

AN ALGEBRAIC MULTIGRID APPROACH BASED ON A COMPATIBLE GAUGE REFORMULATION OF MAXWELL'S EQUATIONS *

PAVEL B. BOCHEV[†], JONATHAN J. HU[‡], CHRISTOPHER M. SIEFERT[†], AND
RAYMOND S. TUMINARO[‡]

Abstract. With the rise in popularity of compatible finite element, finite difference and finite volume discretizations for the time domain eddy current equations, there has been a corresponding need for fast solvers of the resulting linear algebraic systems. However, the traits that make compatible discretizations a preferred choice for the Maxwell's equations also render these linear systems essentially intractable by truly black-box techniques. We propose a new algebraic reformulation of the discrete eddy current equations along with a new algebraic multigrid technique (AMG) for this reformulated problem. The reformulation process takes advantage of a discrete Hodge decomposition on co-chains to replace the discrete eddy current equations by an equivalent 2×2 block linear system whose diagonal blocks are discrete Hodge Laplace operators acting on 1-cochains and 0-cochains, respectively. While this new AMG technique requires somewhat specialized treatment on the finest mesh, the coarser meshes can be handled using standard methods for Laplace-type problems. Our new AMG method is applicable to a wide range of compatible methods on structured and unstructured grids, including edge finite elements, mimetic finite differences, co-volume methods and Yee-like schemes. We illustrate the new technique, using edge elements in the context of smoothed aggregation AMG, and present computational results for problems in both two and three dimensions.

Key words. Maxwell's equations, eddy currents, algebraic multigrid, multigrid, discrete Hodge Laplacian, compatible discretizations, edge elements.

AMS subject classifications. 65F10, 65F30, 78A30

1. Introduction. Due to the pioneering work of Bossavit [9], it is now well-known that stable and accurate numerical solution of Maxwell's equations can be achieved by using discrete spaces from a finite-dimensional analogue of the differential De Rham complex. Numerical methods that are based on such discretizations are now commonly referred to as *compatible*, or *mimetic* methods [6].

However, the same traits that make compatible methods a natural choice for Maxwell's equations also render the ensuing linear system essentially intractable by truly general purpose black-box multigrid solvers. For example, a compatible discretization of the curl-curl operator gives rise to a symmetric, semi-definite linear system whose null-space is of approximately the same size as the number of nodes in the computational grid. Multilevel solution of such systems often utilizes special smoothing, prolongation and restriction operators that separate error components in the null-space and its complement and satisfy a commuting diagram property. Formulation of such operators is well-understood in geometric multigrid settings [22], where

*Received by the editors Month ??, 2007; accepted for publication (in revised form) Month ??, 2007; published electronically Month ??, 2007. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.
<http://www.siam.org/journals/sisc/????/?????.html>

[†]Sandia National Laboratories, Computational Math/Algorithms, P.O. Box 5800, MS 1320, Albuquerque, NM 87185-1320 (pbboche@sandia.gov, csiefer@sandia.gov).

[‡]Sandia National Laboratories, Computational Math/Algorithms, P.O. Box 969, MS 9159, Livermore, CA 94551-0969 (jhu@sandia.gov, rstumin@sandia.gov).

availability of nested grids greatly simplifies the characterization of the null-space at all grid levels. In the algebraic multigrid setting, however, a proper characterization of the null-space at the coarser levels is not immediately obvious. As a result, algebraic multigrid (AMG) for Maxwell’s equations, that is comparable with AMG methods for the Poisson equation, has been elusive.

In this paper we propose a new algebraic reformulation of the discrete eddy current equations along with a new AMG technique for this reformulated problem. The reformulation process takes advantage of a discrete Hodge decomposition on co-chains to replace the discrete eddy current equations by an equivalent 2×2 block linear system whose diagonal blocks are discrete Hodge Laplace operators acting on 1-cochains and 0-cochains, respectively. While this new AMG technique requires somewhat specialized treatment on the fine mesh, the coarser meshes can be handled using standard methods for Laplace-type problems. An attractive computational feature of our new AMG method is its applicability to a wide range of compatible methods on structured and unstructured grids, including edge finite elements [9], mimetic finite differences [40], co-volume methods [33] and staggered grid (Yee) schemes [44].

There are two basic approaches to reformulation of Maxwell’s equations that differ in the order in which the discretization and reformulation steps are applied. The methods in [3, 10, 19, 20] are examples of the *reformulate and then discretize* approach in which the differential equations are first transformed to an equivalent form and then discretized. Typically, methods of this kind have been restricted to staggered grids because of the need to effectively discretize a vector Laplacian operator acting on fields that are only tangentially or normally continuous.

Our approach belongs to the category of *discretize and then reformulate* methods in which reformulation is applied after the discretization step. This approach offers greater flexibility in the choice of the discretization because it relies on a discrete Hodge decomposition to derive the reformulated equations. Therefore, it is applicable to any discretization setting that offers such a discrete decomposition, including mixed finite elements, mimetic finite differences and co-volume methods. Other examples of this approach can be found in [11, 16, 21], however, these papers are focused solely on the reformulation and do not address solvers. Of special note are the recent papers [23, 28, 29, 30] which uses the idea of *auxiliary space* preconditioners and nodal vector Laplacians. The approach in this paper bears some similarity to our proposed method; however, it is based on a discrete version of a non-orthogonal “regular decomposition” [34] of $H(\text{curl})$, as opposed to a discrete Hodge decomposition in our case.

Some reformulation techniques can also be viewed as gauging approaches, although in a different sense from standard gauges [10, 12, 13, 14]. Our proposed technique is rightfully called a compatible gauge, but it differs from standard gauges in three ways. First, our compatible gauge is applied after the discretization, rather than at the PDE level. Second, it applies to compatible discretizations rather than nodal discretizations. Third, the addition of our gauge does not change the discrete solution. Standard gauges are often applied to massage the problem into a form that can be discretized using nodal elements¹. Our gauge is applied to shorten the solu-

¹The use of nodal elements to discretize the vector Laplacian $(\nabla \times \nabla - \nabla \nabla \cdot) \mathbf{E}$ often overlooks the fact that the space \mathbf{H}^1 may have infinite co-dimension in $H(\text{div}) \cap H(\text{curl})$, i.e., a nodal discretization of this operator may fail to produce convergent approximations. On the other hand, on a single mesh, discretization of a field in $H(\text{div}) \cap H(\text{curl})$ must be simultaneously tangentially and normally continuous, i.e., it has to be of class C^0 . This explains why *reformulate and then discretize* approaches have been so far limited to staggered grids - such grids provide discretizations of $H(\text{div}) \cap H(\text{curl})$ that are not necessarily C^0 and can handle material discontinuities that lead to fields that are only

tion time for a compatibly discretized problem. Like our proposed method, algebraic reformulations can also be considered compatible gauges in this sense [11, 16, 21].

Previous work on AMG techniques for Maxwell's equations has focused on specialized techniques for the curl-curl operator. These methods usually require special aggregation [35], coarsening [32], interpolation [8] or smoothing techniques [2, 4, 22]. These techniques can be enhanced in a number of ways, such as by smoothed prolongation [5] and higher-order interpolation [5, 24] to yield effective solvers for fairly difficult problems. However, these techniques are specialized for curl-curl problems and are limited in terms of the benefit they can derive from advances in standard AMG techniques.

The paper is organized as follows. §2 introduces some notation and the eddy current Maxwell's equations. §3 reviews basic facts about the discretization framework used in the paper. In §4 we apply this framework to obtain a compatible discretization of the eddy current equations and its equivalent reformulation. AMG solvers for the reformulated system are developed in §5. In §6 we present computational results in two and three dimensions that illustrate the new technique in the context of smoothed aggregation AMG. In all experiments we use finite element discretizations based on the lowest order edge elements on both structured and unstructured triangular, quadrilateral and hexahedral elements.

2. The model equations. Let Ω denote a bounded, simply connected, contractible domain in \mathbb{R}^d , $d = 2, 3$ with Lipschitz continuous boundary $\partial\Omega$. We assume that $\partial\Omega$ consists of two disjoint parts Γ and Γ^* , i.e., $\partial\Omega = \Gamma \cup \Gamma^*$ and $\Gamma \cap \Gamma^* = \emptyset$. The eddy current equations in terms of the electric field \mathbf{E} are given by

$$\left\{ \begin{array}{l} \sigma \frac{\partial \mathbf{E}}{\partial t} + \nabla \times \frac{1}{\mu} \nabla \times \mathbf{E} = 0 \quad \text{in } \Omega \\ \mathbf{n} \times \mathbf{E} = 0 \quad \text{on } \Gamma \\ \mathbf{n} \times \frac{1}{\mu} \nabla \times \mathbf{E} = 0 \quad \text{on } \Gamma^* \end{array} \right. , \quad (2.1)$$

augmented with the initial condition

$$\mathbf{E}(\mathbf{x}, 0) = \mathbf{E}_0(\mathbf{x}) \quad \text{in } \Omega. \quad (2.2)$$

In (2.1) \mathbf{E} is the electric field, σ is the electrical conductivity and μ is the magnetic permeability. We assume that σ and μ are positive throughout the computational domain.

The boundary condition on Γ is called a tangential induction condition and represents a Dirichlet (essential) boundary condition. The boundary condition prescribed on Γ^* is the normal field condition; see [9], and it is of the Neumann type (natural boundary condition). Note that the boundary conditions in (2.1) imply that

$$\nabla \cdot \sigma \mathbf{E} = 0 \quad \text{on } \Gamma \quad \text{and} \quad \mathbf{n} \cdot \sigma \mathbf{E} = 0 \quad \text{on } \Gamma^*. \quad (2.3)$$

We assume that the initial data satisfies the compatibility condition $\nabla \cdot \sigma \mathbf{E}_0 = 0$. Then, from the first equation in (2.1) it is easy to see that the eddy current problem has the involution $\nabla \cdot \sigma \mathbf{E} = 0$ for all $t > 0$.

tangentially or normally continuous.

3. Compatible discretization framework. To approximate the eddy current equations (2.1) we use a general framework for compatible discretizations developed in [6]. This framework is based on algebraic topology and includes certain finite element [7, 41], finite volume [33], and finite difference [40] schemes as particular cases. As a result, the AMG algorithm developed in this paper is readily applicable to discrete problems generated by any of these schemes.

3.1. Computational grid. We will consider computational grids Ω^h consisting of 0-cells (nodes), 1-cells (edges), 2-cells (faces), and 3-cells (volumes). Formal linear combinations of k -cells are called k -chains [17]. The sets of k -chains forming Ω^h are denoted by C_k . We will assume that Ω^h is such that the collection $\{C_0, C_1, C_2, C_3\}$ is a complex, i.e., for any $c \in C_k$, $\partial_k c \in C_{k-1}$, where $\partial_k : C_k \mapsto C_{k-1}$ is the boundary operator on k -chains [15]. Together with the identity $\partial_k \partial_{k+1} = 0$ this gives rise to the exact sequence

$$0 \longleftarrow C_0 \xleftarrow{\partial_1} C_1 \xleftarrow{\partial_2} C_2 \xleftarrow{\partial_3} C_3 \longleftarrow 0. \quad (3.1)$$

The dual of C_k is denoted by C^k and its members are called k -cochains [17]. While C_k and C^k are isomorphic, they have different meanings in our discretization framework. The sets C_k represent the physical objects that form the grid, while C^k are collections of real numbers associated with the grid objects. For example, $c_1 \in C_1$ is a formal sum of (oriented) grid edges, while its isomorphic image $c^1 \in C^1$ is a set of real numbers² assigned to the edges of c_1 .

Therefore, the elements of C^0 provide values associated with the 0-cells (grid nodes); the elements of C^1 are values associated with the 1-cells (grid edges); C^2 contains values assigned to the 2-cells (grid faces) of the grid, and C^3 are the values assigned to the 3-cells (grid volumes). We will use C^0 and C^3 to approximate scalar functions and C^1 and C^2 - to approximate vector functions. In particular, the electric field \mathbf{E} in (2.1) will be approximated by 1-cochains.

The symbols C_Γ^k will denote the subspaces of C^k constrained by zero on the Dirichlet boundary Γ for $k = 0, 1, 2$. Such spaces are needed to approximate scalar and vector functions subject to appropriate boundary conditions³.

3.2. Natural operators. Let $\langle \cdot, \cdot \rangle$ denote the duality pairing of C_k and C^k . The adjoint of ∂_k , defined by $\langle a, \partial_k c \rangle = \langle \delta_k a, c \rangle$, induces an operator $\delta_k : C_\Gamma^k \mapsto C_\Gamma^{k+1}$ called coboundary. This operator satisfies $\delta_{k+1} \delta_k = 0$ and gives rise to the exact sequence

$$\mathbb{R} \longrightarrow C_\Gamma^0 \xrightarrow{\delta_0} C_\Gamma^1 \xrightarrow{\delta_1} C_\Gamma^2 \xrightarrow{\delta_2} C_\Gamma^3 \longrightarrow 0. \quad (3.2)$$

It is not hard to see that the matrix representation \mathbb{D}_k of δ_k is the signed incidence matrix between C_k and C_{k+1} . Following [26] we call \mathbb{D}_0 , \mathbb{D}_1 , and \mathbb{D}_2 *natural* approximations of the gradient, curl and divergence operators. Note that from $\delta_{k+1} \delta_k = 0$ it follows that

$$\mathbb{D}_{k+1} \mathbb{D}_k = 0; \quad k = 0, 1, 2, \quad (3.3)$$

²Clearly, C^k are isomorphic to $\mathbf{R}^{\tilde{k}}$, where $\tilde{k} = \dim C^k$. For simplicity, the isomorphic image of the cochain $c^k \in C^k$ in $\mathbf{R}^{\tilde{k}}$ will be denoted by the same symbol.

³For example, C_Γ^0 approximates scalar functions such that $\phi = 0$ on Γ ; C_Γ^1 can be used to approximate vector fields \mathbf{E} such that $\mathbf{n} \times \mathbf{E} = 0$ on Γ . The space C_Γ^2 is appropriate for vector fields \mathbf{B} that have a vanishing normal component on Γ .

and so our natural operators mimic the well-known vector calculus identities $\nabla \times \nabla = 0$, and $\nabla \cdot \nabla \times = 0$. In [25], it is pointed out that natural operations are not enough to provide compatible discretizations of the basic second order operators because their ranges and domains do not match. For example, we cannot approximate $\nabla \times \nabla \times$ by $\mathbb{D}_1 \mathbb{D}_1$ because \mathbb{D}_1 is in general a rectangular matrix. The number of its columns and rows equals the number of 1-cells and 2-cells in the grid, which are not the same.

3.3. Metric structures and derived operators. Let $\mathbb{M}_k : C_\Gamma^k \mapsto C_\Gamma^k$, $k = 0, 1, 2, 3$ denote symmetric positive definite matrices. The matrix \mathbb{M}_k endows C_Γ^k with an inner product structure,

$$(a^k, b^k)_{C^k} = (a^k)^T \mathbb{M}_k (b^k). \quad (3.4)$$

The matrices \mathbb{M}_0 and \mathbb{M}_3 approximate weighted L^2 inner products of scalar functions:

$$\mathbb{M}_0 \longrightarrow \int_\Omega \gamma p \hat{p} d\Omega; \quad \mathbb{M}_3 \longrightarrow \int_\Omega \lambda \phi \hat{\phi} d\Omega,$$

while \mathbb{M}_1 and \mathbb{M}_2 approximate the weighted L^2 inner products of vector functions

$$\mathbb{M}_1 \longrightarrow \int_\Omega \sigma \mathbf{E} \hat{\mathbf{E}} d\Omega; \quad \mathbb{M}_2 \longrightarrow \int_\Omega \mu^{-1} \mathbf{B} \hat{\mathbf{B}} d\Omega,$$

that are needed in the formulation of the discrete eddy current equations.

We define the derived operator $\mathbb{D}_k^* : C_\Gamma^{k+1} \mapsto C_\Gamma^k$ as the adjoint of \mathbb{D}_k with respect to the inner product (3.4):

$$(\mathbb{D}_k^* a^{k+1}, b^k)_{C^k} = (a^{k+1}, \mathbb{D}_k b^k)_{C^{k+1}}. \quad (3.5)$$

From (3.5) it is easy to see that for $k = 0, 1, 2$

$$\mathbb{D}_k^* = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1}. \quad (3.6)$$

The matrices \mathbb{D}_2^* , \mathbb{D}_1^* and \mathbb{D}_0^* provide a second set of discrete differential operators. Specifically, they are approximations of scaled gradient, curl and divergence operators

$$\mathbb{D}_2^* \rightarrow -\mu \nabla \lambda; \quad \mathbb{D}_1^* \rightarrow \sigma^{-1} \nabla \times \mu^{-1}; \quad \mathbb{D}_0^* \rightarrow -\gamma^{-1} \nabla \cdot \sigma,$$

augmented with the boundary conditions

$$\lambda \phi = 0; \quad \mathbf{n} \times \mu^{-1} \mathbf{B} = 0; \quad \text{and} \quad \mathbf{n} \cdot \sigma \mathbf{E} = 0 \quad \text{on} \quad \Gamma^*,$$

respectively. Using (3.6) and (3.3)

$$\mathbb{D}_k^* \mathbb{D}_{k+1}^* = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1} \mathbb{M}_{k+1}^{-1} \mathbb{D}_{k+1}^T \mathbb{M}_{k+2} = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{D}_{k+1}^T \mathbb{M}_{k+2} = 0,$$

and so, the basic vector calculus identities hold for the derived operators as well.

Because the range of \mathbb{D}_k is contained in the domain of \mathbb{D}_k^* and vice versa we can use the natural and the derived operators to define discrete versions of the basic second order differential operators, including a discrete Hodge Laplace operator. Specifically, we have the second order operators

$$\mathbb{D}_k^* \mathbb{D}_k : C_\Gamma^k \mapsto C_\Gamma^k; \quad \mathbb{D}_k^* \mathbb{D}_k = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1} \mathbb{D}_k; \quad k = 0, 1, 2 \quad (3.7)$$

$$\mathbb{D}_k \mathbb{D}_k^* : C_\Gamma^{k+1} \mapsto C_\Gamma^{k+1}; \quad \mathbb{D}_k \mathbb{D}_k^* = \mathbb{D}_k \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1}; \quad k = 1, 2, 3 \quad (3.8)$$

and the discrete Hodge Laplacian

$$\mathbb{L}_k : C_\Gamma^k \mapsto C_\Gamma^k; \quad \mathbb{L}_k = \mathbb{D}_k^* \mathbb{D}_k + \mathbb{D}_{k-1} \mathbb{D}_{k-1}^*; \quad k = 0, 1, 2, 3 \quad (3.9)$$

with the understanding that $\mathbb{D}_3 = 0$ and $\mathbb{D}_{-1}^* = 0$.

The discrete operators in (3.7)-(3.9) approximate basic second order elliptic differential operators. For example, $\mathbb{D}_1^* \mathbb{D}_1$ is a compatible discretization of $\sigma^{-1} \nabla \times \mu^{-1} \nabla \times \mathbf{E}$, augmented with the essential boundary conditions from (2.1). The operator $\mathbb{L}_1 = \mathbb{D}_1^* \mathbb{D}_1 + \mathbb{D}_0 \mathbb{D}_0^*$ is a compatible discretization of the Hodge Laplacian

$$\sigma^{-1} \nabla \times \mu^{-1} \nabla \times \mathbf{E} - \nabla \gamma^{-1} \nabla \cdot \sigma \mathbf{E}, \quad (3.10)$$

augmented with the boundary conditions

$$\mathbf{n} \times \mathbf{E} = 0 \quad \text{on } \Gamma \quad \text{and} \quad \mathbf{n} \cdot \sigma \mathbf{E} = 0 \quad \text{on } \Gamma^*. \quad (3.11)$$

In §4.1 we will use this operator to motivate and explain our reformulation strategy. A key ingredient in this strategy will be to endow C_Γ^1 with a second inner product defined by a matrix $\tilde{\mathbb{M}}_1$ that uses a unit weight, i.e.,

$$\tilde{\mathbb{M}}_1 \longrightarrow \int_\Omega \mathbf{E} \hat{\mathbf{E}} \, d\Omega.$$

The reasons to consider this inner product will be also explained in §4.1. There we will see that by using $\tilde{\mathbb{M}}_1$, instead of \mathbb{M}_1 , in the discrete Hodge decomposition of the electric field, the scaling of the reformulated system can be significantly improved for materials with widely varying conductivities σ .

The second inner product on C_Γ^1 gives rise to a second set of derived operators $\tilde{\mathbb{D}}_0^* : C_\Gamma^1 \mapsto C_\Gamma^0$ and $\tilde{\mathbb{D}}_1^* : C_\Gamma^2 \mapsto C_\Gamma^2$, given by

$$\tilde{\mathbb{D}}_0^* = \mathbb{M}_0^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1 \quad \text{and} \quad \tilde{\mathbb{D}}_1^* = \tilde{\mathbb{M}}_1^{-1} \mathbb{D}_1^T \mathbb{M}_2,$$

respectively, and such that $\tilde{\mathbb{D}}_0^* \tilde{\mathbb{D}}_1^* = 0$ and $\tilde{\mathbb{D}}_1^* \mathbb{D}_2^* = 0$. These operators give rise to the discrete Hodge Laplace operators

$$\tilde{\mathbb{L}}_0 : C_\Gamma^0 \mapsto C_\Gamma^0; \quad \tilde{\mathbb{L}}_0 = \tilde{\mathbb{D}}_0^* \mathbb{D}_0;$$

and

$$\tilde{\mathbb{L}}_1 : C_\Gamma^1 \mapsto C_\Gamma^1; \quad \tilde{\mathbb{L}}_1 = \tilde{\mathbb{D}}_1^* \mathbb{D}_1 + \mathbb{D}_0 \tilde{\mathbb{D}}_0^*;$$

that are different versions of \mathbb{L}_0 and \mathbb{L}_1 , respectively.

The following general result from [6] provides the results needed for the reformulation of the discrete eddy current equations.

THEOREM 3.1. *The size of the kernel of the analytic and discrete Hodge Laplacians is the same.*

Theorem 3.1 reveals that the null-space of the discrete Hodge Laplacian and, by extension the structure of the discrete Hodge decomposition of discrete functions in C_Γ^k , are topological invariants that are independent of the particular choice of metric, i.e., the matrices \mathbb{M}_k . As a result, the assertion of this theorem is valid for both $\mathbb{L}_0, \mathbb{L}_1$

and $\tilde{\mathbb{L}}_0, \tilde{\mathbb{L}}_1$. The properties of these operators, relevant to the reformulation process, are summarized in the following corollary.

COROLLARY 3.2. *Assume that Ω is contractible. Then, every $e^1 \in C_\Gamma^1$ has the discrete Hodge decomposition*

$$e^1 = \mathbb{D}_0 p^0 + \tilde{\mathbb{D}}_1^* b^2 \quad (3.12)$$

where $p^0 \in C_\Gamma^0$ and $b^2 \in C_\Gamma^2$ solve the equations

$$\tilde{\mathbb{D}}_0^* \mathbb{D}_0 p^0 = \tilde{\mathbb{D}}_0^* e^1 \quad \text{and} \quad \mathbb{D}_1 \tilde{\mathbb{D}}_1^* b^2 = \mathbb{D}_1 e^1, \quad (3.13)$$

respectively. The equation

$$\tilde{\mathbb{D}}_0^* \mathbb{D}_0 q^0 = 0 \quad (3.14)$$

has only the trivial solution $q^0 = 0$.

Proof. If Ω is contractible, then the analytic vector Laplacian (3.10) with the boundary conditions (3.11) has a trivial null-space and Theorem 3.1 implies that $\ker \tilde{\mathbb{L}}_1 = \{0\}$. As a result, the decomposition of any $e^1 \in C_\Gamma^1$ does not include discrete harmonics, i.e., it has the form (3.12) where p^0 and b^2 solve the equations in (3.13); see [6]. The second assertion is simply the statement that $\ker \tilde{\mathbb{L}}_0 = \{0\}$. It follows from Theorem 3.1 and the fact that the corresponding analytic Laplacian $\gamma^{-1} \nabla \cdot \sigma \nabla$ with a Dirichlet boundary condition has a trivial null-space. \square

4. Compatible discretization of the eddy current equations. To obtain the fully discrete eddy-current equations we proceed to apply Rothe's method (semi-discretization in time). For simplicity we consider the Backward Euler scheme and use \mathbf{E} and \mathbf{E}_o to denote the unknown value of the electric field at the new time step and the value computed at the previous time step. The discretized in time equations are

$$\mathbf{E} + \Delta t \sigma^{-1} \nabla \times \mu^{-1} \nabla \times \mathbf{E} = \mathbf{E}_o.$$

Using the discrete operators defined in the last section, a compatible discretization of the semi-discrete in time equation is straightforward. Specifically, we approximate \mathbf{E} by a 1-cochain $e^1 \in C_\Gamma^1$, i.e., by values associated with the 1-cells (the edges) of the mesh that are not in Γ . Then, the compatible discrete version of the curl-curl operator is provided by the second order discrete operator $\mathbb{D}_1^* \mathbb{D}_1$. As a result, the compatible, fully discrete eddy current model is given by

$$e^1 + \Delta t \mathbb{D}_1^* \mathbb{D}_1 e^1 = e_o^1, \quad (4.1)$$

where e_o^1 is the approximate electric field at the old time step. Because the actual value of the time step is irrelevant for the subsequent developments, we set $\Delta t = 1$ for simplicity. An equivalent "weak" form of (4.1) is given by the variational equation: seek $e^1 \in C_\Gamma^1$ such that

$$(e^1, \hat{e}^1)_{C^1} + (\mathbb{D}_1 e^1, \mathbb{D}_1 \hat{e}^1)_{C^2} = (e_o^1, \hat{e}^1) \quad \forall \hat{e}^1 \in C_\Gamma^1. \quad (4.2)$$

The following theorem shows that the discrete equations (4.1) or (4.2) inherit the involution of the continuous problem.

THEOREM 4.1. *The discrete problem (4.1) has the involution $\mathbb{D}_0^* e^1 = 0$.*

Proof. Clearly, it suffices to prove that $\mathbb{D}_0^* e^1 = 0$ provided $\mathbb{D}_0^* e_o^1 = 0$. Assuming that the latter is true, we proceed to rewrite (4.1) by using the definitions of the natural and derived operations as follows:

$$e^1 + \mathbb{M}_1^{-1} \mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 e^1 = e_o^1,$$

or, which is the same:

$$\mathbb{M}_1 e^1 = -\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 e^1 + \mathbb{M}_1 e_o^1.$$

We left-multiply this equation by $\mathbb{M}_0^{-1} \mathbb{D}_0^T$ and use that $\mathbb{D}_0^* = \mathbb{M}_0^{-1} \mathbb{D}_0^T \mathbb{M}_1$, $\mathbb{D}_0^T \mathbb{D}_1^T = 0$ (from (3.3)), and $\mathbb{D}_0^* e_o^1 = 0$ (by assumption) to find that

$$\mathbb{D}_0^* e^1 = -\mathbb{M}_0^{-1} \mathbb{D}_0^T \mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 e^1 + \mathbb{D}_0^* e_o^1 = 0,$$

Therefore, the discrete solutions stays discretely divergence free at all time steps. \square

4.1. Reformulation. If $\mathbb{D}_0^* e_o^1 = 0$, then from Theorem 4.1 we know that $\mathbb{D}_0^* e^1 = 0$ and so, it is easy to see that the reformulated problem

$$e^1 + (\mathbb{D}_1^* \mathbb{D}_1 + \mathbb{D}_0 \mathbb{D}_0^*) e^1 = e_o^1, \quad (4.3)$$

is completely equivalent to (4.1). Compared to this problem, (4.3) has the advantage of using the operator $\mathbb{D}_1^* \mathbb{D}_1 + \mathbb{D}_0 \mathbb{D}_0^* = \mathbb{L}_1$. We remind that \mathbb{L}_1 was a compatible approximation of the Hodge Laplacian (3.10) with the boundary conditions (3.11) and that, according to Theorem 3.1, it has trivial null-space.

Notice that when $\sigma = \mu = \gamma = 1$, the operator (3.10) corresponds to an unscaled vector Laplacian. Since AMG methods typically work well for such Laplacians, this suggests an approach wherein AMG is applied to the reformulated equation (4.3) rather than to the original problem (4.1). Then, we would be able to devise AMG solvers for (4.1) by leveraging proven and efficient techniques developed for Laplace-like equations.

Unfortunately, the straight-forward application of AMG to \mathbb{L}_1 for arbitrary σ , μ and γ is likely to be problematic due to scaling issues associated with the material properties. One example of particular interest to us is the Z-pinch model, where the magnitude of σ can range from 1 to 10^7 ; see [7] for an example. In this case, in the regions where σ is much smaller than μ^{-1} and γ^{-1} the term $\sigma^{-1} \nabla \times \mu^{-1} \nabla \times$ will completely dominate (3.10). Translated to \mathbb{L}_1 , this means that $\mathbb{D}_1^* \mathbb{D}_1$ will completely dominate the discrete Hodge Laplacian and the beneficial effect from the gauge term $\mathbb{D}_0 \mathbb{D}_0^*$ will be lost, i.e., the system (4.3) effectively behaves similar to the original eddy current equations (4.1).

In this section we propose an alternative reformulation of (4.1) that retains the null-space property of (4.3) but offers a much better scaling for the resulting discrete Laplace-like operator acting on e^1 . The key idea is to consider an orthogonal decomposition of e^1 with respect to the inner product defined by the matrix \mathbb{M}_1 , instead of the more obvious choice of decomposition with respect to \mathbb{M}_1 . This leads to a system with the Laplace-like operator $\mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1^{-1} \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{D}_0^*$. This operator can be thought of as a compatible discretization of the second order differential operator

$$\sigma^{-1} (\nabla \times \mu^{-1} \nabla \times - \nabla \gamma^{-1} \nabla \cdot). \quad (4.4)$$

Furthermore, if $\mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1^{-1} \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{D}_0^*$ is multiplied on the left by \mathbb{M}_1 , the resulting operator $\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_0^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ is a compatible discretization of

$$(\nabla \times \mu^{-1} \nabla \times - \nabla \gamma^{-1} \nabla \cdot). \quad (4.5)$$

By choosing $\gamma = \mu$ and defining \mathbb{M}_0 accordingly, we can make the terms in (4.5) and its compatible discretization perfectly balanced. As a result, one can expect that AMG will perform well for the matrix $\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_0^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$.

There are two formal complications arising from this approach. First, even if $\mathbb{D}_0^* e_o^1 = 0$ we are not guaranteed that $\tilde{\mathbb{D}}_0^* e^1 = 0$ and so, the decomposition (3.12) of e^1 will necessarily include a function $p^0 \in C_\Gamma^0$. As a result, in contrast with (4.3), the alternative reformulated problem will have a 2×2 block structure. However, in practice it often happens that $\mathbb{D}_0^* e_o^1 \neq 0$, and so it is desirable to have algorithms that work in this general case as well. A second, seemingly more serious complication, is that using (3.12) to reformulate (4.1) leads to a problem with a Laplace-like operator that combines the term $\mathbb{D}_1^* \mathbb{D}_1$ from \mathbb{L}_1 and the term $\mathbb{D}_0 \tilde{\mathbb{D}}_0^*$ from $\tilde{\mathbb{L}}_1$, premultiplied by $\mathbb{M}_1^{-1} \tilde{\mathbb{M}}_1$, i.e., this operator is neither \mathbb{L}_1 , nor $\tilde{\mathbb{L}}_1$. As a result, Theorem 3.1 cannot be applied directly to show that this “mismatched” Laplacian will have a trivial null-space. Fortunately, using the results from Corollary 3.2 we show that this is indeed true.

THEOREM 4.2. *Assume that e^1 is a solution of (4.1) and let*

$$e^1 = \mathbb{D}_0 p^0 + \tilde{\mathbb{D}}_1^* b^2$$

denote its discrete Hodge decomposition with respect to the inner product induced by $\tilde{\mathbb{M}}_1$. The pair (a^1, p^0) , where $a^1 = \tilde{\mathbb{D}}_1^ b^2$, solves the linear system*

$$\begin{bmatrix} \mathbb{M}_1 + \mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_0^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1 & \mathbb{M}_1 \mathbb{D}_0 \\ \mathbb{D}_0^T \mathbb{M}_1 & \mathbb{D}_0^T \mathbb{M}_1 \mathbb{D}_0 \end{bmatrix} \begin{bmatrix} a^1 \\ p^0 \end{bmatrix} = \begin{bmatrix} \mathbb{M}_1 e_o^1 \\ \mathbb{D}_0^T \mathbb{M}_1 e_o^1 \end{bmatrix}. \quad (4.6)$$

Proof. To make the proof more transparent we eschew for a moment the matrix form (4.1) and work with the “weak” equation (4.2). Using the ansatz $e^1 = \mathbb{D}_0 p^0 + a^1$ in (4.2) together with the property of the natural operators that $\mathbb{D}_1 \mathbb{D}_0 \equiv 0$ gives the identity

$$(a^1, \hat{e}^1)_{C^1} + (\mathbb{D}_1 a^1, \mathbb{D}_1 \hat{e}^1)_{C^2} + (\mathbb{D}_0 p^0, \hat{e}^1)_{C^1} = (e_o^1, \hat{e}^1)_{C^1} \quad \forall \hat{e}^1 \in C_\Gamma^1.$$

Because $a^1 = \tilde{\mathbb{D}}_1^* b^2$ and $\tilde{\mathbb{D}}_0^* \tilde{\mathbb{D}}_1^* \equiv 0$, it follows that $\mathbb{D}_0 \tilde{\mathbb{D}}_0^* a^1 = 0$, or, using weak forms:

$$(\tilde{\mathbb{D}}_0^* a^1, \tilde{\mathbb{D}}_0^* \hat{e}^1)_{C^0} = 0 \quad \forall \hat{e}^1 \in C_\Gamma^1. \quad (4.7)$$

As a result, this term can be added to the last equation without changing it

$$(a^1, \hat{e}^1)_{C^1} + (\mathbb{D}_1 a^1, \mathbb{D}_1 \hat{e}^1)_{C^2} + (\tilde{\mathbb{D}}_0^* a^1, \tilde{\mathbb{D}}_0^* \hat{e}^1)_{C^0} + (\mathbb{D}_0 p^0, \hat{e}^1)_{C^1} = (e_o^1, \hat{e}^1)_{C^1} \quad \forall \hat{e}^1 \in C_\Gamma^1.$$

To complete the proof we now switch back to matrix notation. Using the definition of the inner product (3.4) and the matrix representation (3.6) of the derived operators, it is easy to see that the above weak equation is equivalent to the matrix equation

$$\mathbb{M}_1 a^1 + \left(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_0^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1 \right) a^1 + \mathbb{M}_1 \mathbb{D}_0 p^0 = \mathbb{M}_1 e_o^1$$

which is the first equation in (4.6). To obtain the second equation we left-multiply (4.1) by \mathbb{D}_0^* , use that $\mathbb{D}_0^* \mathbb{D}_1^* \equiv 0$, and replace e^1 by its Hodge decomposition:

$$\mathbb{D}_0^* e_o^1 = \mathbb{D}_0^* (\mathbb{D}_1^* \mathbb{D}_1 e^1 + e^1) = \mathbb{D}_0^* e^1 = \mathbb{D}_0^* (\mathbb{D}_0 p^0 + a^1) = \mathbb{D}_0^* a^1 + \mathbb{D}_0^* \mathbb{D}_0 p^0.$$

Using (3.6) to expand the matrices gives the equation

$$\mathbb{M}_0^{-1} \mathbb{D}_0^T \mathbb{M}_1 a^1 + \mathbb{M}_0^{-1} \mathbb{D}_0^T \mathbb{M}_1 \mathbb{D}_0 p^0 = \mathbb{M}_0^{-1} \mathbb{D}_0^T \mathbb{M}_1 e_o^1.$$

After left-multiplying by \mathbb{M}_0 we obtain the second equation in (4.6). It is clear that (4.6) and the discrete eddy current equation (4.1) are equivalent in the sense that if (a^1, p^0) is a solution of (4.6), then $e^1 = \mathbb{D}_0 p^0 + a^1$ is a solution of (4.1). \square

REMARK 4.3. Let $[r_a, r_p]^T$ denote the residual vector of (4.6) and r_e - the residual vector of (4.1). From the proof of Theorem 4.2 it follows that

$$\begin{bmatrix} r_a \\ r_p \end{bmatrix} = \begin{bmatrix} \mathbb{I} \\ \mathbb{D}_0^T \end{bmatrix} [r_e] \quad (4.8)$$

REMARK 4.4. By left multiplying the first and second equations in (4.6) by \mathbb{M}_1^{-1} and \mathbb{M}_0^{-1} , respectively we obtain the equivalent problem

$$\begin{bmatrix} \mathbb{I} + \mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1^{-1} \tilde{\mathbb{M}}_1 \mathbb{D}_0 \tilde{\mathbb{D}}_0^* & \mathbb{D}_0 \\ \mathbb{D}_0^* & \mathbb{D}_0^* \mathbb{D}_0 \end{bmatrix} \begin{bmatrix} a^1 \\ p^0 \end{bmatrix} = \begin{bmatrix} e_o^1 \\ \mathbb{D}_0^* e_o^1 \end{bmatrix}. \quad (4.9)$$

The second diagonal block in (4.9) is the operator \mathbb{L}_0 which, according to Theorem 3.1, has trivial null-space. However, the first diagonal block of (4.9) contains the Laplace-like operator $\mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1^{-1} \tilde{\mathbb{M}}_1 \mathbb{D}_0 \tilde{\mathbb{D}}_0^*$ which is neither \mathbb{L}_1 nor $\tilde{\mathbb{L}}_1$, and at this point we cannot describe its null-space.

REMARK 4.5. If we set $\tilde{\mathbb{M}}_1 = \mathbb{M}_1$, the term $\mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1^{-1} \tilde{\mathbb{M}}_1 \mathbb{D}_0 \tilde{\mathbb{D}}_0^*$ reduces to \mathbb{L}_1 and the reformulated system becomes

$$\begin{bmatrix} \mathbb{I} + \mathbb{L}_1 & \mathbb{D}_0 \\ \mathbb{D}_0^* & \mathbb{L}_0 \end{bmatrix} \begin{bmatrix} a^1 \\ p^0 \end{bmatrix} = \begin{bmatrix} e_o^1 \\ \mathbb{D}_0^* e_o^1 \end{bmatrix}. \quad (4.10)$$

Moreover, if $\mathbb{D}_0^* e^1 = 0$ the right hand side of the equation for p^0 in (3.13) equals to zero, and so $p^0 = 0$. Therefore, $e^1 = \mathbb{D}_1^* b^2 = a^1$ and (4.10) reduces to (4.3).

The following theorem shows that the Laplace-like operator in (4.9) has the same null-space as the proper Laplacians \mathbb{L}_1 and $\tilde{\mathbb{L}}_0$.

THEOREM 4.6. Assume that Ω is contractible and let $\mathbb{K}_1 = \mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1^{-1} \tilde{\mathbb{M}}_1 \mathbb{D}_0 \tilde{\mathbb{D}}_0^*$. Then $\ker \mathbb{K}_1 = \{0\}$.

Proof. The assertion of the theorem is equivalent⁴ to the statement that the variational equation: seek $z^1 \in C_\Gamma^1$ such that

$$\left(\mathbb{D}_1 z^1, \mathbb{D}_1 \hat{z}^1 \right)_{C^2} + \left(\tilde{\mathbb{D}}_0^* z^1, \tilde{\mathbb{D}}_0^* \hat{z}^1 \right)_{C^0} = 0 \quad \forall \hat{z}^1 \in C_\Gamma^1, \quad (4.11)$$

has only the trivial solution $z^1 = 0$.

⁴This follows from $\ker \mathbb{K}_1 = \ker \mathbb{M}_1 \mathbb{K}_1$ and

$$(z^1)^T \mathbb{M}_1 \mathbb{K}_1 z^1 = (z^1)^T \left(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + (\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_0^{-1}) \mathbb{M}_0 (\mathbb{M}_0^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1) \right) z^1.$$

Let $z^1 = \mathbb{D}_0 p^0 + \tilde{\mathbb{D}}_1^* b^2$ denote the discrete Hodge decomposition of z^1 with respect to the inner product induced by $\tilde{\mathbb{M}}_1$. We set $\hat{z}^1 = \mathbb{D}_0 p^0$ in (4.11), substitute z^1 by its decomposition and use the identities $\mathbb{D}_1 \mathbb{D}_0 = 0$ and $\tilde{\mathbb{D}}_0^* \tilde{\mathbb{D}}_1^* = 0$ to find that

$$0 = \left(\tilde{\mathbb{D}}_0^* \mathbb{D}_0 p^0, \tilde{\mathbb{D}}_0^* \mathbb{D}_0 p^0 \right)_{C^0} = \|\tilde{\mathbb{L}}_0 p^0\|_{C^0}^2.$$

From Corollary 3.2 we know that $\ker \tilde{\mathbb{L}}_0 = \{0\}$ and so, it follows that $p^0 = 0$. As a result, (4.11) reduces to

$$\left(\mathbb{D}_1 \tilde{\mathbb{D}}_1^* b^2, \mathbb{D}_1 \hat{z}^1 \right)_{C^2} = 0 \quad \forall \hat{z}^1 \in C^1.$$

After setting $\hat{z}^1 = \tilde{\mathbb{D}}_1^* b^2$ in this equation we find that

$$0 = \left(\mathbb{D}_1 \tilde{\mathbb{D}}_1^* b^2, \mathbb{D}_1 \tilde{\mathbb{D}}_1^* b^2 \right)_{C^2} = \|\mathbb{D}_1 \tilde{\mathbb{D}}_1^* b^2\|_{C^2}^2.$$

Therefore, $\mathbb{D}_1 \tilde{\mathbb{D}}_1^* b^2 = 0$ and because the complex (3.2) is exact, it follows that $\tilde{\mathbb{D}}_1^* b^2 = \mathbb{D}_0 r^0$ for some $r^0 \in C_1^0$. On the other hand, using that $\tilde{\mathbb{D}}_0^* \tilde{\mathbb{D}}_1^* = 0$ gives the identity

$$\tilde{\mathbb{D}}_0^* \mathbb{D}_0 r^0 = \tilde{\mathbb{D}}_0^* \tilde{\mathbb{D}}_1^* b^2 = 0.$$

Another application of (3.14) from Corollary 3.2 yields that $r^0 = 0$, and so, $\tilde{\mathbb{D}}_1^* b^2 = \mathbb{D}_0 r^0 = 0$. Therefore, $z^1 = \mathbb{D}_0 p^0 + \tilde{\mathbb{D}}_1^* b^2 = 0$. \square

Let us examine more closely the reformulated linear system (4.9). From the proofs of Theorem 4.2 and Theorem 4.6 it is clear that their assertions remain valid *for any* symmetric and positive definite matrices \mathbb{M}_0 and $\tilde{\mathbb{M}}_1$, i.e., regardless of their choice, (4.6) is always equivalent to (4.1) and $\ker \mathbb{K}_1 = \{0\}$ on any contractible domain Ω . Therefore, we can pick \mathbb{M}_0 and $\tilde{\mathbb{M}}_1$ that offer the most benefits for the AMG algorithm *without* compromising the null-space properties and the accuracy of the reformulated system. For instance, in the context of finite element analysis, this means that instead of the *consistent* nodal mass matrix, which is sparse but has a dense inverse, we can use a diagonal, lumped mass nodal matrix $\mathbb{M}_{0,l}$. This significantly decreases the cost of applying the reformulated system. Furthermore, to improve the scaling of the reformulated equations we choose to define $\tilde{\mathbb{M}}_1$ using a unit weight and \mathbb{M}_0 using $\gamma = \mu$. We recall that with this choice the terms in \mathbb{K}_1 and the associated Hodge Laplacian (4.5) are well-balanced.

In §5.1 we will construct a specialized prolongator for the reformulated system by using the *near* null-space of \mathbb{K}_1 . This construction will take advantage of the fact that not only does \mathbb{K}_1 have the same null-space as a proper Laplacian, but it also has a *near* null-space defined by

$$\phi_1 = \mathbb{D}_0 N_x \quad \phi_2 = \mathbb{D}_0 N_y \quad \phi_3 = \mathbb{D}_0 N_z, \quad (4.12)$$

where N_x , N_y and N_z are vectors corresponding to the x , y and z nodal coordinates, respectively. A near null-space vector, ϕ_k , has the property that $(\mathbb{K}_1)_{\mathcal{I},\cdot} \phi_k = 0$ where the subscript \mathcal{I},\cdot takes all matrix rows that are a graph distance of four or more from the boundary. This follows from the fact that $\mathbb{D}_1 \mathbb{D}_0 = 0$ and that $\tilde{\mathbb{D}}_0^* \mathbb{D}_0$ is a linear nodal finite element approximation to $u_{xx} + u_{yy} + u_{zz}$. The simple form of this Laplacian (without varying PDE coefficients) is due to the unit weight matrix associated with $\tilde{\mathbb{M}}_1$. Since the approximation is linear, it must annihilate all linear

functions away from the boundary. Analogous to a standard vector Laplacian, this near null-space corresponds to constants in each of the component directions. In particular, a gradient operator is applied to the functions x , y , and z . Further, this near null-space does not rely on the properties of \mathbb{M}_0 and so a diagonal lumped mass matrix can be used.

The use of a different inner product in the reformulation step is a key feature of our approach that distinguishes it from other proposed techniques. For instance, Bossavit [11] considers a transformation which, assuming that the initial data satisfies $\mathbb{D}_0^* e_o^1 = 0$, yields the system

$$(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1 \mathbb{D}_0 \mathbb{M}_0 (1/\mu\sigma^2) \mathbb{D}_0^T \mathbb{M}_1 + \mathbb{M}_1) e^1 = \mathbb{M}_1 e_o^1,$$

where $\mathbb{M}_0(1/\mu\sigma^2)$ is a diagonal matrix defined by $\mathbb{M}_0(1/\mu\sigma^2)_{nn} = \int_{\hat{c}_0} 1/\mu\sigma^2$. In this formula, \hat{c}_0 is the dual volume of the 0-cell c_0 and n is the global number of that cell. This problem is essentially the same as (4.3) which, according to Remark 4.5 can be obtained from our reformulation approach, by setting $\tilde{\mathbb{M}}_1 = \mathbb{M}_1$.

However, this Laplacian reformulation, discounting the mass term, is affected by changes (especially jumps) in the conductivity, σ . By using $\tilde{\mathbb{M}}_1$, independent of σ , our approach avoids these difficulties. Moreover, our analysis shows that it is not necessary to use dual volumes in the definition of \mathbb{M}_0 . Virtually any mass lumping procedure, in a consistent mass matrix will generate $\mathbb{M}_{0,l}^{-1}$ with the desired properties.

It is also instructive to compare our approach with the *reformulate and then discretize* method of [19]. Their reformulation begins with the Hodge decomposition of the original electric field $\mathbf{E} = \mathbf{A} + \nabla\phi$. Substitution into Maxwell's equations leads to a gauged system of equations that involves the operator $\nabla \times \mu^{-1} \nabla \times \mathbf{A} - \nabla \mu^{-1} \nabla \cdot \mathbf{A}$. A weak formulation of this operator is well-posed on the space $H(\text{div}) \cap H(\text{curl})$. This, however, constitutes a problem if the discretization of this operator has to be accomplished on a single grid. Indeed, a finite dimensional subspace of $H(\text{div}) \cap H(\text{curl})$ must contain fields that are both tangentially and normally continuous. On a single grid the only possible realization of such a field by, e.g., finite elements, is given by nodal C^0 elements. Such elements are clearly inappropriate for problems with material discontinuities because their solutions may have only tangential or normal continuity. As a result, the method of [19] relies on a staggered grid Yee-like scheme [44]. Effectively, a staggered grid method is equivalent to a primal-dual grid discretization in which the divergence and the curl operators are discretized by stencils on the primal and the dual grids, respectively.

In contrast, our approach starts with a curl-compatible discretization and then builds a discrete Laplacian that is consistent with the chosen discrete field representation. In this way, the problem of dealing with discretization of $H(\text{div}) \cap H(\text{curl})$ is avoided, and the reformulation can be carried for any discrete problem that fits in the framework of [6].

5. Multigrid solvers. We now combine reformulation and preconditioning to develop a linear solver for the compatible discretization of the eddy current equations. Specifically, consider the linear system

$$(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1) e = b \tag{5.1}$$

arising from (4.1). Once again, $\Delta t = 1$ to simplify the presentation and b is some given right hand side. As discussed in §4.1, an equivalent 2×2 block system

$$\begin{bmatrix} \mathbb{M}_1 + \mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1 & \mathbb{M}_1 \mathbb{D}_0 \\ \mathbb{D}_0^T \mathbb{M}_1 & \mathbb{D}_0^T \mathbb{M}_1 \mathbb{D}_0 \end{bmatrix} \begin{bmatrix} a^1 \\ p^0 \end{bmatrix} = \begin{bmatrix} b \\ \mathbb{D}_0^T b \end{bmatrix} \quad (5.2)$$

could alternatively be solved. While (5.1) is a smaller system, its nontrivial kernel makes preconditioning difficult. Therefore, we use (5.2) within the preconditioner.

There are several possible block preconditioning strategies to approximate the solution to (5.2). The approach considered in this paper focuses on developing AMG methods for the (1,1) and (2,2) blocks separately. The diagonal blocks (\mathbb{K}_1 and \mathbb{L}_0) are compatible discretizations of second order differential operators, and the off-diagonal blocks are first order, so (5.2) is dominated by the diagonal blocks. The key is that these diagonal blocks are Laplace-like. The (2,2) block of (5.2) is, in fact, amenable to any number of standard AMG techniques for variable nodal Laplace operators. Once constructed these AMG solvers can be combined in different ways to precondition (5.2). For example, a Jacobi-like preconditioner would ignore off-diagonal blocks and apply one AMG V-cycle to approximately invert the (1,1) block and another to approximately invert the (2,2) block. Other preconditioners could be based on methods like symmetric block Gauss-Seidel. One intriguing variant combines prolongators for the (1,1) and (2,2) blocks into a composite prolongator. In this way a V-cycle could be developed for the entire 2×2 system where coupling between a^1 and p^0 is maintained on all levels.

Unfortunately, while applying AMG to the (2,2) block is trivial, it is somewhat more complicated for the (1,1) block even though it is Laplace-like in nature. There are two difficulties associated with AMG and the (1,1) block of (5.2). The first is connected to the directionality of 1-cochain representations. In particular, each degree of freedom (DOF) in C^1 essentially corresponds to the tangent component of the electric field along the edges in C_1 . When restricting or prolongating, this directionality must somehow be taken into account. As a simple example, consider a standard orthogonal mesh in two dimensions. Horizontal edges have information about the field in the x direction while vertical edges have only information in the y direction. Thus, interpolation to a horizontal (or vertical) edge should only use information from coarse horizontal (or vertical) edges. Things are obviously more complex on unstructured meshes where tangent components of the electric field are not in general aligned with the coordinate axes. Unfortunately, such orientation issues are not generally considered in most standard AMG methods.

A second difficulty associated with the (1,1) block is the term $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$. For a general unstructured mesh, this term has a sparsity pattern similar to A^3 where A represents the original curl-curl operator. Simply forming this operator is expensive. As an example, consider a point disturbance at a single edge. The rightmost $\tilde{\mathbb{M}}_1$ transfers this disturbance to all adjacent edges (distance-1 edges). \mathbb{D}_0^T sends the disturbance to nodes. Since it is lumped, $\mathbb{M}_{0,l}^{-1}$ has no effect, but \mathbb{D}_0 transfers the disturbance pattern to edges adjacent to the nodes. This includes edges that are a distance-2 from the original edge. Finally, the leftmost $\tilde{\mathbb{M}}_1$ sends the disturbance as far as edges which are a distance of three from the initial edge. For a two-dimensional orthogonal mesh, the reformulated operator has the 31 edge stencil pattern shown in Figure 5.1. In 3D, or on non-orthogonal meshes, the stencil is larger.

Of course a primary goal of the reformulation is that it should be possible to

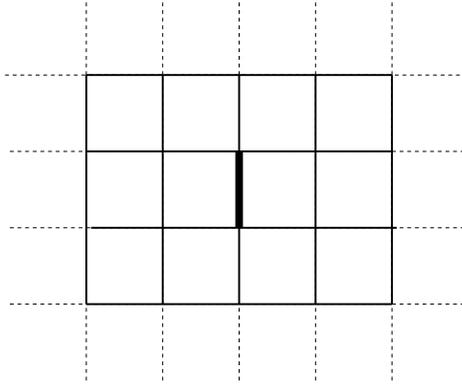


FIG. 5.1. The 31 edge stencil of $(1,1)$ block of (5.2) for a 2D orthogonal mesh. The thick black edge is the original edge and other solid edges represent the stencil.

leverage standard AMG solvers for Laplace-like operators. To address these concerns, we propose an AMG technique for the $(1,1)$ block which employs a somewhat specialized method at the finest level (i.e. for the first grid transfer), but allows subsequent levels and transfers to be handled with any standard AMG method. To do this, the edge element version of the $(1,1)$ block must be converted to a more standard nodal form on the coarse mesh⁵. This is accomplished by a special prolongator that not only transfers solutions from a coarse to a fine resolution but also transfers solutions from a nodal to an edge representation. The net effect of this special prolongator is that the corresponding Galerkin projection of the $(1,1)$ block will, in fact, yield a coarse operator resembling a vector nodal Laplacian which is amenable to any standard AMG method for further coarsening.

5.1. The specialized prolongator. For the specialized prolongator, we follow a smoothed aggregation philosophy. Smoothed aggregation is a successful AMG technique for solving second order elliptic systems. It is now available within several solver packages (both open source and commercial) and is described in [42, 43]. The basic idea is that a tentative (or simple) prolongator is first constructed and then later improved. The tentative prolongator is typically defined using a basis for a near null-space of dimension d . The near null-space corresponds to the true null-space of the PDE operator, ignoring boundary conditions. That is, one can consider the null-space of a modified PDE operator which usually consists of natural boundary conditions. For a nodal vector Laplace operator the near null-space is of dimension three, corresponding to a constant in each of the three coordinate directions.

In addition to a near null-space basis, the tentative prolongator requires a set of aggregates. These aggregates must be disjoint and must cover all DOFs. Ideally, an aggregate would include a root DOF and all of its neighbors, leading to aggregate widths of three in each dimension in the orthogonal mesh case. Each vector in the near null-space basis is then partitioned between aggregates to produce the tentative prolongator, \hat{P} . For each aggregate, \hat{P} normally has d columns which are only nonzero for DOFs within the aggregate. These d columns are often just the restriction of the near null-space basis to the aggregates (though orthogonalization is sometimes used).

⁵It is important to recall that the $(1,1)$ block is a Laplace-like operator *compatible* with 1-cochains and so, concerns associated with nodal representations and the eddy current equations do not apply.

By construction the tentative prolongator perfectly interpolates the near null-space. While \hat{P} could be used in the AMG V-cycle, it is typically improved by applying a single Jacobi iteration to the tentative prolongator. This improves the quality of the interpolant, P , and creates overlap between some columns associated with different aggregates. In a multilevel setting (e.g. $O(\log n)$ levels), the improvement of the tentative prolongator is usually necessary to obtain convergence rates which are independent of the number of mesh points, n .

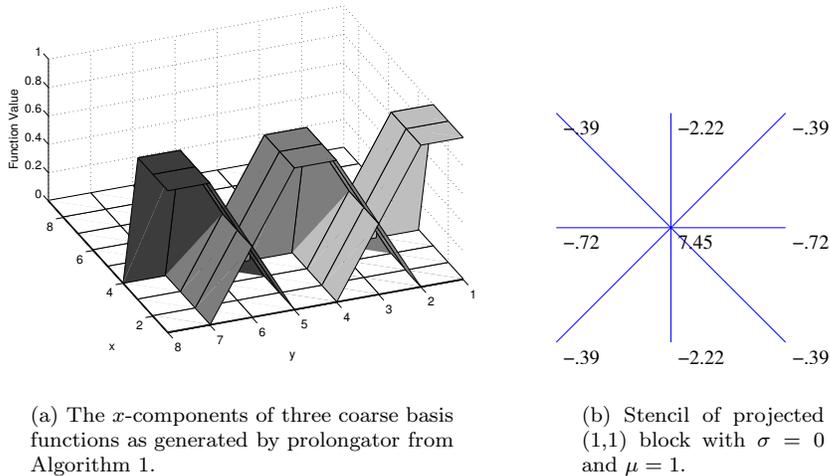
To apply the smoothed aggregation idea to the (1,1) block, the near null-space defined by (4.12) is used. We recall that this near null-space effectively corresponds to a constant electric field in the three coordinate directions. As in standard smoothed aggregation, aggregates must also be chosen. These aggregates are based on nodes as nodal quantities are employed on coarse levels. There are several ways to obtain aggregates corresponding to nodes. One possibility involves applying standard aggregation techniques to a nodal matrix $A_{nodal} = \mathbb{D}_0^T \mathbb{M}_1(1/\mu) \mathbb{D}_0$, where $\mathbb{M}_1(\mu)$ uses μ as the weight. This matrix captures the magnetic permeability in the (1,1) block. Notice that because of the special mass matrix, $\tilde{\mathbb{M}}_1$, σ only appears in the lowest order term of the (1,1) block and need not be considered when forming aggregates. Once the aggregates are formed, the near null-space must be partitioned to construct the tentative prolongator, denoted by \hat{P}_{11} , to emphasize that it is applied to the (1,1) block. It is important to notice that the near null-space is defined over edges while the aggregates are defined over nodes. Two cases must be considered when partitioning the near null-space: edges where endpoints lie within the same aggregate and edges where endpoints lie within different aggregates. In the former case, near null-space components are assigned to the corresponding aggregate columns. In the later case, there is some choice. In this paper we split equally the near null-space components between columns of the two different aggregates. Large variations in permeability may warrant a more careful choice of the near null-space partitioning. The detailed construction of the special prolongator for the (1,1) block is given in Algorithm 1, where $N = \{n_i\}$ is the set of nodes and $E = \{e_i\}$ is the set of edges with $e_j = (n_l, n_m)$ for some pair of nodes n_l and n_m . Notice that the net effect of \hat{P}_{11} is to interpolate coarse nodal quantities to fine edge-oriented quantities. Specifically, each aggregate corresponds to a coarse node and each DOF associated with a coarse node corresponds to a different near null-space component (or different coordinate direction). Figure 5.2(a) illustrates the support of three prolongator basis functions for

Algorithm 1: $\hat{P}_{11} = \text{Coarse_Node_to_Edge_Prolongator}(N, E, A_{nodal}, \{\phi_i\})$

- 1: $\{\mathcal{A}_i\} \leftarrow \text{Aggregate}(A_{nodal})$.
- 2: Let $d = |\{\phi_i\}|$, the number of near null-space vectors.
- 3: For $k = 1, \dots, d$ define

$$\hat{P}_{11}(i, (j-1)d + k) = \begin{cases} \phi_k(i) & \text{if } n_l \in \mathcal{A}_j, n_m \in \mathcal{A}_j \text{ with } e_i = (n_l, n_m) \\ \frac{1}{2}\phi_k(i) & \text{if } n_l \in \mathcal{A}_j, n_m \notin \mathcal{A}_j \text{ with } e_i = (n_l, n_m) \\ \frac{1}{2}\phi_k(i) & \text{if } n_l \notin \mathcal{A}_j, n_m \in \mathcal{A}_j \text{ with } e_i = (n_l, n_m) \\ 0 & \text{otherwise.} \end{cases}$$

\mathcal{A}_j is the j^{th} aggregate and $\phi_k(i)$ is the i^{th} component of the k^{th} near null-space vector defined by (4.12).



(a) The x -components of three coarse basis functions as generated by prolongator from Algorithm 1.

(b) Stencil of projected (1,1) block with $\sigma = 0$ and $\mu = 1$.

FIG. 5.2. 2D model problem with orthogonal structured mesh.

a two-dimensional structured mesh with 3×3 nodal aggregates. Figure 5.2(b) gives an interior coarse stencil corresponding to the projection of the (1,1) block using \hat{P}_{11} . For the figure, a two-dimensional structured mesh is used in conjunction with 3×3 nodal aggregates, $\sigma = 0$ and $\mu = 1$. Inspecting the coarse stencil, it is clear that this resembles a standard 9-point discretization of a slightly anisotropic Poisson operator. This anisotropy is due to the slightly anisotropic nature of the coarse basis functions. Other more complex choices for aggregates and near null-space partitioning would avoid this anisotropy.

\hat{P}_{11} has several important features that merit discussion. The first is that the number of nonzeros in the resulting coarse discretization stencil is quite reasonable. For example, 9-point stencils (as in Figure 5.2(b)) arise when a logically rectangular two-dimensional mesh is covered by 3×3 aggregates. This is in contrast with $\tilde{M}_1 \mathbb{D}_0 \tilde{M}_0^{-1} \mathbb{D}_0^T \tilde{M}_1$ which has significantly more nonzeros per matrix row. For this reason we omit the standard prolongator smoothing step normally used to improve the prolongator. That is, we take $P_{11} = \hat{P}_{11}$ to limit the size of the coarse stencil. It is important to note that this step is only omitted for the special prolongator used to transfer between the first coarse mesh and the finest mesh. A completely standard AMG method is applied to the resulting coarse operator. If, for example, the standard AMG method is smoothed aggregation, then it would apply the standard prolongator smoothing step to construct the coarser level prolongators. The significance of this is that while prolongator smoothing is important in a multilevel setting, it is not essential in a two-level setting. In particular, it is well-known that mesh independent convergence rates can be obtained when the tentative prolongator is used on elliptic problems within two-level domain decomposition schemes (see [27, 31, 37, 38, 39]). Thus, it is *reasonable* to expect that omitting the prolongator smoothing for only one grid transfer in the hierarchy will also give mesh independent convergence rates. Additionally, it is important to keep in mind that our prolongator has a degree of smoothness and overlapping support that is not present in the standard smoothed aggregation tentative prolongator. This gives a further expectation that mesh inde-

pendent convergence rates can be obtained. Numerical results will be presented which support this expectation. Finally, it should be noted that omitting the prolongator smoothing step avoids the need to explicitly form $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ in the coarsening process. Specifically, the Galerkin coarse discretization is given by

$$A_H = P_{11}^T A_{11} P_{11}$$

where A_{11} is the (1,1) block of (5.2) and A_H refers to its projection on a coarse mesh. Projection of the $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ term can be done efficiently by first forming $Z = \tilde{\mathbb{M}}_1 P_{11}$ followed by the computation $Z^T \mathbb{M}_{0,l}^{-1} Z$. Notice that if μ does not vary over time, then $Z^T \mathbb{M}_{0,l}^{-1} Z$ needs to be done only once over the entire time sequence. Further, computation of Z only needs to be done once if μ varies but the aggregation is held fixed (e.g. the variation in μ does not warrant changing aggregates).

REMARK 5.1. *Algorithm 1 has similarities to methods in [4, 23] which are used to interpolate scalar edge quantities to vector nodal quantities. The primary difference is that their interpolation, Π , translates between nodal and edge spaces on the **fine** mesh. In fact, our approach can be viewed as a composition of Π with a nodal prolongator. The nodal prolongator transfers coarse nodal quantities to fine nodal quantities which are then converted to edge quantities via Π . What is interesting is that depending on boundary conditions, it is possible that the dimension of Π 's domain is larger than the dimension of its range. In this case $\Pi^T A_f \Pi$ would be singular. Fortunately, this singularity does not arise with P_{11} as the nodal coarsening guarantees that the dimension of P_{11} 's domain is smaller than its range.*

5.2. Relaxation. While $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ is not explicitly needed during coarsening, the AMG relaxation must also avoid forming $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$. One possibility is to use Chebyshev relaxation methods for the (1,1) block [36, Algorithm 2.1]. These techniques make effective smoothers, work well on parallel computers, and are quite competitive with Gauss-Seidel methods even on traditional single CPU machines [1]. More importantly, they can be completely implemented using matrix-vector products and thus avoid the explicit formation of $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$. Of course, the matrix-vector product is costly due to the individual matrix-vector products (e.g. with $\tilde{\mathbb{M}}_1$ and \mathbb{D}_0^T) that now must be carried out.

An alternative idea that we advocate is to completely omit the $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ term during the fine grid (1,1) relaxation! To understand this, consider the following hybrid scheme. Suppose that the conjugate gradient iteration is actually applied to (5.1) and that (5.2) is *only used within the preconditioner*. To do this, it is necessary to convert residuals of (5.1) to right hand sides of (5.2) within the preconditioner. This is done by applying $[\mathbb{I} \quad \mathbb{D}_0]^T$ to the residual (see Remark 4.3). Approximate solutions to (5.2) are then converted back to a form suitable for (5.1) via $\mathbb{D}_0 p^0 + a^1$. While it might seem that there is no advantage, we have noticed that convergence rates are in fact better for the hybrid scheme as opposed to applying conjugate gradient directly to (5.2). This may be due to the fact that (5.1) is a smaller linear system (though we have not studied this issue). Further, when conjugate gradient is applied to (5.1), the matrix-vector product needed by conjugate gradient does not involve $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ and so it is less expensive. That is, utilization of (5.1) within conjugate gradient avoids one application of the (1,1) block.

Let us take the above idea one step further by considering the following hybrid preconditioner:

$$\begin{aligned}
\tilde{u} &\leftarrow \text{FineRelaxation}(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1, \mathbb{D}_0, 0, r) \\
r_{2 \times 2} &\leftarrow [I \quad \mathbb{D}_0]^T (r - (\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1) \tilde{u}) \\
(a, p) &\leftarrow \text{Solve}(A_{2 \times 2}, r_{2 \times 2}) \\
\tilde{u} &\leftarrow \tilde{u} + a + \mathbb{D}_0 p \\
\tilde{u} &\leftarrow \text{FineRelaxation}(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1, \mathbb{D}_0 \tilde{u}, r)
\end{aligned}$$

where $A_{2 \times 2}$ is the matrix in (5.2), $\text{Solve}()$ represents a single block AMG application to $A_{2 \times 2}$, and $\text{FineRelaxation}()$ is a method that smooths errors (including those within the curl-curl null-space) associated with solving (5.1). Algorithm 2 illustrates such a smoother proposed by Hiptmair that combines standard smoothing of the original equations with standard smoothing of the equations projected to the near null-space [22]. The key is that the error is smooth after this initial relaxation. Since

Algorithm 2: $\tilde{u} = \text{FineRelaxation}(A, \mathbb{D}_0, \tilde{u}, b)$

- 1: $\tilde{u} \leftarrow \text{StandardRelaxation}(A, \tilde{u}, b)$
 - 2: $c \leftarrow \text{StandardRelaxation}(\mathbb{D}_0^T A \mathbb{D}_0, 0, \mathbb{D}_0^T (b - A \tilde{u}))$
 - 3: $\tilde{u} \leftarrow \tilde{u} + \mathbb{D}_0 c$
 - 4: $\tilde{u} \leftarrow \text{StandardRelaxation}(A, \tilde{u}, b)$
-

the error is smooth, fine grid relaxation may be omitted from the AMG V-cycles in $\text{Solve}()$, as (5.1) and (5.2) are equivalent. This implies that $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ is completely avoided during fine grid relaxation. It is important to understand that $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ cannot be omitted during the Galerkin coarsening process. This term avoids the large null-space associated with the curl-curl operator on coarse grids. If not retained, the approximate inversion of the coarse operator would amplify modes in this large null-space. It is also important to realize that the somewhat specialized smoother is only needed on the finest level. A standard smoother can be used on coarse levels within the AMG procedures for the (1,1) and (2,2) blocks. Finally, it is interesting to note that an additive version of the Hiptmair smoother may also be considered for $\text{FineRelaxation}()$. In this case, the hybrid method is *almost* identical to the non-hybrid method where standard smoothing is used on the fine grid (1,1) and (2,2) blocks with the exception that $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1$ is omitted during the (1,1) fine grid relaxation.

5.3. AMG algorithm preconditioner. We now give the entire AMG-based preconditioner for the block Jacobi version in Algorithm 3. $\text{PreFineRelaxation}()$ is identical to Algorithm 2 except step one is omitted. This also avoids the residual calculation in step two as the initial guess to a preconditioner is always zero. $\text{PostFineRelaxation}()$ is identical to Algorithm 2 except step four is omitted to keep the preconditioner symmetric when $\text{StandardRelaxation}()$ employs a symmetric algorithm. Of course, residual calculations can also be avoided using additive forms of this smoother. It is important to notice that the only non-standard component in this AMG procedure is an algorithm to build the special interpolation operator, P_{11} . As already noted, this operator can be constructed using a standard AMG nodal aggregation method. The only new capabilities are the near null-space formation and a scheme for partitioning the near null-space over the aggregates. Constructing the near null-space amounts to applying \mathbb{D}_0 to the nodal coordinates; see (4.12) Partitioning the near null-space over aggregates is accomplished by Algorithm 1.

Algorithm 3: $\tilde{u} = \text{Block Preconditioner}(r)$

```

% Setup Phase
 $P_{11} = \text{Coarse\_Node\_to\_Edge\_Interpolant}(N, E, A_{nodal}, \{\phi_i\})$ 
Form  $A_H \leftarrow P_{11}^T (\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1 + \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{\mathbb{M}}_1) P_{11}$  efficiently
Standard_AMG_Setup( $A_H$ )
Standard_AMG_Setup( $\mathbb{D}_0^T \mathbb{M}_1 \mathbb{D}_0$ )

```

```

% Solve Phase
 $\tilde{u} \leftarrow \text{PreFineRelaxation}(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1, \mathbb{D}_0, 0, r)$ 
 $\tilde{r} \leftarrow r - (\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1) \tilde{u}$ 

% Perform V-cycles skipping fine grid smoothing
 $a \leftarrow \text{Standard\_AMG\_Vcycle}(A_H, 0, P_{11}^T \tilde{r})$  % Skip fine smoothing
 $p \leftarrow \text{Standard\_AMG\_Vcycle}(\mathbb{D}_0^T \mathbb{M}_1 \mathbb{D}_0, 0, \mathbb{D}_0^T \tilde{r})$  % Skip fine smoothing
 $\tilde{u} \leftarrow \tilde{u} + P_{11} a + \mathbb{D}_0 p$ 
 $\tilde{u} \leftarrow \text{PostFineRelaxation}(\mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \mathbb{M}_1, \mathbb{D}_0, \tilde{u}, r)$ 

```

It is interesting to notice that this procedure shares characteristics with the the multigrid methods given in [4, 23]. These methods essentially involve two AMG solves: one corresponding to the (2,2) block and the other corresponding to a nodal vector Laplacian. In addition, some relaxation must be performed on the original fine mesh system. The primary difference between these methods and Algorithm 3 is that the inversion of a vector Laplacian is replaced with the inversion of A_H which is a vector Laplace-like operator defined on a nodal coarse mesh (though our Laplace-like operator does not reside in a product of C^0 spaces). The advantage of our scheme is that it is completely algebraic and so there is no need to form a vector Laplacian on the fine mesh. There is also an exact relationship between the Laplace-like operator and the original eddy current system (as opposed to an asymptotic relationship). This exact relationship ensures that proper boundary conditions are associated with the Laplace-like operator. It is interesting to note that when μ is highly variable, the underlying PDE associated with our (1,1) block is slightly different from that in [4, 23]. Their schemes, however, do have an advantage in that they do not need to create a special prolongator operator. Additionally, the theoretical justification for the two methods is completely different.

6. Experiments and results. The proposed solver was implemented using CG in MATLAB. The first level and the first grid transfer of Algorithm 3 is also implemented in MATLAB. The coarse solver for the (1,1) and the solver for the (2,2) block use ML's smoothed aggregation solver, through the mlmex MATLAB interface [18]. A single V-cycle of AMG is used for both the (1,1) and (2,2) block, using the efficient variant of Algorithm 2 (smoother) described in §5.3. Unless otherwise stated, we use two steps of symmetric Gauss-Seidel sub-smoothing on both edges and nodes. The relatively small size of our examples is due to the MATLAB implementation, not any inherent limits in the solver.

For all experiments the CG tolerance is 1×10^{-10} . For each CG run, the iteration count, estimated overall convergence rate and the multigrid operator complexity of the preconditioner for the (1,1) block are reported. Specifically, this complexity is

Refinement	cplx	2 SGS Steps		3 SGS Steps		4 SGS Steps	
		Its.	CR	Its.	CR	Its.	CR
5.00e-01	1.23	14	0.20	13	0.18	12	0.16
2.50e-01	1.25	16	0.25	15	0.23	14	0.21
1.25e-01	1.24	17	0.28	16	0.25	15	0.23
6.25e-02	1.26	19	0.31	17	0.27	16	0.26

TABLE 6.1

Operator complexity (cplx), number of iterations (Its) and convergence rate (CR) for CG-accelerated AMG on the 2D triangular mesh problem with constant σ , using Algorithm 3. The size of the problem and the number of SGS smoothing steps are varied.

Grid	cplx	σ_2								
		10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
9^2	1.07	7	7	7	7	7	7	7	7	7
27^2	1.20	12	12	12	12	12	12	12	12	12
81^2	1.25	15	16	16	16	16	16	16	16	16
243^2	1.27	17	18	18	18	18	18	18	18	18

TABLE 6.2

Operator complexity (cplx) and iteration counts for CG-accelerated AMG on the 2D quad mesh problem, using Algorithm 3. The size of the problem and the internal square conductivity σ_2 are varied. The conductivity of the external region is $\sigma_1 = 1$.

defined as

$$\text{cplx} = \frac{\sum_{i=1}^{N\text{Levels}} \text{nnz}(A_i)}{\text{nnz}(A_1)}$$

where $\text{nnz}(A_i)$ gives the number of nonzeros for the discretization matrix associated with the i^{th} level in the AMG method for the (1,1) block. For the finest mesh, $i = 1$, and $\text{nnz}(A_1)$ does not include nonzeros that are caused solely by the $\tilde{M}_1 \mathbb{D}_0 \tilde{M}_{0,l}^{-1} \mathbb{D}_0^T \tilde{M}_1$ term. The zero initial guess is used, and the right hand side is chosen to be the vector of ones times the matrix. Unless otherwise stated, $\mu = 1$.

6.1. 2D triangular mesh. A two-dimensional unit square domain is considered with Neumann boundary conditions on the top and left and Dirichlet boundary conditions on the bottom and right. The domain is meshed using non-uniform triangles of approximately equal size. Four problems are considered corresponding to a mesh refinement of .5 (with 1,354 edges), .25 (with 5,021 edges), .125 (with 20,116 edges) and .0625 (with 79,874 edges). Table 6.1 reports the convergence results. We note that the convergence rates remain small over a range of mesh sizes though there is some mild deterioration as the mesh is refined. The multigrid operator complexity of the (1,1) block is also acceptable and seems relatively insensitive to mesh refinement.

6.2. 2D quads with variable conductivity. A two-dimensional unit square domain with homogeneous Neumann boundaries conditions is considered. The conductivity is varied from $\sigma_1 = 1$ to $\sigma_1 = 10^{-8}$ within an inner square of size $1/3 \times 1/3$ centered around the domain center. Outside of this inner square the conductivity is $\sigma = 1$. Table 6.2 reports iteration counts for this system. Note that the iteration counts are practically insensitive to jumps in the conductivity of up to eight orders of magnitude.

Grid	cplx	μ_2						
		10^0	10^{-1}	10^{-2}	10^{-3}	10^1	10^2	10^3
9^2	1.07	7	7	7	7	7	8	9
27^2	1.34	12	12	13	12	12	13	13
81^2	1.24	15	18	19	20	19	21	21
243^2	1.27	17	22	25	26	24	29	31

TABLE 6.3

Operator complexity (cplx) and iteration counts for CG-accelerated AMG on the 2D quad mesh problem, using Algorithm 3. The size of the problem and the internal square permeability μ_2 are varied. The permeability of the external region is $\mu_1 = 1$.

Refinement	cplx	2 SGS Steps		3 SGS Steps		4 SGS Steps	
		Its.	CR	Its.	CR	Its.	CR
$50 \times 10 \times 10$	1.11	11	0.14	10	0.10	9	0.08
$100 \times 20 \times 20$	1.09	14	0.21	13	0.18	12	0.16

TABLE 6.4

Operator complexity (cplx), number of iterations (Its) and convergence rate (CR) for CG-accelerated AMG on the 3D bar problem with varying σ , using Algorithm 3. The size of the problem and the number of SGS smoothing steps are varied.

6.3. 2D quads with variable permeability. We now consider the same domain as in §6.2, but we vary the permeability rather than the conductivity. The permeability ranges from $\mu_1 = 10^{-3}$ to $\mu_1 = 10^3$ within the inner square. Outside of this inner square we have $\mu = 1$. We set $\sigma = 1$ everywhere. Table 6.3 reports iteration counts for this system. Note that the iteration counts are moderately sensitive to jumps in the permeability. As noted earlier, a simple aggregation algorithm and a simple near null-space partitioning scheme is used on the finest mesh. This leads to prolongator stencils that cross between the two regions and probably causes this mild growth in iterations.

6.4. 3D bar. We consider an axially-aligned three dimensional bar-shaped region defined by $0 \leq x \leq 5$, $0 \leq y \leq 1$, and $0 \leq z \leq 1$. Dirichlet boundary conditions are applied to the $y = 0$ plane and Neumann boundary conditions are applied on all other faces. Conductivity is defined as

$$\sigma(x, y, z) = \begin{cases} 1. & 0 \leq x < 1 \\ .5 & 1 \leq x < 2 \\ .1 & 2 \leq x < 3 \\ .05 & 3 \leq x < 4 \\ .01 & 4 \leq x \leq 5 \end{cases}$$

The bar is meshed uniformly with either $50 \times 10 \times 10$ elements or $100 \times 20 \times 20$ elements, yielding a system with 17,270 or 128,940 edge unknowns, respectively. Table 6.4 reports the convergence results. We note that convergence deteriorates modestly with respect to grid refinement, and that the operator complexities of the (1,1) block are reasonable.

7. Conclusions. We have proposed an algebraic reformulation of the eddy current Maxwell's equations and an AMG technique for solving the reformulated problem.

The reformulation, which uses a discrete Hodge decomposition, replaces the discrete eddy current equations by an equivalent 2×2 block linear system whose diagonal blocks are discrete Hodge Laplace operators acting on 1-cochains and 0-cochains, respectively. This reformulation preserves the discrete solution and is applicable to a wide range of compatible methods on structured and unstructured grids, including edge finite elements, mimetic finite differences, co-volume methods and Yee-like schemes.

While our new AMG technique requires a specialized solver on the fine mesh, the coarser meshes can be handled using standard methods. This enables us to capitalize on existing technology for vector nodal Laplace-type solvers on the coarse levels. We have also presented numerical results in both two and three dimension showing very little sensitivity to jumps in the conductivity σ , moderate sensitivity to jumps in the permeability μ , good scaling with increasing mesh size and good operator complexity.

REFERENCES

- [1] M. ADAMS, M. BREZINA, J. HU, AND R. TUMINARO, *Parallel multigrid smoothing: polynomial versus Gauss-Seidel*, J. Comp. Phys., 188 (2003), pp. 593–610.
- [2] D. ARNOLD, R. FALK, AND R. WINTHER, *Multigrid in $h(\text{div})$ and $h(\text{curl})$* , Numer. Math., 85 (2000), pp. 197–217.
- [3] D. ARULIAH, U. ASCHER, E. HABER, AND D. OLDENBURG, *A method for the forward modelling of 3d electromagnetic quasi-static problems*, M3AS, 11 (2001), pp. 1–21.
- [4] R. BECK, *Algebraic multigrid by component splitting for edge elements on simplicial triangulations*, Tech. Report SC 99-40, Zuse Institute Berlin, December 1999.
- [5] P. BOCHEV, C. GARASI, J. HU, A. ROBINSON, AND R. TUMINARO, *An improved algebraic multigrid method for solving Maxwell's equations*, SIAM J. Sci. Comput., 25 (2003), pp. 623–642.
- [6] P. BOCHEV AND J. HYMAN, *Principles of mimetic discretizations of differential operators*, in Compatible Spatial Discretizations, D. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and M. Shashkov, eds., Springer-Verlag, 2006.
- [7] P. BOCHEV AND A. ROBINSON, *Matching algorithms with physics: exact sequences of finite element spaces*, in Preservation of stability under discretization, D. Estep and S. Tavener, eds., Philadelphia, 2001, SIAM, pp. 145–165.
- [8] T. BOONEN, G. DELIÉGE, AND S. VANDEWALLE, *On algebraic multigrid methods derived from partition of unity nodal prolongators*, Numerical Linear Algebra with Applications, 13 (2006), pp. 105–131.
- [9] A. BOSSAVIT, *Whitney forms: A class of finite elements for three dimensional computations in electromagnetism*, IEEE Proceedings, 135 (1988), pp. 493–500.
- [10] ———, *On the Lorenz gauge*, COMPEL, 18 (1999), pp. 323–336.
- [11] ———, *“Stiff” problems in eddy-current theory and the regularization of Maxwell's equations*, IEEE Transactions on Magnetics, 37 (2001), pp. 3542–3545.
- [12] C. BRYANT, C. EMSON, AND C. TROWBRIDGE, *A comparison of Lorentz gauge formulations in eddy current computations*, IEEE Transactions on Magnetics, 26 (1990), pp. 430–433.
- [13] ———, *General purpose 3-d formulation for eddy currents using the Lorenz gauge*, IEEE Transactions on Magnetics, 26 (1990), pp. 2373–2375.
- [14] C. BRYANT, C. EMSON, C. TROWBRIDGE, AND P. FERNANDES, *Lorentz gauge formulations involving piecewise homogeneous conductors*, IEEE Transactions on Magnetics, 34 (1998), pp. 2559–2562.
- [15] S. CAIRNS, *Introductory topology*, Ronald Press Co., New York, 1961.
- [16] M. CLEMENS, S. FEIGH, AND T. WEILAND, *Construction principles of multigrid smoothers for curl-curl equations*, IEEE Transactions on Magnetics, 41 (2005), pp. 1680–1683.
- [17] A. DEZIN, *Multidimensional analysis and discrete models*, CRC Press, Boca Raton, 1995.
- [18] M. GEE, C. SIEFERT, J. HU, R. TUMINARO, AND M. SALA, *ML 5.0 smoothed aggregation user's guide*, Tech. Report SAND2006-2649, Sandia National Laboratories, 2006.
- [19] E. HABER AND U. ASCHER, *Fast finite volume simulation of 3d electromagnetic problems with highly discontinuous coefficients*, SIAM J. Sci. Comput., 22 (2001), pp. 1943–1961.
- [20] E. HABER, U. ASCHER, D. ARULIA, AND D. OLDENBURG, *Fast simulation of 3d electromagnetic problems using potentials*, J. Comp. Phys., 163 (2000), pp. 150–171.
- [21] E. HABER AND S. HELDMANN, *An ocTree multigrid method for quasi-static Maxwell's equations*

- with highly discontinuous coefficients.* Submitted for publication.
- [22] R. HIPTMAIR, *Multigrid method for Maxwell's equations*, SIAM J. Numer. Anal., 36 (1998), pp. 204–225.
 - [23] R. HIPTMAIR AND J. XU, *Nodal auxiliary space preconditioning in $H(\text{curl})$ and $H(\text{div})$ spaces*, tech. report, 2006.
 - [24] J. HU, R. TUMINARO, P. BOCHEV, C. GARASI, AND A. ROBINSON, *Toward an h -independent algebraic multigrid method for Maxwell's equations*, SIAM J. Sci. Comput., 27 (2006), pp. 1669–1688.
 - [25] J. HYMAN AND M. SHASHKOV, *Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids*, Appl. Num. Math., 25 (1997), pp. 413–442.
 - [26] ———, *Natural discretizations for the divergence, gradient and curl on logically rectangular grids*, Comput. Math. Appl., 33 (1997), pp. 88–104.
 - [27] E. W. JENKINS, C. E. KEES, C. T. KELLEY, AND C. T. MILLER, *An aggregation-based domain decomposition preconditioner for groundwater flow*, SIAM Journal on Scientific Computing, 23 (2002), pp. 430–441.
 - [28] T. KOLEV, J. PASCIAK, AND P. VASSILEVSKI, *$H(\text{curl})$ auxiliary mesh preconditioning*, Tech. Report UCRL-JRNL-224227, Lawrence Livermore National Labs, June 2006.
 - [29] T. KOLEV AND P. VASSILEVSKI, *Parallel $H1$ -based auxiliary space AMG solver for $H(\text{curl})$ problems*, Tech. Report UCRL-TR-222763, Lawrence Livermore National Labs, July 2006.
 - [30] ———, *Some experience with a $H1$ -based auxiliary space AMG for $H(\text{curl})$ problems*, Tech. Report UCRL-TR-21841, Lawrence Livermore National Labs, June 2006.
 - [31] C. LASSER AND A. TOSELLI, *Convergence of some two-level overlapping domain decomposition preconditioners with smoothed aggregation coarse spaces*, in Recent Developments in Domain Decomposition Methods, L. Pavarino and A. Toselli, eds., vol. 23 of LNCSE, Springer Verlag, 2002.
 - [32] B. LEE AND C. TONG, *A novel algebraic multigrid-based approach for Maxwell's equations*, Tech. Report UCRL-JC-218750, Lawrence Livermore National Laboratory, 2006.
 - [33] R. NICOLAIDES, *Direct discretization of planar div-curl problems*, SIAM J. Numer. Anal., 29 (1992), pp. 32–56.
 - [34] J. PASCIAK AND J. ZHAO, *Overlapping schwartz methods in $h(\text{curl})$ on polyhedral domains*, Numer. Math, 10 (2002), pp. 221–234.
 - [35] S. REITZINGER AND J. SCHÖBERL, *An algebraic multigrid method for finite element discretizations with edge elements*, Numer. Linear Algebra Appl., 9 (2002), pp. 223–238.
 - [36] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, SIAM, 2nd ed., 2003.
 - [37] M. SALA, *Domain Decomposition Preconditioners: Theoretical Properties, Application to the Compressible Euler Equations, Parallel Aspects*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2003.
 - [38] ———, *Analysis of two-level domain decomposition preconditioners based on aggregation*, Mathematical Modelling and Numerical Analysis, 38 (2004), pp. 765–780.
 - [39] M. SALA, J. SHADID, AND R. TUMINARO, *An improved convergence bound for aggregation-based domain decomposition preconditioners*, SIAM J. Matrix Anal., 27 (2005), pp. 744–756.
 - [40] M. SHASHKOV, *Conservative finite difference methods on general grids*, CRC Press, Boca Raton, FL, 1996.
 - [41] J. VAN WELIJ, *Calculation of eddy currents in terms of H on hexahedra*, IEEE Transactions on Magnetics, 21 (1985), pp. 2239–2241.
 - [42] P. VANĚK, M. BREZINA, AND J. MANDEL, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.
 - [43] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing, 56 (1996), pp. 179–196.
 - [44] K. YEE, *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media*, IEEE Trans. Antennas and Propagation, 14 (1966), pp. 302–307.