Preprint
UCRL-JC-218750

A Novel Algebraic Multigrid-Based Approach for Maxwell's Equations

B. Lee
C. Tong

This article was submitted to Siam Journal of Scienti£c Computing

Feb. 9, 2006

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

Approved for public release; further dissemination unlimited

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any speci£c commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or re¤ect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

A NOVEL ALGEBRAIC MULTIGRID-BASED APPROACH FOR MAXWELL'S EQUATIONS

B. LEE * AND C. TONG *

Abstract. This paper presents a new algebraic multigrid-based method for solving the curl-curl formulation of Maxwell's equations discretized with edge elements. The ultimate goal of this approach is two-fold. The first is to produce a multiple-coarsening multigrid method with two approximately decoupled hierarchies branching off at the initial coarse level, one resolving the divergence-free error and the other resolving the curl-free error, i.e., a multigrid method that couples only on the finest level and mimics a Helmholtz decomposition on the coarse levels. The second consideration is to produce the hierarchies using a non-agglomerate coarsening scheme. To roughly attain this two-fold goal, this new approach constructs the first coarse level using topological properties of the mesh. In particular, a discrete orthogonal decomposition of the finest edges is constructed by dividing them into two sets, those forming a spanning tree and the complement set forming the cotree. Since the cotree edges do not form closed cycles, these edges cannot support "complete" near-nullspace gradient functions of the curl-curl Maxwell operator. Thus, partitioning the finest level matrix using this tree/cotree decomposition, the cotree-cotree submatrix does not have a large near-nullspace. Hence, a non-agglomerate algebraic multigrid method (AMG) that can handle strong positive and negative off-diagonal elements can be applied to this submatrix. This cotree operator is related to the initial coarse-grid operator for the divergence-free hierarchy. The curl-free hierarchy is generated by a nodal Poisson operator obtained by restricting the Maxwell operator to the space of gradients. Unfortunately, because the cotree operator itself is not the initial coarse-grid operator for the divergence-free hierarchy, the multiple-coarsening scheme composed of the cotree matrix and its coarsening, and the nodal Poisson operator and its coarsening does not give an overall efficient method. Algebraically, the tree/cotree coupling on the finest level, which is accentuated through smooth divergence-free error, is too strong to be handled sufficiently only on the finest level. In this new approach, these couplings are handled using oblique/orthogonal projections onto the space of discretely divergence-free vectors. In the multigrid viewpoint, the initial coarsening from the target fine level to the divergence-free subspace is obtained using these oblique/orthogonal restriction/interpolation operators in the Galerkin coarsening procedure. The resulting coarse grid operator can be preconditioned with a product operator involving a cotree-cotree submatrix and a topological matrix related to a discrete Poisson operator. The overall iteration is then a multigrid cycle for a nodal Poisson operator (the curl-free branch) coupled on the finest grid to a preconditioned Krylov iteration for the fine grid Maxwell operator restricted to the subspace of discretely divergence-free vectors. Numerical results are presented to verify the effectiveness and difficulties of this new approach for solving the curl-curl formulation of Maxwell's equations.

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

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

1. Introduction.

In a recent paper ([19]), the first author co-presented a multigrid method for the curl-curl formulation of Maxwell's equations discretized with edge elements. The focus of that paper was to advocate the explicit use of nodal and edge interpolation operators in order to give good coarse grid approximation to both the curl-free and divergence-free error components. The nodal interpolation operators can be constructed using a Dendy operator-collapsing approach ([11]-[13]) or a standard Ruge-Stuben AMG approach ([25] and [27]), both which are known to handle Poisson equations well. The edge interpolation operators can be constructed using an AMGe element agglomeration approach ([9]), which involves small local inversions to determine the interpolation weights. With these two types of interpolation operators, the coarse grid problems are generated. On any level, viewing the "hybrid" node-edge relaxation as a block Gauss-Seidel process, the nodal equations are first relaxed and then the edge equations are relaxed with an adjustment to the right-hand side given by a node-to-edge transformation of the computed nodal correction. Hence, at each

*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA. *emails*: lee123@llnl.gov, tong10@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.

level the node and edge degree of freedoms are coupled through a node-to-edge transformation operator. This is a shortfall of that method, i.e., node-to-edge operators must be computed and stored on all levels. Another shortfall is the need to construct topological relations on all levels for the element agglomeration routine. Yet another problem is that a scheme for defining the interpolation weights for the interior edges of faces for 3-d problems has not been fully developed. In this paper, we develop an approach that evades these shortfalls. In particular, this approach requires the node-to-edge variables to couple only on the finest level, and produces matrices that are more amenable to non-agglomerate AMG coarsening procedures.

An initial method that seemly achieves the above goals is to directly apply a standard AMG method to the finest level nodal Poisson operator and edge Maxwell operator. These two operators are coarsened separately so that decoupled multigrid structures are produced, with the node and edge variables coupled only on the finest level. However, a problem with this naive approach is that the generated edge coarse grid operators generally will have poor quality. Not only does the discrete edge operator have large positive off-diagonal elements, but it also has many near-nullspace gradient components. Hence, a standard AMG coarsening procedure may generate interpolation operators with poor approximation properties. But since these gradient components are handled through the Poisson solve, they should not even be considered when coarsening the edge equations. Thus, unless special care is taken in avoiding the complete gradient functions in the coarsening procedure, the quality of the interpolation and coarse grid operators will be poor. This is so even for an AMG method that handles matrices with large positive off-diagonal elements.

Although the above method fails, its multilevel structure is desirable. With the node and edge hierarchies decoupled on all coarse levels, the coarse node-to-edge transformation operators are not used. On the other hand, these transformation operators are needed for incorporating into the overall iteration the gradient error components re-introduced by the edge correction. These two conflicting conditions illustrate an inability to construct and preserve a discrete Helmholtz decomposition in the multilevel structure. Specifically, the decoupled multigrid hierarchies for the node and edge variables form a multiple coarsening of the target fine grid problem. The node branch is for the curl-free error and the edge branch is for the divergence-free error. Since the edge correction re-introduces gradient error, the coarse grid problems are not defined for orthogonal components, i.e., the branches do not mimic a Helmholtz decomposition. Hence, the branches must be bridged using the node-to-edge transformation operators. Now, because it is generally impossible even to orthogonally decouple the coarse degrees of freedom into gradients and weakly divergence-free components (since an edge element always has a gradient component), the Helmholtz decomposition should be compromised. An alternative, computable decomposition is a partitioning of the edges into those that can form "complete" gradients and those that cannot[1]. The obvious reason for choosing this partitioning is to insure a separate hierarchy for the edges that can form the troublesome complete gradients. This hierarchy in turn can be replaced by the nodal Poisson hierarchy. Moreover, although the complement set of edges do not correspond to divergence-free functions (i.e., the Nedelec basis functions on these edges do not form divergence-free functions), if the discretely divergence-free functions can be expressed in terms of the complementary set of edges, then this decomposition can lead to a good multiple-coarsening scheme. Of course, such a decomposition generally does not produce orthogonal branch hierarchies, but chosen correctly, it can lead to hierarchies that are only weakly coupled on the coarse levels. One incorrect partitioning consists of a complete gradient edge set that spans

[1]Throughout this paper, we will indiscriminately call a Nedelec basis function an edge, and vice versa. Thus, the phrase "the set of edges span a subspace" translates to "the set of Nedelec basis functions located on the set of edges span a subspace".

all gradients. In this case, the complete gradient set contains all the edges, and thus, the complement set cannot represent the divergence-free functions. Thus, the strategy considered in this paper is to form the edge set that cannot support a gradient function, so that its complementary set supports only some of the gradients. The former set is also chosen under the constraint that divergence-free functions can be expressed in terms of its edges. Having determined these orthogonal sets, the finest level edge matrix can be explicitly re-ordered into a 2-2 block matrix with each diagonal sub-block coupling within only one of these two sets. It would appear then that a multiple-coarsening scheme should involve the nodal Poisson operator and the diagonal submatrix for the edges that cannot form gradients. However, we will see that this is not so, although a modification to this approach can lead to an efficient scheme.

To extract the edges that cannot form complete gradients with the constraint that discretely divergence-free vectors can be expressed in terms of these edges, we use a simple graph search algorithm. Given a node-to-edge relation on the finest grid (for example, through the fine level gradient matrix), a graph relating the nodes and edges is available. With this graph, the edges making up a spanning tree form one of the two sets of edges, the tree edges. The remaining edges form the other edge set, the cotree edges. Such tree/cotree extraction can be achieved within $O(|\mathbf{E}^L|)$ comparisons, where $|\mathbf{E}^L|$ is the number of finest level edges. Hence, this procedure is computationally negligible. The cotree edges cannot support complete gradients and can be used to express the discretely divergence-free vectors. Also, the tree edges support only some of the gradient functions. Having extracted the tree and cotree edge sets, the fine edge matrix can be re-ordered so that the block diagonal sub-matrices involve only tree-tree or cotree-cotree connections. Since it is well-known that the edge subspace of gradients span the tree edges ([1], [15], [20], [22], [24], [26], [28]), as mentioned earlier, one would expect that only the cotree-cotree and nodal Poisson operators are needed in the solver. Such a solver would consists of a multigrid branch generated for the cotree-cotree operator and a multigrid branch generated for the nodal Poisson operator. The first step of this iteration would be to apply multigrid cycles to the Poisson equation for the purpose of sufficiently reducing the gradient error components. Error components that are not gradient-like are next reduced by applying multigrid cycles to the cotree-cotree branch. However, this method generally converges slowly. A partial explanation for this poor performance is that the tree edges are corrected only on the finest level through the nodal correction via the fine level discrete gradient operator. But including an additional multigrid branch for the tree edges does not guarantee a resolution to this problem. The poor convergence occurs because the cotree-cotree operator itself does not give a good initial coarse-grid operator for the divergence-free hierarchy. Viewing the cotree operator as the result of a Galerkin coarsening procedure applied to the target level Maxwell operator, the corresponding interpolation operator gives poor approximation to the divergence-free subspace. In particular, there are a few smooth divergence-free near-nullspace vectors that cannot be approximated through the cotree edges, and so clearly cannot be resolved by the cotree-cotree branch. These smooth vectors are supported by tree and cotree edges. Algebraically, these vectors expose the cotree-tree coupling connections. Thus, the cotree-cotree and tree-tree multigrid branches cannot eliminate these components. Now the approach presented in this paper uses an interpolation operator that relates the cotree edge vectors to the divergence-free vectors. With this interpolation and a restriction operator, the Galerkin coarsening procedure generates a good initial coarse-grid operator for the divergence-free hierarchy. The corresponding coarse-grid problems are solved using a preconditioned Krylov iteration with a preconditioner that includes a cotree-cotree multigrid branch. The complete preconditioner consists of a cotree-cotree multigrid branch and multigrid branches for operators that can

be factored into a product that includes a discrete Poisson operator. The overall solver then consists of a multigrid branch for the nodal Poisson problem and this preconditioned Krylov iteration for the divergence-free coarse grid problems.

An immediate question one may consider is whether multilevel exactness is preserved or approximated by this new method. Clearly, since the multigrid hierarchies are decoupled on the coarse levels, multilevel exactness is not even defined. But De Rham and exactness sequences exist on the finest level. Neither is multilevel exactness approximated in this method as in [19], where the nodal equations are explicitly introduced on all levels. But as expounded in [19], multilevel exactness is required only to generate representable near-nullspace gradients of all scales. Thus, the purpose of multilevel exactness is to preserve the gradient near-nullspace on the coarse levels. With this preservation, a relaxation scheme can be developed to handle these near-nullspace gradient components on all scales, countering any re-introduction of these components from the edge corrections. However, if the hierarchy of edge grids is chosen so that edge corrections cannot re-introduce complete gradients, then this counter-action may not even be needed. This is indeed what the new solver is designed to accomplish since the cotree edge correction cannot re-introduce complete gradients.

To the authors' knowledge, this is the first method based on non-agglomerate AMG for solving the curl-curl formulation of Maxwell's equations discretized with edge elements. In [5] though, an AMG method for the edge elements is derived through an auxiliary component splitting of the edge elements into Lagrangian bases. But the construction of the prolongation operator between the Langrangian bases and Nedelec bases requires a fair amount of geometric information from the triangulation, and the generated finest edge matrix used in the solver is substantially denser than the original edge matrix. The methods of [5] and this paper are totally different. As for other AMG methods for Maxwell's equations, they are based on aggregation/agglomeration coarsening, which results in coarse degree of freedoms defined on non-nested edges. These AMG methods also require multilevel exactness or an approximation to it ([23], [6], and [19]).

Turning to tree and cotree partitioning, tree extraction approaches for solving the magnetostatic problem has been utilized in the electromagnetic community for decades ([1], [7], [15], [20], [24], [28]). The basis for this extraction is the enforcement of a gauge condition, which is required in magnetostatics. Solving this problem over just the cotree edges indeed enforces this condition. Similar tree extraction methods were also developed in [14] and [26] for solving saddle-point problems. But in [26], the cotree is used to determine a divergence-free basis set. The continuous problem is then discretized over this subspace. In all these studies, tree/cotree extractions were not used in developing a solver.

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, for self-containment and since the new approach somewhat continues to adhere to the general technique, we describe the basic structure of a multigrid method for Maxwell's equation. Brief descriptions of some existing methods also are given in this section. In section 4, we review graphs, trees, and cotrees, and relate these objects to the edge finite element discretization of Maxwell's equations. We examine the cotree-cotree submatrix of the re-ordered target fine grid matrix to show why it is more amenable to AMG processing. In particular, we show that the near-nullspace of this submatrix is much smaller than the near-nullspace of the full target grid matrix. Also in this section, we perform a frequency analysis to determine which modes the cotree can and cannot eliminate. This analysis reveals the poor quality of the cotree operator for the divergence-free hierarchy. Section 5 presents the new approach. We first introduce

an interpolation operator that relates the cotree edge values to the tree edge values for discretely divergence-free vectors. With this operator, we form the initial coarse grid operator for the divergence-free branch. We relate this coarse grid operator to a product operator involving a cotree-cotree submatrix and a discrete Poisson matrix. This then leads to the preconditioned Krylov method for the divergence-free coarse grid problem. Finally, section 5 ends with descriptions of two simplified methods for regularized magnetostatics and structured 2-d problems with structured tree/cotree extractions. In section 6, we present some preliminary numerical results illustrating the effectiveness of these new approaches.

2. Curl-Curl Formulation for the Electric Field. Let Ω be a bounded simply-connected domain $\Omega \in \Re^n, n = 2, 3$. To guarantee appropriate regularity of the problem, let Ω have a smooth or polygonal boundary Γ. Assuming that the boundary surface is perfectly conducting, the curl-curl boundary-value problem for the electric field $\mathbf{E}$ is

$$\begin{aligned} \nabla\times\alpha\nabla\times\mathbf{E} + \beta\mathbf{E} &= \mathbf{f} \quad \text{in } \Omega, \\ \mathbf{n}\times\mathbf{E} &= \mathbf{0} \quad \text{on } \Gamma, \end{aligned} \tag{2.1}$$

where α and β are positive functions (in $L_\infty(\Omega)$). For time-dependent problems, the parameter β is related to the time step size ($\beta = O(\frac{1}{\Delta t})$) and may be quite small if large time steps are taken. In this case, at each time step of the solution process, an equation of the form (2.1) must be solved.

To describe the variational formulation of (2.1), 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. The variational formulation of (2.1) is then to find $\mathbf{E} \in H_0(\mathrm{curl};\Omega)$ such that

$$(\alpha\nabla\times\mathbf{E}, \nabla\times\mathbf{v}) + (\beta\mathbf{E}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \qquad \forall\mathbf{v} \in H_0(\mathrm{curl};\Omega). \tag{2.2}$$

To discretize variational problem (2.2), 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 (2.2), the discrete variational problem is to find $\mathbf{E}_h \in \mathcal{ND}(\mathcal{T}_h)$ such that

$$(\alpha\nabla\times\mathbf{E}_h, \nabla\times\mathbf{v}_h) + (\beta\mathbf{E}_h, \mathbf{v}_h) = (\mathbf{f}, \mathbf{v}_h) \qquad \forall\mathbf{v}_h \in \mathcal{ND}(\mathcal{T}_h). \tag{2.3}$$

We further 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 just in $\mathcal{ND}(\mathcal{T}_h)$ but is 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 (2.3) requires simple-connectedness of Ω. For a domain such as a torus, an additional procedure is needed to handle special troublesome subspaces of low dimension. These subspaces can be treated on a coarse grid ([17]).)

3. Review of Existing Methods. As noted in the previous section, the important property of $\mathcal{ND}(\mathcal{T}_h)$ is its explicit representation of the curl nullspace of gradients. Since the leading order term of (2.3) is the curl part, this means that the near-nullspace of (2.3) is representable in $\mathcal{ND}(\mathcal{T}_h)$. These components exist on a number of scales, from the target fine scale down to extremely coarser scales. It is in fact this multiscale, high-dimensional characteristic of the near-nullspace that hampers easy construction of efficient multigrid solvers for the Maxwell equations. In particular, there are many of these components, coarse scale ones must be both representable on the coarse grids and approximately in the range of the interpolation operators, and relaxation must effectively reduce them. Furthermore, a scheme that handles these problems must also effectively eliminate non-gradient troublesome components.

Existing multigrid solvers have been designed to handle some or all of the above problems. For geometric methods, Hiptmair is the original developer of an effective scheme ([16]). All methods, both geometric and algebraic, have essentially followed the structure of this method. The form of it is

Given $\mathbf{E}^L$ and a righthand 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, $\mathbf{E}^0 \leftarrow [A_{ee}^0]^{-1}\mathbf{f}^0$.
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($l-1$, $\mathbf{E}^{l-1}$, $[P_{l-1}^l]^t(\mathbf{f}^l - A_{ee}^l\mathbf{E}^l)$)
   d. coarse grid correction $\mathbf{E}^l \leftarrow \mathbf{E}^l + P_{l-1}^l\mathbf{E}^{l-1}$
   e. post-smooth $\mathbf{E}^l \leftarrow S_l(\mathbf{E}^l, \mathbf{f}^l)$,

where A_{ee}^l is the $l'th$ level discretized edge operator, and S_l and P_{l-1}^l are respectively the $l'th$ level smoothing and interpolation operators. The smoother has the form

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

where G_{en}^l is the level l discrete gradient operator mapping a node-based scalar potential into the Nedelec element space, and A_{nn}^l is a discrete node-based diffusion operator obtained by post- and pre-multiplying A_{ee}^l with G_{en}^l and $[G_{en}^l]^t$.

For the geometric case, when the coefficients α and β are very smooth, the interpolation can be the natural Nedelec interpolation. This interpolation operator satisfies not only the property that coarse gradients are in its range, but also that the coarse edges linearly interpolate to fine edges. This second property permits good handling of non-gradient error components.

AMG extensions of Hiptmair's method have been presented in [23], [6], and [19]. The methods of [6] are generalizations to [23]'s scheme. The central theme of [23] and [6] is to construct interpolation operators that satisfy multilevel commutativity. This means that edge interpolation operators are designed particularly to map coarse gradients to fine gradients. In the two-level case, this is symbolically expressed by the relation

$$\nabla^h P^h_{H^1} = P^h_{H(\mathrm{curl})} \nabla^{2h}, \tag{3.1}$$

where ∇^h and ∇^{2h} are fine and coarse gradient operators, and $P^h_{H^1}$ and $P^h_{H(\mathrm{curl})}$ are the nodal and edge interpolation operators, respectively. Hence, (3.1) says that edge interpolating the gradient of a coarse nodal function is equivalent to nodally interpolating this function and then taking its gradient, i.e., the end result of edge interpolating a coarse gradient is a fine gradient. Of course, edge interpolation operators constructed to satisfy this constraint generally will not handle non-gradient error components well, which in turn may lead to scalability difficulties. Nonetheless, the approach of [23] and [6] generate these interpolation operators by first applying a standard aggregration AMG procedure to the Poisson equation obtained by restricting the Maxwell operator to the gradient subspace. The chosen aggregated coarse nodal points are then joined to form non-nested coarse edges, and the constructed nodal interpolation operator is used in determining an edge interpolation operator that satisfies (3.1). With these two interpolation operators, a Galerkin procedure is applied to create the nodal and edge coarse grid operators. This whole procedure is then recursively applied down to the coarsest level. Except for the nodal interpolation operator, every part of the resulting multigrid hierarchy is used in the solve phase.

The method of [19] does not create edge interpolation operators that satisfy (3.1). Hence, it does not produce multilevel commutativity complexes. To guarantee that gradients are represented and resolved on coarser levels, the multigrid hierarchy is built for the overdetermined system operator

$$\begin{bmatrix} A^L_{ee} & A^L_{en} \\ A^L_{ne} & A^L_{nn} \end{bmatrix}, \tag{3.2}$$

where $A^L_{ne} = (A^L_{en})^t$ is a discretization of $(\nabla q, \mathbf{r}),\ q \in \mathcal{H}_L,\ \mathbf{r} \in \mathcal{ND}_L$. (This overdetermined system is used only in the solve phase, not in the continuous formulation.) The nodal interpolation P_n is constructed using a standard AMG or Dendy operator-collapsing scheme applied to A^L_{nn}, and the edge interpolation operator P_e is constructed using an AMGe scheme on A^L_{ee}. The block diagonal interpolation operator

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

is then utilized in a Galerkin procedure applied to (3.2). A recursive application of this procedure generates the multigrid hierarchy, which can be viewed as multiple-coarsening with a branch for the edge problem (A^L_{ee}) and a branch for the node problem (A^L_{nn}). Unlike the methods of [23] and [6], both the P_n's and P_e's are used in the solve phase. P_n guarantees that gradients are interpolated well and P_e, constructed without the multilevel commutativity constraint (3.1), guarantees that non-gradient near-nullspace components are interpolated well. Note that this

unconstrained method of building interpolation operators permits a level of decoupling for the choice of coarse degrees of freedom. Specifically, coarsening of nodes and edges can be performed to an extent independently, and coarse gradient operators are not needed (gradient operators couple the choice of nodes and edges).

A disadvantage of the method of [19] is the need to compute and store the A_{ne}'s at all levels. These operators describe the coupling between the edges making up gradients and the collection of edges not forming gradients, i.e., edges making up divergence-free functions. They are particularly needed in the hybrid relaxation: an edge relaxation of the whole edge grid re-introduces gradients, and hence, the nodal problem must be adjusted using A_{en} to reflect this re-introduction. An intriguing question to examine is, to what extent are these matrices needed at all levels, or equivalently, to what extent can the node and edge multigrid hierarchies be decoupled on all coarse levels? Clearly, these hierarchies can be decoupled if the node and edge problems are defined and remain in orthogonal subspaces.

4. Tree/Cotree Edge Decomposition. The obvious orthogonal decomposition we would like the node and edge multigrid hierarchies to reflect is the Helmholtz decomposition: for arbitrary $\mathbf{E} \in \mathcal{ND}_h$,

$$\mathbf{E} = \nabla p + \mathbf{E}_d, \tag{4.1}$$

where $\mathbf{E}_d$ is divergence-free. More precisely, $\mathbf{E}_d$ is weakly divergence-free in the sense that

$$(\mathbf{E}_d, \nabla q) = 0 \quad \text{for all } q \in \mathcal{H}_h.$$

The purpose of the node and edge solvers should be to eliminate only the gradient and weakly divergence-free components of the error, respectively. In this ideal situation, the two hierarchies decouple on all coarse levels. However, to attain this decoupling, the exact Helmholtz decomposition must be obtainable in $\mathcal{ND}_h$, and this is just as difficult as solving (2.3) itself. Thus, we opt for a more practical decomposition.

The form of this practical, computable decomposition can be determined by noting that branch coupling is needed to handle *complete* gradient functions re-introduced by the edge correction. The nodal correction does not re-introduce error components that can be resolvable only by the edge solver. For complete gradient error, a simple analysis shows that the smoother slowly reduces them, e.g., Gauss-Seidel is slow on an error vector which is a gradient of a nodal function, whereas Gauss-Seidel can be satisfactory on an error vector that has components of gradients but does not contain a complete gradient. In the overall iteration then, error containing only components of complete gradients can be sufficiently handled on the edge branch, but error containing complete gradients must be transferred to the nodal branch using A_{ne}^L. Thus, in order to decouple the branches, a property of the decomposition is that one of the resulting sets, the set that will be used in the divergence-free hierarchy, should not contain complete gradients. This property is moreover advantageous for AMG processing since the Maxwell operator restricted to the corresponding subspace of edge elements will not have a large near-nullspace. To obtain such decomposition, the simplest strategy is to choose a set of edges that cannot geometrically support complete gradients. With this objective, we turn to the seemingly unrelated topic of graph theory.

Definition: A graph $\mathcal{G}$ is a pair $(\mathcal{X},\mathcal{U})$, where $\mathcal{X}$ is a set of elements $\{x_i\}$ called nodes, and $\mathcal{U}$ is a subset of $\mathcal{X} \times \mathcal{X}$ called edges. For any edge $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$, node x_i is the initial point and x_j is the terminal point.

Definition: A walk $\mathcal{W}$ is an alternating sequence of nodes and edges that begins and ends with a node and with each of its edges connecting the node preceding it to the

node following it.

Definition: A cycle $\mathcal{C}$ is a walk consisting of distinct edges and nodes except the initial and terminal nodes, which are the same.

Definition: A path $\mathcal{P}$ is a walk that contains distinct nodes.

Definition: A graph is connected if there us a path between every pair of nodes.

Definition: A tree is a connected graph that has no cycles. A spanning tree is a tree that connects every node of $\mathcal{G}$.

Of particular interest is the extraction of a spanning tree from a graph. This can be achieved in $O(|\mathcal{U}|)$ comparisons using a standard first-in/first-out (FIFO) search algorithm [10]. The subset of edges removed from the graph is called the cotree. Each cotree edge can be added to a path in the tree to create a cycle.

Now let $\mathcal{G}^L$ be the target fine grid where Maxwell equation (2.3) is discretized on. Since $\mathcal{G}^L$ consists of nodes and edges, it forms a graph. The first step in our proposed algorithm is to decompose $\mathcal{G}^L$ into a spanning tree and cotree. Clearly, this creates a discrete orthogonal decomposition of the edges of $\mathcal{G}^L$. More relevant are the properties of the cotree and tree. First, there does not exist a non-trivial gradient function supported by only cotree edges. For suppose otherwise, that there is an edge function $\tilde{\mathbf{E}} = \sum_{i \in \text{cotree}} \tilde{\mathbf{E}}_i N_i = \nabla\phi$ for some $\phi \in \mathcal{H}_h$. Consider cotree edge $\mathbf{e}_i$ with initial and terminal nodes x_i and x_j. Since a spanning tree connects all nodes of $\mathcal{G}^L$, there is a path in the tree that connects x_i to x_j such that adding cotree edge $\mathbf{e}_i$ to this path forms a closed loop l. Because $\tilde{\mathbf{E}}$ is defined on the whole grid but is non-zero only on the cotree edges, and because the domain is simply-connected, we have

$$0 = \oint_l \nabla\phi \cdot \mathbf{t}\, d\Gamma = \oint_l \tilde{\mathbf{E}} \cdot \mathbf{t}\, d\Gamma = \int_{\text{cotree edge}_i} \tilde{\mathbf{E}} \cdot \mathbf{t}\, d\Gamma = \tilde{\mathbf{E}}_i.$$

Repeating this procedure for each cotree edge, $\tilde{\mathbf{E}}$ must be identically zero.

Although a complete gradient cannot be supported by only cotree edges, it definitely will be shared between the tree and cotree edges. Such partitioning fragments a majority of the full gradients suported on $\mathcal{G}^L$. This implies that the tree supports far fewer gradients than the whole grid. An example of this reduction is illustrated in Figure 4.1 for a structured 2-d grid. In the interior of the grid, there exists 9 gradient functions given by taking the gradient of the bilinear hat functions centred at each of the interior nodes. Each of these local gradient functions are fragmented into tree and cotree edges so that none of them are supported only on the tree or cotree grids, although the gradients of appropriate linear combinations of hat functions can be supported on the tree grid.

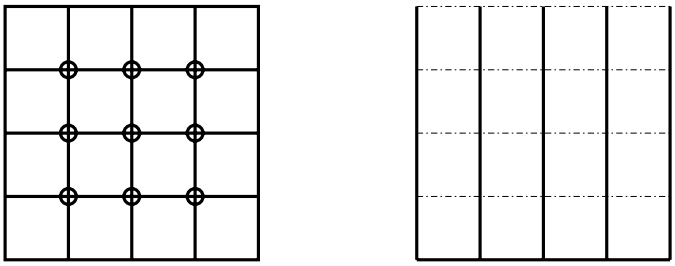

FIG. 4.1. *Left diagram: whole grid $\mathcal{G}^L$; right diagram: tree= solid lines and cotree= hashed lines.*

Thus, the cotree grid does not support a full near-nullspace vector of the Maxwell

operator and the tree grid supports a dramatically fewer number of them. The next task, which pertains to AMG coarsening of the edges, is to determine precisely the near-nullspace components of the Maxwell operator restricted to the cotree and tree grids. Re-ordering matrix A^L_{ee} into

$$\begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \\ A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix}, \tag{4.2}$$

where $A^L_{ee,cc}$ and $A^L_{ee,tt}$ are respectively the cotree-cotree and tree-tree submatrices, and $A^L_{ee,ct}$ and $A^L_{ee,tc}$ are the cotree-tree and tree-cotree cross-coupling submatrices, we examine the near-nullspaces of the diagonal blocks.

In the preceding, we will distinguish between the near-nullspace functions that are annihilated by the curl term of the Maxwell operator and the near-nullspace functions that give relatively small $H(\text{curl})$ semi-norm, $\|\nabla\times\cdot\|$. We will call the former oscillatory, although they may not be geometrically nor algebraically oscillatory. The latter functions will be called smooth. In the continuum, disregarding boundary conditions, gradients are oscillatory and divergence-free eigenfunctions corresponding to the small eigenvalues of the vector Laplacian are smooth:

$$\begin{aligned}(\nabla\times\mathbf{E},\nabla\times\mathbf{E}) &= |(\nabla\times\nabla\times\mathbf{E},\mathbf{E})| \\ &= |(-\underline{\Delta}\mathbf{E}+\nabla\nabla\cdot\mathbf{E},\mathbf{E})| \\ &= \lambda(\mathbf{E},\mathbf{E}).\end{aligned}$$

For example, in 2-d, a smooth function is $\mathbf{E} = (\sin\pi x\sin\pi y, \cos\pi x\cos\pi y)^t$ with $\lambda = 2\pi^2$.

We first consider the oscillatory components of the cotree-cotree operator. Assume that there is an oscillatory coefficient vector $\mathbf{E}_c \in \Re^{n_c}$, where n_c is the dimension of the cotree set. The finite element function, which we denote by $\mathbf{E}_c$ also, corresponds to the padded extension of the coefficient vector. Since $\alpha > 0$,

$$0 = (\alpha\nabla\times\mathbf{E}_c,\nabla\times\mathbf{E}_c) \geq \alpha_{\min}(\nabla\times\mathbf{E}_c,\nabla\times\mathbf{E}_c)$$

so that $\nabla\times\mathbf{E}_c = \mathbf{0}$. But since the domain is simply-connected, this implies the contradiction that $\mathbf{E}_c$ is a non-trivial gradient supported only on the cotree edges. Turning to the tree-tree operator, clearly the gradients supported on the tree edges are oscillatory. Using the above padded extension argument, all other functions supported on only tree edges are not oscillatory. Thus, the only oscillatory near-nullspace components of the restricted Maxwell operators are the gradients supported on the tree grid.

Observe that applying the padded extension procedure and using the positive-definiteness of A^L_{ee}, both $A^L_{ee,cc}$ and $A^L_{ee,tt}$ are also positive-definite. Moreover, because $A^L_{ee,cc}$ has no oscillatory components, the curl part of $A^L_{ee,cc}$ is positive-definite itself. This will be relevant later when we introduce a preconditioner based on the curl part of $A^L_{ee,cc}$.

To examine the smooth near-nullspace of $A^L_{ee,cc}$ and $A^L_{ee,tt}$, we consider the frequencies of the full matrix A^L_{ee} and assume that β is constant and relatively small so that the gradients are indeed the near-nullspace components of A^L_{ee}. In fact, we will assume that the gradients satisfy

$$[\nabla p]^t A^L_{ee}\nabla p = (\beta\nabla p,\nabla p) = \epsilon,$$

for small ϵ. Now for an arbitrary function $\mathbf{E} \in \mathcal{ND}_L$ with a discrete version of Helmholtz decomposition (4.1), we have

$$\begin{aligned}\mathbf{E}^t A^L_{ee}\mathbf{E} &= [\mathbf{E}_d]^t A^L_{ee}\mathbf{E}_d + (\beta\nabla p,\nabla p) \\ &= [\mathbf{E}_d]^t A^L_{ee}\mathbf{E}_d + \epsilon.\end{aligned}$$

Moreover, for weakly divergence-free functions in $\mathcal{ND}_L$, there exists the bound

$$\|\mathbf{E}_d\|^2 \leq C\|\nabla \times \mathbf{E}_d\|^2$$

[18], and hence, for arbitrary $\mathbf{E} \in \mathcal{ND}_L$,

$$C_1\|\nabla \times \mathbf{E}_d\|^2 + \epsilon \leq \mathbf{E}^t A_{ee}^L \mathbf{E} \leq C\|\nabla \times \mathbf{E}_d\|^2 + \epsilon. \tag{4.3}$$

In particular, $\mathbf{E}$ is a smooth frequency of A_{ee}^L if and only if it contains a smooth divergence-free frequency. Next, consider a coefficient vector defined on the cotree or tree grid, $\tilde{\mathbf{E}}_i, i = c, t$. Appropriately padding $\tilde{\mathbf{E}}_i$ with zeros so that it belongs in $\Re^{|\mathbf{E}^L|}$, this extension has decomposition (4.1). Thus, $\tilde{\mathbf{E}}_i$ is a smooth near-nullspace of $A_{ee,ii}^L$ if and only if its padded extension contains a smooth divergence-free component of A_{ee}^L. For example, with $(\nabla p + \mathbf{E}_d)$, $\mathbf{E}_d \neq \mathbf{0}$, denoting the Helmholtz decomposition of the padded extension of the cotree vector $\tilde{\mathbf{E}}_c$, we have

$$\begin{aligned}
[\tilde{\mathbf{E}}_c]^t A_{ee,cc}^L \tilde{\mathbf{E}}_c &= \begin{pmatrix} \tilde{\mathbf{E}}_c \\ \mathbf{0} \end{pmatrix}^t \begin{bmatrix} A_{ee,cc}^L & A_{ee,ct}^L \\ A_{ee,tc}^L & A_{ee,tt}^L \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{E}}_c \\ \mathbf{0} \end{pmatrix} \\
&= (\nabla p + \mathbf{E}_d)^t A_{ee}^L (\nabla p + \mathbf{E}_d) \\
&= [\mathbf{E}_d]^t A_{ee}^L \mathbf{E}_d + \epsilon,
\end{aligned}$$

so that (4.3) gives the desired result. This relation expresses the fact that the tree/cotree decomposition is not a Helmholtz decomposition.

A special type of smooth cotree frequencies are ones whose padded extensions are themselves approximately divergence-free. This special form will be exploited to develop a multigrid-based solver for regularized magnetostatics. On the other hand, the above analysis shows that the cotree correction can resolve only some of the divergence-free error of A_{ee}^L. There are smooth divergence-free components of A_{ee}^L that cannot be approximated well by the smooth cotree and tree frequencies. For example, consider the smooth divergence-free component $\mathbf{E} = (\mathbf{E}_c, \mathbf{E}_t)^t$, $\mathbf{E}_t \neq \mathbf{0}$, of A_{ee}^L. Then for any smooth cotree frequency $(\mathbf{E}_c, \mathbf{0})^t$, we have

$$\left\| \mathbf{E} - \begin{pmatrix} \mathbf{E}_c \\ \mathbf{0} \end{pmatrix} \right\| \geq \|\mathbf{E}_t\|,$$

which can be large. Similarly, the smooth tree frequencies can give poor approximation to $\mathbf{E}$. This shows that a multiple-coarsening scheme consisting of a multigrid hierarchy for the cotree-cotree submatrix and a multigrid hierarchy for the nodal Poisson matrix will converge slowly because these smooth divergence-free components are handled only by the finest level relaxation. In general, adding a multigrid hierarchy for the tree-tree submatrix will not substantially improve the rate either. This is especially true if these smooth divergence-free components expose strong cotree-tree connections.

Summarizing, the ideal structure we would like the multiple-coarsening scheme to reflect is a Helmholtz decomposition. But to obtain a sufficient branch decoupling, the computable tree/cotree decomposition can be used instead. This latter decomposition though does not permit good approximation to all smooth divergence-free functions of the target fine edge matrix. In fact, as of now, a connection between the cotree space and the divergence-free space has not even been established.

5. The New Approach. The purpose of the cotree subspace is to weaken the coupling between the edge and node branches and to approximate the divergence-free subspace. So far, we have shown that the cotree subspace indeed weakens the branch coupling, but this subspace by itself cannot approximate the whole divergence-free

subspace. In fact, the only established connection between these two subspaces is that padded smooth frequencies of the cotree-cotree operator contain smooth divergence-free frequencies of the target fine edge operator. Unfortunately, because of the form of these padded frequencies, only certain types of divergence-free functions can be extracted from the cotree corrections. To enable a better approximation to the divergence-free subspace, a relation, i.e., an interpolation, between the cotree and divergence-free subspaces must be developed. To this end, we consider the discrete gradient and divergence operators.

The fine discrete gradient operator G^L_{en} has dimensions $|\mathbf{E}^L| \times n^L$, where n^L is the size of fine nodal grid. To be precise, one node is taken to be grounded and the corresponding column removed from G^L_{en}. Let this grounded operator be denoted also by G^L_{en}. Its transpose is the discrete divergence operator defined on the fine edge grid. Using the tree/cotree partitioning, G^L_{en} can be re-ordered as

$$\hat{G}_{en} := \begin{bmatrix} G_{en,c} \\ G_{en,t} \end{bmatrix},$$

where $G_{en,c}$ and $G_{en,t}$ correspond respectively to the cotree and tree rows of G^L_{en}. Now, $G_{en,t}$ is sparse and invertible. Thus,

$$\tilde{G} := \hat{G}_{en} G^{-1}_{en,t} = \begin{bmatrix} F \\ I \end{bmatrix},$$

where $F = G_{en,c} G^{-1}_{en,t}$. $\tilde{G}$ is an incidence matrix for the independent cutsets of graph $\mathcal{G}^L$ [21]. It can be constructed without computing $G^{-1}_{en,t}$ in the traditional way. Throughout this paper, we will assume that $\tilde{G}$ has been formed using a graph algorithm or $G^{-1}_{en,t}$ has been efficiently decomposed (or factorized so that only the action of solving a linear system involving $G_{en,t}$) using a sparse matrix algorithm so that its formation is comparatively cheap (recall that the number of nodes can be substantially smaller than the number of edges and G^L_{en} has only two elements per row). Corresponding to $\tilde{G}$ is a transformed divergence operator whose nullspace is again the discrete divergence-free space: for divergence-free $\mathbf{E}$,

$$0 = \hat{G}^t_{en}\mathbf{E} = \begin{bmatrix} G^t_{en,c} & G^t_{en,t} \end{bmatrix} \begin{pmatrix} \mathbf{E}_c \\ \mathbf{E}_t \end{pmatrix},$$

and hence,

$$\begin{aligned} 0 = G^{-1,t}_{en,t}\hat{G}^t_{en}\mathbf{E} &= \begin{bmatrix} G^{-1,t}_{en,t}G^t_{en,c} & I \end{bmatrix} \begin{pmatrix} \mathbf{E}_c \\ \mathbf{E}_t \end{pmatrix} \\ &= \begin{bmatrix} F^t & I \end{bmatrix} \begin{pmatrix} \mathbf{E}_c \\ \mathbf{E}_t \end{pmatrix} = \tilde{G}^t\mathbf{E}. \end{aligned}$$

The last expression gives a relation between the cotree vectors and the divergence-free vectors:

$$0 = \begin{bmatrix} F^t & I \end{bmatrix} \begin{pmatrix} \mathbf{E}_c \\ \mathbf{E}_t \end{pmatrix} \Rightarrow \mathbf{E}_t = -F^t\mathbf{E}_c.$$

The interpolation operator taking a cotree function to a divergence-free function is then

$$\begin{bmatrix} I \\ -F^t \end{bmatrix}.$$

Consider the subspace of discretely divergence-free vectors

$$V := \left\{ \mathbf{E} \in \Re^{|\mathbf{E}^L|} : \mathbf{E} = \begin{pmatrix} I \\ -F^t \end{pmatrix} \mathbf{E}_c \right\}.$$

Given the re-ordered fine linear system

$$\begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \\ A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix} \begin{pmatrix} \mathbf{E}_c \\ \mathbf{E}_t \end{pmatrix} = \begin{pmatrix} \mathbf{f}_c \\ \mathbf{f}_t \end{pmatrix}, \tag{5.1}$$

the systems

$$\begin{bmatrix} I & -F \end{bmatrix} \begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \\ A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix} \begin{bmatrix} I \\ -F^t \end{bmatrix} \mathbf{E}_c = \mathbf{f}_c - F\mathbf{f}_t \tag{5.2}$$

and

$$\begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \\ A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix} \begin{bmatrix} I \\ -F^t \end{bmatrix} \mathbf{E}_c = \mathbf{f}_c \tag{5.3}$$

are respectively orthogonal and oblique projections onto V. Equivalently, (5.2) and (5.3) can be viewed as the initial Galerkin coarsening onto the divergence-free subspace with interpolations and restrictions $P = \begin{bmatrix} I \\ -F^t \end{bmatrix}$ and $R = P^t$, and $P = \begin{bmatrix} I \\ -F^t \end{bmatrix}$ and $R = \begin{bmatrix} I & 0 \end{bmatrix}$. The coarse grid operators are

$$\left[A^L_{ee,cc} - A^L_{ee,ct}F^t - FA^L_{ee,tc} + FA^L_{ee,tt}F^t\right] \tag{5.4}$$

and

$$\left[A^L_{ee,cc} - A^L_{ee,ct}F^t\right]. \tag{5.5}$$

These two coarse grid operators clearly are not suitable for further coarsening since they themselves would require explicit formation. However, because their actions do not require their formation, these initial coarse grid problems can be solved using preconditioned Krylov iterations with preconditioners that are more amenable to multigrid coarsening.

To develop such preconditioners, let $\hat{A}^L_{ee}$ denote the curl part of A^L_{ee}. $\hat{A}^L_{ee}$ can be factored as

$$\hat{A}^L_{ee} = C^t M_\alpha C,$$

where C is the discrete edge-to-face curl operator and M_α is the mass matrix weighted by coefficient α :

$$[M_\alpha]_{ij} = \int_\Omega \alpha\, N_i \cdot N_j \, d\Omega.$$

Re-ordering C also with the tree/cotree partitioning and using the topological relation $CG^L_{en} = \mathbf{0}$, we have

$$C \Rightarrow \begin{bmatrix} C_c & C_t \end{bmatrix} = \begin{bmatrix} C_c & -C_cF \end{bmatrix} = C_c \begin{bmatrix} I & -F \end{bmatrix}$$

so that

$$\hat{A}^L_{ee} = C^t M_\alpha C \Rightarrow \begin{bmatrix} I \\ -F^t \end{bmatrix} C_c^t M_\alpha C_c \begin{bmatrix} I & -F \end{bmatrix} = \begin{bmatrix} C_c^t M_\alpha C_c & -C_c^t M_\alpha C_c F \\ -F^t C_c^t M_\alpha C_c & F^t C_c^t M_\alpha C_c F \end{bmatrix}.$$

But this is latter block matrix is exactly

$$\begin{bmatrix} \hat{A}^L_{ee,cc} & \hat{A}^L_{ee,ct} \\ \hat{A}^L_{ee,tc} & \hat{A}^L_{ee,tt} \end{bmatrix}.$$

Thus, we obtain

$$\begin{bmatrix} \hat{A}^L_{ee,cc} & \hat{A}^L_{ee,ct} \\ \hat{A}^L_{ee,tc} & \hat{A}^L_{ee,tt} \end{bmatrix} = \begin{bmatrix} \hat{A}^L_{ee,cc} & -\hat{A}^L_{ee,cc}F \\ -F^t\hat{A}^L_{ee,cc} & F^t\hat{A}^L_{ee,cc}F \end{bmatrix}, \tag{5.6}$$

which expresses the tree-tree and cross-coupling submatrices in terms of the cotree-cotree submatrix. To utilize (5.6) in preconditioners for (5.4) and (5.5), recall that for the subspace of divergence-free functions in $\mathcal{ND}_L$, the equivalence

$$\|\mathbf{E}\|_{H(\mathrm{curl})} \sim \|\nabla \times \mathbf{E}\|, \qquad \mathbf{E} \in \mathcal{ND}_L \text{ and divergence} - \text{free}$$

holds. Thus, $\hat{A}^L_{ee}$ and A^L_{ee} are equivalent in this subspace with an equivalence constant dependent on the material coefficients α and β. We hence can assume that this is approximately so in V. With this assumption, (5.6) implies

$$\begin{aligned} \left[A^L_{ee,cc} - A^L_{ee,ct}F^t - FA^L_{ee,tc} + FA^L_{ee,tt}F^t\right] & \\ \sim \left[\hat{A}^L_{ee,cc} - \hat{A}^L_{ee,ct}F^t - F\hat{A}^L_{ee,tc} + F\hat{A}^L_{ee,tt}F^t\right] & \\ = \left[\hat{A}^L_{ee,cc} + \hat{A}^L_{ee,cc}FF^t + FF^t\hat{A}^L_{ee,cc} + FF^t\hat{A}^L_{ee,cc}FF^t\right] & \\ = \hat{A}^L_{ee,cc}\left[I + FF^t\right] + FF^t\hat{A}^L_{ee,cc}\left[I + FF^t\right] & \\ = \left[I + FF^t\right]\hat{A}^L_{ee,cc}\left[I + FF^t\right] & \end{aligned} \tag{5.7}$$

and

$$\begin{aligned} \left[A^L_{ee,cc} - A^L_{ee,ct}F^t\right] &\sim \left[\hat{A}^L_{ee,cc} - \hat{A}^L_{ee,ct}F^t\right] \\ &= \hat{A}^L_{ee,cc}\left[I + FF^t\right]. \end{aligned} \tag{5.8}$$

These are the preconditioner operators for (5.4) and (5.5). But, whereas $\hat{A}^L_{ee,cc}$ can be coarsened using an appropriate AMG algorithm that can handle matrices with strong positive and negative off-diagonals, since $[I + FF^t]$ must be formed and can be dense, a naive application of (5.7) and (5.8) is not efficient. To obtain an efficient procedure, note that

$$\begin{aligned} \left[I + F^tF\right] &= \begin{bmatrix} I & F^t \end{bmatrix}\begin{bmatrix} I \\ F \end{bmatrix} \\ &= \begin{bmatrix} I & G^{-1,t}_{en,t}G^t_{en,c} \end{bmatrix}\begin{bmatrix} I \\ G_{en,c}G^{-1}_{en,t} \end{bmatrix}. \end{aligned}$$

Pre- and post-multiplying this expression with $G^t_{en,t}$ and $G_{en,t}$ gives

$$G^t_{en,t}\left[I + F^tF\right]G_{en,t} = \begin{bmatrix} G^t_{en,t} & G^t_{en,c} \end{bmatrix}\begin{bmatrix} G_{en,t} \\ G_{en,c} \end{bmatrix} = \hat{G}^t_{en}\hat{G}_{en},$$

a discrete Poisson operator that can be computed efficiently since $\hat{G}_{en}$ is sparse (two elements per row). A standard Ruge-Stuben coarsening procedure then can be applied to $\hat{G}^t_{en}\hat{G}_{en}$. This can be utilized in solving equation

$$\left[I + FF^t\right]\mathbf{x} = \mathbf{y}, \tag{5.9}$$

which is required in (5.4) and (5.5). Specifically, multiplying (5.9) by F^t, we obtain

$$F^t\left[I + FF^t\right]\mathbf{x} = \left[I + F^tF\right]F^t\mathbf{x} = F^t\mathbf{y}. \tag{5.10}$$

Having determined $F^t\mathbf{x}$, (5.9) gives

$$\mathbf{x} = \mathbf{y} - FF^t\mathbf{x}. \tag{5.11}$$

Thus, the whole algorithm for solving (5.9) is

Algorithm $\left[\mathbf{I}+\mathbf{FF^t}\right]$ Solve:

1. $\mathbf{w}_1 = F^t\mathbf{y}$
2. $\mathbf{w}_2 = G^t_{en,t}\mathbf{w}_1$
3. solve $\hat{G}^t_{en}\hat{G}_{en}\mathbf{z}_1 = \mathbf{w}_2$
4. $\mathbf{z}_2 = G_{en,t}\mathbf{z}_1$
5. $\mathbf{x} = \mathbf{y} - F\mathbf{z}_2$.

Steps 2-4 solve $[I + F^tF]\,F^t\mathbf{x} = F^t\mathbf{y}$ using the transformed operator $\hat{G}^t_{en}\hat{G}_{en}$. Note that steps 1-2 can be combined so that just $\mathbf{w}_2 = G^t_{en,c}\mathbf{y}$ has to be computed:

$$\mathbf{w}_2 = G^t_{en,t}\mathbf{w}_1 = G^t_{en,t}F^t\mathbf{y} = G^t_{en,t}G^{-1,t}_{en,t}G^t_{en,c}\mathbf{y} = G^t_{en,c}\mathbf{y}.$$

Also note that neither FF^t nor F^tF need to be explicitly formed, and step 3 can be replaced by several multigrid cycles. A full application of preconditioners (5.7) and (5.8) thus requires three and two sets of multigrid cycles, respectively. The algorithm for solving the preconditioner systems

$$\left[I + FF^t\right]\hat{A}^L_{ee,cc}\left[I + FF^t\right]\mathbf{x} = \mathbf{y} \tag{5.12}$$

or

$$\hat{A}^L_{ee,cc}\left[I + FF^t\right]\mathbf{x} = \mathbf{y} \tag{5.13}$$

is

Algorithm Preconditioner Solve:

1. For system (5.12), solve
$$\left[I + FF^t\right]\mathbf{z} = \mathbf{y}$$
using "Algorithm $[I + FF^t]$ Solve".
Else, for system (5.13),
$$\mathbf{z} = \mathbf{y}.$$
2. Solve
$$\hat{A}^L_{ee,cc}\mathbf{w} = \mathbf{z}$$
using several multigrid cycles.
3. Solve
$$\left[I + FF^t\right]\mathbf{x} = \mathbf{w}$$
using "Algorithm $[I + FF^t]$ Solve".

Finally, the complete multiple-coarsening algorithm for solving Maxwell's equation (2.3) given the tree/cotree partitioning is

Nodal/Cotree Multiple-Coarsening Cycle:

1. Fine grid edge relaxation on $A^L_{ee}\mathbf{E}^L = \mathbf{f}^L$
2. Residual computation: $\mathbf{r}^L = \mathbf{f}^L - A^L_{ee}\mathbf{E}^L$
3. Nodal problem: multigrid cycles on $A^L_{nn}p^L = G^t_{en}\mathbf{r}^L$
4. Nodal correction: $\mathbf{E}^L \leftarrow \mathbf{E}^L + G_{en}p^L$
5. Cotree righthand side: respectively for coarse grid problems (5.4) and (5.5),
$$\mathbf{r}_c = \left[\begin{array}{cc} I & -F \end{array}\right]\left\{\left(\begin{array}{c}\mathbf{f}_c\\ \mathbf{f}_t\end{array}\right) - \left[\begin{array}{cc} A^L_{ee,cc} & A^L_{ee,ct}\\ A^L_{ee,tc} & A^L_{ee,tt}\end{array}\right]\left(\begin{array}{c}\mathbf{E}_c\\ \mathbf{E}_t\end{array}\right)\right\}$$
$$\mathbf{r}_c = \left\{\mathbf{f}_c - \left[\begin{array}{cc} A^L_{ee,cc} & A^L_{ee,ct}\end{array}\right]\left(\begin{array}{c}\mathbf{E}_c\\ \mathbf{E}_t\end{array}\right)\right\}$$

6. Cotree problem: respectively for (5.4) and (5.5), solve

$$\left[A^L_{ee,cc} - A^L_{ee,ct}F^t - FA^L_{ee,tc} + FA^L_{ee,tt}F^t\right]\mathbf{w} = \mathbf{r}_c$$

$$\left[A^L_{ee,cc} - A^L_{ee,ct}F^t\right]\mathbf{w} = \mathbf{r}_c$$

using a preconditioned Krylov iteration with "Algorithm Preconditioner Solve"
7. Edge correction:

$$\mathbf{E}^L \leftarrow \mathbf{E}^L + \begin{pmatrix} \mathbf{w} \\ -F^t\mathbf{w} \end{pmatrix}$$

The effectiveness of this overall method depends on the performance of the preconditioned Krylov iteration. This performance in turn depends on the equivalence between A^L_{ee} and $\hat{A}^L_{ee}$. For smooth coefficients with $\beta < \alpha$, this equivalence is favourable. But for $\alpha \ll \beta$, this equivalence degrades. In this latter case, replacing steps 5-6 with a diagonal scaled conjugate gradient iteration applied to

$$A^L_{ee}\mathbf{v} = \mathbf{f}^L - A^L_{ee}\mathbf{E}^L$$

can be used instead.

5.1. Special Case: Regularized Magnetostatics. The equation for regularized magnetostatics is

$$\nabla \times \alpha\nabla \times \mathbf{E} + \hat{\epsilon}\mathbf{E} = \mathbf{f} \tag{5.14}$$

with $\hat{\epsilon} \ll 1$. Assuming that $\alpha = O(1)$, the method just described is ideal for solving the Nedelec discretization of (5.14). But because of the special form of (5.14), a simpler, more efficient algorithm can be used. We give an intuitive derivation of this algorithm.

Consider (5.14) with $\hat{\epsilon}$ constant and $\mathbf{f}$ divergence-free. Then the solution $\mathbf{E}$ must be divergence-free itself:

$$\nabla \cdot [\nabla \times \alpha\nabla \times \mathbf{E} + \hat{\epsilon}\mathbf{E}] = \hat{\epsilon}\nabla \cdot \mathbf{E} = \nabla \cdot \mathbf{f} = 0.$$

Moreover, since $\hat{\epsilon} \ll 1$, not only is the equivalence between A^L_{ee} and $\hat{A}^L_{ee}$ favourable, but also for the re-ordered matrices, we have

$$\begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \\ A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix} \approx \begin{bmatrix} \hat{A}^L_{ee,cc} & -\hat{A}^L_{ee,cc}F \\ -F^t\hat{A}^L_{ee,cc} & F^t\hat{A}^L_{ee,cc}F \end{bmatrix}.$$

Further, if $\mathbf{f}$ is discretely divergence-free, then

$$\begin{aligned} \begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \\ A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix} \begin{pmatrix} \mathbf{E}^L_c \\ \mathbf{E}^L_t \end{pmatrix} &\approx \begin{bmatrix} \hat{A}^L_{ee,cc} & -\hat{A}^L_{ee,cc}F \\ -F^t\hat{A}^L_{ee,cc} & F^t\hat{A}^L_{ee,cc}F \end{bmatrix} \begin{pmatrix} \mathbf{E}^L_c \\ \mathbf{E}^L_t \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{f}_c \\ -F^t\mathbf{f}_c \end{pmatrix}. \end{aligned} \tag{5.15}$$

Observe that the two equations of the equality relation are compatible, i.e.,

$$\hat{A}^L_{ee,cc}\mathbf{E}^L_c - \hat{A}^L_{ee,cc}F\mathbf{E}^L_t = \mathbf{f}_c \Rightarrow -F^t[\hat{A}^L_{ee,cc}\mathbf{E}^L_c - \hat{A}^L_{ee,cc}F\mathbf{E}^L_t] = -F^t\mathbf{f}_c.$$

Thus, only the first equation needs to be satisfied. Since $\hat{A}^L_{ee,cc}$ is positive-definite, a particular solution of the first equation is

$$\begin{aligned}\begin{pmatrix} \mathbf{E}^L_c \\ \mathbf{E}^L_t \end{pmatrix} &= \begin{pmatrix} \mathbf{E}^L_c \\ \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} \hat{A}^{L,-1}_{ee,cc}\mathbf{f}_c \\ \mathbf{0} \end{pmatrix},\end{aligned} \tag{5.16}$$

the padded extension of the cotree solution. Hence, (5.16) is an approximate solution to

$$\begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \\ A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix} \begin{pmatrix} \mathbf{E}^L_c \\ \mathbf{E}^L_t \end{pmatrix} = \begin{pmatrix} \mathbf{f}_c \\ -F^t\mathbf{f}_c, \end{pmatrix} \tag{5.17}$$

and therefore, must be approximately divergence-free.

We would like to know if the padded cotree corrections from a node/edge multiple-coarsening scheme can suitably represent smooth divergence-free frequencies of A^L_{ee}. So consider an error component that contains a smooth divergence-free frequency $\mathbf{e}_d$,

$$\mathbf{e} = \mathbf{e}_d + \nabla p.$$

The error equation to be solved is

$$A^L_{ee}\mathbf{e} = \mathbf{r}.$$

Since we can assume without loss of generality that the initial approximation is $\mathbf{0}$, the first step in the multiple-coarsening scheme is to solve the nodal equation

$$A^L_{nn}p = [G^L_{en}]^t A^L_{ee} G^L_{en} p = [G^L_{en}]^t \mathbf{r}.$$

Assuming that this is solved exactly, the fine edge correction equation for $\mathbf{e}_d$ is

$$A^L_{ee}\mathbf{e}_d = \hat{\mathbf{r}} = \mathbf{r} - A^L_{ee}G^L_{en}p,$$

which has a discretely divergence-free righthand side:

$$\begin{aligned}[G^L_{en}]^t\hat{\mathbf{r}} &= [G^L_{en}]^t \left[\mathbf{r} - A^L_{ee}G^L_{en}p\right] \\ &= [G^L_{en}]^t\mathbf{r} - [G^L_{en}]^t A^L_{ee}G^L_{en}p \\ &= \mathbf{0}.\end{aligned}$$

Thus, from the above discussion, $\begin{pmatrix} \hat{A}^{L,-1}_{ee,cc}\hat{\mathbf{r}}_c \\ \mathbf{0} \end{pmatrix}$ is a good approximation to $\mathbf{e}_d$, and hence, smooth divergence-free frequencies can be suitably represented by padded cotree corrections. This suggests that an effective solver for regularized magnetostatics is a multiple-coarsening scheme with a branch for the nodal Poisson operator and a branch $\hat{A}^L_{ee,cc}$. Note that now the cotree solution of $\hat{A}^L_{ee,cc}\mathbf{E}_c = \mathbf{f}_c$ is used directly to correct $\mathbf{E}^L$ since $\begin{pmatrix} \hat{A}^{L,-1}_{ee,cc}\mathbf{f}_c \\ \mathbf{0} \end{pmatrix}$ is an approximate solution to (5.17), not $\begin{pmatrix} A^{L,-1}_{ee,cc}\mathbf{f}_c \\ \mathbf{0} \end{pmatrix}$. To insure that the righthand sides of the edge correction equations are approximately divergence-free, a sufficient number of multigrid cycles must be applied to the nodal Poisson problem. The full algorithm is

Algorithm for Regularized Magnetostatics:

1. Fine grid edge relaxation on $A^L_{ee}\mathbf{E}^L = \mathbf{f}^L$
2. Residual computation: $\mathbf{r}^L = \mathbf{f}^L - A^L_{ee}\mathbf{E}^L$

3. Nodal problem: sufficient number of multigrid cycles on $A^L_{nn} p^L = G^t_{en} \mathbf{r}^L$
4. Nodal correction: $\mathbf{E}^L \leftarrow \mathbf{E}^L + G_{en} p^L$
5. Cotree righthand side:

$$\mathbf{r}_c = \left\{ \mathbf{f}_c - \left[\begin{array}{cc} A^L_{ee,cc} & A^L_{ee,ct} \end{array} \right] \left(\begin{array}{c} \mathbf{E}_c \\ \mathbf{E}_t \end{array} \right) \right\}$$

6. Cotree problem: solve

$$\hat{A}^L_{ee,cc} \mathbf{E}_c = \mathbf{r}_c$$

   using multigrid cycles
7. Edge correction:

$$\mathbf{E}^L \leftarrow \mathbf{E}^L + \left(\begin{array}{c} \mathbf{E}_c \\ \mathbf{0} \end{array} \right).$$

5.2. Special Case: Structured 2-d with Structured Tree/Cotree Extractions. Another special case that does not require the preconditioned Krylov iteration is structured 2-d problems with structured tree/cotree extractions. In this case, the smooth divergence-free frequencies of the whole edge matrix can be decoupled into cotree and tree components.

The problem of interest is equation (2.3) discretized on structured quadralaterals with no special restrictions on coefficients α and β. To get some intuition on this algorithm though, let α and β be constant. The smooth divergence-free frequencies of the continuous operator are eigenvectors corresponding to the small eigenvalues of

$$[-\alpha \underline{\Delta} + \beta I] = \left[\begin{array}{cc} -\alpha\Delta + \beta & 0 \\ 0 & -\alpha\Delta + \beta \end{array} \right].$$

The x, y-components of these eigenvectors decouple. Thus, if the tree/cotree extraction permits a similar component decoupling of $\mathbf{E}$, then these smooth divergence-free functions can be resolvable on decoupled tree and cotree equations.

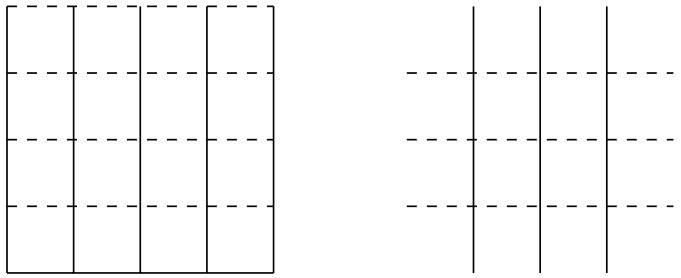

FIG. 5.1. *Left diagram: structured tree/cotree extraction on the whole grid with solid and dashed lines respectively denoting the tree and cotree edges. Right diagram: structured tree/cotree extraction with the boundary edges removed.*

Consider the tree/cotree extractions of the same underlying graph depicted in Figures 5.1 and 5.2. Since the boundary values are known, only the interior edge values need to be determined. For Figure 5.1, the unknown cotree values lie on the horizontal edges, which correspond to the x-component of $\mathbf{E}$ (because for structured quadralaterals, horizontal edges correspond to the x-components of $\mathbf{E}$) and the unknown tree values lie on the vertical edges, which correspond to the y-component of $\mathbf{E}$. In terms of $A^L_{ee,cc}$ and $A^L_{ee,tt}$, the stencil connections are "structured", i.e., at each horizontal/vertical edge, the same 3-point stencil pattern describes the horizontal/vertical connections. For Figure 5.2, even though the full edge matrix has a

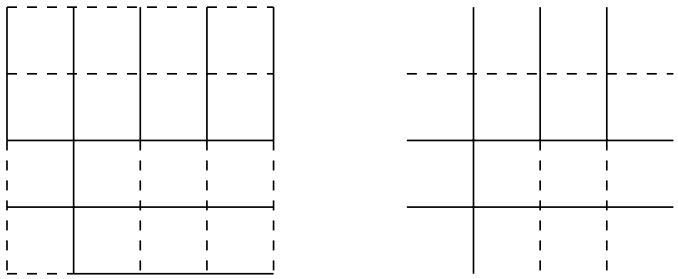

FIG. 5.2. *Left diagram: unstructured tree/cotree extraction on the whole grid with solid and dashed lines respectively denoting the tree and cotree edges. Right diagram: unstructured tree/cotree extraction with the boundary edges removed.*

structured stencil pattern (see stencils (5.18-5.19)), $A^L_{ee,cc}$ and $A^L_{ee,tt}$ have unstructured stencil patterns. For the structured extraction, the components of $\mathbf{E}$ decouple. Thus, the continuous smooth divergence-free functions can be approximated through decoupled tree and cotree problems. A reasonable algorithm is then

Algorithm for Structured Extractions:

1. Fine grid edge relaxation on $A^L_{ee}\mathbf{E}^L = \mathbf{f}^L$
2. Residual computation: $\mathbf{r}^L = \mathbf{f}^L - A^L_{ee}\mathbf{E}^L$
3. Nodal problem: multigrid cycles on $A^L_{nn}p^L = G^t_{en}\mathbf{r}^L$
4. Nodal correction: $\mathbf{E}^L \leftarrow \mathbf{E}^L + G_{en}p^L$
5. Cotree righthand side:

$$\mathbf{r}_c = \left\{\mathbf{f}_c - \begin{bmatrix} A^L_{ee,cc} & A^L_{ee,ct} \end{bmatrix} \begin{pmatrix} \mathbf{E}_c \\ \mathbf{E}_t \end{pmatrix}\right\}$$

6. Cotree problem: multigrid cycles on $A^L_{ee,cc}\mathbf{E}_c = \mathbf{r}_c$
7. Cotree edge correction:

$$\mathbf{E}^L \leftarrow \mathbf{E}^L + \begin{pmatrix} \mathbf{E}_c \\ \mathbf{0} \end{pmatrix}$$

8. Tree righthand side:

$$\mathbf{r}_t = \left\{\mathbf{f}_t - \begin{bmatrix} A^L_{ee,tc} & A^L_{ee,tt} \end{bmatrix} \begin{pmatrix} \mathbf{E}_c \\ \mathbf{E}_t \end{pmatrix}\right\}$$

9. Tree problem: multigrid cycles on $A^L_{ee,tt}\mathbf{E}_t = \mathbf{r}_t$
10. Tree edge correction:

$$\mathbf{E}^L \leftarrow \mathbf{E}^L + \begin{pmatrix} \mathbf{0} \\ \mathbf{E}_t \end{pmatrix}.$$

Note that because the horizontal and vertical stencil connections for the target edge equations are

$$\begin{bmatrix} & \left(-\frac{\alpha}{h^2} + \frac{\beta}{6}\right) & \\ -\frac{\alpha}{h^2} & & \frac{\alpha}{h^2} \\ & \left(\frac{2\alpha}{h^2} + \frac{2\beta}{3}\right) & \\ \frac{\alpha}{h^2} & & -\frac{\alpha}{h^2} \\ & \left(-\frac{\alpha}{h^2} + \frac{\beta}{6}\right) & \end{bmatrix} \tag{5.18}$$

and

$$\begin{bmatrix} & -\frac{\alpha}{h^2} & & \frac{\alpha}{h^2} & \\ \left(-\frac{\alpha}{h^2}+\frac{\beta}{6}\right) & & \left(\frac{2\alpha}{h^2}+\frac{2\beta}{3}\right) & & \left(-\frac{\alpha}{h^2}+\frac{\beta}{6}\right) \\ & \frac{\alpha}{h^2} & & -\frac{\alpha}{h^2} & \end{bmatrix}, \tag{5.19}$$

the stencils for $A^L_{ee,cc}$ and $A^L_{ee,tt}$ are respectively

$$\begin{bmatrix} \left(-\frac{\alpha}{h^2}+\frac{\beta}{6}\right) \\ \left(\frac{2\alpha}{h^2}+\frac{2\beta}{3}\right) \\ \left(-\frac{\alpha}{h^2}+\frac{\beta}{6}\right) \end{bmatrix}$$

and

$$\begin{bmatrix} \left(-\frac{\alpha}{h^2}+\frac{\beta}{6}\right) & \left(\frac{2\alpha}{h^2}+\frac{2\beta}{3}\right) & \left(-\frac{\alpha}{h^2}+\frac{\beta}{6}\right) \end{bmatrix}.$$

These latter stencils correspond to diffusion operators. Thus, a standard Ruge-Stuben coarsening procedure can be applied to $A^L_{ee,cc}$ and $A^L_{ee,tt}$.

6. Numerical Results. The methods described in Section 5 have been tested on a number of problems. We list some of the details of these experiments:

Since convergence is the focus of these experiments, all the problems consist of an edge element discretization of (2.1) on a unit square, with a random initial guess and zero righthand side. Except for the method of Subsection 5.2, the tree/cotree extractions were generated randomly by choosing a random seed tree edge in the FIFO algorithm. For the method of Subsection 5.2, the seed was chosen so that structured tree/cotree extractions were generated. For all experiments, one sweep of the overlapping Schwarz relaxation described in [2] was performed on the target fine level. As for multigrid solvers, a standard Ruge-Stuben AMG method was used on the nodal Poisson problems (A^L_{nn}) and the discrete Poisson problems ($\hat{G}^t_{en}\hat{G}_{en}$), as well as on the cotree and tree problems for the method of Subsection 5.2 (see the discussion at the end of that subsection). Observe that applying a Ruge-Stuben AMG algorithm to the discrete Poisson problem would explicitly require the matrix $\hat{G}^t_{en}\hat{G}_{en}$. This matrix can be formed without a sparse matrix-matrix product: at node i, the centre coefficient is equal to the sum of the edges emanating from node i and the off-diagonal coefficients, which correspond to the connecting coefficients of node i to the nodes at the other ends of the emananting edges, are each -1. Note that $\hat{G}^t_{en}\hat{G}_{en}$ is sparser than A^L_{nn}, for example, for a structured grid, $\hat{G}^t_{en}\hat{G}_{en}$ corresponds to 5 and 7-point stencils in 2 and 3-d, whereas A^L_{nn} corresponds to 9 and 27-point stencils. Thus, a multigrid cycle for this constant coefficient Poisson operator is noticably cheaper than a cycle for the material coefficient operator A^L_{nn}. As for multigrid solver for the cotree problems of the unstructured tree/cotree schemes, the algebraic multilevel multigraph algorithm of Bank et. al. ([3] and [4]) was employed. V(1,0) cycles with an ILU smoother were used in this code. Although this multilevel method is not optimally scalable, its performance is exceptionally good. The development of a scalable solver for $\hat{A}^L_{ee,cc}$ is the topic of a future paper. Finally, the actions of $F = \hat{G}_{en}G^{-1}_{en,t}$ and F^t are obtained through a sparse direct solve and a matrix-vector product. As noted earlier, these can be avoided by forming F using a graph independent cutset algorithm.

Convergence is considered to be attained when the initial residual is reduced by 8 orders of magnitude. The convergence rate of the overall iteration is taken to be

$$\max_i \frac{\|\mathbf{r}^i\|}{\|\mathbf{r}^{i-1}\|}$$

for iteration i.

Preconditioner (5.12): We consider results for the nodal/cotree multiple-coarsening cycle with preconditioner (5.12). Table 6.1 summarizes results for the constant coefficient case. Here, $\alpha = 1$ and β is varied. The 4-number column data registers the total number of overall multiple-coarsening cycles, the total number of inner preconditioned GMRES iterations, the total number of multigraph iterations for the total run, and the convergence rate. For example, for $\beta = 10^{-2}$ and $h = 1/128$, the data 2/3/7/2.1e-5 indicates 2 outer multiple-coarsening cycles, a total of 3 preconditioned GMRES for solving the 2 cotree problems (1 per multiple-coarsening cycle), a total of 7 multigraph V(1,0) cycles for all the preconditioning steps, and an outer iteration convergence rate of 2.1e-5. Here, since 5 V(1,1) cycles were performed for each nodal Poisson solve and discrete Poisson solve, a total of 10 V(1,1) cycles were used on the nodal branch and a total of 30 V(1,1) cycles were used on the sparser, discrete Poisson problems. Also, for the rates attained, a simple calculation reveals that initial residual reductions of more than 8 orders of magnitude are obtained for these experiments. However, the data also shows a performance degradation in the multigraph algorithm, and an increase in the number of GMRES iterations as β increases. The latter is expected since the equivalence between A_{ee}^L and $\hat{A}_{ee}^L$ degrades as β increases.

β/h	1/32	1/64	1/128	1/256
10^{-4}	2/2/2/2.0e-6	2/2/2/1.7e-6	2/2/4/2.1e-6	2/2/12/1.7e-5
10^{-3}	2/2/2/1.9e-5	2/2/2/1.8e-5	2/2/6/1.8e-5	2/2/10/1.8e-5
10^{-2}	2/3/3/1.4e-5	2/3/3/1.9e-5	2/3/7/2.1e-5	2/3/21/2.0e-5
10^{-1}	2/4/4/1.6e-6	2/4/4/1.5e-6	2/4/12/1.3e-6	2/4/31/1.3e-6
10^{0}	2/6/6/5.4e-6	2/6/6/1.5e-6	2/6/24/9.7e-7	2/6/38/6.4e-7
10^{1}	2/9/9/6.2e-5	2/9/9/1.8e-5	2/9/9/3.2e-5	2/9/60/1.5e-5
10^{2}	3/25/25/5.1e-4	3/25/25/1.5e-4	3/24/48/4.6e-5	3/24/170/7.0e-4

TABLE 6.1

Algorithm "Nodal/Cotree Multiple-Coarsening Cycle" using preconditioner (5.12)- random tree/cotree extraction, one finest level multiplicative Schwarz sweep, data= (number of outer cycles)/(number of GMRES iterations)/(total number of multigraph amg cycles)/(convergence rate).

We also consider a variable coefficient problem. As the equivalence between A_{ee}^L and $\hat{A}_{ee}^L$ depends on the coefficients α and β, dramatic jump differences between these coefficients were not permitted. However, the random coefficients

$$\alpha = (1 + |\sin \delta(x,y)|)(4 + \cos 20\pi y)^2(2 + \sin 5\pi x)(20 + \sin 5\pi x)$$
$$\beta = (1 + |\sin \delta(x,y)|)(4 + \sin 20\pi y)^2(2 + \cos 5\pi x)(20 + \cos 5\pi x),$$

where $\delta(x,y)$ is a random integer between 0 and the rand_max variable in C, were chosen (see Figure 6.1). Table 6.2 tabulates the results. As can be observed, the convergence of these runs have the same general behaviour as the constant coefficient case.

Problem/h	1/32	1/64	1/128	1/256
1a	4/12/12/1.9e-2	4/10/11/1.8e-2	3/8/47/1.8e-2	3/8/69/1.8e-2

TABLE 6.2

Problem 1a, Algorithm "Nodal/Cotree Multiple-Coarsening Cycle" using preconditioner (5.12)- random tree/cotree extraction, one finest level multiplicative Schwarz sweep, data= (number of outer cycles)/(number of GMRES iterations)/(total number of multigraph amg cycles)/(convergence rate).

Preconditioner 5.13: Next, we consider results for the nodal/cotree multiple-

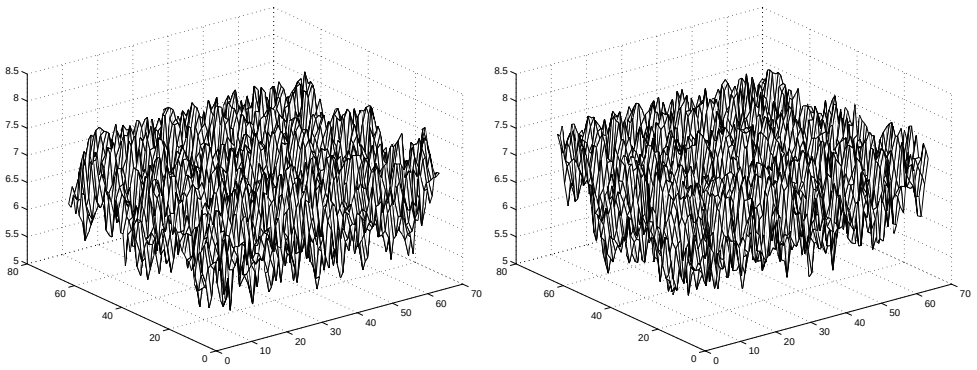

FIG. 6.1. *Log scale of jumps in α and β for Problems 1a & 2a.*

coarsening cycle with preconditioner (5.13) applied to the same problems. Tables 6.3 and 6.4 contain the results. Clearly, the only difference between the results of preconditioners (5.13) and (5.12) is a slight increase in the total number of multigraph V(1,0) cycles.

β/h	1/32	1/64	1/128	1/256
10^{-4}	2/2/2/6.4e-7	2/2/2/1.9e-6	2/2/6/7.6e-7	2/2/10/2.6e-6
10^{-3}	2/2/2/6.6e-6	2/2/2/7.0e-6	2/2/4/6.6e-6	2/2/15/1.8e-5
10^{-2}	2/4/4/7.3e-7	2/4/4/1.0e-6	2/4/12/7.9e-7	2/4/30/2.8e-7
10^{-1}	2/4/4/7.1e-6	2/4/4/3.9e-6	2/4/16/2.0e-6	2/4/26/1.9e-6
10^{0}	2/6/6/6.2e-5	2/6/6/2.3e-5	2/6/12/6.6e-6	2/6/45/6.3e-6
10^{1}	3/14/14/1.0e-3	3/14/14/2.0e-4	2/9/36/1.0e-4	2/9/54/2.0e-5
10^{2}	4/33/33/8.0e-3	3/24/24/2.0e-3	3/24/97/7.0e-4	2/16/125/5.5e-4

TABLE 6.3

Algorithm "Nodal/Cotree Multiple-Coarsening Cycle" using preconditioner (5.13)- random tree/cotree extraction, one finest level multiplicative Schwarz sweep, data= (number of outer cycles)/(number of GMRES iterations)/(total number of multigraph amg cycles)/(convergence rate).

Problem/h	1/32	1/64	1/128	1/256
2a	5/15/15/3.0e-2	5/15/15/2.6e-2	4/12/70/2.2e-2	4/12/86/2.5e-2

TABLE 6.4

Problem 2a, Algorithm "Nodal/Cotree Multiple-Coarsening Cycle" using preconditioner (5.13)- random tree/cotree extraction, one finest level multiplicative Schwarz sweep, data= (number of outer cycles)/(number of GMRES iterations)/(total number of multigraph amg cycles)/(convergence rate).

Regularized Magnetostatics: The scheme for regularized magnetostatics is substantially cheaper than the preconditioned methods. No Krylov wrapper is needed, and neither $(I + FF^t)$ nor F even arises in this method. However, this method is applicable only to small regularization terms, as is the case for practical magnetostatics problems. Table 6.5 shows the results for constant coefficient problems. The 3-number column data registers the number of outer cycles, the total number of multigraph V(1,0) cycles, and the convergence rate. As expected, when $\hat{\epsilon} \to 0$, the solver converges faster. For comparison with the preconditioned method, when $\beta = 10^{-2}$ and $h = 1/128$, since the regularized magnetostatics scheme requires only 2 outer iterations, a total of 10 nodal V(1,1) cycles are needed, rather than 40 (10 for the A^L_{nn} and 30 for $\hat{G}^t_{en}\hat{G}_{en}$).

We also consider two jump coefficient problems. Only α varies since $\hat{\epsilon}$ is a user tuning parameter. The profiles of α are displayed in Figure 6.2. For these coefficients,

β/h	1/32	1/64	1/128	1/256
10^{-6}	1/1/2.0e-9	1/1/1.0e-9	1/4/3.0e-10	1/8/2.0e-10
10^{-4}	2/2/1.0e-4	2/2/1.0e-4	2/8/5.0e-4	2/16/4.0e-4
10^{-2}	3/3/1.0e-3	2/2/5.0e-4	2/5/7.0e-4	2/17/8.0e-4
10^{0}	7/7/1.0e-1	6/6/1.0e-1	6/18/1.0e-1	5/39/1.0e-1

TABLE 6.5

Algorithm for regularized magnetostatics-random tree/cotree extraction, one finest level multiplicative Schwarz sweep, 5 V(1,1) cycles for the nodal branch, data= (number of outer cycles)/(total number of multigraph amg cycles)/(convergence rate).

$\hat{\epsilon}$ was chosen to be $\leq \min \alpha$. Table 6.6 summarizes the results. The performance is even better than when the coefficients are constant.

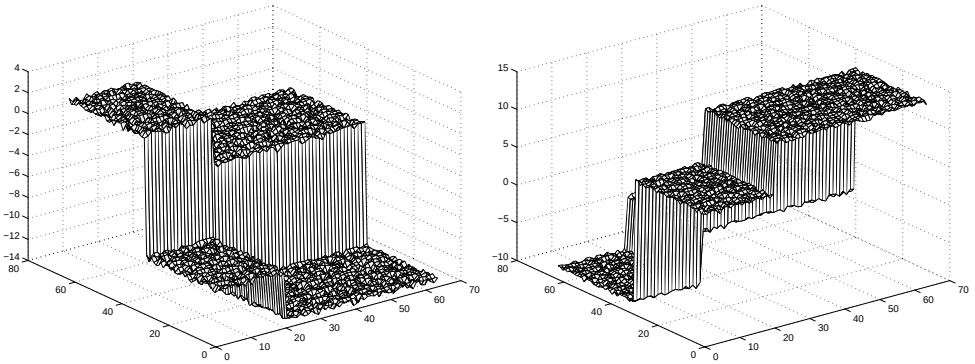

FIG. 6.2. *Log scale of jumps in α for Problems 3a (left) and 3b (right).*

Problem/h	1/32	1/64	1/128	1/256
3a	1/4/4.6e-10	1/5/5.0e-9	1/9/7.8e-9	1/4/1.9e-9
3b	1/5/1.6e-10	1/1/9.0e-10	1/2/3.0e-11	1/6/9.0e-7

TABLE 6.6

Problems 3a & 3b, regularized magnetostatics-random tree/cotree extraction, one finest level multiplicative Schwarz sweep, 5 V(1,1) cycles for the nodal branch, data= (number of outer cycles)/(total number of multigraph amg cycles)/(convergence rate). $\hat{\epsilon}$ =1.0e-6 for Problem 3a and $\hat{\epsilon}$ =1.0e-4 for Problem 3b.

Structured Tree/Cotree Extraction: Our final set of experiments are for structured 2-d problems with structured tree/cotree extractions. As already stated, a standard Ruge-Stuben AMG method is applied to the tree and cotree multigrid hierarchies. Table 6.7 contain results for constant coefficient problems. Here, only 1 V(1,1) cycle is used for each branch. Clearly, we observe scalability for this special method.

Jump coefficients also were considered. Figures 6.3-6.5 illustrate the coefficients. The coefficients in Figure 6.5 is particularly interesting since the magnitude jumps in α and β are in the opposite direction. In the continuum, this leads to characteristically different singularities in the curl-free and divergence-free components of the error. Table 6.8 shows the results. For the first two sets of coefficients (Figures 6.3 and 6.4), the results are even better than for the constant coefficient case. This may be due to the 4 additional V(1,1) cycles applied to the nodal Poisson problem. These additional cycles were added to handle the jumps in β. Only 1 V(1,1) cycle was applied to each tree and cotree branch. However, for the coefficients of Figure 6.5, a Restart(10) GMRES wrapper around the cotree/tree branches was needed to give

β/h	1/32	1/64	1/128	1/256
10^{-4}	7/0.10	7/0.10	7/0.10	7/0.11
10^{-3}	7/0.10	7/0.10	7/0.10	7/0.11
10^{-2}	7/0.10	7/0.11	7/0.11	7/0.11
10^{-1}	7/0.10	7/0.11	7/0.11	7/0.11
10^{0}	7/0.10	7/0.11	7/0.11	7/0.11
10^{1}	7/0.12	7/0.11	7/0.11	7/0.11
10^{2}	7/0.10	7/0.13	7/0.12	7/0.12

TABLE 6.7

Structured 2-d problem with structured tree/cotree extraction. One finest level overlapping Schwarz relaxation and V(1,1) cycle for each branch, nodal, cotree, and tree. Iteration counts and convergence rates are for the overall outer iteration.

good performance. For a total of n outer multiple-coarsening cycles, a total of $10n$ GMRES iterations were needed, giving a total of $10n$ tree and cotree V(1,1) cycles. Although the convergence rates are not too good, only a few outer cycles were needed to attain the 8-order magnitude drop of the initial residual. Figure 6.6 shows the slowly converging error modes.

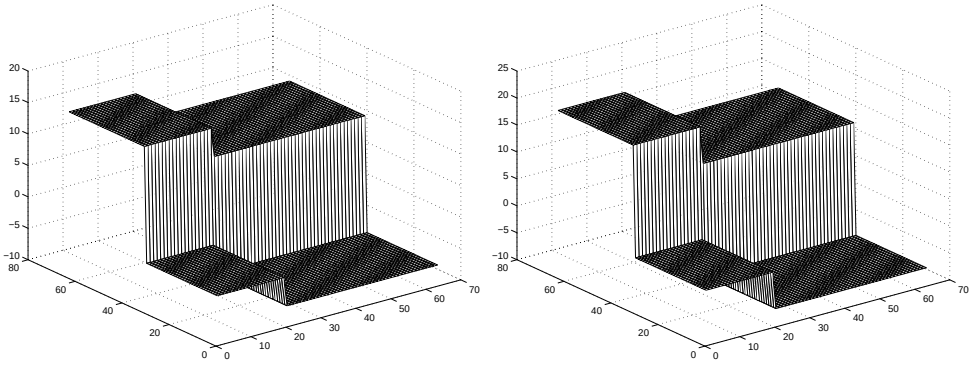

FIG. 6.3. *Log scale of jumps in α and β for Problem 4a.*

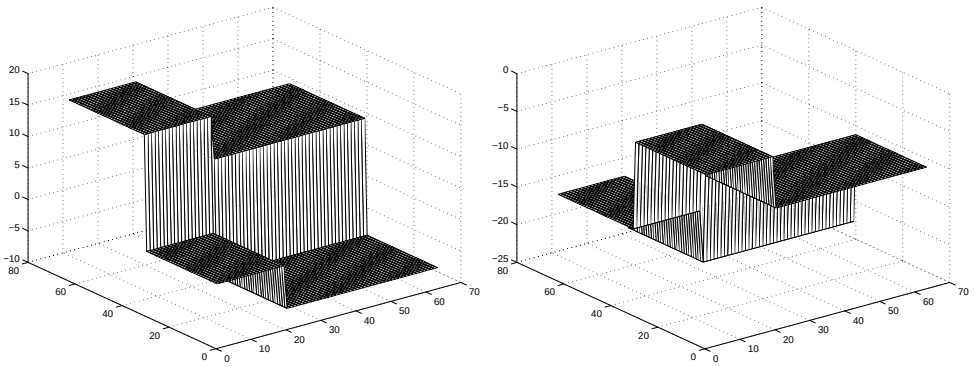

FIG. 6.4. *Log scale of jumps in α and β for Problem 4a.*

Acknowledgement
We like to thank Professor R. Bank for providing us with his algebraic multilevel multigraph code.

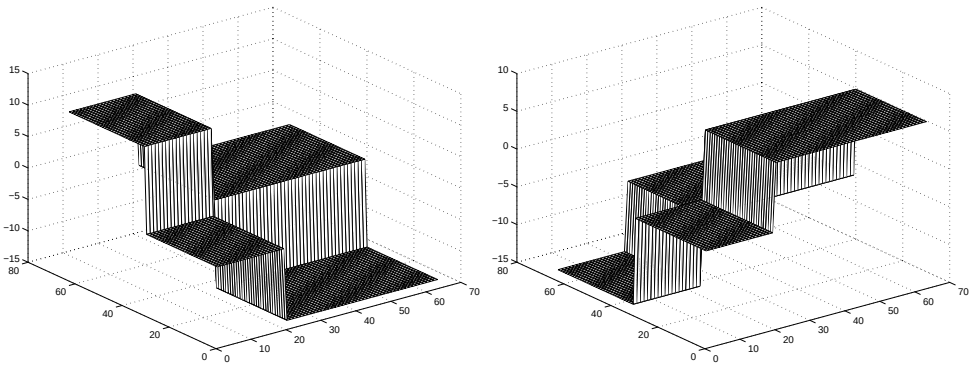

FIG. 6.5. *Log scale of jumps in α and β for Problem 4a.*

Problem/h	1/32	1/64	1/128	1/256
4a	4/3.0e-2	4/2.0e-2	4/1.0e-2	4/8.0e-3
4b	3/1.0e-3	3/1.0e-3	3/1.0e-3	3/4.0e-3
4c	5/4.3e-1	4/4.0e-1	3/2.2e-1	2/6.0e-2

TABLE 6.8

Structured 2-d problem with structured tree/cotree extraction. One finest level overlapping Schwarz relaxation and 5 V(1,1) cycles for the nodal branch. For Problems 4a & 4b, 1 V(1,1) cycle is applied to the cotree and tree branches. For Problem 4c, an outer GMRES wrapper is put around the edge equations. 10 GMRES iterations are taken, with the preconditioner being 1 V(1,1) cycle each for the cotree and tree branches.

REFERENCES

[1] R. ALBANESE AND G. RUBINACCI, *Integral formulation for 3D eddy-current computation using edge elements*, IEE Proceedings A, 135 (1988), pp. 457-462.

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

[3] R. E. BANK AND T. F. CHAN, *An analysis of the composite step biconjugate gradient method*, Num. Math., 66 (1993), 295-319.

[4] R. E. BANK AND R. K. SMITH, *An algebraic multilevel multigraph algorithm*, SIAM J. Sci. Comp., 23 (2002), 1572-1592.

[5] R. BECK, *Algebraic multigrid by component splitting for edge elements on simplicial triangulations*, Preprint SC 99-40, Konrad-Zuse-Zentrum fur Informationstechnic Berlin, 1999.

[6] 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.

[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] T. H. CORMEN, C. E. LEISERSON, AND R. L. RIVEST, *Introduction to Algorithms*, MIT Press, 1997.

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

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

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

[14] F. Dubois, *Discrete vector potential representation of a divergence-free vector field in three-dimensional domains: numerical analysis of a model problem*, SIAM J. Numer. Anal., 27 (1990), pp. 1103-1141.

[15] N. A. GOLIAS AND T. D. TSIBOUKIS, *Magnetostatics with edge elements: a numerical investigation in the choice of the tree*, IEEE Trans. Magn., 30 (1994), pp. 2877-2880.

[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 AND R. W. HOPPE, *Multilevel methods for mixed finite elements in three dimensions*, Numer. Math, 82 (1999), 253-279.

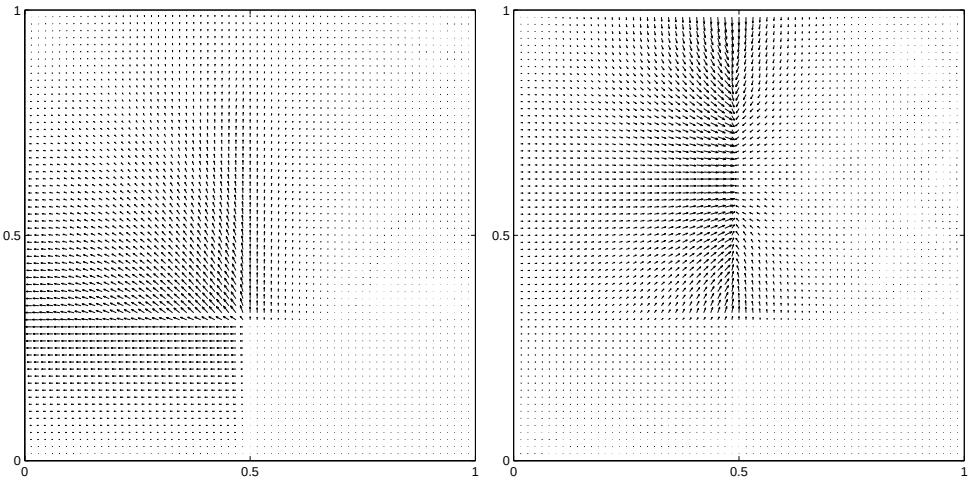

FIG. 6.6. *Slowly converging modes for Problem 4c: left diagram without GMRES, right diagram with GMRES.*

[19] J. JONES AND B. LEE, *A multigrid method for variable coefficient Maxwell's equations*, to appear in Siam J. Sci. Comp..

[20] J. B. MAGNES AND Z. J. CENDES, *A generalized tree-cotree gauge for magnetic field computation*, IEEE Trans. Magn., 31 (1995), pp. 1342-1347.

[21] I. MUNTEANU, *Tree-cotree condensation properties*, preprint, 2004.

[22] F. Rapetti, F. Dubois, and A. Bossavit, *Discrete vector potentials for nonsimply connected three-dimensional domains*, SIAM J. Numer. Anal., 41 (2003), pp. 1505-1527.

[23] S. REITZINGER AND J. SCHOBERL, *An algebraic multigrid method for finite element discretizations with edge elements*, Num. Lin. Alg. Appl., 9 (2002), 223-238.

[24] Z. Ren and A. Razek, *Boundary edge elements and spanning tree technique in three-dimensional electromagnetic field computation*, Int. J. Numer. Meth. Engng., 36 (1993), pp. 2877-2893.

[25] J. W. Ruge and K. Stuben, *Algebraic multigrid (AMG)*, in *Multigrid Methods*, S. F. McCormick, ed., SIAM Frontiers in Applied Mathematics, Vol. 5, Philadelphia, 1986.

[26] R. Scheichl, *Decoupling three-dimensional mixed problems using divergence-free finite elements*, SIAM J. Sci. Comput., 23 (2002), pp. 1752-1776.

[27] U. Trottenberg, C. W. Oosterlee and A. Schuller, *Multigrid*, Academic Press, 2001.

[28] J. P. WEBB, *Edge elements and what they can do for you*, IEEE Trans. Magn., 29 (1993), pp. 1460-1465.