PARALLEL NUMERICAL SOLUTION OF THE TIME-HARMONIC MAXWELL EQUATIONS IN MIXED FORM

DAN LI*, CHEN GREIF*, AND DOMINIK SCHÖTZAU†

Abstract. We present a fully scalable parallel iterative solver for the time-harmonic Maxwell equations in mixed form with small wave numbers. It is based on mixed finite element discretization on tetrahedral meshes, using the lowest-order Nédélec elements of the first kind for the approximation of the vector field and standard nodal elements for the Lagrange multiplier associated with the divergence constraint. The corresponding linear system has saddle point form, and is solved using preconditioned MINRES. We adopt a block diagonal preconditioning approach, with a nodal auxiliary space preconditioning technique as an inner iteration for the shifted curl-curl operator. This approach leads to a sequence of scalar elliptic problems, which we solve by applying conjugate gradients, preconditioned with algebraic multigrid. We demonstrate the performance of our parallel solver on problems with constant and variable coefficients with up to approximately 40 million degrees of freedom. Our numerical results indicate very good scalability with the mesh size, on uniform, unstructured and locally refined meshes.

Key words. Parallel iterative solvers, saddle point linear systems, preconditioners, Maxwell equations

1. Introduction.

The time-harmonic Maxwell equations in a lossless medium with perfectly conducting boundaries are

$$\begin{aligned} \nabla \times (\mu_r^{-1} \nabla \times E) - k^2 \epsilon_r E &= f \quad \text{in} \quad \Omega, \\ n_\Gamma \times E &= 0 \quad \text{on} \quad \Gamma. \end{aligned} \tag{1.1}$$

Here $\Omega \in \mathbb{R}^3$ is a polyhedral domain, which we assume is simply connected with a connected boundary $\Gamma = \partial\Omega$, E is the electric field, f is a generic source, and n_Γ denotes the outward unit normal on Γ. The electromagnetic parameters μ_r and ϵ_r denote the relative permeability and permittivity, respectively, which are scalar functions of position. We assume that

$$0 < \mu_{\min} \leq \mu_r \leq \mu_{\max} < \infty \qquad \text{and} \qquad 0 < \epsilon_{\min} \leq \epsilon_r \leq \epsilon_{\max} < \infty.$$

The wave number k is assumed small throughout this paper,

$$k \ll 1,$$

and is given by $k^2 = \omega^2 \epsilon_0 \mu_0$, where $\mu_0 = 4\pi \times 10^{-7}$ [H/m] and $\epsilon_0 = \frac{1}{36\pi} \times 10^{-9}$ [F/m] are the permeability and permittivity in vacuum, respectively, and $\omega \neq 0$ is the angular frequency. We assume throughout that $k^2\epsilon_r$ is not a Maxwell eigenvalue.

We are interested in numerically solving (1.1) using mixed finite element methods. Such formulations may improve the stability for vanishing wave numbers [5, 21] and yield a well defined problem. A mixed approach has been used in [4] for treating non-matching meshes. To derive a mixed formulation we apply the Helmholtz decomposition [19], and write

$$E = u + \nabla \hat{p},$$

*Department of Computer Science, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada, {danli,greif}@cs.ubc.ca.

†Mathematics Department, University of British Columbia, Vancouver, BC, V6T 1Z2, Canada, schoetzau@math.ubc.ca.

where $\epsilon_r u$ is divergence-free, and $\hat{p}$ has homogeneous Dirichlet boundary conditions. Setting $p = -k^2\hat{p}$, we obtain the following mixed problem: find the vector field u and the scalar multiplier p such that

$$\begin{aligned} \nabla \times \mu_r^{-1} \nabla \times u - k^2 \epsilon_r u + \epsilon_r \nabla p &= f \quad \text{in} \quad \Omega, && (1.2\text{a}) \\ \nabla \cdot (\epsilon_r u) &= 0 \quad \text{in} \quad \Omega, && (1.2\text{b}) \\ n_\Gamma \times u &= 0 \quad \text{on} \quad \Gamma, && (1.2\text{c}) \\ p &= 0 \quad \text{on} \quad \Gamma. && (1.2\text{d}) \end{aligned}$$

Note that (1.2) is well-posed for $k = 0$. In this case, if $\epsilon_r \equiv 1$, this corresponds to the mixed formulation of the magnetostatic problem in terms of the vector potential u, along with Coulomb's gauge $\nabla \cdot u = 0$. In our numerical tests, we will also consider $k = 0$ and variable ϵ_r.

Finite element discretization using Nédélec elements of the first kind [18] for the approximation of u and standard nodal elements for p yields a saddle point linear system of the form

$$\mathcal{K}x \equiv \begin{pmatrix} A - k^2 M & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \equiv b, \tag{1.3}$$

whose size is $(m+n) \times (m+n)$. The matrix $A \in \mathbb{R}^{n \times n}$ corresponds to the μ_r^{-1}-weighted discrete curl-curl operator; $B \in \mathbb{R}^{m \times n}$ is the ϵ_r-weighted divergence operator with full row rank; $M \in \mathbb{R}^{n \times n}$ is the ϵ_r-weighted vector mass matrix; $f \in \mathbb{R}^n$ is now the load vector associated with the right-hand side in (1.2a), and the vectors $u \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ represent the finite element coefficients. Note that A is symmetric positive semidefinite with nullity m.

In [8], block diagonal preconditioners were designed for iteratively solving system (1.3) with constant coefficients. These preconditioners were motivated by spectral equivalence properties. Each iteration of the scheme requires inverting $A + \gamma M$, where $\gamma > 0$ is a given parameter. There exist several effective multigrid methods for doing so. The standard V-cycle multigrid algorithm has been proved to be an effective solver or preconditioner for this problem in [7]. When a hierarchy of structured meshes is available, geometric multigrid can be applied [11]; for unstructured meshes, algebraic multigrid (AMG) approaches have been explored in [2, 14, 20], using the smoothers introduced in [11]. A different approach has been proposed by Hiptmair and Xu in [13]: a nodal auxiliary space preconditioner, which reduces the problem into solving two scalar elliptic problems on the nodal finite element space. In [16], a parallel implementation of the nodal auxiliary space preconditioners was discussed.

In this paper we develop a fully scalable parallel implementation for solving (1.3) in complicated domains in three dimensions. The preconditioned iterations are based on outer iterations of the form introduced in [8], extended to the variable coefficient case, and inner iterations of the form derived in [13]. We use algebraic multigrid solvers for each elliptic problem, and accomplish almost linear complexity in the number of degrees of freedom. With our implementation we can solve problems of dimensions of up to approximately 40 million degrees of freedom. Our numerical results scale very well with the mesh size, on uniform, unstructured, and locally refined meshes.

The remainder of the paper is structured as follows. In Section 2 we analyze the properties of the discrete operators. The preconditioning approach is presented in Section 3. In Section 4 we provide numerical examples to demonstrate the scalability and performance of the proposed solvers. Finally, we draw some conclusions in Section 5.

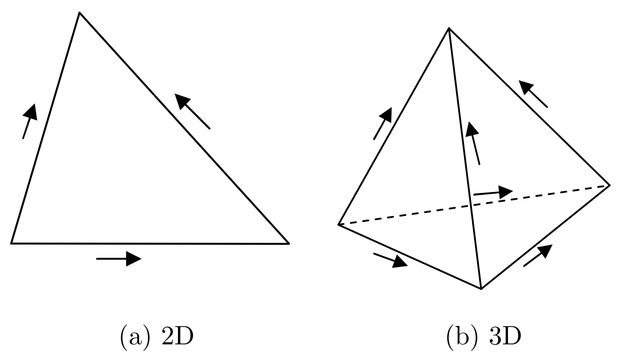

FIG. 2.1. *A graphical illustration of the degrees of freedom for the lowest order Nédélec element in 2D and 3D. Degrees of freedom are the average value of tangential component of the vector field on each edge.*

2. Finite element discretization. To discretize problem (1.2), we partition the domain Ω into shape-regular tetrahedra of a sufficiently small mesh size h. The electric field is approximated with Nédélec elements of the first family and the multiplier is approximated with nodal elements of order ℓ [17, 18]. We denote the two resulting finite element spaces by V_h and Q_h, respectively. On V_h we enforce the homogeneous boundary condition (1.2c), where as on Q_h we impose (1.2d). Figure 2.1 shows the degrees of freedom on the lowest order Nédélec elements in 2D and 3D.

Let $\langle \psi_j \rangle_{j=1}^n$ and $\langle \phi_i \rangle_{i=1}^m$ be finite element bases for the spaces V_h and Q_h respectively:

$$V_h = \text{span}\langle \psi_j \rangle_{j=1}^n, \qquad Q_h = \text{span}\langle \phi_i \rangle_{i=1}^m. \tag{2.1}$$

Then, the weak formulation of (1.2) yields a linear system of the form (1.3), see [8, 17], where the entries of the matrices and the load vector are given by

$$\begin{aligned}
A_{i,j} &= \int_\Omega \mu_r^{-1} \left(\nabla \times \psi_j\right) \cdot \left(\nabla \times \psi_i\right) \, dx, && 1 \le i, j \le n, \\
M_{i,j} &= \int_\Omega \epsilon_r \psi_j \cdot \psi_i \, dx, && 1 \le i, j \le n, \\
B_{i,j} &= \int_\Omega \epsilon_r \psi_j \cdot \nabla \phi_i \, dx, && 1 \le i \le m, \ 1 \le j \le n, \\
f_i &= \int_\Omega f \cdot \psi_i \, dx, && 1 \le i \le n.
\end{aligned}$$

Let us introduce a few additional matrices that play an important role in this formulation. First, note that $\nabla Q_h \subset V_h$, and define the matrix $C \in \mathbb{R}^{n \times m}$ by

$$\nabla \phi_j = \sum_{i=1}^n C_{i,j} \psi_i, \qquad j = 1, \ldots, m. \tag{2.2}$$

For a function $q_h \in Q_h$ given by $q_h = \sum_{j=1}^m q_j \phi_j$, we then have

$$\nabla q_h = \sum_{i=1}^n \sum_{j=1}^m C_{i,j} q_j \psi_i,$$

so Cq is the coefficient vector of ∇q_h in the basis $\langle \psi_i \rangle_{i=1}^n$. In the lowest order case, the entries of C are

$$C_{i,j} = \begin{cases} 1 & \text{if node } j \text{ is the head of edge } i, \\ -1 & \text{if node } j \text{ is the tail of edge } i, \\ 0 & \text{otherwise.} \end{cases}$$

Define the ϵ_r-weighted scalar Laplacian on Q_h as $L = (L_{i,j})_{i,j=1}^m \in \mathbb{R}^{m \times m}$ with

$$L_{i,j} = \int_\Omega \epsilon_r \nabla \phi_j \cdot \nabla \phi_i \, dx. \tag{2.3}$$

Finally, we set $Q = (Q_{i,j})_{i,j=1}^m \in \mathbb{R}^{m \times m}$ as the ϵ_r-weighted scalar mass matrix on Q_h, that is

$$Q_{i,j} = \int_\Omega \epsilon_r \phi_j \cdot \phi_i \, dx.$$

Let us state a few stability results that extend the analysis in [8] from constant material coefficients to the variable coefficient case.

Denote by $\langle \cdot, \cdot \rangle$ the standard Euclidean inner product in $\mathbb{R}^n$ or $\mathbb{R}^m$, and by $\text{null}(\cdot)$ the null space of a matrix. For a given positive (semi)definite matrix W and a vector x, we define the (semi)norm

$$|x|_W = \sqrt{\langle Wx, x \rangle}.$$

Proposition 2.1. *The following stability properties of the matrices A and B hold:*

(i) Continuity of A:

$$|\langle Au, v \rangle| \le |u|_A |v|_A, \qquad u, v \in \mathbb{R}^n. \tag{2.4}$$

(ii) Continuity of B:

$$|\langle Bv, q \rangle| \le |v|_M |q|_L, \qquad v \in \mathbb{R}^n, \ q \in \mathbb{R}^m. \tag{2.5}$$

(iii) The matrix A is positive definite on $\text{null}(B)$ *and*

$$\langle Au, u \rangle \ge \alpha \left(|u|_A^2 + |u|_M^2 \right), \qquad u \in \text{null}(B), \tag{2.6}$$

with a stability constant α which is independent of the mesh size.

(iv) The matrix B satisfies the discrete inf-sup condition

$$\inf_{0 \ne q \in \mathbb{R}^m} \sup_{0 \ne v \in \text{null}(A)} \frac{\langle Bv, q \rangle}{|v|_M |q|_L} \ge 1. \tag{2.7}$$

Proof. The first two properties follow directly from the Cauchy-Schwarz inequality.

To show (iii), we first recall the discrete Poincaré–Friedrichs inequality from [12, Theorem 4.7]. Let $u \in \text{null}(B)$ and let u_h be the associated finite element function. Then, we have

$$\int_\Omega |\nabla \times u_h|^2 \, dx \ge \beta \int_\Omega |u_h|^2 \, dx,$$

where $\beta > 0$ is independent of the mesh size.

Consequently, we bound $\langle Au, u\rangle$ as follows

$$\begin{aligned}
\langle Au, u\rangle &= \frac{1}{2}\langle Au, u\rangle + \frac{1}{2}\langle Au, u\rangle \\
&= \frac{1}{2}|u|_A^2 + \frac{1}{2}\int_\Omega \mu_r^{-1}|\nabla\times u_h|^2\,dx \\
&\geq \frac{1}{2}|u|_A^2 + \frac{1}{2\mu_{\max}}\int_\Omega |\nabla\times u_h|^2\,dx \\
&\geq \frac{1}{2}|u|_A^2 + \frac{\beta}{2\mu_{\max}}\int_\Omega |u_h|^2\,dx \\
&\geq \frac{1}{2}|u|_A^2 + \frac{\beta}{2\mu_{\max}\epsilon_{\max}}\int_\Omega \epsilon_r|u_h|^2\,dx \\
&= \frac{1}{2}|u|_A^2 + \frac{\beta}{2\mu_{\max}\epsilon_{\max}}|u|_M^2, \\
&\geq \min\left(\frac{1}{2}, \frac{\beta}{2\mu_{\max}\epsilon_{\max}}\right)(|u|_A^2 + |u|_M^2) = \alpha(|u|_A^2 + |u|_M^2),
\end{aligned}$$

where $\alpha = \min\left(\frac{1}{2}, \frac{\beta}{2\mu_{\max}\epsilon_{\max}}\right)$. Note that, since $\langle Au, u\rangle = |u|_A^2$, we must have $0 < \alpha < 1$, and then also

$$|u|_A^2 \geq \bar{\alpha}|u|_M^2, \qquad u \in \mathrm{null}(B), \tag{2.8}$$

with

$$\bar{\alpha} = \frac{\alpha}{1-\alpha}. \tag{2.9}$$

To prove (iv), let $0 \neq q_h \in Q_h$ and v be the coefficient vector of $v_h = \nabla q_h$ in the basis $\langle\psi_i\rangle_{i=1}^n$. Then it follows that $v \in \mathrm{null}(A)$ and

$$\begin{aligned}
\sup_{0\neq v\in\mathrm{null}(A)} \frac{\langle Bv, q\rangle}{|v|_M|q|_L} &= \sup_{0\neq v\in\mathrm{null}(A)} \frac{\int_\Omega \epsilon_r v_h\cdot\nabla q_h\,dx}{(\int_\Omega \epsilon_r v_h\cdot v_h dx)^{\frac{1}{2}}|q|_L} \\
&\geq \frac{\int_\Omega \epsilon_r\nabla q_h\cdot\nabla q_h\,dx}{(\int_\Omega \epsilon_r\nabla q_h\cdot\nabla q_h dx)^{\frac{1}{2}}|q|_L} = \frac{|q|_L^2}{|q|_L^2} = 1,
\end{aligned}$$

which shows (iv). □

The properties stated in Proposition 2.1 and the theory of mixed finite element methods [3, Chapter 2] ensure that the saddle point system (1.3) is invertable (provided that the mesh size is sufficiently small).

3. The solver.

3. The solver. To iteratively solve the saddle point system (1.3) we use MINRES as an *outer solver*. This is discussed in Section 3.1. To solve each outer iteration, we apply an *inner solver* based on [13] and presented in Section 3.2. In Section 3.3, we outline the complete solution procedure.

3.1. The outer solver. Following the analysis for constant coefficients in [8], we propose the following block diagonal preconditioner to iteratively solve (1.3):

$$\mathcal{P}_{M,L} = \begin{pmatrix} \mathcal{P}_M & 0 \\ 0 & L \end{pmatrix}, \tag{3.1}$$

where

$$\mathcal{P}_M = A + \gamma M \tag{3.2}$$

and

$$\gamma = 1 - k^2 > 0. \tag{3.3}$$

We have the following result:

THEOREM 3.1. *The preconditioned matrix $\mathcal{P}_{M,L}^{-1}\mathcal{K}$ has two eigenvalues $\lambda_+ = 1$ and $\lambda_- = -\frac{1}{1-k^2}$, each with algebraic multiplicity m. The remaining eigenvalues satisfy the bound*

$$\frac{\bar{\alpha} - k^2}{\bar{\alpha} + 1 - k^2} < \lambda < 1, \tag{3.4}$$

where $\bar{\alpha}$ is the constant in (2.9).

The proof follows along the lines of [8, Theorem 5.2].

3.2. The inner solver. The overall computational cost of using $\mathcal{P}_{M,L}$ depends on the ability to efficiently solve linear systems whose associated matrices are $\mathcal{P}_M$ in (3.2) and L in (2.3).

The linear system L arises from a standard scalar elliptic problem, for which many efficient solution methods exist. On the other hand, efficiently inverting $\mathcal{P}_M$ is the computational bottleneck in the inner iteration. Recently, Hiptmair and Xu proposed effective auxiliary space preconditioners for linear systems arising from conforming finite element discretizations of $H(\text{curl})$-elliptic variational problems [13], based on fictitious spaces as developed in [9, 22]. The preconditioner is:

$$\mathcal{P}_V^{-1} = \text{diag}(\mathcal{P}_M)^{-1} + P(\bar{L} + \gamma\bar{Q})^{-1}P^T + \gamma^{-1}C(L^{-1})C^T, \tag{3.5}$$

with γ as in (3.3). The matrix $\bar{L} = \text{diag}(L, L, L)$ is the ϵ_r-weighted vector Laplacian on Q_h^3, $\bar{Q} = \text{diag}(Q, Q, Q)$ is the ϵ_r-weighted vector mass matrix on Q_h^3, C is the null-space matrix in (2.2), and P is the matrix representation of the nodal interpolation operator $\Pi_h^{\text{curl}} : Q_h^3 \to V_h$. In the lowest order case, the operator Π_h^{curl} is based on path integrals along edges; for a finite element function $w_h \in Q_h^3$ it is given by

$$\Pi_h^{\text{curl}} w_h = \sum_j \left(\int_{e_j} w_h \cdot d\vec{s} \right) \psi_j,$$

where e_j is the interior edge associated with the basis function ψ_j. We have $P = [P^{(1)}, P^{(2)}, P^{(3)}]$, where $P \in \mathbb{R}^{n \times 3m}$ and $P^{(k)}$ are matrices in $\mathbb{R}^{n \times m}$. The entries of $P^{(k)}$ are given by

$$P_{i,j}^{(k)} = \begin{cases} 0.5\, d_i\, t_i^{(k)} & \text{if node } j \text{ is the head/tail of edge } i, \\ 0 & \text{otherwise,} \end{cases}$$

where $t_i^{(k)}$ is the kth component of the unit tangential vector on edge i, and d_i is the length of edge i.

In the constant coefficient case, it was shown in [13, Theorem 7.1] that for $0 < \gamma \leq 1$, the spectral condition number $\kappa_2(\mathcal{P}_V^{-1}\mathcal{P}_M)$ is independent of the mesh size. Even though there seems to be no theoretical analysis available for the variable coefficient case, the preconditioner $\mathcal{P}_V$ was experimentally shown to be effective in this case as well [13].

3.3. Solution algorithm. We run preconditioned MINRES as the outer solver for the linear system (1.3). The preconditioner is the block diagonal matrix $\mathcal{P}_{M,L}$, defined in (3.1). For each outer iteration, we need to solve a linear system of the form

$$\begin{pmatrix} \mathcal{P}_M & 0 \\ 0 & L \end{pmatrix} \begin{pmatrix} v \\ q \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}. \tag{3.6}$$

Two Krylov subspace solvers are applied as the inner iterations. The linear system associated with the $(1,1)$ block,

$$\mathcal{P}_M v = c, \tag{3.7}$$

is solved using conjugate gradient (CG) with the preconditioner $\mathcal{P}_V$, which is defined in (3.5). In each CG iteration, we need to solve a linear system of the form

$$\mathcal{P}_V w = r. \tag{3.8}$$

Following (3.5), this can be done by solving the two linear systems

$$(\bar{L} + \gamma\bar{Q})y = s, \tag{3.9a}$$
$$\bar{L}z = t, \tag{3.9b}$$

where $s = P^T r$ and $t = C^T r$. We run one AMG V-cycle to compute y and z, and we set

$$w = \operatorname{diag}(\mathcal{P}_M)^{-1} r + Py + \gamma^{-1} Cz. \tag{3.10}$$

The linear system associated with the $(2,2)$ block of (3.6),

$$Lq = d, \tag{3.11}$$

is solved using CG with an AMG preconditioner.

Our approach is summarized in Algorithm 1. The inner iteration for (3.7) is initialized in line 4 and laid out in lines 5–10, where CG iterations preconditioned with $\mathcal{P}_V$ are used. The inner iteration for (3.11) is initialized in line 11 and provided in lines 12–15, where a CG scheme with an AMG preconditioner is used. Once the two iterative solvers converge, we update the approximated solution x for the next outer iterate in line 16.

4. Numerical experiments.

This section is devoted to assessing the numerical performance and parallel scalability of our implementation on different test cases. In all experiments, the relative residual of the outer iteration is set to 1e-6 and the relative residual of the inner iteration is set to 1e-8, unless explicitly specified. The code is executed on a cluster with up to 12 nodes. Each node has eight 2.6GHz Intel processors and 16G RAM.

The following notation is used to record our results: np denotes the number of processors of the run, its is the number of outer MINRES iterations, its_{i_1} is the number of inner CG iterations for solving $\mathcal{P}_M$, its_{i_2} is the number of CG iterations for solving L, while t_s and t_a denote the average times needed for the assemble phase and the solve phase in seconds, respectively. The parameter t_{AMG} is the time spent in seconds in one BoomerAMG V-cycle for solving L.

Algorithm 1 Solve $\mathcal{K}x = b$ in (1.3)

```
1: initialize MINRES for (1.3)
2: while MINRES not converged do
3:   set $c$, $d$ to be the right-hand-side for the current inner iteration; see (3.6)
4:   initialize CG for (3.7)
5:   while CG not converged do
6:     run one AMG V-cycle to approximate $(\bar{L} + \gamma\bar{Q})^{-1}$ and update $y$ in (3.9a)
7:     run one AMG V-cycle to approximate $L^{-1}$ and update $z$ in (3.9b)
8:     update $w$ in (3.8), using (3.10)
9:     update $v$ in (3.7)
10:  end while
11:  initialize CG for (3.11)
12:  while CG not converged do
13:    apply AMG preconditioner to approximate $L$
14:    update $q$ in (3.11)
15:  end while
16:  update $x$ in (1.3)
17: end while
```

4.1. Example 1. The first example is a simple domain with a structured mesh. The domain is a cube, $\Omega = (-1, 1)^3$. We test both homogeneous and inhomogeneous coefficient cases. In the homogeneous case, we set $\mu_r = \epsilon_r = 1$. In the variable coefficient case, we assume that there are eight subdomains in the cube as shown in Figure 4.1 and each subdomain has piecewise constant coefficients. The coefficients are

$$\mu_r = \epsilon_r = \begin{cases} 1a & \text{if } x < 0 \text{ and } y < 0 \text{ and } z < 0, \\ 2a & \text{if } x > 0 \text{ and } y < 0 \text{ and } z < 0, \\ 3a & \text{if } x < 0 \text{ and } y > 0 \text{ and } z < 0, \\ 4a & \text{if } x > 0 \text{ and } y > 0 \text{ and } z < 0, \\ 5a & \text{if } x < 0 \text{ and } y < 0 \text{ and } z > 0, \\ 6a & \text{if } x > 0 \text{ and } y < 0 \text{ and } z > 0, \\ 7a & \text{if } x < 0 \text{ and } y > 0 \text{ and } z > 0, \\ 8a & \text{otherwise,} \end{cases} \tag{4.1}$$

where a is a constant. We set the right-hand side so that the solution of (1.2) is given by

$$u(x, y, z) = \begin{pmatrix} u_1(x, y, z) \\ u_2(x, y, z) \\ u_3(x, y, z) \end{pmatrix} = \begin{pmatrix} (1 - y^2)(1 - z^2) \\ (1 - x^2)(1 - z^2) \\ (1 - x^2)(1 - y^2) \end{pmatrix} \tag{4.2}$$

and

$$p(x, y, z) = (1 - x^2)(1 - y^2)(1 - z^2). \tag{4.3}$$

In this example, the homogeneous boundary conditions in (1.1) are satisfied.

Uniformly refined meshes are constructed shown in Figure 4.2(a). The number of elements and matrix sizes are given in Table 4.1. Figure 4.2(b) shows how Grid C1 is partitioned across 3 processors. Elements with the same color are stored on the

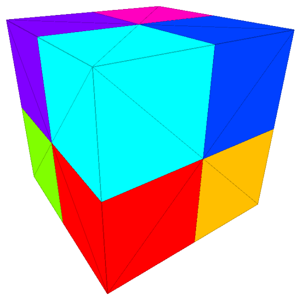

FIG. 4.1. *Example 1. Distribution of material coefficients.*

same processor. Elements with the same color are clustered together, which means the communication cost is minimal. Table 4.2 shows the local numbers of elements and degrees of freedom on each processor for Grid C1. The number of degrees of freedom on each processor is roughly the same, which indicates the load is balanced.

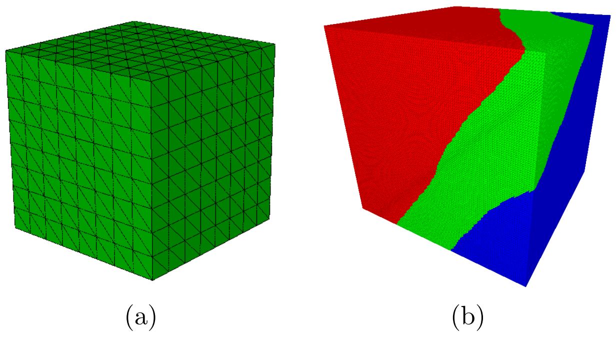

FIG. 4.2. *Example 1. (a) Structured mesh. (b) Grid C1 partitioned on 3 processors.*

Grid	Nel	$n+m$
C1	7,146,096	9,393,931
C2	14,436,624	19,034,163
C3	29,478,000	38,958,219

TABLE 4.1
Example 1. Number of elements (Nel) and the size of the linear systems ($n+m$) for Grids C1–C3.

processor	local elements	local DOFs
1	2,359,736	3,119,317
2	2,424,224	3,199,406
3	2,362,136	3,075,208

TABLE 4.2
Example 1. Partitioning of Grid C1.

In our first experiment, material coefficients are homogeneous. Scalability results are shown in Table 4.3 and results with different values of k are shown in Table 4.4.

To test scalability, we refine the mesh and increase the number of processors in a proportional manner, so that the problem size per processor remains constant. Full scalability would then imply that the computation time also remains constant. We observe in Table 4.3 that when the mesh is refined, the numbers of outer and inner iterations stay constant, which demonstrates the scalability of our method. The time spent in the assembly also scales very well. The time spent in the solve increases slightly. This is because each BoomerAMG V-cycle seems to take more time when the mesh is refined. For different values of k, we have observed very similar computation times as in Table 4.3. Table 4.4 shows that the iteration counts stay the same for the values of k that we have selected. This demonstrates the scalability of our solver.

np	Grid	its	its_{i_1}	its_{i_2}	t_s(sec)	t_a(sec)	t_{AMG}(sec)
3	C1	5	34	7	1,473.58	44.83	16.03
6	C2	5	35	9	1,634.17	45.26	20.55
12	C3	5	34	9	1,879.06	48.93	25.39

TABLE 4.3
Example 1. Iteration counts and computation times for various grids, $k = 0$.

		$k=0$			$k=\frac{1}{8}$			$k=\frac{1}{4}$		
np	Grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
3	C1	5	34	7	5	34	7	5	34	7
6	C2	5	35	9	5	35	9	5	35	9
12	C3	5	34	9	5	34	9	5	34	9

TABLE 4.4
Example 1. Iteration counts for various values of k.

Setting the outer tolerance as 1e-6, we test our solver with different inner tolerances. The results are given in Table 4.5. We see that when the inner tolerance is looser, fewer inner iterations are required, however if the tolerance is too loose, more outer iterations are required. For this example, 1e-6 is the optimal inner tolerance.

In the remaining examples, we stick to 1e-6 and 1e-8 as outer and inner tolerances respectively. We select a tight inner tolerance since one of our goals is to investigate the speed of convergence of outer iterations.

inner tol	its	its_{i_1}	its_{i_2}	t_s(sec)
1e-10	5	43	9	557.08
1e-9	5	39	8	505.30
1e-8	5	35	8	440.92
1e-7	5	30	7	383.34
1e-6	6	21	7	375.91
1e-5	8	20	6	409.57
1e-4	21	16	5	794.15

TABLE 4.5
Example 1. Iteration counts and computation times for Grid C1 on 16 processors for inexact inner iterations, $k = 0$.

Next we test the variable coefficient case. Table 4.6 shows the iteration counts for different variable coefficient cases. Note that the larger a is, the more variant the coefficients are in different regions. As expected, the eigenvalue bound depends on the coefficients. Table 4.6 shows that as a increases, the material coefficients vary more dramatically, the eigenvalue bounds in Theorem 3.1 get looser and the iteration counts increase. In the variable coefficient case, both the inner and outer iterations are not sensitive to changes of the mesh size, which demonstrates that our method is scalable in the variable coefficient case.

		$a = 1$			$a = 10$			$a = 20$		
np	Grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
3	C1	10	81	8	58	147	8	107	135	8
6	C2	10	87	10	59	154	10	105	145	10
12	C3	10	87	11	58	178	10	107	187	10

TABLE 4.6
Example 1. Iteration counts for various values of a, $k = 0$.

4.2. Example 2. In this example, we test the problem in a complicated domain with a quasi-uniform mesh. The domain is a complicated 3D gear, which is bounded in $(0.025, 0.975) \times (0.025, 0.975) \times (0.025, 0.15292)$. We test two different cases: constant and variable coefficient cases. In the constant coefficient test, we set $\mu_r = \epsilon_r = 1$. In the variable coefficient case, we assume that there are four subdomains in the gear as shown in Figure 4.3 and each subdomain has piecewise constant coefficients. The coefficients are

$$\mu_r = \epsilon_r = \begin{cases} 1a & \text{if } x < 0.5 \text{ and } y < 0.5, \\ 2a & \text{if } x > 0.5 \text{ and } y < 0.5, \\ 3a & \text{if } x < 0.5 \text{ and } y > 0.5, \\ 4a & \text{otherwise,} \end{cases}$$

where a is a constant. In both tests, we set the right-hand side function so that the exact solution is given by (4.2) and (4.3), and enforce the inhomogeneous boundary conditions in a standard way.

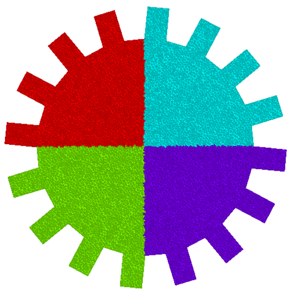

FIG. 4.3. *Example 2. Distribution of material coefficients.*

The domain is meshed with quasi-uniform tetrahedra as shown in Figure 4.4(a). The number of elements and matrix sizes are given in Table 4.7. Figure 4.4(b) shows how to partition G1 across 4 processors.

First, we test our approach with constant coefficients. The scalability results are shown in Table 4.8. Results with different values of k are shown in Table 4.9. We observe that when the mesh is refined, both the inner and outer iteration counts stay

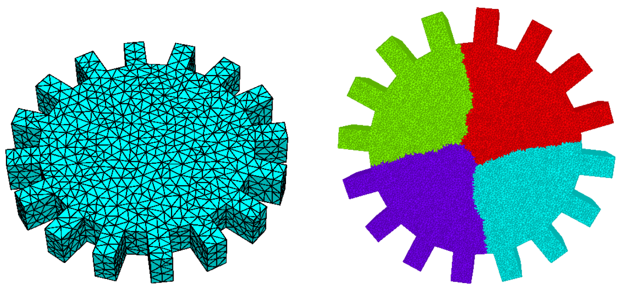
(a) (b)

FIG. 4.4. *Example 2. (a) Quasi-uniform mesh on the gear. (b) Grid G1 partitioned on 4 processors.*

Grid	Nel	$n+m$
G1	723,594	894,615
G2	1,446,403	1,810,413
G3	2,889,085	3,650,047
G4	5,778,001	7,354,886

TABLE 4.7
Example 2. Number of elements (Nel) and the size of the linear systems ($n+m$) for Grids G1–G4.

constant. Again, the time spent in the assembly scales very well. The time spent in the solve is increasing slightly, which can be explained by the increase in t_{AMG}. We note that the computational times for different values of k are similar to those reported for $k=0$ in Table 4.8. Table 4.9 shows that the iteration counts do not change for different k values.

np	Grid	its	its_{i_1}	its_{i_2}	t_s(sec)	t_a(sec)	t_{AMG}(sec)
12	G1	4	58	8	72.11	2.50	0.42
24	G2	4	61	9	102.27	2.56	0.67
48	G3	4	64	9	138.87	2.55	1.05
96	G4	4	66	10	190.19	2.68	1.52

TABLE 4.8
Example 2. Iteration counts and computation times for various grids, $k=0$.

		$k=0$			$k=\frac{1}{8}$			$k=\frac{1}{4}$		
np	Grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
12	G1	4	58	8	4	58	8	4	58	8
24	G2	4	61	9	4	61	9	4	61	9
48	G3	4	64	9	4	64	9	4	64	9
96	G4	4	66	10	4	66	10	4	66	10

TABLE 4.9
Example 2. Iteration counts for various values of k.

Next we test the variable coefficient example. Table 4.10 shows the results for different variable coefficient cases. Again, we observe that when the variation of the material coefficients increases, the iteration counts increase. We also observe that both the inner and outer iterations are not sensitive to changes of the mesh size for the example with unstructured meshes.

		$a = 1$			$a = 10$			$a = 20$			$a = 50$		
np	Grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
12	G1	5	160	8	21	291	8	35	251	8	71	141	8
24	G2	5	167	9	20	332	9	35	295	9	72	169	9
48	G3	5	177	9	20	391	9	34	303	9	69	192	9
96	G4	5	186	10	20	432	10	34	408	10	71	230	10

TABLE 4.10
Example 2. Iteration counts for various values of a, $k = 0$.

		$\epsilon_r = 10$			$\epsilon_r = 100$			$\epsilon_r = 1000$		
np	Grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
12	G1	5	141	8	9	258	8	19	210	8
24	G2	5	148	9	9	284	9	19	274	9
48	G3	5	157	9	9	316	9	19	325	9
96	G4	5	164	10	9	340	10	19	387	10

TABLE 4.11
Example 2. Iteration counts for various values of ϵ_r, $\mu_r = 1$, $k = 0$.

4.3. Example 3. The last example is the Fichera corner problem. We are interested in testing our solver on a series of locally refined meshes. The domain $\Omega = (-1, 1)^3 \backslash [0, 1) \times [0, -1) \times [0, 1)$ is a cube with a missing corner. We also test both homogeneous and inhomogeneous coefficients. In the homogeneous coefficient case, we set $\mu_r = \epsilon_r = 1$. In the inhomogeneous case, we assume that there are seven subdomains in the domain as shown in Figure 4.5 and each subdomain has piecewise constant coefficients. The coefficients are the same as (4.1). In both tests, we set the right-hand side function so that the exact solution is given by (4.2) and (4.3), and enforce the inhomogeneous boundary conditions in a standard way.

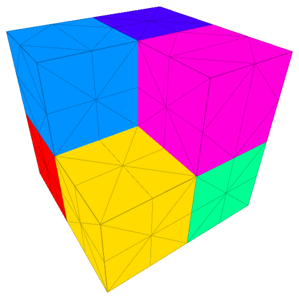

FIG. 4.5. *Example 3. Distribution of material coefficients.*

The domain is discretized with locally refined meshes towards the corner. Figure 4.6 shows an example of a sequence of locally refined meshes. The number of elements and matrix sizes are given in Table 4.12. Figure 4.7 shows the partitioning of F1 on 4 processors.

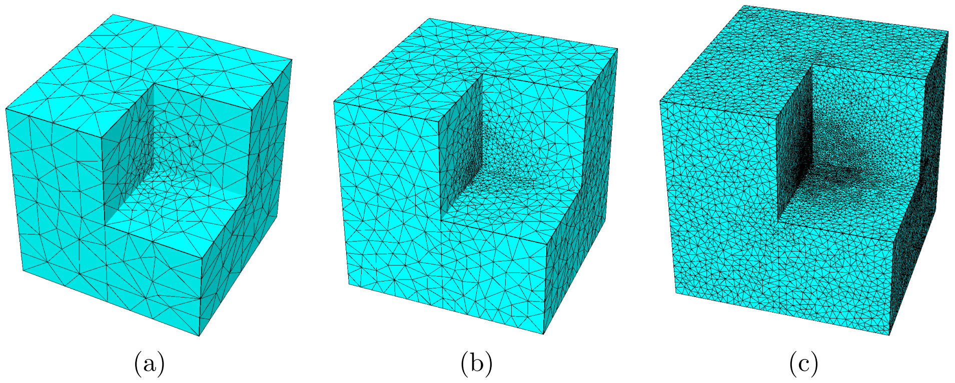

FIG. 4.6. *Example 3. A sequence of locally refined meshes.*

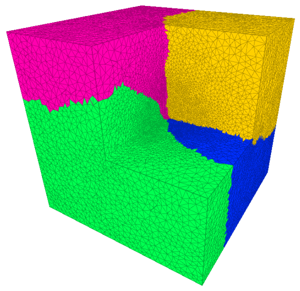

FIG. 4.7. *Example 3. Illustration of Grid F1 partitioned on 4 processors.*

Grid	Nel	$n+m$
F1	781,614	957,277
F2	1,543,937	1,917,649
F3	3,053,426	3,832,895
F4	6,072,325	7,689,953

TABLE 4.12
Example 3. Number of elements (Nel) and the size of the linear systems ($n+m$) for Grids F1–F4.

First, we assume that the material coefficients are homogeneous. The scalability results are shown in Table 4.13. The results with different values of k are shown in Table 4.14. On the locally refined meshes, we also observe that when the mesh is refined, both the inner and outer solvers are scalable. Again, the time spent in the assembly scales very well. The time spent in the solve is increasing, which is due to the increasing cost of BoomerAMG V-cycles. Table 4.14 shows that the iteration counts do not change for different wave numbers. As in the previous examples, in our experiments the computational times are also roughly the same as in Table 4.13 for different values of k.

np	Grid	its	its_{i_1}	its_{i_2}	t_s(sec)	t_a(sec)	t_{AMG}(sec)
4	F1	5	59	8	204.82	6.54	0.88
8	F2	5	56	9	213.16	6.39	1.17
16	F3	5	58	10	253.81	6.29	1.65
32	F4	5	62	10	314.88	6.38	1.99

TABLE 4.13

Example 3. Iteration counts and computation times for various grids, $k = 0$.

		$k = 0$			$k = \frac{1}{8}$			$k = \frac{1}{4}$		
np	Grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
4	F1	5	59	8	5	59	8	5	59	8
8	F2	5	56	9	5	56	9	5	56	9
16	F3	5	58	10	5	58	10	5	58	10
32	F4	5	62	10	5	62	10	4	63	10

TABLE 4.14

Example 3. Iteration counts for various values of k.

Next, we test the variable coefficient case. Table 4.15 shows the results for different variable coefficient cases. Again, we observe that when the coefficients vary more, the iteration counts increase, but the scalability is very good.

		$a = 1$			$a = 10$			$a = 20$		
np	Grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
4	F1	7	85	9	40	157	9	74	134	9
8	F2	7	92	9	40	178	9	73	157	9
16	F3	7	100	10	39	192	10	72	167	10
32	F4	7	107	10	39	216	10	71	206	10

TABLE 4.15

Example 3. Iteration counts for various values of a, $k = 0$.

5. Conclusions.

We have developed and implemented a fully scalable parallel iterative solver for the time-harmonic Maxwell equations with variable coefficients, on unstructured complicated domains with heterogeneous coefficients. Our code for preconditioned iterative solvers is specialized, dealing with the specific features of the discretization, such as the use of edge elements. We use state-of-the-art software packages (PETSc [1], BoomerAMG [10], Hypre [6], METIS [15]) to optimize the performance of our solvers. We have also developed our own mesher for structured meshes, and we use TetGen for unstructured and locally refined meshes.

The preconditioned solvers are based on inner-outer iterations. For outer iterations we adapt [8] to the variable coefficient case. The inner iterations are based on the auxiliary spaces technique developed in [13].

We have shown that the the preconditioned matrix is well conditioned and its eigenvalues are tightly clustered; this is key to the effectiveness of the proposed approach. As our extensive numerical experiments indicate, the inner and outer iterations are scalable in terms of iteration counts and computation times, and the solver is minimally sensitive to changes in the mesh size. Highly varying coefficients result in higher iteration counts, but those barely change as the mesh is refined.

Our results apply to low wave numbers; dealing with high wave numbers is a primary challenge of much interest and remains an item for future work.

REFERENCES

[1] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. PETSc users manual. Technical Report ANL-95/11 - Revision 2.3.2, Argonne National Laboratory, 2006.

[2] P. Bochev, C. Garasi, J. Hu, A. Robinson, and R. Tuminaro. An improved algebraic multigrid method for solving Maxwell's equations. *SIAM Journal on Scientific Computing*, 25:623–642, 2003.

[3] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Springer-Verlag New York, Inc., New York, NY, USA, 1991.

[4] Z. Chen, Q. Du, and J. Zou. Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients. *SIAM Journal on Numerical Analysis*, 37(5):1542–1570, 2000.

[5] L. Demkowicz and L. Vardapetyan. Modeling of electromagnetic absorption/scattering problems using *hp*–adaptive finite elements. 152:103–124, 1998.

[6] R. Falgout, A. Cleary, J. Jones, E. Chow, V. Henson, C. Baldwin, P. Brown, P. Vassilevski, and U. M. Yang. Hypre: high performace preconditioners. http://acts.nersc.gov/hypre/, 2010.

[7] J. Gopalakrishnan, J. E. Pasciak, and L. F. Demkowicz. Analysis of a multigrid algorithm for time harmonic maxwell equations. *SIAM Journal on Numerical Analysis*, 42:90–108, 2004.

[8] C. Greif and D. Schötzau. Preconditioners for the discretized time-harmonic Maxwell equations in mixed form. *Numerical Linear Algebra with Applications*, 14(4):281–297, 2007.

[9] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numerische Mathematik*, 70(2):163–180, 1995.

[10] V. E. Henson and U. M. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41:155–177, 2000.

[11] R. Hiptmair. Multigrid method for Maxwell's equations. *SIAM Journal on Numerical Analysis*, 36(1):204–225, 1998.

[12] R. Hiptmair. Finite elements in computational electromagnetism. *Acta Numerica*, 11(237-339), 2002.

[13] R. Hiptmair and J. Xu. Nodal auxiliary space preconditioning in H(curl) and H(div) spaces. *SIAM Journal on Numerical Analysis*, 45(6):2483–2509, 2007.

[14] J. Jones and B. Lee. A multigrid method for variable coefficient Maxwell's equations. *SIAM Journal on Scientific Computing*, 27(5):1689–1708, 2006.

[15] G. Karypis. METIS: Family of multilevel partitioning algorithms. http://glaros.dtc.umn.edu/gkhome/views/metis/, 2010.

[16] T. V. Kolev and P. S. Vassilevski. Parallel auxiliary space AMG for H(curl) problems. *Journal of Computational Mathematics*, 27(5):604–623, 2009.

[17] P. Monk. *Finite Element Methods for Maxwell's Equations.* Oxford Science Publications, 2003.

[18] J. C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35:315–341, 1980.

[19] I. Perugia, D. Schötzau, and P. Monk. Stabilized interior penalty methods for the time-harmonic Maxwell equations. *Computer Methods in Applied Mechanics and Engineering*, 191:4675–4697, 2002.

[20] S. Reitzinger and J. Schöberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numerical Linear Algebra with Applications*, 9(3):223–238, 2002.

[21] L. Vardapetyan and L. Demkowicz. *hp*-adaptive finite elements in electromagnetics. *Computer Methods in Applied Mechanics and Engineering*, 169:331–344, 1999.

[22] J. Xu. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. *Computing*, 56(3):215–235, 1996.