

**NOVEL MULTILEVEL PRECONDITIONERS FOR THE SYSTEMS ARISING
FROM PLANE WAVE DISCRETIZATION OF HELMHOLTZ EQUATIONS
WITH LARGE WAVE NUMBERS**

QIYA HU AND XUAN LI

ABSTRACT. In this paper we are concerned with fast algorithms for the systems arising from the plane wave discretizations for two-dimensional Helmholtz equations with large wave numbers. We consider the plane wave weighted least squares (PWLS) method and the plane wave discontinuous Galerkin (PWDG) method. The main goal of this paper is to construct multilevel parallel preconditioners for solving the resulting Helmholtz systems. To this end, we first build a multilevel overlapping space decomposition for the plane wave discretization space based on a multilevel overlapping domain decomposition method. Then, corresponding to the space decomposition, we construct an additive multilevel preconditioner for the underlying Helmholtz systems. Further, we design both additive and multiplicative multilevel preconditioners with smoothers, which are different from the standard multigrid preconditioners. We apply the proposed multilevel preconditioners with a *constant* coarsest mesh size to solve two dimensional Helmholtz systems generated by PWLS method or PWDG method, and we find that the new preconditioners possess nearly stable convergence, i.e., the iteration counts of the preconditioned iterative methods (PCG or PGMRES) with the preconditioners increase very slowly when the wave number increases (and the fine mesh size decreases).

Keywords: Helmholtz equation, large wave numbers, plane wave methods, multilevel overlapping domain decomposition, multilevel overlapping preconditioner, smoothers

AMS subject classifications. 65N22, 65N55, 65N06, 65F10

1. INTRODUCTION

The plane wave method, which falls into the class of Trefftz methods [38], differs from the traditional finite element method and the boundary element method 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. Examples of this approach include the Ultra Weak Variational Formulation (UWVF) (see [9, 15]), the weighted plane wave least-squares (PWLS) method (see [23, 35]), the plane wave discontinuous Galerkin methods (PWDG) (see [17, 20]), the plane wave Lagrangian multiplier (PWLM) method [14, 37] and the Variational Theory of Complex Rays (VTCR) introduced in [29, 28, 36]. This kind of method can generate higher accuracy approximations than the other methods for Helmholtz equations with large wave numbers. The plane wave discretization methods have been extended to discretization of Maxwell's equations recently (see [21, 24, 26]). The PWLS method has an advantage over the other plane wave methods: the stiffness matrix associated with the PWLS method is Hermitian positive definite, so the resulting system can be solved by the PCG method. Like the other discretization methods, the Helmholtz systems arising from the plane wave

LSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China. (hxy@lsec.cc.ac.cn and lixuan@lsec.cc.ac.cn). This research was supported by the Natural Science Foundation of China G11571352.

discretization are also highly ill-conditioned when the wave number is large. Comparing with many works on the plane wave discretizations, there are only a few articles (refer to [14, 23, 37]) to study fast solver for the resulting Helmholtz systems.

It is well known that multilevel methods are powerful algorithms for solving the systems generated by finite element discretization of elliptic-type partial differential equations (see, for example, [1, 3, 4, 6, 7, 19]). However, the standard multilevel methods (and domain decomposition methods) are ineffective for Helmholtz equations (and time-harmonic Maxwell's equations) with large wave numbers, unless the sizes of coarse meshes are chosen as $O(1/\omega)$ (see, for example, [2, 8, 11, 13, 27, 16, 18, 30, 33, 39]), where ω denotes the fixed wave number. It is clear that the restriction on the coarse mesh sizes is limiting in applications. How to construct an effective parallel preconditioner for Helmholtz equations (and time-harmonic Maxwell's equations) with large wave numbers seems an open problem. The wave-ray multigrid method for Helmholtz equations was proposed in [5, 32] (a further development of this method was made in [31]), in which the approximations of oscillatory error components were transformed into the approximations of smooth ray envelope functions by using the exponential interpolations. The wave-ray multigrid method can improve the performance of the standard multigrid methods for Helmholtz equations with large wave numbers. Recently, a kind of successive preconditioner based on a decomposition of the domain into strips was proposed in [10, 12] to solve Helmholtz equations with large wave numbers. The preconditioners can be viewed as physically-based approximations of direct solvers. It has been shown that such kind of preconditioner possesses the optimal convergence independent of the mesh sizes [10], which is a very important result in the solution method for Helmholtz equations with large wave numbers.

In the present paper, we consider the PWLS method and the PWDG method for the discretization of Helmholtz equations in two dimensions, and explore a new way to construct multilevel preconditioners for the resulting Helmholtz systems. At first we design a multilevel overlapping domain decomposition method to build a multilevel space decomposition for the plane wave discretization space. Then, based on the space decomposition, we construct an additive multilevel overlapping preconditioner for the underlying Helmholtz systems. Finally, we replace the solvers in the previous preconditioner by block Jacobi-type smoothers to get cheaper (both additive and multiplicative) multilevel overlapping preconditioners. The multilevel overlapping preconditioners with smoothers are different from the standard multigrid preconditioners, since the space decomposition defining such new preconditioners has different overlapping structure from the one corresponding to the standard multigrid preconditioners. We apply the proposed preconditioners to solve Helmholtz systems generated by PWLS method or PWDG method. Numerical results indicate that the new preconditioners possess nearly stable convergence, i.e., the iteration counts of the corresponding iterative methods (PCG or PGMRES) increase very slowly when the wave number increases (and the mesh size decreases), without the limiting condition mentioned in the last paragraph. In particular, the multilevel overlapping preconditioners with smoothers possess almost optimal convergence.

The paper is organized as follows: In Section 2, we recall the PWLS method and the PWDG method for Helmholtz equations. In section 3, we design a multilevel space decomposition of the solution space and describe the corresponding additive multilevel preconditioner. An additive multilevel overlapping preconditioner with smoothers is introduced in Section 4. In Section 5, we define several multiplicative variants of the additive multilevel overlapping preconditioner with smoothers. In Section 6, we apply the proposed preconditioners to solve several Helmholtz systems and report some numerical results.

2. PLANE WAVE METHODS FOR HELMHOLTZ EQUATIONS

For convenience, we only consider the two-dimensional case in this paper. In this section, we briefly review the plane wave methods for Helmholtz equations. At first the original problem to be solved is defined. Then the variational formulations are given out in detail.

2.1. The reference problem. Firstly, we present the mathematical model of Helmholtz equations. Let Ω be a bounded and connected Lipschitz domain in two dimensions. We consider Helmholtz equations with Robin boundary conditions.

$$\begin{cases} -\Delta u - \omega^2 u = 0 & \text{in } \Omega, \\ (\partial_{\mathbf{n}} + i\omega)u = g & \text{on } \gamma = \partial\Omega, \end{cases} \quad (2.1)$$

where $\partial_{\mathbf{n}}$ and ω denote the outer normal derivative and the angular frequency.

Let Ω be divided into a partition as follows:

$$\overline{\Omega} = \bigcup_{k=1}^N \overline{E}_k, \quad E_k \cap E_j = \emptyset \quad \text{for } k \neq j.$$

We assume that the subdomains E_1, E_2, \dots, E_N are geometrical conforming, i.e., the intersection of any two adjoining subdomains is just the common vertex or the common edge of them. Here, we do not require that the intersection of two adjoining elements is a straight line segment. In practice, the partition is a mesh of domain, and E_1, \dots, E_N are the elements. As usual, we assume that $\{E_k\}$ is quasi-uniform and regular. Let \mathcal{T}_h denote the set of the elements E_1, \dots, E_N , where h is the size of the elements. Define

$$\gamma_{kj} = \partial E_k \cap \partial E_j \quad (\text{when } E_k \text{ and } E_j \text{ are adjoining})$$

and

$$\gamma_k = \partial E_k \cap \partial\Omega \quad (\text{if } E_k \text{ closes } \partial\Omega).$$

Let $V(E_k)$ denote the space of the functions which verify Helmholtz's homogeneous equation (2.1) on the element E_k :

$$V(E_k) = \{v_k \in H^1(E_k); \Delta v_k + \omega^2 v_k = 0\}. \quad (2.2)$$

Define

$$V(\mathcal{T}_h) = \prod_{k=1}^N V(E_k),$$

with the natural scalar product

$$(u, v)_V = \sum_{k=1}^N \int_{E_k} u_k \cdot \bar{v}_k \, d\mathbf{x}, \quad \forall u, v \in V(\mathcal{T}_h).$$

2.2. The PWLS method. In this subsection, we review the PWLS method introduced in [35] and [23].

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

$$\begin{cases} -\Delta u_k - \omega^2 u_k = 0 & \text{in } E_k, \\ (\partial_{\mathbf{n}} + i\omega)u = g & \text{on } \gamma_k \text{ (if } \gamma_k \neq \emptyset), \end{cases} \quad (2.3)$$

and

$$\begin{cases} u_k - u_j = 0 & \text{over } \gamma_{kj}, \\ \partial_{\mathbf{n}_k} u_k + \partial_{\mathbf{n}_j} u_j = 0 & \text{over } \gamma_{kj} \end{cases} \quad (k \neq j; k, j = 1, 2, \dots, N). \quad (2.4)$$

Let α and β be two given positive real numbers to be specified later. Corresponding to the boundary condition in (2.3) and the interface continuity condition (2.4), we define the functional

$$J(v) = \sum_{k=1}^N \int_{\gamma_k} |(\partial_{\mathbf{n}} + i\omega)v_k - g|^2 ds + \sum_{j \neq k} \left(\alpha \int_{\gamma_{kj}} |v_k - v_j|^2 ds + \beta \int_{\gamma_{kj}} |\partial_{\mathbf{n}_k} v_k + \partial_{\mathbf{n}_j} v_j|^2 ds \right), \quad v \in V(\mathcal{T}_h). \quad (2.5)$$

It is clear that $J(v) \geq 0$. Consider the minimization problem: find $u \in V(\mathcal{T}_h)$ such that

$$J(u) = \min_{v \in V(\mathcal{T}_h)} J(v) \quad (2.6)$$

If u is the solution of the problem (2.1), i.e., $u \in V(\mathcal{T}_h)$ satisfies the boundary condition in (2.3) and the interface continuity condition (2.4), then we have $J(u) = 0$, which implies that u is also the solution of the minimization problem (2.6).

Define the sesquilinear form $a(\cdot, \cdot)$ by

$$a(u, v) = \sum_{k=1}^N \int_{\gamma_k} ((\partial_{\mathbf{n}} + i\omega)u_k) \cdot \overline{(\partial_{\mathbf{n}} + i\omega)v_k} ds + \sum_{j \neq k} \left(\alpha \int_{\gamma_{kj}} (u_k - u_j) \cdot \overline{(v_k - v_j)} ds + \beta \int_{\gamma_{kj}} (\partial_{\mathbf{n}_k} u_k + \partial_{\mathbf{n}_j} u_j) \cdot \overline{(\partial_{\mathbf{n}_k} v_k + \partial_{\mathbf{n}_j} v_j)} ds \right), \quad \forall v \in V(\mathcal{T}_h), \quad (2.7)$$

and define the functional $\mathcal{L}(\cdot)$ by

$$\mathcal{L}(v) = \sum_{k=1}^N \int_{\gamma_k} g \cdot \overline{(\partial_{\mathbf{n}} + i\omega)v_k} ds \quad \forall v \in V(\mathcal{T}_h). \quad (2.8)$$

The variational problem associated with the minimization problem (2.6) can be expressed as:

$$\begin{cases} \text{Find } u \in V(\mathcal{T}_h), \text{ s.t.} \\ a(u, v) = \mathcal{L}(v), \quad \forall v \in V(\mathcal{T}_h). \end{cases} \quad (2.9)$$

The reference problem (2.3) and (2.4) is equivalent to the new variational problem (2.9) (see [23] Theorem 3.1). In applications, we usually choose the two parameters in (2.5) as $\alpha = \omega^2$ and $\beta = 1$.

2.3. The PWDG method. In this subsection, we review the PWDG method introduced in [20].

Let u and σ be a piecewise smooth function and vector field on \mathcal{T}_h respectively. On γ_{kj} , we define

$$\begin{aligned} \text{the averages:} & \quad \{u\} = \frac{1}{2}(u_k + u_j), \quad \{\sigma\} = \frac{1}{2}(\sigma_k + \sigma_j), \\ \text{the jumps:} & \quad [u] = u_k \mathbf{n}_k + u_j \mathbf{n}_j, \quad [\sigma] = \sigma \cdot \mathbf{n}_k + \sigma \cdot \mathbf{n}_j. \end{aligned}$$

Set

$$\mathcal{F}_h^I = \bigcup_{k \neq j} \gamma_{kj} \quad \text{and} \quad \mathcal{F}_h^B = \bigcup_{k=1}^N \gamma_k.$$

With these definitions, we can write the PWDG method as follows:

$$\begin{cases} \text{Find } u \in V(\mathcal{T}_h), \text{ s.t.} \\ a(u, v) = \mathcal{L}(v), \forall v \in V(\mathcal{T}_h), \end{cases} \quad (2.10)$$

where (see [20])

$$\begin{aligned} a(u, v) = & \int_{\mathcal{F}_h^I} \left(\{u\}[\overline{\nabla}v] - \frac{\beta}{i\omega} [\nabla u][\overline{\nabla}v] - \{\nabla u\}[\overline{v}] + \alpha \cdot i\omega[u][\overline{v}] \right) ds \\ & + \int_{\mathcal{F}_h^B} \left((1-\delta)u\overline{\nabla}v \cdot \mathbf{n} - \frac{\delta}{i\omega} \nabla u \cdot \mathbf{n} \overline{\nabla}v \cdot \mathbf{n} - \delta \nabla u \cdot \mathbf{n} \overline{v} + (1-\delta)i\omega u \overline{v} \right) ds, \end{aligned} \quad (2.11)$$

and

$$\mathcal{L}(v) = \int_{\mathcal{F}_h^B} \left(-\frac{\delta}{i\omega} g \overline{\nabla}v \cdot \mathbf{n} + (1-\delta)g \overline{v} \right) ds. \quad (2.12)$$

Here α, β and δ are given positive parameters. The simplest choice of the parameters in the above two expressions is $\alpha = \beta = \delta = \frac{1}{2}$.

2.4. Discretization of the variational formulations. Before building discrete variational problems, we need to approximate the space $V(\mathcal{T}_h)$ by a suitable finite dimensional subspace, which is spanned by some plane wave basis functions, i.e., solutions of homogeneous Helmholtz equation without boundary condition.

For convenience, we assume that the number of plane wave basis functions equals a same positive integer p for every elements Ω_k . Let y_l be the wave shape functions, which satisfy

$$\begin{cases} y_l(\mathbf{x}) = e^{i\omega(\boldsymbol{\alpha}_l \cdot \mathbf{x})}, \mathbf{x} \in \Omega, \\ \boldsymbol{\alpha}_l \cdot \boldsymbol{\alpha}_l = 1, \\ l \neq s \rightarrow \boldsymbol{\alpha}_l \neq \boldsymbol{\alpha}_s, \end{cases} \quad (2.13)$$

where $\boldsymbol{\alpha}_l$ ($l = 1, \dots, p$) are unit wave propagation directions to be specified later. The plane wave basis functions can be defined as

$$\phi_l^{(k)}(\mathbf{x}) = \begin{cases} y_l(\mathbf{x}), & \mathbf{x} \in E_k, \\ 0, & \mathbf{x} \notin E_k \end{cases} \quad (k = 1, \dots, N; l = 1, \dots, p). \quad (2.14)$$

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

$$V_p(\mathcal{T}_h) = \text{span} \left\{ \phi_l^{(k)} : k = 1, \dots, N; l = 1, \dots, p \right\}. \quad (2.15)$$

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

$$\boldsymbol{\alpha}_l = \begin{pmatrix} \cos(2\pi(l-1)/p) \\ \sin(2\pi(l-1)/p) \end{pmatrix} \quad (l = 1, \dots, p).$$

Let $V_p(\mathcal{T}_h)$ be the plane wave space defined above. Then the discrete variational problems associated with (2.10) and (2.9) can be described as follows:

$$\begin{cases} \text{Find } u_h \in V_p(\mathcal{T}_h), \text{ s.t.} \\ a(u_h, v_h) = \mathcal{L}(v_h), \forall v_h \in V_p(\mathcal{T}_h). \end{cases} \quad (2.16)$$

Let $A : V_p(\mathcal{T}_h) \rightarrow V_p(\mathcal{T}_h)$ be the discrete operator defined by the sesquilinear form $a(\cdot, \cdot)$. The discrete variational problem (2.16) can be written in the operator form

$$Au_h = f_h, \quad u_h \in V_p(\mathcal{T}_h). \quad (2.17)$$

Let \mathcal{A} be the stiffness matrix generated by the sesquilinear form $a(\cdot, \cdot)$ on the space $V_p(\mathcal{T}_h)$, and let b denote the vector associated with $\mathcal{L}(v_h)$. Namely, the entries of the matrix

\mathcal{A} are computed by $a_{k,j}^{l,m} = a(\phi_m^{(j)}, \phi_l^{(k)})$; and the complements of the vector b are defined as $b_{k,l} = \mathcal{L}(\phi_l^{(k)})$. The discretized problem (2.16) leads to the algebraic system below:

$$\mathcal{A}X = b, \quad (2.18)$$

where $X = (x_{11}, x_{12}, \dots, x_{1p}, x_{21}, \dots, x_{2p}, \dots, x_{N1}, \dots, x_{Np})^t \in \mathbb{C}^{pN}$ is the unknown vector.

In general the system (2.16) is solved by an iterative method, for example, the preconditioned GMRES method or the PCG method. In this paper, we solve the system arising from the PWDG method by preconditioned GMRES method, and solve the system arising from the PWLS method by PCG method since the system of the PWLS method is Hermitian positive definite. Notice that implementation of an iterative step in PCG method is cheaper than that in the preconditioned GMRES method. We need to construct an efficient preconditioner \mathcal{B} for the matrix \mathcal{A} , and solve the equivalent system

$$\mathcal{B}^{-1}\mathcal{A}X = \mathcal{B}^{-1}b. \quad (2.19)$$

The main goal of this paper is to construct efficient multilevel preconditioners \mathcal{B} , especially multilevel preconditioners with overlapping smoothers. In order to make the ideas easily understood, we first construct a basic preconditioner directly from multilevel overlapping domain decompositions, and then we define multilevel preconditioners with overlapping smoothers based on the basic preconditioner. For convenience, we shall describe the preconditioners in operator forms, instead of matrix forms.

3. A PRECONDITIONER BASED ON MULTILEVEL OVERLAPPING DOMAIN DECOMPOSITION

In this section, we construct an additive multilevel preconditioner B for the operator A based on overlapping domain decompositions.

3.1. A multilevel overlapping space decomposition. Let N_0 be a fixed positive integer, which is independent of ω , h and p . For simplicity of exposition, we use D to denote a generic domain that is the union of some elements in \mathcal{T}_h , where D can be the domain Ω itself or a subdomain of Ω .

Let D be decomposed into the union of non-overlapping subdomains D_1, D_2, \dots, D_{N_0} such that: (1) each subdomain D_r is just the union of several elements in \mathcal{T}_h ; (2) the subdomains D_1, D_2, \dots, D_{N_0} are quasi-uniform, regular and geometrical conforming (refer to Subsection 2.1). Here, we do not require that the intersection of two adjoining subdomains is a straight line segment. Then D_1, \dots, D_{N_0} can be viewed as coarse elements of D and they constitute a (coarse) finite element partition \mathcal{T}_d^D of D , where d denotes the size of these elements.

Based on the partition \mathcal{T}_d^D , we can define an overlapping domain decomposition of D as usual. For a constant $\theta_0 \in [\frac{1}{2}, 1]$, we enlarge each coarse element D_r by the thickness $\theta_0 d$, and generate a larger domain \tilde{D}_r satisfying: (1) $D_r \subset \tilde{D}_r \subset D$; (2) \tilde{D}_r is just the union of some (fine) elements in \mathcal{T}_h ; (3) the distance between the internal boundaries $\partial\tilde{D}_r \setminus \partial D$ and $\partial D_r \setminus \partial D$ is about $\theta_0 d$. Then

$$\bar{D} = \bigcup_{r=1}^{N_0} \tilde{D}_r$$

constitutes an overlapping domain decomposition of D with ‘‘large overlap’’. For convenience, we call the parameter θ_0 as ‘‘overlapping degree’’. When $\theta_0 = 1$ (rep. $\theta_0 = \frac{1}{2}$), each subdomain \tilde{D}_r is the union of D_r itself and all the neighboring coarse elements (rep. the half of every neighboring coarse elements) with it. Thus, the case with $\theta_0 = 1$ (rep.

$\theta_0 = \frac{1}{2}$) is called “complete overlap” (rep. “half overlap”). We point out that the case with a small θ_0 , i.e., small overlap (for example, $\theta_0 = \frac{h}{d}$) is not considered in this paper, since the numerical results for this case are not satisfactory (see Table 9 in Section 6).

For convenience, the above process to generate the coarse elements $\{D_r\}$ and the overlapping subdomains $\{\tilde{D}_r\}$ from D is called a “decomposition operation” of D . The subdomain \tilde{D}_r is called the “enlarged subdomain” of D_r .

When D is just Ω itself, we let $\mathcal{T}_{d_0}^\Omega$ denote the set of the resulting coarse elements $\Omega_1, \dots, \Omega_{N_0}$, where d_0 is the size of the elements $\Omega_1, \dots, \Omega_{N_0}$. Moreover, we use \mathcal{S}_0 to denote the set of the “enlarged subdomains” $\tilde{\Omega}_1, \tilde{\Omega}_2, \dots, \tilde{\Omega}_{N_0}$.

For each subdomain $D \in \mathcal{S}_0$, let $\mathcal{T}_{d_1}^D$ be the set of the coarse elements D_1, \dots, D_{N_0} defined by the “decomposition operation” of D , where d_1 denote the size of D_1, \dots, D_{N_0} . Let $\tilde{D} = \bigcup_{r=1}^{N_0} \tilde{D}_r$ denote the overlapping domain decomposition of D , where \tilde{D}_r is the “enlarged subdomain” of D_r . With all the “enlarged subdomains” at 1th-level, define the set

$$\mathcal{S}_1 = \{\tilde{D}_r : r = 1, \dots, N_0; \text{ for every } D \in \mathcal{S}_0\}.$$

We can repeat the above process. Let $J \geq 1$. For an integer j satisfying $1 \leq j \leq J$, we assume that the set \mathcal{S}_{j-1} consisting of overlapping subdomains of Ω has been defined. For each subdomain $D \in \mathcal{S}_{j-1}$, we use $\mathcal{T}_{d_j}^D$ to denote the set of the coarse elements D_1, \dots, D_{N_0} defined by the “decomposition operation” of D , with d_j being the size of the subdomains D_1, \dots, D_{N_0} . Let \tilde{D}_r be the “enlarged subdomain” of D_r , and let $\tilde{D} = \bigcup_{r=1}^{N_0} \tilde{D}_r$ denote the resulting overlapping domain decomposition of D . Define the set of j th-level “enlarged subdomains” as

$$\mathcal{S}_j = \{\tilde{D}_r : r = 1, \dots, N_0; \text{ for every } D \in \mathcal{S}_{j-1}\} \quad (j = 1, \dots, J).$$

We would like to point out that the numbers of the coarse elements generated by “decomposition operation” of two different subdomains may be different in applications, here the choice of the same number N_0 of coarse elements is only to simplify the description. When choosing N_0 properly, we have $d_0 > d_1 > \dots > d_j > h$. Then the number of *fine* elements contained in each $K \in \mathcal{S}_j$ decreases rapidly when j increases.

Corresponding to a “decomposition operation” of a subdomain, we can build a local space decomposition on the subdomain.

As in Section 2, let $y_l(\mathbf{x})$ denote the plane wave shape function $e^{i\omega(a_l \mathbf{x})}$ ($l = 1, \dots, p$). Let \mathcal{Q}_p be the space consisting of the p plane wave shape functions, i.e.,

$$\mathcal{Q}_p = \text{span}\{y_l : l = 1, \dots, p\}.$$

Define the *coarsest* plane wave space on Ω as

$$V_p(\mathcal{T}_{d_0}^\Omega) = \{v \in L^2(\Omega) : v|_K \in \mathcal{Q}_p \text{ for every } K \in \mathcal{T}_{d_0}^\Omega\}.$$

Similarly, for each $D \in \mathcal{S}_{j-1}$ with $j \geq 1$, define the *coarse* plane wave space on D by

$$V_p(\mathcal{T}_{d_j}^D) = \{v \in L^2(\Omega) : \text{supp } v \subset D; v|_K \in \mathcal{Q}_p \text{ for every } K \in \mathcal{T}_{d_j}^D\} \quad (j = 1, \dots, J),$$

namely, $V_p(\mathcal{T}_{d_j}^D)$ is the plane wave finite element space associated with the coarse partition $\mathcal{T}_{d_j}^D$. Notice that the spaces $V_p(\mathcal{T}_{d_0}^\Omega)$ and $V_p(\mathcal{T}_{d_j}^D)$ ($j = 1, \dots, J$) have the dimension $N_0 p$ and possess the same structure with the original plane wave finite element space $V_p(\mathcal{T}_h)$ defined in Subsection 2.4.

For a subdomain K that is the union of some fine elements in \mathcal{T}_h , we always use \mathcal{T}_h^K to denote the restriction of the original partition \mathcal{T}_h on K , and define the *fine* plane wave

space on K by

$$V_p(\mathcal{T}_h^K) = \{v \in V_p(\mathcal{T}_h) : \text{supp } v \subset K\}.$$

As in the standard overlapping domain decomposition method, we can obtain the initial space decomposition on Ω (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)

$$V_p(\mathcal{T}_h) = V_p(\mathcal{T}_{d_0}^\Omega) + \sum_{r=1}^{N_0} V_p(\mathcal{T}_h^{\tilde{\Omega}_r}) = V_p(\mathcal{T}_{d_0}^\Omega) + \sum_{D \in \mathcal{S}_0} V_p(\mathcal{T}_h^D). \quad (3.1)$$

Similarly, for each $D \in \mathcal{S}_{j-1}$ with $j \geq 1$, we can build the local space decomposition on D

$$V_p(\mathcal{T}_h^D) = V_p(\mathcal{T}_{d_j}^D) + \sum_{r=1}^{N_0} V_p(\mathcal{T}_h^{\tilde{D}_r}). \quad (3.2)$$

Set $j = 1$ in (3.2), and substituting the resulting decomposition into (3.1), yields

$$\begin{aligned} V_p(\mathcal{T}_h) &= V_p(\mathcal{T}_{d_0}^\Omega) + \sum_{D \in \mathcal{S}_0} V_p(\mathcal{T}_{d_1}^D) + \sum_{D \in \mathcal{S}_0} \sum_{r=1}^{N_0} V_p(\mathcal{T}_h^{\tilde{D}_r}) \\ &= V_p(\mathcal{T}_{d_0}^\Omega) + \sum_{D \in \mathcal{S}_0} V_p(\mathcal{T}_{d_1}^D) + \sum_{D \in \mathcal{S}_1} V_p(\mathcal{T}_h^D). \end{aligned}$$

Combining the above decomposition with (3.2) for $j = 2, \dots, J$, and using the relation

$$\sum_{D \in \mathcal{S}_{j-1}} \sum_{r=1}^{N_0} V_p(\mathcal{T}_h^{\tilde{D}_r}) = \sum_{D \in \mathcal{S}_j} V_p(\mathcal{T}_h^D) \quad (j \geq 2),$$

we recursively obtain the multilevel space decomposition

$$V_p(\mathcal{T}_h) = V_p(\mathcal{T}_{d_0}^\Omega) + \sum_{j=1}^J \sum_{D \in \mathcal{S}_{j-1}} V_p(\mathcal{T}_{d_j}^D) + \sum_{K \in \mathcal{S}_J} V_p(\mathcal{T}_h^K). \quad (3.3)$$

For ease of notation, we would like to give a terser expression of the above space decomposition.

For convenience, we write $V_p(\mathcal{T}_{d_0}^\Omega)$ as $V_p(\mathcal{T}_{d_0})$. For $j = 1, \dots, J$, define the *set of j th-level coarse elements*

$$\mathcal{T}_{d_j} = \bigcup_{D \in \mathcal{S}_{j-1}} \mathcal{T}_{d_j}^D$$

and *j th-level coarse space*

$$V_p(\mathcal{T}_{d_j}) = \sum_{D \in \mathcal{S}_{j-1}} V_p(\mathcal{T}_{d_j}^D).$$

Notice that, for $j \geq 1$, the set \mathcal{T}_{d_j} does not constitute a (coarse) finite element partition of Ω since the elements in $\mathcal{T}_{d_j}^D$ may be overlapping with the elements in $\mathcal{T}_{d_j}^{D'}$ when D is different from D' .

Moreover, we define the set of *J th-level fine elements*

$$\tilde{\mathcal{T}}_h^J = \bigcup_{K \in \mathcal{S}_J} \mathcal{T}_h^K$$

and *J th-level fine space*

$$V_p(\tilde{\mathcal{T}}_h^J) = \sum_{K \in \mathcal{S}_J} V_p(\mathcal{T}_h^K).$$

Also, the set $\tilde{\mathcal{T}}_h^J$ is not a (fine) finite element partition of Ω .

Therefore, the space decomposition (3.3) can be simplified as

$$V_p(\mathcal{T}_h) = V_p(\tilde{\mathcal{T}}_h^J) + \sum_{j=0}^J V_p(\mathcal{T}_{d_j}). \quad (3.4)$$

In the rest of this paper, we construct several multilevel preconditioners for A based on the above multilevel space decomposition.

3.2. A multilevel overlapping preconditioner. In this subsection, we construct a basic preconditioner of A by the multilevel space decomposition (3.4).

Let $A_0 : V_p(\mathcal{T}_{d_0}) \rightarrow V_p(\mathcal{T}_{d_0})$ be the restriction of the discrete operator A on the coarsest space $V_p(\mathcal{T}_{d_0})$, namely,

$$(A_0 v_0, w_0) = a(v_0, w_0), \quad v_0 \in V_p(\mathcal{T}_{d_0}), \quad \forall w_0 \in V_p(\mathcal{T}_{d_0}).$$

As usual, A_0 is called the *coarsest* solver.

Let $j = 1, \dots, J$. For $D \in \mathcal{S}_{j-1}$, let $V_p(\mathcal{T}_{d_j}^D)$ be the local coarse spaces defined in the last subsection. Define j th-level local coarse solvers $A_{d_j}^D : V_p(\mathcal{T}_{d_j}^D) \rightarrow V_p(\mathcal{T}_{d_j}^D)$ by

$$(A_{d_j}^D v, w) = a(v, w), \quad v \in V_p(\mathcal{T}_{d_j}^D), \quad \forall w \in V_p(\mathcal{T}_{d_j}^D) \quad (1 \leq j \leq J; D \in \mathcal{S}_{j-1}).$$

Then we define *inexact* solver $B_j : V_p(\mathcal{T}_{d_j}) \rightarrow V_p(\mathcal{T}_{d_j})$ at j th-level coarse space as:

$$B_j^{-1} = \sum_{D \in \mathcal{S}_{j-1}} (A_{d_j}^D)^{-1} Q_{d_j}^D \quad (j = 1, \dots, J),$$

where $Q_{d_j}^D : V_p(\mathcal{T}_{d_j}) \rightarrow V_p(\mathcal{T}_{d_j}^D)$ denote the L^2 projectors. Notice that the operator B_j can be viewed as a “block-diagonal” preconditioner for the restriction of A on j th-level coarse subspace $V_p(\mathcal{T}_{d_j})$, where the order of each “block” equals N_{0p} .

Similarly, for each $K \in \mathcal{S}_j$, define J th-level *local* solver $\tilde{A}_J^K : V_p(\tilde{\mathcal{T}}_h^K) \rightarrow V_p(\tilde{\mathcal{T}}_h^K)$ by

$$(\tilde{A}_J^K v, w) = a(v, w), \quad v \in V_p(\tilde{\mathcal{T}}_h^K), \quad \forall w \in V_p(\tilde{\mathcal{T}}_h^K) \quad (K \in \mathcal{S}_j),$$

and define J th-level *fine inexact* solver $\tilde{B}_J : V_p(\tilde{\mathcal{T}}_h^J) \rightarrow V_p(\tilde{\mathcal{T}}_h^J)$ as

$$\tilde{B}_J^{-1} = \sum_{K \in \mathcal{S}_j} (\tilde{A}_J^K)^{-1} \tilde{Q}_J^K,$$

where $\tilde{Q}_J^K : V_p(\tilde{\mathcal{T}}_h^J) \rightarrow V_p(\tilde{\mathcal{T}}_h^K)$ denote the L^2 projectors. It is clear that \tilde{B}_J is also a “block-diagonal” preconditioner for the restriction of A on the fine subspace $V_p(\tilde{\mathcal{T}}_h^J)$.

Finally, corresponding to the multilevel space decomposition (3.4), an additive multilevel preconditioner $B : V_p(\mathcal{T}_h) \rightarrow V_p(\mathcal{T}_h)$ is naturally defined as

$$B^{-1} = A_0^{-1} Q_0 + \sum_{j=1}^J B_j^{-1} Q_j + \tilde{B}_J^{-1} \tilde{Q}_J, \quad (3.5)$$

where Q_j ($j = 0, \dots, J$) and \tilde{Q}_J denote the L_2 projectors into $V_p(\mathcal{T}_{d_j})$ and $V_p(\tilde{\mathcal{T}}_h^J)$, respectively.

The action of B^{-1} can be described by the following algorithm.

Algorithm 3.1. For $\xi \in V_p(\mathcal{T}_h)$, the function $u_\xi = B^{-1}\xi \in V_p(\mathcal{T}_h)$ can be obtained as follows:

Step 1. Computing $u_0 \in V_p(\mathcal{T}_{d_0})$ by

$$(A_0 u_0, v_0) = (\xi, v_0), \quad \forall v_0 \in V_p(\mathcal{T}_{d_0});$$

Step 2. For $j = 1, \dots, J$, computing $u_{d_j} \in V_p(\mathcal{T}_{d_j})$ in parallel by

$$(B_j u_{d_j}, v) = (\xi, v), \quad \forall v \in V_p(\mathcal{T}_{d_j});$$

Step 3. Computing $\tilde{u}_h^J \in V_p(\tilde{\mathcal{T}}_h^J)$ by

$$(\tilde{B}_J \tilde{u}_h^J, v_h) = (\xi, v_h), \quad \forall v_h \in V_p(\tilde{\mathcal{T}}_h^J);$$

Set

$$u_\xi = u_0 + \sum_{j=1}^J u_{d_j} + \tilde{u}_h^J.$$

By the definitions of the solvers B_j ($j = 1, \dots, J$) and \tilde{B}_J , Step 2-Step 3 in **Algorithm 3.1** can be implemented in smaller spaces (otherwise, **Algorithm 3.1** has no significance).

The action of B_j^{-1} ($j = 1, \dots, J$) appeared in Step 2 of **Algorithm 3.1** can be described by the following algorithm

Algorithm 3.2. For $\eta \in V_p(\mathcal{T}_{d_j})$, the function $w_\eta = B_j^{-1}\eta \in V_p(\mathcal{T}_{d_j})$ can be obtained by two steps:

Step 1. For $D \in \mathcal{S}_{j-1}$, computing $w_{d_j}^D \in V_p(\mathcal{T}_{d_j}^D)$ in parallel by

$$a(w_{d_j}^D, v) = (\eta, v), \quad \forall v \in V_p(\mathcal{T}_{d_j}^D);$$

Step 2. Set

$$w_\eta = \sum_{D \in \mathcal{S}_{j-1}} w_{d_j}^D.$$

Similarly, the action of \tilde{B}_J^{-1} appeared in Step 3 of **Algorithm 3.1** can be described by the following algorithm

Algorithm 3.3. For $\eta \in V_p(\tilde{\mathcal{T}}_h^J)$, the function $\tilde{w}_\eta = \tilde{B}_J^{-1}\eta \in V_p(\tilde{\mathcal{T}}_h^J)$ can be obtained by two steps:

Step 1. For $K \in \mathcal{S}_J$, computing $\tilde{w}_h^K \in V_p(\tilde{\mathcal{T}}_h^K)$ in parallel by

$$a(\tilde{w}_h^K, v_h) = (\eta, v), \quad \forall v_h \in V_p(\tilde{\mathcal{T}}_h^K);$$

Step 2. Set

$$\tilde{w}_\eta = \sum_{K \in \mathcal{S}_J} \tilde{w}_h^K.$$

In applications, the action of B^{-1} is implemented in parallel by Step 1 in **Algorithm 3.1-Algorithm 3.3**.

Remark 3.1. Notice that the dimension of the coarsest space $V_p(\mathcal{T}_{d_0})$ and each local “coarse” space $V_p(\mathcal{T}_{d_j}^D)$ equals $N_0 p$. Moreover, the number of fine elements contained in $K \in \mathcal{S}_J$ monotonically decreases when J increases (assume that N_0 is chosen in a suitable rule). Therefore, in order to guarantee that every local space has almost the same dimension, we should choose J to be large enough such that each domain $K \in \mathcal{S}_J$ contains almost N_0 fine elements in \mathcal{T}_h . Then each subproblem needed to be solved in Step 1 of **Algorithm 3.2-Algorithm 3.3** has nearly $N_0 p$ unknowns only.

3.3. Further discussions on the proposed multilevel method. In this subsection we first give some comparisons between the proposed multilevel method and two existing multigrid methods, and then investigate more details on the proposed preconditioner B .

- Comparisons with the standard multigrid method with overlapping Schwarz smoothers

The preconditioner B defined in the previous two subsection looks like the standard multigrid preconditioner with overlapping Schwarz smoothers, but the two preconditioners have essential differences. In order to explain the differences in details, we first describe this standard preconditioner for the current situation.

As in Subsection 3.1, let Ω be decomposed into the union of several quasi-uniform and regular coarse elements with the size h_0 , where each coarse element is just the union of some fine elements in \mathcal{T}_h . Let \mathcal{T}_{h_0} denote the resulting partition, i.e., the set of all the coarse elements. For every element in \mathcal{T}_{h_0} , we continue such decomposition and obtain several quasi-uniform and regular coarse elements with the size $h_1 < h_0$. The resulting partition is denoted by $\hat{\mathcal{T}}_{h_1}$. As usual, we repeat the above decomposition process and we can build refining finite element partitions: $\mathcal{T}_{h_0}, \hat{\mathcal{T}}_{h_1}, \dots, \hat{\mathcal{T}}_{h_J}$ with the sizes h_0, h_1, \dots, h_J satisfying $h < h_J < \dots < h_1 < h_0$. For $j = 1, \dots, J$, let $V_p(\hat{\mathcal{T}}_{h_j})$ denote the plane wave finite element space associated with the finite element partitions $\hat{\mathcal{T}}_{h_j}$. Then we obtain the multilevel space decomposition

$$V_p(\mathcal{T}_h) = V_p(\mathcal{T}_{h_0}) + \sum_{j=1}^J V_p(\hat{\mathcal{T}}_{h_j}). \quad (3.6)$$

In order to define overlapping Schwarz smoothers, we decompose the space $V_p(\hat{\mathcal{T}}_{h_j})$ ($j \geq 1$) into the sum of smaller subspaces. For each $D \in \hat{\mathcal{T}}_{h_j}$, we enlarge D with the thickness of one (coarse) element to a larger domain \tilde{D} , i.e., \tilde{D} is the union of D and the coarse elements adjoining D , where the added elements belong to $\hat{\mathcal{T}}_{h_j}$. Then D and the added elements constitute a coarse finite element partition of \tilde{D} , which is denoted by $\hat{\mathcal{T}}_{h_j}^{\tilde{D}}$. Let $V_p(\hat{\mathcal{T}}_{h_j}^{\tilde{D}})$ denote the plane wave finite element space associated with the finite element partition $\hat{\mathcal{T}}_{h_j}^{\tilde{D}}$, i.e., the restriction of $V_p(\hat{\mathcal{T}}_{h_j})$ on the subdomain \tilde{D} . Then we have the ‘‘overlapping’’ space decomposition of the j th-level coarse space

$$V_p(\hat{\mathcal{T}}_{h_j}) = \sum_{D \in \hat{\mathcal{T}}_{h_j}} V_p(\hat{\mathcal{T}}_{h_j}^{\tilde{D}}) \quad (j \geq 1). \quad (3.7)$$

Combing this decomposition with (3.6), gives the new multilevel decomposition of the original space

$$V_p(\mathcal{T}_h) = V_p(\mathcal{T}_{h_0}) + \sum_{j=1}^J \sum_{D \in \hat{\mathcal{T}}_{h_j}} V_p(\hat{\mathcal{T}}_{h_j}^{\tilde{D}}). \quad (3.8)$$

As in Subsection 3.2, let A_0 be the coarsest solver associated with $V_p(\mathcal{T}_{h_0})$. We define $\hat{A}_{h_j}^{\tilde{D}} : V_p(\hat{\mathcal{T}}_{h_j}^{\tilde{D}}) \rightarrow V_p(\hat{\mathcal{T}}_{h_j}^{\tilde{D}})$ as the restriction of A on $V_p(\hat{\mathcal{T}}_{h_j}^{\tilde{D}})$, and use $\hat{Q}_{h_j}^{\tilde{D}} : V_p(\hat{\mathcal{T}}_{h_j}) \rightarrow V_p(\hat{\mathcal{T}}_{h_j}^{\tilde{D}})$ to denote the L^2 projector. Then we define j th-level solver

$$\hat{B}_j^{-1} = \sum_{D \in \hat{\mathcal{T}}_{h_j}} (\hat{A}_{h_j}^{\tilde{D}})^{-1} \hat{Q}_{h_j}^{\tilde{D}} \quad (1 \leq j \leq J)$$

and the preconditioner

$$\hat{B}^{-1} = A_0^{-1}Q_0 + \sum_{j=1}^J \hat{B}_j^{-1}Q_j. \quad (3.9)$$

For $j \geq 1$, the operator \hat{B}_j^{-1} is called the overlapping Schwarz smoother at j th-level, and the preconditioner \hat{B} is called the multigrid preconditioner with overlapping Schwarz smoothers (MG-Schwarz). For each coarse element $D \in \hat{\mathcal{T}}_{h_j}$, if we do not enlarge D into the larger subdomain \tilde{D} and replace the subspace $V_p(\hat{\mathcal{T}}_{h_j}^D)$ in (3.8) by $V_p(\hat{\mathcal{T}}_{h_j}^D)$ itself, then the corresponding preconditioner \hat{B} is just the multigrid preconditioner with Jacobi smoothers (MG-Jacobi). In applications, the action of the smoother \hat{B}_j^{-1} may be repeated several times by Richardson iterations. Notice that we have not considered the more general situation, in which the subdomain \tilde{D} contains more elements for each $D \in \hat{\mathcal{T}}_{h_j}$, since the implementation of the resulting smoothers has greater cost.

Now we give some comparisons between the preconditioner B defined in (3.5) and the preconditioner \hat{B} defined in (3.9). We need only to compare the two multilevel space decompositions (3.3) and (3.8).

Similarity: for both multilevel space decompositions, the subspaces in each level (except the coarsest level) are overlapping each other.

Differences:

(1) the two space decompositions are constructed in different ways. For the space decomposition (3.8), we first have the multilevel decomposition (3.6), and then construct independently the overlapping decomposition (3.7) for each level coarse space. However, for the space decompositions (3.3), we first construct the overlapping decomposition (3.2) on each “enlarged subdomain”, and then use all these local overlapping decompositions to derive recursively the global multilevel space decomposition (3.3).

(2) the two space decompositions have different structures. The design of the overlapping decomposition (3.7) only changes the structure of j th-level space itself, but does not improve the relation of the coarse spaces at different levels. This means that the structure of the space decomposition (3.8) has no essential difference from that in the multilevel preconditioner with Jacobi smoothers. From the construction of the space decomposition (3.3), we know that the space decomposition (3.3) locally possesses the structure of the space decomposition in the overlapping domain decomposition method, and so the overlapping subspaces $V_p(\mathcal{T}_{d_j}^D)$ at different levels have inherent connections. It is easy to see that the space decomposition (3.3) is independent of the space decomposition in the standard multigrid preconditioners. Some comparison results for them will be given in Table 10 of Section 6.

- Comparisons with the wave-ray multigrid methods

The wave-ray multigrid methods (see [32] and [31]) were designed for solving Helmholtz system generated by the discretization with finite difference or the nodal finite elements.

As in the first part of this subsection, let h_j denote the size of the coarse elements at j -th level. It is well known that, when h_j is relatively large comparing the value of $1/\omega$, the oscillatory error components at j -th level can not be efficiently reduced by the standard multigrid methods. The basic idea of the wave-ray multigrid methods is to approximate such oscillatory error components at j -th level by the following functions

$$w^j(\mathbf{x}) = \sum_{l=1}^{L_j} a_l^j(\mathbf{x}) e^{i\omega(\alpha_l \cdot \mathbf{x})},$$

where $a_l^j(\mathbf{x})$ are smooth functions, which are called *ray envelope functions* in [32]; the wave direction vectors $\{\alpha_l\}$ may be different from that given in Subsection 2.4. The number L_j of the wave directions increases when the value ωh_j increases.

Since the original error components can not be directly expressed as the form of the function $w^j(\mathbf{x})$, some exponential interpolations need to be constructed (see [31]). These exponential interpolations were defined by the Fourier components (ray elements, plane wave functions) $e^{i\omega(\alpha_l \cdot \mathbf{x})}$, and were used to achieve a transformation between the original error components and the ray envelope functions. In the wave-ray multigrid methods, the approximation of oscillatory error components was transformed into the approximation of smooth ray envelope functions by using the exponential interpolations. While the smooth ray envelope functions can be approximated by the standard multigrid methods. Then the oscillatory error components can be reduced on relatively coarse grids. The implementation of the wave-ray multigrid methods involves many technical details, for example, how to choose suitable wave direction vectors $\{\alpha_l\}$. The cost in the wave-ray multigrid methods depends on the value of the wave number L_j and the calculation of the exponential interpolations.

Notice that both the wave-ray multigrid method and the multilevel method introduced in this paper are based on the plane wave functions $e^{i\omega(\alpha_l \cdot \mathbf{x})}$, in essence, use the “good” approximate property of the plane wave functions for oscillatory solutions. However, the roles of the plane wave functions are different in the two kinds of methods: the plane wave functions are used to define discretization basis functions in this paper; while, the plane wave functions are only auxiliary *weight* functions in the wave-ray multigrid methods. As to the multilevel methods themselves, the multilevel method described in the previous two subsections has no relation with the wave-ray multigrid methods, since the wave-ray multigrid methods still use the standard multigrid framework to approximate the smooth ray envelope functions.

- On the efficiency of the proposed multilevel method.

In Section 6, we will test several examples to illustrate the efficiency of the proposed multilevel preconditioners (including some variants of B , see Sections 4-5). Besides, we will give numerical comparisons among the proposed multilevel preconditioner B , the multigrid preconditioner \hat{B} with overlapping Schwarz smoothers and the multigrid preconditioner with Jacobi smoothers. As we will see, the multilevel preconditioner B designed in the previous two subsections is robust even for large ω . However, the multilevel preconditioner \hat{B} with overlapping Schwarz smoothers can only slightly improve the convergence rate of the multilevel preconditioner with Jacobi smoothers. In this part, we try to give some explanations to the effectiveness of the preconditioner B .

The first reason is that the plane wave functions can approximate the oscillatory solution of the Helmholtz equation very well (which is just the motive of the wave-ray multigrid methods), but it is not the unique reason of the effectiveness. In fact, if we decrease the thickness of the overlap to be one fine element in the overlapping space decomposition (3.3), then the resulting multilevel preconditioner has almost the same convergence rate with the multigrid preconditioner \hat{B} with Schwarz smoothers (see the results reported in Table 9 of Section 6). The second reason is that the space decomposition (3.3) possesses “good” structure, as explained in the first part of this subsection. We would like to explain this point more clearly. It is known that an overlapping domain decomposition preconditioner with several subdomains only is always stable even for the Helmholtz equations with large wave numbers (see the results listed in Table 8 of Section 6). Thus, since the number N_0 of overlapping subdomains is fixed and not large, the overlapping decomposition (3.2)

(and (3.1)) is stable for each $D \in \mathcal{S}_{j-1}$ even for large ω . This means that the global space decomposition (3.3), which is defined by the local space decompositions (3.1) and (3.2), should be also stable even for large ω . Notice that each local space $V_p(\mathcal{T}_h^{\tilde{D}_r})$ has too high dimension unless j is large, so we have to make multilevel decomposition.

- Computational cost for the implementation of the proposed preconditioner B .

In applications, the action of B^{-1} is implemented *in parallel*. Thus we should not investigate the computational complexity for the implementation of B^{-1} as successive algorithm. But, for completeness, we still estimate the computational complexity in the usual way.

As in Section 2, let N denote the number of the fine elements in \mathcal{T}_h . It is easy to see that the numbers of different subproblems needed to be solved in **Algorithm 3.2** and **Algorithm 3.3** are not greater than $\min\{N_0^{j-1}, N\}$ ($j = 1, \dots, J$) and $\min\{N_0^J, N\}$, respectively. If we require that the number of the fine elements contained in each $K \in \mathcal{S}_J$ almost equals N_0 (refer to Remark 3.1), we can verify that the level number J should be $c_0(\log_{N_0} N)$, where c_0 is a positive constant depending on N_0 and θ_0 . Notice that each subproblem to be solved has $N_0 p$ unknowns, so its solution has the computational cost $O((N_0 p)^3)$. Then the computational complexity for the implementation of B^{-1} can be estimated as follows

$$N_{cost} \leq C(N_0 p)^3 J \min\{N_0^J, N\} \leq C N_0^3 p^3 (\log_{N_0} N) N = C N_0^3 p^2 (\log_{N_0} N) (pN).$$

Then we have

$$N_{cost} \leq C N_0^3 p^2 (\log_{N_0} N) N_{dof},$$

where $N_{dof} = pN$ denotes the dimension of the original fine grid system (2.18). Since N_0 is a constant, the computational cost is estimated by $N_{cost} = O(p^2 (\log_{N_0} N) N_{dof})$. This means that, even if we implement the action of B^{-1} in *successive* manner, the resulting computational complexity is almost the optimal. Since the solution of each subproblem has very small cost $O((pN_0)^3)$, the preconditioner B implemented in parallel should be much cheaper than the direct solvers.

4. A MULTILEVEL OVERLAPPING PRECONDITIONER WITH SMOOTHERS

In this section, we design an improvement of the preconditioner B to further reduce the cost for implementing the solvers B_j^{-1} ($j = 1, \dots, J$) and \tilde{B}_J^{-1} described in **Algorithm 3.2** and **Algorithm 3.3**. The basic idea is to replace the solvers B_j^{-1} ($j = 1, \dots, J$) and \tilde{B}_J^{-1} by Jacobi-type smoothers. To this end, we first give exact definitions of the smoothers.

For $j = 1, \dots, J$ and $D \in \mathcal{S}_{j-1}$, let $V_p(\mathcal{T}_{d_j}^D)$ denote the local coarse space defined in Subsection 3.1. We want to further decompose each space $V_p(\mathcal{T}_{d_j}^D)$ into the sum of several smaller spaces. Notice that the support set of the functions in $V_p(\mathcal{T}_{d_j}^D)$ is D , which is the union of N_0 coarse elements D_1, \dots, D_{N_0} in $\mathcal{T}_{d_j}^D$. Thus we need only to define subspaces on the coarse elements.

As in Subsection 3.1, let \mathcal{Q}_p denote the space of p plane wave shape functions. For a coarse element D_r in $\mathcal{T}_{d_j}^D$, define

$$V_p(D_r) = \text{span}\{v \in L^2(\Omega) : v|_{D_r} \in \mathcal{Q}_p; \text{supp } v \subset D_r\} = \{v \in V_p(\mathcal{T}_{d_j}^D) : \text{supp } v \subset D_r\}.$$

$$(j = 1, \dots, J; D \in \mathcal{S}_{j-1}; r = 1, \dots, N_0)$$

In other words, $V_p(D_r)$ is the restriction space of $V_p(\mathcal{T}_{d_j}^D)$ on $D_r \subset D$. It is clear that the space $V_p(D_r)$ has the dimension p (but the dimension of $V_p(\mathcal{T}_{d_j}^D)$ equals $N_0 p$). Then

$$V_p(\mathcal{T}_{d_j}^D) = \sum_{r=1}^{N_0} V_p(D_r),$$

and so

$$V_p(\mathcal{T}_{d_j}) = \sum_{D \in \mathcal{S}_{j-1}} \sum_{r=1}^{N_0} V_p(D_r).$$

Similarly, for each *fine* element $E \in \tilde{\mathcal{T}}_h^K$, define

$$V_p(E) = \{v \in V_p(\tilde{\mathcal{T}}_h^K) : \text{supp } v \subset E\} \quad (K \in \mathcal{S}_J; E \in \tilde{\mathcal{T}}_h^K).$$

It is clear that the dimension of $V_p(E)$ equals p and we have

$$V_p(\tilde{\mathcal{T}}_h^K) = \sum_{E \in \tilde{\mathcal{T}}_h^K} V_p(E).$$

Then

$$V_p(\tilde{\mathcal{T}}_h^J) = \sum_{K \in \mathcal{S}_J} \sum_{E \in \tilde{\mathcal{T}}_h^K} V_p(E).$$

Based on the above space decompositions, we can define Jacobi-type smoothers in the natural manner.

Let m_0 be a given positive integer. The desired smoothers $R_j^{(m_0)}$ ($j = 1, \dots, J$) and $\tilde{R}_J^{(m_0)}$ are defined by the following algorithms.

Algorithm 4.1. For $\eta \in V_p(\mathcal{T}_{d_j})$, the function $w_\eta = (R_j^{(m_0)})^{-1} \eta \in V_p(\mathcal{T}_{d_j})$ can be obtained as follows:

Step 1. Let $w^0 \in V_p(\mathcal{T}_{d_j})$ be an initial guess. Assume that w^{l-1} ($l = 1, \dots, m_0$) has been gotten. For $D \in \mathcal{S}_{j-1}$ and elements $D_r \in \mathcal{T}_{d_j}^D$, computing $\hat{w}_{D_r}^l \in V_p(D_r)$ in parallel by

$$a(\hat{w}_{D_r}^l, v) = (\eta, v) - a(w^{l-1}, v), \quad \forall v \in V_p(D_r),$$

and set

$$w^l = w^{l-1} + \sum_{D \in \mathcal{S}_{j-1}} \sum_{r=1}^{N_0} \hat{w}_{D_r}^l \quad (l = 1, \dots, m_0);$$

Step 2. Define $w_\eta = w^{m_0}$.

Algorithm 4.2. For $\eta \in V_p(\tilde{\mathcal{T}}_h^J)$, the function $w_\eta = (\tilde{R}_J^{(m_0)})^{-1} \eta \in V_p(\tilde{\mathcal{T}}_h^J)$ can be obtained as follows:

Step 1. Let $w^0 \in V_p(\tilde{\mathcal{T}}_h^J)$ be an initial guess. Assume that w^{l-1} ($l = 1, \dots, m_0$) has been gotten. For $K \in \mathcal{S}_J$ and *fine* elements $E \in \tilde{\mathcal{T}}_h^K$, computing $\hat{w}_E^l \in V_p(E)$ in parallel by

$$a(\hat{w}_E^l, v) = (\eta, v) - a(w^{l-1}, v), \quad \forall v \in V_p(E),$$

and set

$$w^l = w^{l-1} + \sum_{K \in \mathcal{S}_J} \sum_{E \in \tilde{\mathcal{T}}_h^K} \hat{w}_E^l \quad (l = 1, \dots, m_0);$$

Step 2. Define $w_\eta = w^{m_0}$.

Next we define a new multilevel preconditioner.

Let m_0 be a given positive integer, and let $(R_j^{(m_0)})^{-1}$ and $(\tilde{R}_J^{(m_0)})^{-1}$ denote the smoothers defined by **Algorithm 4.1** and **Algorithm 4.2**, respectively. Define the additive preconditioner

$$(B_s^{(m_0)})^{-1} = A_0^{-1}Q_0 + \sum_{j=1}^J (R_j^{(m_0)})^{-1}Q_j + (\tilde{R}_J^{(m_0)})^{-1}\tilde{Q}_J.$$

In applications, we can choose the positive integer m_0 as $m_0 = 2, 3$. The action of $(B_s^{(m_0)})^{-1}$ can be implemented as in **Algorithm 3.1**, provided that the solvers B_j ($j = 1, \dots, J$) and \tilde{B}_J are replaced with $R_j^{(m_0)}$ ($j = 1, \dots, J$) and $\tilde{R}_J^{(m_0)}$ defined by **Algorithm 4.1** and **Algorithm 4.2**. Since the actions of $(R_j^{(m_0)})^{-1}$ ($j = 1, \dots, J$) and $(\tilde{R}_J^{(m_0)})^{-1}$ are implemented in smaller spaces, one of which is defined on an (coarse or fine) element and has only p degree of freedoms, the preconditioner $B_s^{(m_0)}$ is cheaper than the preconditioner B . Numerical experiments in Section 6 will indicate that the new variant has faster convergence than the preconditioner B .

Remark 4.1. According to the discussions in Subsection 3.3, the preconditioner $B_s^{(m_0)}$ is different from the standard multigrid preconditioners since the space decomposition defining $B_s^{(m_0)}$ possesses different structure from the one corresponding to the standard multigrid preconditioners. The differences between $B_s^{(m_0)}$ and the multigrid preconditioner \hat{B} with overlapping Schwarz smoothers are more obvious: each subproblem (except the coarsest problem) to be solved in $B_s^{(m_0)}$ has p unknowns only, but each subproblem to be solved in \hat{B} has $n_{\tilde{D}} \times p$ unknowns, where $n_{\tilde{D}}$ denotes the number of the (coarse) elements contained in a subdomain \tilde{D} . The proposed method is not called as multigrid method, since the sets \mathcal{T}_{d_j} and $\tilde{\mathcal{T}}_h^J$ defining the multilevel spaces do not constitute grids on Ω yet. For convenience, we called the preconditioner $B_s^{(m_0)}$ as *multilevel overlapping preconditioners with smoothers* (MOPS).

Remark 4.2. Notice that the dimension of the coarsest space $V_p(\mathcal{T}_{d_0}^\Omega)$ equals N_0p , with N_0 being a constant independent of ω , h and p . Thus, it is cheap to realize the action of A_0^{-1} appearing in the preconditioner $B_s^{(m_0)}$ by the direct method (the values of N_0 and p are not large). Of course, the action of A_0^{-1} can be also replaced by implementing a cheaper preconditioner of A_0 . It is easy to construct such a cheaper preconditioner for A_0 since the space $V_p(\mathcal{T}_{d_0}^\Omega)$ is defined on N_0 coarsest elements with fixed size d_0 .

5. MULTIPLICATIVE VARIANTS OF THE PRECONDITIONER $B_s^{(m_0)}$

In this section, we design several multiplicative multilevel preconditioners to accelerate the convergence of the additive preconditioner $B_s^{(m_0)}$.

5.1. A basic multiplicative preconditioner. In this subsection, we introduce a simple multiplicative preconditioner.

Define the operator

$$P_0 = A_0^{-1}Q_0A.$$

Then P_0 is the energy projector from $V_p(\mathcal{T}_h)$ into the coarsest space $V_p(\mathcal{T}_{d_0})$. Let $R_j^{(m_0)}$ ($j = 1, \dots, J$) and $\tilde{R}_J^{(m_0)}$ be the smoothers defined in the last section, and set

$$T_j^{(m_0)} = (R_j^{(m_0)})^{-1}Q_jA \quad (j = 1, \dots, J) \quad \text{and} \quad \tilde{T}_J^{(m_0)} = (\tilde{R}_J^{(m_0)})^{-1}\tilde{Q}_JA.$$

Let I denote the identity operator on $V_p(\mathcal{T}_h)$. Associated with the space decomposition (3.4), a multiplicative variant of $B_s^{(m_0)}$ is defined by

$$(M_1^{(m_0)})^{-1} = (I - (I - P_0)(I - T_1^{(m_0)})) \cdots (I - T_J^{(m_0)})(I - \tilde{T}_J^{(m_0)})A^{-1}.$$

The error propagation operator of $M_1^{(m_0)}$ is

$$I - (M_1^{(m_0)})^{-1}A = (I - P_0)(I - T_1^{(m_0)}) \cdots (I - T_J^{(m_0)})(I - \tilde{T}_J^{(m_0)}).$$

The action of $(M_1^{(m_0)})^{-1}$ can be described by the following algorithm.

Algorithm 5.1. For $\xi \in V_p(\mathcal{T}_h)$, the function $u_\xi = (M_1^{(m_0)})^{-1}\xi \in V_p(\mathcal{T}_h)$ can be obtained as follows:

Step 1. Computing $\tilde{u}_h^J \in V_p(\tilde{\mathcal{T}}_h^J)$ by

$$(\tilde{R}_J^{(m_0)}\tilde{u}_h^J, v_h) = (\xi, v_h), \quad \forall v_h \in V_p(\tilde{\mathcal{T}}_h^J);$$

Step 2. Computing $u_{d_j} \in V_p(\mathcal{T}_{d_j})$ by

$$(R_J^{(m_0)}u_{d_j}, v) = (\xi, v) - a(\tilde{u}_h^J, v), \quad \forall v \in V_p(\mathcal{T}_{d_j}),$$

and set $u_J = \tilde{u}_h^J + u_{d_j}$;

Step 3. Let $j = J, \dots, 2$. If we have obtained $u_j \in V_p(\mathcal{T}_h)$, then compute $u_{d_{j-1}} \in V_p(\mathcal{T}_{d_{j-1}})$ by

$$(R_{j-1}^{(m_0)}u_{d_{j-1}}, v) = (\xi, v) - a(u_j, v), \quad \forall v \in V_p(\mathcal{T}_{d_{j-1}}),$$

and set

$$u_{j-1} = u_j + u_{d_{j-1}} \quad (j = J, \dots, 2);$$

Step 4. Computing $u_0 \in V_p(\mathcal{T}_{d_0})$ by

$$(A_0u_0, v) = (\xi, v) - a(u_1, v), \quad \forall v \in V_p(\mathcal{T}_{d_0});$$

Step 5. Set

$$u_\xi = u_1 + u_0.$$

5.2. The standardly symmetrized multiplicative preconditioner. In this subsection we consider the case of PWLS method. Then the operator A is Hermitian positive definite with respect to the inner product $(A \cdot, \cdot)$. Thus, we need to define a symmetrization of the preconditioner $M_1^{(m_0)}$.

The standardly symmetrized preconditioner of $M_1^{(m_0)}$ is defined as

$$(M_2^{(m_0)})^{-1} = (I - (I - \tilde{T}_J^{(m_0)})(I - T_J^{(m_0)})) \cdots (I - T_1^{(m_0)})(I - P_0)(I - T_1^{(m_0)}) \cdots (I - T_J^{(m_0)})(I - \tilde{T}_J^{(m_0)})A^{-1}.$$

The error propagation operator of $M_2^{(m_0)}$ is

$$I - (M_2^{(m_0)})^{-1}A = (I - \tilde{T}_J^{(m_0)})(I - T_J^{(m_0)}) \cdots (I - T_1^{(m_0)})(I - P_0)(I - T_1^{(m_0)}) \cdots (I - T_J^{(m_0)})(I - \tilde{T}_J^{(m_0)}).$$

For the case of PWLS method, the operators $T_j^{(m_0)}$ and $\tilde{T}_j^{(m_0)}$ are Hermitian positive definite with respect to the inner product $(A \cdot, \cdot)$. As a result, the operator $(M_2^{(m_0)})^{-1}$ is also Hermitian and positive definite with respect to the same inner product.

The action of $(M_2^{(m_0)})^{-1}$ can be described by the following algorithm.

Algorithm 5.2. For $\xi \in V_p(\mathcal{T}_h)$, the function $u_\xi = (M_2^{(m_0)})^{-1}\xi \in V_p(\mathcal{T}_h)$ can be obtained as follows:

Step 1. Computing $\tilde{w}_h^J \in V_p(\tilde{\mathcal{T}}_h^J)$ by

$$(\tilde{R}_J^{(m_0)} \tilde{w}_h^J, v_h) = (\xi, v_h), \quad \forall v_h \in V_p(\tilde{\mathcal{T}}_h^J);$$

Step 2. Computing $w_{d_j} \in V_p(\mathcal{T}_{d_j})$ by

$$(\mathcal{R}_J^{(m_0)} w_{d_j}, v) = (\xi, v) - a(\tilde{w}_h^J, v), \quad \forall v \in V_p(\mathcal{T}_{d_j}),$$

and set $w_J = \tilde{w}_h^J + w_{d_J}$;

Step 3. Let $j = J, \dots, 2$. If we have obtained $w_j \in V_p(\mathcal{T}_h)$, then compute $w_{d_{j-1}} \in V_p(\mathcal{T}_{d_{j-1}})$ by

$$(\mathcal{R}_{j-1}^{(m_0)} w_{d_{j-1}}, v) = (\xi, v) - a(w_j, v), \quad \forall v \in V_p(\mathcal{T}_{d_{j-1}}),$$

and set

$$w_{j-1} = w_j + w_{d_{j-1}} \quad (j = J, \dots, 2);$$

Step 4. Computing $u_{d_0} \in V_p(\mathcal{T}_{d_0})$ by

$$(A_0 u_{d_0}, v) = (\xi, v) - a(w_1, v), \quad \forall v \in V_p(\mathcal{T}_{d_0}),$$

and set $u_0 = w_1 + u_{d_0}$;

Step 5. Let $j = 1, \dots, J$. If we have obtained $u_{j-1} \in V_p(\mathcal{T}_h)$, then compute $u_{d_j} \in V_p(\mathcal{T}_{d_j})$ by

$$(\mathcal{R}_j^{(m_0)} u_{d_j}, v) = (\xi, v) - a(u_{j-1}, v), \quad \forall v \in V_p(\mathcal{T}_{d_j}),$$

and set

$$u_j = u_{j-1} + u_{d_j} \quad (j = 1, \dots, J).$$

Step 6. Computing $\tilde{u}_h^J \in V_p(\tilde{\mathcal{T}}_h^J)$ by

$$(\tilde{R}_J^{(m_0)} \tilde{u}_h^J, v_h) = (g, v_h) - a(u_J, v_h), \quad \forall v_h \in V_p(\tilde{\mathcal{T}}_h^J);$$

Step 7. Set

$$u_\xi = u_J + \tilde{u}_h^J.$$

5.3. A non-standard symmetrized multiplicative preconditioner. In this subsection, we still consider the case of PWLS method. Define the operator $T^{(m_0)} : V_p(\mathcal{T}_h) \rightarrow V_p(\mathcal{T}_h)$ by

$$T^{(m_0)} = I - (I - T_1^{(m_0)}) \cdots (I - T_J^{(m_0)})(I - \tilde{T}_J^{(m_0)})(I - T_J^{(m_0)}) \cdots (I - T_1^{(m_0)}).$$

Then $T^{(m_0)}$ is Hermitian positive definite with respect to the inner product $(A \cdot, \cdot)$. A non-standard symmetrized preconditioner of $M_1^{(m_0)}$ can be defined as (refer to [22])

$$(M_3^{(m_0)})^{-1} = (I - (I - P_0)(I - T^{(m_0)}))A^{-1} \quad (\text{the PWLS method}).$$

It can be verified that the restriction of $M_3^{(m_0)}$ on $(V_p(\mathcal{T}_{d_0}))^\perp$ is Hermitian positive definite with respect to the inner product $(A \cdot, \cdot)$ (refer to [22]). The error propagation operator of $M_3^{(m_0)}$ is

$$I - (M_3^{(m_0)})^{-1}A = (I - P_0)(I - T).$$

The action of $(M_3^{(m_0)})^{-1}$ can be described by the following algorithm.

Algorithm 5.3. For $\xi \in V_p(\mathcal{T}_h)$, the function $u_\xi = (M_3^{(m_0)})^{-1}\xi \in V_p(\mathcal{T}_h)$ can be obtained as follows:

Step 1. Computing $w_1 \in V_p(\mathcal{T}_{d_1})$ by

$$(\mathcal{R}_1^{(m_0)} w_1, v) = (\xi, v), \quad \forall v \in V_p(\mathcal{T}_{d_1});$$

Step 2. Let $j = 2, \dots, J$. If we have obtained $w_{j-1} \in V_p(\mathcal{T}_h)$, then compute $w_{d_j} \in V_p(\mathcal{T}_{d_j})$ by

$$(R_j^{(m_0)} w_{d_j}, v) = (\xi, v) - a(w_{j-1}, v), \quad \forall v \in V_p(\mathcal{T}_{d_j}),$$

and set

$$w_j = w_{j-1} + w_{d_j} \quad (j = 2, \dots, J);$$

Step 3. Computing $\tilde{w}_h^J \in V_p(\tilde{\mathcal{T}}_h^J)$ by

$$(\tilde{R}_J^{(m_0)} \tilde{w}_h^J, v_h) = (\xi, v_h) - a(w_J, v_h), \quad \forall v_h \in V_p(\tilde{\mathcal{T}}_h^J),$$

and set $\tilde{w}_J = w_J + \tilde{w}_h^J$;

Step 4. Computing $u_{d_j} \in V_p(\mathcal{T}_{d_j})$ by

$$(R_J^{(m_0)} u_{d_j}, v) = (\xi, v) - a(\tilde{w}_J, v), \quad \forall v \in V_p(\mathcal{T}_{d_j}),$$

and set $u_J = \tilde{w}_J + u_{d_j}$;

Step 5. Let $j = J, \dots, 2$. If we have obtained $u_j \in V_p(\mathcal{T}_h)$, then compute $u_{d_{j-1}} \in V_p(\mathcal{T}_{d_{j-1}})$ by

$$(R_{j-1}^{(m_0)} u_{d_{j-1}}, v) = (\xi, v) - a(u_j, v), \quad \forall v \in V_p(\mathcal{T}_{d_{j-1}}),$$

and set

$$u_{j-1} = u_j + u_{d_{j-1}} \quad (j = J, \dots, 2);$$

Step 6. Computing $u_0 \in V_p(\mathcal{T}_{d_0})$ by

$$(A_0 u_0, v) = (\xi, v) - a(u_1, v), \quad \forall v \in V_p(\mathcal{T}_{d_0});$$

Step 7. Set

$$u_\xi = u_1 + u_0.$$

Remark 5.1. The actions of $(R_j^{(m_0)})^{-1}$ ($j = 1, \dots, J$) and $(\tilde{R}_J^{(m_0)})$ used in **Algorithm 5.1** -**Algorithm 5.3** are implemented by **Algorithm 4.1** and **Algorithm 4.2**, respectively. Notice that the solver $(\tilde{R}_J^{(m_0)})^{-1}$ is implemented only one time in **Algorithm 5.3** (such solver needs to be implemented for two times in **Algorithm 5.2**), so the preconditioner $M_3^{(m_0)}$ is cheaper than $M_2^{(m_0)}$. It is interesting that the numerical results reported in Section 6 indicate that $M_3^{(m_0)}$ has faster convergence than $M_2^{(m_0)}$ (some explanations to the kind of phenomenon have been given in [22]).

6. NUMERICAL EXPERIMENTS

In this section we report numerical results to illustrate that the new preconditioners are effective for solving Helmholtz equations with large wave numbers.

In the examples tested in this part, we choose Ω as the rectangle $[0, 2] \times [0, 1]$, and we adopt a uniform partition \mathcal{T}_h for the domain Ω as follows: Ω is divided into some small rectangles with the same size, where h denotes the length of the longest edge of the elements. Let n_h denote the number of elements generated by the partition \mathcal{T}_h , and let p denote the number of plane wave basis functions in one element. Then the dimension of the original fine grid system (2.18) is $N_{dof} = n_h \times p$.

We choose the mesh size h and the number p of plane wave basis functions in one element according to the following rule: when the wave numbers increase, the scale of the discrete problem is increased (either h decreases or p increases) in a suitable manner such that accepted relative L^2 errors of the approximation can be kept. In the numerical experiments below, we choose $h \approx 2/\omega$ and slightly increase p when ω increases.

We need to give a rule for the multilevel overlapping domain decomposition. For convenience, we consider only an easily implemented rule, i.e., the overlap degree $\theta_0 = 1$, for the main experiments. Let Ω be divided into $2^n \times 2^n$ ($n \geq 3$) rectangle elements with the same size. We divide Ω into 4 parts in each direction (x -coordinate axis direction or y -coordinate axis direction) to build the coarsest partition \mathcal{T}_{d_0} , with d_0 being a constant independent of the wave number ω and the fine mesh size h . This means that the coarsest partition contains 4×4 (coarse) rectangular elements with the same size, and so $N_0 = 16$. Define the enlarged subdomain of each (coarse) element as the union of the (coarse) element itself and its neighboring (coarse) elements, where the definition of the enlarged subdomain was given in Subsection 3.1. We repeat the above process to decompose each enlarged subdomain into 4×4 rectangles, but the rectangles may have different sizes since the number of the elements contained in a enlarged rectangle may be not divisible by 16. For this case, we still divide the enlarged rectangle into 4 parts in each direction such that the number of elements in each part is almost the same. We continue the above process, and the decomposition stops when the number of elements in each enlarged subdomain associated with the current level is less than 5×5 .

Throughout this section, we always use B , $B_s^{(m_0)}$ and $M_l^{(m_0)}$ ($l = 1, 2, 3$) to denote the proposed multilevel preconditioners with the above decomposition rule.

For the PWLS method, we set $\alpha = \omega^2$ and $\beta = 1$; for the PWDG method, we set $\alpha = \beta = \delta = \frac{1}{2}$. Since the stiffness matrix of PWLS method is Hermitian positive definite, we can solve the system by PCG method. While the stiffness matrix of PWDG is not Hermitian, we solve it by PGMRES method. For one iterative step, PCG method is cheaper than PGMRES method. The stopping criterion in the iterative algorithms is that the relative L^2 -norm ϵ of the residual of the iterative approximation satisfies $\epsilon < 1.0e - 6$.

Let N_{iter} represent the iteration count for solving the algebraic system. When the wave number ω increases (and the mesh size h decreases), the iteration count N_{iter} also increases. In order to describe the growth rate of the iteration count N_{iter} with respect to the wave number ω , we introduce a new notation ρ . Let ω_1 and ω_2 be two wave numbers, and let $N_{iter}^{(1)}$ and $N_{iter}^{(2)}$ denote the corresponding iteration counts, respectively. Then we define the positive number ρ by

$$\left(\frac{\omega_2}{\omega_1}\right)^\rho = \frac{N_{iter}^{(2)}}{N_{iter}^{(1)}}.$$

For example, when $\rho = 1$, the growth is linear; if $\rho \rightarrow 0^+$, then the preconditioner possesses the optimal convergence. For a preconditioner, the positive number ρ defined above is called as ‘‘relative growth rate’’ of the iteration count. Of course, we hope that the relative growth rate ρ is sufficiently small. In particular, a preconditioner is almost the optimal if the relative growth rate ρ is much less than 1.

6.1. An example with known analytic solution. The first model problem is the problem with the Robin boundary condition (refer to [25]):

$$\begin{aligned} \Delta u + \omega^2 u &= 0 \quad \text{in } \Omega, \\ \frac{\partial u}{\partial \mathbf{n}} + i\omega u &= g \quad \text{on } \partial\Omega, \end{aligned} \tag{6.1}$$

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

The analytic solution of the problem can be given 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

$$\begin{pmatrix} \omega_x & -\omega_x \\ (\omega - \omega_x)e^{-2i\omega_x} & (\omega + \omega_x)e^{2i\omega_x} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} -i \\ 0 \end{pmatrix} \quad (6.2)$$

In applications, the parameter k may have different values. According to our numerical experiments, different values of k do not affect the efficiency of the preconditioners (refer to Table 4 and Table 5 in [23]). Thus, in order to shorten the length of the paper, we only choose $k = 10$ in the experiments for the example.

Let u_h denote the approximate solution generated by an iterative method, we introduce the following relative error:

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

We use the above relative L^2 error to measure the accuracy of the approximate solution u_h .

6.1.1. *Results on the PWDG method.* In this part, we apply the PWDG method to the discretization of this example and solve the resulting algebraic system by PGMRES method, with the preconditioners B , $B_s^{(m_0)}$ and $M_1^{(m_0)}$. In Table 1, Table 2 and Table 3, we list the iteration counts and the L^2 errors of the resulting approximations.

TABLE 1
PWDG discretization and PGMRES iteration
(with the preconditioner B)

ω	p	n_h	N_{iter}	ρ	err.
20π	10	32^2	38		8.13e-4
40π	11	64^2	47	0.3067	7.69e-4
80π	12	128^2	58	0.3034	6.57e-4
160π	15	256^2	71	0.2918	6.02e-4
320π	16	512^2	87	0.2932	5.98e-4

TABLE 2
PWDG discretization and PGMRES iteration
(with the preconditioner $B_s^{(m_0)}$)

ω	p	n_h	$m_0 = 2$			$m_0 = 3$		
			N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	44		4.13e-4	42		4.23e-4
40π	11	64^2	53	0.2685	6.27e-4	50	0.2515	6.21e-4
80π	12	128^2	64	0.2721	5.21e-4	59	0.2388	3.97e-4
160π	15	256^2	77	0.2668	3.87e-4	69	0.2259	4.27e-4
320π	16	512^2	92	0.2568	3.96e-4	80	0.2134	4.12e-4

TABLE 3
PWDG discretization and PGMRES iteration
(with the preconditioner $M_1^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	39		6.52e-4	36		6.38e-4
40π	11	64^2	44	0.1740	5.87e-4	40	0.1520	5.06e-4
80π	12	128^2	50	0.1844	6.28e-4	44	0.1375	5.39e-4
160π	15	256^2	56	0.1635	6.14e-4	48	0.1255	5.22e-4
320π	16	512^2	63	0.1699	6.29e-4	52	0.1155	7.81e-4

The results in the above tables indicate that the proposed preconditioners are robust for Helmholtz equation with large wave numbers (some detailed comments will be given later).

6.1.2. *Results on the PWLS method.* In this part, we apply the PWLS method to the discretization of this example and solve the resulting systems by PCG method, with the preconditioners B , $B_s^{(m_0)}$, $M_2^{(m_0)}$ and $M_3^{(m_0)}$. We report the iteration counts and the L^2 errors of the resulting approximations in the following four tables.

TABLE 4
PWLS discretization and PCG iteration
(with the preconditioner B)

ω	p	n_h	N_{iter}	ρ	err.
20π	10	32^2	41		9.25e-4
40π	11	64^2	51	0.3149	3.60e-3
80π	14	128^2	63	0.3049	3.88e-4
160π	15	256^2	78	0.3081	2.31e-4
320π	16	512^2	96	0.2996	3.27e-4

TABLE 5
PWLS discretization and PCG iteration
(with the preconditioner $B_s^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	47		3.64e-4	45		3.67e-4
40π	11	64^2	57	0.2783	1.79e-3	53	0.2361	1.78e-3
80π	14	128^2	69	0.2756	2.63e-4	62	0.2263	3.89e-4
160π	15	256^2	83	0.2665	3.91e-4	72	0.2157	2.67e-4
320π	16	512^2	100	0.2688	4.37e-4	84	0.2224	4.63e-4

TABLE 6
PWLS discretization and PCG iteration
(with the preconditioner $M_2^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	28		6.97e-4	26		6.23e-4
40π	11	64^2	32	0.1926	2.37e-3	29	0.1575	2.13e-3
80π	14	128^2	36	0.1699	4.07e-4	32	0.1420	6.94e-4
160π	15	256^2	41	0.1876	6.24e-4	35	0.1293	7.83e-4
320π	16	512^2	46	0.1660	4.51e-4	38	0.1186	5.68e-4

TABLE 7
PWLS discretization and PCG iteration
(with the preconditioner $M_3^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	23		8.24e-4	22		8.29e-4
40π	11	64^2	26	0.1769	2.69e-3	24	0.1255	2.70e-3
80π	14	128^2	29	0.1575	3.91e-4	26	0.1155	3.90e-4
160π	15	256^2	33	0.1864	2.61e-4	28	0.1069	2.34e-4
320π	16	512^2	37	0.1651	2.97e-4	30	0.0995	2.28e-4

It can be seen, from the above tables, that the proposed multilevel preconditioners for Helmholtz equation with large wave numbers have relatively stable convergence. Namely, the iteration counts of the corresponding iterative methods (PCG or PGMRES) increase slowly when the wave number increases (and the mesh size decreases). In particular, for the multiplicative multilevel overlapping preconditioners with smoothers, the relative growth rates ρ of the iteration counts with respect to the wave numbers are very small. In fact, the rates are about 0.1 when the smoothing step $m_0 = 3$. This means that the multiplicative multilevel overlapping preconditioners with smoothers are almost optimal. We also notice that, for the PWLS method, the non-standard symmetrized preconditioner $M_3^{(m_0)}$ is more effective than the standardly symmetrized preconditioner $M_2^{(m_0)}$. We would like to emphasize that all the results are obtained without the limiting condition on the coarsest mesh size d_0 (see Section 1 for the details), which can be chosen as a constant independent of ω and the mesh size h .

In the next part, we report some results to explain why the proposed preconditioners are robust for the considered model, and illustrate the differences between the proposed preconditioners and several existing preconditioners.

6.1.3. *Results on some other related preconditioners.* In this part, we only apply the PWLS method to the discretization of this example and solve the resulting systems by PCG method with the considered preconditioners.

At first we consider the preconditioners generated by the non-overlapping domain decomposition method, the domain decomposition method with one element overlap and the domain decomposition method with complete overlap, respectively. Here we consider only the usual one-level domain decomposition (i.e., $J = 1$), in which Ω is decomposed into 4×4 rectangles with the same size. The resulting preconditioners are denoted by M_{non} , M_{small} and M_{large} . We give the iteration counts of the PCG methods with the three preconditioners in Table 8.

TABLE 8
 PWLS discretization and PCG iteration
 (with the preconditioners M_{non} , M_{small} and M_{large})

			M_{non}		M_{small}		M_{large}	
ω	p	n_h	N_{iter}	ρ	N_{iter}	ρ	N_{iter}	ρ
20π	10	32^2	108		82		20	
40π	11	64^2	139	0.3641	101	0.3007	21	0.0704
80π	14	128^2	178	0.3568	125	0.3076	22	0.0671
160π	15	256^2	229	0.3635	154	0.3010	23	0.0641

The above results indicate that, when we decompose Ω into several subdomains only, all the standard domain decomposition preconditioners have stable convergence (of course, the preconditioner with large overlap converges more rapidly). But, for this one-level decomposition, each subdomain still contains too many fine elements when h is small (i.e., ω is large). Because of this, we have to design multilevel domain decomposition in Section 3, such that each considered domain is decomposed into only several subdomains, and every subdomain at the final level contains several fine elements. Then each local space decomposition (3.2) is stable, and so the global space decomposition (3.3) should be stable too. This can intuitively explain why the proposed multilevel preconditioners are effective for Helmholtz equations with large wave numbers.

Then we investigate the influence of the overlapping degree θ_0 to the effectiveness of the multilevel preconditioner defined by (3.5). When decreasing the thickness of the overlap to be one fine element (i.e., $\theta_0 = \frac{h}{d}$), the resulting multilevel preconditioner is denoted by B_{small} (the preconditioner with small overlap). Let B_{half} denote the multilevel preconditioner with $\theta_0 = \frac{1}{2}$ (half overlap). In the table below, we list the iteration counts of the PCG methods with the two preconditioners and the errors of the resulting approximations.

TABLE 9
 PWLS discretization and PCG iteration
 (with the preconditioners B_{small} and B_{half})

			B_{small}			B_{half}		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	71		3.86e-4	49		4.91e-4
40π	11	64^2	99	0.4796	5.34e-4	61	0.3160	6.35e-4
80π	14	128^2	138	0.4792	2.74e-4	76	0.3172	2.71e-4
160π	15	256^2	193	0.4839	1.67e-4	95	0.3219	1.66e-4

The above results tell us that the multilevel preconditioner with small overlap is not satisfactory. Fortunately, the multilevel preconditioner with half overlap possesses almost the same convergence rate as the multilevel preconditioner B with complete overlap (comparing the results in Table 4). Notice that the overlap degree of the small overlap case depends on h , but the overlap degree for the case of complete overlap or half overlap is independent of h . This means that the convergence rate of the proposed preconditioner is mainly determined by the overlap degree, as in the standard overlapping domain decomposition method for diffusion equations.

In the following we compare the proposed preconditioner B with two standard multilevel preconditioners. Let \hat{B} (MG-Schwarz) be the multilevel preconditioner defined by (3.9), and let MG-Jacobi denote the multilevel preconditioner with Jacobi smoothers (see

the first part in Subsection 3.3). For the comparison, we use 4×4 refinement for all cases, i.e., choosing $N_0 = 4 \times 4$ in Subsection 3.1 and setting $h_j = h_{j-1}/4$ in Subsection 3.3. We report the iteration counts of the PCG methods with the three preconditioners in Table 10

TABLE 10
PWLS discretization and PCG iteration
(with the preconditioners MG-Jacobi, MG-Schwarz and B)

			MG-Jacobi		MG-Schwarz		B	
ω	p	n_h	N_{iter}	ρ	N_{iter}	ρ	N_{iter}	ρ
20π	10	32^2	78		62		41	
40π	11	64^2	113	0.5348	86	0.4721	51	0.3124
80π	14	128^2	163	0.5285	119	0.4686	63	0.3049
160π	15	256^2	235	0.5278	164	0.4627	78	0.3081

The results given in the above table indicate that the proposed preconditioner B is essentially different from the standard multilevel preconditioners and is obviously more effective than the considered two preconditioners (see Subsection 3.3 for the detailed explanations). We point out that, when setting $h_j = h_{j-1}/2$ in Subsection 3.3 or implementing more smoothing steps of the smoothers B_j^{-1} and \hat{B}_j^{-1} , this conclusion still holds.

Now we compare three preconditioners, in which each subproblem to be solved has p unknowns. When setting $h_j = h_{j-1}/2$ and implementing m_0 smoothing steps for the Jacobi smoothers, the resulting multigrid preconditioner with Jacobi smoothers is denoted by MG-Jacobi $^{(m_0)}$. If the smoothing step m_0 in the preconditioner $B_s^{(m_0)}$ described in Section 4 is not fixed, but it is determined by Krylov method (see [11]) with the control accuracy η , the resulting preconditioner is denoted by $B_{s,\eta}$. As an example, we choose $m_0 = 3$ and $\eta = \frac{1}{5}$, for which the average time for implementing smoothers in $B_{s,\eta}$ is about 2.7. In table 11, we list the iteration counts of the PCG methods with the three preconditioners.

TABLE 11
PWLS discretization and PCG iteration
(with the preconditioners MG-Jacobi $^{(m_0)}$, $B_{s,\eta}$ and $B_s^{(m_0)}$, where $m_0 = 3, \eta = 1/5$)

			MG-Jacobi $^{(m_0)}$		$B_{s,\eta}$		$B_s^{(m_0)}$	
ω	p	n_h	N_{iter}	ρ	N_{iter}	ρ	N_{iter}	ρ
20π	10	32^2	61		48		45	
40π	11	64^2	83	0.4443	58	0.2730	53	0.2361
80π	14	128^2	112	0.4323	70	0.2713	62	0.2263
160π	15	256^2	151	0.4310	84	0.2630	72	0.2157

It can be seen from the above results that the proposed preconditioner $B_s^{(m_0)}$ is obviously more effective than the multigrid preconditioner with m_0 Jacobi smoothing steps, and it is so effective as the preconditioner $B_{s,\eta}$. As pointed out in [11], the use of Krylov methods often plays an important role in other methods, but the conclusion is not true in the current multilevel method.

Notice that we have not reported the errors of the approximations in Table 8, Table 10 and Table 11 because of the limitation of the space in these tables. In fact, all the errors are less than 10^{-3} and have not large difference.

6.2. An example whose analytic solution is unknown. The example tested in the last subsection is too special. In this subsection, we consider the model with an arbitrary function g , which is not determined by an analytic solution. The example can be described as

$$\begin{aligned} \Delta u + \omega^2 u &= 0 \quad \text{in } \Omega, \\ \frac{\partial u}{\partial \mathbf{n}} + i\omega u &= g \quad \text{on } \partial\Omega, \end{aligned} \quad (6.3)$$

where $\Omega = [0, 2] \times [0, 1]$, and $g = x * y$.

In this example, since we do not know its analytic solution, we can only compute an approximate solution for the comparison with the iterative solution. Let \hat{u}_h be the approximate solution obtained by the direct method for the discrete system, i.e.,

$$\hat{u}_h = A^{-1} f_h.$$

To measure the accuracy of the approximate solution u_h generated by an iterative method, we introduce the following relative error:

$$\text{err.} = \frac{\|\hat{u}_h - u_h\|_{L^2(\Omega)}}{\|\hat{u}_h\|_{L^2(\Omega)}}.$$

6.2.1. Results on the PWDG method. In this part we apply the PWDG method to the discretization of this example and solve the resulting algebraic system by PGMRES method, with the preconditioners B , $B_s^{(m_0)}$ and $M_1^{(m_0)}$. In Table 12, Table 13 and Table 14, we report the iteration counts and the L^2 errors of the resulting approximations.

TABLE 12
PWDG discretization and PGMRES iteration
(with the preconditioner B)

ω	p	n_h	N_{iter}	ρ	err.
20π	10	32^2	44		$7.21\text{e-}4$
40π	11	64^2	54	0.2955	$7.39\text{e-}4$
80π	12	128^2	66	0.2895	$6.33\text{e-}4$
160π	15	256^2	80	0.2775	$6.19\text{e-}4$
320π	16	512^2	98	0.2928	$5.63\text{e-}4$

TABLE 13
PWDG discretization and PGMRES iteration
(with the preconditioner $B_s^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	51		$8.71\text{e-}4$	48		$6.22\text{e-}4$
40π	11	64^2	61	0.2583	$5.46\text{e-}4$	57	0.2479	$5.83\text{e-}4$
80π	12	128^2	73	0.2591	$6.74\text{e-}4$	67	0.2332	$5.91\text{e-}4$
160π	15	256^2	87	0.2531	$7.93\text{e-}4$	78	0.2193	$6.08\text{e-}4$
320π	16	512^2	104	0.2575	$6.28\text{e-}4$	91	0.2224	$5.69\text{e-}4$

TABLE 14
 PWDG discretization and PGMRES iteration
 (with the preconditioner $M_1^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	47		7.08e-4	41		8.26e-4
40π	11	64^2	53	0.1733	6.13e-4	45	0.1343	7.93e-4
80π	12	128^2	60	0.1790	6.29e-4	49	0.1229	6.15e-4
160π	15	256^2	68	0.1806	6.37e-4	53	0.1132	7.04e-4
320π	16	512^2	77	0.1793	7.24e-4	57	0.1050	6.87e-4

The above results indicate that the proposed preconditioners are also robust for this example.

6.2.2. *Results on the PWLS method.* In this part we apply the PWLS method to the discretization of this example and solve the resulting systems by PCG method, with the preconditioners B , $B_s^{(m_0)}$, $M_2^{(m_0)}$ and $M_3^{(m_0)}$. We list the iteration counts and the L^2 errors of the resulting approximations in the following four tables.

TABLE 15
 PWLS discretization and PCG iteration
 (with the preconditioner B)

ω	p	n_h	N_{iter}	ρ	err.
20π	10	32^2	49		6.27e-4
40π	11	64^2	60	0.2922	9.19e-3
80π	14	128^2	73	0.2829	4.70e-4
160π	15	256^2	89	0.2859	3.08e-4
320π	16	512^2	109	0.2925	5.81e-4

TABLE 16
 PWLS discretization and PCG iteration
 (with the preconditioner $B_s^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	56		7.89e-4	53		7.61e-4
40π	11	64^2	67	0.2587	2.96e-3	63	0.2494	2.37e-3
80π	14	128^2	80	0.2558	3.27e-4	74	0.2322	3.82e-4
160π	15	256^2	96	0.2630	5.14e-4	86	0.2168	6.35e-4
320π	16	512^2	115	0.2605	6.27e-4	100	0.2176	5.23e-4

TABLE 17
 PWLS discretization and PCG iteration
 (with the preconditioner $M_2^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	35		5.81e-4	33		6.14e-4
40π	11	64^2	40	0.1926	3.19e-3	36	0.1651	3.25e-3
80π	14	128^2	45	0.1699	6.34e-4	40	0.1520	5.12e-4
160π	15	256^2	51	0.1806	4.88e-4	44	0.1375	3.09e-4
320π	16	512^2	57	0.1605	5.33e-4	48	0.1225	5.87e-4

TABLE 18
PWLS discretization and PCG iteration
(with the preconditioner $M_3^{(m_0)}$)

			$m_0 = 2$			$m_0 = 3$		
ω	p	n_h	N_{iter}	ρ	err.	N_{iter}	ρ	err.
20π	10	32^2	31		4.76e-4	29		4.91e-4
40π	11	64^2	35	0.1751	8.68e-3	32	0.1420	9.01e-3
80π	14	128^2	39	0.1561	4.71e-4	35	0.1293	4.70e-4
160π	15	256^2	44	0.1740	3.92e-4	38	0.1186	3.91e-4
320π	16	512^2	49	0.1553	6.13e-4	41	0.1096	6.19e-4

From the above results, we know that the proposed multilevel preconditioners are also very effective for the Helmholtz equation considered in this subsection.

7. CONCLUSION

In this paper we have constructed several multilevel preconditioners for the Helmholtz systems generated by the plane wave discretization (PWLS or PWDG), based on a multilevel overlapping domain decomposition method. In particular, we have designed multilevel overlapping preconditioners with smoothers, which are almost the optimal. The numerical results have illustrated that the proposed preconditioners possess nearly stable convergence for the two-dimensional Helmholtz equations with large wave numbers, without the limiting condition on the coarse mesh size. In the next work we shall extend the proposed methods (with some modifications) to solving three-dimensional Helmholtz equations with large wave numbers.

REFERENCES

- [1] R. Bank, T. Dupont and H. Yserentant, The hierarchical basis multigrid method, Numer. Math., 52(1988), pp. 427-458.
- [2] J. H. Bramble, J. E. Pasciak, and J. Xu, The analysis of multigrid algorithms for nonsymmetric and indefinite elliptic problems, Math. Comp., 51 (1988), pp. 389-414.
- [3] J. Bramble, J. Pasciak and J. Xu, Parallel multilevel preconditioners, Math. Comp., 55 (1990), pp. 1-21.
- [4] A. Brandt, Multilevel adaptive solutions to boundary-value problems, Math. Comp., 31 (1977), pp. 333-309.
- [5] Brandt A, Livshits I. Wave-ray multigrid method for standing wave equations. ETNA (Electronic Transactions on Numerical Analysis) 1997;6:162-182.
- [6] M. Dryja, M.V. Sarkis and O. B. Widlund, Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions, Numer. Math., 72(1996): 313-348
- [7] W. Hackbusch, Multi-Grid Methods and Applications, Springer-Verlag, Berlin, Heidelberg, 1985.
- [8] 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.
- [9] 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), No.1, pp.255-299.

- [10] 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.
- [11] H. Elman, O. Ernst, and D. OLeary, A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations, *SIAM J. Sci. Comput.*, 23 (2001), pp. 1291-1315
- [12] B. Engquist, L. Ying. Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. *Multiscale Modeling Simulation*, 9(2011), No.2, pp.686-710.
- [13] Y. Erlangga, C. Osterlee and C. Vuik, A novel multigrid based preconditioner for heterogeneous Helmholtz problems, *SIAM J. Sci. Comput.*, 27(2006): 1471-1492
- [14] C. Farhat, R. Tezaur, J. Toivanen. A domain decomposition method for discontinuous Galerkin discretizations of Helmholtz problems with plane waves and Lagrange multipliers. *International journal for numerical methods in engineering*, 78(2009), No.13, pp.1513-1531.
- [15] P. Gamallo, R. Astley, A comparison of two Trefftz-type method: the ultra weak variational formulation and the least-squares method, for solving shortwave 2-D Helmholtz problems, *Int. J. Numer. Meth. Engng.*, 71(2007), No.4, pp.406-432.
- [16] M. Gander, F. Magoules, F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 24(2002), No.1, pp. 38-60.
- [17] C. Gittelsohn, R. Hiptmair and I. Perugia, Plane wave discontinuous Galerkin methods: Analysis of the h -version, *ESAIM: M2AN Math. Model. Numer. Anal.*, 43(2009), No.2, pp. 297-331.
- [18] J. Gopalakrishnan and J. Pasciak, Overlapping Schwarz preconditioners for indefinite time-harmonic Maxwell's equations, *Math. Comp.*, 72 (2003), 1-16.
- [19] R. Hiptmair, Multigrid method for Maxwell's equations, *SIAM J. Numer. Anal.*, 36(1998), 204-225.
- [20] 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.
- [21] R. Hiptmair, A. Moiola, and I. Perugia, Error analysis of Trefftz-discontinuous Galerkin methods for the time-harmonic Maxwell equations, *Math. Comp.*, 82(2013), No.281, pp. 247-268.
- [22] Q. Hu and G. Liang, Acceleration of the non-symmetrized, two-level iteration, *Appl. Numer. Math.*, 41(2002), No.2, pp.305-323.
- [23] Q. Hu and L. Yuan, A weighted variational formulation based on plane wave basis for discretization of Helmholtz equations, *Intern. J. of numer. anal. and model.*, 11(2014), No.3, pp.587-607.
- [24] 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), No.4, pp.1937-1959
- [25] T. Huttunen, P. Gamallo and R. Astley, Comparison of two wave element methods for the Helmholtz problem, *Commun. Numer. Meth. Engng.*, 25(2009), pp. 35-52.
- [26] T. Huttunen, M. Malinen, and P. Monk, Solving Maxwell's equations using the ultra weak variational formulation, *J. Comput. Phys.*, 223(2007), No.2, pp.731-758.
- [27] S. Kim and S. Kim, Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media, *SIAM J. Sci. Comput.*, 24 (2002), pp. 684C701.
- [28] L. Kovalevsky, P. Ladevèze, H. Riou, The Fourier version of the variational theory of complex rays for medium-frequency acoustics. *Computer Methods in Applied Mechanics and Engineering*, 225-228(2012), pp.142-153.
- [29] P. Ladevèze, A new computational approach for structure vibrations in the medium frequency range, *Comptes Rendus Académie des Sciences Paris*. 322(IIb) (1996), pp.849-856.
- [30] 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.
- [31] B. Lee, T. A. Manteuffel, S. F. McCormick and J. Ruge, First-order System least-squares for the Helmholtz equation, *SIAM J. Sci. Comput.*, 21(2000), No. 5, pp. 1927-1949
- [32] I. Livshits and A. Brandt, Accuracy properties of the wave-ray multigrid algorithm for Helmholtz equations, *SIAM J. Sci. Comput.*, 28(2006), 1228-1251.
- [33] F. Magoules, F. Roux, S. Salmon. Optimal discrete transmission conditions for a nonoverlapping domain decomposition method for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 25(2004), No.5, pp.1497-1515.
- [34] A. Moiola, R. Hiptmair and I. Perugia, Plane wave approximation of homogeneous Helmholtz solutions, *Z. Angew. Math. Phys.* 62(2011), No.5, pp.809-837.
- [35] P. Monk and D. Wang, A least-squares method for the Helmholtz equation, *Comput. Meth. Appl. Mech. Engng.*, 175(1999), No.1, pp.121-136

- [36] H. Riou, P. Ladevèze, B. Sourcis, B. Faverjon and L. Kovalevsky, An adaptive numerical strategy for the medium-frequency analysis of Helmholtz's problem, *J. Comput. Acous.*, 20(2012), No.1, DOI: 10.1142/S0218396X11004481.
- [37] 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), No.5, pp.796-815.
- [38] E. Trefftz, Ein gegenstück zum ritzschen verfahren, *Sec. Inte. Cong. Appl. Mech.*, (1926), pp.131-137.
- [39] H. Yserentant, On the multi-level splitting of finite element spaces for indefinite elliptic boundaryvalue problems, *SIAM J. Numer. Anal.*, 23 (1986), pp. 581-595.