TOWARDS ROBUST 3D Z-PINCH SIMULATIONS: DISCRETIZATION AND FAST SOLVERS FOR MAGNETIC DIFFUSION IN HETEROGENEOUS CONDUCTORS.[1]

PAVEL BOCHEV[2,4] JONATHAN HU[5], ALLEN C. ROBINSON[3], AND RAY TUMINARO[5]

Abstract. The mathematical model of the Z-pinch is comprised of many interacting components. One of these components is magnetic diffusion in highly heterogeneous media. In this paper we discuss finite element approximations and fast solution algorithms for this component, as represented by the eddy current equations. Our emphasis is on discretizations that match the physics of the magnetic diffusion process in heterogeneous media in order to enable reliable and robust simulations for even relatively coarse grids. We present an approach based on the use of exact sequences of finite element spaces defined with respect to unstructured hexahedral grids. This leads to algorithms that effectively capture the physics of magnetic diffusion. For the efficient solution of the ensuing linear systems we consider an algebraic multigrid method that appropriately handles the nullspace structure of the discretization matrices.

Accepted for publication in Elec. Trans. Numer. Analysis, 14 (2002).

Key words. Maxwell's equations, eddy currents, De Rham complex, finite elements, AMG.

AMS subject classifications. 76D05, 76D07, 65F10, 65F30

1. Introduction.

The Z-pinch is a technique for generating large material compressions and energies by generating a cylindrical implosion using focused magnetic field energy. Wire array implosions for example are used to generate extremely large X-ray power pulses [29]. Modeling requires a multiphysics approach which must include several interacting components. Components are coupled through interactions of forces, exchange of energy and etc. Our immediate interest is in developing a technology for Z-pinch modeling which falls within the constraints of an Arbitrary Lagrangian-Eulerian (ALE) modeling approach inherent in the framework of the ALEGRA code [21, 19]. In this code various physics components are modeled and coupled using operator splitting. Managing the complexity of the fully coupled model is made more tractable by examining each component separately and ensuring its reliability. In this report we focus on magnetic diffusion represented by a subset of the full Maxwell's equations referred to as the eddy current equations.

While finite element analysis of eddy currents is a relatively well-studied subject, placing this problem in the context of Z-pinch simulations brings up some specific modeling and computational issues. Most notably, conducting and non-conducting regions are not separated by a well-defined static interface. As a result, implementa-

[1]Sandia National Laboratories, SAND 2001-8363J, Submitted to Electronic Transactions of Numerical Analysis, Special Issue for Tenth Copper Mountain Conference on Multigrid Methods, February 2002

[2]This work was sponsored by NSF under grant number DMS-0073698 and the Computer Science Research Institute (CSRI) at Sandia National Laboratories. Permanent address: Department of Mathematics, Box 19408, University of Texas at Arlington, Arlington, TX 76019-0408 (bochev@uta.edu).

[3]Sandia National Laboratories, Computational Physics R&D Department, Albuquerque, NM 87185-0819 (acrobin@sandia.gov).

[4]CSRI, Sandia National Laboratories, P.O. Box 5800, MS 1110, Albuquerque, NM 87185-1110 (pbboche@sandia.gov).

[5]Sandia National Laboratories, PO Box 969, MS 9217, Livermore, CA 94551 (jhu@sandia.gov, rstumin@sandia.gov)

tion of standard methods based on the use of different magnetic potentials in conducting/nonconducting regions; see [4, 3], becomes prohibitively expensive (and complex). This forces consideration of the eddy current equations on a single, but highly heterogeneous, conductor as the only acceptable modeling choice.

For the finite element analysis of the ensuing problem with *nodal spaces* one can still formally adopt a potential approach based on a vector magnetic potential $\mathbf{A}$. The difficulties that arise in this context stem from the need to gauge the resulting boundary value problem, i.e., augment it by additional equations and boundary conditions. The Coulomb gauge $\nabla \cdot \mathbf{A} = 0$ is hard to satisfy numerically and must be added implicitly to the formulation which creates a cascading effect of adding more and more equations; see [3]. Another choice is the Lorentz gauge; see [14, 12, 13]. For heterogeneous conductors this gauge leads to nonsymmetric weak equations and is thus undesirable. Application of standard nodal spaces complicates imposition of tangential and normal boundary conditions which are typical for the eddy current equations.

An alternative to nodal approximations of gauged vector potential equations is to discretize directly the eddy current equations using exact sequences of finite element spaces. The rationale behind this approach is that such finite element spaces represent approximations of a De Rham complex that describes the mathematical structure of Maxwell's equations; see [5, 6]. These spaces have the important advantage of providing natural degrees of freedom for purposes of implementing tangential Dirichlet boundary conditions.

In this paper we pursue two main objectives. The first one is to develop such finite element spaces on unstructured hexahedral grids and test their use for finite element analysis of the eddy current model relevant to the Z-pinch. Here, our main focus is on the development of the discrete model and verifying its fidelity to the physics of magnetic diffusion. The finite element spaces are considered in section 3 followed by development of the fully discrete equations in section 4. The formulation is tested for a model 2D problem in section 6.

The second objective is to develop fast, scalable solvers for the discrete eddy current equations. These solvers must address the special structure of the linear systems inherent in the use of the exact finite element sequences. They also must work well for realistic values of the material modeling parameters which may vary over many orders of magnitude in a highly heterogeneous way. Although a hierarchical grid is available in the ALEGRA framework, restricting application modelers to such grids is considered to be unacceptable. As a result, the main focus here is on the development and implementation of a precisely designed algebraic multigrid method which operates directly on the assembled discrete matrix.

Throughout the paper bold face is used to denote vector quantities. The symbols $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ stand for the Cartesian coordinate vectors in $\mathbf{R}^3$, equipped with the Euclidean norm $\|\cdot\|$, while $\mathbf{n}$ and $\mathbf{t}$ denote an outward unit normal field to a surface and a unit tangent field (to a curve), respectively. The symbols $L^2(\Omega)$ and $\mathbf{L}^2(\Omega)$ will denote the spaces of all square integrable scalar and vector functions on Ω.

2. The model problem. The eddy current equations are obtained by neglecting the displacement current in the full Maxwell equations. This amounts to neglecting high frequency speed-of-light time scale electromagnetic waves in a conducting media. The model problem considered in this paper is that of a single conducting region Ω in $\mathbf{R}^3$ with non-constant conductivity σ and permeability μ. We assume that the boundary Γ of this region consists of two disjoint parts denoted by Γ^* and Γ, respectively.

Furthermore, it is assumed that the conductivity σ and the permeability μ are single valued bounded non vanishing functions depending only on the spatial position $\mathbf{x}$. No particular smoothness of the coefficients can be assumed. For the most pressing application of interest, however, μ is constant. Furthermore, we assume that

$$0 < \sigma_{\min} \leq \sigma(\mathbf{x}) \leq \sigma_{\max} \quad \forall \mathbf{x} \in \Omega \tag{2.1}$$

$$0 < \mu_{\min} \leq \mu(\mathbf{x}) \leq \mu_{\max} \quad \forall \mathbf{x} \in \Omega. \tag{2.2}$$

The governing equations for the electromagnetic field in Ω are given by

$$\nabla \times \mathbf{H} = \mathbf{J} \quad \text{in } \Omega \tag{2.3}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad \text{in } \Omega \tag{2.4}$$

$$\nabla \cdot \mathbf{B} = 0 \qquad \text{in } \Omega \tag{2.5}$$

$$\nabla \cdot \mathbf{J} = 0 \qquad \text{in } \Omega. \tag{2.6}$$

where $\mathbf{H}$ is the magnetic field, $\mathbf{J}$ is the current density, $\mathbf{E}$ is the electric field, and $\mathbf{B}$ is the magnetic flux density. Initial values of the magnetic flux density $\mathbf{B}$ are required to satisfy (2.5). These fields are connected by the constitutive relations

$$\mathbf{B} = \mu \mathbf{H} \tag{2.7}$$

$$\mathbf{J} = \sigma \mathbf{E}. \tag{2.8}$$

Eq. (2.3) is the Ampere's theorem and (2.4) is the Faraday's law, while (2.8) is the Ohm's law. System (2.3)-(2.6) must be closed by choosing appropriate boundary conditions. Here we consider *Type I* conditions

$$\mathbf{n} \times \mathbf{E} = \mathbf{n} \times \mathbf{E}_{\mathrm{b}} \quad \text{and} \quad \mathbf{n} \cdot \mathbf{B} = \mathbf{n} \cdot \mathbf{B}_{\mathrm{b}} \quad \text{on } \Gamma^* \tag{2.9}$$

and *Type II* conditions

$$\mathbf{n} \times \mathbf{H} = \mathbf{n} \times \mathbf{H}_{\mathrm{b}} \quad \text{and} \quad \mathbf{n} \cdot \mathbf{J} = \mathbf{n} \cdot \mathbf{J}_{\mathrm{b}} \quad \text{on } \Gamma. \tag{2.10}$$

To deliver robust, 3D fully integrated Z-pinch calculations, finite element simulations of the eddy current equations (2.3)-(2.6), (2.7)-(2.8) and (2.9)-(2.10) must meet certain requirements. From the modeling point of view the main requirement is to obtain high fidelity simulation of the magnetic field diffusion in highly heterogeneous media. This fidelity must be maintained both at the ideal MHD limit $\sigma \to \infty$, as well as at the highly diffusive limit $\sigma \to 0$. Furthermore, it is desirable to advance the magnetic flux density in a manner which maintains $\nabla \cdot \mathbf{B} = 0$ at all time steps. From the computational point of view the demand is on scalability of the solvers of the discrete linear system for realistic values of the modeling parameters. Scalability implies approximately linear work in the number of unknowns to find a high quality solution to the discrete linear eddy current system. Scalability will be discussed in §5 and §6.2.

3. Approximation of De Rham's complex on hexahedra. In this section we develop exact sequences of finite element spaces on unstructured hexahedral and quadrilateral grids. This choice is dictated by the ALEGRA computing framework which supports ALE hydrodynamics on arbitrary quadrilaterial and hexahedral grids

[19]. An intuitive method for developing edge and face elements on arbitrary hexahedra (isoparametric bricks) was first given by van Welij [28]. The van Welij elements are defined directly in the computational domain using the coordinate functions of the inverse mapping between a reference and computational elements. Here we develop a general approach that follows this idea and includes the edge elements of van Welij as a special case. For parallelepipeds or parallelograms these finite elements also include the well-known spaces of Nedelec, Brezzi, Douglas, Fortin and Marini, among others; see [10], [15], [17] and [18]. However, for general hexahedral grids the elements used here are quite different because they do not form an affine family of finite element spaces ([8, p.72]). We also show how to obtain proper restrictions of these spaces in two dimensions and discuss specifics of the exactness relation in $\mathbf{R}^2$.

The notions of exactness and De Rham complex are closely related to the mathematical structure of Maxwell's equations. The domains of the differential operators gradient, curl and divergence, relative to Γ are

$$H_0(\Omega, \mathrm{grad}) = \{\phi \in H(\Omega, \mathrm{grad}) | \phi = 0 \quad \text{on } \Gamma\}, \tag{3.1}$$

$$H_0(\Omega, \mathbf{curl}) = \{\mathbf{u} \in H(\Omega, \mathbf{curl}) | \mathbf{u} \times \mathbf{n} = 0 \quad \text{on } \Gamma\}, \tag{3.2}$$

$$H_0(\Omega, \mathrm{div}) = \{\mathbf{u} \in H(\Omega, \mathrm{div}) | \mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma\}, \tag{3.3}$$

where

$$H(\Omega, \mathrm{grad}) = \{\phi \in L^2(\Omega) | \nabla\phi \in \mathbf{L}^2(\Omega)\}, \tag{3.4}$$

$$H(\Omega, \mathbf{curl}) = \{\mathbf{u} \in \mathbf{L}^2(\Omega) | \nabla \times \mathbf{u} \in \mathbf{L}^2(\Omega)\}, \tag{3.5}$$

$$H(\Omega, \mathrm{div}) = \{\mathbf{u} \in \mathbf{L}^2(\Omega) | \nabla \cdot \mathbf{u} \in L^2(\Omega)\}. \tag{3.6}$$

The four spaces $H_0(\Omega, \mathrm{grad})$, $H_0(\Omega, \mathbf{curl})$, $H_0(\Omega, \mathrm{div})$, $L^2(\Omega)$ and the three operators ∇, $\nabla\times$ and $\nabla\cdot$ form a *De Rham complex relative to* Γ.

The dual complex can be introduced by using the adjoint differential operators ∇^*, $(\nabla\times)^*$ and $(\nabla\cdot)^*$. A fundamental property of the De Rham complex is the exactness of the sequence

$$H(\Omega, \mathrm{grad}) \xrightarrow{\nabla} H(\Omega, \mathbf{curl}) \xrightarrow{\nabla\times} H(\Omega, \mathrm{div}) \xrightarrow{\nabla\cdot} L^2(\Omega). \tag{3.7}$$

Exactness means that each differential operator maps the space to its left into the kernel of the next differential operator. Importance of this property stems from the fact that Maxwell's equations can be described in terms of a *Tonti* diagram built upon this complex; see [6]:

$$\begin{array}{ccccc}
\hline
 & Ampere & & Faraday & \\
\hline
H_0(\Omega, \mathrm{grad}) & \psi & & 0 & L_0^2(\Omega) \\
\nabla & \downarrow & & \Uparrow & \nabla\cdot \\
H_0(\Omega, \mathbf{curl}) & \mathbf{H} & \Rightarrow \ \mu\mathbf{H} = \mathbf{B} \ \Rightarrow & \mathbf{B} & H_0^*(\Omega, \mathrm{div}) \\
\nabla\times & \Downarrow & & \Uparrow & \nabla\times \\
H_0(\Omega, \mathrm{div}) & \mathbf{J} & \Leftarrow \ \mathbf{J} = \sigma\mathbf{E} \ \Leftarrow & \mathbf{E} & H_0^*(\Omega, \mathbf{curl}) \\
\nabla\cdot & \Downarrow & & \uparrow & \nabla \\
L_0^2(\Omega) & 0 & & \phi & H_0^*(\Omega, \mathrm{grad}) \\
\hline
\end{array} \tag{3.8}$$

Suppose now that $\mathcal{W}^i$, $i = 0,\ldots,3$ are finite element subspaces of $H_0(\Omega,\mathrm{grad})$, $H_0(\Omega,\mathbf{curl})$, $H_0(\Omega,\mathrm{div})$, and $L^2(\Omega)$ defined with respect to some triangulation $\mathcal{T}_h$ of Ω into finite elements. Furthermore, suppose that $\mathcal{W}^i$ form an exact sequence, i.e., they approximate not only the individual spaces but the De Rham complex as a whole. Then, a discretization of the Maxwell's equations can be obtained by substituting the De Rham complex in (3.8) by the exact sequence $\mathcal{W}^i$; see [7], [5]. This approach will be pursued in 4.

3.1. Exact sequence on a generalized hexahedral. Consider $\mathbf{R}^3$ endowed with a physical coordinate frame $(x_1,x_2,x_3) \equiv \mathbf{x}$ and a parameter (or reference) frame $(\xi_1,\xi_2,\xi_3) \equiv \boldsymbol{\xi}$. In what follows the indices α, β and γ take the values ± 1 and the indices i,j,k form an even permutation of the numbers $1,2,3$. Let $\hat{K}$ denote the open cube $(-1,1)^3$ in the reference space and let K denote its image under a smooth deformation $F : \hat{\mathbf{R}}^3 \mapsto \mathbf{R}^3$ of $\mathbf{R}^3$. We refer to K as *generalized* hexahedral. Construction of an exact sequence on K will be carried for general F assuming only that

- $F = (F_1,F_2,F_3)$ is invertible when restricted to $\hat{K}$
- $G = (G_1,G_2,G_3) = F^{-1}$ is such that $G(K) = \hat{K}$.

Restriction of F to a particular class of mappings will specialize further the exact sequence to a desired class of hexahedral grids. Since here we will be ultimately concerned with trilinear mappings F for simplicity we only consider unisolvency sets consisting of the vertices, edges, and faces

$$\begin{aligned}\boldsymbol{\xi}^{\alpha\beta\gamma} &= \{\xi_i = \alpha,\ \xi_j = \beta,\ \xi_k = \gamma\}\\ \boldsymbol{\xi}_{ij}^{\alpha\beta} &= \{\xi_i = \alpha,\ \xi_j = \beta,\ -1 \le \xi_k \le 1\}\\ \boldsymbol{\xi}_i^{\alpha} &= \{\xi_i = \alpha,\ -1 \le \xi_j,\xi_k \le 1\}.\end{aligned}$$

and the hexahedral $K = \{\mathbf{x}\,|\,\mathbf{x} = F(\boldsymbol{\xi}); \boldsymbol{\xi} \in \hat{K}\}$ itself[1]. Restriction of F to the sets above induces "vertices", "edges", and "faces" on K according to

$$\mathbf{x}^{\alpha\beta\gamma} = F(\boldsymbol{\xi}^{\alpha\beta\gamma}); \quad \mathbf{x}_{ij}^{\alpha\beta} = F(\boldsymbol{\xi}_{ij}^{\alpha\beta}); \quad \text{and} \quad \mathbf{x}_i^{\alpha} = F(\boldsymbol{\xi}_i^{\alpha}),$$

respectively. Note that

$$\mathbf{x}_i^{\alpha} \cap \mathbf{x}_j^{\beta} = \mathbf{x}_{ij}^{\alpha\beta} \quad \text{and} \quad \mathbf{x}_i^{\alpha} \cap \mathbf{x}_j^{\beta} \cap \mathbf{x}_k^{\gamma} = \mathbf{x}^{\alpha\beta\gamma}.$$

Next consider the Jacobians $J_F = (V_1,V_2,V_3)$ and $J_G = (\nabla G_1, \nabla G_2, \nabla G_3)^T$ where $V_i = (\partial F_1/\partial\xi_i,\ldots,\partial F_3/\partial\xi_i)^T$. Clearly, $\det J_F = V_i \cdot (V_j \times V_k)$ and $\det J_G = \nabla G_i \cdot (\nabla G_j \times \nabla G_k)$. From the identity $(F\circ G)(\mathbf{x}) = \mathbf{x}$ it follows that $J_F J_G = J_G J_F = I$. This relation means that

$$V_i \cdot \nabla G_j = \delta_{ij}, \tag{3.9}$$

i.e., the columns V_i of J_F and the rows ∇G_j^T of J_G are bi-orthogonal. Solving (3.9) for V_i and ∇G_j gives

$$V_i = (\nabla G_j \times \nabla G_k)\det J_F \quad \text{and} \quad \nabla G_i = (V_j \times V_k)\det J_G \tag{3.10}$$

[1] A unisolvency set for a given class of functions is a collection of data and data location pairs that defines a unique function out of the class. For instance, the unisolvency set for linear polynomials in one dimension consists of two distinct points with two prescribed values. For higher order polynomials and/or higher space dimensions these sets have to be expanded by including more nodes, edges and surfaces to the unisolvency sets.

The unit normal to a face $\mathbf{x}_i^\alpha$ and the unit tangent to an edge $\mathbf{x}_{ij}^{\alpha\beta}$ are given by

$$\mathbf{n} = \frac{\nabla G_i}{\|\nabla G_i\|} \quad \text{and} \quad \mathbf{t} = \frac{(\nabla G_i \times \nabla G_j)}{\|\nabla G_i \times \nabla G_j\|} \tag{3.11}$$

respectively. Changing variables in (3.11) and using (3.10) shows that corresponding vector fields on $\hat{K}$ are

$$(\mathbf{n} \circ F) = \frac{V_j \times V_k}{\|(V_j \times V_k)\|} \quad \text{and} \quad (\mathbf{t} \circ F) = \frac{V_k}{\|V_k\|}, \tag{3.12}$$

respectively. Let $\phi_i^\alpha(\mathbf{x}) = \frac{1}{2}(1 + \alpha G_i(\mathbf{x}))$. We consider four sets of functions defined on K as follows:

$$W_{ijk}^{\alpha\beta\gamma} = \phi_i^\alpha \phi_j^\beta \phi_j^\gamma \tag{3.13}$$

$$W_{ij}^{\alpha\beta} = \phi_i^\alpha \phi_j^\beta \nabla \phi_k^\gamma \tag{3.14}$$

$$W_i^\alpha = \phi_i^\alpha (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma) \tag{3.15}$$

$$W = \nabla \phi_i^\alpha \cdot (\nabla \phi_j^\beta \times \nabla \phi_k^\gamma). \tag{3.16}$$

These sets span four spaces denoted by $\mathcal{W}^0(K)$, $\mathcal{W}^1(K)$, $\mathcal{W}^2(K)$ and $\mathcal{W}^3(K)$, respectively. Fundamental properties of (3.13)–(3.16) are associated with the "nodes", "edges", and "faces" of K. The "point" mass of the scalar functions in (3.13) is

$$\int_K W_{ijk}^{\alpha\beta\gamma}(\mathbf{x}) \cdot \delta(\mathbf{x}^{\kappa\mu\nu}) d\mathbf{x} = \begin{cases} 1 & \text{if } \mathbf{x}^{\kappa\mu\nu} = \mathbf{x}^{\alpha\beta\gamma} \\ 0 & \text{at all other nodes} \end{cases}.$$

Thus, $\mathcal{W}^0(K)$ is "nodal" space with basis $\{W_{ijk}^{\alpha\beta\gamma}\}$. Circulations of the vector fields in (3.14) are

$$\int_{\mathbf{x}_{st}^{\kappa\mu}} W_{ij}^{\alpha\beta}(\mathbf{x}) \cdot \mathbf{t} dl = \begin{cases} 1 & \text{if } \mathbf{x}_{st}^{\kappa\mu} = \mathbf{x}_{ij}^{\alpha\beta} \\ 0 & \text{along all other edges} \end{cases}.$$

so we call $W_{ij}^{\alpha\beta}$ "edge" basis and $\mathcal{W}^1(K)$ - edge space. The vector fields in (3.15) have similar property with respect to their fluxes across the faces of K:

$$\int_{\mathbf{x}_s^\kappa} W_i^\alpha(\mathbf{x}) \cdot \mathbf{n} dS = \begin{cases} 1 & \text{if } \mathbf{x}_s^\kappa = \mathbf{x}_i^\alpha \\ 0 & \text{all other faces} \end{cases}.$$

Thus, W_i^α is "face" basis and $\mathcal{W}^2(K)$ is a face space. Lastly,

$$\int_K W(\mathbf{x}) d\mathbf{x} = 1.$$

so W is a "volume" basis and $\mathcal{W}^3(K)$ is volume space. Degrees of freedom (DOF) for $W^0(K)$ are "point masses", or simply the nodal values of a scalar function. DOFs for $W^1(K)$ are circulations of a vector field along the edges of K, DOF's for $W^2(K)$ are fluxes across the faces, and the DOF for $W^3(K)$ is the total mass of K for a given scalar density function.

To show that $\mathcal{W}^i(K)$ form an exact sequence on K recall that

$$\nabla \times (u\mathbf{V}) = u\nabla \times \mathbf{V} + \nabla u \times \mathbf{V}, \tag{3.17}$$

$$\nabla \cdot (u\mathbf{V}) = \nabla u \cdot \mathbf{V} + u\nabla \cdot \mathbf{V}, \tag{3.18}$$

$$\nabla \cdot (\nabla f \times \nabla g) = 0 \tag{3.19}$$

for smooth vectors fields $\mathbf{U}$, $\mathbf{V}$ and scalar function u. Using the chain rule, (3.17)–(3.19) and definitions of $W_{ijk}^{\alpha\beta\gamma}$, $W_{ij}^{\alpha\beta}$, W_i^α and W gives

$$\begin{aligned}
\nabla W_{ijk}^{\alpha\beta\gamma} &= \sigma_{ij} W_{ij}^{\alpha\beta} + \sigma_{jk} W_{jk}^{\beta\gamma} + \sigma_{ki} W_{ki}^{\gamma\alpha} \\
\nabla \times W_{ij}^{\alpha\beta} &= \sigma_i W_i^\alpha + \sigma_j W_j^\beta \\
\nabla \cdot W_i^\alpha &= \sigma W,
\end{aligned}$$

where σ_{ij}, σ_i and σ take the values ± 1. It follows that $\mathcal{W}^i(K)$ is exact sequence, i.e.,

$$\mathcal{W}^0(K) \quad \overset{\nabla}{\longmapsto} \quad \mathcal{W}^1(K) \quad \overset{\nabla\times}{\longmapsto} \quad \mathcal{W}^2(K) \quad \overset{\nabla\cdot}{\longmapsto} \quad \mathcal{W}^3(K). \tag{3.20}$$

Using (3.10) in (3.13)-(3.16) yields explicit formulae for the basis functions on $\hat{K}$:

$$\begin{aligned}
\hat{W}_{ijk}^{\alpha\beta\gamma} &= \frac{1}{8}(1+\alpha\xi_i)(1+\beta\xi_j)(1+\gamma\xi_k) \\
\hat{W}_{ij}^{\alpha\beta} &= \frac{1}{8}(1+\alpha\xi_i)(1+\beta\xi_j)(V_i \times V_j)/\det J_F \\
\hat{W}_i^\alpha &= \frac{1}{8}(1+\alpha\xi_i)V_i/\det J_F \\
\hat{W} &= \frac{1}{8}V_i \cdot (V_j \times V_k)/\det J_F
\end{aligned}$$

3.2. Exact sequence on hexahedral grids. For the magnetic diffusion application we are mainly interested in standard isoparametric hexahedral grids. Such grids consist of convex, nondegenerate hexahedrals K with vertices $\mathbf{x}^{\alpha\beta\gamma}$, $\alpha,\beta,\gamma = \pm 1$. In this case

$$F_K(\boldsymbol{\xi}) = \sum_{\alpha\beta\gamma=\pm 1} \mathbf{x}^{\alpha\beta\gamma} \hat{W}_{ijk}^{\alpha\beta\gamma}(\boldsymbol{\xi}). \tag{3.21}$$

is the unique mapping between $\hat{K}$ and a given element K. Note that F_K is a linear combination of the nodal basis functions $\hat{W}_{ijk}^{\alpha\beta\gamma}(\boldsymbol{\xi})$ on the reference element. Therefore, F_K is an incomplete cubic polynomial (a trilinear function), whose restrictions to the faces and the edges are bilinear and linear polynomials, respectively.

Let $\mathcal{N}$, $\vec{\mathcal{E}}$, $\vec{\mathcal{F}}$ and $\mathcal{K}$ denote the sets of all nodes, oriented edges and faces, and hexahedrals in the grid. Furthermore, for $K \in \mathcal{K}$ let $\mathcal{W}^l(K)$ denote the exact sequence induced by the mapping (3.21) on this element. To form an exact sequence $\mathcal{W}^l(\Omega)$ on the hexahedral grid we introduce four sets of functions parametrized by $\mathcal{N}$, $\vec{\mathcal{E}}$, $\vec{\mathcal{F}}$

and $\mathcal{K}$, and such that

$$\int_{\Omega} W_{\mathcal{N}_i}(\mathbf{x}) \cdot \delta(\mathcal{N}_j) d\mathbf{x} = \delta_{ij}; \qquad W_{\mathcal{N}_i|_K} \in \mathcal{W}^0(K)$$
$$\int_{\vec{\mathcal{E}}_j} W_{\vec{\mathcal{E}}_i}(\mathbf{x}) \cdot \mathbf{t} dl = \delta_{ij}; \qquad W_{\vec{\mathcal{E}}_i|_K} \in \mathcal{W}^1(K)$$
$$\int_{\vec{\mathcal{F}}_j} W_{\vec{\mathcal{F}}_i}(\mathbf{x}) \cdot \mathbf{n} dS = \delta_{ij}; \qquad W_{\vec{\mathcal{F}}_i|_K} \in \mathcal{W}^2(K)$$
$$\int_{\mathcal{K}_j} W_{\mathcal{K}_i}(\mathbf{x}) d\mathbf{x} = \delta_{ij}; \qquad W_{\mathcal{K}_i|_K} \in \mathcal{W}^3(K)$$

The sets $\{W_{\mathcal{N}}\}$, $\{W_{\vec{\mathcal{E}}}\}$, $\{W_{\vec{\mathcal{F}}}\}$, and $\{W_{\mathcal{K}}\}$ span the spaces $\mathcal{W}^i(\Omega)$.

The space $\mathcal{W}^0(\Omega)$ is $H(\Omega, \mathrm{grad})$ conforming because it contains continuous functions. Definition of $W_{\vec{\mathcal{E}}}$ and Eq. (3.11) imply that $\mathcal{W}^1(\Omega)$ contains vector fields that are tangentially continuous along the edges in $\vec{\mathcal{E}}$. Therefore, this space is $H(\Omega, \mathbf{curl})$ conforming. Likewise, $\mathcal{W}^2(\Omega)$ contains fields that are normally continuous across the faces $\vec{\mathcal{F}}$. This makes $\mathcal{W}^2(\Omega)$ $H(\Omega, \mathrm{div})$ conforming. Clearly, $\mathcal{W}^3(\Omega) \subset L^2(\Omega)$. Exactness of this sequence follows easily from the exactness of the element spaces $\mathcal{W}^i(K)$.

3.3. Exact sequence on quadrilateral grids in 2-D. It suffices to construct an exact sequence for one generalized quadrilateral. Then, spaces on quadrilateral grids can be formed as in the three-dimensional case.

We consider the open square $\hat{K} = (-1,1)^2$ in the reference frame $\boldsymbol{\xi} = (\xi_1, \xi_2)$ and a smooth mapping $F : \mathbf{R}^2 \mapsto \mathbf{R}^2$. Next we imbed K into the *virtual* generalized hexahedral

$$\tilde{K} = \{\mathbf{x} \,|\, (x_1, x_2) \in K, -1 < x_3 < 1\}.$$

Let $\tilde{\mathcal{W}}^i$ denote an exact sequence defined on $\tilde{K}$. Since the virtual hexahedral is image of $(-1,1)^3$ under the mapping $\tilde{F} = (F, \xi_3)$,

$$V_3 = \mathbf{k} \quad \text{and} \quad \nabla G_3 = \mathbf{k}.$$

As a result, (3.10) specialize to

$$\begin{aligned} \nabla G_1 &= (V_2 \times \mathbf{k}) / \det J_F \\ \nabla G_2 &= (\mathbf{k} \times V_1) / \det J_F. \end{aligned}$$

Inserting these expressions in (3.13)–(3.16) yields after some manipulation four pairs of basis function sets on K and $\hat{K}$:

$$\begin{array}{lcllcl} W_{ij*}^{\alpha\beta*} & = & \phi_i^\alpha \phi_j^\beta; & \hat{W}_{ij*}^{\alpha\beta*} & = & \frac{1}{4}(1+\alpha\xi_i)(1+\beta\xi_j) \\ W_{ij}^{\alpha*} & = & \phi_i^\alpha \nabla \phi_j^\beta; & \hat{W}_{ij}^{\alpha*} & = & \frac{1}{4}(1+\alpha\xi_i)(V_j \times \mathbf{k})/\det J_F \\ W_i^\alpha & = & \phi_i^\alpha (\nabla \phi_j^\beta \times \mathbf{k}); & \hat{W}_i^\alpha & = & \frac{1}{4}(1+\alpha\xi_i) V_i / \det J_F; \\ W & = & \nabla \phi_i^\alpha \cdot (\nabla \phi_j^\beta \times \dfrac{\mathbf{k}}{2}); & \hat{W} & = & \frac{1}{4} V_i \cdot (V_j \times V_k)/\det J_F = \frac{1}{4} \end{array}$$

The two-dimensional complex $\mathcal{W}^i(K)$ is defined by taking the spans of each basis set in K. By the chain rule

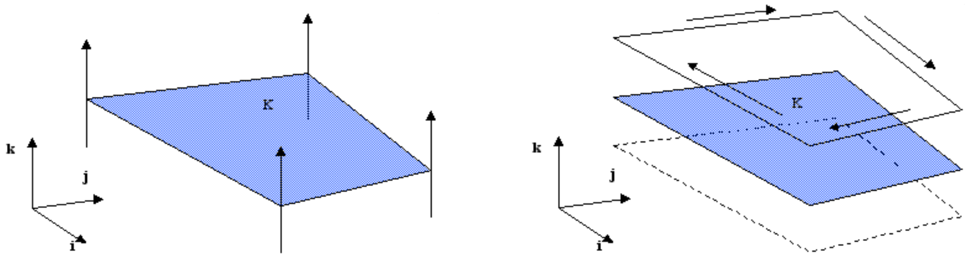

FIG. 3.1. *Virtual (perpendicular) and parallel edges on $\tilde{K}$.*

$$\nabla W_{ij*}^{\alpha\beta *} = \phi_i^\alpha \nabla \phi_j^\beta + \phi_j^\beta \nabla \phi_i^\alpha$$

which is a sum of $\mathcal{W}^1(K)$ basis functions, and

$$\nabla \cdot W_i^\alpha = \phi_i^\alpha (\nabla \phi_j^\beta \times \mathbf{k})$$

which is a $\mathcal{W}^3(K)$ function. Therefore $\nabla \mathcal{W}^0(K) \subset \mathcal{W}^1(K)$ and $\nabla \cdot \mathcal{W}^2(K) \subset \mathcal{W}^3(K)$. Showing the curl inclusion is somewhat more involved as it splits into two relations. This corresponds to the two possibilities of restricting curls[2] to a plane. The first way is to apply the curl to vector fields perpendicular to the plane and set

$$\nabla \times \phi := \nabla \times (\phi \mathbf{k}) = \nabla \phi \times \mathbf{k} = \phi_y \mathbf{i} - \phi_x \mathbf{j}. \tag{3.22}$$

The virtual hehaxedral $\tilde{K}$ has four vertical edges; see Fig.3.1. The 3D edge basis functions associated with these edges are

$$W_{ij}^{\alpha\beta} = \phi_i^\alpha \phi_j^\beta \frac{\mathbf{k}}{2} = \frac{1}{2} W_{ij*}^{\alpha\beta *}\mathbf{k}.$$

Therefore, $\nabla \times W_{ij}^{\alpha\beta}$ gives the two-dimensional curl of the two-dimensional *nodal* function $W_{ij}^{\alpha\beta *}$. On the other hand,

$$\begin{aligned}\nabla \times W_{ij}^{\alpha\beta} &= \frac{1}{2}\nabla \times \left(W_{ij*}^{\alpha\beta *}\mathbf{k}\right) = \frac{1}{2}\Big(\phi_i^\alpha \left(\nabla \phi_j^\beta \times \mathbf{k}\right) + \phi_j^\beta \left(\nabla \phi_i^\alpha \times \mathbf{k}\right)\Big) \\ &= \frac{1}{2}(W_i^\alpha - W_j^\beta)\end{aligned}$$

which establishes the inclusion $\nabla \times \mathcal{W}^0(K) \subset \mathcal{W}^2(K)$.

The second way is to restrict the curl to planar vectors. The result is identified with a scalar field according to

$$\nabla \times \mathbf{u} := \nabla \times (u_1 \mathbf{i} + u_2 \mathbf{j}) = (u_{2x} - u_{1y})\mathbf{k}. \tag{3.23}$$

The virtual hexahedral has two pairs of edges parallel to K, see Fig.3.1. The 3D edge basis functions for the edges on the top face (where $\phi_3^+ = 1$) are

$$W_{i3}^{\alpha +} = \phi_i^\alpha \phi_3^+ \nabla \phi_j^\beta = \phi_i^\alpha \nabla \phi_j^\beta = W_{ij}^{\alpha *}.$$

[2] In the literature the operators engendered by the restriction procedure are sometimes denoted by *rot* and *curl*, respectively. Here we employ the same symbol for both operators in order to emphasize that they are merely restrictions of the same three-dimensional operator.

Therefore, $\nabla \times W_{i3}^{\alpha +}$ gives the two-dimensional curl of the two-dimensional *edge* basis function $W_{ij}^{\alpha *}$. Since

$$\nabla \times \left[W_{23}^{\alpha +} \right]_{\phi_3^+ = 1} = \nabla \times \left(\phi_i^\alpha \nabla \phi_j^\beta \right) = \nabla \phi_i^\alpha \times \nabla \phi_j^\beta .$$

this establishes the inclusion $\mathcal{W}^1(K) \subset \mathcal{W}^3(K)$. The two-dimensional exactness structure is summarized in (3.24)

$$\boxed{\begin{array}{ccccccc} \mathcal{W}^1 & \xleftarrow{\nabla} & \mathcal{W}^0 & \xmapsto{\nabla\times} & \mathcal{W}^2 & \xmapsto{\nabla\cdot} & \mathcal{W}^3 \\ \mathcal{W}^1 & & & \xmapsto{\nabla\times} & & & \mathcal{W}^3 \end{array}} \tag{3.24}$$

4. Transient magnetics solution using the exact sequence. For the magnetic diffusion application considered here we are interested in divergence free approximations of the magnetic induction $\mathbf{B}$. To accomplish this $\mathbf{H}$ and $\mathbf{J}$ are eliminated from the system by (2.7)-(2.8) and the exact sequence $\mathcal{W}^i$ is used on the Faraday's side of Tonti's diagram (3.8)

(4.1)

	Ampere		*Faraday*	
$\mathcal{W}^2$	$\frac{1}{\mu}\mathbf{B}_h$	$\dots$	$\mathbf{B}_h$	$\mathcal{W}^2$
$\nabla\times$	$\vdots$		$\Uparrow$	$\nabla\times$
$\mathcal{W}^1$	$\sigma\mathbf{E}_h$	$\dots$	$\mathbf{E}_h$	$\mathcal{W}^1$

The finite element model that corresponds to Diagram 4.1 is

$$\nabla \times \frac{1}{\mu}\mathbf{B}_h = \sigma \mathbf{E}_h \quad \text{in } \Omega \tag{4.2}$$

$$\nabla \times \mathbf{E}_h = -\frac{\partial \mathbf{B}_h}{\partial t} \quad \text{in } \Omega \tag{4.3}$$

where

$$\mathbf{B}_h = \sum_{\vec{\mathcal{F}}} \Phi_{\vec{\mathcal{F}}}(t) W_{\vec{\mathcal{F}}}; \qquad \mathbf{E}_h = \sum_{\vec{\mathcal{E}}} C_{\vec{\mathcal{E}}}(t) W_{\vec{\mathcal{E}}},$$

are expansions of $\mathbf{E}_h$ and $\mathbf{B}_h$ in terms of edge and face basis functions. The proper boundary conditions for this formulation are

$$\mathbf{n} \times \mathbf{E}_h = \mathbf{n} \times \mathbf{E}_b \quad \text{on Type I;} \qquad \mathbf{n} \times \frac{1}{\mu}\mathbf{B}_h = \mathbf{n} \times \mathbf{H}_b \quad \text{on Type II.} \tag{4.4}$$

System (4.2)-(4.3) and (4.4) requires proper interpretation. The discrete Faraday law (4.3) holds exactly thanks to the inclusion $\nabla \times \mathcal{W}^1 \subset \mathcal{W}^2$. The Ampere's theorem (4.2) and the boundary condition on Type II segments are, in contrast, interpreted as a weak equation

$$\int_\Omega \frac{1}{\mu}\mathbf{B}_h \cdot \nabla \times \hat{\mathbf{E}}_h d\Omega + \int_\Gamma (\mathbf{n} \times \mathbf{H}_\mathrm{b}) \cdot \hat{\mathbf{E}}_h d\Gamma = \int_\Omega \sigma \mathbf{E}_h \cdot \hat{\mathbf{E}}_h d\Omega \qquad \forall \hat{\mathbf{E}}_h \in \mathcal{W}^1 \tag{4.5}$$

in which tangential magnetic field appears as natural boundary condition. The fully discrete system is then derived by replacing the time derivative by a finite difference. The ensuing algebraic system for $\mathbf{E}_h^{n+1}$ and $\mathbf{B}_h^{n+1}$ is

$$\int_\Omega \sigma \mathbf{E}_h^{n+1} \cdot \hat{\mathbf{E}}_h - \frac{1}{\mu}\mathbf{B}_h^{n+1} \cdot \nabla \times \hat{\mathbf{E}}_h d\Omega = \int_\Gamma (\mathbf{n} \times \mathbf{H}_\mathrm{b}) \cdot \hat{\mathbf{E}}_h d\Gamma \qquad \forall \hat{\mathbf{E}}_h \in \mathcal{W}^1 \tag{4.6}$$

$$-\frac{\mathbf{B}_h^{n+1} - \mathbf{B}_h^n}{\Delta t} = \nabla \times \mathbf{E}_h^{n+1}. \tag{4.7}$$

The fully discrete equations combine conventional Galerkin formulation for (4.6) with a finite-volume like form of the discrete Faraday's law (4.7). However, (4.7) is not a bona-fide finite volume scheme because it is based on functional representation of the fields rather than a discrete set of values. Methods of this kind for exact sequences on tetrahedral grids (Whitney elements) were introduced by Bossavit and Verite in [7]. They considered formulation in $\mathbf{H}$ and $\mathbf{J}$ in which discrete Ampere's theorem is satisfied exactly, while the Faraday's law holds weakly.

To solve (4.6)-(4.7) we proceed as follows. Because $\nabla \times \mathbf{E}_h^{n+1}$ is in $\mathcal{W}^2$ the second equation can be solved exactly for $\mathbf{B}_h^{n+1}$:

$$\mathbf{B}_h^{n+1} = \mathbf{B}_h^n - \Delta t \nabla \times \mathbf{E}_h^{n+1}.$$

This expression is substituted in (4.6) to obtain an equation in terms of $\mathbf{E}_h^{n+1}$:

$$\begin{aligned}&\int_\Omega \sigma \mathbf{E}_h^{n+1} \cdot \hat{\mathbf{E}}_h + \frac{\Delta t}{\mu}\Big(\nabla \times \mathbf{E}_h^{n+1}\Big) \cdot \Big(\nabla \times \hat{\mathbf{E}}_h\Big) d\Omega \\ &= \int_\Omega \frac{1}{\mu}\mathbf{B}_h^n \cdot \Big(\nabla \times \hat{\mathbf{E}}_h\Big) d\Omega + \int_\Gamma \Big(\mathbf{n} \times \mathbf{H}_\mathrm{b}\Big) \cdot \hat{\mathbf{E}}_h d\Gamma \qquad \forall \hat{\mathbf{E}}_h \in \mathcal{W}^1.\end{aligned} \tag{4.8}$$

This scheme has very attractive computational properties. First, it ensures that the approximate magnetic flux density is divergenceless provided $\nabla \cdot \mathbf{B}_h^0 = 0$. This can be accomplished by setting $\mathbf{B}_h^0 = \nabla \times \mathbf{A}_h^0$ for some potential $\mathbf{A}_h^0 \in \mathcal{W}^1$. Second, it allows imposition of Type I and Type II boundary conditions in a simple and efficient manner. For the formulation considered here, tangential $\mathbf{E}$ are essential boundary conditions and tangential $\mathbf{H}$ are natural boundary conditions. Because the degrees of freedom for $\mathbf{E}$ are the circulations of the electric field along the edges, the essential boundary condition is trivial to satisfy. For example, setting $\mathbf{n} \times \mathbf{E} = 0$ on Type I boundaries amounts to setting all coefficients associated with Type I edges to zero. This situation sharply contrasts with the use of nodal elements where tangential and normal boundary conditions pose a difficult problem.

5. Fast iterative solvers. Solution of the discrete linear system (4.8) is complicated by the nontrivial discrete kernel corresponding to the **curl** operator (referred to as ker(**curl**) throughout the rest of the paper). When σ is large this **curl** operator is less important and relaxation alone (i.e. without multigrid) is effective. In regions where σ is small, however, the **curl** operator dominates and the ker(**curl**) can cause difficulties for iterative methods. Any efficient preconditioner or solution technique must approximate all scales associated with the operator and so this discrete kernel must be addressed. In this section, we consider the application of a multigrid method to (4.8) and the proper treatment of the ker(**curl**).

Multigrid methods approximate the partial differential equation (PDE) of interest on a hierarchy of grids and use solution updates from coarse grids to accelerate the convergence on the finest grid. An example multilevel iteration is given in Figure 5.1 to solve

$$A_1 u = b.$$

In Figure 5.1, the $S_k()$'s are approximate solvers corresponding to pre and post

/* Solve $A_k u = b$ (k is current grid level) */
procedure multilevel(A_k, b, u, k)
  $u = S_k(A_k, b, u)$;
  if ($k \neq$ **Nlevel**)
    $\hat{r} = P_k^T(b - A_k u)$;
    $\hat{A}_{k+1} = \begin{cases} P_k^T A_k P_k \\ \text{or} \\ \text{discretized PDE on coarser mesh} \end{cases}$
    $v = 0$;
    multilevel($\hat{A}_{k+1}, \hat{r}, v, k+1$);
    $u = u + P_k v$;
    $u = S_k(A_k, b, u)$;

FIG. 5.1. *High level multigrid V cycle consisting of 'Nlevel' grids to solve $A_1 u = b$.*

smoothing. These are used to reduce high frequency errors. Once smoothed, errors can be approximated well on a coarser grid and so the linear system of equations is projected on to a coarser space via the grid transfer operator P_k. The coarse grid equations are approximately solved by recursively applying the multigrid idea. The resulting coarse grid solution is then interpolated and used to correct the fine grid solution. The two primary multigrid components are the smoothers, $S_k()$'s, and the grid transfers, P_k's. See [11, 23] for more on multigrid methods. Standard multigrid methods fall into two categories: geometric and algebraic. Geometric algorithms use a hierarchy of meshes covering the same physical domain. Usually, the grid transfers correspond to standard interpolation (e.g. linear) between the meshes and the $\hat{A}_k$'s are built by discretizing the PDE on each mesh. In contrast to geometric methods, algebraic methods use only A_1. Coarse grid meshes are constructed automatically by coarsening the matrix graph associated with A_1 and the P_k's are determined algebraically. The primary advantage of algebraic multigrid techniques is that a hierarchy of meshes and coarse grid discretizations need not be supplied.

When solving (4.8), both the smoother and the coarse grid correction must properly treat the ker(**curl**). This is because ker(**curl**) contains both high and low frequency components. We want high frequency ker(**curl**) error components reduced by the smoother and low frequency ker(**curl**) error components to be accurately represented on the next coarser grid (where they will be reduced). Within most geometric schemes, the coarse grid interpolation correctly approximates the smooth ker(**curl**). Specifically, linear interpolation applied to the discrete coarse grid kernel of the **curl** lies within the discrete fine grid kernel of the **curl**. Hence the primary multigrid task is the development of a suitable smoother. It is important to notice that the discrete ker(**curl**) exists on all levels and so the smoother on all levels must appropriately address these components. Smoothing error components that lie in the space orthogonal to ker(**curl**) operator is relatively straight-forward (e.g., standard Gauss-Seidel methods are suitable). However, high frequency error components lying in the subspace of the ker(**curl**) are poorly reduced by standard smoothers when σ is small. This is because the mass term ($\int_\Omega \sigma \mathbf{E}_h^{n+1} \cdot \hat{\mathbf{E}}_h$) in (4.8) governs the error within ker(**curl**). Most smoothers, however, do not treat the two terms of Equation (4.8) separately and thus focus only on the (**curl**, **curl**) term that dominates when σ is small. Geometric multigrid techniques addressing the smoothing issue have been proposed by Vassilevski/Wang [27], Hiptmair [16], and Arnold/Falk/Winter [1]. These methods are discussed in §5.2.

In contrast to geometric multigrid methods, algebraic multigrid techniques must also determine coarse grid spaces, and these coarse grid spaces must take into account ker(**curl**). For this reason, traditional AMG methods that have been designed for H_1 elliptic systems fail. Reitzinger and Schöberl [20] propose an algebraic method that specifically addresses equations of the form (4.8). This approach is described in §5.3. Here, the idea is to preserve the kernel of the discrete **curl** on coarser spaces. In particular, when the discrete kernel subspace associated with a coarse grid **curl** is interpolated to the fine grid, it lies within the discrete kernel subspace of the fine grid **curl**. In this paper, we pursue the Reitzinger/Schöberl approach.

5.1. Discrete Gradient. In order to explain the multigrid method, we need to discuss the discrete analog of the operator '$\nabla \times \nabla \times$' (referred to as the (**curl**, **curl**) operator) and its kernel. In continuous space it is well known that

$$\nabla \times (\nabla \phi) = 0 \qquad \phi \in H_0(\Omega, \mathbf{curl}). \tag{5.1}$$

In §3 the De Rham complex was introduced. Recall that the continuous gradient maps $H(\Omega, \mathrm{grad})$ to ker(**curl**) $\subset H(\Omega, \mathbf{curl})$ and that the continuous gradient of $\mathcal{W}^0$ exactly corresponds to ker(**curl**) in $\mathcal{W}^1$ where $\mathcal{W}^0$ and $\mathcal{W}^1$ are the finite element subspaces of $H_0(\Omega, \mathrm{grad})$ and $H_0(\Omega, \mathbf{curl})$, respectively. This implies that a matrix spanning the discrete ker(**curl**) can be constructed one column at a time by taking the gradient of each basis function in $\mathcal{W}^0$. The resulting matrix, T, is a discrete approximation to the continuous gradient operator and $T\hat{\phi}$ (where $\hat{\phi} \in \mathcal{W}^0$) is a discrete analogue of $\nabla \phi$ given in (5.1). When $\mathcal{W}^0$ corresponds to linear basis functions and Ω has Neumann boundary conditions, T is $N_{edges} \times N_{nodes}$ where N_{edges} is the number of mesh edges and N_{nodes} is the number of mesh nodes. Column (node) i has '+1' and '-1' entries for each edge (row) that has node i as an endpoint. The sign depends on the direction imposed on the edge in the edge element discretization. The null space of the discrete (**curl**, **curl**) operator has dimension $N_{nodes} - 1$ and is spanned by T [3]. It is important to note that T is developed via an algebraic (actually a matrix graph) process using just nodal connectivity information. Hence the construction can be repeated on coarser grids (see §5.4).

5.2. Smoothing. In the context of geometric multigrid, several smoothers have been proposed for problems in $H(\Omega, \mathrm{div})$ and $H(\Omega, \mathbf{curl})$ [27, 1, 16, 2]. Each of these smoothers is designed to damp both error components in ker(**curl**) and in its orthogonal complement. One such method is an overlapping Schwarz block smoother by Arnold, Falk, and Winther [1]. The central idea is to break the grid into overlapping patches of edges with one patch for each node. Patch i consists of all edges having node i as an endpoint. Arnold et. al. use block Jacobi or Gauss-Seidel smoothers based on this decomposition in conjunction with a geometric V cycle multigrid method. They

[3] When Dirichlet boundary conditions are present, the dimension of the discrete null space is smaller and is related to the number of node groups (where a group consists of nodes that are connected together via Dirichlet edges). Additionally, Dirichlet edges are omitted when forming T.

Perform symmetric Gauss-Seidel on $A^{(e)}v^{(e)} = f^{(e)}$.
Calculate residual $r^{(e)} = f^{(e)} - A^{(e)}v^{(e)}$.
Transfer edge residual to nodes: $f^{(n)} = T^T r^{(e)}$.
Perform symmetric Gauss-Seidel on $T^T A^{(e)} T v^{(n)} = f^{(n)}$ with zero initial guess.
Update edge-based solution: $v^{(e)} = v^{(e)} + T v^{(n)}$.

FIG. 5.2. *Distributed relaxation algorithm applied to $A^{(e)}$ (Vasselevski/Wang, Hiptmair).*

prove that the convergence of the resulting algorithm is independent of the number of mesh points and invariant with respect to the material properties such as conductivity and permeability.

Another effective smoother can be viewed as a special case of distributed relaxation first proposed by Brandt [9]. This form of distributed relaxation was considered for a divergence equation (arising from mixed finite elements) in 2D by Vassilevski and Wang [27] and extended to 3D and for Maxwell's equations by Hiptmair [16]. The central idea is to explicitly smooth on both $\ker(\mathbf{curl})$ and on its orthogonal complement. The smoothing algorithm proposed by Hiptmair is given in Figure 5.2. In the first stage the smoother relaxes on the entire space. When σ is small, the smoother effectively focuses on error components that are in $\ker(\mathbf{curl})^{\perp}$. In the second stage the smoother relaxes error components that are in $\ker(\mathbf{curl})$. This is done by using T and T^T to project the system of equations into $\ker(\mathbf{curl})$. In both stages the distributed relaxation uses one step of symmetric Gauss-Seidel. Using this smoother in conjunction with a geometric V cycle multigrid algorithm, Hiptmair also proves convergence independent of the number of mesh points and invariance to material properties.

5.3. Algebraic Coarsening. In order to address Z-pinch simulations within the ALEGRA framework, a multigrid linear solver must function with highly irregular unstructured meshes and highly heterogenous material properties. Schemes restricted to either regular meshes or to meshes that are refinements of coarser grids are not desirable. We pursue algebraic multigrid methods as they free application users from grid hierarchy requirements and free finite element developers from constructing complex operators as a prerequisite for applying multigrid. Unfortunately, however, the proper handling of the low frequency $\ker(\mathbf{curl})$ subspace is quite complicated within algebraic methods. In particular, standard algebraic multigrid techniques will fail as the coarse grid correction will not adequately damp low frequency error components in $\ker(\mathbf{curl})$. In contrast to algebraic methods, typical geometric schemes automatically handle the coarse grid $\ker(\mathbf{curl})$ properly.

The key idea to properly capturing the $\ker(\mathbf{curl})$ on coarse grids is to work with nodal basis functions. In particular, the De Rham complex tells us that the $\ker(\mathbf{curl})$ can be obtained by taking the gradient of nodal basis functions. Thus, if we take nodal basis functions corresponding to the fine grid mesh, coarsen them, and then take their discrete gradient we can properly capture the low frequency $\ker(\mathbf{curl})$ space. An overview of the multigrid hierarchy construction follows. A hierarchy of nodal discretization matrices is created by doing unsmoothed aggregation on a closely related nodal problem. Using meshes defined by the nodal hierarchy, an edge based multigrid hierarchy is developed: inter-grid transfer operators, coarse grid discretizations, and coarse grid discrete gradients. The nodal discretization matrices are then discarded, and the fine grid edge based problem is solved with CG preconditioned by AMG using Hiptmair's implementation of Brandt's distributed relaxation as the smoother on all levels. The main idea is capturing the null space of the $(\mathbf{curl}\,, \mathbf{curl}\,)$ operator on each of the coarser levels. By choosing an appropriate interpolation operator, Reitzinger and Schöberl [20] show that each coarse level gradient prolongates to a fine level gradient, i.e., into the null space of the $(\mathbf{curl}\,, \mathbf{curl}\,)$ operator on the fine grid. We now discuss the individual steps in more detail.

The first step is to build a multigrid hierarchy for a related PDE problem that is discretized using nodal piecewise linear FE basis functions. Reitzinger and Schöberl advocate using the related PDE problem:

$$\int \frac{\Delta t}{\mu} \nabla u \cdot \nabla v + \int \sigma u \cdot v. \tag{5.2}$$

Note that the coefficients of this problem are the same as those in (4.8). Building the multigrid hierarchy consists of two primary steps: coarsening the matrix graph and building the interpolation operator. Specifically, an undirected graph, $\mathcal{G}$, is constructed from the discrete matrix $A_1^{(n)}$ associated with (5.2). The number of graph vertices is equal to the number of matrix equations and an undirected edge between node i and j is added if and only if the upper triangular matrix entry $A_1^{(n)}(i,j)$ is nonzero. This matrix graph can then be coarsened by any one of several aggregation techniques. Typically, these schemes work incrementally by creating one new aggregate at a time. A new aggregate is defined by taking an unaggregated root node and grouping it with its immediate neighbors. To encourage aggregates to be approximately the same size, several heuristics are applied to 'clean up' aggregates and to choose unaggregated root nodes wisely [26, 25, 24]. Additional heuristics are used to ignore 'weak' matrix couplings (e.g. $|a(i,j)| \ll max\{|a(i,i)|, |a(j,j)|\}$) during the coarsening phase. Thus, the inclusion of the coefficients σ and $\frac{\Delta t}{\mu}$ in (5.2) gives the aggregation scheme the opportunity to detect coefficient jumps when coarsening. The aggregates can now be thought of as coarse mesh points.

Once the aggregates are created, a grid transfer operator $P_1^{(n)}$ between the coarse and fine mesh points is constructed. $P_1^{(n)}$ corresponds to piecewise constant interpolation and is given by

$$P_1^{(n)}(i,j) = \begin{cases} 1 & \text{if } j \text{ is in aggregate } i \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

A "coarse" discretization matrix is then defined by a Galerkin approach:

$$A_2^{(n)} = (P_1^{(n)})^{\mathrm{T}} A_1^{(n)} P_1^{(n)}. \tag{5.4}$$

The matrix (5.4) can be thought of as an adjacency matrix and so defines a "coarse" mesh. This process of unsmoothed aggregation can be applied recursively to build a hierarchy of grid transfer matrices, $P_1^{(n)}, \ldots, P_k^{(n)}$, and discretization matrices, $A_1^{(n)}, \ldots, A_k^{(n)}$, corresponding to a non-nested mesh hierarchy.

After the nodal mesh hierarchy has been created, the next step is to define a sequence of edge based interpolation operators, $P_1^{(e)}, \ldots, P_k^{(e)}$, based on this hierarchy. The hierarchy of edge based matrices is the one that is actually used in the multigrid iterations. If defined properly, the prolongation operator $P_k^{(e)}$ should interpolate the discrete gradient of coarse grid nodal basis functions into the $\ker(\mathbf{curl}\,)$ on fine grids. When used with a Galerkin approach, this guarantees that the discrete gradient of

$$\begin{tikzcd}
\mathcal{W}_{0,h} \arrow[r, "\nabla"] & \ker(\mathbf{curl}_h) \\
\mathcal{W}_{0,H} \arrow[r, "\nabla"] \arrow[u, "P_k^{(n)}"] & \ker(\mathbf{curl}_H) \arrow[u, "P_k^{(e)}"']
\end{tikzcd}$$

FIG. 5.3. *Commuting diagram for two levels.*

coarse grid nodal functions are in the coarse grid approximation to $\ker(\mathbf{curl})$. As shown in [20], this is accomplished if

$$\nabla_h(P_k^{(n)}\phi_H) = P_k^{(e)}(\nabla_H\phi_H). \tag{5.5}$$

where ϕ_H is a coarse level nodal basis function and ∇_h (∇_H) is the discrete gradient on the fine (coarse) grid. In effect, proper construction of $P_k^{(e)}$ ensures the diagram in Figure 5.3 commutes where h and H are used to denote fine and coarse grid spaces.

To define the interpolation operator, we first consider the mapping $\mathbf{agg}$: nodes $\rightarrow$ aggregates by

$$\mathbf{agg}(i) = \begin{cases} j & \text{if } i \text{ belongs to aggregate } j \\ 0 & \text{otherwise.} \end{cases}$$

$P_k^{(e)}$ is a rectangular matrix that maps coarse grid edges, $e_2 = (i_2, j_2)$, to fine grid edges, $e_1 = (i_1, j_1)$ where $P_k^{(e)}(e_1, e_2)$ is given by

$$P_k^{(e)}(e_1, e_2) = \begin{cases} 1 & \text{if } (i_2, j_2) = (\mathbf{agg}(i_1), \mathbf{agg}(j_1)) \\ -1 & \text{if } (i_2, j_2) = (\mathbf{agg}(j_1), \mathbf{agg}(i_1)) \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

Essentially, the prolongator $P_k^{(e)}$ is piecewise constant. A value is interpolated from a coarse grid edge (i, j) to each fine grid edge that connects the two aggregates corresponding to coarse nodes i and j.No values are interpolated to fine edges whose endpoints are in the same aggregate. The interpolation process is illustrated in Figure 5.4 for one coarse edge. The fine grid mesh is given by straight solid lines, and nodal aggregates are denoted by dashed lines. A coarse grid edge that connects two coarse grid nodes (aggregates) and that has value c is represented by a curving solid line. Finally, the edge based coarse grid matrix is defined with a Galerkin approach:

$$A_{k+1}^{(e)} = (P_k^{(e)})^{\mathrm{T}} A_k^{(e)} P_k^{(e)}.$$

The key idea is to coarsen the nodal graph and then define nodal basis functions that are piecewise constants. The discrete gradient of a piecewise constant function defined over aggregate j is a function ψ_j that is nonzero only at the interface between aggregate j and neighboring aggregates. The exact interpolation of ψ_j is then insured by (5.6). Alternatively, we can view the algorithm as a way of coarsening the fine grid null space. We can coarsen the null space by summing columns of T associated with nodes in an aggregate. Recall that each edge in T contains a '+1' and '−1' entry associated with the edge's endpoints. Thus, the resulting 'summed' null space

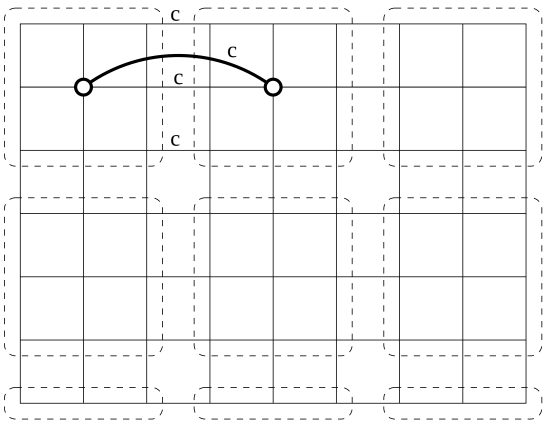

FIG. 5.4. *Example of edge based interpolation. Coarse grid edge values are interpolated only to fine edges passing between aggregates.*

vector for aggregate j is nonzero only at the interface between aggregate j and other aggregates. Once coarsened, each null space vector, ψ_j, is defined as a sum of local basis functions

$$\psi_j = \sum_{i=1,\ldots,N} \phi_{ij}$$

where ϕ_{ij} has support only at the interface between aggregates i and j, and N is the number of neighboring aggregates. These local basis functions essentially form the columns of $P_k^{(e)}$. This alternative view of the method is closely related to the smoothed aggregation multigrid method[26, 25]. In this scheme, the operator's null space[4] is partitioned over local basis functions associated with aggregates and these local basis functions form an initial prolongation operator. A key improvement in smoothed aggregation is that this initial prolongation is enhanced via a smoothing step. Without this smoothing step it has been shown that the convergence is not independent of the number of mesh points. Thus, we should not expect the use of $P_k^{(e)}$ to yield a multigrid method that converges independent of the number of mesh points. We are currently experimenting with applying a smoothing step to improve $P_k^{(e)}$. This modification follows standard smoothed aggregation and uses

$$\tilde{P}_k^{(e)} = (I - \alpha D^{-1} A_k^{(e)}) P_k^{(e)} \tag{5.7}$$

where $\alpha = \frac{4}{3}\lambda_{max}$, $D = diag(A_k^{(e)}$, and λ_{max} is obtained by applying a couple of eigenvalue iterations to $D^{-1/2} A_k^{(e)} D^{-1/2}$. This technique has not been fully implemented and the idea will be pursued in detail in a future paper. More information on smoothed aggregation can be found in [26, 25].

5.4. Implementation. The edge element algebraic multigrid preconditioner is implemented in the ML package [22], an AMG package intended for distributed memory computers. This package requires users to furnish vectors and matrices. Matrices

[4]Smoothed aggregation is normally applied to problems with a small global null space (e.g. in elasticity the null space corresponds to six rigid body modes: rotations and translations in three dimensions). Thus, no coarsening of the null space is needed.

are supplied by providing size information, a matrix-vector product, and a *getrow* function (used to obtain nonzeros and column numbers within a single row). The ML package runs on distributed memory machines. Parallelism is achieved by assigning a subset of rows for each matrix to different processors. The ML package already contains the smoothed aggregation multigrid method[26, 25] and many of the needed kernels: parallel matrix-matrix multiply, a variety of parallel smoothers (damped Jacobi, symmetric processor-block Gauss-Seidel [5], block symmetric processor-block Gauss-Seidel, etc.) and a coarse direct solver. Additionally, the ML package is designed to facilitate the use of other software packages. ML's existing smoothed aggregation multigrid method (with smoothing disabled) is used to generate the complete nodal multigrid hierarchy: $P_k^{(n)}$'s and $A_k^{(n)}$'s. The fine grid nodal matrix, $A_1^{(n)}$, is a discrete Laplace operator and is constructed by setting $A_1^{(n)}(i,j)$ to '-1' for each (i,j) corresponding to a mesh edge. The matrix diagonal is then chosen so that the sum of matrix entries within a row is zero. In the future, we will replace the discrete Laplacian with an approximation to (5.2) so that our aggregation scheme can detect coefficient jumps. The coarse grid $A_k^{(n)}$'s are then used to generate the coarse grid T_k's. Specifically, on level $k > 1$ each undirected edge $A_k^{(n)}(i,j)$ is assigned a unique number: $1 \le \tilde{e} \le N_{edges}$ where N_{edges} is the total number of undirected edges. Then,

$$\begin{array}{rcll} T_k(\tilde{e}, i) & = & 1 & (\text{or } -1 \text{ if } j > i) \\ T_k(\tilde{e}, j) & = & -1 & (\text{or } 1 \text{ if } j > i). \end{array}$$

Finally, the edge-element grid transfers, $P_k^{(e)}$'s, are obtained by performing a matrix triple product

$$\hat{P}_k^{(e)} = T_k P_k^{(n)} T_{k+1}^T \tag{5.8}$$

and culling entries:

$$P_k^{(e)}(i,j) = \begin{cases} 1 & \text{if } \hat{P}_k^{(e)}(i,j) = 2 \\ -1 & \text{if } \hat{P}_k^{(e)}(i,j) = -2 \\ 0 & \text{otherwise} \end{cases} . \tag{5.9}$$

This corresponds exactly to (5.6) and allowed us to implement the $P_k^{(e)}$ construction quickly via existing ML kernels.

Most of the parallel issues are handled by ML's existing parallel kernels. Two exceptions are the formulation of the coarse grid discrete gradients and a matrix free version of Hiptmair's smoother. Both required new parallel code. The distributed relaxation algorithm (5.2) is implemented in two different ways. In one implementation, the matrix products $T_k^T A_k^{(e)} T_k$ are calculated in a preprocessing step for each level. This allows the use of ML's fast parallel matrix kernels. In the other implementation, the smoother application is matrix free [20]. The nodal space projection and update to the edge based solution is done node by node, so that the triple matrix product is never formed.

6. Numerical studies. To deliver usable computations for the Z-pinch simulations, the fidelity and scalability of the solvers must be tested for realistic test

[5] Processor-block means that each processor performs Gauss-Seidel locally and uses off-processor information corresponding to the previous iteration.

problems. In particular the conductivity may vary over many orders of magnitude and it is important to understand how this will affect not only the representation of the solution but also the requirements for iterative solution technology. The linear system (4.8) can be represented in matrix form as

$$\left[\sigma M + \frac{\Delta t}{\mu} K\right] x = b. \tag{6.1}$$

The scaling of the M and K matrices with respect to the element length scale, h, goes as h^3 and h respectively. Thus we obtain

$$\left[\sigma h^3 \hat{M} + \frac{\Delta t}{\mu} h \hat{K}\right] x = b \tag{6.2}$$

where $\hat{M}$ and $\hat{K}$ contain entries of $O(1)$ size. Let c represent a typical sound speed or velocity in the problem of interest. We expect in general for the time step to be limited by the hydrodynamic Courant scales so that $\Delta t \sim h/c$. Thus we can define the mesh magnetic Reynolds number

$$R_m = \mu\sigma c h \tag{6.3}$$

If R_m is large then the linear system is mass matrix dominated and diffusion times are slower than hydrodynamic propagation times. If R_m is small then we are in a diffusion dominated region. It is possible to model regions containing no mass using a very small pseudo conductivity in order to propagate the field within the magnetoquasistatic approximation of magnetohydrodynamics which implies in MKS units that $\epsilon \ll \sigma\Delta t$ where the permittivity of free space is $\epsilon \approx 8.85 \cdot 10^{-12}$. We have $\mu \sim 4\pi \cdot 10^{-7}$ and we estimate roughly $\sigma \sim 1. \rightarrow \sigma \sim 10^6$, $c \sim 10^4$ and for large problems $h \sim 10^{-4}$. This gives $R_m \sim 1$ for regions with large conductivities. However, σ may drop by several orders of magnitude in material state transitions from solid through melt before returning to high values for high temperature plasma states. Void conductivity values should be lower than any material state values and we estimate values from 10^3 to 1. may be utilized. Thus the stiffness matrix will dominate by factors of 10^3 to 10^6 respectively in these void regions. Such low R_m states drive the requirement for an implicit magnetic diffusion solution methodology.

6.1. Physics fidelity studies. To validate the approach described above we consider a two-dimensional model problem obtained from the eddy current equations (2.3)-(2.6) by the ansatz

$$\mathbf{H} = H_z\mathbf{k} \quad \text{and} \quad \mathbf{E} = E_x\mathbf{i} + E_y\mathbf{j}.$$

The main objective is to verify correct initial transient phase and the steady state limit. The region Ω is a rectangle that is 0.003mm wide and 0.004mm high. The low conductivity region occupies a slot in the middle of the rectangle that is 0.003mm deep and 0.001mm wide; see Fig.6.1. The material permeability is

$$\mu = 4\pi \times 10^{-7}$$

in the whole region, while conductivity is a discontinuous function given by

$$\sigma = \begin{cases} 1 & \text{if } 0.001 < x < 0.002 \text{ and } 0.001 < y < 0.004 \\ 63.3 \times 10^6 & \text{otherwise} \end{cases}$$

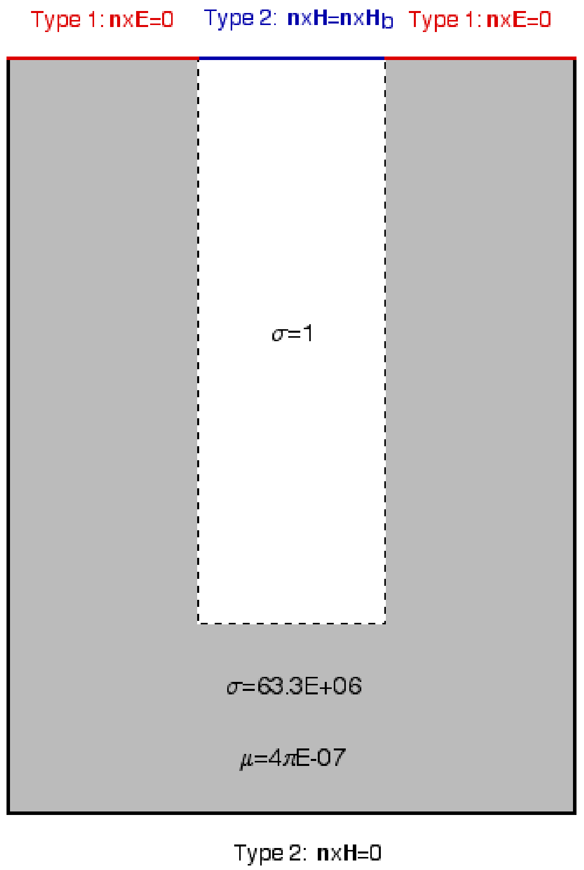

FIG. 6.1. *Model problem in two-dimensions*

The fields in the model problem are driven by a combination of Type I and Type II boundary conditions. Type II boundary consists of the center slot on the top side, the bottom side and the left and right sides of Ω. Type I boundary is the complement of Type II and contains the two segments on the left and right of the center slot on the top side; see Fig. 6.1. On Type I we prescribe homogeneous tangential **E**:

$$\mathbf{n} \times \mathbf{E} = 0 \quad \text{on } y = 0.004 \text{ and } 0 < x < 0.001 \text{ or } 0.002 < x < 0.003.$$

The boundary condition on Type II boundaries is natural for the weak equations. The tangential magnetic field is set to one at the center slot and zero elsewhere:

$$\mathbf{n} \times \mathbf{H} = \begin{cases} 1 & \text{on } 0.001 < x < 0.002,\ y = 0.004 \\ 0 & \text{all other parts of Type II boundary} \end{cases}$$

The fully discrete magnetic diffusion problem in two-dimensions is developed according to §4. The spatial discretization is effected using the 2D complex from §3.3. on uniform grids. Specifically, we employ grids consisting of 30×30 rectangles. Because the goal of the experiments in this section is to validate the discretization *qualitatively* the linear system (6.1) is fully assembled and solved "exactly" using banded Chlolesky factorization routine.

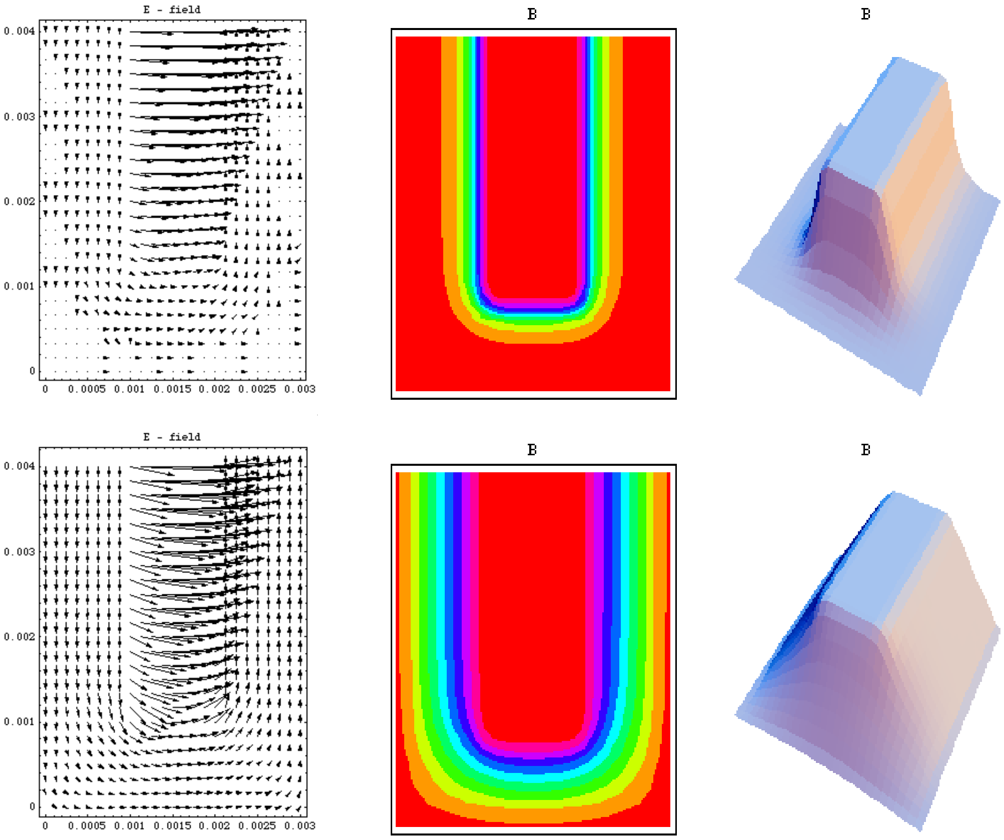

FIG. 6.2. *Electric field and magnetic flux density: initial and steady states.*

Numerical simulations were run for $\Delta t = 5 \times 10^{-6}$sec. and $\Delta t = 2.5 \times 10^{-6}$sec. In both cases steady state was reached at $t \approx 50 \times 10^{-6}$sec. This diffusion time is consistent with the prescribed material parameters. Figure 6.2 shows the initial electric field $\mathbf{E}$ and magnetic flux density $\mathbf{B}$ and their steady states obtained after 20 time steps with $\Delta t = 2.5 \times 10^{-6}$sec.

6.2. Scalability Studies. To demonstrate the performance of the edge-element based algebraic multigrid method, two test problems are solved within the ALEGRA framework. In all of our results, the notation $\mathrm{V}(k,k)$ (or $\mathrm{W}(k,k)$) indicates a multigrid V cycle (or W cycle) with k pre and post Hiptmair smoothing steps on each level. It is important to note that one multigrid cycle is used as a preconditioner to a conjugate gradient solver. Thus, the iteration counts correspond to conjugate gradient iterations.

The first test problem corresponds to a three-dimensional box on the unit cube (i.e. $\Omega = [0,1]^3$) with Neumann boundary conditions on the surface. Different experiments are performed by varying the conductivity (which is constant throughout the entire region) and by varying the mesh spacing. The conductivity is a weighting factor on the mass term of Equation (4.8). Hence, decreasing the conductivity emphasizes the $(\mathbf{curl}\,,\mathbf{curl}\,)$ term and makes the problem harder to solve.

Table 6.1 illustrates the results corresponding to the first linear solve for a $32 \times 32 \times 32$ mesh. The initial guess is identically zero and the right hand side is random.

	iterations					
conductivity	V(1,1)	V(2,2)	V(3,3)	W(1,1)	W(2,2)	W(3,3)
100	15	11	9	13	10	8
10	26	19	16	22	17	14
1	29	22	18	25	19	16

TABLE 6.1
Medium cube problem iterations.

	Iterations					
Conductivity	V(1,1)	V(2,2)	V(3,3)	W(1,1)	W(2,2)	W(3,3)
100	27	19	15	22	16	13
10	48	33	27	38	25	22
1	52	37	30	44	30	25

TABLE 6.2
Large cube problem iterations.

Convergence is declared when $||r||_2/||b||_2 \leq 10^{-11}$. In Table 6.2, a cube problem on a $64 \times 64 \times 64$ mesh is solved using the same initial guess and right hand side. ¿From the tables, it is clear that while the number of iterations initially grows with reduced conductivity, it does level off. That is, the convergence of the method can be experimentally bounded independently of the conductivity. Unfortunately, however, there is growth in the number of iterations as the grid is refined. (See Table 6.3.) This is to be expected as discussed in §5. The growth observed in the W cycle iterations is somewhat less than in the corresponding V cycle, especially for smaller conductivity (stiffer problem). Note that the stopping tolerance is quite small, which tends to emphasize this growth in iterations. For example, if we had required only $||r||_2/||b||_2 \leq 10^{-6}$ the iteration counts for a V(2,2) cycle for the $32 \times 32 \times 32$ and the $64 \times 64 \times 64$ meshes would be 9 and 14, respectively, a growth of 1.55. While there is growth, the symmetric Gauss-Seidel preconditioned method required 614 iterations as compared to no more than 37 for the multigrid runs. We are currently experimenting with techniques to improve the scalability of the multigrid technique by smoothing the interpolation operator as discussed in §5. While preliminary, encouraging results have been obtained for a model two dimensional problem:

$$\nabla \times \nabla \times u + \sigma u = f$$

on the unit square with Dirichlet boundary conditions. This problem is discretized with edge elements on a regular mesh with $\sigma = 1000$ and f taken as a random vector. Table 6.4 illustrate iteration counts required to reduce the initial residual by 10^{10} using a zero initial guess and defining the $\tilde{P}_k^{(e)}$'s via (5.7). While growth persists, it is much less significant for the smoothed interpolant which converges three times faster than the standard interpolant. This 'smoothing' technique will be pursued in a future paper.

Our second test problem corresponds to a more realistic model and is run in serial with two different mesh sizes. Figure 6.3 illustrates the solution after a single time step on a three dimensional domain consisting of a cylinder of highly conductive material with a cylindrical slot modeled by a very low conductivity region. The first test problem is meshed with $44,544$ hexahedral elements resulting in $46,761$ nodes and $130,008$ edges. This mesh is approximately 8 times larger than the one shown in Figure 6.3.

	Growth in Iterations					
Conductivity	V(1,1)	V(2,2)	V(3,3)	W(1,1)	W(2,2)	W(3,3)
100	1.80	1.72	1.66	1.69	1.60	1.62
10	1.85	1.73	1.68	1.73	1.47	1.57
1	1.79	1.68	1.66	1.76	1.57	1.56

TABLE 6.3
Growth in iteration count.

Grid Size	$P^{(e)}$	$\tilde{P}^{(e)}$
25×25	24	14
50×50	42	19
100×100	76	27
150×150	93	30

TABLE 6.4
V(1,1) iteration counts using both the standard and 'smoothed' interpolation operators.

A sharp jump in the conductivity occurs between the slot and the material regions. Specifically, the conductivity of the outer region is 6.33×10^7 while the conductivity of the inner "void" region is given the small value 1.0. Homogeneous electric Dirichlet boundary conditions are applied on the center and outside top surfaces. Homogeneous Neumann boundary conditions are applied to the outer and bottom surfaces and an inhomogeneous azimuthal tangential field condition is applied on the top middle ring surface. For the time step chosen, the field fills the slot immediately. This problem is intended to be a first approximation to the Z-pinch apparatus described in §1. In the second run the mesh contains $116,473$ nodes and $345,768$ edges. Table 6.5 illustrates the results for the first linear solve for each problem size. The initial guess is the zero vector, and the stopping criteria is $||r||_2/||b||_2 \leq 10^{-8}$. Both the V(1,1) and W(2,2) converge in a small number of iterations. Once again, there is modest growth in the number of iterations. By comparision, a conjugate gradient with symmetric Gauss-Seidel preconditioning requires 684 iterations for the smaller problem. Given that this problem is still relatively small, it is unlikely that the Gauss-Seidel preconditioning will lead to convergence on significantly larger problems.

At this time the parallel code is still being optimized. Given the promising serial results, we expect that the parallel version should also perform reasonably well. The major change is the use of processor-block Gauss-Seidel within the Hiptmair smoother. (See the footnote in §5.4.)

7. Conclusions. We have described an edge and face finite element discretization for the eddy current equations on arbitrary quadrilateral and hexahedral meshes in heterogeneous media and described a particular implementation of an algebraic multigrid technique appropriate to this discretization. Numerical results are given indicating both the fidelity of the representation and the efficacy of the algebraic multigrid methodology.

8. Acknowledgements. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

REFERENCES

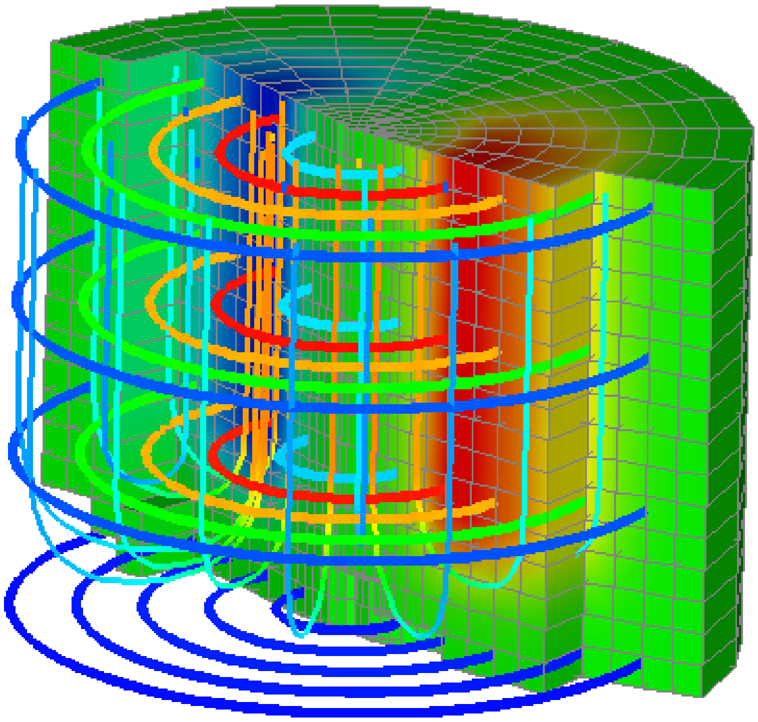

FIG. 6.3. *Cutaway of axial slot showing Y component of magnetic flux density after one time step and streamlines of current density (thin) and magnetic flux density (thick).*

	Iterations			
Problem Size	V(1,1)	V(2,2)	W(1,1)	W(2,2)
130,008	42	20	35	18
345,768	54	28	42	22

TABLE 6.5
Slot problem iterations.

[1] D. N. Arnold, R. S. Falk, and R. Winther, *Multigrid in H(div) and H(curl)*, Numer. Math., 85 (2000), pp. 197–217.

[2] T. Austin, *Advances on a Scaled Least-Squares Method for the Neutron Transport Equation*, PhD thesis, University of Colorado – Boulder, 2001.

[3] O. Biro and K. Preis, *On the use of the magnetic vector potential in the finite element analysis of three-dimensional eddy currents*, IEEE Transactions on Magnetics, 25 (1989), pp. 3145–3159.

[4] O. Biro and K. Preis, *Finite element analysis of 3-D eddy currents*, IEEE Transactions on Magnetics, 26 (1990), pp. 418–423.

[5] A. Bossavit, *A rationale for "edge-elements" in 3-D fields computations*, IEEE Transactions on Magnetics, 24 (1988), pp. 74–79.

[6] ———, *Computational electromagnetism*, Academic Press, 1998.

[7] A. Bossavit and J. Verite, *A mixed fem-biem method to solve 3-d eddy current problems*, IEEE Transactions on Magnetics, MAG-18 (1982), pp. 431–435.

[8] D. Braess, *Finite elements. Theory, fast solvers, and applications in solid mechanics*, Cambridge University Press, Cambridge, 1997.

[9] A. Brandt, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Tech. Report Nr. 85, GMD-Studie, Sankt Augustin, West Germany, 1984.

[10] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, 1991.

[11] W. L. Briggs, V. E. Henson, and S. McCormick, *A Multigrid Tutorial, Second Edition*, SIAM, Philadelphia, 2000.

[12] C. F. Bryant, C. R. I. Emson, and C. W. Trowbridge, *A comparison of Lorentz gauge formulations in eddy current computations*, IEEE Transactions on Magnetics, 26 (1990), pp. 430–433.

[13] ———, *A general purpose 3-d formulation for eddy currents using the Lorentz gauge*, IEEE Transactions on Magnetics, 26 (1990), pp. 2373–2375.

[14] C. F. Bryant, C. R. I. Emson, C. W. Trowbridge, and P. Fernandes, *Lorentz gauge formulations involving piecewise homogeneous conductors*, IEEE Transactions on Magnetics, 34 (1998), pp. 2559–2562.

[15] M. F. F. Brezzi, J. Douglas and D. Marini, *Efficient rectangular mixed finite elements in two and three space variables*, M^2AN Math. Model. Numer. Anal., 21 (1987), pp. 581–604.

[16] R. Hiptmair, *Multigrid method for Maxwell's equations*, SIAM J. Numer. Anal., 36 (1998), pp. 204–225.

[17] J. Nedelec, *Mixed finite elements in* $\mathbf{R}^3$, Numerische Mathematik, 35 (1980), pp. 315–341.

[18] ———, *A new family of finite element methods in* $\mathbf{R}^3$, Numerische Mathematik, 50 (1986), pp. 57–81.

[19] J. S. Peery and D. E. Carroll, *Multi-material ALE methods in unstructured grids*, Computer Methods in Applied Mechanics and Engineering, 187 (2000), pp. 591–619.

[20] S. Reitzinger and J. Schöberl, *Algebraic multigrid for finite element discretizations with edge elements.* Unpublished manuscript. reitz@sfb013.uni-linz.ac.at.

[21] R. M. Summers, J. S. Peery, M. W. Wong, and et. al., *Recent progress in ALEGRA development and application to ballistic impacts*, Int. J. Impact Engng., 20 (1997), pp. 779–788.

[22] C. Tong and R. S. Tuminaro, *ML 2.0 smoothed aggregation user's guide*, Tech. Report SAND2001-8028, Sandia National Laboratories, December 2000.

[23] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[24] R. Tuminaro and C. Tong, *Parallel smoothed aggregation multigrid: Aggregation strategies on massively parallel machines*, in SuperComputing 2000 Proceedings, J. Donnelley, ed., 2000.

[25] P. Vaněk, M. Brezina, and J. Mandel, *Convergence of Algebraic Multigrid Based on Smoothed Aggregation*, Tech. Report report 126, UCD/CCM, Denver, CO, 1998.

[26] 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.

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

[28] J. S. V. Welij, *Calculation of eddy currents in terms of H on hexahedra*, IEEE Transactions on Magnetics, MAG-21 (1985), pp. 2239–2241.

[29] G. Yonas, *Fusion and the Z pinch*, Scientific American, 279 (1998), pp. 22–27.