INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 13, Number 5, Pages 802–819

PARALLEL PRECONDITIONERS FOR PLANE WAVE HELMHOLTZ AND MAXWELL SYSTEMS WITH LARGE WAVE NUMBERS

LONG YUAN, QIYA HU, AND HENGBIN AN

Abstract. A kind of non-overlapping domain decomposition preconditioner was proposed to solve the systems generated by the plane wave least-squares (PWLS) method for discretization of Helmholtz equation and Maxwell equations respectively in [13] and [14]. In this paper we introduce overlapping variants of this kind of preconditioner and give some comparison among these domain decomposition preconditioners. The main goal of this paper is to implement in parallel these domain decomposition preconditioners for the system with large wave numbers. The numerical results indicate that the preconditioners are highly scalable and are effective for solving Helmholtz equation and Maxwell's equations with large wave numbers.

Key words. Helmholtz equations, Maxwell's equations, large wave number, variational formulation, plane-wave basis, preconditioner, iteration counts.

1. Introduction

The plane-wave method differs from the traditional finite-element method (FEM) and the boundary-element method (BEM) in the sense that the basis functions are chosen as exact solutions of the governing differential equation without boundary conditions. This type of numerical method was first introduced to solve Helmholtz equations and was then extended to solve Maxwell's equations. Examples of this approach include the discontinuous enrichment method [1, 9], the variational theory of complex rays (VTCR) [21, 22], the ultra weak variational formulation (UWVF) [3, 4, 16], the plane-wave discontinuous Galerkin (PWDG) method [10, 12, 26], and the plane-wave least-squares (PWLS) method [20, 13, 14].

All methods described above fall into the class of Trefftz methods. An important advantage of the PWLS method over the others is that the stiffness matrix generated by the PWLS method is Hermitian positive definite, so it is easier to construct efficient preconditioners for this matrix. For example, a simple non-overlapping domain decomposition preconditioner for such stiffness matrix was constructed in [13] and [14]. The numerical results indicate that the system with middle wave numbers can be solved rapidly by the preconditioned CG method with the proposed preconditioner.

It is a difficult topic to construct an efficient preconditioner for Helmholtz equation or Maxwell's equations with large wave numbers. In fact, the existing domain decomposition methods (and multilevel methods) are inefficient to these equations except that the sizes of the coarse meshes are chosen as $O(1/\omega)$ (see, for example, [5, 8, 17, 25]), where ω denotes the wave number. It is clear that the restriction on the coarse mesh size is fatal in applications. Recently a kinds of successive preconditioners based on PML method are proposed to solve Helmholtz equations with large wave numbers (see [6, 7]). It has been shown that such preconditioners

Received by the editors February 2, 2015 and, in revised form, July 14, 2016.
2000 *Mathematics Subject Classification.* 65N30, 65N55.
This research was supported by the Natural Science Foundation of China G11571352 and G11501529.

possess the optimal convergence independent of the mesh sizes [6]. It is certain that the results are most important advance in the solution method for Helmholtz equations with large wave numbers.

In this paper we are mainly interested in the parallel implementation of domain decomposition preconditioners for the systems generated by the PWLS method for discretization of Helmholtz equation and Maxwell equations with large wave numbers. Motivated by the non-overlapping domain decomposition preconditioner in [13] and [14], we construct overlapping domain decomposition preconditioner for such systems in the present paper. We give some comparison of iteration counts and computing times spent in PCG method with the non-overlaping preconditioner and the overlaping preconditioner. Numerical results indicate that the domain decomposition preconditioner with small overlap is more effective than the others when the wave number is large and the mesh size is small. In particular, we implement in parallel the domain decomposition preconditioner with one element overlap for solving the systems with large wave numbers, and we find that such preconditioner is strongly scalable and is very effective, without the restriction that the size of coarse meshes is $O(1/\omega)$.

The paper is organized as follows: In Section 2, we recall the proposed variational formulation for homogeneous Helmholtz equation and Maxwell's equations. In Section 3, we describe the plane wave discretization of the variational problem. In Section 4, we construct domain decomposition preconditioners for the stiffness matrix associated with the new variational problem. In Section 5, we address some key issues in the parallel implementation of the domain decomposition preconditioner in JASMIN framework. In Section 6, we report some numerical results to confirm the effectiveness of the new preconditioner for solving the system with large wave numbers.

2. Variational formulation for Helmholtz equation and Maxwell's equations

In this section we recall the Helmholtz equation and second-order system of Maxwell's equations.

The considered variational formulation is based on a triangulation of the solution domain. Suppose Ω is a bounded polyhedral domain in $\mathbb{R}^n$ $(n = 2, 3)$. Let Ω be divided into a partition in the sense that

$$\overline{\Omega} = \bigcup_{k=1}^{N} \overline{\Omega}_k, \quad \Omega_l \bigcap \Omega_j = \emptyset \quad \text{for } l \neq j.$$

Let $\mathcal{T}_h$ denote the triangulation comprising the elements $\{\Omega_k\}$, where h is the meshwidth of the triangulation. Define

$$\Gamma_{lj} = \partial\Omega_l \bigcap \partial\Omega_j \quad \text{for } l \neq j$$

and

$$\gamma_k = \overline{\Omega}_k \bigcap \partial\Omega \quad (k = 1, \ldots, N),\ \gamma = \bigcup_{k=1}^{N} \gamma_k.$$

2.1. The case of Helmholtz equation. Consider Helmholtz equations which is formalized, normalizing the wave's velocity to 1, by

$$\begin{cases} -\Delta u - \omega^2 u = 0 & \text{in} \quad \Omega, \\ (\partial_{\mathbf{n}} + i\omega)u = g & \text{on} \quad \gamma. \end{cases} \tag{1}$$

The outer normal derivative is referred to by $\partial_{\mathbf{n}}$ and the angular frequency by ω.

Set $u|_{\Omega_k} = u_k$ $(k = 1, \cdots, N)$. Then the reference problem (1) to be solved consists in finding the local acoustic pressures $u_k \in H^1(\Omega_k)$ such that

$$(2)\quad \begin{cases} -\Delta u_k - \omega^2 u_k = 0 & \text{in} \quad \Omega_k, \\ (\partial_{\mathbf{n}} + i\omega)u_k = g & \text{on} \quad \gamma \ (if\ \partial\Omega_k \cap \gamma \neq \emptyset) \end{cases} \qquad (k = 1, 2, \ldots, N),$$

and

$$(3)\quad \begin{cases} u_k - u_j = 0 & \text{over} \quad \Gamma_{kj}, \\ \partial_{\mathbf{n}_k} u_k + \partial_{\mathbf{n}_j} u_j = 0 & \text{over} \quad \Gamma_{kj} \end{cases} \qquad (k \neq j;\ k, j = 1, 2, \cdots, N).$$

Define

$$V(\mathcal{T}_h) = \{v \in L^2(\Omega);\ \Delta v + \omega^2 v = 0 \text{ in each } \Omega_k\}.$$

In the PWLS method (see [13]), the variational problem of (2) and (3) can be expressed as follows: find $u \in V(\mathcal{T}_h)$ such that

$$\begin{aligned}(4)\quad &\sum_{k=1}^{N} \int_{\gamma_k} ((\partial_{\mathbf{n}} + i\omega)u_k - g) \cdot \overline{(\partial_{\mathbf{n}} + i\omega)v_k} ds + \sum_{j \neq k} \Big(\alpha \int_{\Gamma_{kj}} (u_k - u_j) \cdot \overline{(v_k - v_j)} ds \\ &+ \beta \int_{\Gamma_{kj}} (\partial_{\mathbf{n}_k} u_k + \partial_{\mathbf{n}_j} u_j) \cdot \overline{(\partial_{\mathbf{n}_k} v_k + \partial_{\mathbf{n}_j} v_j)} ds \Big) = 0,\ \forall v \in V(\mathcal{T}_h),\end{aligned}$$

where α and β are given positive numbers. In general we choose $\alpha = \omega^2$ and $\beta = 1$.

Equivalently, Eq.(4) can be written as

$$(5)\quad \begin{cases} \text{Find } u \in V(\mathcal{T}_h)\ s.t. \\ a(u, v) = (\xi,\ v)_V, \quad \forall v \in V(\mathcal{T}_h), \end{cases}$$

where the sesquilinear form $a(\cdot, \cdot)$ and the linear functional $(\xi,\ \cdot)_V$ were defined exactly in [13].

2.2. The case of Maxwell's equations. Consider time-harmonic Maxwell equations in three-dimensional (3D) domain, written as a second-order system of equations:

$$(6)\quad \begin{cases} \nabla \times (\frac{1}{i\omega\mu} \nabla \times \mathbf{E}) + i\omega\varepsilon \mathbf{E} = 0 & \text{in} \quad \Omega, \\ -\mathbf{E} \times \mathbf{n} + \frac{\sigma}{i\omega\mu}((\nabla \times \mathbf{E}) \times \mathbf{n}) \times \mathbf{n} = \mathbf{g} & \text{on} \quad \gamma, \end{cases}$$

Here, $\omega > 0$ is the temporal frequency of the field, and $\mathbf{g} \in L^2_{\mathbf{T}}(\partial\Omega)$. The material coefficients ε, μ and σ are understood as usual. In particular, if ε is complex valued, then the material is known as an absorbing medium; otherwise the material is called a non-absorbing medium.

For each element Ω_k, let $\mathbf{E}|_{\Omega_k} = \mathbf{E}_k$ $(k = 1, \ldots, N)$. As usual, we assume that ω, μ and ε are constants on each element. Then the reference problem (6) to be solved consists of finding the local electric field $\mathbf{E}_k$ such that

$$\nabla \times (\frac{1}{i\omega\mu} \nabla \times \mathbf{E}_k) + i\omega\varepsilon \mathbf{E}_k = 0 \qquad \text{in} \quad \Omega_k,$$

namely,

$$(7)\quad \nabla \times (\nabla \times \mathbf{E}) - \kappa^2 \mathbf{E} = 0 \text{ in} \quad \Omega_k$$

with $\kappa = \omega\sqrt{\mu\varepsilon}$, and the interface conditions (note that $\mathbf{n}_l = -\mathbf{n}_j$)

$$(8)\quad \begin{cases} \mathbf{E}_l \times \mathbf{n}_l + \mathbf{E}_j \times \mathbf{n}_j = 0 \\ (\frac{1}{i\omega\mu} \nabla \times \mathbf{E}_l) \times \mathbf{n}_l + (\frac{1}{i\omega\mu} \nabla \times \mathbf{E}_j) \times \mathbf{n}_j = 0 \end{cases} \text{on} \quad \Gamma_{lj} \quad (l < j;\ l, j = 1, 2, \ldots, N).$$

The boundary condition becomes

$$-\mathbf{E}_k \times \mathbf{n} + \frac{\sigma}{i\omega\mu}((\nabla \times \mathbf{E}_l) \times \mathbf{n}) \times \mathbf{n} = \mathbf{g} \qquad \text{on} \quad \gamma_k = \partial\Omega_k \cap \gamma. \tag{9}$$

For $\mathbf{F} \in \mathbf{V}(\mathcal{T}_h)$, set $\mathbf{F}|_{\Omega_k} = \mathbf{F}_k$. For ease of notation, define

$$\Phi(\mathbf{F_k}) = \frac{\sigma}{i\omega\mu}((\nabla \times \mathbf{F_k}) \times \mathbf{n_k}) \qquad \text{on} \quad \gamma_k = \partial\Omega_k \cap \gamma.$$

For each Ω_k, set

$$\Psi(\mathbf{F}_k) = \frac{1}{i\omega\mu}(\nabla \times \mathbf{F}_k), \text{ on } \Omega_k.$$

Let ρ_1 and ρ_2 be two given constants, which satisfy $\rho_1 \neq \rho_2$. A simple choice of the constants is $\rho_1 = 1$ and $\rho_2 = -1$. For each local interface Γ_{lj} $(l < j)$, we define the jumps on Γ_{lj} as follows (note that $\mathbf{n}_l = -\mathbf{n}_j$):

$$\begin{aligned}[[\mathbf{F} \times \mathbf{n} + \rho_\mathbf{r}\mathbf{\Psi}(\mathbf{F}) \times \mathbf{n}]] &= [[\mathbf{F} \times \mathbf{n}]] + \rho_\mathbf{r}[[(\mathbf{\Psi}(\mathbf{F}) \times \mathbf{n})]] \\ &= \mathbf{F}_l \times \mathbf{n}_l + \mathbf{F}_j \times \mathbf{n}_j + \rho_r\Big(\Psi(\mathbf{F}_l) \times \mathbf{n}_l + \Psi(\mathbf{F}_j) \times \mathbf{n}_j\Big) \ (r = 1, 2).\end{aligned} \tag{10}$$

In the PWLS method (see [14]), the variational problem of (7), (8), and (9) can be expressed as follows: Find $\mathbf{E} \in V(\mathcal{T}_h)$ such that

$$\begin{aligned}&\sum_{k=1}^{N} \int_{\gamma_k} (-\mathbf{E}_k \times \mathbf{n}_k + \Phi(\mathbf{E}_k) \times \mathbf{n}_k - \mathbf{g}) \cdot \overline{-\mathbf{F}_k \times \mathbf{n}_k + \Phi(\mathbf{F}_k) \times \mathbf{n}_k} ds \\ &+ \sum_{l<j} \left(\int_{\Gamma_{lj}} [[\mathbf{E} \times \mathbf{n} + \rho_1 \Psi(\mathbf{E}) \times \mathbf{n}]] \cdot \overline{[[\mathbf{F} \times \mathbf{n} + \rho_1 \Psi(\mathbf{F}) \times \mathbf{n}]]}\ ds \right. \\ &\left. + \int_{\Gamma_{lj}} [[\mathbf{E} \times \mathbf{n} + \rho_2 \Psi(\mathbf{E}) \times \mathbf{n}[[\cdot \overline{[[\mathbf{F} \times \mathbf{n} + \rho_2 \Psi(\mathbf{F}) \times \mathbf{n}]]}\ ds \right) = 0, \ \forall\ \mathbf{F} \in V(\mathcal{T}_h).\end{aligned} \tag{11}$$

Equivalently, Eq.(11) can be written as

$$\begin{cases} \text{Find } \mathbf{E} \in V(\mathcal{T}_h) \ s.t. \\ A(\mathbf{E}, \mathbf{F}) = (\boldsymbol{\xi},\ \mathbf{F})_\mathbf{V}, \quad \forall \mathbf{F} \in V(\mathcal{T}_h), \end{cases} \tag{12}$$

where the sesquilinear form $A(\cdot, \cdot)$ and the linear functional $(\boldsymbol{\xi},\ \cdot)_\mathbf{V}$ were defined exactly in [14].

3. Discretization of the variational problems

In this section, we describe the discretization of the variational problem. The discretization is based on a finite-dimensional space $V_p(\mathcal{T}_h) \subset V(\mathcal{T}_h)$. We first give the precise definition of such a space $V_p(\mathcal{T}_h)$.

3.1. Basis functions of the finite-dimensional space. In this section we construct basis functions with which to discretize the PWLS method.

3.1.1. The case of Helmholtz equation. In each element Ω_k, we introduce a finite number of functions y_{kl} $(l = 1, 2, \cdots, p)$ supported in Ω_k and that are independent solutions of the homogeneous Helmholtz equation (without boundary condition) in the element Ω_k $(k = 1, 2, \cdots, N)$.

To simplify, we consider some constant number p of basis functions for all elements Ω_k. Particularly, in this paper we will choose y_{kl} as the wave shape functions on Ω_k, which satisfy

$$\begin{cases} y_{kl}(\mathbf{x}) = e^{i\omega(\mathbf{x}\cdot\boldsymbol{\alpha}_l)},\ \mathbf{x}\in\Omega_k, \\ \boldsymbol{\alpha_l}\cdot\boldsymbol{\alpha_l} = 1, \\ l\neq s \to \boldsymbol{\alpha_l}\neq\boldsymbol{\alpha_s}, \end{cases} \tag{13}$$

where $\boldsymbol{\alpha_l}$ $(l=1,\cdots,p)$ are unit wave propagation directions to be specified later.

The basis functions of $V_p(\mathcal{T}_h)$ can be defined as

$$\phi_{kl}(\mathbf{x}) = \begin{cases} y_{kl}(\mathbf{x}),\ \mathbf{x}\in\Omega_k, \\ 0,\ \mathbf{x}\in\Omega_j \text{ satisfying } j\neq k \end{cases} \quad (k,j=1,\cdots,N;\ l=1,\cdots,p). \tag{14}$$

Thus the space $V(\mathcal{T}_h)$ is discretized by the subspace

$$V_p(\mathcal{T}_h) = \text{span}\Big\{\phi_{kl}:\ k=1,\cdots,N;\ l=1,\cdots,p\Big\}. \tag{15}$$

During numerical simulations, the directions of the wave vectors of these wave functions, for two-dimensional problems, are uniformly distributed as follows:

$$\boldsymbol{\alpha_l} = \begin{pmatrix} (cos(2\pi(l-1)/p) \\ sin(2\pi(l-1)/p)) \end{pmatrix}\ (l=1,\cdots,p).$$

For three-dimensional problems, we use the optimal spherical codes from [23] to generate the wave propagation derections $\boldsymbol{\alpha_l}, l=1,\cdots,p$, where $p=(q+1)^2, q\in\mathbb{N}$.

3.1.2. The case of Maxwell's equations. In practice, following [2], a suitable family of plane waves, which are solutions of the constant-coefficient Maxwell equations, are generated on Ω_k by choosing p unit propagation directions $\mathbf{d}_l,\ l=1,\ldots,p$ (we also use the optimal spherical codes from [23]), and defining a real unit polarization vector $\mathbf{G}_l$ orthogonal to $\mathbf{d}_l$. Then the propagation directions and polarization vectors define the complex polarization vectors $\mathbf{F}_l$ and $\mathbf{F}_{l+p}$ by

$$\mathbf{F}_l = \mathbf{G}_l + \mathrm{i}\mathbf{G}_l\times\mathbf{d}_l, \quad \mathbf{F}_{l+p} = \mathbf{G}_l - \mathrm{i}\mathbf{G}_l\times\mathbf{d}_l\ (l=1,\ldots,p).$$

Note that the complex polarization vectors are the same as in [4, 16], but differ slightly from those in [12]. We then define the complex functions $\mathbf{E}_l$:

$$\mathbf{E}_l = \sqrt{\mu}\ \mathbf{F}_l\ \exp(\mathrm{i}\kappa\mathbf{d}_l^*\cdot\mathbf{x})\ (l=1,\ldots,2p), \tag{16}$$

where $d_l^* = d_l$ when $l=1,\cdots,p$ and $d_l^* = d_{l-p}$ when $l=p+1,\cdots,2p$. It is easy to verify that every function $\mathbf{E}_l$ $(l=1,\ldots,2p)$ satisfies Maxwell's system (7).

Let $\mathcal{Q}_{2p}$ denote the space spanned by the $2p$ plane-wave functions $\mathbf{E}_l$ $(l=1,\ldots,2p)$. Define the finite-element space

$$\mathbf{V}_p(\mathcal{T}_h) = \Big\{\mathbf{v}\in L^2(\Omega):\ \mathbf{v}|_K\in\mathcal{Q}_{2p} \text{ for any } K\in\mathcal{T}_h\Big\}. \tag{17}$$

It is easy to see that the above space has $N\times 2p$ basis functions, which are defined by

$$\phi_l^k(\mathbf{x}) = \begin{cases} \mathbf{E}_l(\mathbf{x}),\ \mathbf{x}\in\Omega_k, \\ 0,\ \mathbf{x}\in\Omega_j \text{ satisfying } j\neq k \end{cases} \quad (k=1,\ldots,N;\ l=1,\ldots,2p). \tag{18}$$

3.2. Discrete variational problems and algebraic forms. Using the definition of finite dimensional plane wave discrete space $V_p(\mathcal{T}_h) \subset V(\mathcal{T}_h)$ in [13], then the discrete variational problem associated with (5) can be described as follows:

$$\begin{cases} \text{Find } u_h \in V_p(\mathcal{T}_h) \ \ s.t. \\ a(u_h, v_h) = (\xi,\ v_h)_V,\ \forall v_h \in V_p(\mathcal{T}_h). \end{cases} \tag{19}$$

Using the definition of finite dimensional plane wave discrete space $V_p(\mathcal{T}_h) \subset V(\mathcal{T}_h)$ in [14], then the discrete variational problem associated with (12) can be described as follows:

$$\begin{cases} \text{Find } \mathbf{E}_h \in V_p(\mathcal{T}_h) \ \ s.t. \\ A(\mathbf{E}_h, \mathbf{F}_h) = (\boldsymbol{\xi},\ \mathbf{F}_h)_{\mathbf{V}},\ \forall \mathbf{F}_h \in V_p(\mathcal{T}_h). \end{cases} \tag{20}$$

Let $\mathcal{A}$ be the stiffness matrix associated with the sesquilinear form $a(\cdot,\cdot)$ or $A(\cdot,\cdot)$, and let b denote the vector associated with the vector product $(\xi,\ \cdot)_V$ or $(\boldsymbol{\xi},\ \cdot)_{\mathbf{V}}$. In this case, the discretized problem (19) and (20) leads to the algebraic system

$$\mathcal{A}X = b. \tag{21}$$

In general system (21) is solved by an iterative method, for example, the preconditioned CG method. Then we need to construct an efficient preconditioner $\mathcal{B}$ for the matrix $\mathcal{A}$, and use CG method to solve the equivalent system

$$\mathcal{B}^{-1}\mathcal{A}X = \mathcal{B}^{-1}b. \tag{22}$$

In Section 4, we construct an effective preconditioner $\mathcal{B}$.

3.3. Error estimates of the approximate solutions. In this subsection we give some error estimates of the approximations u_h defined by (19).

As in [11], for a given a domain $D \subset \mathbb{R}^l, (l = 2, 3)$, let $||\cdot||_{s,\omega,D}$ be the $\omega-$ weighted Sobolev norm defined by

$$||v||_{s,\omega,D} = \sum_{j=0}^{s} \omega^{2(s-j)} |v|^2_{j,D}.$$

The following error estimates can be obtained by Theorem 5.2, Theorem 5.3 in [19] and Theorem 4.1 in [13].

Theorem 3.1 [19] *For an integer $q \geq 2$, set $p = 2q+1$ (for 2d case) or $p = (q+1)^2$ (for 3d case). Assume that $u \in C^{r+1}(\Omega_k)$ for each element Ω_k and the assumptions of Theorem 5.2 and Theorem 5.3 in* [19] *are satisfied. Let u and u_h denote the solutions of (5) and (19), respectively. Then*

$$\begin{cases} ||u - u_h||_V \leq Ch^{r-\frac{1}{2}} \left(\frac{\log p}{p}\right)^{r-\frac{1}{2}} (\sum_{k=1}^{N} ||u||^2_{\omega,m+1,\Omega_k})^{\frac{1}{2}}, \\ ||u - u_h||_{0,\Omega} \leq C(1 + (h\omega)^{-1}) h^r \left(\frac{\log p}{p}\right)^{r-\frac{1}{2}} (\sum_{k=1}^{N} ||u||^2_{\omega,m+1,\Omega_k})^{\frac{1}{2}} \end{cases} \tag{23}$$

for 2d case, and

$$\begin{cases} ||u - u_h||_V \leq Ch^{r-\frac{1}{2}} q^{-\lambda(r-\frac{1}{2})} (\sum_{k=1}^{N} ||u||^2_{\omega,m+1,\Omega_k})^{\frac{1}{2}}, \\ ||u - u_h||_{0,\Omega} \leq C(1 + (h\omega)^{-1}) h^r q^{-\lambda(r-\frac{1}{2})} (\sum_{k=1}^{N} ||u||^2_{\omega,m+1,\Omega_k})^{\frac{1}{2}} \end{cases} \tag{24}$$

for 3d case, where C is the constant independent of p and u, but dependent on ω and h only through the product ωh as an increasing function, and may depend on the shape of the elements $K \in \mathcal{T}_h$ and r.

Assume that the mesh triangulation $\mathcal{T}_h$ satisfies the definition stated in Ref. [12, Section 5] and set $\lambda = \min_{K\in\mathcal{T}_h}\lambda_K$, where λ_K is the positive parameter that depends only on the shape of an element K of $\mathcal{T}_h$ introduced in Ref. [19, Th. 3.2]. Let r and m be given positive integers satisfying $m \geq 2r-1$ and $m \geq 2(1+2^{1/\lambda})$. Let the number p of plane wave propagation directions be chosen as $p = (m+1)^2$.

The error estimates of the approximate solutions $\mathbf{E}_h$ of Maxwell's equations can be obtained from [14, Th. 4.3] (refer to Theorem 5.4 of [12]).

Theorem 3.2 L*et λ, m and p be defined above. Assume that $\mathbf{E} \in H^{r+1}(curl;\Omega)$. Let $\mathbf{E}$ and $\mathbf{E}_h$ denote the solutions of (12) and (20), respectively. Then*

$$||\mathbf{E}-\mathbf{E}_h||_{\mathbf{V}} \leq C\omega^{-2}\left(\frac{h}{m^{\lambda}}\right)^{(r-\frac{3}{2})}||\nabla\times\mathbf{E}||_{r+1,\omega,\Omega}. \tag{25}$$

4. Domain decomposition preconditioners $\mathcal{B}$

In this section, we construct the preconditioner $\mathcal{B}$ based on the domain decomposition method.

4.1. Space decomposition of $V_p(\mathcal{T}_h)$. We first coarsen the triangulation $\{\Omega_k\}$ as follows: let Ω be decomposed into the union of non-overlapping coarse elements $D_1, D_2, \ldots, D_{n_0}$ such that D_r is just the union of several elements in $\{\Omega_k\}$.

Each coarse element D_r is extended to a larger region $D_r^{'}$, i.e. $D_r \subset D_r^{'}$. We assume that each $D_r^{'}$ is also the union of some elements in $\{\Omega_k\}$, i.e., $\partial D_r^{'}$ does not cut through any fine elements.

The domain Ω has the following domain decomposition:

$$\begin{cases} \text{nonoverlapping domain decomposition}\{D_r\}, D_r \bigcap D_l = \emptyset \text{ for } r \neq l, \\ \text{overlapping domain decomposition}\{D_r^{'}\}, D_r^{'} \bigcap D_l^{'} \neq \emptyset \text{ for } r \neq l. \end{cases} \tag{26}$$

For convenience, we use $\mathcal{T}_h^r$ to denote the restriction of the triangulation $\mathcal{T}_h$ on the subdomain (coarse element) D_r $(r = 1, \ldots, n_0)$ or overlapping subdomain $D_r^{'}$ $(r = 1, \ldots, n_0)$. For $r = 1, \ldots, n_0$, define the *local* space

$$V_p(\mathcal{T}_h^r) = \left\{\mathbf{v} \in V_p(\mathcal{T}_h) : \text{ supp } \mathbf{v} \subset \mathcal{T}_h^r\right\};$$

namely, $V_p(\mathcal{T}_h^r)$ is just the restriction space of $V_p(\mathcal{T}_h)$ on the subdomain D_r or $D_r^{'}$.

In order to define coarse subspace, let d denote the meshwidth of coarse elements $D_1, \ldots, D_{n_0}$, and let $\mathcal{T}_d$ denote the triangulation associated with coarse elements $D_1, D_2, \ldots, D_{n_0}$. For the case of Helmholtz equation, set $y_{rl}^d(\mathbf{x}) = e^{i\omega(\mathbf{x}\cdot\overrightarrow{\alpha_l})}(\mathbf{x} \in D_r \text{ or } D_r^{'}; r = 1, \cdots, n_0; l = 1, 2, \cdots, p)$. Define the *coarse* basis functions

$$\tilde{\phi}_{rl} = \begin{cases} y_{rl}^d, & on\ \phi_k\ satisfying\ \phi_k \subset D_r \text{ or } D_r^{'} \\ 0, & on\ \phi_k\ satisfying\ \phi_k \not\subset D_r \text{ or } D_r^{'} \end{cases} \quad (r = 1, \cdots, n_0; l = 1, \cdots, p)$$

and the *coarse* space (refer to [13])

$$V_p(\mathcal{T}_d) = \text{span}\left\{\tilde{\phi}_{rl} : \ r = 1, \cdots, n_0; \ l = 1, \cdots, p\right\}.$$

For the case of Maxwell's equations, define (see [14])

$$V_p(\mathcal{T}_d) = \Big\{ \mathbf{v} \in V_p(\mathcal{T}_h) : \ \mathbf{v}|_K \in \mathcal{Q}_{2p} \text{ for any } K \in \mathcal{T}_d \Big\}.$$

Noting that the *coarse* space $V_p(\mathcal{T}_d)$ has the same structure as the original space $V_p(\mathcal{T}_h)$.

As in the standard overlapping domain decomposition method, we can obtain the space decomposition (here we can easily define weight functions satisfying the partition of unity, since we do not require the continuity of functions in the considered spaces)

$$V_p(\mathcal{T}_h) = \sum_{r=1}^{n_0} V_p(\mathcal{T}_h^r) + V_p(\mathcal{T}_d). \tag{27}$$

If the coarse space $V_p(\mathcal{T}_d)$ is dropped, the above space decomposition still holds, but the corresponding preconditioners will not be scalable.

4.2. Preconditioners. Based on the space decomposition (27), we can construct the desired preconditioner by the general framework of space decomposition. For the purpose of implementation, we would like to describe the preconditioner in algebraic form.

For $r = 1, \ldots, n_0$, let $\mathcal{A}_r$ denote the stiffness matrix induced from the sesquilinear form $A(\cdot, \cdot)$ on the subspace $V_p(\mathcal{T}_h^r)$. We define the transfer matrix $\mathcal{C}_r$ from $V_p(\mathcal{T}_h)$ to $V_p(\mathcal{T}_h^r)$ as the matrix whose entries are simply 1 or 0, then it is easy to verify that $\mathcal{A}_r = \mathcal{C}_r \mathcal{A} \mathcal{C}_r^t$. Particularly, when the order of the basis functions is arranged in a suitable manner, for the nonoverlapping domain decomposition case, the original stiffness matrix $\mathcal{A}$ can be written as a block matrix with the diagonal submatrices $\mathcal{A}_1, \ldots, \mathcal{A}_{n_0}$. Set

$$\mathcal{D} = \operatorname{diag}(\mathcal{A}_1, \ldots, \mathcal{A}_{n_0}).$$

In the following we define the coarse solver. From the definition of $V_p(\mathcal{T}_d)$, we know that each basis function of $V_p(\mathcal{T}_d)$ can be expressed as a linear combination of the basis functions of $V_p(\mathcal{T}_h)$, with the coordinates being 1 or 0. We define the transfer matrix $\mathcal{C}_d$ from $V_p(\mathcal{T}_h)$ to $V_p(\mathcal{T}_d)$ as the matrix whose entries are simply all such coordinates. In other words, $\mathcal{C}_d^t$ is the embedding of the coarse space $V_p(\mathcal{T}_d)$ into the fine space $V_p(\mathcal{T}_h)$. Let $\mathcal{A}_d$ denote the stiffness matrix induced from the sesquilinear form $A(\cdot, \cdot)$ on the coarse subspace $V_p(\mathcal{T}_d)$. As usual the matrix $\mathcal{A}_d$ is called a *coarse solver*. It is easy to verify that $\mathcal{A}_d = \mathcal{C}_d \mathcal{A} \mathcal{C}_d^t$.

In this case, the preconditioner associated with the space decomposition (27) is defined as

$$\mathcal{B}^{-1} = \sum_{r=1}^{n_0} \mathcal{C}_r^t \mathcal{A}_r^{-1} \mathcal{C}_r + \mathcal{C}_d^t \mathcal{A}_d^{-1} \mathcal{C}_d, \tag{28}$$

which is the desired preconditioner for the original stiffness matrix $\mathcal{A}$. because, in general, both $\mathcal{A}_r$ and $\mathcal{A}_d$ have much smaller size than the original stiffness matrix $\mathcal{A}$; implementing the action of $\mathcal{B}^{-1}$ is much cheaper than implementing the action of $\mathcal{A}^{-1}$. Particularly, when the computed domain Ω is decomposed into nonoverlapping subdomains, the preconditioner can be simplified as follows:

$$\mathcal{B}^{-1} = \mathcal{D}^{-1} + \mathcal{C}_d^t \mathcal{A}_d^{-1} \mathcal{C}_d, \tag{29}$$

5. Parallel implementation of preconditioner $\mathcal{B}$ in JASMIN framework

In this section, we address some key issues in the parallel implementation of the domain decomposition preconditioner.

At first we would like to introduce the JASMIN framework (see [18]). JASMIN is developed by the institute of applied physics and computational mathematics in Beijing. The core of JASMIN framework is the data structure. The data structure adopted by JASMIN framework is as follows: mesh element–Patch(PatchData)–Patch Level–Patch Hierarchy, where the former is contained by the latter and PatchData is defined in the Patch which is core of the data structure.

The parallel implementation of action of preconditioner $\mathcal{B}^{-1}$ in JASMIN framework can be decomposed into two steps: setup and solving.

5.1. Setup of the preconditioner. The setup procedure is to implement LU factorization of submatrices $\mathcal{A}_r (r = 1, \cdots, n_0)$ and coarse solver $\mathcal{A}_d$.

• Factorizing submatrices

These submatrices can be factorized in a mutually independent way, so the procedure for LU factorization can be implemented in parallel. We distribute the subproblems among each group of processors, which guarantees parallel implementation. Particularly, each subproblem is assigned to a processor; namely, the number of subproblems is equal to the number of processors.

• Factorizing the coarse matrix

Before implementing LU factorization of the coarse matrix $\mathcal{A}_d$, a coarse patch level needs to be created. Then the coarse matrix is computed in the coarse patch level. A coarse mesh element (i.e., a subdomain) contains $(d/h)^n$ fine mesh elements, and each coarse mesh element corresponds to one fine patch. Notice that the fine patches are distributed on the processors. Thus the coarse matrix is computed in parallel. After the entries of the coarse matrix is obtained, all the entries are gathered to a processor associated with the coarse patch. Then the LU factorization for $\mathcal{A}_d$ can be implemented.

Finally, we point out that the cost of LU factorization for sub-matrices depends on the choice of LU factorization procedure. For convenience, here we only use the standard LU factorization procedure for sparse matrix. If a more advanced LU factorization procedure (for example, the MUMPS format) is used, then CPU times in our numerical experiments can be greatly reduced.

5.2. Solving the subproblems. This step is to solve subproblems and coarse problem, and realize the sum of subsolutions and coarse-solution. Solving subproblems and coarse problem instead of the original problem shall reduce the computational overhead and memory requirement. The solution over the whole domain can be obtained by the sum, which does not induce any additional error.

After subproblems are solved, realizing the sum of subsolutions is to add the subsolutions restricted on the overlapping fine elements. The overlapping fine elements can be divided into two groups: one is defined inside the patch, the other is defined in a patch boundary. For the former group, the math operation can realize the sum; For the latter group, the sum can be realized by the integrator component. In particular, if the preconditioner is nonoverlapping, the solution on the whole domain can be achieved by simply assembling the subsolutions.

Realizing the action of coarse solver can be decomposed into following steps: (i) by the transfer matrix $\mathcal{C}_d$, the coarse vector defined in the coarse patch level can be computed; (ii) by the vector communication algorithm, the coarse vector is copied

into the temporary coarse patch level; (iii) solving the coarse linear system; (iv) by the vector communication algorithm, the temporary coarse vector is copied into the coarse patch level; (v) by the embedding matrix $\mathcal{C}_d^t$, the corrected solution defined in the fine patch level can be computed.

Finally, by the math operation, we can realize the sum of subsolutions and coarse-solution.

6. Numerical experiments

In this section we report some numerical results to confirm the effectiveness of the preconditioner $\mathcal{B}$ defined in Section 4 for solving the linear system (21).

We consider two-dimensional and three-dimensional Helmholtz equation and three -dimensional Maxwell equations with large wave number. For the case of Helmholtz equation, as pointed out in Section 3 from [13], we choose the weighted parameters α and β in the variational problem (4) as $\alpha = \omega^2$ and $\beta = 1$ (when the analytic solution is smooth) or $\alpha = \omega$ and $\beta = \frac{1}{\omega}$ (if the analytic solution is singular). Besides, as pointed out in [16] (p.733), *the permittivity of the material in the computational domain for Maxwell's equations is in general complex valued* (i.e., the material is an absorbing medium). We apply the PWLS method with $\rho_1 = 1$ and $\rho_2 = -1$ to solve time-harmonic Maxwell equations (6) in three-dimensional absorbing media.

For the examples tested in this section, we adopt a uniform triangulation $\mathcal{T}_h$ for the domain Ω as follows. Ω is divided into small cubes (for three-dimensional case), rectangles (for two-dimensional case) with equal meshwidth, where h is the length of the longest edge of the elements. As described in section 3.1, we choose the same number p of basis functions for all elements Ω_k.

To determine the coarse solver $\mathcal{A}_d$ described in Section 5.2, we define the subdomain D_r ($r = 1, \ldots, n_0$) as follows: each subdomain (coarse element) D_r is a cube, which is just the union of several fine elements, and every subdomain D_r has the same size. Let d denote the length of the longest edge of the subdomains D_r. Then the orders of each submatrix $\mathcal{A}_r$ and the coarse matrix $\mathcal{A}_d$ respectively equal $\left(\frac{c_1 d}{h}\right)^n \times p$ and $\left(\frac{c_2}{d}\right)^n \times p$, where c_1 and c_2 are constants. In general, from the viewpoint of loading balance, we choose $d \approx \sqrt{h}$ such that the order $\left(\frac{c_1 d}{h}\right)^n \times p$ are almost same the order $\left(\frac{c_2}{d}\right)^n \times p$, which indicates that the computing time to factorize each submatrix $\mathcal{A}_r$ almost equals the computing time to factorize the coarse matrix $\mathcal{A}_d$. However, we consider different choices of d in the following tests to see how the iteration number and computing time depend on the different values of d/h. Note that the value d/h determines the number of elements contained in each subdomain D_r and so determines the number of coarse elements D_r for a fixed fine-mesh meshwidth h. More precisely, for a fixed h, when the value of d/h increases (so d increase), the number of coarse elements D_r decreases and so the order of the matrix $\mathcal{A}_d$ decreases.

To measure the accuracy of the numerical solution u_h and $\mathbf{E}_h$, we introduce the relative numerical error

$$\text{err.} = \frac{||u_{ex} - u_h||_{L^2(\Omega)}}{||u_{ex}||_{L^2(\Omega)}}\ , \text{ or err.} = \frac{||\mathbf{E}_{ex} - \mathbf{E}_h||_{L^2(\Omega)}}{||\mathbf{E}_{ex}||_{L^2(\Omega)}}$$

for the exact solution $u_{ex} \in L^2(\Omega)$, or $\mathbf{E}_{ex} \in (L^2(\Omega))^3$ and

$$\text{err.} = \sqrt{\frac{\sum_j^{Num} |u_{ex,j} - u_{h,j}|^2}{\sum_j^{Num} |u_{ex,j}|^2}}\ , \text{ or err.} = \sqrt{\frac{\sum_j^{Num} |\mathbf{E}_{ex,j} - \mathbf{E}_{h,j}|^2}{\sum_j^{Num} |\mathbf{E}_{ex,j}|^2}}$$

for the exact solution $u_{ex} \notin L^2(\Omega)$, or $\mathbf{E}_{ex} \notin (L^2(\Omega))^3$, where $u_{ex,j}, \mathbf{E}_{ex,j}$ and $u_{h,j}, \mathbf{E}_{h,j}$ are the exact solution and numerical approximation, respectively, of the problem at vertices referred to by the subscript j.

For the example discussed in the third section, we assume that $\mu = 1$. The stopping criterion in the iterative algorithms is that the relative L^2 norm ϵ of the residual of the iterative approximation satisfies $\epsilon < 1.0\text{e} - 6$ (we make the initial guess $X_0 = 0$ in the iteration), and the maximum number of iteration steps is $maxit = 1000$. Moreover, Ndof, Cores, Patch and $\mathrm{N_{iter}}$ represent the number of degree freedoms, the number of processors, the size of Patch and iteration numbers, respectively. $\mathrm{T_{setup}}$ represents the computing time of factorizing submatrices or factorizing the coarse matrix, $\mathrm{T_{solve}}$ and $\mathrm{T_{total}}$ represent the time of implementing the preconditioner and the overall run time of linear system, respectively.

6.1. An example of two-dimensional Helmholtz equation. The first model problem is the following Helmholtz equations for the acoustic pressure u and associated boundary conditions (see [13]):

$$\begin{aligned} &\Delta u + \omega^2 u = 0 \quad \text{in} \quad \Omega, \\ &\frac{\partial u}{\partial \mathbf{n}} + i\omega u = g \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{30}$$

where $\Omega = [0, 2] \times [0, 1]$, and $g = (\frac{\partial}{\partial \mathbf{n}} + i\omega)u_{ex}$.

The exact solution to the problem can be obtained in the closed form as

$$u_{ex}(x, y) = \cos(k\pi y)(A_1 e^{-i\omega_x x} + A_2 e^{i\omega_x x})$$

where $\omega_x = \sqrt{\omega^2 - (k\pi)^2}$, and coefficients A_1 and A_2 satisfy the equation

$$\begin{pmatrix} \omega_x & -\omega_x \\ (\omega - \omega_x)e^{-2i\omega_x} & (\omega + \omega_x)e^{2i\omega_x} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} -i \\ 0 \end{pmatrix} \tag{31}$$

The solution respectively represents propagating modes and evanescent modes when the mode number k is below the cut-off value $k \leqslant k_{\text{cut-off}} = \frac{\omega}{\pi}$ and up the cut-off value $k > k_{\text{cut-off}}$. To be simple, we only compute the approximate solution for the highest propagating mode with $k = \omega/\pi - 1$ in the following tests.

We first compare the efficiency of the non-overlapping domain decomposition preconditioner and the overlapping domain decomposition preconditioner with different overlaps. The numerical results are listed in Table 1 below, where "overlap" denote the numbers of elements describing the width of the square ring $D_r^{'} \backslash D_r$.

In the above table, "overlap=0" denote the non-overlapping domain decomposition preconditioner; "overlap=1" denote the domain decomposition preconditioner with one element overlap (small overlaps); "overlap=8" denote the domain decomposition preconditioner with half subdomain overlap; "overlap=16" denote the domain decomposition preconditioner with one subdomain overlap. The results indicate that the domain decomposition preconditioner with small overlaps is more effective than others when the wave number is large. Because of this, we only report numerical results for the domain decomposition preconditioner with small overlaps in the rest of this section except special emphasis.

TABLE 1. Iteration number and computing time for different overlaps.

	d/h	overlap	N_{iter}	T_{setup}	T_{solve}	T_{total}
$\omega = 160\pi, h = \frac{1}{256}$	16	0	601	2.84	13.6	16.7
		1	321	2.73	11.1	13.8
		8	134	29.8	22.4	54.3
		16	73	171	38.0	214

TABLE 2. Computing time for different patch.

Patch	$\mathrm{T_{setup}}$	$\mathrm{T_{solve}}$	$\mathrm{T_{total}}$
32×32	36.1	122	165
64×64	81.8	312	409
128×128	303	1.18e+3	1.53e+3

TABLE 3. Iteration number and computing time for different coarse elements.

ω	Ndof	Cores	d/h	$\mathrm{N_{iter}}$	$\mathrm{T_{setup}}$	$\mathrm{T_{solve}}$	$\mathrm{T_{total}}$
160π	$256^2 \times 12$	1024	8	380	0.51	6.30	6.42
		256	16	297	1.92	7.25	10.35
		64	32	229	16.3	10.8	27.9
320π	$512^2 \times 17$	1024	16	599	24.4	82.2	111
		256	32	420	57.1	105	170
		512	64	289	832	477	1.35e+3

We then numerically investigate how to determine the size of patch. We list the computing time for the case of $\omega = 640\pi, \mathrm{Ndof} = 1024^2 \times 17, d/h = 32, \mathrm{Cores} = 1024$ in Table 2.

The results listed in the above table indicate that by adjusting the patch size, the cache hit rate of parallel program can be improved, so the computing time is greatly saved. Here we propose that a patch size is equal to the coarse element size.

Table 3 gives the number of iterations and computing time for coarse elements of different size.

The results listed in above table indicate that when the size of coarse elements increases within a suitable region, the iteration number decreases. However, the Setup and Solve time of the preconditioner both increases. Especially, from viewpoint of load balancing in the domain decomposition method, we choose $d \approx \sqrt{h}$ such that the size of each subproblem is equal to the size of coarse problem. Therefore the computing time of factorizing submatrices is equal to the computing time of factorizing the coarse matrix.

Table 4 gives the parallel efficiency for the case of $\omega = 640\pi, \mathrm{Ndof} = 1024^2 \times 12, d/h = 32$. We point out that each patch only contain one coarse element.

It can be seen, from Table 4, that when the number of processors is less than 1024, the parallel efficiency of the preconditioner has better strong scalability. When the number of processors increases from 512 to 1024, the dramatic increase in the amount of communication hinders strong scalability.

Finally we show the efficiency of the proposed preconditioner for the case with large wave numbers. When the wave numbers increase, the scale of the discrete problem is increased in a suitable manner such that accepted relative L^2 errors of

TABLE 4. Parallel efficiency.

Cores	T_{setup}	T_{solve}	T_{total}	Parallel efficiency
32	589	2.23e+3	2.93e+3	
64	300	1.13e+3	1.48e+3	98.7%
128	154	582	764	97.0%
256	81.0	301	385	99.2%
512	44.5	168	221	87.1%
1024	36.1	122	165	67.0%

TABLE 5. Numerical results for large wave numbers.

ω	Ndof	Cores	d/h	N_{iter}	T_{setup}	T_{solve}	T_{total}	err.
320π	$512^2 \times 17$	1024	16	599	24.4	82.2	111	2.95e-4
640π	$1024^2 \times 17$	1024	32	731	80.1	247	342	5.63e-4
1280π	$2048^2 \times 17$	4096	32	1094	301	750	1.05e+3	1.06e-3
2560π	$2304^2 \times 19$	5184	32	1545	971	2.31e+3	3.28e+3	7.01e-3

the approximation can be kept. In numerical experiments, the patch size is equal to the coarse element size and the number of processors is equal to the number of patches. Table 5 below gives the iteration counts and the computing time spent on PCG iteration with the proposed preconditioner.

It is well known that, for the discretization method with the standard linear finite element, at least 10 points per wavelength is required to guarantee some accuracy of the approximate solution. It can be seen, from Table 5, that about 3 points per wavelength is enough to guarantee satisfactory accuracy of the plane wave approximations since the approximations generated by the PWLS method have high accuracy. This advantage thanks to the $hp-$ error estimate (23) given in Theorem 3.3.

It is well known that, when the linear finite element method is used to discretization of Helmholtz equation with large wave number, the mesh size h needs to be chosen carefully such that $h = \theta \cdot \frac{1}{\omega}$, where θ satisfies $\theta \to 0^+$ when $\omega \to +\infty$ (because of the pollution of large wave number). This means that the mesh size h should be less than $\frac{1}{2}h$ when ω is doubled, so the time costed in the solution of the discrete system increases as four times (for a two-dimensional problem) at least even if the solver is the optimal. The results listed in the above table indicate that the iteration counts and the computing time spent on the implementation of the new solver increase slowly when the wave numbers increase, and so the proposed method is very effective for the case with large wave number.

Here only the simplest format is used in LU factorization of the matrices $\mathcal{A}_r$ and $\mathcal{A}_d$. As pointed out at the end of Subsection 5.1, if a more advanced format is used in LU factorization of these matrices, the computing times can be greatly reduced.

6.2. An example of three-dimensional Helmholtz equation. The following test problem consists of a point source and associated boundary conditions for homogeneous Helmholtz equations (see [15]):

$$
\begin{aligned}
& u(r, r_0) = \frac{1}{4\pi} \frac{e^{i\omega|r-r_0|}}{|r-r_0|} \text{ in } \Omega, \\
& \frac{\partial u}{\partial \mathbf{n}} + i\omega u = g \quad \text{over} \quad \partial\Omega,
\end{aligned} \tag{32}
$$

TABLE 6. Iteration number and computing time for different overlaps.

	d/h	overlap	N_{iter}	T_{setup}	T_{solve}	T_{total}
$\omega = 32\pi, h = \frac{1}{32}$	4	0	189	460	90.6	552
		1	105	309	48.7	374
		2	77	905	131	1.04e+3
		4	56	3.52e+3	491	4.01e+3

TABLE 7. Iteration number and computing time for different coarse elements.

ω	Ndof	Cores	d/h	$\mathrm{N_{iter}}$	$\mathrm{T_{setup}}$	$\mathrm{T_{solve}}$	$\mathrm{T_{total}}$
16π	$16^3 \times 25$	512	2	59	564	42.3	609
		64	4	72	132	17.5	161
		8	8	50	2.57e+3	63.8	2.69e+3
32π	$32^3 \times 25$	512	4	105	309	48.7	374
		64	8	90	1.37e+3	211	1.58e+3
		128	16	69	2.97e+3	411	3.39e+3

TABLE 8. Parallel efficiency.

Cores	$\mathrm{T_{setup}}$	$\mathrm{T_{solve}}$	$\mathrm{T_{total}}$	Parallel efficiency
64	1.54e+3	302	1.92e+3	
128	914	167	1.13e+3	85.0%
256	511	88.4	627	76.7%
512	309	48.7	374	64.3%

in a cubic computational domain $\Omega = [0,1] \times [0,1] \times [0,1]$ centred at the origin. The location of the source is off-centred at $r_0 = (1,1,1)$ and $r = (x,y,z)$ is an observation point. Note that the analytic solution of the Helmholtz equation with such boundary condition has a singularity at $r = r_0$.

As in the last subsection, we first compare the efficiency of the preconditioners with different overlaps in Table 6 below.

The results listed in above table indicate that the domain decomposition preconditioner with small overlaps is more effective than others.

Table 7 gives the number of iterations and computing time for coarse elements of different size.

Similarly to the first example, from viewpoint of load balancing in the domain decomposition method, we choose $d \approx \sqrt{h}$ such that the size of each subproblem equals to the size of coarse problem.

Table 8 gives the parallel efficiency for the case of $\omega = 32\pi, \mathrm{Ndof} = 32^3 \times 25, d/h = 4$. We also point out that each patch only contain one coarse element.

It can be seen that, from Table 8, when the number of processors increases, the dramatic increase in the amount of communication hinders strong scalability.

Table 9 gives the relative L^2 error of approximations for large wave numbers. Similarly to the first example, we choose that each patch size equals to the size of coarse element and the number of processors equals to the number of patches.

The results listed in Table 9 indicate that the preconditioner is effective for the case of large wave numbers. Notice that the analytic solution u is singular at the vertex of Ω, the accuracy of the approximate solutions is low.

TABLE 9. Numerical results for large wave numbers.

ω	Ndof	Cores	d/h	N_{iter}	T_{setup}	T_{solve}	T_{total}	err.
16π	$16^3 \times 49$	64	4	85	264	23.8	289	9.98e-2
32π	$32^3 \times 49$	512	4	157	649	105	756	1.35e-1
64π	$64^3 \times 49$	512	8	279	2.61e+3	432	3.04e+3	2.39e-1

Fig.1 shows the solutions for the point source at $\omega = 32\pi$, where the solutions are computed by (32) and the PWLS method, respectively. The top row shows the real parts of the acoustic fields generated by the two methods, and the bottom row shows the full amplitude of these acoustic fields.

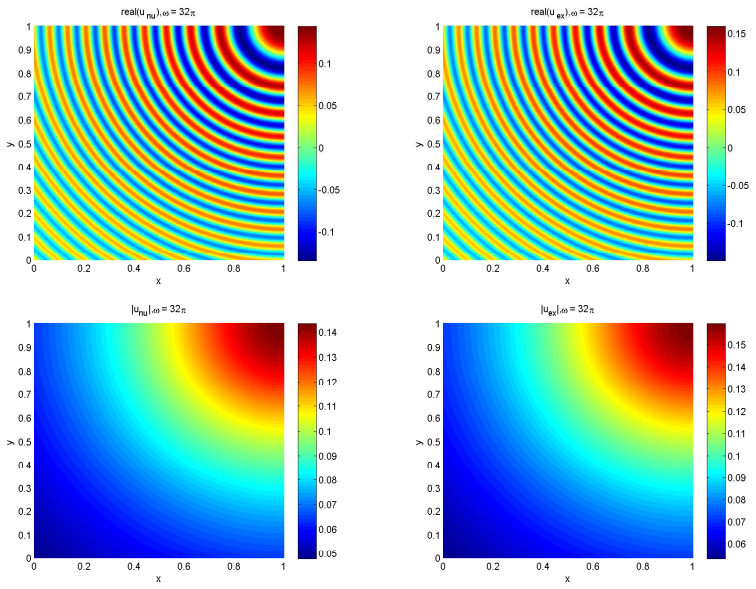

FIGURE 1. Comparision of solutions for the point source at $\omega = 32\pi$.

Fig. 1 shows that the approximate solution computed by the PWLS method is almost indistinguishable from the exact solution.

6.3. An example of three-dimensional Maxwell's equations. We compute the electric field due to an electric dipole source at the point $\mathbf{x}_0 = (0.6, 0.6, 0.6)$. The dipole point source can be defined as the solution of a homogeneous Maxwell system (6). The exact solution of the problems is

$$\mathbf{E}_{\text{ex}} = -\mathrm{i}\omega I\phi(\mathbf{x}, \mathbf{x}_0)\mathbf{a} + \frac{I}{\mathbf{i}\omega\varepsilon}\nabla(\nabla\phi \cdot \mathbf{a}), \tag{33}$$

where

$$\phi(\mathbf{x}, \mathbf{x}_0) = \frac{\exp(\mathrm{i}\omega\sqrt{\varepsilon}|\mathbf{x} - \mathbf{x}_0|)}{4\pi|\mathbf{x} - \mathbf{x}_0|}$$

and $\Omega = [-0.5, 0.5]^3$. To keep the exact solution smooth in Ω, we move the singularity $\mathbf{x}_0$ from $(0.2, 0.2, 0.2)$ (see Ref. [16]) in the computational domain to $(0.6, 0.6, 0.6)$ outside the region. We investigate homogeneous media and assume that $\varepsilon = 1 + i$.

TABLE 10. Iteration number and computing time for different overlaps.

	d/h	overlap	N_{iter}	T_{setup}	T_{solve}	T_{total}
$\omega = 16\pi, h = \frac{1}{16}$	4	0	27	273	9.75	283
		1	23	233	6.75	246
		2	21	705	24.1	731
		4	18	2.82e+3	115	2.94e+3

TABLE 11. Iteration number and computing time for different coarse elements.

ω	Ndof	Cores	d/h	$\mathrm{N_{iter}}$	$\mathrm{T_{setup}}$	$\mathrm{T_{solve}}$	$\mathrm{T_{total}}$
16π	$16^3 \times 50$	512	2	28	973	20.8	996
		64	4	23	233	6.75	246
		8	8	18	4.74e+3	24.5	4.84e+3

TABLE 12. Parallel efficiency.

Cores	$\mathrm{T_{setup}}$	$\mathrm{T_{solve}}$	$\mathrm{T_{total}}$	Parallel efficiency
64	2.43e+3	49.0	2.49e+3	
128	1.40e+3	27.1	1.43e+3	87.3%
256	814	16.4	834	74.6%
512	467	9.31	478	65.1%

TABLE 13. Numerical results for large wave numbers.

ω	Ndof	Cores	d/h	$\mathrm{N_{iter}}$	$\mathrm{T_{setup}}$	$\mathrm{T_{solve}}$	$\mathrm{T_{total}}$	err.
8π	$16^3 \times 50$	64	4	24	129	6.59	145	2.06e-2
16π	$16^3 \times 98$	64	4	22	491	14.6	505	1.14e-2
32π	$32^3 \times 98$	512	4	25	1.41e+3	43.8	1.45e+3	1.05e-2
64π	$64^3 \times 50$	512	8	25	4.21e+3	167	4.38e+3	2.26e-2

We firt list the iteration number and computing time for the preconditioners with different overlaps in Table 10.

The results listed in above table indicate that the domain decomposition preconditioner with only one overlap is more effective than others.

Table 11 gives the number of iterations and computing time for coarse elements of different size.

Similarly to above examples, from viewpoint of load balancing we choose $d \approx \sqrt{h}$.

Table 12 gives the parallel efficiency for the case of $\omega = 32\pi, \mathrm{Ndof} = 32^3 \times 50, d/h = 4$.

It can be seen that, from Table 12, when the number of processors increases, the dramatic increase in the amount of communication hinders strong scalability.

Table 13 gives the relative L^2 error of approximations for large wave numbers. Here we choose that each patch size equals to the coarse element size and the number of processors equals to the number of patches.

The results listed in the above table indicate that the iteration counts are almost stable and the computing time increases slowly when the wave numbers increase.

References

[1] M. Amara, R. Djellouli and C. Farhat, Convergence analysis of a discontinuous Galerkin method with plane waves and Lagrange multipliers for the solution of Helmholtz problems, SIAM J. Numer. Anal., 47(2009), pp. 1038-1066.

[2] O. Cessenat, Application d'une nouvelle formulation variationnelle aux équations d'ondes harmoniques, Problèmes de Helmholtz 2D et de Maxwell 3D, Ph.d. thesis, Université Paris IX Dauphine, 1996.

[3] O. Cessenat and B. Despres, Application of an ultra weak variational formulation of elliptic PDEs to the two-dimensional helmholtz problem, SIAM J. Numer. Anal., 35(1998), pp. 255-299.

[4] O. Cessenat and B. Despres, Using plane waves as basis functions for solving time harmonic equations with the ultra weak variational formulation, J. Comput. Acous., 11(2003), pp. 227-238.

[5] X. Cai, O. Widlund. Domain decomposition algorithms for indefinite elliptic problems. SIAM Journal on Scientific and Statistical Computing, 13(1992), No.1, pp.243-258.

[6] Z. Chen, X. Xiang. A Source Transfer Domain Decomposition Method for Helmholtz Equations in Unbounded Domain. SIAM Journal on Numerical Analysis, 51(2013), No.4, pp.2331-2356.

[7] B. Engquist, L. Ying. Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers[J]. Multiscale Modeling Simulation, 9(2011), No.2, pp.686-710.

[8] C. Farhat, A. Macedo, M. Lesoinne. A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. Numerische Mathematik, 85(2000), No.2, pp.283-308.

[9] C. Farhat, R. Tezaur and P. Weidemann, Higher-order extensions of a discontinuous Galerkin method for mid-frequency Helmholtz problems, Internat. J. Numer. Methods Engrg., 61(2004), pp. 1938-1956.

[10] C. Gittelson, R. Hiptmair and I. Perugia, Plane wave discontinuous Galerkin methods: Analysis of the h-version, ESAIM Math. Model. Numer. Anal., 43(2009), pp. 297-331.

[11] R. Hiptmair, A. Moiola, and I. Perugia, Plane wave discontinuous Galerkin methods for the 2D Helmholtz equation: analysis of the p-version. SIAM J. Numer. Anal., 49(2011), No.1, pp. 264-284.

[12] R. Hiptmair, A. Moiola, and I. Perugia, Error analysis of Trefftz-discontinuous Galerkin methods for the time-harmonic Maxwell equations, Math. Comp., 82(2013), pp. 247-268.

[13] Q. Hu and L. Yuan, A weighted variational formulation based on plane wave basis for discretization of Helmholtz equations, Int. J. Numer. Anal. Model., 11(2014), pp. 587-607.

[14] Q. Hu and L. Yuan, A Plane Wave Least-Squares Method for Time-Harmonic Maxwell's Equations in Absorbing Media, SIAM J. Sci. Comput., 36(2014), pp. A1911-A1936.

[15] T. Huttunen, J. Kaipio and P. Monk. The perfectly matched layer for the ultra weak variational formulation of the 3D Helmholtz equation, Int. J. Numer. Meth. Engng., 61(2004), pp. 1072-1092.

[16] T. Huttunen, M. Malinen, P. Monk, Solving Maxwell's equations using the ultra weak variational formulation, J. Comput. Phys., 223 (2007), pp. 731-758.

[17] C. Lasser, A. Toselli. An overlapping domain decomposition preconditioner for a class of discontinuous Galerkin approximations of advection-diffusion problems. Mathematics of Computation, 72(2003), No.243, pp.1215-1238.

[18] Z. Mo and A. Zhang (eds.): User's guide for JASMIN, Technical Report No. T09-JMJL-01(2009). http://www.iapcm.ac.cn/jasmin

[19] A. Moiola, R. Hiptmair, and I. Perugia, Plane wave approximation of homogeneous Helmholtz solutions, Z. Angew. Math. Phys., 62(2011), pp. 809-837.

[20] P. Monk and D. Wang, A least-squares method for the helmholtz equation, Comput. Methods Appl. Mech. Engrg., 175(1999), pp. 121-136.

[21] H. Riou, P. Ladevèze, B. Sourcis, The multiscale VTCR approach applied to acoustics problems, J. Comput. Acous., 16(2008), pp. 487-505.

[22] H. Riou, P. Ladevèze, B. Sourcis, B. Faverjon and L. Kovalevsky, An adaptive numerical strategy for the media-frequency analysis of Helmholtz's problem, J. Comput. Acous., 20(2012).

[23] N. Sloane, Tables of spherical codes, http://neilsloane.com/packings/dim3(2014).

[24] R. Tezaur, C. Farhat, Three-dimensional directional discontinuous Galerkin elements with plane waves and Lagrange multipliers for the solution of mid-frequency Helmholtz problems, Int. J. Numer. Meth. Engng., 66(2006), pp. 796-815.

[25] A. Toselli, O. Widlund. Domain decomposition methods: algorithms and theory. Berlin: Springer, 2005.
[26] Long Yuan and Qiya Hu, A plane wave discontinuous Petrov-Galerkin method for Helmholtz equation and time-harmonic Maxwell equations with complex wave numbers, J. Numerical Methods and Computer Applications, 36(1015), No. 4, 185-196

College of Mathematics and Systems Science, Shan Dong University of Science and Technology, 579 Qian Wan Gang Road, Qingdao 266590, China
E-mail: yuanlong@lsec.cc.ac.cn

LSEC, Institute of Computational Mathematics and Scientic/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
E-mail: hqy@lsec.cc.ac.cn

Institute of Applied Physics and Computational Mathematics, Beijing 100088, China
E-mail: an_hengbin@iapcm.ac.cn