A MULTIGRID METHOD FOR VARIABLE COEFFICIENT MAXWELL'S EQUATIONS

J. JONES * AND B. LEE †

Abstract. This paper presents a multigrid method for solving variable coefficient Maxwell's equations. The novelty in this method is the use of interpolation operators that do not produce multilevel commutativity complexes that lead to multilevel exactness. Rather, the effects of multilevel exactness are built into the level equations themselves- on the finest level using a discrete $T-V$ formulation, and on the coarser grids through the Galerkin coarsening procedure of a $T-V$ formulation. These built-in structures permit the levelwise use of an effective hybrid smoother on the curl-free near-nullspace components, and permit the development of interpolation operators for handling the curl-free and divergence-free error components separately. The resulting block diagonal interpolation operator does not satisfy multilevel commutativity but has good approximation properties for both of these error components. Applying operator-dependent interpolation for each of these error components leads to an effective multigrid scheme for variable coefficient Maxwell's equations, where multilevel commutativity-based methods can degrade. Numerical results are presented to verify the effectiveness of this new scheme.

Key words. Maxwell's equations, multigrid method, edge finite elements.

AMS(MOS) subject classifications. 65N55, 65N30

1. Introduction. In recent years, there has been substantial interest in solving the curl-curl formulation of Maxwell's equations for the electric or magnetic field. This formulation commonly arises in the time-domain and frequency-domain approaches for solving time-dependent electromagnetic problems. For both approaches, the challenges in developing an efficient multigrid algorithm are formidable: the time-domain leads to variable-coefficient equations with large near-nullspaces, as with the frequency-domain but with the added challenge of being indefinite. In this paper, we consider only the definite curl-curl formulation.

There exist several successful geometric ([16], [2]) and algebraic ([20], [5]) multigrid methods for solving the definite curl-curl equations. But the performance of these methods degrade as the variation of the material coefficients increases. The reason for this degradation can be traced back to the interpolation operator. For geometric schemes, even with nested finite element spaces, the natural finite element interpolation fails when the coefficients strongly vary much the same way natural finite element interpolation fails for scalar diffusion equations with rapidly varying coefficients, i.e. finite element interpolation leads to simple arithmetic averaging of the coefficients on the coarser grids. For algebraic schemes, since the interpolation operator is constructed with the constraint that a multilevel commutativity complex is formed (see 3.4), divergence-free error components can be untouched or even amplified by this type of interpolation. In particular, interpolation operators constructed in this manner can handle at most only the curl-free error components, and thus, only half of the Helmholtz decomposition of the error. Thus, current geometric and algebraic multigrid methods do not satisfy the approximation property when the material coefficients vary substantially.

Satisfying the approximation property is not the only difficulty. For Maxwell's equations, the smoothing property is hard to obtain. For this system pde, near-nullspace components occur on all grid scales, and thus, relaxation must be designed to handle error components of these form on all grid scales. This is in fact the purpose of having a multilevel commutativity complex. Such complex permits easy movement between two-term exact sequences on different grid levels. With a discretization that

[*] Department of Mathematical Sciences, Florida Institute of Technology, 150 West University Boulevard, Melbourne, Florida 32901-6975. *email*: jim@fit.edu

[†] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA. *email*: lee123@llnl.gov. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48.

preserves this two-term exact sequence, nullspace components of the curl operator are explicitly known, and hence, hybrid smoothers ([16], [2]) can be constructed to effectively handle curl-free error components. Having this two-term sequence on each level then guarantees level smoothers that eliminate curl-free error of any grid scale, and having a multilevel commutativity complex guarantees curl-free coarse grid corrections are indeed curl-free on the finer grid. But ideally, as previously explained, one would like a multilevel commutativity complex that spans over an exact sequence that also includes the divergence operator. This would involve developing separate interpolation operators for the curl-free and divergence-free components, which implicitly implies the existence of an exact discrete Helmholtz decomposition. Or, if an exact discrete Helmholtz decomposition does not exist, one needs an interpolation operator that acts appropriately (i.e., has good approximation) on both the curl-free and divergence-free components. Such an interpolation operator is unfortunately extremely difficult to construct.

The challenge in developing a multigrid scheme for variable coefficient curl-curl equations is then constructing interpolation operators that have good approximation property for both curl-free and divergence-free errors, and, such that the coarse grid problems constructed using these operators have the relevant effects of two-term exact sequences in order for a hybrid smoother or overlapping Schwarz smoother to be effective on coarser levels. Rather than constructing interpolation operators that satisfy these constraints, in this paper, we develop a multigrid scheme that relaxes on the multilevel commutativity constraint, and hence, has more freedom in the construction of interpolation operators to handle the curl-free and divergence-free error components. This method involves a discrete $T - V$ formulation ([19]) on the finest level. The construction of this $T - V$ formulation can be obtained from the given curl-curl formulation matrix and a discrete gradient on the finest level.

The goal of this formulation is to introduce the nullspace of the curl operator, i.e. gradients, explicitly into the set of equations. This leads to a discrete system for the curl-free and approximate divergence-free components of the solution, which corresponds to a system pde for these solution components. Separate operator-based interpolation operators are then constructed using the original curl-curl equations and the Laplace operator derived from the introduction of the gradients. These block diagonal interpolation operators are used in the Galerkin coarsening procedure to generate coarse grid problems for the curl-free and divergence-free components of the error. Thus, at all levels, the systems are blocked 2×2, with the diagonal blocks describing the coupling within the curl-free or divergence-free components, and with the off-diagonal blocks describing their coupling. Now equations for the curl-free, or near-nullspace gradients, are explicitly available at all levels, and so a hybrid smoother can be applied at each level.

It must be noted that the $T-V$ formulation is introduced discretely only to obtain an efficient multigrid solver. The target continuous curl-curl formulation of Maxwell's equations is not explicitly replaced by a continuous semi-definite $T - V$ formulation. Also, it must be noted that the above scheme resembles Griebel's subspace correction method [14]. However, unlike Griebel's scheme which creates linearly dependent generating functions by collecting the finite element basis functions on all levels, the above scheme considers the "generating functions" from a Helmholtz decomposition perspective, at each level. Indeed, the Helmholtz decomposition perspective gives guidance on developing effective interpolation operators and relaxation schemes, and is much more insightful for $H(\text{div})$ and $H(\text{curl})$ system pdes than a general subspace correction perspective. In fact, the $T - V$ formulation exposes the extent to which the components of the Helmholtz decomposition can be coarsened separately.

This paper is organized as follow. In section 2, we introduce the curl-curl formulation of the definite Maxwell's equations, the functional setting for the variational problem, and the finite element spaces for the discretization. In section 3, because of their importance in understanding solution methods for the curl-curl formulation, we review the de Rham complex for the curl-curl formulation and the multilevel commutativity diagrams connecting the complexes on different levels. In particular, because the multigrid scheme developed in this paper can be viewed as a generalization of Hiptmair's approach ([16]) that does not satisfy multilevel commutativity, we need to examine what multilevel commutativity achieves and how much of it is needed in a multigrid solver. In fact, one purpose of this paper is to "isolate" de Rham sequences and commutativity from multigrid. To achieve this, in section 3, we will distinguish between multilevel commutativity and multilevel de Rham complexes, and show that multigrid schemes with commutativity-based interpolation operators generally only enforce exactness whereas multilevel de Rham complexes are really needed for multigrid effectiveness. In section 4, we describe our $T-V$ formulation and the operator-based interpolation operators to handle variable coefficients. An intuitive justification for our choice of operator-based interpolation operators is presented. In particular, since the $T-V$ system is used in the multigrid procedure, we examine the near-nullspace components of this system to show that operator collapsing ([10]-[12]) can be used to handle the curl-free error components and AMGe element agglomeration ([9]) can be used to handle the remaining error components. Finally, in section 5, we present some preliminary numerical results illustrating the effectiveness of this multigrid scheme, and equally important, illustrating no need for multilevel commutativity complexes.

2. Curl-Curl Formulation in the Electric Field.

Let $\Omega\times T$ be the Cartesian product of a bounded simply-connected domain $\Omega \in \Re^n, n=2,3$, and a non-negative time interval T. To guarantee appropriate regularity of the problem, let Ω have a smooth or polygonal boundary Γ. Electromagnetic phenomenom in $\Omega\times T$ can be described by the differential equations

$$\begin{aligned} \nabla\times\mathbf{E} &= -\tfrac{\partial \mathbf{B}}{\partial t} && \text{in } \Omega\times T \\ \nabla\times\mathbf{H} &= \tfrac{\partial \mathbf{D}}{\partial t}+\mathbf{j} && \text{in } \Omega\times T \end{aligned} \tag{2.1}$$

with the linear material constituitive relations

$$\begin{aligned} \mathbf{D} &= \epsilon\mathbf{E} && \text{in } \Omega\times T \\ \mathbf{B} &= \mu\mathbf{H} && \text{in } \Omega\times T \\ \mathbf{j} &= \sigma\mathbf{E} && \text{in } \Omega\times T \end{aligned} \tag{2.2}$$

([19]). The equations in (2.1) are Faraday and Ampere's laws of Maxwell's equations, and the third equation in (2.2) is a simple Ohm's law. Here, $\mathbf{E},\mathbf{D},\mathbf{H},\mathbf{B}$, and $\mathbf{j}$ are respectively the electric field, electric flux, magnetic field, magnetic flux, and electric current; and ϵ,μ, and σ are respectively the electric permittivity, magnetic permeability, and electric conductivity, which we assume all to be spatially dependent but independent of t. Using (2.1) and (2.2), we have

$$\nabla\times\frac{1}{\mu}\nabla\times\mathbf{E}+\epsilon\frac{\partial^2\mathbf{E}}{\partial^2 t}+\sigma\frac{\partial\mathbf{E}}{\partial t}=\mathbf{0}.$$

Differencing the time derivatives, and for simplicity, assuming that the boundary surface is perfectly conducting, we obtain a boundary-value problem of the form

$$\begin{aligned} \nabla\times\alpha\nabla\times\mathbf{E}+\beta\mathbf{E} &= \mathbf{f} && \text{in } \Omega, \\ \mathbf{n}\times\mathbf{E} &= \mathbf{0} && \text{on } \Gamma, \end{aligned} \tag{2.3}$$

where α and β are positive functions (in $L_\infty(\Omega)$). The parameter β) is related to the time step size ($\beta=\frac{\alpha}{\Delta t}$ and may be quite small if large time steps are taken. We will call (2.3) the *curl-curl formulation* for the electric field. At each time step of the solution process for the time-dependent problem, an equation of the form (2.3) must be solved.

To describe the variational formulation of (2.3), several functional spaces are needed. So, we will denote the usual $k'th$ order Sobolev spaces by $H^k(\Omega)$ and their homogeneous trace subspaces by $H_0^k(\Omega)$. We also need spaces

$$\begin{aligned} H(\mathrm{div};\Omega) &= \{\mathbf{v} \in [L^2(\Omega)]^n : \nabla\cdot\mathbf{v} \in L^2(\Omega)\} \\ H(\mathrm{curl};\Omega) &= \{\mathbf{v} \in [L^2(\Omega)]^n : \nabla\times\mathbf{v} \in [L^2(\Omega)]^{2n-3}\}. \end{aligned}$$

We denote their homogeneous trace subspaces by $H_0(\mathrm{div};\Omega)$ and $H_0(\mathrm{curl};\Omega)$, respectively. Now, the variational formulation of (2.3) is to find $\mathbf{E} \in H_0(\mathrm{curl};\Omega)$ such that

$$(2.4) \qquad (\alpha\nabla\times\mathbf{E}, \nabla\times\mathbf{v}) + (\beta\mathbf{E}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \qquad \forall\mathbf{v} \in H_0(\mathrm{curl};\Omega).$$

To discretize variational problem (2.4), the lowest-order Nedelec finite elements are used. Let $\mathcal{T}_h := \{T_i\}$ be a quasi-uniform and shape-regular triangulation of Ω with mesh-size h ([8]). The lowest-order local Nedelec finite element space is

$$\mathcal{ND}(T_i) := \{\mathbf{v} = \mathbf{a} + \mathbf{b}\times\mathbf{x};\ \mathbf{x} \in T_i,\ \mathbf{a},\mathbf{b} \in \Re^n\}.$$

The degrees of freedom are the tangential components along the edges of T_i- along the $l'th$ edge e_l,

$$\int_{e_l} \mathbf{t}\cdot\mathbf{v}\, ds.$$

The global finite element space is

$$\mathcal{ND}(\mathcal{T}_h) := \left\{\mathbf{v}_h \in H_0(\mathrm{curl};\Omega) : \mathbf{v}_{h|T_i} \in \mathcal{ND}(T_i)\ \ \forall T_i \in \mathcal{T}_h\right\}.$$

This choice of finite element space guarantees continuity of the tangential component of $\mathbf{v}_h$. With $\mathbf{E}_h$ denoting the discrete approximation to the solution of (2.4), the discrete variational problem is to find $\mathbf{E}_h \in \mathcal{ND}(\mathcal{T}_h)$ such that

$$(2.5) \qquad (\alpha\nabla\times\mathbf{E}_h, \nabla\times\mathbf{v}_h) + (\beta\mathbf{E}_h, \mathbf{v}_h) = (\mathbf{f}, \mathbf{v}_h) \qquad \forall\mathbf{v}_h \in \mathcal{ND}(\mathcal{T}_h).$$

We also will need the standard first-order scalar Lagrange finite element space:

$$\mathcal{H}_h(\mathcal{T}_h) := \left\{v \in H_0^1(\Omega) : v_{|T_i} \in \mathcal{P}_1(T_i)\ \ \forall T_i \in \mathcal{T}_h\right\}$$

with the usual nodal degrees of freedom. Note that the gradient of an element of $\mathcal{H}_h(\mathcal{T}_h)$ is not only in $\mathcal{ND}(\mathcal{T}_h)$ but also in the nullspace of the curl operator. In fact, the assumption of simple-connectedness of Ω guarantees that the nullspace of the curl operator consists of only gradients. By cohomology theory, this is also true in the triangulated domain ([7]). (The algorithm of this paper as with other multigrid algorithms for solving (2.5) requires simple-connectedness of Ω. For domains such as tori, an additional procedure is needed to handle special troublesome subspaces of low dimension. These subspaces can be treated on a coarse grid ([17]).)

3. De Rham and Commutativity Complexes, and Multigrid. The multigrid scheme we will develop is a generalization of the algorithm described in [16]. But a major difference between these two schemes is, the method described in this paper does not produce multilevel commutativity complexes even when the levels are nested, whereas the method described in [16] does. Because of this, we explore de Rham and commutativity complexes and multigrid in this section. We examine if multigrid efficiency is really a result of these multilevel complexes. In particular, our goal in this section is to isolate commutativity from the multigrid process, and to show that effective multilevel complexes are too difficult to produce for variable coefficient Maxwell's equations. This will then lead to a simpler scheme, described in Section 4.

Before reviewing de Rham and commutativity complexes, we outline the algorithm of [16] in order to have a base method to illustrate multilevel complexes. This algorithm has the standard multigrid structure:

```
Given, an initial guess $\mathbf{E}^L$ and a right hand side $\mathbf{f}^L$ on the finest level $L$
MG($l$, $\mathbf{E}^l$, $\mathbf{f}^l$) Cycle:
  1. if $l = 0$, the coarsest level, solve exactly, $\mathbf{E}^0 \leftarrow [A^0_{ee}]^{-1}\mathbf{f}^0$.
  2. else,
     a. pre-smooth $\mathbf{E}^l \leftarrow S_l(\mathbf{E}^l, \mathbf{f}^l)$
     b. $\mathbf{E}^{l-1} \leftarrow \mathbf{0}$
     c. MG($l-1$, $\mathbf{E}^{l-1}$, $[P^l_{l-1}]^t(\mathbf{f}^l - A^l_{ee}\mathbf{E}^l)$ )
     d. coarse grid correction $\mathbf{E}^l \leftarrow \mathbf{E}^l + P^l_{l-1}\mathbf{E}^{l-1}$
     e. post-smooth $\mathbf{E}^l \leftarrow S_l(\mathbf{E}^l, \mathbf{f}^l)$,
```

where A^l_{ee} is the $l'th$ level discretized edge operator, and S_l and P^l_{l-1} are respectively the $l'th$ level smoothing and interpolation operators. However, the smoother is nonstandard:

```
$S_l(\mathbf{E}^l, \mathbf{f}^l)$ Sweep:
  1. Gauss-Seidel sweep on edge equations $A^l_{ee}\mathbf{E}^l = \mathbf{f}^l$
  2. edge residual computation, $\mathbf{r}^l \leftarrow \mathbf{f}^l - A^l_{ee}\mathbf{E}^l$
  3. transfer to scalar "potential" space (or set up of distributive relaxation),
     $\mathbf{w}^l \leftarrow [G^l_{en}]^t\mathbf{r}^l$
  4. Gauss-Seidel sweep on scalar potential equation $A^l_{nn}\phi^l = \mathbf{w}^l$
  5. potential solution correction $\mathbf{E}^l$, $\mathbf{E}^l \leftarrow \mathbf{E}^l + G^l_{en}\phi^l$,
```

where G^l_{en} is the level l discrete gradient operator mapping a node-based scalar potential into the Nedelec element space, and A^l_{nn} is a discrete node-based diffusion operator. When the level edges are nested and the $P^l_{l-1}{}'s$ are the natural Nedelec finite element interpolation operators, then the multilevel de Rham and commutativity complexes are hidden in the interpolation and discrete gradient operators. When the level edges are non-nested, as generally is the case for unstructured grids, the $P^l_{l-1}{}'s$ can be constructed so that a multilevel commutativity complex is formed (see [20] and [5]).

So, what are de Rham complexes? De Rham complexes describe a relationship between the gradient, curl, and divergence operators over appropriate spaces. They are an amazing tool for systematically describing stable discretizations of pde systems ([1]). Given a scalar or system of partial differential equations, subtle geometric properties of the pde's can be exposed by de Rham sequences. These properties should be retained in the discretization for the sake of numerical stability. For the Maxwell's equations, which have intrinsic topological properties reflective of de Rham complexes ([3]), these complexes have been used to explain the stablility of Nedelec finite elements for the curl-curl formulation ([6], [1]).

In the continuum, the de Rham complex of interest for Maxwell's equations is

$$H^1(\Omega) \xrightarrow{\nabla} H(\text{curl};\Omega) \xrightarrow{\nabla\times} H(\text{div};\Omega). \tag{3.1}$$

Here, we have a sequence of vector spaces defined on our domain Ω and a sequence of operators interrelating them. These vector spaces are *exact* in the sense that the range of an operator defined on the space to its left is in the kernel of the operator and domain space to its right- e.g., the kernel of $\nabla\times$ defined on $H(\text{curl};\Omega)$ is $\nabla H^1(\Omega)$. Let $\{H^1_h,\ H_h(\text{curl}),\ H_h(\text{div})\}$ and $\{S_{2h},\ W_{2h},\ V_{2h}\}$ be two sets of finite elements that need not be nested and of which the first set forms a discrete analogue to (3.1):

$$H^1_h(\Omega) \xrightarrow{\nabla^h} H_h(\text{curl};\Omega) \xrightarrow{\nabla\times^h} H_h(\text{div};\Omega). \tag{3.2}$$

Of course, exactness means that the finite element spaces in the first set cannot be chosen separately. In our case, choosing the lowest-order Nedelec elements ($\mathcal{ND}_h$) for $H_h(\text{curl};\Omega)$, the other finite element spaces are the first-order Lagrangian elements ($\mathcal{H}_h$) and lowest-order Raviart-Thomas elements ($\mathcal{RT}_h$) for $H^1_h(\Omega)$ and $H_h(\text{div};\Omega)$. This consistent choice of finite element spaces is in fact a geometric property (e.g., for the above multigrid algorithm, $\mathbf{E}^l$ is defined on edges ($\mathcal{ND}_h$) and ϕ^l is defined on nodes ($\mathcal{H}_h$)). This shows that a discrete de Rham sequence is not simply a discretization of the functional spaces involved, but is a "discretization" of a calculus defined over a smooth manifold and its sub-manifolds ([4], [13], and [15]). In particular, a discrete de Rham sequence satisfies three properties. It is (i) a discretization of a smooth n-dimensional manifold $\mathcal{M}$ with so-called p-cubes or Euclidean simplices; (ii) a discretization of the *p-forms* defined on the manifold and its sub-manifolds; and (iii) a discretization of a so-called exterior differential operator so that a generalized Stokes theorem holds over the manifold and its sub-manifolds (this gives the exactness property). These discretizations are constructed so that the p-forms are simple, unisolvent, conforming, etc. ([15]). In our case, $\mathcal{M}$ is a domain in $\Re^3$, the p-cubes are points, edges, and faces, and the p-forms are the above finite element spaces. The location of the degrees of freedom for these finite element spaces indeed reflects a *consistent* choice of simplices to guarantee exactness.

Now, denoting the induced finite element interpolation operators of $\mathcal{H}_h$, $\mathcal{ND}_h$, and $\mathcal{RT}_h$ by $\Pi^h_{\mathcal{H}}$, $\Pi^h_{\mathcal{ND}}$, and $\Pi^h_{\mathcal{RT}}$, the following commutativity complex holds:

$$\begin{tikzcd}
H^1(\Omega) \arrow[r, "\nabla"] \arrow[d, "\Pi^h_{\mathcal{H}}"] & H(\text{curl};\Omega) \arrow[r, "\nabla\times"] \arrow[d, "\Pi^h_{\mathcal{ND}}"] & H(\text{div};\Omega) \arrow[d, "\Pi^h_{\mathcal{RT}}"] \\
\mathcal{H}_h \arrow[r, "\nabla^h"] & \mathcal{ND}_h \arrow[r, "\nabla\times^h"] & \mathcal{RT}_h .
\end{tikzcd} \tag{3.3}$$

This complex relates a set of continuous spaces to a set of discretizations, under a collection of commuting differential and interpolation operators. For the above multigrid algorithm and several others ([20] and [5]), additional hierarchies of commutativity relating the level spaces are needed. These methods require a two-level commutativity complex of the form

$$\begin{tikzcd}
H^1(\Omega) \arrow[r, "\nabla"] \arrow[d, "\Pi^h_{\mathcal{H}}"] & H(\text{curl};\Omega) \arrow[r, "\nabla\times"] \arrow[d, "\Pi^h_{\mathcal{ND}}"] & H(\text{div};\Omega) \arrow[d, "\Pi^h_{\mathcal{RT}}"] \\
\mathcal{H}_h \arrow[r, "\nabla^h"] & \mathcal{ND}_h \arrow[r, "\nabla\times^h"] & \mathcal{RT}_h \\
S_{2h} \arrow[r, "\nabla^{2h}"] \arrow[u, "P^h_{H^1}"] & W_{2h} \arrow[r, "\nabla\times^{2h}"] \arrow[u, "P^h_{H(\text{curl})}"] & V_{2h}(\text{div}), \arrow[u, "P^h_{H(\text{div})}"]
\end{tikzcd} \tag{3.4}$$

where $P^h_{(\cdot)}$ are multilevel interpolation operators that are constructed to satisfy commutativity. Note that the second layer of (3.4) is a discrete de Rham sequence since it satisfies the above three properties. However, depending on the choice of $\{S_{2h},\ W_{2h},\ V_{2h}\}$, the third layer *only* forms an exact sequence between these sets since it may not satisfy condition (ii). Although multilevel exactness requires a consistent choice of coarse simplices, a multilevel de Rham structure also requires $\{S_{2h},\ W_{2h},\ V_{2h}\}$ to consists of proper $p-$forms. To stress this, consider the case when the coarse edges do not nest the fine edges, as is the case for the algorithms of [20] and [5]. Because of this non-nesting, $\{S_{2h},\ W_{2h}\}$ consists of *weighted* averages generated by the $P^h_{(\cdot)}$, which are not the proper $p-$forms for the coarse simplices.

Let's return to the above multigrid algorithm and reflect on these complexes. For the curl-curl formulation of Maxwell's equation, some of the near-nullspace components are gradients, and these occur on all grid scales. Thus, not only must coarse scale gradients be in the range of the interpolation operator, but also relaxation on a level must effectively eliminate gradients of scale equal to the level's mesh size. This latter condition leads to the non-standard smoother, using the level gradient operator G^l_{en}, in the algorithm at the beginning of this section. Of course, effective elimination of these level gradients by level relaxation assumes that each level operator has a curl-curl form. But all of this is reflective of a multigrid solver built on a multilevel commutativity complex: level exactness guarantees the above smoother to be effective on the level gradients.

A multilevel commutativity complex also guarantees good approximation to coarse scale near-nullspace gradient components. Consider the case where $\{S_{2h},\ W_{2h}\}$ is a nested subspace of $\{\mathcal{H}_h,\ \mathcal{ND}_h\}$ and the $P_{(\cdot)}$'s are the natural finite element interpolation operators. Because of our assumption of quasi-uniformity, finite element approximation theory implies multilevel approximation property for the algebraically smooth eigenfunctions that are gradients. For algebraic multigrid on non-nested finite elements, a multilevel approximation property for the algebraically smooth gradient eigenfunctions also holds. Physically, this seems strange since the coarse degrees of freedom in algebraic multigrid are located on coarse edges that are generally not a subset of fine edges, i.e. physically, electric fields that have zero circulation over a given closed cycle (e.g., coarse cycle) generally do not have zero circulation over closed cycles not contained in that given cycle (e.g., fine cycles). But this "strangeness" can be explained away through the commutativity relation

$$\nabla^h P^h_{H^1} = P^h_{H(\mathrm{curl})} \nabla^{2h}, \tag{3.5}$$

i.e. a field with zero circulation on a coarse cycle corresponds to a coarse gradient, whose generating potential can be interpolated to a fine scalar function and then operated on with the fine gradient operator, to produce a zero circulation field over the fine cycles. Hence, non-nesting of the coarse edges *does not pose* a problem for approximating gradient near-nullspace components.

Now, given an arbitrary coarse edge function that has a curl-free and a divergence-free component, commutativity interpolation interpolates the curl-free component correctly, and achieves this without explicitly using $P^h_{H^1}$. But the fact that the divergence-free component may not be interpolated correctly begs the question of overall multigrid efficiency. We contend that multilevel de Rham sequences are needed to give multigrid efficiency. Suppose the $\{S_{2h},\ W_{2h}\}$ generated by $P^h_{H^1}$ and $P^h_{H(\mathrm{curl})}$ contains the proper discrete $p-$forms that approximate the continuous $p-$forms of $\{H^1(\Omega),\ H(\mathrm{curl};\Omega)\}$, i.e. $P^h_{H^1}$ and P^h_{curl} are the natural finite element interpolation operators. Then the set of coarse finite elements are stable, and by the finite element approximation property for each of these spaces, we have good multigrid approximation property everywhere. For example, let $\mathbf{v}_h$ be any type of near-nullspace component and let $\mathbf{v}$ be the continuous function it approximates. On the fine and coarse grids, by finite element approximation property, we have

$$\|\mathbf{v}-\mathbf{v}_h\|_{\mathrm{curl}} \le Ch \quad \text{and} \quad \|\mathbf{v}-P^h_{\mathrm{curl}}\mathbf{v}_{2h}\|_{\mathrm{curl}} = \|\mathbf{v}-\mathbf{v}_{2h}\|_{\mathrm{curl}} \le Ch,$$

where we used the induced finite element interpolation property of P^h_{curl} in the second inequality. Thus, we have good coarse grid approximation to $\mathbf{v}_h$:

$$\|\mathbf{v}_h-P^h_{\mathrm{curl}}\mathbf{v}_{2h}\|_{\mathrm{curl}} \le Ch.$$

When a multilevel de Rham structure does not exists, $P^h_{H(\mathrm{curl})}$ is generally not Nedelec interpolation, and so,

$$\|\mathbf{v}-P^h_{H(\mathrm{curl})}\mathbf{v}_{2h}\|_{\mathrm{curl}}$$

can be large for divergence-free $\mathbf{v}$. Hence,

$$\|\mathbf{v}_h - P^h_{H(\text{curl})}\mathbf{v}_{2h}\|_{\text{curl}}$$

can be large, and multigrid performance degrades.

Except for geometric multigrid, constructing multilevel de Rham complexes is very difficult. Moreover, even if we have multilevel de Rham complexes, when the material coefficients vary substantially, $P^h_{H(\text{curl})}$ should depend on these coefficients, which then may break multilevel commutativity. When multilevel de Rham complexes do not exist, to obtain multilevel efficiency, an interpolation operator that acts like $P^h_{H(\text{curl})}$ on the curl-free components and $P^h_{H(\text{div})}$ on the divergence-free components is needed. Construction of such an operator is also difficult. The fundamental misunderstanding is that multilevel communativity considers only a subspace of the whole space, whereas good multigrid performance requires good handling of the whole space (i.e. good handling of the whole multilevel nodal decomposition [16]). For an efficient multigrid scheme, one needs some explicit type of Helmholtz decomposition on each level, and one needs interpolation operators that give good coarse grid approximation to each part of this decomposition. To achieve both of these, explicitly having nodal (to represent the curl-free vector) and edge (to represent the divergence-free vector) degrees of freedom and separate interpolation operators for each part of the Helmholtz decomposition is desirable. This will require more computer memory and additional computational cost per multigrid cycle, but will give robustness and simplicity. We consider such an algorithm in the next section.

4. $T-V$ **Formulation.** As one can derive from the last section, commutativity interpolation is difficult to construct and gives poor approximation to divergence-free error components. There are some computational reduction in using these operators since nodal interpolation is never performed in the solve phase. In some algebraic multigrid algorithms ([20], [5]), though, this reduction is only minor since the formation of auxiliary nodal interpolation and coarse grid operators are explicitly done in the setup phase but then discarded afterwards. In this section, we will develop an algorithm that recycles this discarded nodal information in the solve phase. Of course, this will lead to actual nodal structures in the solve phase. But, by keeping these structures, interpolation operators can be easily constructed without the commutativity constraints, and be constructed such that the coarse grid approximation property holds for all required bad components. The motivation behind such an approach is to develop an algorithm that can be easily fitted into the framework of existing multigrid codes.

We note that we could present this algorithm as a subspace iteration, like the one presented in [14]. However, we opt to present it in terms of a so-called $T-V$ formulation ([19]) since this formulation gives guidance on developing effective interpolation operators. We also note that the $T-V$ formulation is used only in the multigrid solution phase- we do not replace continuous problem (2.3) with a continuous semi-definite $T-V$ formulation. Nevertheless, this multigrid solver iterates over the pde system of the $T-V$ formulation. Hence, interpolation operators must be based on the non-trivial near-nullspace components of this pde system. Thus, in this section, we characterize these components. We consider the quantative form of these near-nullspace components in order to justify our choice of interpolation operators.

So, consider curl-curl equation (2.4) with the solution decomposition

$$\mathbf{E} = \mathbf{E}_1 + \nabla\phi, \qquad \mathbf{E}_1 \in H_0(\text{curl};\Omega),\ \phi \in H^1_0(\Omega),$$

and the variational problem: find $(\mathbf{E}_1, \phi) \in H_0(\text{curl};\Omega) \times H^1_0(\Omega)$ such that

$$(\alpha\nabla\times(\mathbf{E}_1 + \nabla\phi), \nabla\times(\mathbf{v} + \nabla\psi)) + (\beta(\mathbf{E}_1 + \nabla\phi), (\mathbf{v} + \nabla\psi)) = (\mathbf{f}, (\mathbf{v} + \nabla\psi)) \tag{4.1}$$

for all $\mathbf{v} \in H_0(\text{curl};\Omega)$, $\psi \in H_0^1(\Omega)$. Respectively denoting the basis functions of $H_0(\text{curl};\Omega)$ and $H_0^1(\Omega)$ as $\{\mathbf{v}_l\}$ and $\{\psi_i\}$, (4.1) can be written as a system variational problem

$$\begin{bmatrix} A_{ee} & A_{en} \\ A_{ne} & A_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{E}_1 \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{f}_e \\ \mathbf{f}_n \end{bmatrix}, \tag{4.2}$$

where

$$A_{ee} = [(\alpha\nabla\times\mathbf{v}_l, \nabla\times\mathbf{v}_m) + (\beta\mathbf{v}_l, \mathbf{v}_m)] \quad A_{en} = [(\beta\mathbf{v}_l, \nabla\psi_j)]$$
$$A_{ne} = [(\beta\nabla\psi_i, \mathbf{v}_m)] \qquad\qquad A_{nn} = [(\beta\nabla\psi_i, \nabla\psi_j)]$$

and

$$\begin{bmatrix} \mathbf{f}_e \\ \mathbf{f}_n \end{bmatrix} = \begin{bmatrix} (\mathbf{f}, \mathbf{v}_l) \\ (\mathbf{f}, \nabla\psi_i), \end{bmatrix}$$

and for all $(\mathbf{v}_l, 0)^t, (\mathbf{0}, \psi_i)^t \in H_0(\text{curl};\Omega) \times H_0^1(\Omega)$. To examine the near-nullspace component $\left(\tilde{\mathbf{E}}, \tilde{\phi}\right)^t$ of (4.2), let $\tilde{\mathbf{E}}$ have the Helmholtz decomposition

$$\tilde{\mathbf{E}} = \tilde{\mathbf{E}}_d + \tilde{\mathbf{E}}_c,$$

where $\tilde{\mathbf{E}}_d$ is divergence-free and $\tilde{\mathbf{E}}_c$ is curl-free. We first consider the quantative form of two special near-nullspace components, and then consider the quantative form of a general near-nullspace component. The two special forms are $\left(\tilde{\mathbf{E}}, \tilde{\phi}\right)^t$ where $\tilde{\mathbf{E}}$ is curl-free and $\tilde{\mathbf{E}}$ is divergence-free.

For near-nullspace component with curl-free $\tilde{\mathbf{E}}$, $\tilde{\mathbf{E}} = \tilde{\mathbf{E}}_c = \nabla\eta$, let us determine the corresponding nodal component $\tilde{\phi}$. From the second equation of (4.2), the nodal component must satisfy

$$(\beta\nabla\tilde{\phi}, \nabla\psi_i) \approx -(\beta\nabla\psi_i, \tilde{\mathbf{E}}_c) = -(\beta\nabla\psi_i, \nabla\eta) \qquad \forall i.$$

Thus $\tilde{\phi}$ is approximately $-\eta$. Substituting this into the first equation of (4.2), we have

$$(\alpha\nabla\times\tilde{\mathbf{E}}_c, \nabla\times\mathbf{v}_l) + (\beta\tilde{\mathbf{E}}_c, \mathbf{v}_l) - (\beta\mathbf{v}_l, \nabla\eta) = (\beta\nabla\eta, \mathbf{v}_l) - (\beta\mathbf{v}_l, \nabla\eta) = 0$$

so that $(\tilde{\mathbf{E}}, \tilde{\phi})^t = (\nabla\eta, -\eta)^t$ is indeed a near-nullspace component.

Next, consider the near-nullspace component with divergence-free $\tilde{\mathbf{E}}$, $\tilde{\mathbf{E}} = \tilde{\mathbf{E}}_d$. We will assume that α, β are sufficiently smooth and consider only interior estimates so that boundary terms produced by integration by parts can be omitted (near-nullspace components are locally interior). From the second equation of (4.2) and integration by parts,

$$\begin{aligned} -(\nabla\cdot\beta\tilde{\mathbf{E}}_d, \psi_i) + (\beta\nabla\tilde{\phi}, \nabla\psi_i) &= -(\nabla\beta\cdot\tilde{\mathbf{E}}_d + \beta\nabla\cdot\tilde{\mathbf{E}}_d, \psi_i) + (\beta\nabla\tilde{\phi}, \nabla\psi_i) \\ &= -(\nabla\beta\cdot\tilde{\mathbf{E}}_d, \psi_i) + (\beta\nabla\tilde{\phi}, \nabla\psi_i) \approx 0 \qquad \forall i, \end{aligned} \tag{4.3}$$

or

$$(\beta\nabla\tilde{\phi}, \nabla\psi_i) \approx (\nabla\beta\cdot\tilde{\mathbf{E}}_d, \psi_i) \qquad \forall i.$$

If we further assume that β is smooth in the sense that $||\nabla\beta|| \leq \epsilon$, then using elliptic regularity and the Cauchy-Schwarz inequality,

$$||\tilde{\phi}||_1 \leq C||\nabla\beta\cdot\tilde{\mathbf{E}}_d|| \leq C||\nabla\beta||||\tilde{\mathbf{E}}_d|| \leq C\epsilon||\tilde{\mathbf{E}}_d||.$$

That is, $\tilde{\phi}$ has small $H1$ energy, or equivalently, is a smooth eigenfunction of the diffusion operator $[\nabla \cdot \beta \nabla]$. Now for the first equation of (4.2), we have

$$(\alpha \nabla \times \tilde{\mathbf{E}}_d, \nabla \times \mathbf{v}_l) + (\beta \tilde{\mathbf{E}}_d, \mathbf{v}_l) \approx -(\beta \nabla \tilde{\phi}, \mathbf{v}_l) \qquad \forall l, \tag{4.4}$$

and because $\tilde{\phi}$ has small $H1$ energy, $\tilde{\mathbf{E}}_d$ has small scaled $H(\text{curl})$ energy in the order of the size of the eigenvalue corresponding to $\tilde{\phi}$ i.e., coercivity of the curl-curl bilinear form implies

$$||\tilde{\mathbf{E}}_d||_{H(\text{curl})} \leq C||\beta \nabla \tilde{\phi}||$$

and $\tilde{\phi}$ being a smooth eigenfuction implies

$$||\tilde{\phi}||_1 \leq ||\tilde{\phi}||_2 \leq C\lambda.$$

So, $\tilde{\mathbf{E}}_d$ is a smooth eigenfunction of the operator $[\nabla \times \alpha \nabla \times + \beta I]$. Moreover, using the identity

$$\begin{aligned} \nabla \times \alpha \nabla \times \mathbf{v} &= \nabla \alpha \times (\nabla \times \mathbf{v}) + \alpha \nabla \times \nabla \times \mathbf{v} \\ &= \nabla \alpha \times (\nabla \times \mathbf{v}) - \alpha \mathbf{\Delta} \mathbf{v} + \nabla \nabla \cdot \mathbf{v}, \end{aligned}$$

$\tilde{\mathbf{E}}_d$ is a smooth eigenfunction of

$$[-\alpha \mathbf{\Delta} + \nabla \alpha \times \nabla \times + \beta I]. \tag{4.5}$$

Thus, quantative form of two types of near-nullspace components of (4.2) are

$$\begin{pmatrix} \tilde{\mathbf{E}} \\ \tilde{\phi} \end{pmatrix} = \begin{pmatrix} \nabla \eta \\ -\eta \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \tilde{\mathbf{E}} \\ \tilde{\phi} \end{pmatrix} = \begin{pmatrix} \text{small scaled H(curl) norm} \\ \text{small scaled H1 norm} \end{pmatrix}. \tag{4.6}$$

Multigrid interpolation operators must be designed to approximate at least these components well. In fact, interpolation operators designed to handle these components will also handle other near-nullspace components. For consider the general case,

$$\tilde{\mathbf{E}} = \tilde{\mathbf{E}}_d + \tilde{\mathbf{E}}_c = \tilde{\mathbf{E}}_d + \nabla \eta.$$

Again using the second equation of (4.2) first, integration by parts leads to

$$\begin{aligned} (\tilde{\mathbf{E}}_d + \tilde{\mathbf{E}}_c, \beta \nabla \psi_i) + (\beta \nabla \tilde{\phi}, \nabla \psi_i) &= (\tilde{\mathbf{E}}_d, \beta \nabla \psi_i) + (\nabla \eta, \beta \nabla \psi_i) + (\beta \nabla \tilde{\phi}, \nabla \psi_i) \\ &= -(\nabla \beta \cdot \tilde{\mathbf{E}}_d, \psi_i) + (\beta \nabla (\eta + \tilde{\phi}), \nabla \psi_i) \\ &\approx 0, \end{aligned}$$

or

$$(\beta \nabla (\eta + \tilde{\phi}), \nabla \psi_i) \approx (\nabla \beta \cdot \tilde{\mathbf{E}}_d, \psi_i).$$

Elliptic regularity and the above smoothness assumption on β leads to

$$||\eta + \tilde{\phi}||_1 \leq C\epsilon ||\tilde{\mathbf{E}}_d||.$$

Thus, $(\eta + \tilde{\phi})$ has small scaled $H1$ norm, from which we conclude $\tilde{\phi} \approx -\eta$. Using this result in the first equation of (4.2) then gives

$$(\alpha \nabla \times \tilde{\mathbf{E}}_d, \nabla \times \mathbf{v}_l) + (\beta \tilde{\mathbf{E}}_d, \mathbf{v}_l) \approx 0.$$

Hence, the general near-nullspace components of (4.2) have the form

$$\begin{pmatrix} \tilde{\mathbf{E}} \\ \tilde{\phi} \end{pmatrix} = \begin{pmatrix} \text{small scaled H(curl) norm} + \nabla \eta \\ -\eta \end{pmatrix}.$$

But the components that must be handled well by coarse grid correction are the components that are left after relaxation. Applying a block relaxation on (4.2) for the nodes and edges, η will be smooth after relaxation on the nodes. After relaxation then, η will have small scaled H1 norm and $\tilde{\mathbf{E}}$ will have small scaled $H(\text{curl})$ norm. Therefore, interpolation will be constructed based on (4.6).

Note that the $\tilde{\mathbf{E}}$ component of the system near-nullspace is the near-nullspace of (2.4). These must be well approximated by $P^h_{H(\text{curl})}$. Note also that when α, β are constants, the divergence-free near-nullspace components are smooth eigenfunctions of $[-\alpha\boldsymbol{\Delta} + \beta I]$. Hence, to obtain good approximation to these near-nullspace components, $P^h_{H(\text{curl})}$ should act like linear interpolation on the edge degrees of freedom. Constructing $P^h_{H(\text{curl})}$ under the commutativity constraints using $P^h_{H^1}$ generally will not attain this property. A special case when this is attained occurs when the edges are nested and $P^h_{H(\text{curl})}$ is the Nedelec finite element interpolation operator (see Fig. 4.1).

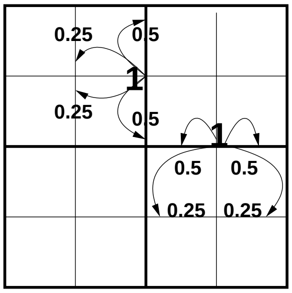

FIG. 4.1. *Nedelec linear interpolation of horizontal and vertical edges.*

Having determined the quantative form of the near-nullspace of (4.2), we now develop a multigrid algorithm. First, system (4.2) is discretized over $\mathcal{ND}(\mathcal{T}_h) \times \mathcal{H}_h(\mathcal{T}_h)$, which approximates the continuous near-nullspace well ([6]):

$$\begin{bmatrix} A^h_{ee} & A^h_{en} \\ A^h_{ne} & A^h_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{E}^h_1 \\ \psi^h \end{bmatrix} = \begin{bmatrix} \mathbf{f}^h_e \\ \mathbf{f}^h_n \end{bmatrix}. \tag{4.7}$$

From the above discussion, for the curl-free near nullspace, since $\eta = -\tilde{\phi}$, we need only to approximate the diffusion operator $[-\nabla \cdot \beta \nabla]$ on the coarser grids. Good approximation to this same operator is also required in handling the nodal $\tilde{\phi}$ component of the divergence-free near-nullspace. For the edge component of this divergence-free vector, good approximation to vector "convection"-diffusion operator (4.5) or the curl-curl operator itself is needed on the coarser levels. To accomplish both of these approximation requirements, separate interpolation operators are used for the nodal and edge degrees of freedom. The nodal interpolation P_n will be based on the diffusion operator and the edge interpolation P_e will be based on the curl-curl operator. P_n can be constructed using AMG or BoxMG techniques ([10]), and P_e can be constructed using AMGe ([9]) or AMG techniques, since the convection-diffusion operator is being implicitly coarsened. The overall interpolation operator for (4.7) is

$$\begin{bmatrix} P_e & 0 \\ 0 & P_n \end{bmatrix}.$$

The validity of this block-diagonal interpolation operator was based under the assumption that $||\nabla\beta||$ is small. When $||\nabla\beta||$ is large, a full 2×2 interpolation operator may be needed (see [11], [12] for a numerical comparison of diagonal block and full block interpolation operators on system pde's).

With block-diagonal interpolation, the coarse grid operator is formed using Galerkin coarsening:

$$\begin{aligned}\begin{bmatrix} A^{2h}_{ee} & A^{2h}_{en} \\ A^{2h}_{ne} & A^{2h}_{nn}\end{bmatrix} &= \begin{bmatrix} P_e & 0 \\ 0 & P_n\end{bmatrix}^t \begin{bmatrix} A^{h}_{ee} & A^{h}_{en} \\ A^{h}_{ne} & A^{h}_{nn}\end{bmatrix} \begin{bmatrix} P_e & 0 \\ 0 & P_n\end{bmatrix} \\ &= \begin{bmatrix} P^t_e A^h_{ee} P_e & P^t_e A^h_{en} P_n \\ P^t_n A^h_{ne} P_e & P^t_n A^h_{nn} P_n \end{bmatrix}. \end{aligned} \tag{4.8}$$

Applying this recursively, the coarse grid operator on level i can be constructed.

This algorithm requires the additional computation and storage of the off-diagonal blocks $A^{2^ih}_{en} = (A^{2^ih}_{ne})^t$. However, as the above discussion suggests, the coarse grid problems capture the divergence-free and curl-free components separately, i.e. respectively using $A^{2^ih}_{ee}$ and $A^{2^ih}_{nn}$ only. Hence, omitting the off-diagonal blocks can also lead to an effective scheme. With the off-diagonal blocks, curl-free error components re-introduced when solving the edge correction can be surpressed.

This algorithm also appears to require more user data than just the fine grid stiffness matrix A^h_{ee} and a fine grid discrete gradient operator G^h_{en} : nodes → edges, as required by commutativity-based multigrid methods. But, as in the commutativity-based methods, A^h_{nn} can be gotten using sparse matrix products and $A^h_{en} = (A^h_{ne})^t$ can be gotten freely from this same matrix product:

$$\begin{aligned} A^h_{nn} &= (G^h_{en})^t \left[A^h_{ee} G^h_{en}\right] \\ A^h_{en} &= A^h_{ee} G^h_{en}. \end{aligned}$$

Note that once these operators are formed, discrete gradient operators are not needed on the coarser levels. This apparent minor detail actually implicates some structural differences between this multigrid algorithm and commutativity-based ones. Because these operators are not needed on the coarser grids, there need not be a connection between the coarse nodal and edge grids, although the nodal and edge degrees of freedom are connected through the off-diagonal blocks of (4.8). This implies that the coarse simplices do not have to satisfy a consistency condition, i.e. nodes and edges can be coarsened separately. In turn, this means that a two-term exactness property does not hold on the coarser levels. Even if coarse simplices were consistency chosen, an exactness property may still not hold because the weights generated by the separate interpolation operators lead to discrete differential operators that may not be consistent. All of this should not be surprising because we independently constructed the edge interpolation to capture mainly the divergence-free error and the nodal interpolation to capture the curl-free error, rather than having one edge interpolation constructed to produce exactness to capture the curl-free error. There also should be no concern that the exactness property does not hold on coarser levels because the nodal coarse grid problem defines the equation for the curl-free error, i.e. we have a relaxation procedure for these errors on the coarser levels. With no exactness property on the coarser levels, the multilevel complex for this algorithm is

$$\begin{tikzcd}
H^1(\Omega) \arrow[r, "\nabla"] \arrow[d, "\Pi^h_{\mathcal{H}}"'] & H(\mathrm{curl};\Omega) \arrow[d, "\Pi^h_{\mathcal{ND}}"'] \\
\mathcal{H}_h \arrow[r, "\nabla^h"] & \mathcal{ND}_h \\
H^1_{2h} \arrow[u, "P^h_n"] \arrow[r, dash, dotted, "A^{2h}_{en}"] & H_{2h}(\mathrm{curl}) \arrow[u, "P^h_e"] \\
\vdots & \vdots \\
H^1_{2^ih} \arrow[r, dash, dotted, "A^{2^ih}_{en}"] & H_{2^ih}(\mathrm{curl}),
\end{tikzcd}$$

where $H^1_{2^ih}$ and $H_{2^ih}(\text{curl})$ are subspaces of $\mathcal{H}_h$ and $\mathcal{ND}_h$, and $\overset{A^{2^ih}_{en}}{\cdots}$ denotes only a connection between the node and edge degrees of freedom, not an exactness relation.

4.1. Interpolation Schemes. There are many well-developed interpolation schemes for variable-coefficient problems. In this paper, we will consider only structured grids, and hence, employ a BoxMG scheme ([10]) for P_n and an element agglomeration scheme ([9]) for P_e.

The BoxMG technique is well known. Let the 9-point stencil of A^h_{nn} be

$$\begin{bmatrix} a^{nw}_{nn} & a^{n}_{nn} & a^{ne}_{nn} \\ a^{w}_{nn} & a^{c}_{nn} & a^{e}_{nn} \\ a^{sw}_{nn} & a^{s}_{nn} & a^{se}_{nn} \end{bmatrix}_{IJ}$$

at coarse node IJ. This technique constructs P_n to satisfy

$$A^h_{nn} P_n \phi^{2h} = 0. \tag{4.1}$$

For fine nodes that underly a coarse node, the interpolation weight is one; for fine nodes that are horizontally or vertically adjacent to a coarse node, stencil collapsing is used to determine the weights- e.g. for fine nodes that are directly east or west of node IJ, the interpolation weights are respectively

$$P^e_n = -\frac{(a^{ne}_{nn} + a^{e}_{nn} + a^{se}_{nn})_{IJ}}{(a^{n}_{nn} + a^{c}_{nn} + a^{s}_{nn})_{IJ}} \qquad P^w_n = -\frac{(a^{nw}_{nn} + a^{w}_{nn} + a^{sw}_{nn})_{IJ}}{(a^{n}_{nn} + a^{c}_{nn} + a^{s}_{nn})_{IJ}}.$$

The weights for all other fine nodes are obtained using these computed weights and relation (4.1).

To describe the AMGe technique to construct P_e, we will refer to the two-dimensional uniform agglomerate shown in Fig. 4.2. We divide the fine edges of this agglomerate into boundary (b) and interior (i) edges, and order them so that locally A^h_{ee} and P_e can be written as

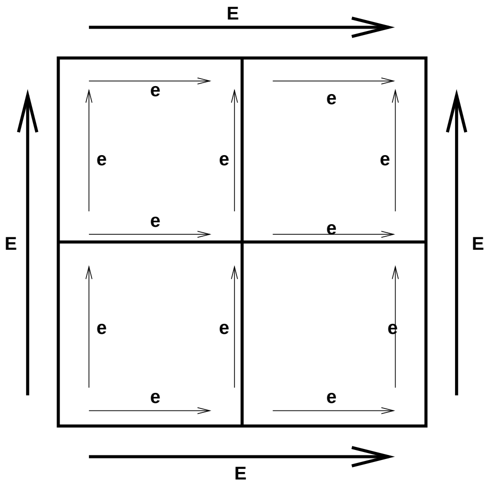

FIG. 4.2. *Coarse cell agglomerate. Coarse edges are labeled with E and fine edges are labeled with e.*

$$A^h_{ee} = \begin{bmatrix} A^{bb}_{ee} & A^{bi}_{ee} \\ A^{ib}_{ee} & A^{ii}_{ee} \end{bmatrix} \qquad P_e = \begin{bmatrix} P^b_e \\ P^i_e \end{bmatrix}.$$

The boundary edges are the fine edges e that nest a coarse edge E, and the interior edges are the remaining fine edges of the agglomerate. For the boundary edges, the interpolation weights can be defined physically. Since the edge degrees of freedom are voltage drops along the edges, the boundary weights are simply the ratio of the length of the boundary edge to the length of coarse edge that it nests. With these boundary weights, the interior weights are given by the local inversion

$$P_e^i = -[A_{ee}^{ii}]^{-1} A_{ee}^{ib} P_e^b.$$

4.2. Relaxation Schemes. Since the equations for both the nodes and edges are Poisson-like, pointwise Gauss-Seidel can be used for each degree of freedom. But as in system pde's, the sweeping order of relaxation produces different smoothers. One can relax all the nodes of the nodal grid and then all the edges of the edge grid, to give the hybrid block Gauss-Seidel scheme of [16]. One can also apply a multiplicative Schwarz smoother similar to the one of [2], but where now a node and all the edges branching out of it are solved simultanuously. This is more costly than the pointwise smoother, but it gives robustness. Yet an intermediate ordering is to update a node and then pointwise update all the edges branching out of this node in turn. Since an edge connects two nodes, a given edge is updated more than once in a sweep. Hence, this smoother is more costly than the pointwise hybrid smoother.

5. Numerical Experiments. This multigrid algorithm has currently been tested on two-dimensional uniform grids. Three-dimensional problems and algebraic multigrid variants of this algorithm will be presented in a forthcoming paper. The problems solved in this paper are (2.3) on a unit square. Problems with constant and variable α and β are examined, both standard Nedelec interpolation and the operator-based interpolations described in the previous section are used, and the smoothers described in the previous section are employed. (Note that operator-based interpolation produces multilevel structures that do not satisfy exactness.) $V(1,1)$ cycles are used, and all experiments are performed with a zero righthand side, a random initial guess, and a stopping criterion of twelve order magnitude reduction of the $l2-$norm of the initial residual. The schemes used are

- Method 1: Hybrid Gauss-Seidel smoother– all nodes updated then all edges.
- Method 2: Intermediate Gauss-Seidel smoother– pointwise update one node and then all edges branching out of it in turn.
- Method 3: Node-edge decoupling on coarse grids– i.e. only the block-diagonal terms of (4.8) are used on the coarse grids (see the discussion after (4.8)).
- Method 4: Multiplicative Schwarz smoother– simultaneously solve all edges connected to a node. The nodal degrees of freedom are not involved; this method is used for comparison.

Problem 1: $\alpha = 1$, $\beta =$ constant. This experiment was conducted to illustrate the robustness of the scheme as $\beta \to 0$, so that the operator becomes singular. Table 5.1 contains the $V(1,1)$ iteration counts. For each iteration column, the left count is for Nedelec interpolation and the right count for operator-based interpolation. From these results, all methods are robust with respect to the size of β. In particular, not preserving exactness on coarse grids does not affect convergence. Also, the number of iterations decreases as the smoother becomes "stronger" (Method 2), and Method 3 performs just as well as Method 1 because $\nabla\beta = 0$ so that the edge and nodal components of the divergence-free near-nullspace of (4.2) decouple.

Problem 2: Variable α, β. We consider the performance of this multigrid method on variable coefficient problems. Problem 2a will involve coefficients

$$\alpha = (2 + \sin 40\pi x)^2 (2 + \cos 40\pi y)^2$$

$$\beta = (2 + \cos 40\pi x)^2 (2 + \sin 40\pi y)^2,$$

and Problem 2b will involve grid-aligned jump coefficients

$$\begin{aligned} \alpha &= C(2 + \sin 40\pi x)^2 (2 + \cos 40\pi y)^2 \\ \beta &= C(2 + \cos 40\pi x)^2 (2 + \sin 40\pi y)^2, \end{aligned}$$

where jump C is described in Fig. 5.1. The results are tabulated in Table 5.2. For both problems, we see poor performance when Nedelec interpolation is used, and a dramatic improvement when operator-based interpolation is used. Note that Method 3 performs just as Method 1, which is a bit surprising. Also note that Method 2 performs better than Method 4. Both of these methods do not satisfy exactness on the coarse grids, but only Method 2 has systems for the curl-free component on the coarse grids. Lastly, note that the performance of methods 1-3 with operator-based interpolation are unaffected by jumps in the coefficients, whereas coefficient jumps noticeably affects the convergence when Nedelec interpolation is used in these methods.

0.1	10^2
10	10^4

FIG. 5.1. *Jump C of Problem 2b.*

We also consider an example that demonstrates the importance of using operator-based interpolation in the Galerkin coarsening procedure. In Table 5.3, we consider Problem 2a again but this time with operator-based interpolation used in the solve phase and Nedelec interpolation used in the Galerkin coarsening procedure. As this table shows, there is almost no improvement in using operator-based interpolation in the solve phase only (compare with the left columns of Table 5.2, Problem 2a). The coarse grid operators must be formed with these operators also.

Problem 3: Constant coefficient jumps. Our final experiment further examines jump coefficients, but now with jumps that are not grid-aligned on the coarse grids. The log scale of the coefficients are shown in Fig. 5.2 (Problem 3a) and Fig. 5.3 (Problem 3b), which has a more dramatic jump pattern for α and β. The order of magnitude of the jumps are around 10. Results are given in Table 5.4. Again operator-based interpolation performs better than Nedelec interpolation but not as dramatically this time, especially for Problem 3a. This can be accredited to the localization of the coefficient variations. Also, Problem 3b shows Method 1 performing better than Method 3.

6. Conclusions.

We presented a multigrid method for solving variable coefficient Maxwell's equations. This method does not construct interpolation operators based on multilevel commutativity constraints, and hence, does not satisfy exactness on the coarse levels. Rather, interpolation operators are separately developed for the nodes and edges, and operator-based techniques are used to capture the coefficient variations. Moreover, the relevant effects of exactness are obtained by having equations for the curl-free gradients on the coarse levels. Numerical results demonstrate the effectiveness of this algorithm.

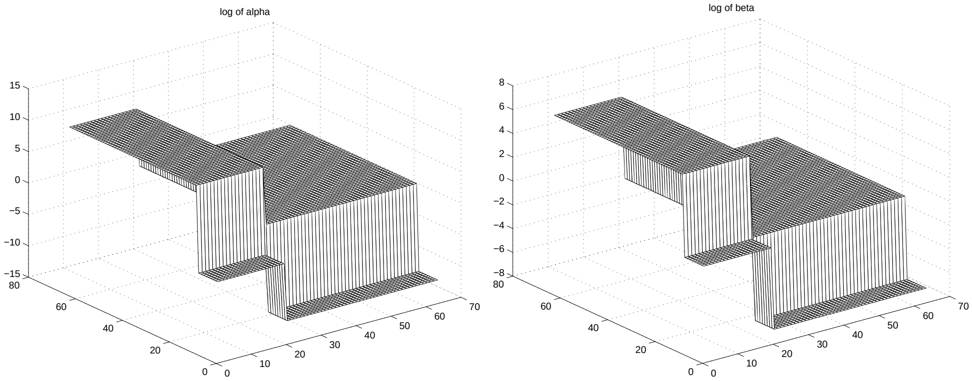

FIG. 5.2. *Log scale of jumps in α and β for Problem 3a.*

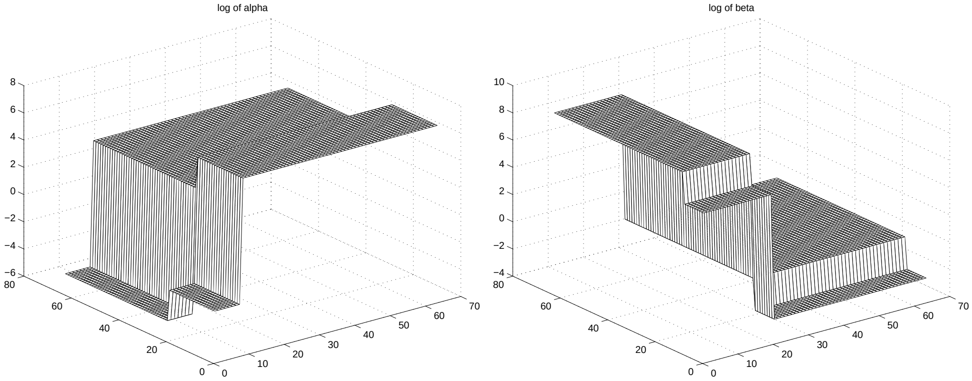

FIG. 5.3. *Log scale of jumps in α and β for Problem 3b.*

REFERENCES

[1] D. N. ARNOLD, *Differential complexes and numerical stability*, manuscript, 2003.

[2] D. N. ARNOLD, R. S. FALK, AND R. WINTHER, *Multigrid in $H(\mathrm{div})$ and $H(curl)$*, Numer. Math., 85 (2000), 197-218.

[3] D. BALDOMIR, *Geometry of Electromagnetic Systems*, Oxford University Press, New York, 1996.

[4] R. L. BISHOP AND S. I. GOLDBERG, *Tensor Analysis on Manifolds*, Dover Publication, New York, 1980.

[5] P. B. BOCHEV, C. J. GARASI, J. J. HU, A. C. ROBINSON, R. S. TUMINARO, *An improved algebraic multigrid method for solving Maxwell's equations*, SIAM J. Sci. Comp., to appear.

[6] D. BOFFI, *Fortin operator and discrete compactness for edge elements*, Numer. Math., 87 (2000), 229-246.

[7] A. BOSSAVIT, *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*, Academic Press, San Diego, 1998.

[8] S. C. BRENNER AND L. R. SCOTT, *The Mathematical Theory of Finite Element Methods*, Springer, New York, 1994.

[9] M. BREZINA, A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, J. W. RUGE, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comp., 22 (2000), 1570-1592.

[10] J. E. DENDY, JR., *Black box multigrid*, J. Comput. Phys., 48 (1982), 366-386.

[11] J. E. DENDY, JR., *Black box multigrid for systems*, Appl. Math. Comp., 19 (1986), 57-74.

[12] J. E. DENDY, JR., *Semicoarsening multigrid for systems*, ETNA, 6 (1997), 97-105.

[13] H. FLANDERS, *Differential Forms with Applications to the Physical Sciences*, Dover Publication, New York, 1989.

[14] M. GRIEBEL, *Multilevel algorithms considered as iterative methods on semidefinite systems*, SIAM J. Sci. Stat. Comp., 15 (1994), 547-565.

[15] R. HIPTMAIR, *Canonical construction of finite elements*, Math. Comp., 68 (1999), 1325-1346.

[16] R. HIPTMAIR, *Multigrid method for Maxwell's equations*, SIAM J. Numer. Anal., 36 (1999), 204-225.

[17] R. Hiptmair, *Multilevel gauging for edge elements*, Comput., 64 (2000), 97-122.

[18] R. Hiptmair, *Finite elements in computational electromagnetism*, Acta Numerica, 11 (2002), 237-340.

[19] J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley, New York, 2002.

[20] S. Reitzinger and J. Schoberl, *An algebraic multigrid method for finite element discretizations with edge elements*, Num. Lin. Alg. Appl., 9 (2002), 223-238.

[21] P. S. Vassilevski and J. Wang, *Multilevel iterative methods for mixed finite element discretizations of elliptic problems*, Numer. Math., 63 (1992), 503-520.

[22] H. Whitney, *Integration Theory*, Princeton University Press, 1957.

β	h	Method 1	Method 2	Method 3	Method 4
10^3	1/16	12/11	8/8	12/13	7/7
	1/32	19/19	12/12	20/19	10/11
	1/64	24/24	14/15	24/24	12/13
	1/128	26/26	15/15	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	26/26	15/15	26/26	13/13
10^2	1/16	17/17	11/12	18/19	10/10
	1/32	23/23	15/15	23/23	12/13
	1/64	26/26	15/16	26/26	13/13
	1/128	26/26	16/16	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	27/27	15/15	27/27	13/13
10^1	1/16	18/17	12/13	19/19	11/11
	1/32	23/24	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
	1/128	26/26	16/16	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	26/26	15/15	26/26	13/13
10^0	1/16	17/18	13/12	19/18	12/11
	1/32	23/24	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
	1/128	26/26	16/16	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	26/26	15/15	26/26	13/13
10^{-1}	1/16	17/18	12/13	17/17	12/11
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
	1/128	26/26	15/15	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	27/27	15/15	27/27	13/13
10^{-2}	1/16	18/17	13/13	16/17	11/12
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
	1/128	26/26	16/16	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	26/26	15/15	26/26	13/13
10^{-3}	1/16	17/17	13/13	17/18	11/11
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
	1/128	26/26	16/16	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	27/27	15/15	27/27	13/13
10^{-4}	1/16	17/17	13/12	18/17	11/11
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	15/16	26/26	13/13
	1/128	26/26	16/16	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	26/26	15/15	26/26	13/13
10^{-5}	1/16	17/16	13/13	17/17	11/11
	1/32	23/23	15/15	24/23	13/13
	1/64	26/26	16/16	26/26	13/13
	1/128	26/26	16/16	26/26	13/13
	1/256	26/26	15/15	26/26	13/13
	1/512	27/27	15/15	27/27	13/13

Table 5.1

V(1,1)-cycle results for Problem 1, $\alpha = 1$, $\beta =$ constant. *The iteration counts on the left and right are for Nedelec interpolation and operator-based interpolation, respectively.*

Problem	h	Method 1	Method 2	Method 3	Method 4
2a	1/16	54/15	48/11	54/17	48/18
	1/32	64/18	58/13	64/20	54/20
	1/64	71/19	59/12	71/20	53/19
	1/128	57/23	53/14	57/23	50/16
	1/256	53/25	49/15	53/25	47/13
	1/512	49/26	45/15	49/26	44/13
2b	1/16	69/13	62/11	94/15	59/44
	1/32	127/15	96/12	126/18	85/26
	1/64	108/18	85/12	110/23	77/45
	1/128	141/23	102/13	140/23	87/43
	1/256	154/25	106/15	153/25	89/31
	1/512	127/26	92/15	127/26	78/21

TABLE 5.2

V(1,1)-cycle results for variable coefficients, Problems 2a & 2b.

h	Method 1	Method 2	Method 3	Method 4
1/16	47	47	47	46
1/32	61	56	61	54
1/64	62	58	62	53
1/128	51	48	51	46
1/256	47	45	47	43
1/512	43	41	43	39

TABLE 5.3

V(1,1)-cycle with operator-based interpolation in the solve phase and Nedelec interpolation in Galerkin coarsening, Problems 2a.

Problem	h	Method 1	Method 2	Method 3	Method 4
3a	1/16	86/25	65/26	86/25	48/26
	1/32	37/18	35/11	37/18	29/12
	1/64	31/20	29/13	31/20	24/12
	1/128	31/25	30/15	31/25	26/13
	1/256	27/26	26/15	27/26	23/13
	1/512	28/26	28/15	28/26	25/13
3b	1/16	144/44	148/43	161/60	124/37
	1/32	120/38	122/38	134/46	106/26
	1/64	86/42	87/42	92/46	72/25
	1/128	80/31	81/30	78/36	68/20
	1/256	75/29	75/28	77/33	66/18
	1/512	69/27	68/25	68/30	58/17

TABLE 5.4

V(1,1)-cycle results for "non-aligned" jump coefficients.