Front. Math. China 2012, 7(2): 347–363
DOI 10.1007/s11464-012-0191-8

RESEARCH ARTICLE

A sweeping preconditioner for Yee's finite difference approximation of time-harmonic Maxwell's equations

Paul TSUJI[1], **Lexing YING**[1,2]

1 ICES, University of Texas at Austin, Austin, TX 78712, USA
2 Department of Mathematics, University of Texas at Austin, Austin, TX 78712, USA

Abstract This paper is concerned with the fast iterative solution of linear systems arising from finite difference discretizations in electromagnetics. The sweeping preconditioner with moving perfectly matched layers previously developed for the Helmholtz equation is adapted for the popular Yee grid scheme for wave propagation in inhomogeneous, anisotropic media. Preliminary numerical results are presented for typical examples.

Keywords Electromagnetic scattering, Yee grid, finite difference methods, perfectly matched layers, LDL^{T} factorizations, multifrontal method, wave propagation in inhomogeneous and anisotropic media, matrix preconditioners
MSC 65F08, 65N22, 65N80, 35Q61

1 Introduction

Medium-to-high frequency wave propagation in complex media is a problem of great interest in many fields; in electromagnetics, these problems arise in applications such as metamaterials and radar imaging. In order to compute the electromagnetic fields in an arbitrary medium, one must solve Maxwell's equations numerically for the given permittivity and permeability tensors $\varepsilon(\mathbf{r})$ and $\mu(\mathbf{r})$, respectively; the tensors here are functions of the spatial variable $\mathbf{r} \in \mathbb{R}^3$. This paper is concerned with solving Maxwell's equations with finite difference methods in the frequency domain.

When the wavelength of the incident or incoming field is much smaller in comparison to the size of the truncated domain, there are many oscillations in the solution and interesting scattering effects can occur. In order to physically

Received January 29, 2011; accepted December 22, 2011
Corresponding author: Lexing YING, E-mail: lexing@math.utexas.edu

capture these details, it is a commonly accepted practice to use discretizations with a constant number of samples s per λ, where λ is the operating wavelength. For a 3-D problem in Cartesian coordinates that is K wavelengths in diameter, the number of unknowns N grows as $O((Ks)^3)$. Thus, the number of unknowns grows very quickly with the size of the problem; because of the large memory requirements and long computational time, using a sparse direct solver becomes impractical for 3D applications, so iterative methods are employed. However, using iterative methods poses an alternative challenge: the matrix resulting from finite difference or finite element discretization is ill-conditioned and indefinite, hence rendering most existing multiscale techniques inefficient. For high-frequency problems, a large number of iterations is necessary for convergence even to a modest error tolerance; as the problem grows, this number increases significantly.

We propose using the sweeping preconditioner presented in [6] to reduce the prohibitively high number of iterations in the electromagnetics case. Originally introduced for the Helmholtz equation, this approach constructs an LDL^{T} factorization by eliminating the unknowns layer by layer starting from an absorbing boundary, and then approximates the Schur complement matrices of the factorization using moving perfectly matched layers (PMLs) introduced in the interior of the domain. This paper reports the current progress on applying this idea to Maxwell's equations. The rest of the paper is as follows. In Section 2, we review the frequency-domain finite-difference method using the Yee grid scheme in Cartesian coordinates. In section 3, we show how the sweeping preconditioner can be adapted for Maxwell's equations. In Section 4, we provide some preliminary numerical results that show the complexity of the algorithm and effectiveness of the preconditioner. Finally, we conclude with some remarks and comments on future work in Section 5.

2 FDFD method for electromagnetics

2.1 Perfectly matched layers

We begin with the time-harmonic Maxwell equations for general media on an infinite domain, which are

$$\begin{gathered}\nabla \times \mathbf{E} = -\mathrm{i}\omega\mu\mathbf{H}, \quad \nabla \times \mathbf{H} = \mathrm{i}\omega\varepsilon\mathbf{E} + \mathbf{J}, \\ \nabla \cdot (\varepsilon\mathbf{E}) = \rho, \quad \nabla \cdot (\mu\mathbf{H}) = 0,\end{gathered} \tag{2.1}$$

$$\lim_{|\mathbf{r}|\to\infty} (\mathbf{H} \times \mathbf{r} - |\mathbf{r}|\,\mathbf{E}) = 0, \tag{2.2}$$

$$\lim_{|\mathbf{r}|\to\infty} (\mathbf{E} \times \mathbf{r} + |\mathbf{r}|\,\mathbf{H}) = 0, \tag{2.3}$$

where $\mathrm{i} = \sqrt{-1}$, ω is the angular frequency, $\mathbf{J}$ is the current distribution, and ρ is the charge distribution; it should be noted that the current and charge satisfy the continuity equation

$$\nabla \cdot \mathbf{J} = -\mathrm{i}\omega\rho.$$

The last two limits are the Silver-Müller radiation conditions, which enforce the fields to radiate away from the current source and dissipate as $|\mathbf{r}|$ goes to infinity. We take the permittivity and permeability tensors $\varepsilon(\mathbf{r})$ and $\mu(\mathbf{r})$ to be measurable with respect to the spatial variable $\mathbf{r} \in \mathbb{R}^3$; the entries of these matrices are

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

Since it is impossible to numerically solve Maxwell's equations in all of $\mathbb{R}^3$, the computational domain is truncated and a boundary condition which emulates the radiation condition is introduced; for our problem, the domain that we consider is the unit cube $D = [0,1]^3$. Absorbing boundary conditions [4,9] have been very popular for the wave equation. Here, we choose to use the perfectly matched layer derived by complex-stretched coordinates [3] because of its ubiquity in computational electromagnetics.

Let us define the width of the PML as ℓ, so that the non-PML region in D is $[\ell, 1-\ell]^3$. Consider the complex stretching variables s_ξ for $\xi = x, y, z$ used in the homogeneous, isotropic case. These functions are of the form

$$s_\xi(\xi) = a(\xi) + \mathrm{i}\sigma(\xi), \tag{2.4}$$

with $a \geqslant 1$ and $\sigma \geqslant 0$; in the physical space outside the PML, we have $s_\xi = 1$. Typically, we choose $a = 1$ everywhere, and σ as the ramp-like function

$$\sigma(\xi) = \begin{cases} \theta\Big(\dfrac{\xi-\ell}{\ell}\Big)^2, & \xi \in [0,\ell], \\ 0, & \xi \in [\ell, 1-\ell], \\ \theta\Big(\dfrac{\xi-1+\ell}{\ell}\Big)^2, & \xi \in [1-\ell, 1], \end{cases} \tag{2.5}$$

where θ is an optimal constant inversely proportional to frequency [7]. Now, define the matrix

$$\mathbf{S} = \begin{pmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1/s_z \end{pmatrix}, \tag{2.6}$$

therefore,

$$\det(\mathbf{S}) = (s_x s_y s_z)^{-1}.$$

It has been shown in [2] that Maxwell's equations in complex stretched coordinates can be written in terms of this operator; when these equations are recast in the non-stretched coordinates, the material tensors take on the form

$$\widetilde{\varepsilon} = (\det(\mathbf{S}))^{-1}(\mathbf{S}\varepsilon\mathbf{S}), \quad \widetilde{\mu} = (\det(\mathbf{S}))^{-1}(\mathbf{S}\mu\mathbf{S}).$$

Explicitly, these matrix entries are

$$\begin{aligned}\widetilde{\varepsilon} &= \begin{pmatrix} \varepsilon_{xx}s_y s_z/s_x & \varepsilon_{xy}s_z & \varepsilon_{xz}s_y \\ \varepsilon_{yx}s_z & \varepsilon_{yy}s_x s_z/s_y & \varepsilon_{yz}s_x \\ \varepsilon_{zx}s_y & \varepsilon_{zy}s_x & \varepsilon_{zz}s_x s_y/s_z \end{pmatrix}, \\ \widetilde{\mu} &= \begin{pmatrix} \mu_{xx}s_y s_z/s_x & \mu_{xy}s_z & \mu_{xz}s_y \\ \mu_{yx}s_z & \mu_{yy}s_x s_z/s_y & \mu_{yz}s_x \\ \mu_{zx}s_y & \mu_{zy}s_x & \mu_{zz}s_x s_y/s_z \end{pmatrix}.\end{aligned} \tag{2.7}$$

Now, we can use (2.7) as the material tensors in the whole computational domain, since $s_\xi = 1$ in $[\ell, 1-\ell]^3$ and the PML reduces to the actual material. At the boundary of the domain ∂D, we artificially use PEC boundary conditions to close the system; because of the exponential decay of any plane wave entering the perfectly matched layer, the field is so small that the reflections off of the PEC outside of the PML are deemed insignificant. The infinite domain problem is now reduced to the truncated problem:

$$\left.\begin{aligned} \nabla\times\mathbf{E} = -\mathrm{i}\omega\widetilde{\mu}\mathbf{H}, \quad \nabla\times\mathbf{H} = \mathrm{i}\omega\widetilde{\varepsilon}\mathbf{E}+\mathbf{J}, \\ \nabla\cdot(\widetilde{\varepsilon}\mathbf{E}) = \rho, \quad \nabla\cdot(\widetilde{\mu}\mathbf{H}) = 0 \end{aligned}\right. \quad \text{in } D,$$
$$\hat{\mathbf{n}}\times\mathbf{E} = 0, \quad \hat{\mathbf{n}}\times\mathbf{H} = 0 \quad \text{on } \partial D.$$

2.2 Yee grid discretization

For the scalar Helmholtz equation, standard central differencing methods are sufficient and produce reasonably accurate results. In the case of Maxwell's equations, however, dispersion becomes a major problem if the components of the electric field **E** and magnetic field **H** are all defined at the same locations. The celebrated Yee grid, which was originally used for finite-difference time-domain simulations [10], is also applicable to the frequency domain [1]. In this scheme, the components of **E** and **H** are defined on a staggered grid. The advantages to using the Yee grid are two-fold: the divergence equations in (2.1) are implicitly satisfied, and boundary conditions between materials are naturally handled.

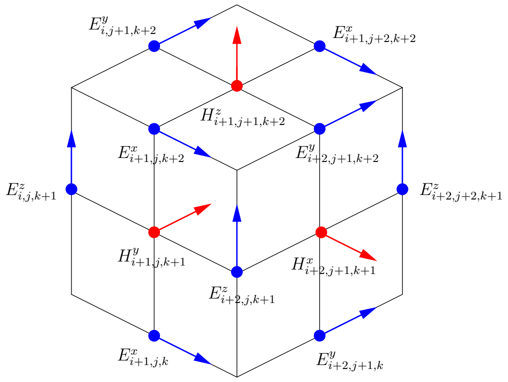


Fig. 1 Yee grid on a cubic cell. Vectors perpendicular to faces are components of **H** and vectors along edges are components of **E**.

Figure 1 illustrates the locations of the field components in Yee's scheme for an orthogonal Cartesian grid; with this structure, the finite difference formulas are

$$\begin{aligned}
&E^y_{i,j+1,k} - E^y_{i,j+1,k+2} + E^z_{i,j+2,k+1} - E^z_{i,j,k+1} + 2\mathrm{i}\omega h(\widetilde{\mu}\mathbf{H})^x_{i,j+1,k+1} = 0,\\
&E^z_{i,j,k+1} - E^z_{i+2,j,k+1} + E^x_{i+1,j,k+2} - E^x_{i+1,j,k} + 2\mathrm{i}\omega h(\widetilde{\mu}\mathbf{H})^y_{i+1,j,k+1} = 0,\\
&E^x_{i+1,j,k} - E^x_{i+1,j+2,k} + E^y_{i+2,j+1,k} - E^y_{i,j+1,k} + 2\mathrm{i}\omega h(\widetilde{\mu}\mathbf{H})^z_{i+1,j+1,k} = 0,\\
&\qquad H^z_{i+1,j+1,k} - H^z_{i+1,j-1,k} + H^y_{i+1,j,k-1}\\
&\qquad\quad -H^y_{i+1,j,k+1} - 2h(\mathrm{i}\omega(\widetilde{\varepsilon}\mathbf{E})^x_{i+1,j,k}) = 2h(J^x_{i+1,j,k}),\\
&\qquad H^x_{i,j+1,k+1} - H^x_{i,j+1,k-1} + H^z_{i-1,j+1,k}\\
&\qquad\quad -H^z_{i+1,j+1,k} - 2h(\mathrm{i}\omega(\widetilde{\varepsilon}\mathbf{E})^y_{i,j+1,k}) = 2h(J^y_{i,j+1,k}),\\
&\qquad H^y_{i+1,j,k+1} - H^y_{i-1,j,k+1} + H^x_{i,j-1,k+1}\\
&\qquad\quad -H^x_{i,j+1,k+1} - 2h(\mathrm{i}\omega(\widetilde{\varepsilon}\mathbf{E})^z_{i,j,k+1}) = 2h(J^z_{i,j,k+1}),
\end{aligned}\tag{2.8}$$

where

$$h = \frac{1}{n+1}$$

is the distance between two nodes in the grid, and the subscript notation for each component follows the convention

$$E^x_{i,j,k} \approx E_x(ih, jh, kh).$$

In the derivation of the finite difference formulas, the tensor product terms are the result of using the midpoint rule on the integrals

$$\int_{D_1} (\widetilde{\mu}\mathbf{H}) \cdot \hat{n}\,\mathrm{d}A, \qquad \int_{D_2} (\widetilde{\varepsilon}\mathbf{E}) \cdot \hat{n}\,\mathrm{d}A,$$

where

D_1 is a square face orthogonal to $\mathbf{H}$,

D_2 is a square face orthogonal to $\mathbf{E}$, and

$\hat{n}$ is the unit normal vector.

Observing the diagram in Fig. 1, we see that only one component of the electric and magnetic fields is defined at each point. This creates a problem when $\widetilde{\mu}$ and $\widetilde{\varepsilon}$ have non-zero off-diagonal entries, as the individual components of $\widetilde{\mu}\mathbf{H}$ and $\widetilde{\varepsilon}\mathbf{E}$ will be summations of these terms; we must somehow define a local approximation of the field components which are not defined at the midpoints of D_1 and D_2. To this end, we use the simple schemes introduced in previous works on FDTD methods [11], which take the average of the nearest four field values. The products $\widetilde{\mu}\mathbf{H}$ and $\widetilde{\varepsilon}\mathbf{E}$ on the grids are then

$$
\begin{aligned}
(\widetilde{\mu}\mathbf{H})^x_{i,j+1,k+1} &\approx \widetilde{\mu}^{xx}_{i,j+1,k+1}H^x_{i,j+1,k+1} + \frac{\widetilde{\mu}^{xy}_{i,j+1,k+1}}{4}(H^y_{i+1,j,k+1} + H^y_{i+1,j+2,k+1} \\
&\quad + H^y_{i-1,j,k+1} + H^y_{i-1,j+2,k+1}) + \frac{\widetilde{\mu}^{xz}_{i,j+1,k+1}}{4}(H^z_{i+1,j+1,k} \\
&\quad + H^z_{i+1,j+1,k+2} + H^z_{i-1,j+1,k} + H^z_{i-1,j+1,k+2}), \\
(\widetilde{\varepsilon}\mathbf{E})^x_{i+1,j,k} &\approx \widetilde{\varepsilon}^{xx}_{i+1,j,k}E^x_{i+1,j,k} \\
&\quad + \frac{\widetilde{\varepsilon}^{xy}_{i+1,j,k}}{4}(E^y_{i,j+1,k} + E^y_{i+2,j+1,k} + E^y_{i,j-1,k} + E^y_{i+2,j-1,k}) \\
&\quad + \frac{\widetilde{\varepsilon}^{xz}_{i+1,j,k}}{4}(E^z_{i,j,k+1} + E^z_{i+2,j,k+1} + E^z_{i,j,k-1} + E^z_{i+2,j,k-1})
\end{aligned}
\tag{2.9}
$$

for the x-components; the other components can be defined similarly. Now, these approximations can be inserted into (2.8) to get the finite difference formulas for fully anisotropic media.

2.3 Block tridiagonal structure and node ordering

Observe the finite difference equations given by (2.8) and (2.9), and consider the unknowns on the layer $z = kh$. We see that each node on this layer of the grid only interacts with nodes that satisfy

$$(k-1)h \leqslant z \leqslant (k+1)h.$$

Denoting u_k as the vector of unknowns on the layer $z = kh$, we can order the unknowns within each layer lexicographically by the x-coordinate first and y-coordinate second. We can then write the full vector of unknowns as $u = (u_1^{\mathrm{T}}, u_2^{\mathrm{T}}, \dots, u_n^{\mathrm{T}})^{\mathrm{T}}$, where

$$
\begin{aligned}
u_1 &= (H^x_{2,1,1}, H^x_{4,1,1}, \dots, H^y_{1,2,1}, E^z_{2,2,1}, \dots, H^x_{n-3,n,1}, H^x_{n-1,n,1})^{\mathrm{T}}, \\
u_2 &= (H^z_{1,1,2}, E^y_{2,1,2}, \dots, E^x_{1,2,2}, E^x_{3,2,2}, \dots, E^y_{n-1,n,2}, H^z_{n,n,2})^{\mathrm{T}}, \\
&\cdots, \\
u_{n-1} &= (H^z_{1,1,n-1}, E^y_{2,1,n-1}, \dots, E^x_{1,2,n-1}, E^x_{3,2,n-1}, \dots, E^y_{n-1,n,n-1}, H^z_{n,n,n-1})^{\mathrm{T}} \\
u_n &= (H^x_{2,1,n}, H^x_{4,1,n}, \dots, H^y_{1,2,n}, E^z_{2,2,n}, \dots, H^x_{n-3,n,n}, H^x_{n-1,n,n})^{\mathrm{T}}.
\end{aligned}
$$

Similarly, the right-hand side f contains the information on the current source and can be written as $f = (f_1^{\mathrm{T}}, f_2^{\mathrm{T}}, \dots, f_n^{\mathrm{T}})^{\mathrm{T}}$ following the same ordering.

Given the full vector of unknowns $u = (u_1^{\mathrm{T}}, u_2^{\mathrm{T}}, \dots, u_n^{\mathrm{T}})^{\mathrm{T}}$ and the right-hand side $f = (f_1^{\mathrm{T}}, f_2^{\mathrm{T}}, \dots, f_n^{\mathrm{T}})^{\mathrm{T}}$, we can write the linear system $Au = f$ in the block tridiagonal form

$$
\begin{pmatrix}
A_{1,1} & A_{1,2} & & \\
A_{2,1} & A_{2,2} & \ddots & \\
 & \ddots & \ddots & A_{n-1,n} \\
 & & A_{n,n-1} & A_{n,n}
\end{pmatrix}
\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}
=
\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}.
\tag{2.10}
$$

It is important to note that the off-diagonal blocks here are not square; because of the staggered grid, there is a slightly different number of unknowns in each layer.

3 Sweeping preconditioner

In [6], the sweeping preconditioner with moving PML was developed for the scalar Helmholtz equation with variable media. Here, we present the preconditioner adapted to Maxwell's equations and the Yee grid.

3.1 Block LDL^{T} factorization

After arriving at (2.10), we are in position to discuss the sweeping factorization. Let P_k be the unknowns on the k-th layer. By eliminating the unknowns layer by layer, the block LDL^{T} factorization for the matrix A can be written as

$$A = L_1 \cdots L_{n-1} \begin{pmatrix} S_1 & & & \\ & S_2 & & \\ & & \ddots & \\ & & & S_n \end{pmatrix} L_{n-1}^{\mathrm{T}} \cdots L_1^{\mathrm{T}}, \tag{3.1}$$

where

$$S_1 = A_{1,1}, \quad S_m = A_{m,m} - A_{m,m-1} S_{m-1}^{-1} A_{m-1,m}, \quad m = 2, \dots, n,$$

and L_k are the block lower triangular matrices given by

$$L_k(P_i, P_j) = \begin{cases} A_{k+1,k} S_k^{-1}, & i = k+1,\, j = k, \\ I, & 1 \leqslant i = j \leqslant n, \\ 0, & \text{otherwise}. \end{cases} \tag{3.2}$$

Inverting this factorization and applying it to the right-hand side f, we can arrive at the solution

$$u = (L_1^{\mathrm{T}})^{-1} \cdots (L_{n-1}^{\mathrm{T}})^{-1} \begin{pmatrix} S_1^{-1} & & & \\ & S_2^{-1} & & \\ & & \ddots & \\ & & & S_n^{-1} \end{pmatrix} L_{n-1}^{-1} \cdots L_1^{-1} f. \tag{3.3}$$

Our goal is to find an approximate inverse M^{-1} efficiently and solve the preconditioned system $M^{-1}Au = M^{-1}f$ iteratively. Here, the main computational task is constructing the inverse operators of $S_1, \dots, S_n$, as these matrices are dense; we will focus on this problem very shortly.

3.2 Main physical observation

To obtain an accurate approximation for the inversion of the Schur complement matrices, we attempt to gain some physical intuition by restricting the problem to the first m layers. Let us observe the upper $m \times m$ blocks of the block tridiagonal matrix A; if we only consider the degrees of freedom for layers $1, \ldots, m$, where layer m is outside the PML, we can obtain the smaller linear system

$$\begin{pmatrix} A_{1,1} & A_{1,2} & & \\ A_{2,1} & A_{2,2} & \ddots & \\ & \ddots & \ddots & A_{m-1,m} \\ & & A_{m,m-1} & A_{m,m} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix}. \tag{3.4}$$

Removing the degrees of freedom for layers $m+1$ to n essentially strips the domain of these layers and enforces PEC boundary conditions at layer $m+1$; that is, the matrix equation (3.4) corresponds to the discretization of the half-space PEC plane problem for Maxwell's equations on layers 1 to m, where the PEC plane is located at layer $m+1$. If we take the inverse of the operator on the left-hand side, we have

$$\begin{aligned} &\begin{pmatrix} A_{1,1} & A_{1,2} & & \\ A_{2,1} & A_{2,2} & \ddots & \\ & \ddots & \ddots & A_{m-1,m} \\ & & A_{m,m-1} & A_{m,m} \end{pmatrix}^{-1} \\ &= (L_1^{\mathrm{T}})^{-1} \cdots (L_{m-1}^{\mathrm{T}})^{-1} \begin{pmatrix} S_1^{-1} & & \\ & \ddots & \\ & & S_m^{-1} \end{pmatrix} L_{m-1}^{-1} \cdots L_1^{-1}. \end{aligned} \tag{3.5}$$

We note that the inversion formula above is similar in structure to the inverse of the full matrix A; this time, however, we only need the lower triangular matrices $L_1, \ldots, L_{m-1}$ and Schur complements $S_1, \ldots, S_m$. The left-hand side in equation (3.5) is Green's function for the half-space problem, a dense matrix which can be written in block form as

$$\begin{pmatrix} A_{1,1} & A_{1,2} & & \\ A_{2,1} & A_{2,2} & \ddots & \\ & \ddots & \ddots & A_{m-1,m} \\ & & A_{m,m-1} & A_{m,m} \end{pmatrix}^{-1} = \begin{pmatrix} G_{1,1} & G_{1,2} & \cdots & G_{1,m} \\ G_{2,1} & G_{2,2} & \cdots & G_{2,m} \\ \vdots & \vdots & & \vdots \\ G_{m,1} & G_{m,2} & \cdots & G_{m,m} \end{pmatrix}, \tag{3.6}$$

where the entries in each block $G_{i,j}$ give the fields in layer i due to sources in layer j. The crucial observation here is that S_m^{-1} remains untouched by the left operator $(L_1^{\mathrm{T}})^{-1} \cdots (L_{m-1}^{\mathrm{T}})^{-1}$ and right operator $L_{m-1}^{-1} \cdots L_1^{-1}$, due to the definition of $L_1, \ldots, L_{m-1}$; that is, if the matrix multiplications on the right-hand side of (3.5) are carried out, then the matrix in the (m,m)-th block is just S_m^{-1}. Algebraically, this gives us the result $G_{m,m} = S_m^{-1}$. The physical knowledge we have gained is that S_m^{-1} *is approximately the discrete half-space Green's function for Maxwell's equations with PEC boundary conditions on layer* $m+1$, *restricted to degrees of freedom on the m-th layer.* By solving the half-space problem on a grid of the first m layers, we can construct an operator which reproduces S_m^{-1}. However, this becomes very costly as m approaches n, so our goal is to approximate S_m^{-1} efficiently. The authors of [5,6] have proposed two methods for this approximation: the hierarchical matrix approach and the moving PML approach. Here, we will focus on the latter technique.

3.3 Moving PML method

The application of S_m^{-1} only involves the degrees of freedom on the m-th layer; that is, the half space Green's function matrix which we are trying to approximate ($G_{m,m}$) only maps the right-hand side on layer m to the solution on layer m. Therefore, the solution of the half-space problem only needs to be accurate in a small neighborhood of $z = mh$. Recall that the purpose of the PML is to be an absorbing boundary; in the original problem, we were not concerned with fields outside the cube $[\ell, 1-\ell]^3$, so we placed a PML outside of this cube and made the computational domain of the full problem to be $[0,1]^3$. We can use the same exact reasoning to truncate each half-space problem; if we are only concerned with the layers in the immediate vicinity of the m-th layer, then we can treat layers 1 to $m-1$ as buffer layers or "white space". Thus, we can push the PML up to the edge of our domain of interest and still get a good approximation of the solution on layer m. The computational advantage here is that we need to solve a much smaller subproblem for each S_m^{-1}. The technique of truncating the domain and pushing the PML closer is called the *moving PML method.*

To be more precise, let $b = \ell/h$ be the number of PML layers and consider the domain

$$D_m = [0,1] \times [0,1] \times [(m-b)h, (m+1)h].$$

Let us also define the shifted PML function

$$s_z^m(z) = 1 + \mathrm{i}\sigma(z - (m-b)h), \tag{3.7}$$

and shifted material tensors

$$\begin{aligned}\widetilde{\varepsilon}_m &= \begin{pmatrix} \varepsilon_{xx} s_y s_z^m / s_x & \varepsilon_{xy} s_z^m & \varepsilon_{xz} s_y \\ \varepsilon_{yx} s_z^m & \varepsilon_{yy} s_x s_z^m / s_y & \varepsilon_{yz} s_x \\ \varepsilon_{zx} s_y & \varepsilon_{zy} s_x & \varepsilon_{zz} s_x s_y / s_z^m \end{pmatrix}, \\ \widetilde{\mu}_m &= \begin{pmatrix} \mu_{xx} s_y s_z^m / s_x & \mu_{xy} s_z^m & \mu_{xz} s_y \\ \mu_{yx} s_z^m & \mu_{yy} s_x s_z^m / s_y & \mu_{yz} s_x \\ \mu_{zx} s_y & \mu_{zy} s_x & \mu_{zz} s_x s_y / s_z^m \end{pmatrix}.\end{aligned} \tag{3.8}$$

With the PML pushed to the edge, the subproblem to be solved for each S_m^{-1} is

$$\begin{aligned} &\left.\begin{aligned}\nabla \times \mathbf{E} = -\mathrm{i}\omega\widetilde{\mu}_m\mathbf{H}, &\quad \nabla \times \mathbf{H} = \mathrm{i}\omega\widetilde{\varepsilon}_m\mathbf{E} + \mathbf{J}, \\ \nabla \cdot (\widetilde{\varepsilon}_m\mathbf{E}) = \rho, &\quad \nabla \cdot (\widetilde{\mu}_m\mathbf{H}) = 0\end{aligned}\right\} \text{ in } D_m, \\ &\hat{\mathbf{n}} \times \mathbf{E} = 0, \quad \hat{\mathbf{n}} \times \mathbf{H} = 0 \quad \text{on } \partial D_m,\end{aligned} \tag{3.9}$$

and the subgrid on which the above equations are discretized is

$$G_m = \{(ih, jh, kh) \mid 1 \leqslant i, j \leqslant n,\ m-b+1 \leqslant k \leqslant m\}. \tag{3.10}$$

To solve each discretized subproblem, a version of the multifrontal method [6,8] is utilized, with a modification that allows the Yee grid to be handled naturally. Consider the matrix H_m resulting from the discretization of (3.9) on G_m. Essentially, each subproblem can be viewed as a quasi-2D problem, since the number of PML and buffer layers is small. The first step of the algorithm is to partition the nodes of G_m hierarchically in the x-y plane, i.e., nodes with the same x and y indices remain in the same group. In the Helmholtz case, every node is associated with a variable; in the Maxwell case, however, the Yee grid does not have an unknown assigned to every node. To keep the same efficiency of the method with the Helmholtz grid, we leave these empty nodes in at the hierarchical partitioning stage; after each cluster is set, we then remove any empty nodes, as they contain no information relevant to the factorization. The unknowns associated with the full nodes are then reordered according to their hierarchical groups to minimize the number of fill-ins in the LDL^{T} factorization of H_m. The costs of computing the factorization and applying to a vector are $O(b^3n^3)$ and $O(b^2n^2\log n)$, respectively; we refer to the previous works for details.

Once the multifrontal factorization of H_m is constructed, we can approximate the application of S_m^{-1} to a vector as follows. Given a vector of values g_m defined on the grid points of layer m, we must construct a longer vector padded with zeros, which will correspond to the grid points on layers $m-b$ to $m-1$; the zeros are necessary to ensure that the solution is not corrupted by Green's function on these layers. After the vector is made long enough to match the dimensions of H_m, we can compute the matrix-vector product

$$H_m^{-1}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_m \end{pmatrix} = \begin{pmatrix} * \\ \vdots \\ * \\ v_m \end{pmatrix} \tag{3.11}$$

and extract the vector v_m; this will give us the approximation of $S_m^{-1}g_m = v_m$ we need. We denote this combined process of concatenation, application of H_m^{-1}, and extraction as the operator

$$\widetilde{T}_m\colon\ g_m \to v_m.$$

We can finally summarize the moving PML preconditioner in two stages: the construction of the approximate sweeping factorization, and the application to an arbitrary vector. We stay consistent to [6] and define

$$u_F = (u_1^{\mathrm{T}}, \ldots, u_b^{\mathrm{T}}), \quad f_F = (f_1^{\mathrm{T}}, \ldots, f_b^{\mathrm{T}}), \tag{3.12}$$

while rewriting A as

$$\begin{pmatrix} A_{F,F} & A_{F,b+1} & & \\ A_{b+1,F} & A_{b+1,b+1} & \ddots & \\ & \ddots & \ddots & A_{n-1,n} \\ & & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} u_F \\ u_{b+1} \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_F \\ f_{b+1} \\ \vdots \\ f_n \end{pmatrix}, \tag{3.13}$$

where $A_{F,F}$ is the upper-left block of A for the first b layers.

Algorithm 3.1 *Construct the approximate sweeping factorization of A.*

```
1. Let $G_F$ be the subgrid of the first $b$ layers and $H_F = A_{F,F}$; construct the multifrontal factorization of $H_F$.
2. for $m = b+1, \dots, n$ do
       Let $G_m$ be as defined in (3.10) and $H_m$ be the matrix resulting from the finite difference discretization of (3.9). Construct the multifrontal factorization of $H_m$.
   end for
```

Algorithm 3.2 *Apply the approximate inverse to get $u \approx A^{-1}f$.*

```
1. $u_F = f_F$ and $u_m = f_m$ for $m = b+1, \dots, n$.
2. $u_{b+1} = u_{b+1} - A_{b+1,F}(H_F^{-1}u_F)$, using the multifrontal factorization of $H_F$.
3. for $m = b+1, \dots, n-1$ do
       $u_{m+1} = u_{m+1} - A_{m+1,m}(\widetilde{T}_m u_m)$, where $\widetilde{T}_m u_m$ is computed by the process described earlier in Section 3.3.
   end for
4. $u_F = H_F^{-1}u_F$, using the multifrontal factorization of $H_F$.
5. for $m = b+1, \dots, n$ do
       $u_m = \widetilde{T}_m u_m$. See previous steps for the application of $\widetilde{T}_m$.
   end for
6. for $m = n-1, \dots, b+1$ do
       $u_m = u_m - \widetilde{T}_m(A_{m,m+1}u_{m+1})$. See previous steps for the application of $\widetilde{T}_m$.
   end for
7. $u_F = u_F - H_F^{-1}(A_{F,b+1}u_{b+1})$, using the multifrontal factorization of $H_F$.
```

Because the total number of degrees of freedom is $N = n^3$ and there are n layers, it is clear that the cost of Algorithms 3.1 and 3.2 are

$$O(b^3n^4) = O(b^3N^{4/3}), \quad O(b^2n^3 \log n) = O(b^2N \log N),$$

respectively.

In practice, the approach is slightly different. First, the block LDL^{T} factorization is constructed so that multiple layers can be preconditioned for each subproblem; that is, S_m^{-1} is not just applied to layer m, but layers $m-d+1$

to m, where d is a specified number of buffer layers. The subproblem for each S_m is instead defined on an expanded domain

$$D'_m = [0,1] \times [0,1] \times [(m-b-d+1)h, (m+1)h].$$

Second, the perturbed matrix associated with the curl equations

$$\nabla \times \mathbf{E} = -\mathrm{i}(\omega + \mathrm{i}\alpha)\widetilde{\mu}\mathbf{H}, \quad \nabla \times \mathbf{H} = \mathrm{i}(\omega + \mathrm{i}\alpha)\widetilde{\varepsilon}\mathbf{E} + \mathbf{J} \tag{3.14}$$

is used instead of the true matrix A, for some small damping constant α; the preconditioner produced from this matrix is much more stable and effective. We will denote the approximate inverse operator constructed by Algorithm 3.1 with constant α as M_α^{-1}; we use this approximate inverse as a left preconditioner and solve the preconditioned linear system

$$M_\alpha^{-1} A u = M_\alpha^{-1} f. \tag{3.15}$$

4 Numerical results

In this section, we provide some preliminary results illustrating the effectiveness of the sweeping preconditioner for the Yee grid. The general setup of our problems is as follows. We have a point source oriented in the z-direction with current magnitude $|\mathbf{J}| = 1$, which is embedded in the material domain at $(0.5, 0.25, 0.5)$. As mentioned before, the domain is the unit cube $[0,1]^3$, and has a PEC boundary. We denote K as the size of the domain in terms of wavelengths, so $\lambda = 1/K$. For the Yee grid, we choose the number of points per wavelength at 6 to attain reasonably accurate results; this implies that field components of the same kind are separated by $\lambda/6$, but the distance between two nodes in the physical grid is $h = \lambda/12$. We choose the PML width to be $\ell = \lambda/2$, and the number of buffer layers $d = 12$. For the damping constant, we choose $\alpha = 1$. As for the iterative solver, we use GMRES iteration with a relative residual tolerance of $1\mathrm{e}-3$. For ease of implementation and generating figures, all of the code is run serially in MATLAB.

We have chosen three example mediums. The first example is the converging lens with a Gaussian profile centered at $\mathbf{r}_c = (0.5, 0.5, 0.5)$ for both permittivity and permeability; that is, the material is isotropic, with

$$\widetilde{\varepsilon} = \varepsilon(\mathbf{r})\overline{\overline{\mathbf{I}}}, \quad \widetilde{\mu} = \mu(\mathbf{r})\overline{\overline{\mathbf{I}}},$$

and

$$\varepsilon(\mathbf{r}) = \varepsilon_0 \frac{1}{\frac{4}{3}\left(1 - \frac{1}{2}\mathrm{e}^{-32|\mathbf{r}-\mathbf{r}_c|^2}\right)}, \quad \mu(\mathbf{r}) = \mu_0 \frac{1}{\frac{4}{3}\left(1 - \frac{1}{2}\mathrm{e}^{-32|\mathbf{r}-\mathbf{r}_c|^2}\right)}.$$

Our second example is a random isotropic medium, which we form with a random smooth perturbation function $\delta(\mathbf{r})$ which satisfies $0 < \delta < 1$; once again, the material tensors are formed by multiplying the identity tensor with scalar functions

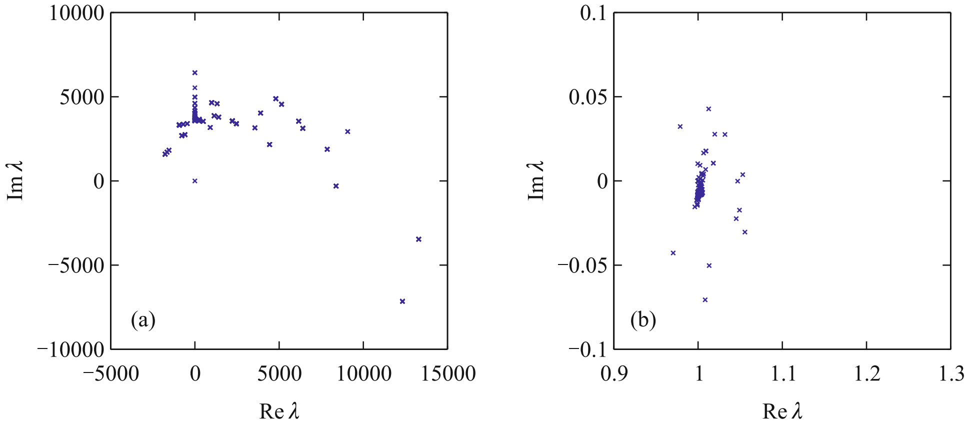
$$\varepsilon(\mathbf{r}) = \varepsilon_0 \frac{4}{3}\Big(1 - \frac{1}{2}\,\delta(\mathbf{r})\Big), \quad \mu(\mathbf{r}) = \mu_0 \frac{4}{3}\Big(1 - \frac{1}{2}\,\delta(\mathbf{r})\Big).$$

Finally, for an anisotropic, inhomogeneous example, we have the medium

$$\varepsilon(\mathbf{r}) = \varepsilon_0 \begin{pmatrix} 1.1 & 0.1\,\mathrm{i} & 0 \\ -0.1\,\mathrm{i} & 0.9 & 0 \\ 0 & 0 & \gamma(\mathbf{r}) \end{pmatrix}, \quad \mu(\mathbf{r}) = \mu_0 \begin{pmatrix} 1.1 & 0.1\,\mathrm{i} & 0 \\ -0.1\,\mathrm{i} & 0.9 & 0 \\ 0 & 0 & \gamma(\mathbf{r}) \end{pmatrix},$$

where γ is a smooth random perturbation function satisfying $0.7 < \gamma < 1.3$. This medium is similar to gyrotropic mediums found in crystals, except the z-component of the tensor is randomized.

To start, we must test how the preconditioner affects the eigenvalues and condition numbers of the original matrix. Fig. 2 shows that the original problem is indefinite and very ill-conditioned; a plot of the eigenvalues for the converging lens media illustrates that the original matrix A has large eigenvalues which can have a negative real part, as well as eigenvalues close to the origin. After preconditioning, the spectrum is clustered around $(1, 0)$ in the positive real half of the complex plane, which is conducive for the convergence of GMRES. Using MATLAB's condition number estimator condest, we have estimated the 1-norm condition numbers for small problems ($K = 2$) before and after preconditioning; the method reduces the condition number by several orders of magnitude.



Medium	cond(A)	cond($M_\alpha^{-1}A$)
converging lens	1.875e+07	42.407
random isotropic	2.959e+07	22.729
random anisotropic	3.305e+07	11.841

Fig. 2 Eigenvalues of coefficient matrix A in converging lens problem, (a) before and (b) after preconditioning, along with condition number estimates for different media.

For each medium, we have plotted the E_z component in the x-y plane at $z = 0.5$; because the source is aligned in the z-direction, the most significant behavior in the isotropic case happens in this component. Figs. 3–5 give the numerical results for each particular example. To illustrate the characteristics of the material, we show the relative material parameters for the isotropic cases and the relative ε_{zz} for the anisotropic example. In the table, we list the size of the problem K, number of unknowns N, preconditioner setup time T_{setup}, iterative solver time T_{solve}, and number of iterations N_{iter}. It is observed that even if the size of the problem is increased, the number of iterations remains almost constant. We note that the preconditioner takes the same time for the anisotropic medium as it does for the isotropic mediums; thus, there is no dependency on the type of medium for the method to work.

Next, we test the accuracy of the preconditioning method by running GMRES to a smaller residual tolerance and comparing with the numerical solution of the linear system using MATLAB's backslash operator; in these examples, we set the tolerance to be $1\text{e}-6$. We denote the relative 2-norm error between the direct solver and iterative solver result as ε. For the converging lens problem, Table 1 shows that the preconditioner agrees with numerically stable direct solvers, as the relative error does not exceed the given residual tolerance. We note that when the residual tolerance has decreased, the number of iterations still remains essentially independent of frequency. For each digit of accuracy, about one additional iteration is needed.

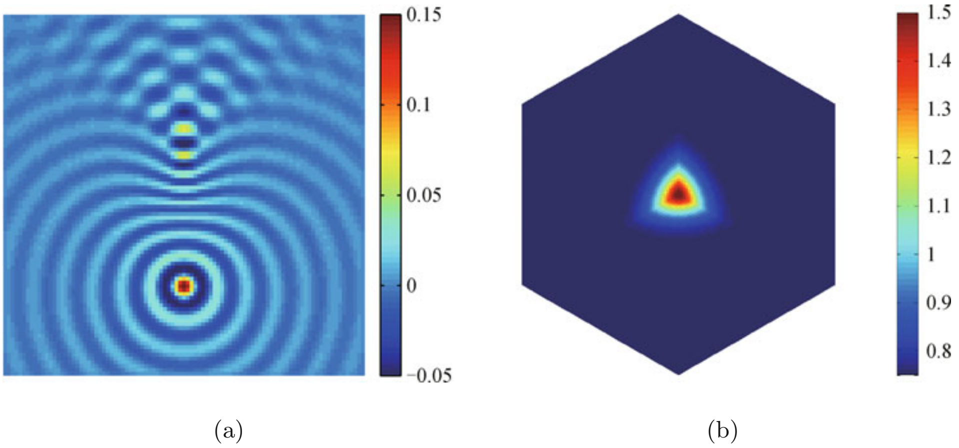
Table 1 Accuracy of preconditioning method for converging lens problem

K	N	N_{iter}	ε
4	7.783e+4	7	2.4572e−7
8	6.429e+5	7	3.5560e−7
16	5.225e+6	8	4.3398e−7

Finally, to show the complexities for the setup and apply stages, we have plotted the setup time and apply time against the total number of degrees of freedom. Fig. 6 illustrates the almost linear complexity of the sweeping preconditioner. Since we are keeping the number of gridpoints per wavelength constant, each time we double the frequency, the total number of DOFs should increase by a factor of 8; this implies the setup time should increase by a factor of $8^{4/3} = 16$. However, we typically see an increase by a factor of 12 or 13; this trends with $O(N^{6/5})$ complexity instead. The apply time, as stated, is $O(N \log N)$.

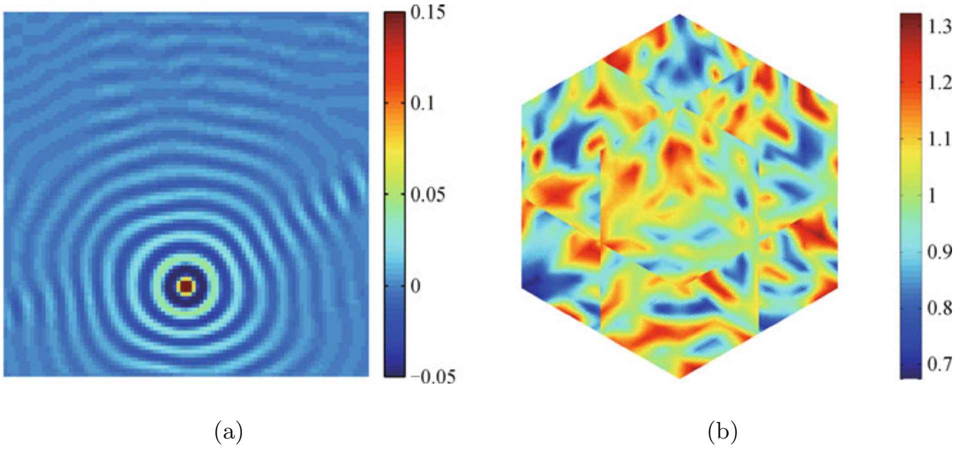
5 Conclusions

In this paper, we have adapted the sweeping preconditioner originally used for the Helmholtz equation to speed up the solution of Maxwell's equations through the Yee grid. We have shown that the setup time and solve time scale



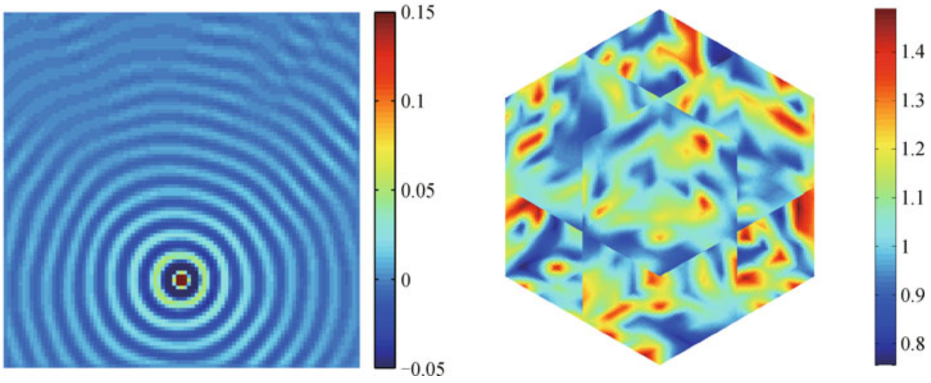
K	N	T_{setup}	T_{solve}	N_{iter}
4	7.783e+4	24	8	4
8	6.429e+5	303	84	4
16	5.225e+6	3938	851	4

Fig. 3 Converging lens example.
(a) $\operatorname{Re} E_z$ in x-y plane at $z = 0.5$; (b) relative ε and μ in medium.



K	N	T_{setup}	T_{solve}	N_{iter}
4	7.783e+4	23	8	4
8	6.429e+5	301	105	5
16	5.225e+6	3946	1064	5

Fig. 4 Random isotropic media example.
(a) $\operatorname{Re} E_z$ in x-y plane at $z = 0.5$; (b) relative ε and μ in medium.



(a) (b)

K	N	T_{setup}	T_{solve}	N_{iter}
4	7.783e+4	24	8	4
8	6.429e+5	305	85	4
16	5.225e+6	3942	859	4

Fig. 5 Random anisotropic media example.
(a) $\operatorname{Re} E_z$ in x-y plane at $z = 0.5$; (b) relative ε_{zz} in medium.

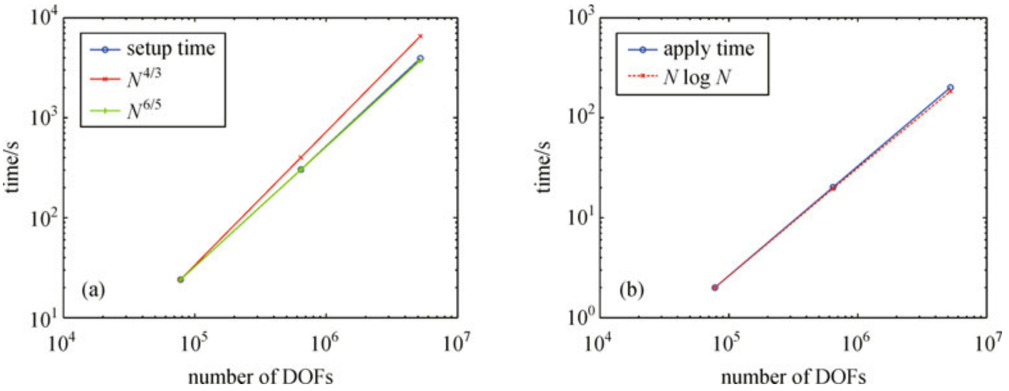


Fig. 6 (a) Setup and (b) apply times plotted against number of DOFs

as $O(N^{4/3})$ and $O(N \log N)$, respectively, just as they did in the Helmholtz case. Since there are roughly six times the number of unknowns in the Maxwell case, the prefactor of these estimates is considerably larger; however, this is the unavoidable nature of working with vector quantities.

Although we have presented some preliminary examples of anisotropic and inhomogeneous media problems, there is a considerable amount of future work that can be done to generalize the method.

First, for scattering problems with curved geometries, using the Yee grid in Cartesian coordinates is not optimal because of the staircase approximation; in this case, some special boundary condition must be implemented, or a more general method such as finite elements must be introduced. In either case, the adaptation of the preconditioner is non-trivial.

Second, due to the six components of **E** and **H**, the number of unknowns is very high even for a small problem; an alternative method that could be implemented would be the vector Helmholtz formulation, where the two curl equations are combined to eliminate either **E** or **H**. This would cut the number of unknowns in half, but the multifrontal solver would have to be modified because of the expanded stencil. In comparison to the Yee grid, there exist more efficient higher-order discretizations such as the spectral element method; these methods would decrease the degrees of freedom per wavelength significantly, allowing us to solve much larger problems.

Third, numerical results have shown that the sweeping preconditioner is an efficient approximate inverse of discretized Maxwell equations. A natural question is whether its accuracy can be systematically improved. A possible solution is to increase the width of the moving PML, however, it would also increase the computational complexity significantly. It remains an interesting question to see what the right balance is.

Finally, the structure of the algorithm and the use of the multifrontal method suggests that a parallel implementation would be natural.

References

1. Champagne N J, Berryman J G, Buettner H M. FDFD: A 3D Finite-difference frequency-domain code for electromagnetic induction tomography. J Comput Phys, 2001, 170(2): 830–848
2. Chew W C, Jin J, Michielssen E, Song J. Fast and Efficient Algorithms in Computational Electromagnetics. London: Artech House, 2001
3. Chew W C, Weedon W H. A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates. Microwave Opt Tech Lett, 1994, 7(13): 599–604
4. Engquist B, Majda A. Absorbing boundary conditions for the numerical simulation of waves. Math Comp, 1977, 31: 629–651
5. Engquist B, Ying L. Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation. Comm Pure Appl Math, 2011, 64: 697–735
6. Engquist B, Ying L. Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. Multiscale Model Simul, 2011, 9: 686–710
7. Jin J. The Finite Element Method in Electromagnetics. Hoboken: Wiley-IEEE Press, 2002
8. Lin L, Lu J, Ying L, Car R, E W. Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems. Commun Math Sci (to appear)
9. Mur G. Absorbing boundary conditions for the finite-difference approximation of time-domain electromagnetic field equations. IEEE Trans Electromag Compat, 1981, 23: 377–382
10. Taflove A, Hagness S. Computational Electrodynamics: the Finite-difference Time-domain Method. London: Artech House, 2005
11. Werner G R, Cary J R. A stable FDTD algorithm for non-diagonal, anisotropic dielectrics. J Comput Phys, 2007, 226(1): 1085–1101