Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **4**, No. 2, pp. 175-189

DOI: 10.4208/aamm.10-m11166
April 2012

An Iterative Two-Grid Method of A Finite Element PML Approximation for the Two Dimensional Maxwell Problem

Chunmei Liu[1], Shi Shu[1,*], Yunqing Huang[1], Liuqiang Zhong[2] and Junxian Wang[1]

[1] *Hunan Key Laboratory for Computation & Simulation in Science and Engineering and Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, Xiangtan University, Hunan 411105, China*
[2] *School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China*

Received 21 June 2011; Accepted (in revised version) 6 January 2012

Available online 26 March 2012

Abstract. In this paper, we propose an iterative two-grid method for the edge finite element discretizations (a saddle-point system) of Perfectly Matched Layer(PML) equations to the Maxwell scattering problem in two dimensions. Firstly, we use a fine space to solve a discrete saddle-point system of $H(grad)$ variational problems, denoted by auxiliary system 1. Secondly, we use a coarse space to solve the original saddle-point system. Then, we use a fine space again to solve a discrete $\boldsymbol{H}(curl)$-elliptic variational problems, denoted by auxiliary system 2. Furthermore, we develop a regularization diagonal block preconditioner for auxiliary system 1 and use *H-X* preconditioner for auxiliary system 2. Hence we essentially transform the original problem in a fine space to a corresponding (but much smaller) problem on a coarse space, due to the fact that the above two preconditioners are efficient and stable. Compared with some existing iterative methods for solving saddle-point systems, such as PMinres, numerical experiments show the competitive performance of our iterative two-grid method.

AMS subject classifications: 65F10, 65N30, 78A46
Key words: Maxwell scattering, edge finite element, PML, iterative two-grid method.

1 Introduction

An early paper of Bérenger [3] proposed a perfectly matched layer (PML) method for

*Corresponding author.
URL: `http://math.xtu.edu.cn/myphp/math/personal/shushi/`
Email: liuchunmei8080@yahoo.com.cn (C. Liu), shushi@xtu.edu.cn (S. Shu), huangyq@xtu.edu.cn (Y. Huang), zhong@scnu.edu.cn (L. Zhong), xianxian.student@sina.com (J. Wang)

time-dependent Maxwell equations. The idea is to construct a fictitious absorbing layer outside the "region of interest" so that plane waves passed into the layer without reflection. This approach was then applied to various time domain problems (cf. [1, 4, 8, 9]). PML methods were also developed in terms of a complex change of variable (or stretching) for frequency domain Maxwell problems (cf. [5, 10, 17]). Especially, Bramble and Pasciak [5] have proved existence and uniqueness of the solutions to the infinite domain and truncated domain PML equations provided that the truncated domain is sufficiently large. Furthermore, they also showed the PML reformulation preserves the solution in the layer while decaying exponentially outside of the layer. However, the corresponding edge finite element discretization is an indefinite saddle-point system which is usually large and higher ill-condition. Hence constructing the corresponding fast algorithms is necessary for realistic computational electromagnetism.

Nowadays, there are only few research results for fast algorithms of PML equations. For example, Botros and Volakis [6] presented a generalized minimal residual (GMRES) solver which coupled with an approximate inverse preconditioner. Botros and Volakis [7] given an optimal selection of the PML parameters and tested the GMRES solver.

The two-grid methods are developed originally for nonsymmetric or indefinite linear elliptic partial differential equations (PDEs) [15, 18–20], and the basic idea is first to solve the original problem in a coarse mesh space with mesh size H and then solve a symmetric positive definite (SPD) problem on a fine mesh space with mesh size h. This method was later extended to other problems (cf. [13,16,21,22]). However, the extension of the two grid method to the Maxwell equations is not straightforward, since the leading term *curl* for Maxwell's equations has a large kernel. Noting that the behaviors of the system PML problems in our paper are different in different regions and the parameter before the operator *curl* don't maintain sign in some regions, so the system is more complex and then the usual multigrid methods for Maxwell problem won't work.

In this paper, we will propose an iterative two-grid method for the edge finite element discretizations (a saddle-point system) of PML equations for a Maxwell scattering problem in two dimensions. Unlike the traditional two-grid method for elliptic problems, we need to take care of the kernel of operator *curl*. In detail, we first use a fine space to solve a discrete saddle-point system of $H(grad)$ variational problems, denoted by auxiliary system 1. Secondly, we use a coarse space to solve the original saddle-point system. At last, we use a fine space again to solve a discrete $\boldsymbol{H}(curl)$-elliptic variational problems, denoted by auxiliary system 2. Furthermore, we design a regularization diagonal block preconditioner for auxiliary system 1 since its algebraic system is still an indefinite saddle-point system. In view of the algebraic system of auxiliary system 2 it is a diagonal block matrix with each diagonal elements is a $\boldsymbol{H}(curl)$-elliptic operator, we choose PCG method based on H-X preconditioner [14] as a solver. Numerical experiments show that the above two preconditioners are efficient and stable. With this method, the solution of the original problem in a fine grid is

essentially reduced to the solution of a corresponding (but much smaller) problem on a coarser grid. Compared with some iterative methods to solve saddle-point systems, such as PMinres, numerical experiments show that our iterative two-grid method is much more efficient.

The rest of the paper is organized as follows. In Section 2, we introduce the Maxwell PML equations and the corresponding edge finite element discretization. In Section 3, we construct an iterative two-grid method and two preconditioners for auxiliary systems. Finally, we report several numerical examples to illustrate the competitive behavior of the method in Section 4.

2 A Model PML Problem

Let

$$\Omega_M = \Omega_1 \cup \Omega_2 \cup \Omega_3 \subset \mathbb{R}^2,$$

be a bounded computational domain with the inner boundary Γ and the outer boundary Γ_M (see Fig. 1), where

$$\Omega_1 = [-a,a]^2 \setminus [-1,1]^2, \quad \Omega_2 = [-b,b]^2 \setminus [-a,a]^2 \quad \text{and} \quad \Omega_3 = [-M,M]^2 \setminus [-b,b]^2,$$

with $1 < a < b < M$.

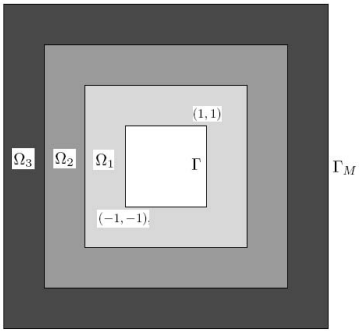


Figure 1: Computational domain Ω_M.

We define an even function $\sigma \in C^0(\mathbb{R})$ as follows (see [5])

$$\sigma(t) = \begin{cases} 0, & \text{for } |t| \le a, \\ \sigma_0(|t|-a), & \text{for } a < |t| < b, \\ \sigma_0, & \text{for } |t| \ge b. \end{cases} \tag{2.1}$$

Here $\sigma_0 > 0$ is a parameter (the PML strength). Furthermore, we need to introduce another function [5,11]

$$d(t) = 1 + i\sigma(t).$$

For any given complex function $\boldsymbol{g} \in \boldsymbol{L}^2(\Gamma)$, we define the Sobolev space

$$\boldsymbol{H}_{\boldsymbol{g}}(curl;\Omega_M) = \big\{\boldsymbol{u} : \boldsymbol{u} = \boldsymbol{u}_r + i\boldsymbol{u}_i,\ \ curl\boldsymbol{u}_r \in L^2(\Omega_M),\ \ curl\boldsymbol{u}_i \in L^2(\Omega_M),\\ \boldsymbol{u}\cdot\boldsymbol{t}|_{\Gamma_M} = \boldsymbol{0},\ \ \boldsymbol{u}\cdot\boldsymbol{t}|_{\Gamma} = \boldsymbol{g}\cdot\boldsymbol{t}|_{\Gamma}\big\}.$$

In this paper, we consider the following variational problem of PML equations [5]: find $\boldsymbol{u} \in \boldsymbol{H}_{\boldsymbol{g}}(curl;\Omega_M)$ such that

$$a(\boldsymbol{u},\boldsymbol{\psi}) = \boldsymbol{0}, \quad \forall \boldsymbol{\psi} \in \boldsymbol{H}_0(curl;\Omega_M), \tag{2.2}$$

where

$$a(\boldsymbol{u},\boldsymbol{\psi}) = \int_\Omega \boldsymbol{S}^{-1}(\nabla\times\boldsymbol{u})(\overline{\nabla\times\boldsymbol{\psi}})\mathrm{d}\boldsymbol{x} - \int_\Omega (\boldsymbol{\mu}\boldsymbol{u})\cdot\overline{\boldsymbol{\psi}}\mathrm{d}\boldsymbol{x}.$$

Here

$$\boldsymbol{S} = d(x)d(y), \qquad \boldsymbol{\mu} = \mathrm{diag}\Big(\frac{d(y)}{d(x)}, \frac{d(x)}{d(y)}\Big).$$

For simplicity, let

$$\boldsymbol{S}^{-1} = S^r + iS^i, \qquad \boldsymbol{\mu} = D^r + iD^i.$$

We also write complex function $\boldsymbol{u}$ as $\boldsymbol{u} = \boldsymbol{u}_r + i\boldsymbol{u}_i$ for the purpose of computation, and still use the notation $\boldsymbol{H}_{\boldsymbol{g}}(curl;\Omega_M)$ for the Sobolev space of real functions.

A straightforward computation gives the equivalent weak formulation of (2.2): Find $\boldsymbol{u}_r \in \boldsymbol{H}_{\boldsymbol{g}_r}(curl;\Omega_M)$, $\boldsymbol{u}_i \in \boldsymbol{H}_{\boldsymbol{g}_i}(curl;\Omega_M)$ such that

$$a(\boldsymbol{u}_r,\boldsymbol{u}_i;\boldsymbol{\psi}_r,-\boldsymbol{\psi}_i) = \boldsymbol{0}, \quad \forall \boldsymbol{\psi}_r,\boldsymbol{\psi}_i \in \boldsymbol{H}_0(curl;\Omega_M), \tag{2.3}$$

where

$$\begin{aligned} a(\boldsymbol{u}_r,\boldsymbol{u}_i;\boldsymbol{\psi}_r,-\boldsymbol{\psi}_i) = &\int_{\Omega_M} (S^r(\nabla\times\boldsymbol{u}_r)(\nabla\times\boldsymbol{\psi}_r) - S^r(\nabla\times\boldsymbol{u}_i)(\nabla\times\boldsymbol{\psi}_i))\mathrm{d}\boldsymbol{x}\\ &- \int_{\Omega_M} (S^i(\nabla\times\boldsymbol{u}_i)(\nabla\times\boldsymbol{\psi}_r) + S^i(\nabla\times\boldsymbol{u}_r)(\nabla\times\boldsymbol{\psi}_i))\mathrm{d}\boldsymbol{x}\\ &- \int_{\Omega_M} (\boldsymbol{u}_r^T D^r\boldsymbol{\psi}_r - \boldsymbol{u}_i^T D^r\boldsymbol{\psi}_i)\mathrm{d}\boldsymbol{x} + \int_{\Omega_M} (\boldsymbol{u}_i^T D^i\boldsymbol{\psi}_r + \boldsymbol{u}_r^T D^i\boldsymbol{\psi}_i)\mathrm{d}\boldsymbol{x}. \end{aligned} \tag{2.4}$$

Assume that Ω_M is covered by a quasi-uniform mesh of tetrahedron $\mathcal{T}_h$, where h is the maximum diameter of the tetrahedrons in $\mathcal{T}_h$. We introduce the lowest order edge elements of the first family

$$\mathbb{V}_{h,\boldsymbol{g}}(\mathcal{T}_h) = \big\{\boldsymbol{u}_h \in \boldsymbol{H}_{\boldsymbol{g}}(curl;\Omega_M) : \boldsymbol{u}_h|_\tau \in \boldsymbol{\mathcal{R}}_1,\ \ \forall \tau \in \mathcal{T}_h\big\}, \tag{2.5}$$

where

$$\boldsymbol{\mathcal{R}}_1 = (\mathcal{P}_0)^2 \oplus \big\{\boldsymbol{p} \in (\tilde{\mathcal{P}}_1)^2 : \boldsymbol{x}\cdot\boldsymbol{p} = 0\big\},$$

$\mathcal{P}_0$ denotes the space of constants, $\tilde{\mathcal{P}}_1$ denotes the space of homogeneous linear polynomials.

Then the algebraic system of weak formulation (2.3) can be described as

$$MX = F, \tag{2.6}$$

where

$$M = \begin{pmatrix} A & B \\ B & -A \end{pmatrix}. \tag{2.7}$$

Here the corresponding bilinear forms of matrices A and B are $a(\boldsymbol{u}_r, \mathbf{0}; \boldsymbol{\psi}_r, \mathbf{0})$ and $a(\mathbf{0}, \boldsymbol{u}_i; \boldsymbol{\psi}_r, \mathbf{0})$, respectively.

Noting that the matrix M in (2.7) is a symmetric indefinite matrix and has significantly deteriorate condition number if the mesh size becomes small. Hence convergence of the iterative solver is substantially affected and construction of fast solver is necessary.

3 An iterative two-grid method

In this section, we first introduce some preliminaries which will be used in two-grid algorithm, then present an iterative two-grid method for solving (2.6) and discuss some corresponding solvers.

Assume the coarse mesh $\mathcal{T}_H$ is a nested mesh of the fine mesh $\mathcal{T}_h$, which implies

$$\mathbb{V}_{H,\mathbf{0}}(\mathcal{T}_H) \subset \mathbb{V}_{h,\boldsymbol{g}}(\mathcal{T}_h).$$

We define the following Lagrange finite element space $S^1_{h,0}$ corresponding to $H^1_0(\Omega_M)$:

$$S_{h,0} = \big\{ v_h \in C(\Omega_M) : v_h|_{\Gamma \cup \Gamma_M} = 0, \;\; v_h|_\tau \in \mathcal{P}_1, \;\; \forall \tau \in \mathcal{T}_h \big\}.$$

From the exact sequence property of discrete finite element space, we know that

$$\mathbf{grad} S_{h,0} := \big\{ \nabla v_h : v_h \in S_{h,0} \big\},$$

is the kernel of operator *curl* in the fine space $\mathbb{V}^{1,1}_{h,\mathbf{0}}(\mathcal{T}_h)$, and

$$\mathbf{grad} S_{h,0} \subset \mathbb{V}^{1,1}_{h,\mathbf{0}}(\mathcal{T}_h),$$

holds.

In order to discuss the algorithm conveniently, let us introduce another bilinear form $a_g(\cdot,\cdot;\cdot,\cdot)$ that is defined by

$$a_g(\phi_r, \phi_i; \varphi_r, -\varphi_i) = a(\nabla\phi_r, \nabla\phi_i; \nabla\varphi_r, \nabla\varphi_i), \quad \forall \phi_r, \phi_i, \varphi_r, \varphi_i \in S_{h,0}. \tag{3.1}$$

Similar to the standard two-grid method, we need to introduce the decomposition of bilinear form $a(\cdot,\cdot;\cdot,\cdot)$ given by (2.4) as follows

$$a(\boldsymbol{u}_r,\boldsymbol{u}_i;\boldsymbol{\psi}_r,-\boldsymbol{\psi}_i)=\hat{a}(\boldsymbol{u}_r,\boldsymbol{u}_i;\boldsymbol{\psi}_r,-\boldsymbol{\psi}_i)+N(\boldsymbol{u}_r,\boldsymbol{u}_i;\boldsymbol{\psi}_r,-\boldsymbol{\psi}_i),$$

where

$$\begin{aligned}\hat{a}(\boldsymbol{u}_r,\boldsymbol{u}_i;\boldsymbol{\psi}_r,-\boldsymbol{\psi}_i)=&\int_{\Omega_M}|S^r|(\nabla\times\boldsymbol{u}_r)(\nabla\times\boldsymbol{\psi}_r)\mathrm{d}\boldsymbol{x}+\int_{\Omega_M}\boldsymbol{u}_r^TD^r\boldsymbol{\psi}_r\mathrm{d}\boldsymbol{x}\\&-\int_{\Omega_M}|S^r|(\nabla\times\boldsymbol{u}_i)(\nabla\times\boldsymbol{\psi}_i)\mathrm{d}\boldsymbol{x}-\int_{\Omega_M}\boldsymbol{u}_i^TD^r\boldsymbol{\psi}_i\mathrm{d}\boldsymbol{x},\end{aligned}\tag{3.2a}$$

$$\begin{aligned}N(\boldsymbol{u}_r,\boldsymbol{u}_i;\boldsymbol{\psi}_r,-\boldsymbol{\psi}_i)=&\int_{\Omega_M}(S^r-|S^r|)\big((\nabla\times\boldsymbol{u}_r)(\nabla\times\boldsymbol{\psi}_r)\big)\mathrm{d}\boldsymbol{x}-2\int_{\Omega_M}\boldsymbol{u}_r^TD^r\boldsymbol{\psi}_r\mathrm{d}\boldsymbol{x}\\&-\int_{\Omega_M}(S^r-|S^r|)\big((\nabla\times\boldsymbol{u}_i)(\nabla\times\boldsymbol{\psi}_i)\big)\mathrm{d}\boldsymbol{x}+2\int_{\Omega_M}\boldsymbol{u}_i^TD^r\boldsymbol{\psi}_i\mathrm{d}\boldsymbol{x}\\&-\int_{\Omega_M}(S^i(\nabla\times\boldsymbol{u}_i)(\nabla\times\boldsymbol{\psi}_r)-\boldsymbol{u}_i^TD^i\boldsymbol{\psi}_r)\mathrm{d}\boldsymbol{x}\\&+\int_{\Omega_M}(S^i(\nabla\times\boldsymbol{u}_r)(\nabla\times\boldsymbol{\psi}_i)-\boldsymbol{u}_r^TD^i\boldsymbol{\psi}_i)\mathrm{d}\boldsymbol{x}.\end{aligned}\tag{3.2b}$$

3.1 Iterative two-grid method

Using the above preparation, we present an iterative two-gird method for approximating the finite element solution of original variational problem (2.3). We also provide the corresponding algebraic system in the meantime.

Algorithm 3.1 (Iterative two-grid method)**.** *For given positive integer J, let $\boldsymbol{u}_r^0=\boldsymbol{0},\boldsymbol{u}_i^0=\boldsymbol{0}$. Assume that $\boldsymbol{u}_r^j\in\mathbb{V}_{h,\boldsymbol{g}_r}(\mathcal{T}_h)$ and $\boldsymbol{u}_i^j\in\mathbb{V}_{h,\boldsymbol{g}_i}(\mathcal{T}_h)$ have been obtained, $\boldsymbol{u}_r^{j+1}\in\mathbb{V}_{h,\boldsymbol{g}_r}(\mathcal{T}_h)$ and $\boldsymbol{u}_i^{j+1}\in\mathbb{V}_{h,\boldsymbol{g}_i}(\mathcal{T}_h)$ $(1\le j+1\le J)$ defined as follows:*

- *Find $\phi_r^j\in S_{h,0}^1$ and $\phi_i^j\in S_{h,0}^1$ such that*

$$\begin{aligned}a_g(\phi_r^j,\phi_i^j;\varphi_{r,h},-\varphi_{i,h})=&(f_{r,g},f_{i,g};\nabla\varphi_{r,h},-\nabla\varphi_{i,h})\\&-a(\boldsymbol{u}_r^j,\boldsymbol{u}_i^j;\nabla\varphi_{r,h},-\nabla\varphi_{i,h}),\quad\forall\varphi_{r,h}\in S_{h,0}^1,\ \varphi_{i,h}\in S_{h,0}^1,\end{aligned}\tag{3.3}$$

  where $a_g(\cdot,\cdot;\cdot,\cdot)$ and $a(\cdot,\cdot;\cdot,\cdot)$ are defined by (3.1) *and* (2.4)*, respectively.*

  Let

$$X_g=(X_{r,g},X_{i,g})^T\quad\text{and}\quad X^j=(X_r^j,X_i^j)^T,$$

  where $X_{r,g}^T$, $X_{i,g}^T$, $(X_r^j)^T$ and $(X_i^j)^T$ are column vectors formed by freedoms of ϕ_r^j, ϕ_i^j, $\boldsymbol{u}_r^j$ and $\boldsymbol{u}_i^j$, respectively. The algebraic system of weak formulation (3.3) *can be described as*

$$M_gX_g=\boldsymbol{P}_g(F-MX^j),\tag{3.4}$$

with

$$M_g = \begin{pmatrix} A_g & B_g \\ B_g & -A_g \end{pmatrix}, \tag{3.5}$$

$\boldsymbol{P}_g$ *is a restricted matrix from* $(\mathbb{V}_{h,\mathbf{0}}(\mathcal{T}_h))^2$ *to* $(S^1_{h,0})^2$, F *is the right side vector of* (2.6) *and the matrix* M *is defined by* (2.7).

- *Find* $\boldsymbol{e}^j_{r,H} \in \mathbb{V}_{H,\mathbf{0}}(\mathcal{T}_H)$ *and* $\boldsymbol{e}^j_{i,H} \in \mathbb{V}_{H,\mathbf{0}}(\mathcal{T}_H)$ *such that*

$$\begin{aligned} &a(\boldsymbol{e}^j_{r,H}, \boldsymbol{e}^j_{i,H}; \boldsymbol{\psi}_{r,H}, -\boldsymbol{\psi}_{i,H}) \\ =&(f_{r,H}, f_{i,H}; \boldsymbol{\psi}_{r,H}, -\boldsymbol{\psi}_{i,H}) - a(\boldsymbol{u}^j_r + \nabla\phi^j_{r,h}, \boldsymbol{u}^j_i + \nabla\phi^j_{r,h}; \boldsymbol{\psi}_{r,H}, \\ &-\boldsymbol{\psi}_{i,H}), \quad \forall \boldsymbol{\psi}_{r,H} \in \mathbb{V}_{H,\mathbf{0}}(\mathcal{T}_H), \ \ \boldsymbol{\psi}_{i,H} \in \mathbb{V}_{H,\mathbf{0}}(\mathcal{T}_H). \end{aligned} \tag{3.6}$$

Let $X_H = (X_{r,H}, X_{i,H})^T$, *where* $(X_{r,H})^T$ *and* $(X_{i,H})^T$ *are column vectors formed by freedoms of* $e^j_{r,H}$ *and* $e^j_{i,H}$, *respectively. The algebraic system of* (3.6) *can be written as*

$$M_H X_H = \boldsymbol{R}_H(F - M(\boldsymbol{P}_g^T X_g + X^j)), \tag{3.7}$$

with

$$M_H = \begin{pmatrix} A_H & B_H \\ B_H & -A_H \end{pmatrix},$$

$\boldsymbol{R}_H$ *is a restricted matrix from* $(\mathbb{V}_{h,\mathbf{0}}(\mathcal{T}_h))^2$ *to* $(\mathbb{V}_{H,\mathbf{0}}(\mathcal{T}_h))^2$.

- *Find* $\boldsymbol{u}^{j+1}_r \in \mathbb{V}_{h,\boldsymbol{g}_r}(\mathcal{T}_h)$ *and* $\boldsymbol{u}^{j+1}_i \in \mathbb{V}_{h,\boldsymbol{g}_i}(\mathcal{T}_h)$ *such that*

$$\begin{aligned} \hat{a}(\boldsymbol{u}^{j+1}_r, \boldsymbol{u}^{j+1}_i; \boldsymbol{\psi}_{r,h}, -\boldsymbol{\psi}_{i,h}) =& -N(\boldsymbol{u}^j_r + \nabla\phi^j_{r,h} + \boldsymbol{e}^j_{r,H}, \boldsymbol{u}^j_i + \nabla\phi^j_{i,h} + \boldsymbol{e}^j_{i,H}; \\ &\boldsymbol{\psi}_{r,h}, -\boldsymbol{\psi}_{i,h}), \ \ \forall \boldsymbol{\psi}_{r,h} \in \mathbb{V}_{h,\mathbf{0}}(\mathcal{T}_h), \ \ \boldsymbol{\psi}_{i,h} \in \mathbb{V}_{h,\mathbf{0}}(\mathcal{T}_h), \end{aligned} \tag{3.8}$$

where $\boldsymbol{g}_r$ *and* $\boldsymbol{g}_i$ *are the real part and the imaginary part of* $\boldsymbol{g}$, $\hat{a}(\cdot,\cdot;\cdot,\cdot)$ *and* $N(\cdot,\cdot;\cdot,\cdot)$ *defined by* (3.2a) *and* (3.2b), *respectively.*

Let $X_{j+1} = (X_{r,j+1}, X_{i,j+1})^T$, *where* $X^T_{r,j+1}$ *and* $X^T_{i,j+1}$ *are column vectors formed by freedoms of* $\boldsymbol{u}^{j+1}_r$ *and* $\boldsymbol{u}^{j+1}_i$, *respectively. The algebraic system of weak formulation* (3.8) *can be described as*

$$\hat{M} X_{j+1} = F - (M - \hat{M})(X_g + X^j + X_H), \tag{3.9}$$

here

$$\hat{M} = \begin{pmatrix} \hat{A} & \\ & -\hat{A} \end{pmatrix}, \tag{3.10}$$

is the corresponding matrix of bilinear form $\hat{a}(\cdot,\cdot;\cdot,\cdot)$.

In Algorithm 3.1, we need to solve three variational problems (3.3), (3.6) and (3.8). Although the variational problem (3.6) on a coarse mesh space is an original weak formulation, we can use some iterative methods (e.g., PMinres) since the corresponding degrees of freedom are relatively small. So, we focus on the solvers of (3.3) and (3.8) in the following.

For the sake of simplicity, the corresponding discrete algebraic system of variational problems (3.3) and (3.8) are called auxiliary systems 1 and 2, respectively.

3.2 Solver for auxiliary system 1

Although Eq. (3.4) is similar to original problem (2.6), it's luck to find a corresponding regularization diagonal block preconditioner $\hat{K}^{-1}$ given by

$$\hat{K} = \begin{pmatrix} \hat{A}_g & 0 \\ 0 & \hat{A}_g \end{pmatrix}, \tag{3.11}$$

where $\hat{A}_g$ is relative to the bilinear form

$$\hat{a}_g(\phi_r, \varphi_r) = \int_{\Omega_M} (\nabla \phi_r)^T D^r (\nabla \varphi_r) \mathrm{d}\boldsymbol{x}. \tag{3.12}$$

Noting that $\hat{K}$ is a diagonal matrix, then using of PMinres method with preconditioner $\hat{K}$ is essentially translated to solve the following system twice

$$\hat{A}_g w = g. \tag{3.13}$$

Furthermore, it is easy to see that matrix $\hat{A}_g$ is relative to $H(\mathbf{grad})$-elliptical problem by using (3.12). Hence we can also use Boomer AMG based on HYPRE and present the corresponding numerical results as follows:

Here we set $J = 2$, the controlling accuracy in PMinres method and Boomer AMG iterative method are both 10^{-6}, iter1 and iter2 denote iteration number of PMinres method and Boomer AMG iterative method, respectively.

Table 1 shows that iteration numbers of PMinres method with preconditioner $\hat{K}$ are independent of the mesh size, and average iteration number of Boomer AMG method between 4 and 5.1, so the former is stable and the latter is basically stable.

Table 1: Iteration numbers of PMinres method and Boomer AMG iterative method.

		$j=1$		$j=2$	
H	h	iter1	iter2	iter1	iter2
$\frac{8}{16}$	$\frac{8}{32}$	47	4.00	53	4.00
$\frac{8}{32}$	$\frac{8}{64}$	47	4.97	55	4.38
$\frac{8}{64}$	$\frac{8}{128}$	45	5.01	55	4.92

3.3 Solver for auxiliary system 2

By (3.10), we know that $\hat{M}$ is a diagonal matrix and SPD matrix $\hat{A}$ is corresponding to the following bilinear form

$$\hat{a}_A(\boldsymbol{u},\boldsymbol{\psi}) = \int_{\Omega_M} |S^r|(\nabla\times\boldsymbol{u})(\nabla\times\boldsymbol{\psi})\mathrm{d}\boldsymbol{x} + \int_{\Omega_M} \boldsymbol{u}^T D^r \boldsymbol{\psi}\mathrm{d}\boldsymbol{x}. \tag{3.14}$$

Remark 3.1. Making use of the definitions of $\sigma(t)$ and $d(t)$, it can prove that the diagonal elements of diagonal matrix D^r are positive if $\sigma_0 > 0$.

Remark 3.2. Bilinear form $\hat{a}_g(\cdot,\cdot)$ given by (3.12) can be viewed as a projection of $\hat{a}_A(\cdot,\cdot)$ restricted on $\mathbf{grad}S_{h,0}$, and it's also included in H-X precondition.

Now we can rewrite the Eq. (3.9) in the auxiliary system 2 as

$$\hat{M}\hat{X} = \hat{F},$$

where $\hat{M}$ given by (3.10), $\hat{F} = (\hat{F}_1,\hat{F}_2)^T$ and $\hat{X} = (\hat{X}_1,\hat{X}_2)^T$ can be reviewed as the approximation of solution X of system (2.6). In order to solve the above algebraic system, we need to solve the following system twice:

$$\hat{A}\hat{X}_i = \hat{F}_i, \qquad i = 1,2. \tag{3.15}$$

We use the PCG method with H-X precondition in [14] to solve (3.15), since that $\hat{A}$ is correspond to a $\boldsymbol{H}(curl)$-elliptic problem (see (3.14)). Now, we present the corresponding numerical results in Table 2. Here we set $J = 2$ and the controlling accuracy is 10^{-6}. Table 2 shows that iteration number are independent of the mesh size.

Table 2: Iteration numbers of PCG method with H-X precondition.

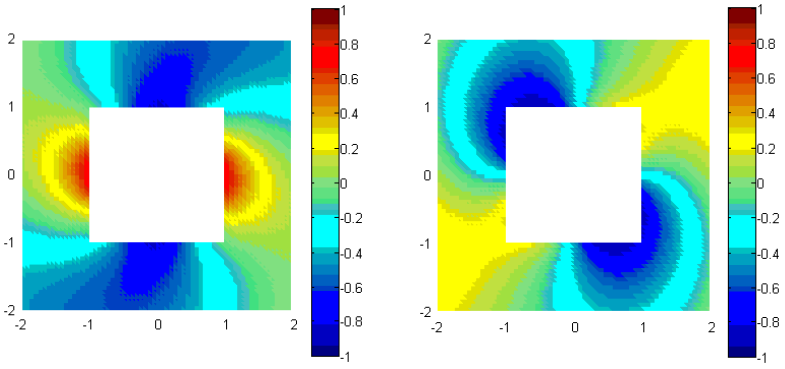
H	h	$j=1$	$j=2$
$\frac{8}{16}$	$\frac{8}{32}$	13	13
$\frac{8}{32}$	$\frac{8}{64}$	13	13
$\frac{8}{64}$	$\frac{8}{128}$	13	13

Summing up, in view of Tables 1 and 2, our solvers for auxiliary systems 1 and 2 are stable. Furthermore, the numerical experiments in the next section show that solving auxiliary systems 1 and 2 requires a very small proportion of the time in our iterative two-grid method (see Tables 10), hence we essentially translate the computation of original problem on a fine space into the one on a coarse space.

4 Numerical experiments

In this section, we report the results of numerical experiments which illustrate the efficiency of the iterative two-grid method in the previous section. In particular, we consider a model problem (2.2) under the following conditions:

$$\Omega_M = [-4,4]^2 \setminus [-1,1]^2, \quad a = 2, \quad b = 3 \quad \text{and} \quad \boldsymbol{g} = \nabla\times\left[H_1^{(1)}(r)e^{i\theta}\right]\big|_{\Gamma},$$



(a) (b)

Figure 2: The electric-field distributions about the first component (a) and the second component (b) of the real part of $\boldsymbol{u}$.

where $H_1^{(1)}(r)$ is the first kind Hankel function. Noting that the region of interest in model problem (2.2) is Ω_1 (see [5]), hence we will focus on the error in Ω_1. According to the boundary condition

$$\boldsymbol{u}|_\Gamma \times \boldsymbol{n} = \boldsymbol{g}|_\Gamma \times \boldsymbol{n},$$

we get an approximated function

$$\boldsymbol{u}|_{\Omega_1} = \nabla \times \left[H_1^{(1)}(r)e^{i\theta}\right],$$

of model problem (2.2). In Fig. 2 we show the electric-field distributions about the first and second component of the real part of $\boldsymbol{u}$ on Ω_1. In the following experiments, we will show the distributions of numerical solution and compare them.

All numerical experiments performance on a PC with Intel(R) P4 3.00 GHz processor, 1GB main memory and Linux operating system.

In the following, we present the numerical results for the PMinres methods and the iterative two-grid method, respectively.

4.1 PMinres method

Firstly, we present the numerical results by using PMinres method with identity precondition (namely Minres method) in Tables 3 and 4, where the controlling accuracy is 10^{-6}, iter and time denote the iteration number and CPU times, respectively.

The results in Table 3 show that the orders of the L^2-error and $\boldsymbol{H}(curl)$-error are $\mathcal{O}(h)$. From Table 4, we observe the following two properties:

- when the mesh size is reduced by half, the iteration number of Minres method increase four times.
- CPU times grow fastly if the discrete systems become large, which implies the corresponding condition numbers also become large.

Table 3: Error for Minres method.

h	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{0,\Omega_1}$	ratio	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{\boldsymbol{H}(curl;\Omega_1)}$	ratio
$\frac{8}{16}$	6.0118e-1		6.8581e-1	
$\frac{8}{32}$	3.1081e-1	1.93	3.4915e-1	1.96
$\frac{8}{64}$	1.5672e-1	1.98	1.7475e-1	2.00
$\frac{8}{128}$	7.8556e-2	2.00	8.7496e-2	2.00

Table 4: Iteration numbers and CPU times (in second) for Minres method.

h	iter	time
$\frac{8}{16}$	3845	2.52
$\frac{8}{32}$	21815	63.39
$\frac{8}{64}$	88610	1352.11
$\frac{8}{128}$	317226	22092.86

Table 5: Iteration numbers and CPU times (in second) for PMinres method.

h	iter	iter_p	time
$\frac{8}{16}$	121	11.98	2.76
$\frac{8}{32}$	169	12.93	16.02
$\frac{8}{64}$	342	13.22	148.85
$\frac{8}{128}$	677	14.23	2078.92

Secondly, we report the numerical results by using PMinres method with the following precondition

$$C = \begin{pmatrix} \hat{A}^{-1} & \\ & \hat{A}^{-1} \end{pmatrix}. \tag{4.1}$$

Here, we use PCG method with H-X precondition to solve the following problem in PMinres method

$$w = Cg, \tag{4.2}$$

where the controlling accuracy is 10^{-6}.

In this case, the corresponding errors are the same as those in Table 3, and iteration number and CPU times will be given in Table 5, where iter and time denote the iteration number and CPU times for PMinres method, and iter_p denotes the average iteration number of PCG method with H-X precondition for solving (4.2).

From Table 5, the mesh size is reduced by half, we observe that the iteration number of PMinres method increased doubly and CPU times increased 9.6 times ($h = 8/128$). Although PMinres method is better than Minres method, it's still difficult to be applied to the larger problem. Hence, it's necessary to design the more efficient fast algorithm for solving (2.6).

4.2 Iterative two-grid method

Firstly, we report the numerical result for errors between the solution $\boldsymbol{u}$ and iterative two-grid solution $\boldsymbol{u}_h^J$ and ratios for a set of combination of h and H with $J = 1, 2$ in Tables 6-7.

Table 6: Errors of Algorithm 3.1 with $J = 1$.

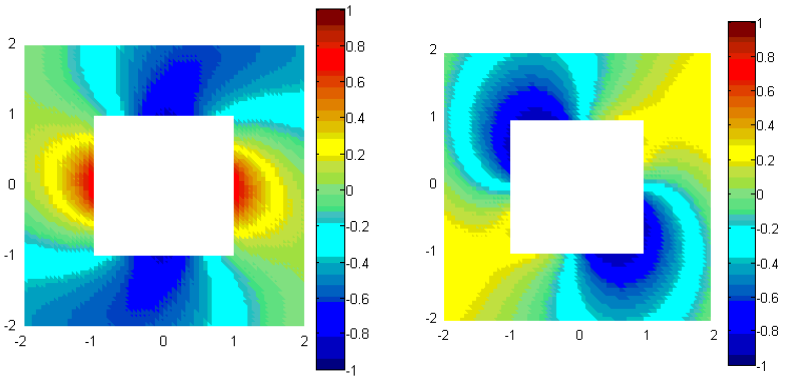
H	h	$\|\boldsymbol{u} - \boldsymbol{u}_h^J\|_{0,\Omega_1}$	ratio	$\|\boldsymbol{u} - \boldsymbol{u}_h^J\|_{\boldsymbol{H}(curl;\Omega_1)}$	ratio
$\frac{8}{16}$	$\frac{8}{32}$	8.4374e-1		8.8640e-1	
$\frac{8}{32}$	$\frac{8}{64}$	5.7837e-1	1.46	5.8867e-1	1.51
$\frac{8}{64}$	$\frac{8}{128}$	4.0461e-1	1.43	4.0755e-1	1.44

Table 7: Errors of Algorithm 3.1 with $J = 2$.

H	h	$\|\boldsymbol{u} - \boldsymbol{u}_h^J\|_{0,\Omega_1}$	ratio	$\|\boldsymbol{u} - \boldsymbol{u}_h^J\|_{\boldsymbol{H}(curl;\Omega_1)}$	ratio
$\frac{8}{16}$	$\frac{8}{32}$	3.1730e-1		3.6128e-1	
$\frac{8}{32}$	$\frac{8}{64}$	1.6207e-1	1.96	1.8187e-1	1.99
$\frac{8}{64}$	$\frac{8}{128}$	7.9254e-2	2.04	8.8707e-2	2.05

Tables 6-7 show the following conclusions. The solution of iterative two-grid method with $J = 1$ does not approximate the real solution well. For $J = 2$, the L^2-error and $\boldsymbol{H}(curl)$-error closely approximate the ones for PMinres method by comparing Table 3 with Table 7, their orders are $\mathcal{O}(h)$. Hence, we only present the numerical results with $J = 2$ in the following.

Since PMinres method is shown to be a better solver of the variational problem (3.6) on a coarse mesh than the standard Minres, we only make use of PMinres method to be a solver of a coarse mesh problem in the following, and $\hat{C}^{-1}$ given by (4.1) to be the corresponding precondition. Fig. 3 shows that the electric-field distributions about the first and second component of the real part of $\boldsymbol{u}_h^J$ on the interest region for our



(a) (b)

Figure 3: The electric-field distributions about the first component (a) and the second component (b) of the real part of $\boldsymbol{u}_h^J$.

Table 8: Error of PMinres method on coarse mesh for Algorithm 3.1.

H	h	$\|\boldsymbol{u}-\boldsymbol{u}_h^J\|_{0,\Omega_1}$	ratio	$\|\boldsymbol{u}-\boldsymbol{u}_h^J\|_{\boldsymbol{H}(curl;\Omega_1)}$	ratio
$\frac{8}{16}$	$\frac{8}{32}$	3.1730e-1		3.6128e-1	
$\frac{8}{32}$	$\frac{8}{64}$	1.6207e-1	1.96	1.8187e-1	1.99
$\frac{8}{64}$	$\frac{8}{128}$	7.9254e-2	2.04	8.8707e-2	2.05
$\frac{8}{128}$	$\frac{8}{256}$	4.0007e-2	1.98	4.4792e-2	1.98

Table 9: Iteration numbers of PMinres method on coarse mesh for Algorithm 3.1.

H	h	iter_c1	iter_p1	iter_c2	iter_p2
$\frac{8}{16}$	$\frac{8}{32}$	121	12.00	127	12.00
$\frac{8}{32}$	$\frac{8}{64}$	165	12.94	175	12.97
$\frac{8}{64}$	$\frac{8}{128}$	340	13.22	301	13.23
$\frac{8}{128}$	$\frac{8}{256}$	713	14.22	545	14.23

Table 10: CPU-times spent at each part of Algorithm 3.1 with PMinres method on coarse mesh.

			time_k		time_c		time_f		
H	h	time_m	t_k1	t_k2	t_c1	t_c2	t_f1	t_f2	total time
$\frac{8}{16}$	$\frac{8}{32}$	0.53	0.63	0.48	2.72	2.72	0.13	0.13	6.93
$\frac{8}{32}$	$\frac{8}{64}$	2.06	2.34	2.42	15.30	17.38	0.45	0.46	38.96
$\frac{8}{64}$	$\frac{8}{128}$	8.56	18.75	22.20	156.92	138.96	2.83	2.85	344.39
$\frac{8}{128}$	$\frac{8}{256}$	96.26	103.06	113.27	2198.13	1679.25	15.65	15.77	4134.15

iterative two-grid method. Compared with Fig. 2, we observe that their distributions are consistent, which implies that our algorithm is reliable.

The corresponding errors in Table 8 show that the orders of L^2-error and $\boldsymbol{H}(curl)$-error are still $\mathcal{O}(h)$. Tables 9 and 10 present the corresponding iteration numbers and CPU-times spent at each part of Algorithm 3.1, where iter_c1 and iter_c2 denote iteration numbers of PMinres method on coarse mesh in Algorithm 3.1 for the first time and second time, and iter_p1 and iter_p2 denote average iteration numbers of PCG method with H-X precondition in Algorithm 3.1 for the first time and second time.

Compared Table 10 with Table 5, we also observe that total CPU times for iterative two-grid method are less than those for PMinres method. For example, when $h = 8/128$, the total time of the former reduce to about one-sixth of the latter's).

In the previous numerical results, we can find that the proportion of CPU times for solving (3.6) on coarse mesh occupies a large share of total time (more than 85%) since the size of coarse mesh is only twice of the size of fine mesh. So, we present some numerical results in Tables 11-13 by enlarging the size between coarse mesh and fine mesh in the following.

Table 11 shows that the orders of L^2-error and $\boldsymbol{H}(curl)$-error are still $\mathcal{O}(h)$. For a fixed h, compared Table 11 with Table 8, we observe that the orders of L^2-error and $\boldsymbol{H}(curl)$-error in Table 11 are between h and $2h$ which both in Table 8, and more closer to h. For example, when $h = 8/128$, L^2-error $\|\boldsymbol{u}-\boldsymbol{u}_h^J\|_{0,\Omega_1}$=1.0330e-1 in Table 11 is between 1.6207e-1 and 7.9254e-2 in Table 8, and more closer to 7.9254e-2.

Table 11: Error of Algorithm 3.1.

H	h	$\|\boldsymbol{u}-\boldsymbol{u}_h^J\|_{0,\Omega_1}$	ratio	$\|\boldsymbol{u}-\boldsymbol{u}_h^J\|_{\boldsymbol{H}(curl;\Omega_1)}$	ratio
$\frac{8}{16}$	$\frac{8}{64}$	1.9363e-1		2.2913e-1	
$\frac{8}{32}$	$\frac{8}{128}$	1.0330e-1	1.87	1.2010e-1	1.91
$\frac{8}{64}$	$\frac{8}{256}$	4.7231e-2	2.19	5.6818e-2	2.11

Table 12: Iteration numbers of PMinres method on coarse mesh for Algorithm 3.1.

H	h	iter_c1	iter_p1	iter_c2	iter_p2
$\frac{8}{16}$	$\frac{8}{64}$	121	11.99	127	12.00
$\frac{8}{32}$	$\frac{8}{128}$	164	12.91	171	12.95
$\frac{8}{64}$	$\frac{8}{256}$	339	13.23	293	13.19

Table 13: CPU-times spent at each part of Algorithm 3.1.

			time_k		time_c		time_f		
H	h	time_m	t_k1	t_k2	t_c1	t_c2	t_f1	t_f2	total time
$\frac{8}{16}$	$\frac{8}{64}$	1.93	2.40	2.40	2.58	2.70	0.45	0.45	11.53
$\frac{8}{32}$	$\frac{8}{128}$	7.06	18.57	22.18	15.90	16.63	2.74	2.94	80.74
$\frac{8}{64}$	$\frac{8}{256}$	42.99	103.59	116.43	151.47	130.97	15.79	16.09	546.88

From Tables 12, we know that the iteration numbers of solving (3.7) reduce, and from Tables 13 that both CPU times for solving (3.7) and total time decrease significantly. Hence, in order to save the total computational time, it is reasonable to enlarge the size of coarse mesh properly in Algorithm 3.1 .

Remark 4.1. It should be remark that the size of coarse mesh in Algorithm 3.1 cannot be enlarged too large, otherwise the approximation order would be influenced. Heuristic arguments are referred to the literature [2].

Acknowledgments

This work was partially supported by NSFC Project (Grant No. 11031006,91130002, 11171281,10971059,11026091), the Key Project of Scientific Research Fund of Hunan Provincial Science and Technology Department (Grant No. 2011FJ2011), and Hunan Provincial Innovation Foundation for Postgraduate (CX2010B245, CX2010B246). The authors would like to thank Prof. Jichun Li, University of Nevada Las Vegas, USA, for providing many constructive suggestions and also Dr. Chunsheng Feng and Xiaoqiang Yue, Xiangtan University, for their help with the numerical experiments.

References

[1] E. BÉCACHE AND P. JOLY, *On the analysis of Bérenger's perfectly matched layers for Maxwell's equations*, M2AN Math. Model. Numer. Anal., 36 (2002), pp. 87–119.

[2] R. BECK, P. DEUFLHARD, R. HIPTMAIR, R. H. W. HOPPE AND B. WOHLMUTH, *Adaptive multilevel methods for edge element discretizations of Maxwell's equations*, Surveys. Math. Indust., 8 (1999), pp. 271–312.
[3] J. P. BÉRENGER, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp.185–200.
[4] J. P. BÉRENGER, *Perfectly matched layer (PML) for computational electromagnetics*, Volume 2 of Synthesis Lectures on Computational Electromagnetics, Morgan & Claypool, 2007.
[5] J. H. BRAMBLE AND J. E. PASCIAK, *Analysis of a cartesian PML approximation to the three dimensional electromagnetiac wave scattering problem*, Preprint, 2010.
[6] Y. BOTROS AND J. VOLAKIS, *Preconditioned generalized minimal residual iterative scheme for perfectly matched layer terminated applications*, IEEE Microw. Guided. W., 9 (1999), pp. 45–47.
[7] Y. BOTROS AND J. VOLAKIS, *Perfectly matched layer termination for finite-element meshed: implementation and application*, Microw. Opt. Tech. Lett., 23 (1999), pp. 166–172.
[8] H. BANKS AND B. BROWNING, *Time domain electromagnetic scattering using finite elements and perfectly matched layers*, Comput. Methods. Appl. Mech. Eng., 194 (2005), pp. 149–168.
[9] G. BAO AND H. WU, *Convergence analysis of the perfectly matched layer problems for time harmonic Maxwell's euqations*, SIAM J. Numer. Anal., 43 (2005), pp. 2121–2143.
[10] W. CHEW AND W. WEEDON, *A 3d perfectly matched medium for modified Maxwell's equations with streched coordinates*, Microw. Opt. Tech. Lett., 13 (1994), pp. 599–604 .
[11] P. MONK, Finite Element Methods for Maxwell's Equations, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, UK, 2003.
[12] R. HIPTMAIR, *Multigrid method For Maxwell's equations*, SIAM J. Numer. Anal., 36 (1998), pp. 204–225.
[13] J. JIN, S. SHU AND J. XU, *A two-grid discretization method for decoupling systems of partial differntial equations*, Math. Comput., 75 (2006), pp. 1617–1626.
[14] R. HIPTMAIR AND J. XU, *Nodal auxiliary spaces preconditions in $H(curl)$ and $H(div)$ spaces*, SIAM J. Numer. Anal., 45 (2007), pp. 2483–2509.
[15] Y. HUANG AND Y. CHEN, *A multilevel iterative method for finite element solutions to singular two-point boundary value problems*, Natur. Sci. J. Xiangtan Univ., 16 (1994), pp. 23–26.
[16] M. MU AND J. XU, *A two-grid method of a mixed Stokes-Darcy model for coupling fluid flow with porous media flow*, SIAM J. Numer. Anal., 45 (2007), pp. 1801–1813.
[17] F. L. TEIXEIRA AND W. C. CHEW, *Differential forms, metrics, and the reflectionless absorption of electromagnetic waves*, J. Electromagn. Waves. Appl., 13 (1999), pp. 665–686.
[18] J. XU, *Iterative methods by SPD and small subspace solvers for nonsymmetric or indefinite problems*, in Proceedings of the 5th International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. Chan, D. Keyes, G. Meurant, J. Scroggs and R. Voigt (eds), Siam, Philadelphia, 1992, pp. 106–118.
[19] J. XU, *Two-grid discretization techniques for linear and nonlinear PDEs*, SIAM J. Numer. Anal., 33 (1996), pp. 1759–1777.
[20] J. XU AND X. CAI, *A preconditioned gmres method for nonsymmetric or indefinite problems*, Math. Comput., 59 (1992), pp. 311–319.
[21] J. XU AND A. ZHOU, *Local and parallel finite element algorithms based on two-grid discretizations for nonlinear problems*, Adv. Comput. Math., 14 (2001), pp. 293–327.
[22] J. XU AND A. ZHOU, *A two-grid discretization scheme for eigenvalue problems*, Math. Comput., 70 (2001), pp. 17–25.