gamma+(m-1)b},\dots,f_{n,n,2\gamma+mb-2}]^T.\end{aligned}$$

4.1.1. Auxiliary domains. The extended subdomains, the extended grids, and the corresponding left and right boundaries are defined by

$$\begin{aligned}D_q^M&:=(0,1)\times(0,1)\times((q-1)bh,2\eta+(qb-1)h),\quad q=1,\dots,m,\\ D_p^R&:=(0,1)\times(0,1)\times(\eta+((p-1)b-1)h,2\eta+(pb-1)h),\quad p=2,\dots,m,\\ D_p^L&:=(0,1)\times(0,1)\times((p-1)bh,\eta+pbh),\quad p=1,\dots,m-1,\\ \partial^L D_p^R&:=(0,1)\times(0,1)\times\{\eta+((p-1)b-1)h\},\quad p=2,\dots,m,\\ \partial^R D_p^L&:=(0,1)\times(0,1)\times\{\eta+pbh\},\quad p=1,\dots,m-1,\\ X_q^M&:=\{(i_1h,i_2h,i_3h):1\le i_1\le n,1\le i_2\le n,(q-1)b+1\le i_3\le 2\gamma+qb-1\},\\ &\quad q=1,\dots,m,\\ X_p^R&:=\{(i_1h,i_2h,i_3h):1\le i_1\le n,1\le i_2\le n,\gamma+(p-1)b\le i_3\le 2\gamma+pb-2\},\\ &\quad p=2,\dots,m,\\ X_p^L&:=\{(i_1h,i_2h,i_3h):1\le i_1\le n,1\le i_2\le n,(p-1)b+1\le i_3\le \gamma+pb-1\},\\ &\quad p=1,\dots,m-1.\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

4.1.2. Auxiliary problems. For each $q = 1, \ldots, m$, $H_q^M \boldsymbol{v} = \boldsymbol{g}$ is defined as the discretization on X_q^M of

$$\begin{cases} \left(\partial_1^2 + \partial_2^2 + (s_q^M(x_3)\partial_3)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = g(x) & \forall x \in D_q^M, \\ v(x) = 0 & \forall x \in \partial D_q^M. \end{cases}$$

For $p = 2, \ldots, m$, $H_p^R \boldsymbol{v} = \boldsymbol{g}$ is defined as the discretization on X_p^R of

$$\begin{cases} \left(\partial_1^2 + \partial_2^2 + (s_p^R(x_3)\partial_3)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = 0 & \forall x \in D_p^R, \\ v(x) = w(x_1, x_2) & \forall x \in \partial^L D_p^R, \\ v(x) = 0 & \forall x \in \partial D_p^R \setminus \partial^L D_p^R, \end{cases}$$

where $\boldsymbol{g} := (-1/h^2)[\boldsymbol{w}^T, 0, \ldots, 0]^T$ and $\boldsymbol{w} := [w_{1,1}, \ldots, w_{n,1}, \ldots, w_{n,n}]$ is the discrete boundary value. Finally, for $p = 1, \ldots, m-1$, $H_p^L \boldsymbol{v} = \boldsymbol{g}$ is the discretization on X_p^L of

$$\begin{cases} \left(\partial_1^2 + \partial_2^2 + (s_p^L(x_3)\partial_3)^2 + \dfrac{\omega^2}{c^2(x)}\right) v(x) = 0 & \forall x \in D_p^L, \\ v(x) = w(x_1, x_2) & \forall x \in \partial^R D_p^L, \\ v(x) = 0 & \forall x \in \partial D_p^L \setminus \partial^R D_p^L, \end{cases}$$

where $\boldsymbol{g} := (-1/h^2)[0, \ldots, 0, \boldsymbol{w}^T]^T$ and $\boldsymbol{w} := [w_{1,1}, \ldots, w_{n,1}, \ldots, w_{n,n}]$.

4.1.3. Auxiliary Green's operators. For $q = 1, \ldots, m$, $\tilde{G}_q^M : \boldsymbol{y} \mapsto \boldsymbol{z}$ is defined using the following operations:

1. Introduce a vector $\boldsymbol{g}$ defined on X_q^M by setting $\boldsymbol{y}$ to X_q and zero otherwise.
2. Solve $H_q^M \boldsymbol{v} = \boldsymbol{g}$ on X_q^M.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_q.

For $p = 2, \ldots, m$, $\tilde{G}_p^R : \boldsymbol{w} \mapsto \boldsymbol{z}$ is given by the following:

1. Set $\boldsymbol{g} = (-1/h^2)[\boldsymbol{w}^T, 0, \ldots, 0]^T$.
2. Solve $H_p^R \boldsymbol{v} = \boldsymbol{g}$ on X_p^R.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_p.

Finally, for $p = 1, \ldots, m-1$, the operator $\tilde{G}_p^L : \boldsymbol{w} \mapsto \boldsymbol{z}$ is introduced to be the following:

1. Set $\boldsymbol{g} = (-1/h^2)[0, \ldots, 0, \boldsymbol{w}^T]^T$.
2. Solve $H_p^L \boldsymbol{v} = \boldsymbol{g}$ on X_p^L.
3. Set $\boldsymbol{z}$ as the restriction of $\boldsymbol{v}$ on X_p.

4.1.4. Putting together. In the 3D case, $\boldsymbol{g}^L$ and $\boldsymbol{g}^R$ for the column-major ordering array $\boldsymbol{g} = [g_{1,1,1}, \ldots, g_{s_1,1,1}, \ldots, g_{s_1,s_2,1}, \ldots, g_{s_1,s_2,s_3}]^T$ induced from some 3D grid with size $s_1 \times s_2 \times s_3$ are given by

$$\boldsymbol{g}^L := [g_{1,1,1}, \ldots, g_{s_1,1,1}, \ldots, g_{s_1,s_2,1}]^T, \quad \boldsymbol{g}^R := [g_{1,1,s_3}, \ldots, g_{s_1,1,s_3}, \ldots, g_{s_1,s_2,s_3}]^T.$$

The subproblems $H_q^M \boldsymbol{v} = \boldsymbol{g}$, $H_p^R \boldsymbol{v} = \boldsymbol{g}$, and $H_p^L \boldsymbol{v} = \boldsymbol{g}$ are quasi-2D. To solve them, we group the elements along dimension 3 first, and then apply the nested dissection method [11, 5] to them, as in [7]. This gives the setup process of the 3D preconditioner in Algorithm 3 and the application process in Algorithm 4.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 3 Construction of the 3D additive sweeping preconditioner of the system (5). Complexity $= O(n^4(b+\gamma)^3/b) = O(N^{4/3}(b+\gamma)^3/b)$.

for $q = 1, \ldots, m$ **do**
  Construct the nested dissection factorization of H_q^M, which defines $\tilde{G}_q^M$.
end for
for $p = 2, \ldots, m$ **do**
  Construct the nested dissection factorization of H_p^R, which defines $\tilde{G}_p^R$.
end for
for $p = 1, \ldots, m-1$ **do**
  Construct the nested dissection factorization of H_p^L, which defines $\tilde{G}_p^L$.
end for

Algorithm 4 Computation of $\tilde{\boldsymbol{u}} \approx G\boldsymbol{f}$ using the preconditioner from Algorithm 3. Complexity $= O(n^3 \log n(b+\gamma)^2/b) = O(N \log N(b+\gamma)^2/b)$.

for $q = 1, \ldots, m$ **do**
  $\tilde{\boldsymbol{u}}_{q,q} = \tilde{G}_q^M \boldsymbol{f}_q$
end for
for $p = 2, \ldots, m$ **do**
  $\tilde{\boldsymbol{u}}_{p,1:p-1} = \tilde{G}_p^R \tilde{\boldsymbol{u}}_{p-1,1:p-1}^R$
  $\tilde{\boldsymbol{u}}_{p,1:p}^R = \tilde{\boldsymbol{u}}_{p,1:p-1}^R + \tilde{\boldsymbol{u}}_{p,p}^R$
end for
for $p = m-1, \ldots, 1$ **do**
  $\tilde{\boldsymbol{u}}_{p,p+1:m} = \tilde{G}_p^L \tilde{\boldsymbol{u}}_{p+1,p+1:m}^L$
  $\tilde{\boldsymbol{u}}_{p,p:m}^L = \tilde{\boldsymbol{u}}_{p,p}^L + \tilde{\boldsymbol{u}}_{p,p+1:m}^L$
end for
$\tilde{\boldsymbol{u}}_1 = \tilde{\boldsymbol{u}}_{1,1} + \tilde{\boldsymbol{u}}_{1,2:m}$
for $p = 2, \ldots, m-1$ **do**
  $\tilde{\boldsymbol{u}}_p = \tilde{\boldsymbol{u}}_{p,1:p-1} + \tilde{\boldsymbol{u}}_{p,p} + \tilde{\boldsymbol{u}}_{p,p+1:m}$
end for
$\tilde{\boldsymbol{u}}_m = \tilde{\boldsymbol{u}}_{m,1:m-1} + \tilde{\boldsymbol{u}}_{m,m}$

For the algorithm analysis, we notice that each quasi-2D subproblem has $O(\gamma+b)$ layers along the third dimension. Therefore, the setup cost for each subproblem is $O((\gamma+b)^3 n^3)$ and the application cost is $O((\gamma+b)^2 n^2 \log n)$. Taking the total number of subproblems into account, the total setup cost for the 3D preconditioner is $O(n^4(b+\gamma)^3/b)$ and the total application cost is $O(n^3 \log n(b+\gamma)^2/b)$.

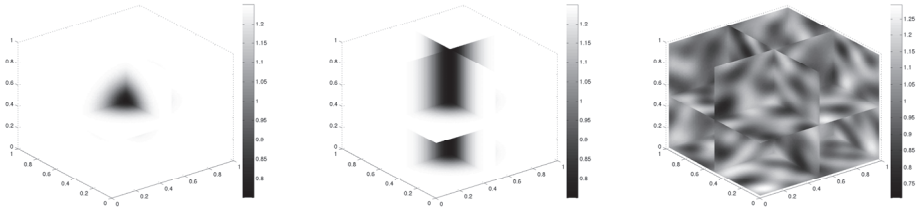
4.2. Numerical results. Here we present the numerical results in three dimensions. All the settings and notations are kept the same as in subsection 3.2 unless otherwise stated. The PMLs are put on all sides of the boundary and the symmetric version of the equation is adopted to save memory cost. The PML width is $\eta = 9h$ for the boundary and is $\eta_{\text{aux}} = 5h$ for the interior auxiliary ones. The number of layers in each subdomain is $b = 4$ for the interior ones, $b+\gamma-1 = 12$ for the leftmost one, and $b+\gamma-2 = 11$ for the rightmost one.

The velocity fields tested are (see Figure 5) the following:

(a) A converging lens with a Gaussian profile at the center of the domain.
(b) A vertical waveguide with a Gaussian cross-section.
(c) A random velocity field.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



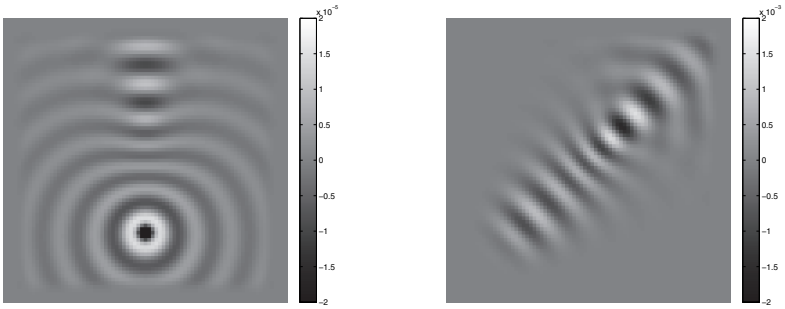
(a) (b) (c)

FIG. 5. *The three velocity fields tested in three dimensions.*

The forces tested for each velocity field are the following:

(a) A Gaussian point source centered at $(1/2, 1/2, 1/4)$.
(b) A Gaussian wave packet with wavelength comparable to the typical wavelength of the domain. The packet centers at $(1/2, 1/4, 1/4)$ and points to the direction $(0, 1/\sqrt{2}, 1/\sqrt{2})$.

TABLE 4
Results for velocity field (a) *in three dimensions. Solutions with* $\omega/(2\pi) = 10$ *at* $x_1 = 0.5$ *are presented.*



Force (a) Force (b)

Velocity field (a)			Force (a)		Force (b)	
$\omega/(2\pi)$	N	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{solve}
5	39^3	2.3304e+01	3	2.9307e+00	4	3.7770e+00
10	79^3	3.2935e+02	3	3.6898e+01	4	4.6176e+01
20	159^2	4.2280e+03	4	4.3999e+02	4	4.6941e+02

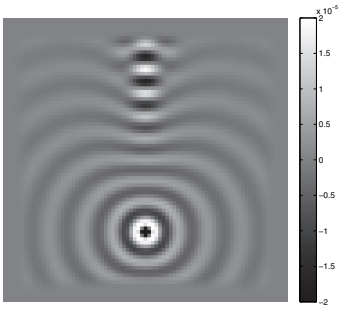
The results are given in Tables 4 to 6. From these tests we see that the iteration number grows mildly as the problem size grows. We also notice that the setup cost scales even better than $O(N^{4/3})$, mainly because MATLAB performs dense linear algebra operations in a parallel way, which gives some extra advantages to the nested dissection algorithm as the problem size grows.

5. Conclusion. In this paper, we proposed a new additive sweeping preconditioner for the Helmholtz equation based on the PML. When combined with the standard GMRES solver, the iteration number grows mildly as the problem size grows. The novelty of this approach is that the unknowns are split in an additive way and the boundary values of the intermediate results are utilized directly. The disadvan-

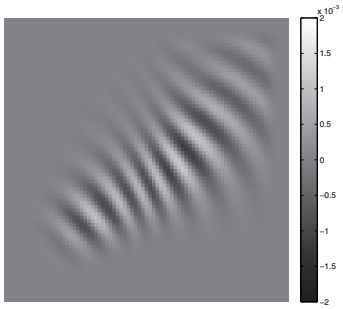
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 5
Results for velocity field (b) *in three dimensions. Solutions with* $\omega/(2\pi) = 10$ *at* $x_1 = 0.5$ *are presented.*



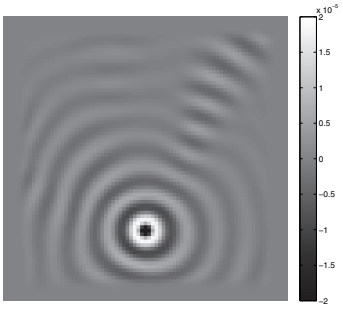
Force (a)



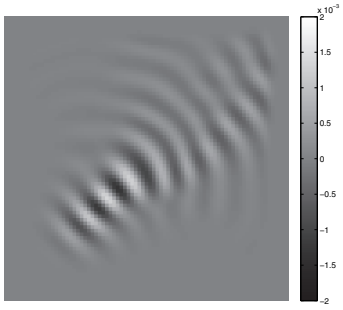
Force (b)

Velocity field (b)			Force (a)		Force (b)	
$\omega/(2\pi)$	N	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{solve}
5	39^3	2.1315e+01	3	2.7740e+00	3	2.7718e+00
10	79^3	3.4256e+02	4	4.4286e+01	3	3.4500e+01
20	159^2	4.3167e+03	5	5.7845e+02	4	4.6462e+02

TABLE 6
Results for velocity field (c) *in three dimensions. Solutions with* $\omega/(2\pi) = 10$ *at* $x_1 = 0.5$ *are presented.*



Force (a)



Force (b)

Velocity field (c)			Force (a)		Force (b)	
$\omega/(2\pi)$	N	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{solve}
5	39^3	2.1063e+01	4	3.8074e+00	4	3.7975e+00
10	79^3	3.4735e+02	4	4.4550e+01	4	4.5039e+01
20	159^2	4.3391e+03	4	4.4361e+02	5	5.8090e+02

tage is that, for each subdomain, three subproblems need to be built up, which is time consuming compared to [7, 15]. However, the costly parts of the algorithm, i.e., the whole setup process and the solve processes of the subproblems $H_q^M \boldsymbol{v} = \boldsymbol{g}$, can be done in parallel. The only parts that must be implemented sequentially are the accumulations of the left-going and right-going waves, where only the solve processes of the subproblems $H_p^L \boldsymbol{v} = \boldsymbol{g}$ and $H_p^R \boldsymbol{v} = \boldsymbol{g}$ are involved, which are the cheapest parts

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

of the algorithm. Besides, we think that the whole approximation process is simple and structurally clear from a physics point of view, and the idea might be easily generalized to other equations.

There are also some other directions to make potential improvements. First, other numerical schemes of the equation and other approximations of the Sommerfeld radiation condition can be used to develop more efficient versions of this additive preconditioner. Second, the parallel version of the nested dissection algorithm can be combined to solve large scale problems. Last, in the 3D case, the quasi-2D subproblems can be solved recursively by sweeping along the x_2 direction with the same technique, which reduces the theoretical setup cost to $O(N)$ and the application cost to $O(N)$. However, compared to [7], the coefficient of the complexity in this new method is larger, so it is not clear whether or not the recursive approach will be more efficient practically. Nevertheless, it is of great theoretical interest to look into it.

Acknowledgments. We thank Lenya Ryzhik for providing computing resources and the anonymous reviewers for comments.

REFERENCES

[1] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200, http://dx.doi.org/10.1006/jcph.1994.1159.

[2] Z. Chen and X. Xiang, *A source transfer domain decomposition method for Helmholtz equations in unbounded domain*, SIAM J. Numer. Anal., 51 (2013), pp. 2331–2356, http://dx.doi.org/10.1137/130917144.

[3] Z. Chen and X. Xiang, *A source transfer domain decomposition method for Helmholtz equations in unbounded domain Part* II: *Extensions*, Numer. Math. Theory Methods Appl., 6 (2013), pp. 538–555, http://dx.doi.org/10.4208/nmtma.2013.1217nm.

[4] W. C. Chew and W. H. Weedon, *A* 3*D perfectly matched medium from modified Maxwell's equations with stretched coordinates*, Microw. Opt. Techn. Let., 7 (1994), pp. 599–604, http://dx.doi.org/10.1002/mop.4650071304.

[5] 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, http://dx.doi.org/10.1145/356044.356047.

[6] B. Engquist and L. Ying, *Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation*, Comm. Pure Appl. Math., 64 (2011), pp. 697–735, http://dx.doi.org/10.1002/cpa.20358.

[7] B. Engquist and L. Ying, *Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers*, Multiscale Model. Simul., 9 (2011), pp. 686–710, http://dx.doi.org/10.1137/100804644.

[8] Y. A. Erlangga, *Advances in iterative methods and preconditioners for the Helmholtz equation*, Arch. Comput. Methods Eng., 15 (2008), pp. 37–66, http://dx.doi.org/10.1007/s11831-007-9013-7.

[9] O. G. Ernst and M. J. Gander, *Why it is difficult to solve Helmholtz problems with classical iterative methods*, in Numerical Analysis of Multiscale Problems, Lect. Notes Comput. Sci. Eng. 83, Springer, Heidelberg, 2012, pp. 325–363, http://dx.doi.org/10.1007/978-3-642-22061-6_10.

[10] M. J. Gander and F. Nataf, *AILU for Helmholtz problems: A new preconditioner based on the analytic parabolic factorization*, J. Comput. Acoust., 9 (2001), pp. 1499–1506, http://dx.doi.org/10.1016/S0764-4442(00)01632-3.

[11] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363, http://dx.doi.org/10.1137/0710032.

[12] S. G. Johnson, *Notes on Perfectly Matched Layers (PMLs)*, Lecture Notes, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[13] F. Liu and L. Ying, *Recursive sweeping preconditioner for the three-dimensional Helmholtz equation*, SIAM J. Sci. Comput., 38 (2016), pp. A814–A832, http://dx.doi.org/10.1137/15M1010154.

[14] J. Poulson, B. Engquist, S. Li, and L. Ying, *A parallel sweeping preconditioner for heterogeneous* 3*D Helmholtz equations*, SIAM J. Sci. Comput., 35 (2013), pp. C194–C212,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

http://dx.doi.org/10.1137/120871985.

[15] C. C. Stolk, *A rapidly converging domain decomposition method for the Helmholtz equation*, J. Comput. Phys., 241 (2013), pp. 240–252, http://dx.doi.org/10.1016/j.jcp.2013.01.039.

[16] P. Tsuji, B. Engquist, and L. Ying, *A sweeping preconditioner for time-harmonic Maxwell's equations with finite elements*, J. Comput. Phys., 231 (2012), pp. 3770–3783, http://dx.doi.org/10.1016/j.jcp.2012.01.025.

[17] P. Tsuji, J. Poulson, B. Engquist, and L. Ying, *Sweeping preconditioners for elastic wave propagation with spectral element methods*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 433–447, http://dx.doi.org/10.1051/m2an/2013114.

[18] P. Tsuji and L. Ying, *A sweeping preconditioner for Yee's finite difference approximation of time-harmonic Maxwell's equations*, Front. Math. China, 7 (2012), pp. 347–363, http://dx.doi.org/10.1007/s11464-012-0191-8.

[19] A. Vion and C. Geuzaine, *Double sweep preconditioner for optimized Schwarz methods applied to the Helmholtz problem*, J. Comput. Phys., 266 (2014), pp. 171–190, http://dx.doi.org/10.1016/j.jcp.2014.02.015.

[20] L. Zepeda-Núñez and L. Demanet, *The method of polarized traces for the* $2D$ *Helmholtz equation*, J. Comput. Phys., 308 (2016), pp. 347–388, http://dx.doi.org/10.1016/j.jcp.2015.11.040.

Downloaded 07/20/16 to 171.67.216.22. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.