Contents lists available at SciVerse ScienceDirect



# Journal of Computational Physics



journal homepage: www.elsevier.com/locate/jcp

# A sweeping preconditioner for time-harmonic Maxwell's equations with finite elements

Paul Tsuji<sup>a,\*</sup>, Bjorn Engquist<sup>b</sup>, Lexing Ying<sup>b</sup>

<sup>a</sup> ICES, University of Texas at Austin, Austin, TX 78712, United States <sup>b</sup> Department of Mathematics and ICES, University of Texas at Austin, TX 78712, United States

#### ARTICLE INFO

Article history: Received 31 August 2011 Received in revised form 30 December 2011 Accepted 19 January 2012 Available online 28 January 2012

Keywords: Maxwell's equations Frequency domain Finite element methods Preconditioners Fast solvers Perfectly matched layers Block LDL<sup>t</sup> factorization High-frequency waves

# 1. Introduction

#### ABSTRACT

This paper is concerned with preconditioning the stiffness matrix resulting from finite element discretizations of Maxwell's equations in the high frequency regime. The moving PML sweeping preconditioner, first introduced for the Helmholtz equation on a Cartesian finite difference grid, is generalized to an unstructured mesh with finite elements. The method dramatically reduces the number of GMRES iterations necessary for convergence, resulting in an almost linear complexity solver. Numerical examples including electromagnetic cloaking simulations are presented to demonstrate the efficiency of the proposed method. © 2012 Elsevier Inc. All rights reserved.

High frequency wave propagation and diffraction in heterogeneous media is a problem of great interest, especially in the field of electromagnetics. The modeling of complex metamaterials and coated scattering objects for high frequencies is in high demand in the electromagnetics community. Solving the forward problem of electromagnetic radiation is also a key part in the inversion methods of biomedical and radar imaging. As a result, fast and accurate numerical solutions to Maxwell's equations have become a top priority.

In the frequency domain, the most widely-used numerical technique for solving the heterogeneous medium problem of Maxwell's equations is the finite element method [14,16]. The reasons for its popularity are clear: it allows the modeling of structures with complex geometry quite naturally, and the stiffness matrix resulting from discretization is sparse, allowing the use of direct solvers such as the multifrontal method. Despite these advantages, however, solving Maxwell's equations in the high frequency regime remains a very difficult task for a few reasons. First, to retain a certain level of accuracy in the solution, each wavelength must be resolved with a reasonable number of elements. In fact, the pollution effect [2] forces the number of elements per wavelength to be increased if the same level of accuracy is desired at a higher frequency. If the domain is *K* wavelengths wide in each dimension, the number of degrees of freedom *N* is at least of order  $O(K^d)$ , where the dimension is d = 2, 3; for high frequencies, this number becomes extremely large. Second, the oscillatory nature of the dyadic Green's function for Maxwell's equations causes the stiffness matrix to be highly indefinite and ill-conditioned.

\* Corresponding author. E-mail address: ptsuji@gmail.com (P. Tsuji).

<sup>0021-9991/\$ -</sup> see front matter @ 2012 Elsevier Inc. All rights reserved. doi:10.1016/j.jcp.2012.01.025

Direct methods work well in 2D (with complexity  $O(N^{3/2})$ ) but become too expensive for 3D problems (with complexity  $O(N^2)$ ). As a result, iterative methods appear as the common choice in 3D. There has been extensive work done on the preconditioning and iterative solution of time-harmonic wave equations; the majority of the literature is focused on finite difference methods for the Helmholtz equation, which can be reviewed in [10]. For most iterative methods, the number of iterations necessary for convergence increases significantly with the wavenumber. For example, a preconditioner that has garnered much attention recently is the shifted Laplacian [3,11]. The main goal of the method is to "shift" the eigenvalues of the Helmholtz operator away from the origin and into the positive half of the complex plane; in doing so, the condition number is improved and the problem becomes less indefinite. For  $i = \sqrt{-1}$  and wavenumber  $\kappa$ , discretizing the complex-perturbed operator  $\Delta + (1 + i\beta)\kappa^2$  for  $\beta > 0$  and applying the inverse approximately can act as an effective preconditioner. The inversion can be achieved by either multigrid or incomplete *LU* decomposition; a comparison of the two methods is done in [12]. Unfortunately, convergence of the iterative solver still deteriorates as frequency increases.

Domain decomposition methods for the Helmholtz equation have also been developed by several research groups, following the early work of [15]. In these methods, for example in [5,7,13], the number of subdomains typically remains constant in order to maintain an iteration number that is essentially independent of the frequency. On the other hand, it has been observed in [5] that when the number of subdomains is increased, the convergence behavior of the iterative method deteriorates. As a result, these methods are mostly suitable for medium frequency problems in order to ensure that the solution of the subdomain problems remains manageable.

Recently, a new family of preconditioners for time-harmonic wave equations, called the *sweeping preconditioners*, was introduced in [8,9]. Developed for Cartesian finite difference grids of the Helmholtz equation, the initial step is to order the degrees of freedom layer by layer, starting from an absorbing side. The algorithm then continues by constructing a block  $LDL^t$  factorization, with each block corresponding to the degrees of freedom in a single layer. The main observation is that the inverse of each Schur complement block of the  $LDL^t$  factorization is the Green's function of the Dirichlet half-space problem, restricted to a line. As a result, it can be represented efficiently using the hierarchical matrix algebra [8] or by truncating the half-space problem further by moving the perfectly matched layer [9]. The approximate inverse of the  $LDL^t$  decomposition can be applied with O(N) complexity in 2D and  $O(N \log N)$  complexity in 3D. As a preconditioner, the number of GMRES iterations necessary for convergence is drastically lower compared to other preconditioning techniques, and is essentially independent of frequency.

The main contribution of this paper is two-fold: first, to generalize the sweeping preconditioner to Maxwell's equations, and second, to present the preconditioner for unstructured finite element meshes. Because the perfectly matched layer (PML) is ubiquitous in computational electromagnetics and is easy to work with in the finite element setting, we choose to adapt the moving-PML method in [9]. We will show that the preconditioner setup time scales as O(N) in 2D and  $O(N^{4/3})$  in 3D, while the application time for each iteration is O(N) in 2D and  $O(N \log N)$  in 3D. In addition, we will give evidence that the rate of convergence of the Krylov iterative solver behaves just as it did in the Helmholtz case; that is, essentially independent of wavenumber and mesh size.

The rest of the paper is as follows: in Section 2, we review the standard Galerkin method for Maxwell's equations, along with the PML formulation and choice of basis functions. In Section 3, we present the sweeping preconditioner for the finite element method. Numerical results for 2D are given in Section 4, while numerical results for 3D are given in Section 5. Finally, we conclude with some remarks and potential for future work in Section 6.

# 2. Finite element methods for Maxwell's equations

To avoid redundancies in the formulation, we proceed with the three-dimensional case; for 2D, we consider the transverse electric (TE) mode, where the fields take the form  $\mathbf{E} = (E_x, E_y, 0)$  and  $\mathbf{H} = (0, 0, H_z)$ . It is easy to obtain the 2D case by eliminating partial derivatives with respect to the *z* variable and setting the PML stretching variable in the *z*-direction,  $s_z$ , to be 1. We will remark on the 2D formulation when clarification is necessary.

#### 2.1. Perfectly matched layers

Consider the infinite-domain electromagnetic radiation problem in  $\mathbb{R}^3$  for electric field **E** = ( $E_x$ ,  $E_y$ ,  $E_z$ ), magnetic field **H** = ( $H_x$ ,  $H_y$ ,  $H_z$ ) and current source **J** = ( $J_x$ ,  $J_y$ ,  $J_z$ ). Maxwell's equations in the frequency domain are

$\nabla \times \mathbf{E} = -\imath \omega \mu_0 \mu_r \mathbf{H}$
$\nabla \times \mathbf{H} = \iota \omega \epsilon_0 \epsilon_r \mathbf{E} + \mathbf{J}$
$\nabla \cdot (\boldsymbol{\epsilon}_0 \boldsymbol{\epsilon}_r \mathbf{E}) = \rho$
$\nabla \cdot (\boldsymbol{\mu}_{0}\boldsymbol{\mu}_{r}\boldsymbol{H}) = \boldsymbol{0}$
$\lim_{ \textbf{r} \to\infty}(\textbf{H}\times\textbf{r}- \textbf{r} \textbf{E})=0$
$\lim_{ \mathbf{r} \to\infty} (\mathbf{E}\times\mathbf{r}+ \mathbf{r} \mathbf{H})=0,$

(2.1)

where  $\mathbf{r} = (x, y, z)$  and the last two conditions are the Silver–Müller radiation conditions; they are required to ensure that the solution radiates away from the source and dissipates to 0 as  $|\mathbf{r}| \to \infty$ . Here,  $\iota = \sqrt{-1}, \omega$  is the angular frequency, **J** is the impressed current distribution,  $\rho$  is the charge distribution satisfying the continuity equation  $\nabla \cdot \mathbf{J} = -\iota \omega \rho$ ,  $\varepsilon_0$  is the permittivity of free space, and  $\mu_0$  is the permeability of free space. The material parameters  $\varepsilon_r$  and  $\mu_r$  are the relative permittivity and permeability tensors with entries

$$\varepsilon_{r} = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix}, \quad \mu_{r} = \begin{pmatrix} \mu_{xx} & \mu_{xy} & \mu_{xz} \\ \mu_{yx} & \mu_{yy} & \mu_{yz} \\ \mu_{zx} & \mu_{zy} & \mu_{zz} \end{pmatrix}$$

In general, these tensors are functions in  $\mathbb{R}^3$ . Due to computational constraints, it is impossible to solve the full radiation problem; since we are usually concerned with the fields inside a compact subset of  $\mathbb{R}^3$ , we truncate the problem by introducing a condition at the boundary of the domain of interest that will emulate the Silver–Müller radiation conditions in (2.1). Here, the subset we consider will be the box  $\Omega = (-1, 1)^3$ . Absorbing boundary conditions have been well investigated for this truncation. In this paper, however, we opt for the perfectly matched layer approach [4,6].

Denote the thickness of the PML as  $\ell$ ; this corresponds to a physical domain  $[-1 + \ell, 1 - \ell]^3$ . Let us define the complex stretching functions  $s_{\xi}$  for  $\xi = x, y, z$  as

$$s_{\xi}(\xi) = 1 + \iota \sigma(\xi),$$

with  $\sigma \ge 0$  in the PML region and 0 everywhere else. We normally choose  $\sigma$  to be the ramp-like function

$$\sigma(\xi) = egin{cases} heta(\xi) = \left\{egin{array}{ll} heta(\xi-1-\xi)^2, & \xi\in [-1,-1+\ell] \ 0, & \xi\in [-1+\ell,1-\ell] \ heta(\xi-1+\ell)^2, & \xi\in [1-\ell,1] \end{array}
ight.$$

where  $\theta$  is a constant determined by the mesh size and frequency. Given these functions, the resulting PML tensors  $\tilde{\epsilon}_r$  and  $\tilde{\mu}_r$  take the form

$$\tilde{\varepsilon}_r = \begin{pmatrix} \varepsilon_{xx} \frac{s_ys_z}{s_x} & \varepsilon_{xy}s_z & \varepsilon_{xz}s_y \\ \varepsilon_{yx}s_z & \varepsilon_{yy} \frac{s_xs_z}{s_y} & \varepsilon_{yz}s_x \\ \varepsilon_{zx}s_y & \varepsilon_{zy}s_x & \varepsilon_{zz} \frac{s_xs_y}{s_z} \end{pmatrix}, \quad \tilde{\mu}_r = \begin{pmatrix} \mu_{xx} \frac{s_ys_z}{s_x} & \mu_{xy}s_z & \mu_{xz}s_y \\ \mu_{yx}s_z & \mu_{yy} \frac{s_xs_z}{s_y} & \mu_{yz}s_x \\ \mu_{zx}s_y & \mu_{zy}s_x & \mu_{zz} \frac{s_xs_y}{s_z} \end{pmatrix}.$$

Note that within the physical domain, the PML tensors reduce to the original material, i.e.  $\tilde{\varepsilon}_r = \varepsilon_r$  and  $\tilde{\mu}_r = \mu_r$  in  $[-1 + \ell, 1 - \ell]^3$ . On the outside of the PML, we artificially place a perfectly conducting surface to close the system. This is a widely accepted practice, as the reflections off the PEC are very small because of the exponential decay inside the PML.

In the 2D TE mode case, Maxwell's curl equations are reduced to

$$\begin{aligned} \frac{\partial E_y}{\partial x} &- \frac{\partial E_x}{\partial y} = -i\omega\mu_0\mu_{zz}H_z\\ \frac{\partial H_z}{\partial y} &= i\omega\varepsilon_0(\varepsilon_{xx}E_x + \varepsilon_{xy}E_y) + J_x\\ &- \frac{\partial H_z}{\partial x} = i\omega\varepsilon_0(\varepsilon_{yx}E_x + \varepsilon_{yy}E_y) + J_y\end{aligned}$$

Defining the partial derivatives in stretched coordinates exactly the same way as in the 3D case, we can follow the same process for deriving the material parameters. The resulting material properties are

$$\tilde{\varepsilon}_r = \begin{pmatrix} \varepsilon_{xx} \frac{s_y}{s_x} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \frac{s_x}{s_y} \end{pmatrix}, \quad \tilde{\mu}_r = s_x s_y \mu_{zz}.$$

### 2.2. Variational formulation

The truncated PML problem of Maxwell's equations is

$$\nabla \times \mathbf{E} = -\iota \omega \mu_0 \tilde{\mu}_r \mathbf{H}$$

$$\nabla \times \mathbf{H} = \iota \omega \varepsilon_0 \tilde{\varepsilon}_r \mathbf{E} + \mathbf{J}$$

$$\nabla \cdot (\varepsilon_0 \tilde{\varepsilon}_r \mathbf{E}) = \rho$$

$$\nabla \cdot (\mu_0 \tilde{\mu}_r \mathbf{H}) = 0 \quad \text{in } \Omega,$$

$$\hat{\mathbf{n}} \times \mathbf{E} = 0$$

$$\hat{\mathbf{n}} \times \mathbf{H} = 0 \quad \text{on } \partial \Omega,$$
(2.3)

where  $\hat{\mathbf{n}}$  is the unit normal vector on  $\partial \Omega$ . We can eliminate the magnetic field variable **H** and use the second-order equation

$$\nabla \times \tilde{\mu}_{r}^{-1} \nabla \times \mathbf{E} - \kappa^{2} \tilde{e}_{r} \mathbf{E} = -\iota \omega \mu_{0} \mathbf{J} \quad \text{in } \Omega,$$
(2.4)

$$\hat{\mathbf{n}} \times \mathbf{E} = \mathbf{0} \quad \text{on } \partial\Omega,$$
 (2.5)

where  $\kappa := \omega \sqrt{\mu_0 \varepsilon_0} = \frac{2\pi}{\lambda}$  is the wavenumber in free space. For a testing function  $\phi \in H_0(\text{curl}, \Omega)$ , where  $H_0(\text{curl}, \Omega) = \{\phi \in H(\text{curl}, \Omega) : \hat{\mathbf{n}} \times \phi(\mathbf{r}) = 0, \forall \mathbf{r} \in \partial \Omega\}$ , the weak form of the equation is

$$\left(\tilde{\mu}_{r}^{-1}\nabla\times\mathbf{E},\nabla\times\boldsymbol{\phi}\right)_{\Omega}-\kappa^{2}(\tilde{\varepsilon}_{r}\mathbf{E},\boldsymbol{\phi})_{\Omega}=-\iota\omega\mu_{0}(\mathbf{J},\boldsymbol{\phi})_{\Omega},$$
(2.6)

with the inner product  $(\mathbf{u}, \mathbf{v})_{\Omega} = \int_{\Omega} \mathbf{u} \cdot \bar{\mathbf{v}} d\Omega$ . Under certain conditions [16], there exists a unique solution to (2.6) in  $H_0(\text{curl}, \Omega)$ .

# 2.3. Vector finite elements

To discretize the variational formulation, we represent the domain with a triangular mesh in 2D and a tetrahedral mesh in 3D. Each mesh is constructed so that the element-to-wavelength ratio is kept constant over all frequencies; for a wavelength  $\lambda$ , we choose the mesh size h so that  $h \approx \lambda/10$ . To avoid the possible spurious solutions given by nodal basis functions, we employ the linear curl-conforming edge functions made famous by Nédélec [17]. These basis functions are divergence-free and have a constant tangential component along the edge on which they are defined.

Given a quasi-uniform tetrahedral mesh of  $\Omega$  with *N* interior edges (i.e. edges with at most one vertex lying on  $\partial \Omega$ ) and edge length bounded by *h*, we have the finite element approximation

$$\mathbf{E}_h = \sum_{i=1}^N c_i \boldsymbol{\psi}_i,$$

where  $\psi_i$  is the edge function defined on edge *i* and  $c_i \in \mathbb{C}$  are the undetermined coefficients. The standard Galerkin formulation yields the sparse linear system

$$Au = b, (2.7)$$

with the matrix and vector entries

$$\begin{split} A_{ij} &= \left(\tilde{\mu}_r^{-1} \nabla \times \psi_j, \nabla \times \psi_i\right)_{\Omega} - \kappa^2 (\tilde{\epsilon}_r \psi_j, \psi_i)_{\Omega}, \\ u_j &= c_j, \\ b_i &= -i \omega \mu_0 (\mathbf{J}, \psi_i)_{\Omega}. \end{split}$$

#### 3. Preconditioner for FEM

The linear system of equations in (2.7) is highly indefinite and ill-conditioned for high frequency problems. As mentioned earlier, the iterative solution of such a system is difficult with current preconditioning techniques. In this section, we detail the sweeping preconditioner for the finite element method, which will allow convergence of a Krylov subspace iterative solver in a small number of iterations.

#### 3.1. Mesh partitioning

The first step to setting up the preconditioner is to divide the mesh into layers or slabs. Consider the problem for a particular wavenumber  $\kappa$  such that the number  $K := \frac{\kappa}{\pi} = \frac{\omega \sqrt{\mu_0 \epsilon_0}}{\pi}$  is an integer; here, K is the width of the domain in wavelengths. Let us also assume for the sake of simplicity that the PML width is  $\ell = \lambda$ . We can then divide the domain into the subdomains  $\Omega_i$ , i = 1, ..., K as

$$\Omega_i = [-1, 1]^{d-1} \times [-1 + (i-1)\lambda, -1 + i\lambda), \text{ for } i = 1, \dots, K-1$$
  
$$\Omega_K = [-1, 1]^{d-1} \times [-1 + (K-1)\lambda, 1].$$

The partition occurs in the *y*-direction for 2D problems and in the *z*-direction for 3D problems. Clearly, we have  $\overline{\Omega} = \bigcup_{i=1}^{k} \Omega_i$ and  $\Omega_i \cap \Omega_i = \emptyset$  if  $i \neq j$ .

For a tetrahedral mesh  $\mathcal{T} = \{t_1, \ldots, t_{N_T}\}$  with edges denoted by  $e_j, j = 1, \ldots, N$ , we define  $\mathbf{v}_{j_k}$  for  $k = 0, \ldots, d$  to be the vertices of tetrahedron  $t_j$  in d dimensions and denote the centroid of  $t_j$  with  $\mathbf{c}_j = \frac{1}{d+1} \sum_{k=0}^d \mathbf{v}_{j_k}$ . Next, we define  $\mathcal{T}_i$  as the union of tetrahedra whose centroids are in  $\Omega_i$ , i.e.,

$$\mathcal{T}_i = \bigcup \{ t_j : \mathbf{c}_j \in \Omega_i \}.$$
(3.1)

Clearly,  $\overline{\Omega} = \bigcup_{i=1}^{K} \mathcal{T}_i$ . Fig. 1 shows an unstructured triangular mesh and tetrahedral mesh partitioned into eight layers, with a different color for each layer; in general, the boundary of each layer is not a smooth surface. (See Fig. 2)

Once the task of partitioning the mesh is completed, we can construct integer sets  $\mathcal{E}_i$ , which will point to which degrees of freedom are associated with layer  $\Omega_i$ . This organizational structure is necessary to obtain the block  $LDL^t$  factorization in the next section. At first, this may seem trivially similar to the algorithms in [8,9], but a conflict occurs at the boundary between two layers; specifically, when a simplex in  $\mathcal{T}_i$  and a simplex in  $\mathcal{T}_{i-1}$  share an edge. This can be avoided by always choosing to associate boundary edges with the upper layer. If we have  $\partial \mathcal{T}_i$  as the boundary and  $\mathcal{T}_i^{\text{int}}$  as the interior of the domain defined by elements in  $\mathcal{T}_i$ , such that  $\mathcal{T}_i = \partial \mathcal{T}_i \cup \mathcal{T}_i^{\text{int}}$ , we can categorize the edges  $e_j$ ,  $j = 1, \ldots, N$  in the following manner:

$$\mathcal{E}_1 = \{j : e_j \text{ is an interior edge of } T_1\}$$
  
$$\mathcal{E}_i = \{j : e_j \text{ is an interior edge of } T_i \text{ or } e_j \in \partial \mathcal{T}_i \cap \partial \mathcal{T}_{i-1}\} \text{ for } i = 2, \dots, K$$

# 3.2. Sweeping factorization

With the degrees of freedom in each layer defined by the integer sets  $\mathcal{E}_i$  for i = 1, ..., K, we can rewrite the linear system in (2.7) in block tridiagonal form. Using MATLAB-style notation for indexing matrices and vectors, we can reorder the system

$$\begin{pmatrix} A(\mathcal{E}_{1},\mathcal{E}_{1}) & A(\mathcal{E}_{1},\mathcal{E}_{2}) & & \\ A(\mathcal{E}_{2},\mathcal{E}_{1}) & A(\mathcal{E}_{2},\mathcal{E}_{2}) & \ddots & \\ & \ddots & \ddots & A(\mathcal{E}_{K-1},\mathcal{E}_{K}) \\ & & A(\mathcal{E}_{K},\mathcal{E}_{K-1}) & A(\mathcal{E}_{K},\mathcal{E}_{K}) \end{pmatrix} \begin{pmatrix} u(\mathcal{E}_{1}) \\ u(\mathcal{E}_{2}) \\ \vdots \\ u(\mathcal{E}_{K}) \end{pmatrix} = \begin{pmatrix} b(\mathcal{E}_{1}) \\ b(\mathcal{E}_{2}) \\ \vdots \\ b(\mathcal{E}_{K}) \end{pmatrix},$$
(3.2)

where  $u(\mathcal{E}_i)$  are the unknown coefficients associated with edges in layer  $i, b(\mathcal{E}_i)$  is the right hand side computed from basis functions defined on edges in layer i, and  $A(\mathcal{E}_i, \mathcal{E}_j)$  are the blocks of the stiffness matrix corresponding to the degrees of freedom in layer i and layer j. This permits the block  $LDL^t$  factorization

$$L_{1} \dots L_{K-1} \begin{pmatrix} S_{1} & & \\ & S_{2} & \\ & & \ddots & \\ & & & \ddots & \\ & & & & S_{K} \end{pmatrix} L_{K-1}^{t} \dots L_{1}^{t},$$
(3.3)

where the Schur complement matrices take the form  $S_1 = A(\mathcal{E}_1, \mathcal{E}_1), S_i = A(\mathcal{E}_i, \mathcal{E}_i) - A(\mathcal{E}_i, \mathcal{E}_{i-1})S_{i-1}^{-1}A(\mathcal{E}_{i-1}, \mathcal{E}_i)$  for i = 2, ..., K. Define the index sets  $\mathcal{P}_i$  for i = 1, ..., K as

$$\mathcal{P}_i = \left\{\sum_{s=1}^{i-1} N_s + 1, \dots, \sum_{s=1}^{i} N_s\right\},\,$$

where  $N_s = |\mathcal{E}_s|$  is the cardinality of set  $\mathcal{E}_s$ . The block lower triangular matrices  $L_i$  are then

 $L_i(\mathcal{P}_{i+1},\mathcal{P}_i) = A(\mathcal{E}_{i+1},\mathcal{E}_i)S_i^{-1}, \quad L_i(\mathcal{P}_i,\mathcal{P}_i) = I(1 \leqslant i \leqslant K), \quad \text{and zero otherwise.}$ 

Explicitly inverting the factorization yields the solution



Fig. 1. Left: triangular mesh partitioned into eight layers. Right: tetrahedral mesh partitioned into eight layers.

$$\begin{pmatrix} u(\mathcal{E}_{1}) \\ u(\mathcal{E}_{2}) \\ \vdots \\ u(\mathcal{E}_{K}) \end{pmatrix} = (L_{1}^{t})^{-1} \cdots (L_{K-1}^{t})^{-1} \begin{pmatrix} S_{1}^{-1} & & \\ & S_{2}^{-1} & \\ & & \ddots \\ & & & S_{K}^{-1} \end{pmatrix} L_{K-1}^{-1} \cdots L_{1}^{-1} \begin{pmatrix} b(\mathcal{E}_{1})b(\mathcal{E}_{2}) \\ \vdots \\ b(\mathcal{E}_{K}) \end{pmatrix}.$$
(3.4)

We can summarize the factorization and inversion process in the following algorithms.

Algorithm 3.1. Construct the sweeping factorization of A

1: Set  $S_1 = A(\mathcal{E}_1, \mathcal{E}_1)$  and compute  $S_1^{-1}$ . 2: for i = 2, ..., K do 3: Set  $S_i = A(\mathcal{E}_i, \mathcal{E}_i) - A(\mathcal{E}_i, \mathcal{E}_{i-1})S_{i-1}^{-1}A(\mathcal{E}_{i-1}, \mathcal{E}_i)$  and compute  $S_i^{-1}$ . 4: end for

# **Algorithm 3.2.** Apply the inverse to get $u = A^{-1}b$

```
1: Set u(\mathcal{E}_i) = b(\mathcal{E}_i), for i = 1, ..., K.
2: Compute u(\mathcal{E}_2) = u(\mathcal{E}_2) - A(\mathcal{E}_2, \mathcal{E}_1)S_1^{-1}u(\mathcal{E}_1).
3: for i = 2, ..., K - 1 do
           Compute u(\mathcal{E}_{i+1}) = u(\mathcal{E}_{i+1}) - A(\mathcal{E}_{i+1}, \mathcal{E}_i)S_i^{-1}u(\mathcal{E}_i).
4:
5: end for
6: Compute u(\mathcal{E}_1) = S_1^{-1}u(\mathcal{E}_1).
7: for i = 2,...,K do
           Compute u(\mathcal{E}_i) = S_i^{-1} u(\mathcal{E}_i).
8:
9: end for
10: for i = K - 1, ..., 2 do
             Compute u(\mathcal{E}_i) = u(\mathcal{E}_i) - S_i^{-1}A(\mathcal{E}_i, \mathcal{E}_{i+1})u(\mathcal{E}_{i+1}).
11:
12:end for
13:Compute u(\mathcal{E}_1) = u(\mathcal{E}_1) - S_1^{-1}A(\mathcal{E}_1, \mathcal{E}_2)u(\mathcal{E}_2).
```

The main computational cost comes from inverting the Schur complement blocks  $S_i$ ; note that for i = 2, ..., K, these matrices are dense, since each  $S_i$  is dependent on the inversion of  $S_{i-1}$ . For Cartesian finite difference grids, the cost of computing the inversion of the above factorization has been shown to be  $O(N^2)$  in 2D and  $O(N^{7/3})$  in 3D [8]. We can make a similar argument for finite element meshes, as the number of edges varies approximately linearly with the number of simplex elements. Our aim is to reduce this computational time to linear or almost linear complexity.

# 3.3. Moving PML method

An important physical observation can be made about each matrix  $S_i$ . Specifically, let us restrict the full problem to the first *m* layers, for *m* < *K*; that is, instead of the whole system in (3.2), consider the smaller system of equations

$(A(\mathcal{E}_1,\mathcal{E}_1))$	$A(\mathcal{E}_1,\mathcal{E}_2)$		)	۱.	$u(\mathcal{E}_1)$		$\int b(\mathcal{E}_1) \setminus$	
$A(\mathcal{E}_2,\mathcal{E}_1)$	$A(\mathcal{E}_2,\mathcal{E}_2)$	·			$u(\mathcal{E}_2)$		$b(\mathcal{E}_2)$	
	·	·	$A(\mathcal{E}_{m-1},\mathcal{E}_m)$		÷	=	÷	•
		$A(\mathcal{E}_m, \mathcal{E}_{m-1})$	$A(\mathcal{E}_m, \mathcal{E}_m)$ /	) '	$\langle u(\mathcal{E}_m) \rangle$		$b(\mathcal{E}_m)/$	

This system corresponds to the discretization of the semi-infinite half-space Maxwell problem with a PEC boundary condition on the boundary of  $\bigcup_{i=1}^{m} T_i$ , i.e.

3775

$$\nabla \times \tilde{\mu}_r^{-1} \nabla \times \mathbf{E} - \kappa^2 \tilde{\varepsilon}_r \mathbf{E} = -\iota \omega \mu_0 \mathbf{J} \quad \text{in } \operatorname{int} \left( \bigcup_{i=1}^m \mathcal{T}_i \right),$$
  
$$\hat{\mathbf{n}} \times \mathbf{E} = \mathbf{0} \quad \text{on } \partial \left( \bigcup_{i=1}^m \mathcal{T}_i \right).$$
(3.5)

If we invert the upper m blocks, we obtain

$$\begin{pmatrix} u(\mathcal{E}_{1}) \\ \vdots \\ u(\mathcal{E}_{m}) \end{pmatrix} = (L_{1}^{t})^{-1} \dots (L_{m-1}^{t})^{-1} \begin{pmatrix} S_{1}^{-1} & & \\ & \ddots & \\ & & S_{m}^{-1} \end{pmatrix} L_{m-1}^{-1} \dots L_{1}^{-1} \begin{pmatrix} b(\mathcal{E}_{1}) \\ \vdots \\ b(\mathcal{E}_{m}) \end{pmatrix}.$$
(3.6)

However, due to the structure of  $L_i$  for i = 1, ..., m, it is noticed that the  $S_m^{-1}$  is unaffected by the left and right operators in (3.6), i.e. the (m,m)-th block in the right hand side is exactly  $S_m^{-1}$ . Based on this fact, we can conclude that  $S_m^{-1}$  is the discrete Green's function for degrees of freedom in the *m*-th layer for (3.5); solving the half-space problem above implicitly constructs an operator for  $S_m^{-1}$ . In [8], a proof was given that this matrix is low rank in the 2D case, and that hierarchical matrices are efficient in representing  $S_m^{-1}$ . In [9], a different approach called the *moving PML method* was introduced as an alternative to approximate  $S_m^{-1}$  efficiently. In this paper, we choose the moving PML approach, as it can be readily applied to general finite element discretizations.

The central idea behind the moving PML method is to take advantage of the radiation boundary condition. Because each  $S_m$  is a Schur complement for the degrees of freedom in layer m, its inverse only acts on quantities defined in this layer. Therefore, as an operator,  $S_m^{-1}$  only needs to be accurate in this region. Recalling that the PML boundary condition is used to emulate the Silver–Müller radiation conditions, the half-space problem in (3.5) can be truncated even further by pushing the PML up to the domain of interest; that is, instead of solving a problem with layers  $1, \ldots, m$ , we can solve an approximate problem with layers m - 1 and m by placing the PML in layer m - 1. As we will see in the numerical results, solving this subproblem will serve as a good approximation to the operator  $S_m^{-1}$ .

When compared to direct inversion to get  $S_i^{-1}$ , the moving PML method has a huge computational advantage. Recall that  $N_i = |\mathcal{E}_i|$  is the number of degrees of freedom in layer *i*. Inverting each  $N_i \times N_i$  matrix  $S_i$  would take  $O(N_i^3)$  steps because  $S_i$  is dense for i = 2, ..., K. On the other hand, the truncated half-space problem is a sparse system roughly of size  $2N_i \times 2N_i$ . In 2D, this corresponds to a quasi-1D problem which can be factorized efficiently using an  $O(N_i)LU$  decomposition that orders the short dimension first. In 3D, it corresponds to a quasi-2D problem that can be factorized in  $O(N_i^{3/2})$  time using the multi-frontal method with nested dissection.

To make things more precise, the process of approximating  $S_i^{-1}$  for i = 2, ..., K in operator form is as follows. Consider the shifted stretching function for subdomain  $\Omega_i$ ,

$$s_{\xi,i}(\xi) = 1 + \iota \sigma_i(\xi),$$

where  $\sigma_i$  is the ramp-like function

$$\sigma_i(\xi) = \begin{cases} \theta \Big(\frac{-1+(i-1)\ell-\xi}{\ell}\Big)^2, & \xi \in [-1+(i-2)\ell, -1+(i-1)\ell] \\ 0, & \xi \in [-1+(i-1)\ell, 1-\ell] \\ \theta \big(\frac{\xi-1+\ell}{\ell}\big)^2, & \xi \in [1-\ell, 1] \end{cases}$$

The truncated half-space problem for layer *i* is then

$$\nabla \times \tilde{\mu}_{r,i}^{-1} \nabla \times \mathbf{E} - \kappa^2 \tilde{\varepsilon}_{r,i} \mathbf{E} = -\iota \omega \mu_0 \mathbf{J} \quad \text{in int}(\mathcal{T}_{i-1} \cup \mathcal{T}_i)$$
  
$$\hat{\mathbf{n}} \times \mathbf{E} = \mathbf{0} \quad \text{on } \partial(\mathcal{T}_{i-1} \cup \mathcal{T}_i), \qquad (3.7)$$

where the material parameters in 2D are

$$\tilde{\varepsilon}_{r,i} = \begin{pmatrix} \varepsilon_{xx} \frac{s_{y,i}}{s_x} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \frac{s_x}{s_{y,i}} \end{pmatrix}, \quad \tilde{\mu}_{r,i} = s_x s_{y,i} \mu_{zz},$$

and the tensors in 3D are

$$\tilde{\varepsilon}_{r,i} = \begin{pmatrix} \varepsilon_{xx} \frac{s_y s_{z,i}}{s_x} & \varepsilon_{xy} s_{z,i} & \varepsilon_{xz} s_y \\ \varepsilon_{yx} s_{z,i} & \varepsilon_{yy} \frac{s_x s_{z,i}}{s_y} & \varepsilon_{yz} s_x \\ \varepsilon_{zx} s_y & \varepsilon_{zy} s_x & \varepsilon_{zz} \frac{s_x s_y}{s_{z,i}} \end{pmatrix}, \quad \tilde{\mu}_{r,i} = \begin{pmatrix} \mu_{xx} \frac{s_y s_{z,i}}{s_x} & \mu_{xy} s_{z,i} & \mu_{xz} s_y \\ \mu_{yx} s_{z,i} & \mu_{yy} \frac{s_x s_{z,i}}{s_y} & \mu_{yz} s_x \\ \mu_{zx} s_y & \mu_{zy} s_x & \mu_{zz} \frac{s_x s_y}{s_{z,i}} \end{pmatrix}.$$
(3.8)

Clearly, subproblem (3.7) requires only the first two layers of the shifted PML function. Denote the stiffness matrix resulting from the discretization of (3.7) as  $H_i$ ; it is crucial that the degrees of freedom in the local subproblem maintain the same

order as in (3.2). Using the multifrontal method with nested dissection, we can construct the optimal sparse *LU* factorization of  $H_i$  and are able to apply the inverse operator  $H_i^{-1}$  efficiently. Now consider the vector  $v \in \mathbb{C}^{N_i}$ , where  $N_i$  is the number of degrees of freedom in layer *i*. If we concatenate a vector of zeros  $0 \in \mathbb{C}^{N_{i-1}}$  with v and apply  $H_i^{-1}$ , the result is

$$H_i^{-1}\begin{pmatrix}0\\\nu\end{pmatrix} = \begin{pmatrix}w_1\\w_2\end{pmatrix}.$$
(3.9)

for vectors  $w_1 \in \mathbb{C}^{N_{i-1}}, w_2 \in \mathbb{C}^{N_i}$ . We can then extract the vector  $w_2$  from the right hand side of (3.9) to obtain the approximation of  $S_i^{-1}v$ ; we define the operator which performs this concatenation/extraction process to be  $\widetilde{S}_i^{-1} : \mathbb{C}^{N_i} \to \mathbb{C}^{N_i}$ .

We are now ready to present the setup and application algorithms of the sweeping preconditioner with the moving PML method; we modify Algorithms 3.1 and 3.2 appropriately with the new operator  $\tilde{S}_i^{-1}$ .

Algorithm 3.3. Setup the sweeping preconditioner of A

1: Let  $H_1 = A(\mathcal{E}_1, \mathcal{E}_1)$ ; construct the sparse *LU* factorization of  $H_1$ .

3: Let  $H_i$  be the stiffness matrix of (3.7). Construct the optimal sparse LU factorization of  $H_i$  using the multifrontal method with nested dissection.

The setup of the preconditioner requires the solution of each subproblem in (3.7). In the 2D case, each subproblem has  $O(\sqrt{N})$  degrees of freedom. The multifrontal method constructs the solution of the quasi-1D problem with linear complexity, so each subproblem can be solved in  $O(\sqrt{N})$  time. Since there are  $O(\sqrt{N})$  subproblems to be solved, the total setup time in 2D is O(N). In the 3D case, the estimate is slightly worse. Each subproblem in 3D contains  $O(N^{1/3}) \times O(N^{1/3}) = O(N^{2/3})$  degrees of freedom; consequently, the multifrontal method can solve the quasi-2D problem in  $O((N^{2/3})^{3/2}) = O(N)$  time. Since there are  $O(N^{1/3})$  layers, the total complexity to setup the 3D preconditioner is  $O(N^{4/3})$ .

**Algorithm 3.4.** Apply the approximate inverse to *b* to get  $u \approx A^{-1}b$ 

1: Set  $u(\mathcal{E}_i) = b(\mathcal{E}_i)$ , for i = 1, ..., K. 2: Compute  $u(\mathcal{E}_2) = u(\mathcal{E}_2) - A(\mathcal{E}_2, \mathcal{E}_1)H_1^{-1}u(\mathcal{E}_1)$ . 3: for i = 2, ..., K - 1 do Compute  $u(\mathcal{E}_{i+1}) = u(\mathcal{E}_{i+1}) - A(\mathcal{E}_{i+1}, \mathcal{E}_i)\widetilde{S}_i^{-1}u(\mathcal{E}_i)$ , where the operator  $\widetilde{S}_i^{-1}$  is described above. 4: 5: end for 6: Compute  $u(\mathcal{E}_1) = H_1^{-1}u(\mathcal{E}_1)$ . 7: **for** *i* = 2,...,*K* **do** Compute  $u(\mathcal{E}_i) = \widetilde{S}_i^{-1} u(\mathcal{E}_i)$ . 8: 9: end for 10: **for** $i = K - 1, \dots, 2$  **do** Compute  $u(\mathcal{E}_i) = u(\mathcal{E}_i) - \widetilde{S}_i^{-1}A(\mathcal{E}_i, \mathcal{E}_{i+1})u(\mathcal{E}_{i+1}).$ 11: 12: end for 13: Compute  $u(\mathcal{E}_1) = u(\mathcal{E}_1) - H_1^{-1}A(\mathcal{E}_1, \mathcal{E}_2)u(\mathcal{E}_2)$ .

The main cost in Algorithm 3.4 is the application of  $\tilde{S}_{1}^{-1}$ . In 2D, the cost of applying the inverse of the sparse *LU* factorization is linear; for each layer, this amounts to  $O(\sqrt{N})$  time. The computation is done  $O(\sqrt{N})$  times, which results in a total complexity of O(N). In 3D, applying each inverse can be done in logarithmic linear time, i.e.  $O(N^{2/3}\log N^{2/3})$ . With  $O(N^{1/3})$  layers, this results in a total complexity of  $O(N \log N)$ .

Algorithms 3.3 and 3.4 define an inverse operator  $M^{-1}$ , which is an approximation of  $A^{-1}$ . For stability reasons, it is more prudent to construct the approximate sweeping factorization for the equation

$$\nabla \times \tilde{\mu}_r^{-1} \nabla \times \mathbf{E} - (\kappa + \iota \alpha)^2 \tilde{\varepsilon}_r \mathbf{E} = -\iota \omega \mu_0 \mathbf{J}, \tag{3.10}$$

where  $\alpha$  is a positive damping constant of O(1). This approach is different from the shifted Laplacian preconditioner, in which the damping constant is proportional to frequency. With  $M_{\alpha}^{-1}$  being the approximate inverse operator for the discretization of (3.10), we can solve the preconditioned linear system

$$M_{\alpha}^{-1}Au = M_{\alpha}^{-1}b.$$

using a Krylov subspace iterative solver. As we will show in the numerical results, the rate of convergence of the iterative solver will be either independent or logarithmically dependent on frequency, with a low number of iterations.

A few remarks must be made about practical issues with the algorithm. First, we have presented the moving PML method in the *z*-direction for 3D; this choice is arbitrary, as we can also partition the mesh into slabs orthogonal to the *x* or *y* axes. For these cases, the appropriate PML functions must be shifted for the material tensors in (3.8). Second, we have defined each subdomain to have the same thickness as the PML; this is not a restriction, as one could configure the subdomains so that each slab is of a different thickness. This is particularly useful when computing on adaptive or locally refined meshes. In addition, the PML used to back each slab does not need to coincide with the adjacent slab; we choose this simplification for ease of implementation. Finally, we would like to note that other absorbing boundary conditions (ABC) can be utilized in place of the PML. The use of an ABC would reduce the memory and computational time, as each subproblem would not need an extra buffer layer.

# 4. Numerical results in 2D

In the 2D case, we present several numerical results to support our claims for the accuracy and linear complexity of the sweeping preconditioner. All of the 2D algorithms are implemented in sequential C++ code on a server equipped with Intel Xeon E7420 2.13 GHz processors. For the multifrontal method used to solve each subproblem, we employed the sequential version of MUMPS [1]. For the iterative solver, we have chosen GMRES iteration with a residual tolerance set to  $10^{-3}$ .

### 4.1. Cartesian PML

For Cartesian PMLs, we have chosen a few examples of heterogeneous media which are of importance to the optics and photonics community:

1. A converging lens profile. Here, we consider the isotropic, heterogeneous material

$$\varepsilon_r = \left(1 + e^{-30(x^2 + y^2)}\right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mu_r = (1 + e^{-30(x^2 + y^2)}).$$

At the center of the lens, the wavespeed is  $\frac{1}{2}c$ , where  $c = \frac{1}{\sqrt{\mu_0 c_0}}$  is the speed of light in free space.

2. A periodic medium. We use the 2D function

$$f(x,y) = 1 + \frac{1}{4}\cos\left(20\left(\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}}\right)\right) + \frac{1}{4}\cos\left(20\left(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}}\right)\right)$$
(4.1)

to form the isotropic material

$$\varepsilon_r = \sqrt{f(x,y)} \begin{pmatrix} 1 & 0\\ 0 & 1 \end{pmatrix}, \quad \mu_r = \sqrt{f(x,y)}. \tag{4.2}$$

In these examples, the current **J** is a solenoidal vector field in the x-y plane derived from a Gaussian point source oriented in the z-direction, i.e.

$$\mathbf{J}(\mathbf{x},\mathbf{y}) = \nabla \times \left( \hat{\mathbf{z}} e^{-\kappa^2 (\mathbf{x}^2 + (\mathbf{y} - \mathbf{0.5})^2)} \right).$$
(4.3)

The preconditioner is constructed with  $\alpha = 1$  and PML width  $\ell = 2\lambda$ . For each experiment, we fix the domain  $[-1,1]^2$  while simultaneously increasing the wavenumber  $\kappa$ , keeping the same resolution for elements per wavelength; we list the width of the problem in wavelengths  $K := \frac{\kappa}{\pi} = \frac{\omega \sqrt{\mu \epsilon_0}}{\pi}$ , the number of degrees of freedom *N*, the preconditioner setup time  $T_{\text{setup}}$  in seconds, the iterative solver time  $T_{\text{solve}}$  in seconds, and the number of iterations necessary for convergence  $N_{\text{iter}}$ .

From the table in Fig. 3, it is observed that when *K* doubles, the number of degrees of freedom increases by a factor of 4. At the same time,  $T_{setup}$  also increases approximately by a factor of 4, which shows the linear complexity of Algorithm 3.3. The time per iteration  $\frac{T_{setup}}{N_{Rer}}$  also grows roughly by a factor of 4; thus, we can infer that the application Algorithm 3.4 is also O(N). As the number of iterations either remains constant or grows very weakly with frequency, the entire solver has O(N) complexity. The complexity graphs in Fig. 4 support these claims.

# 4.2. Cylindrical PML

The sweeping preconditioner can also be used with a cylindrical PML; in this case, the computational domain is a circle of radius 1. Instead of partitioning the domain into equally sized horizontal or vertical layers, a series of concentric shells are



**Fig. 2.** Material  $\varepsilon_r$  and  $\mu_r$  for the 2D converging lens (left) and periodic medium (right).



Fig. 3. Real part of the magnetic field H<sub>z</sub> at K = 64 with computational results for the converging lens (left) and periodic medium (right).



Fig. 4. The complexity graphs for setup time of the preconditioner (left) and time per iteration of the iterative method (right) in the 2D case.



Fig. 5. Layers of the circular domain for the cylindrical PML.



**Fig. 6.** Real part of the scattered field  $H_z$  at K = 64 for the PEC cylinder (left) and real part of the total field  $E_x$  at K = 64 for the transformation optics cloak (right), with computational results.

introduced. This organization is natural because the cylindrical PML is a shell surrounding the domain; the sweeping direction is oriented along the radial direction, towards the center. Fig. 5 illustrates the layer structure for the circle mesh. (See Fig. 6 and 7)

The following are standard examples in the high-frequency scattering and metamaterial communities:

- 1. A cylindrical PEC scatterer. Here, we have a PEC cylinder embedded in free space with a radius of 0.25.
- 2. A transformation optics cloaking device. We consider the cylindrical cloak derived from coordinate transformations [18] with singular parameters near the inner radius of the cloak; inside the cloak lies a PEC. The material is characterized by the relative parameters in cylindrical coordinates ( $\rho$ ,  $\theta$ , z):

$$egin{aligned} arepsilon_{
ho
ho} &= \mu_{
ho
ho} = rac{
ho-a}{
ho} \ arepsilon_{
ho heta} &= \mu_{ heta heta} = rac{
ho}{
ho-a} \ arepsilon_{
m zz} &= \mu_{
m zz} = rac{
ho-a}{
ho} \left(rac{b}{b-a}
ight)^2 \end{aligned}$$

where *a* and *b* are the inner and outer radii of the cloak, respectively. For our experiments, we choose a = 0.25 and b = 0.5. This example is particularly interesting for a few reasons: the medium is discontinuous over the boundary of the cloaking shell, and its material tensor in Cartesian coordinates is anisotropic with off-diagonal entries.

Instead of a current source, we choose a plane wave as the incident field. This requires us to use the scattered field formulation; although the right hand side is altered slightly, this does not change the construction of the stiffness matrices or preconditioner. Here, the plane wave is

$$\mathbf{E}_{\rm inc} = \hat{\mathbf{y}} e^{-i\kappa x} \tag{4.4}$$

The results for the cylindrical PML examples also show the linear complexity of the 2D algorithm. In this instance, every time the frequency is doubled, the radius of the domain in terms of wavelength is multiplied by a factor of 2; this yields an increase in the degrees of freedom by a factor of 4. For the PEC scatterer, we see that the setup time grows roughly by a factor of 4, and the application time also increases by a factor of 4, which is consistent with the O(N) complexity estimate. For the cloaking device, however, we observe that the setup and solve times increase by a factor of 4.2 instead, implying an  $O(N \log N)$  complexity. The reason this example has a less optimal complexity result is because of the dense mesh used to discretize the cloaking layer. Because the cloaking device relies on transformation optics, the oscillations are condensed inside the outer shell; thus, the mesh must be refined to keep the same dispersion relationship. Each subproblem on the cloaking shell results in a thicker 2D problem instead of a quasi-1D strip, resulting in the increase in computational time. The added DOFs also increase the memory requirements for each problem, limiting the largest domain to K = 128.

# 5. Numerical results in 3D

In the 3D case, we present a few examples to show the  $O(N^{4/3})$  complexity of the sweeping preconditioner. The 3D code is implemented in sequential C++ on a server equipped with 2.2 GHz AMD Opteron 6174 processors. Once again, we choose the sequential version of MUMPS for the multifrontal method and GMRES iteration with a residual tolerance set to  $10^{-3}$ . We test the sweeping preconditioner on the following 3D media:

1. A converging lens profile. Here, we consider the isotropic, heterogeneous material

$$\varepsilon_r = \frac{3}{4} \left( 1 - \frac{1}{2} e^{-8(x^2 + y^2 + z^2)} \right)^{-1} \mathbf{I} \quad \mu_r = \frac{3}{4} \left( 1 - \frac{1}{2} e^{-8(x^2 + y^2 + z^2)} \right)^{-1} \mathbf{I}$$

where **I** is the  $3 \times 3$  identity matrix.

2. A periodic medium. Using the 2D function

$$f(x,y) = 1 + \frac{1}{4}\cos\left(20\left(\frac{x}{\sqrt{2}} + \frac{y}{\sqrt{2}}\right)\right) + \frac{1}{4}\cos\left(20\left(\frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}}\right)\right),$$
(5.1)

we can form the oscillatory medium

$$\varepsilon_r = \sqrt{f(x,y)}\mathbf{I}, \quad \mu_r = \sqrt{f(x,y)}\mathbf{I}.$$
 (5.2)

Note that the function is invariant in the *z* direction; we should see caustics similar to the 2D case.

The current source in these examples is a Gaussian point source oriented in the z-direction, i.e.

$$\mathbf{J}(\mathbf{x},\mathbf{y},\mathbf{z}) = \hat{\mathbf{z}} \mathbf{e}^{\frac{\kappa^2}{\pi^2} (\kappa^2 + (\mathbf{y} - \mathbf{0}.5)^2 + \mathbf{z}^2)}.$$
(4.3)

Here, the preconditioner is constructed with  $\alpha = 1$  and PML width  $\ell \approx \lambda$ . In contrast to the 2D case, where  $\ell = 2\lambda$ , we decrease the thickness of each layer to make the 3D algorithm as efficient as possible. Although having thinner slabs increases the number of iterations, we observe that the GMRES convergence does not deteriorate significantly for the problems tested.

The complexity of the 3D algorithm is illustrated in Fig. 8. Keeping the element-to-wavelength ratio constant while doubling the frequency forces the total degrees of freedom to increase by a factor of 8. At the same time, the  $O(N^{4/3})$  complexity estimate implies that the setup time should increase by a factor of  $8^{4/3} = 16$ . However, we observe an increase in setup time by a factor of 11 or 13. The cause of this improvement is clear; in practice, the number of subdomains is closer to  $O(N^{1/5})$  rather than  $O(N^{1/3})$ . Thus, we observe complexity that grows as  $O(N^{6/5})$  instead. For the application of the preconditioner, the  $O(N \log N)$  estimate implies that the solve time should increase roughly by a factor of 10 given the same number of iterations. Fig. 9 supports our claims.

We finish the numerical results section with a few remarks. When comparing the preconditioner of [9] to this paper, there are two main challenges that need to be addressed. First, the vector nature of Maxwell's equations makes it more difficult to reproduce the Green's function in each half-space. This is evident through observing the free space dyadic Green's function, which contains the Hessian of the Helmholtz Green's function. Second, the boundaries between two layers are not flat

P. Tsuji et al./Journal of Computational Physics 231 (2012) 3770-3783



**Fig. 7.** Slice plots of  $\varepsilon_r$  and  $\mu_r$  for the 3D converging lens (left) and periodic medium (right).



Fig. 8. Real part of the field E<sub>z</sub> in the x-y plane at K = 20 with computational results for the converging lens (left) and the periodic medium (right).



Fig. 9. The complexity graphs for setup time of the preconditioner and time per iteration of the iterative method in the 3D case.

planes. The half-space approximation still applies because the algebraic structure of the  $LDL^t$  factorization remains the same; however, the convergence is not as fast as on the uniform grid because the half-space Green's function for a rough surface is harder to compute accurately. To combat these difficulties, we have decreased the damping parameter  $\alpha$  and have refined the mesh slightly.

#### 6. Conclusions and future work

The sweeping preconditioner with moving PMLs originally developed by Engquist and Ying in [8,9] has been presented for finite elements on unstructured meshes, in the context of the time-harmonic Maxwell's equations. The algorithm has been tested on a variety of problems, including a cloaking example with anisotropic and discontinuous material parameters. It has been shown that the preconditioner is essentially unaffected by frequency, requiring only a small number of GMRES iterations which remains almost constant as wavenumber increases. This produces a linear complexity solver in 2D and an almost linear complexity solver in 3D.

A few questions still remain to be answered. First, we have considered low-order finite elements that are standard in computational electromagnetics. In many applications, however, higher-order basis functions are desirable to minimize the dispersion error and drastically reduce the number of degrees of freedom. It would be interesting to see the performance of the algorithm in these situations. Second, there are other physical applications in which frequency domain wave equations on unbounded regions arise, most notably in linear elasticity. In seismic imaging, many researchers resort to solving the scalar Helmholtz equation; due to computational constraints, they are unable to solve the full elastic wave equation. We believe that the sweeping preconditioner would allow the efficient solution of such problems. Finally, improvements can be made for the algorithm in 3D. Instead of using the multifrontal method for each subproblem, one can reuse the sweeping preconditioner within each layer recursively; this would reduce the complexity estimate from  $O(N^{4/3})$  to  $O(N \log N)$ , producing a logarithmic linear complexity solver in 3D. It would be important to compare the accuracy and complexity tradeoff between the sparse direct solver and the recursive sweeping preconditioner.

#### References

- P.R. Amestoy, I.S. Duff, J.Y. L'Excellent, J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 15–41.
- [2] I.M. Babuska, S.A. Sauter, Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?, SIAM Rev 42 (3) (2000) 451–484.
- [3] A. Bayliss, C.I. Goldstein, E. Turkel, An iterative method for the Helmholtz equation, J. Comput. Phys. 49 (3) (1983) 443-457.
- [4] J.P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (2) (1994) 185-200.
- [5] Y. Boubendir, X. Antoine, C. Geuzaine, A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation, J. Comput. Phys. 231 (12) (2012) 262–280.
- [6] W.C. Chew, W.H. Weedon, A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates, Microwave Opt. Tech. Lett. 7 (13) (1994) 599–604.
- [7] F. Collino, S. Ghanemi, P. Joly, Domain decomposition method for harmonic wave propagation: a general presentation, Comput. Methods Appl. Mech. Eng. 184 (2-4) (2000) 171-211.
- [8] B. Engquist, L. Ying, Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation, Commun. Pure Appl. Math. 64 (2011) 697–735.
- [9] B. Engquist, L. Ying, Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers, Multiscale Model. Simul. 9 (2011) 686– 710.
- [10] Y.A. Erlangga, Advances in iterative methods and preconditioners for the Helmholtz equation, Arch. Comput. Methods Eng. 15 (1) (2008) 37-66.
- [11] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, On a class of preconditioners for solving the Helmholtz equation, Appl. Numer. Math. 50 (3-4) (2004) 409–425.
   [12] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, A comparison of multigrid and incomplete LU shifted Laplace preconditioners for the inhomogeneous Helmholtz equation, Appl. Numer. Math. 56 (5) (2006) 648–666.
- [13] M.J. Gander, F. Magoulès, F. Nataf, Optimized Schwarz methods without overlap for the Helmholtz equation, SIAM J. Sci. Comput. 24 (1) (2002) 38–60.
   electronic.
- [14] J. Jin, The Finite Element Method in Electromagnetics, Wiley-IEEE Press, 2002.
- [15] P.-L. Lions, On the Schwarz alternating method. III. A variant for nonoverlapping subdomains, in: Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1990, pp. 202–223. Houston, TX, 1989.
- [16] P. Monk, Finite element methods for Maxwell's equations, Oxford University Press, 2003.
- [17] J.C. Nédélec, Mixed finite elements in  $\mathbb{R}^3$ , Numer. Math. 35 (3) (1980) 315–341.
- [18] J.B. Pendry, D. Schurig, D.R. Smith, Controlling electromagnetic fields, Science 312 (5781) (2006) 1780–1782.