# A QR Accelerated Volume-to-Surface Boundary Condition for the Finite-Element Solution of Eddy-Current Problems

Daniel A. White, Benjamin J. Fasenfest, Robert N. Rieben, and Mark L. Stowell

Defense Sciences Engineering Division, Lawrence Livermore National Laboratory, Livermore, CA 94551 USA

**We are concerned with the solution of time-dependent electromagnetic eddy-current problems using a finite-element formulation on three-dimensional unstructured meshes. We allow for multiple conducting regions, and our goal is to develop an efficient computational method that does not require a computational mesh of the air/vacuum regions. This requires a sophisticated global boundary condition specifying the total fields on the conductor boundaries. To meet this requirement, we propose a volume-to-surface boundary condition based on the Biot–Savart law. We found the Biot–Savart approach to be very accurate. In addition, this approach can be accelerated via a low-rank QR approximation of the discretized Biot–Savart law.**

***Index Terms*—Biot–Savart law, computational electromagnetics, eddy currents, electromagnetic diffusion, low-rank approximation, Maxwell's equations, parallel processing.**

## I. Introduction

In this paper, we present a finite-element method for solving the multiply connected eddy-current problem. While much of this presentation is applicable towards frequency-domain analysis, our emphasis here is on transient simulation. Various formulations for the eddy-current equations exist and have been extensively reviewed and studied in the literature. These include formulations which solve for the electric field (the $\vec{E}$ field formulation) [1]–[3], the magnetic field (the $\vec{H}$ field formulation) [4], [5] or for the potential field (the $\vec{A}$-$\phi$ potential formulation) [6]–[9]. Each formulation has its advantages and disadvantages for problems in computational electromagnetics. However, it has been shown that when using $H(\mathrm{Curl})$ and $H(\mathrm{div})$ conforming finite-element methods, there is no difference in *accuracy* for these three formulations, even for secondary quantities such as $\vec{B}$ and $\vec{J}$ [10]. The difference between the three formulations, which use primary field variables $\vec{E}$, $\vec{H}$, and $\vec{A}$, respectively, is in the boundary conditions and the source terms, and is therefore simply a matter of which formulation is most convenient for a given electromagnetics problem.

The most difficult electromagnetic diffusion problems encountered in practice are those that involve multiple conductors separated by a nonconducting region, the so-called multiply connected eddy-current problem. While the currents are zero in the nonconducting region clearly the fields are not, and some method must be used to account for these fields. One approach is to simply mesh the nonconducting region and use a small value of conductivity in this region. While seemingly a crude approach, it works well in practice for many problems, for example using a conductivity at least $10^3$ times smaller than the metal results in fields correct to within the discretization error [7], [11]. The difficulty with this is twofold; one is the large number of unknowns and the second is matrix ill-conditioning. More sophisticated approaches include forming a magnetostatic problem in the nonconducing region using either the vector or scalar magnetic potential and coupling the two finite-element solutions [6], [12], or employing a surface integral equation to correctly model the global boundary condition [1]–[3].

In the context of Galerkin approximations of electromagnetics partial differential equations (PDEs), the choice of the finite-element space plays a crucial role in the stability and convergence of the discretization. For instance, in numerical approximations of the magnetic and electric field intensities, $H(\mathrm{Curl})$ conforming finite-element spaces (or edge elements) are preferred over traditional nodal vector spaces since they eliminate spurious modes in eigenvalue computations and they prevent fictitious charge buildup in time-dependent computations. The lowest order $H(\mathrm{Curl})$ conforming basis functions were developed by Whitney [14] before the advent of finite-element programs. Arbitrary order versions were introduced by Nédélec [15], [16] as a generalization of the mixed finite-element spaces introduced by P. A. Raviart and J. M. Thomas [17] for $H(\mathrm{div})$ conforming methods. Application of these $H(\mathrm{Curl})$ and $H(\mathrm{div})$ basis functions toward electromagnetics is becoming quite popular and applications can be found in several recent textbooks [18], [5], [19].

In this paper, we focus on the $\vec{A}$ field formulation for the eddy-current problem using $H(\mathrm{Curl})$ basis functions. We assume a given initial condition (which may be zero) and the problem is driven by either a time-varying voltage or a prescribed time-varying current. We allow for multiple conducting regions separated by air. Each individual conducting region is assumed to be homogeneous. The conductors are nonmagnetic.

Digital Object Identifier 10.1109/TMAG.2006.889766

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

TABLE I
ELECTROMAGNETIC QUANTITIES AND THEIR ASSOCIATED DIFFERENTIAL FORMS

| Physical Quantity | Units | Diff. Form |
|---|---|---|
| Scalar Potential, $\phi$ | $Volts/m^0$ | 0-form |
| Vector Potential, $\vec{A}$ | $Webbers/m^1$ | 1-form |
| Electric Field Intensity, $\vec{E}$ | $Volts/m^1$ | 1-form |
| Magnetic Field Intensity, $\vec{H}$ | $Amperes/m^1$ | 1-form |
| Electric Flux Density, $\vec{D}$ | $Coulombs/m^2$ | 2-form |
| Magnetic Flux Density, $\vec{B}$ | $Webbers/m^2$ | 2-form |
| Electric Charge Density, $\rho$ | $Coulombs/m^3$ | 3-form |

Our targeted applications which satisfy these assumptions include induction heating, metal forming, and helical magnetic flux compression generators. Boundary conditions must be applied on the conductor surfaces, and here we propose to compute the surface $\vec{A}$ field (for the inhomogeneous Dirichlet condition) or $\vec{B}$ field (for the inhomogeneous Neumann condition) by a direct discretization of the Biot–Savart law. Compared to surface integral equation methods this approach does not *solve* an integral equation, only the *evaluation* of an integral is required. We do not solve for surface currents and hence need not be concerned with loop-star or spanning-tree processing for suppression of irrotational surface currents. The disadvantage of the Biot–Savart law approach is that it requires a volume-to-surface computation and hence can be expensive. But this computation can be accelerated using a low-rank QR approximation. Low-rank QR approximation has been demonstrated to be effective at accelerating a variety of electromagnetic integral equations [20]–[23]. We do not prove that the low-rank QR approximation is superior to multipole expansions, panel methods, or other acceleration schemes; we employ the low-rank QR approximation because the algorithm is independent of the Green's function and can thus be applied to either form of the Biot–Savart law. Our proposed boundary condition algorithm is verified by comparing to exact analytical solutions and by comparing to simply using a large mesh of the air/vacuum region.

## II. ELECTROMAGNETIC DIFFUSION EQUATION

When working with multiple finite-element spaces, it becomes convenient to use the notation of differential forms as a way of categorizing the various field quantities from Maxwell's equations and the subsequent finite-element spaces used to discretize them. In addition, the calculus of differential forms provides the necessary transformation rules which allow complicated basis functions to be derived on a reference element and then mapped to global mesh elements. Table I lists various physical quantities in electromagnetics and their associated differential form.

In electromagnetics we have the electric and magnetic fields $\vec{E}$, $\vec{H}$, the electric and magnetic flux densities $\vec{D}$, $\vec{B}$, and the constitutive relations

$$\vec{D} = \epsilon\vec{E}$$
$$\vec{B} = \mu\vec{H}.$$

Here, we write Maxwell's Equations in terms of $\vec{E}$ and $\vec{B}$

$$\epsilon\frac{\partial\vec{E}}{\partial t} = \vec{\nabla}\times\frac{1}{\mu}\vec{B} - \sigma\vec{E} - \vec{J}_s \tag{1}$$
$$\frac{\partial\vec{B}}{\partial t} = -\vec{\nabla}\times\vec{E} \tag{2}$$
$$\vec{\nabla}\cdot\epsilon\vec{E} = 0 \tag{3}$$
$$\vec{\nabla}\cdot\vec{B} = 0 \tag{4}$$

with appropriate boundary conditions and initial conditions understood. Note that $\vec{J}_s$ is an independent current source term, which may or may not exist for every problem. In all of our subsequent formulations, the material properties $\epsilon, \mu, \sigma$ are free to be symmetric positive definite tensor functions of space, but we impose the restriction that they are linear time invariant.

Now consider solving Maxwell's equations within a good conductor. A good conductor is defined by the condition

$$\epsilon\frac{\partial\vec{E}}{\partial t} \ll \sigma\vec{E}. \tag{5}$$

Note that (5) depends not only on the material properties $\sigma$ and $\epsilon$, but also on the time rate of change of $\vec{E}$. When (5) is satisfied, Maxwell's equations can be simplified by neglecting the $\epsilon(\partial\vec{E})/(\partial t)$ term altogether, the so-called low-frequency approximation, diffusion approximation, or eddy-current approximation. The diffusion approximation is not valid for most radio-frequency (RF), microwave, or optics problems, but is reasonable for low-frequency electromagnetic (EM) waves in plasmas or in the earth, as well as for engineering design problems such as electric motors, transformers, induction heating, metal forming, and rail-guns. A detailed, mathematical analysis which justifies the approximation is given in [24].

### A. The $\vec{A}$-$\Phi$ Potential Formulation

The divergence condition (4) implies that $\vec{B} = \vec{\nabla}\times\vec{A}$, where $\vec{A}$ is a magnetic vector potential. This in turn implies that the electric field is given by $\vec{E} = -\vec{\nabla}\phi - (\partial)/(\partial t)\vec{A}$, where $\phi$ is an electric scalar potential. Using these two potentials, along with the gauge condition $\vec{\nabla}\cdot\sigma\vec{A} = 0$, the potential diffusion equations in a three-dimensional domain $\Omega$ are

$$\vec{\nabla}\cdot\sigma\vec{\nabla}\phi = 0 \tag{6}$$
$$\sigma\frac{\partial\vec{A}}{\partial t} = -\vec{\nabla}\times\frac{1}{\mu}\vec{\nabla}\times\vec{A} - \sigma\vec{\nabla}\phi + \vec{J}_s \tag{7}$$
$$\vec{E} = -\vec{\nabla}\phi - \frac{\partial}{\partial t}\vec{A} \tag{8}$$
$$\vec{B} = \vec{\nabla}\times\vec{A} \tag{9}$$
$$\vec{J} = \sigma\vec{E}. \tag{10}$$

We refer to $\vec{A}$ and $\phi$ as the primary field variables and $\vec{B}$, $\vec{E}$, and $\vec{J}$ as the secondary variables, with the computation of the secondary variables optional. The inhomogeneous Dirichlet boundary conditions are given by

$$\phi = \phi_{\text{surf}} \text{ on } \Gamma \tag{11}$$
$$\hat{n}\times\vec{A} = A_{\tan} \text{ on } \Gamma \tag{12}$$

and in the Galerkin finite-element method these will appear as constraints on the solution and will be satisfied exactly. The inhomogeneous Neumann boundary conditions are given by

$$\hat{n} \cdot \sigma \vec{\nabla} \phi = J_n \text{ on } \Gamma \tag{13}$$

$$\hat{n} \times \frac{1}{\mu} \vec{\nabla} \times \vec{A} = H_{\tan} \text{ on } \Gamma \tag{14}$$

and in the Galerkin finite-element method these will appear as equivalent sources and will be satisfied in the weak sense. There are divergence constraints on both the primary and secondary fields, namely

$$\vec{\nabla} \cdot \sigma \vec{A} = 0 \tag{15}$$

$$\vec{\nabla} \cdot \vec{B} = 0 \tag{16}$$

and these will be satisfied automatically by the discretization method. No penalty term or projection step will be required.

## III. Finite-Element Discretization

FEMSTER is a class library of finite elements used for the results in this paper; see [25]–[27] for details. It provides discrete numerical implementations of the concepts from differential forms (tangent vectors, wedge product, exterior derivative, hodge-star operator, etc.). The language of differential forms has long been used for elucidating the laws of electromagnetics in a coordinate system free manner [28]–[30]. We have found this language to be extremely useful in providing the abstract software foundation for FEMSTER. In standard finite-element language, FEMSTER contains all the data structures and operations required to compute local finite-element matrices: elements (tetrahedrons, hexahedrons, prisms), basis functions (or shape functions), quadrature rules, linear forms, and bilinear forms. FEMSTER provides the gradient, curl, and divergence operators, as well as the div-grad, curl-curl, and grad-div operators. Note that arbitrary partial derivative operators are not provided, as these do not fit nicely into the framework of differential forms, and fortunately are not needed for computational electromagnetics. The basis function class hierarchy contains four forms of basis functions, simply called 0-forms, 1-forms, 2-forms, and 3-forms. Derived from each of these classes are subclasses for the element types tetrahedron, hexahedron, and prism, and derived from each of these types is a further specialization for the degrees of freedom, i.e., interpolatory, spectral, hierarchical, etc. The critical step in using FEMSTER is to decide which form should be used for each physical quantity. The essential properties of the forms are now summarized.

0-forms are continuous scalar basis functions that have a well-defined gradient. These basis functions are a finite subspace of $H(\mathrm{Grad})$ and are suitable for discretization of the electric potential $\phi$, temperature $T$, etc. The basis functions are dimensionless, hence the degrees of freedom have the same units as the field being approximated. If the field is temperature, the degrees of freedom have units of temperature. The gradient of a 0-form basis function can be represented, exactly, as a combination of 1-form basis functions, i.e., $d\vec{W}^0 \subset \vec{W}^1$.

1-forms are vector basis functions with continuous tangential components across elements, but discontinuous normal components. They have a well-defined curl, but do not have a well-defined divergence. These basis functions are a finite subspace of $H(\mathrm{Curl})$. The basis functions have units of $\mathrm{m}^{-1}$. For example, the electric field has units of $\mathrm{Volts}/\mathrm{m}$ and the degrees-of-freedom will therefore have units of Volts. It is a simple matter to integrate 1-forms along the edges of a mesh, but surface integrals are not well defined. These basis functions are ideally suited for the electric field $\vec{E}$, the magnetic field $\vec{H}$, the magnetic vector potential $\vec{A}$, etc. The curl of a 1-form basis function can be represented, exactly, as a combination of 2-form basis functions, i.e., $d\vec{W}^1 \subset \vec{W}^2$. The null space of the curl operator on 1-forms is, exactly, the space of gradients of 0-forms, $d\vec{W}^1 = 0$ implies $\vec{W}^1 = dW^0$, for simply-connected regions.

2-forms are vector basis functions with continuous normal components across elements, but discontinuous tangential components. They have a well-defined divergence, but do not have a well-defined curl. These basis functions are a finite subspace of $H(\mathrm{div})$. The basis functions have units of $\mathrm{m}^{-2}$. For example, the electric current density has units of $\mathrm{Amperes}/\mathrm{m}^2$; therefore, the degrees of freedom have units of Amperes. It is a simple matter to integrate 2-forms over surfaces of a mesh, but line integrals are not well defined. These basis functions are ideally suited for the electric flux density $\vec{D}$, the magnetic flux density $\vec{B}$, current flux density $\vec{J}$, etc. The divergence of a 2-form basis function can be represented, exactly, as a combination of 3-form basis functions. The null space of the divergence operator on 2-forms is, exactly, the space of curls of 1-forms.

3-forms are discontinuous scalar basis functions. They cannot be differentiated. They can be integrated over a volume, but not over a surface or a line. These basis functions are a finite subspace of $L^2$. The basis functions have units of $\mathrm{m}^{-3}$. For example, charge density has units of $\mathrm{Coulombs}/\mathrm{m}^3$ and the degrees of freedom will therefore have units of Coulombs. These basis functions are ideally suited for the electric charge density $\rho$, the energy density $\varepsilon$, etc.

FEMSTER computes the following "mass," "stiffness," and "derivative" matrices, where the superscript $l = 0, 1, 2, 3$ denotes the degree of the form

$$\mathbf{M}^l(\alpha)_{ij} = \int_\Omega \alpha W_i^l \wedge W_j^l \, d\Omega \tag{17}$$

$$\mathbf{S}^l(\alpha)_{ij} = \int_\Omega \alpha dW_i^l \wedge dW_j^l \, d\Omega \tag{18}$$

$$\mathbf{D}^{l(l+1)}(\alpha)_{ij} = \int_\Omega \alpha dW_i^l \wedge W_j^{l+1} \, d\Omega. \tag{19}$$

Note that the $d$ operator denotes Gradient, Curl, or Divergence, for $l = 0, 1, 2$, respectively. The "mass" matrices $\mathbf{M}$ and the "stiffness" matrices $\mathbf{S}$ are square and map $l$-forms to $l$-forms, the "derivative" matrices $\mathbf{D}$ are rectangular and map $l$-forms to $(l+1)$-forms. Note that $\alpha$ is the Hodge operator which maps $l$-forms to $3-l$-forms, and is associated with material properties such as electric conductivity and magnetic permeability. It can be shown that

$$\mathbf{D}^{l(l+1)} = \mathbf{M}^{l+1} \mathbf{K}^{l(l+1)} \tag{20}$$

$$\mathbf{S}^l = \left( \mathbf{K}^{l(l+1)} \right)^T \mathbf{M}^{l+1} \mathbf{K}^{l(l+1)} \tag{21}$$

where $\mathbf{K}^{l(l+1)}$ is a "topological derivative" matrix. This matrix is the discretization of the exterior derivative operator $d$ from differential geometry, $dW^l = W^{(l+1)}$. This matrix depends

upon the mesh connectivity, but is independent of the nodal coordinates. It does not involve an integral over the element, and it does not involve any material properties. While seemingly abstract, it is enormously valuable in practice. Given an $l$-form quantity $X$ with basis function expansion

$$X = \sum_{i=1}^{n} x_i W_i^l \tag{22}$$

and an $(l+1)$-form quantity $Y$ with basis function expansion

$$Y = \sum_{i=1}^{n} y_i W_i^{(l+1)} \tag{23}$$

the exterior derivative (Gradient, Curl, Divergence for $l = 0$, $l = 1$, and $l = 2$, respectively) is given by

$$\mathbf{y} = \mathbf{K}^{l(l+1)}\mathbf{x}. \tag{24}$$

It can be shown that

$$\mathbf{K}^{12}\mathbf{K}^{01} = 0 \tag{25}$$

$$\mathbf{K}^{23}\mathbf{K}^{12} = 0 \tag{26}$$

which are the discrete versions of $d(dW^l) = 0$. In terms of standard vector calculus, these matrix relations correspond to the identities $\vec{\nabla} \times \vec{\nabla} f = 0$ and $\vec{\nabla} \cdot \vec{\nabla} \times \vec{F} = 0$, respectively. These identities are satisfied in the discrete sense, exactly (to machine precision), for any mesh and any order basis function.

FEMSTER contains some additional miscellaneous functionality. In some circumstances it is necessary to convert an $l$-form to a $(3-l)-$form, i.e., a Hodge-star operation. A classic example is converting a "cell-center" quantity to a "nodal" quantity. In our finite-element setting the Galerkin procedure prescribes rectangular matrices of the form

$$\mathbf{H}_{ij}^{p(3-l)} = \int_{\Omega} W_i^l \wedge W_j^{(3-l)}\, d\Omega \tag{27}$$

which produces optimal (in the least-square error sense) Hodge-star operators for arbitrary order basis functions.

### A. Semi-Discrete Potential Diffusion Equations

As per Table I we employ 0-form basis functions for $\phi$, 1-form basis functions for $A$ and $E$, and 2-form basis functions for $B$ and $J$, leading to the following basis functions expansions:

$$\phi(x,y,z,t) = \sum_{i=1}^{n0} p_i(t)\vec{W}_i^0(x,y,z) \tag{28}$$

$$\vec{A}(x,y,z,t) = \sum_{i=1}^{n1} a_i(t)\vec{W}_i^1(x,y,z) \tag{29}$$

$$\vec{E}(x,y,z,t) = \sum_{i=1}^{n1} e_i(t)\vec{W}_i^1(x,y,z) \tag{30}$$

$$\vec{B}(x,y,z,t) = \sum_{i=1}^{n2} b_i(t)\vec{W}_i^2(x,y,z) \tag{31}$$

$$\vec{J}(x,y,z,t) = \sum_{i=1}^{n2} j_i(t)\vec{W}_i^2(x,y,z) \tag{32}$$

with $x_i$ the $i$th degree of freedom for field $X$. The integers $n0, n1, n2$ refer to the dimensions of the discrete $p$-form spaces, and for lowest order basis functions these correspond to the number of mesh nodes, edges, and faces, respectively.

We employ the Galerkin variational procedure to convert the PDEs (6)–(10) to a semi-discrete system of equations, yielding

$$\mathbf{S}^0(\sigma)\,\mathbf{v} = 0 \tag{33}$$

$$\mathbf{M}^1(\sigma)\,\frac{\partial \mathbf{a}}{\partial t} = -\mathbf{S}^1(\mu^{-1})\,\mathbf{a} - \mathbf{D}^{01}(\sigma)\,\mathbf{v} + \mathbf{j} \tag{34}$$

$$\mathbf{e} = -\mathbf{K}^{01}\mathbf{v} - \frac{\partial \mathbf{a}}{\partial t} \tag{35}$$

$$\mathbf{b} = \mathbf{K}^{12}\mathbf{a} \tag{36}$$

$$\mathbf{M}^2(\sigma^{-1})\mathbf{j} = \mathbf{H}^{12}\mathbf{e} \tag{37}$$

where the matrices have been defined above. The divergence constraints are given by

$$(\mathbf{D}^{01}(\sigma))^T\mathbf{a} = 0 \tag{38}$$

$$(\mathbf{D}^{01}(\sigma))^T\mathbf{e} = 0 \tag{39}$$

$$\mathbf{K}^{23}\mathbf{b} = 0 \tag{40}$$

and from the identities (25) and (26) these constraints are implicitly satisfied for all time, assuming the initial conditions and the source terms are divergence free. The divergence constraints (38) and (39) are often referred to as "weak" or "variational" since they are derived from the integration-by-parts formula

$$\int_{\Omega} dF \wedge W = \int_{\Gamma} F \wedge W - \int_{\Omega} F \wedge dW \tag{41}$$

where $F$ is the field of interest ($A$ or $E$) and $W$ is a "test function" which is zero on the boundary $\Gamma$. These divergence constraints are thus not enforced on the boundary; the divergence of the fields on the boundary is determined by the the choice of Dirichlet and Neumann boundary conditions on $\phi$ and $A$ as described above. Finally, we should note that the electrostatic potential $v$ and the independent current source $j$ may not exisit for all problems of interest; these terms are included here for generality. Also, while it is in principle possible to employ periodic or symmetry boundary conditions, this is not investigated here, as it would require the use of more complicated Green's functions in the Biot–Savart law. The computational mesh must therefore span all the metal regions.

### B. Boundary Conditions

The discrete version of the Dirichlet boundary condition (12) is simply to constrain degree-of-freedom values $\mathbf{a}_i$ on the surface to known values. In finite-element terminology this is an *essential* boundary condition. Since the $\vec{A}$-field update equation involves the solution of a linear system, this constraint is part of the matrix solve step. We briefly review our approach for implementing essential boundary conditions here.

Consider the $n \times n$ linear system of equations $\mathbf{Ax} = \mathbf{b}$ and assume that some subset of the solution vector $\mathbf{x}$ is subject to point constraints of the form $\mathbf{x}_i = \mathbf{q}_i, i = 1, m$. For ease of presentation the solution vector $\mathbf{x}$ is sorted into unconstrained and constrained subvectors, in our application the unconstrained components correspond to internal degrees of freedom and the

constrained components correspond to boundary degrees of freedom. The system of equations can be written in $2 \times 2$ block form as

$$\begin{bmatrix} \mathbf{A}_{II} & \mathbf{A}_{BI} \\ \mathbf{A}_{IB} & \mathbf{A}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{x}_I \\ \mathbf{x}_B \end{bmatrix} = \begin{bmatrix} \mathbf{b}_I \\ \mathbf{b}_B \end{bmatrix}. \tag{42}$$

The approach is to modify the matrix and the right-hand-side vector in a manner that preserves the symmetry of the original system. The solution is given by

$$\begin{bmatrix} \mathbf{A}_{II} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_I \\ \mathbf{x}_B \end{bmatrix} = \begin{bmatrix} \mathbf{b}_I - \mathbf{A}_{IB}\mathbf{q} \\ \mathbf{q} \end{bmatrix}. \tag{43}$$

In practice, diagonally scaled conjugate gradient is effective at solving this system of equations.

The discrete version of the Neumann boundary condition (14) is different. In the Galerkin procedure integration-by-parts is performed on the Curl-Curl operator

$$\int_\Omega \nabla \times \mu^{-1} \nabla \times \vec{A} \cdot \vec{W} d\Omega = \int_\Omega \mu^{-1} (\nabla \times \vec{A}) \cdot (\nabla \times \vec{W}) d\Omega - \oint_\Gamma \mu^{-1} (\vec{W} \times \nabla \times \vec{A}) \cdot \hat{n} d\Gamma. \tag{44}$$

The surface integral can be expressed as

$$\oint_\Gamma \mu^{-1} (\vec{W} \times \nabla \times \vec{A}) \cdot \hat{n} d\Gamma = - \oint_\Gamma \hat{n} \times \vec{H} \cdot \vec{W} d\Gamma \tag{45}$$

clearly this is an effective surface current source $\vec{J}_{\text{eff}} = n \times \vec{H}$ that is added to the right-hand side of (34). In finite-element terminology the special case of $\vec{H} = 0$ on the boundary is referred to as a *natural* boundary condition; this is the boundary condition that is naturally satisfied by the variational formulation. It is also referred to as a weak boundary condition; it is not satisfied exactly but is instead satisfied only to within the approximation error. In the next section, a volume-to-surface integral is proposed to determine $\vec{A}$ and $\vec{H}$ on the boundary.

## IV. Biot–Savart Law

Our goal is to solve the $\vec{A}$-$\phi$ equations in the metal regions using the $H(\text{curl})$-conforming finite-element method described in Section III above. Let a metal-air interface be denoted by $\Gamma$. On each metal region, either $\hat{n} \times \vec{A}$ or $\hat{n} \times \vec{H}$ must be specified on $\Gamma$. In the metal region, the gauge condition $\nabla \cdot \sigma \vec{A} = 0$ is used as this is the the condition that is automatically satisfied by the $H(\text{curl})$-conforming finite-element discretization as discussed above. In the air region we have $\sigma = 0$, there are no currents, and the fields are quasi-static in nature. By quasi-static we mean that the the fields are time varying, but that at every instant of time we have $\nabla \times \vec{B} = \nabla \times \nabla \times \vec{A} = 0$, i.e., the fields have the spatial dependence of a static field. We propose solving the electromagnetic diffusion $\vec{A}$-$\phi$ (6)–(10) using the above finite-element method *in the metal regions only*, and using the Biot–Savart law to compute the fields *on the surface* $\Gamma$ *only*. The key issues are the stable coupling of the Biot–Savart law with the the finite-element discretiation, and the efficiency of the resulting hybrid finite-element boundary-element system of equations, and these issues are discussed here.

The law of Biot and Savart is given by

$$\vec{H}(x) = \frac{1}{\mu} \nabla \times \vec{A} = \frac{1}{4\pi} \int_{\Omega'} \frac{\vec{J}(x') \times (x - x')}{|x - x'|^3} d\Omega' \tag{46}$$

where $\vec{J}(x')$ is a known volume current density at the source point $x'$, and $\vec{H}(x)$ is the desired magnetic field at the observation point $x$. A related equation for the magnetic vector potential $\vec{A}$ is given by

$$\vec{A}(x) = \frac{\mu}{4\pi} \int_{\Omega'} \frac{\vec{J}(x')}{|x - x'|} d\Omega'. \tag{47}$$

Equation (47) is the fundamental solution to the equation $\nabla \times \mu^{-1} \nabla \times \vec{A} = \vec{J}$ in an infinite homogeneous medium. Technically, (47) is the fundamental solution to $\nabla^2 \vec{A} = -\mu \vec{J}$, but since $\sigma = 0$ in the air region, the gauge condition $\nabla \cdot \vec{A} = 0$ is used in the air region, and the standard vector identity $\nabla^2 \vec{A} = \nabla(\nabla \cdot \vec{A}) - \nabla \times \nabla \times \vec{A}$ results in (47). Equation (46) follows directly from $\vec{H} = \mu^{-1} \nabla \times \vec{A}$. We assume that $\mu$ is a constant scalar over the entire problem; we restrict ourselves to problems involving nonmagnetic conducting materials. Inhomogeneous magnetic materials could be accounted for by adding effective surface currents on material interfaces, but this is not considered here.

As discussed in Section III, the finite-element method requires only the tangential component of $\vec{A}$ (or $\vec{H}$) be specified on $\Gamma$, the normal component of $\vec{A}$ (or $\vec{H}$) on $\Gamma$ is not needed and is not computed. Since the fields computed in the air region via the Biot–Savart law are continuous, we are restricted to problems in which the tangential component of $\vec{A}$ is continuous on $\Gamma$. In general the normal component of $\vec{A}$ is discontinuous across $\Gamma$, but this does not pose a computational problem since the normal component of $\vec{A}$ on $\Gamma$ is not needed and is not computed. Note that if the problem of interest consisted of two dissimilar conductors touching each other, and both of these conductors simultaneously touching the air (a "triple point"), the current density $\vec{J}$ would be continuous but the tangential conponent of the $\vec{A}$-field would posses a jump discontinuity due to our choice of gauge; this is illustrated in Fig. 1. For this latter case, the $\vec{A}$-field Biot–Savart law (47) cannot be used, and (46) is the more appropriate boundary condition, as the tangential component of the $\vec{H}$-field is continuous regardless of conductivity. As stated in the Introduction, in this paper we are focused on mutiple homogenous conducting regions separated by air (no "triple points"), and for this class of problem the $\vec{A}$ field is continuous *in the metal*, the $\vec{A}$ field is also continuous *in the air*, and it is possible to enforce tangential continuity of these fields by solving a linear system of equations.

Employing basis function expansions for $\vec{J}$ and $\vec{H}$, multiplying by 1-form test functions $W^1(x)$ and integrating over the surface gives

$$\mathbf{M}^s \mathbf{h} = \mathbf{Z} \mathbf{j} \tag{48}$$

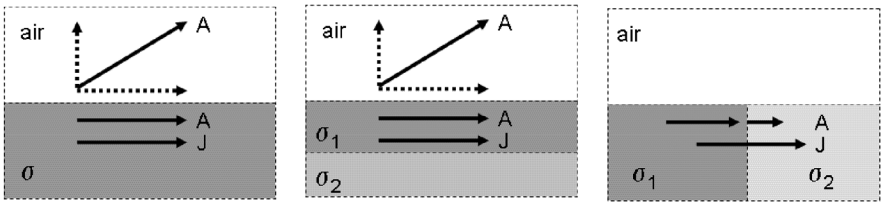


Fig. 1. Continuity of the tangential component of $\vec{A}$ on the air-metal interface. The graphics show $J$ and $A$ in the metal directly below the air-metal interfrace. In each case there is no current flow across the air-metal interface. In the left and middle figure, the tangential component of $\vec{A}$ is continious at the air-metal interface. In the right case, the tangential component of $\vec{A}$ is not continous at the "triple point," and the $\vec{A}$-field Biot–Savart law cannot be applied.

where $\mathbf{h}$ and $\mathbf{j}$ are vectors of degrees of freedom, and the matrices are given by

$$\mathbf{Z}_{ij} = \int_{\Gamma}\int_{\Omega'} \frac{1}{4\pi} \frac{\vec{W}_i^2(x') \times (x-x') \cdot \vec{W}_j^1(x)}{|x-x'|^3} d\Omega' d\Gamma \quad (49)$$

$$\mathbf{M}_{ij}^s = \int_{\Gamma} \vec{W}_i^1(x) \cdot \vec{W}_j^1(x) d\Gamma. \quad (50)$$

The matrix $\mathbf{M}^s$ is a square "mass matrix" defined only over the surface; it is extremely sparse and well-conditioned and is not an issue computationally. The matrix $\mathbf{Z}$ is a completely full rectangular matrix that maps volume currents to surface fields. A similar equation for the $\vec{A}$-field field is given by

$$\mathbf{M}^s \mathbf{a} = \mathbf{Y}\mathbf{j} \quad (51)$$

where

$$\mathbf{Y}_{ij} = \frac{\mu}{4\pi} \int_{\Gamma}\int_{\Omega'} \frac{\vec{W}_i^2(x') \cdot \vec{W}_j^1(x)}{|x-x'|} d\Omega' d\Gamma. \quad (52)$$

The computation of the Biot–Savart matrices involves singular and near singular integrals. The surface integration is performed using standard Gaussian quadrature points for each surface element. The volume integration uses an adaptive integration rule, which varies the order of the Gaussian quadrature based on the distance between the source point $x'$ and the observation point $x$. When the surface element containing $x$ is a face of the volume element containing $x'$, a highly accurate height-based singularity cancellation quadrature rule is used [31].

Here we consider the coupling of the finite-element equations (33)–(37) with the $\vec{A}$-field Biot–Savart equation (51). An implicit backward-difference method is used to update the degrees of freedom $\mathbf{a}$ and $\mathbf{j}$ simultaneously

$$(\mathbf{M}^1(\sigma) + \Delta t \mathbf{S}^1(\mu^{-1}))\mathbf{a}_{n+1} = \mathbf{M}^1(\sigma)\mathbf{a}_n - \Delta t \mathbf{D}^{01}(\sigma)\mathbf{v} + \Delta t \mathbf{M}^1\,(1.0)\mathbf{j}_s \quad (53)$$

$$\mathbf{Y}^{12}\mathbf{j}_{n+1} - \mathbf{M}^1 \mathbf{a}_{n+1} = 0 \quad (54)$$

$$\Delta t \mathbf{M}^2 \mathbf{j}_{n+1} + \mathbf{H}^{12}(\sigma)\,\mathbf{a}_{n+1} = \mathbf{H}^{12}(\sigma)\mathbf{a}_n - \Delta t \mathbf{H}^{12}(\sigma)\mathbf{K}^{01}\mathbf{v} + \Delta t \mathbf{H}^{12}(1.0)\mathbf{j}_s. \quad (55)$$

In the above equation, the voltage and the independent current source are analytical functions evaluated at time $n+1/2$. Note that the third equation follows from the definition of total electric current density

$$\vec{J} = \sigma\left(-\frac{\partial \vec{A}}{\partial t} - \nabla\phi\right) + \vec{J}_s. \quad (56)$$

It is instructive to express the update equation in matrix form. The result is shown in (57) at the bottom of the page. The volume $\vec{A}$-field and the surface $\vec{A}$-field degrees of freedom have been separated for clarity. This system of equations can be solved using an iterative method such as GMRES. But as written this linear system is poorly conditioned. First, note that $\mathbf{a}$ and $\mathbf{j}$ have different units, they differ by a factor $\Delta t/\sigma$. Therefore, every column of the matrix that multiplies $\mathbf{j}$ should be scaled by $\sigma/\Delta t$. Second, note that some of the matrix entries involve $\mathbf{M}^1(\sigma)$ while others involve $\mathbf{M}^1(1)$. This can be fixed by scaling both sides of the Biot–Savart law by $\sigma$. This scaling can be written in standard preconditioner form as shown in (58) at the bottom of the next page.

For this method to be stable we must examine the eigenvalues of the amplification matrix. We do not present an analytical

$$\begin{bmatrix} (\mathbf{M}^1(\sigma)+\Delta t\mathbf{S}^1(\mu^{-1})) & (\mathbf{M}^1(\sigma)+\Delta t\mathbf{S}^1(\mu^{-1})) & 0 \\ 0 & \mathbf{M}^s & -\mathbf{Y} \\ \mathbf{H}^{12}(\sigma) & \mathbf{H}^{12}(\sigma) & \Delta t\mathbf{M}^2(1) \end{bmatrix} \begin{bmatrix} \mathbf{a}_{n+1}^v \\ \mathbf{a}_{n+1}^s \\ \mathbf{j}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{M}^1(\sigma) & \mathbf{M}^1(\sigma) & 0 \\ 0 & 0 & 0 \\ \mathbf{H}^{12}(\sigma) & \mathbf{H}^{12}(\sigma) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_n^v \\ \mathbf{a}_n^s \\ \mathbf{j}_n \end{bmatrix} + \begin{bmatrix} -\Delta t\mathbf{D}^{01}(\sigma)\mathbf{v} + \Delta t\mathbf{j}_s \\ 0 \\ -\Delta t\mathbf{H}^{12}(1)\mathbf{K}^{01}\mathbf{v} + \Delta t\mathbf{j}_s \end{bmatrix} \quad (57)$$

proof of stability, but we have numerically computed the complete set of eigenvalues of the amplification matrix for several problems, and with different values of $\Delta t$, and in each case the eigenvalues are within or on the unit circle; hence the method is stable. The eigenvectors corresponding to eigenvalues on the unit circle correspond to steady-state solutions of the diffusion equations. It should be noted that we investigated lagging the Biot–Savart law in time, i.e., replacing $\mathbf{j}^{n+1}$ with $\mathbf{j}^n$ in (54). This has the advantage of moving the Biot–Savart matrix $\mathbf{Y}$ from the left-hand side to the right-hand side. Unfortunately, this method was unstable *for small time steps*. This method may be applicable if the goal is to quickly reach some steady-state solution, but it is not applicable if the goal of the simulation is to resolve the diffusion dynamics.

There is an issue with the independent current source $\vec{J}_s$. We allow for this source because it is convenient for many problems. However, this source cannot be arbitrary, it is a requirement that this current source be in the range of the $\nabla \times \mu^{-1}\nabla \times$ operator. For example, if the user specifies some current $\vec{J}_s = \nabla \times \mu^{-1}\nabla \times \vec{F} + \delta$, where $\vec{F}$ is arbitrary and $\delta \in (\text{range}(\nabla \times \mu^{-1}\nabla\times))^{\perp}$, the numerical solution will exhibit unbounded linear growth. This linear growth term $\vec{A}_{\text{unbounded}}$ is a solution of $\partial\vec{A}/\partial t = \delta$, and is not an artifact of the proposed finite-element method, it is a valid mathematical solution of the model equations. To eliminate this, care must be taken to ensure the independent current source is divergence free.

## V. Low-Rank QR Acceleration

In this section, we develop the hierarchical low-rank QR approximation of dense matrices representing the Biot–Savart law; the same algorithm is applicable to either (46) or (47). The algorithm has been implemented and tested on a parallel computer consisting of $K$ identical processors, e.g., a cluster of PCs. For uniprocessor computations the parameter $K$ can be considered an arbitrary partitioning of the problem, e.g., the number of computational threads.

We assume that the volume $\Omega$ has been partitioned into $K$ partitions, with each partition having an equal number of volume elements. The volume elements are distributed via the partitioning. The surface $\Gamma$ is also partitioned into $K$ equally sized surface partitions. Note, however, that the surface elements are not distributed via the surface partitions; each processor can access the entire surface mesh. The Biot–Savart matrix is then decomposed into a $K \times K$ block matrix, with every block $Z^{pq}, p \in \{1 : K\}, q \in \{1 : K\}$ representing the interaction of surface partition $\Gamma_p$ with volume $\Omega_q$. The $q$th processor computes blocks $Z^{pq}, p = 1 : K$, i.e., a column of blocks. Note that the matrix is decomposed via a partitioning of elements, hence the matrices $Z^{pq}$ are overlapping in DOF space. For the highest-level partitioning of the elements we employ a graph-based algorithm [32]. This may not be optimal for compression of the Biot–Savart matrix, but it is optimal for the FEM part of the problem. Regardless of the particular partitioning algorithm, the key point is that if the partitions $\Gamma_p$ and $\Omega_q$ are well-separated then the submatrix $Z^{pq}$ will have a low-rank QR decomposition. The procedure for computing the low-rank QR decomposition is described in Section V-A. We define "well-separated" as follows: the bounding spheres for the element partitions $\Gamma_p$ and $\Omega_q$ are computed, if the bounding spheres do not intersect then the partitions are considered well-separated and a low-rank QR representation of $Z^{pq}$ is computed. We employ a recursive procedure for computing $Z^{pq}$ when partitions $\Gamma_p$ and $\Omega_q$ are not well-separated. This results in a hierarchical representation for $Z$. If $\Gamma_p$ and $\Omega_q$ are not well separated, $\Omega_q$ is divided into eight equally sized subpartitions, $\Gamma_p$ is divided into four equally sized subpartitions, and the "well-separated test" is applied to the subpartitions $\Gamma_{pi}$ and $\Omega_{qj}, i = 1 : 4, j = 1 : 8$. A space-filling curve algorithm is used for creating the subpartitions. The process is applied recursively, with a low-rank QR representation computed for well-separated subpartitions. The recursion is halted when a volume subpartition contains fewer than some number of elements, for example 64 elements. If at the lowest level of recursion the interaction is not well separated, it is simply represented by a dense matrix. This is illustrated in Fig. 2.

No parallel communication is required in the construction of the hierarchical Biot–Savart matrix; each processor has the elements that it needs to perform the integrals. Each processor has the same amount of work, hence the computation of the integrals is load balanced. Note, however, that in the low-rank QR approximation the rank $k$ is computed dynamically, and the rank $k$ depends upon the geometry. Hence, the application of the hierarchical Biot–Savart matrix, i.e., the matrix-vector multiplication $\bar{\beta} = \mathbf{Z}\bar{\xi}$, may not be perfectly load balanced. Also note that the application of the hierarchical Biot–Savart matrix does require parallel communication. This communication is as follows: 1) each processor $q$ does a gather operation to get the values of $\bar{\xi}$ that it needs; 2) each processor $q$ loops over the sub-matrices $Z^{pq}, p = 1 : K$ and computes $\bar{\beta}_q = \mathbf{Z}^{pq}\bar{\xi}_q$; and 3) each processor participates in a global reduction on $\bar{\beta}_q$.

### A. Low-Rank QR Decomposition

Every matrix $Z$ has a QR decomposition, $Z = QR$, where $Q$ is a unitary matrix (orthonormal columns), and the matrix $R$ is an upper triangular matrix. These QR decompositions are standard in computational linear algebra as they are key steps in solving least-squares problems and eigenvalue problems. Given a matrix $Z$, there are well-known algorithms such as

$$\begin{bmatrix} I & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \left(\mathbf{M}^1(\sigma) + \Delta t \mathbf{S}^1\left(\mu^{-1}\right)\right) & \left(\mathbf{M}^1(\sigma) + \Delta t \mathbf{S}^1\left(\mu^{-1}\right)\right) & 0 \\ 0 & \mathbf{M}^s & -\mathbf{Y} \\ \mathbf{H}^{12}(\sigma) & \mathbf{H}^{12}(\sigma) & \Delta t \mathbf{M}^2(1) \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \sigma/\Delta t \end{bmatrix} \tag{58}$$

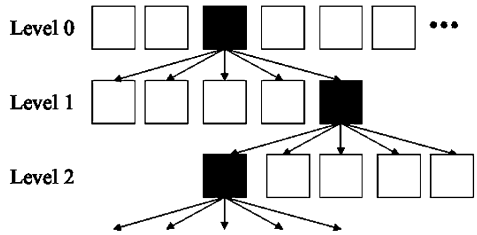


Fig. 2. Hierarchical partitioning of the Biot–Savart matrix. The highest level of partitioning is based on the number of processors. Some of the the interactions at any level $l$ will be full rank (black boxes), and these interactions are subpartitioned by decomposing the corresponding subvolume and subsurface to create Level $l+1$.

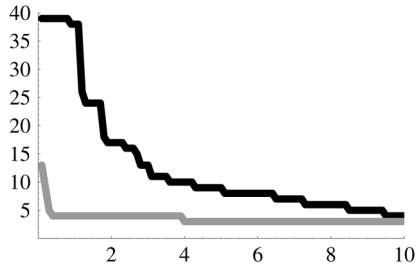


Fig. 3. Rank versus distance. The Black curve is the maximum rank, which is used to determine $s$. The gray curve is the minimum rank. This data is for a thresh of 0.001.

Householder, fast Givens, and modified Gram–Schmidt for computing the QR decomposition [33]. For our application, when $\Gamma_p$ and $\Omega_q$ are well separated the matrix $Z^{pq}$ will have a low-rank decomposition

$$Z^{pq}_{m\times n} \approx Q_{m\times k} \times R_{k\times n} \tag{59}$$

where $k$ is the rank. Clearly, if $m$ and $n$ are large and $k$ is small, the QR decomposition represents a significant compression of the matrix. We do not want to form the entire $Z^{pq}$ and then compress it, rather we sample the matrix by picking $s$ rows and columns of $Z^{pq}$, where $s$ is some predetermined number based on an estimate of the rank. Several algorithms have been proposed for picking the sampled rows and columns. The procedure for picking the sampled rows and columns is ad hoc, the procedure that we employ is described in Section V-B, and is similar to an algorithm that has been successfully applied to electrostatic [21] and frequency-domain [22] boundary integral equations. The sampling procedure is solely linear algebra, the implementation is independent of the particular Green's function, finite-element basis functions, etc., and hence can be used for either (48) or (51). Note that while the *implementation* of the low-rank QR algorithm is independent of the particular Green's function and finite-element basis functions, the *performance* of the algorithm does indeed depend upon these particulars. For the low-rank QR algorithm to be robust we must have $s$ greater than the expected rank, i.e., we over-sample the Biot–Savart matrix. The algorithm for computing $Q_{m\times k}$ and $R_{k\times n}$ is as follows.

1) Form the sampled column matrix $S^c_{m\times s}$ and the sampled row matrix $S^r_{s\times n}$.
2) Compute the rank-revealing QR decomposition $\tilde{Q}_{m\times s}\tilde{R}_{s\times s} = S^c_{m\times s}$ using LAPACK routines DGEQPF and DORGQR. The LAPACK subroutine library is public domain and is described in [34]. The rank $k$ is determined by the criteria $\tilde{R}_{kk} < \text{thresh}\cdot\tilde{R}_{11}$ where thresh is a threshold value. Keep only $k$ columns of $\tilde{Q}$, denote this as $Q_{m\times k}$, and discard $\tilde{R}$.
3) Form a new matrix $\hat{Q}_{s\times k}$ by taking $s$ rows of $Q_{m\times k}$, the exact same rows as used to construct $S^r$.
4) Compute the least-squares solution to $\hat{Q}_{s\times k}R_{k\times n} = S^r_{s\times n}$ using LAPACK routine DGELSS.

At this point we have the desired matrices $Q_{m\times k}$ and $R_{k\times n}$ which approximate $Z^{pq}_{m\times n}$. To perform a matrix-vector multiplication with the compressed matrix, it is necessary to include the permutations due to the column and row sampling, $Z^{pq}_{m\times n} \approx P^c \times Q_{m\times k} \times R_{k\times n} \times P^r$, where $P^r$ and $P^c$ are permutation matrices. The quality of the approximation, and the amount of compression (the rank $k$), are determined by the value of thresh used in Step 2 above. Our approach, being based on highly tuned LAPACK routines, is efficient both in terms of FLOPS and memory usage. The complexity of a single QR decomposition is $O(m\cdot s)+O(s\cdot n)$, using a fixed value of $s$ yields a linear complexity in $m$ and $n$. In our particular implementation the cost of computing the sampled rows and columns dominates over the cost of the LAPACK routines. This obviously depends upon the accuracy of the quadratures employed and on the data structures used to access the elements, and hence may not be true in all implementations.

The two key parameters in the QR decomposition are the threshold used to determine the rank $k$ in Step 2, and the number of sampled rows and columns $s$ in Step 1. In practice we have found $0.001 < \text{thresh} < 0.01$ to yield acceptable results, meaning the the error in the QR compression is less than the intrinsic error of the finite-element computation. Of course this is application dependent. For the parameter $s$ we use a table lookup, where the argument is the normalized distance between the volume and surface regions. This normalized distance is defined as the distance between the centroids of the two bounding spheres divided by the sum of the radii of the spheres, a value of $d = 1$ means that the two bounding spheres are just touching. A table of computed ranks versus distance, generated by running a dozen different problems, is shown in Fig. 3.

### *B. Row and Column Sampling*

The procedure for computing the sample row matrix $S^r_{s\times n}$ and column matrix $S^c_{m\times s}$ is important; the goal is to pick the most independent rows and columns. An ad hoc procedure that works well is described here. The process is initiated by simply computing the first, middle, and last row of $Z^{pq}_{m\times n}$, and the first, middle, and last column. These will be independent based on the fact that the volume and surface elements are sorted via a space-filling curve. The general step for computing a new row, given $s$ previously computed rows and columns, is as follows.

1) Define the matrix $T_{s\times s}$ as the intersection of $S^r_{s\times n}$ and $S^c_{m\times s}$.
2) Define the vector $X_i$ as the $i$th row of $S^c_{m\times s}$.
3) Define the vector $V_j$ as the $j$th row of $T_{s\times s}$.
4) Define the angle $\theta_{ij}$ as the angle between vectors $X_i$ and $V_j$.

5) The next row to compute is given the integer $i$ defined by $\max_{i=1:n}(\min_{j=1:s}\theta_{ij})$, i.e., find the vector $X_i$ that is "most different" from all vectors $V_j$.

This procedure scans the existing sampled data and determines a good candidate for the next row to compute. The same procedure is used for determining what column to compute next. We do not claim that the procedure is optimal; in fact, the reason for requiring oversampling, i.e., $s > k$, is that the sampling procedure is imperfect. If we employed $s = k$, where $k$ was the actual rank, the low-rank QR approximation would likely not satisfy the required error tolerance *thresh*.

## VI. Computational Results

### A. Spherical Shell of Current

In this section, (57) is solved for an eddy-current problem with an exact analytical solution to verify the validity of the approach. For this verification experiment, we construct the exact solution to the problem of a spherical shell of current. There exists a well-known exact analytical solution to the problem of computing the magnetic field $\vec{B}$ due to a sphere of uniform magnetization $\vec{M}$, i.e., a spherical magnet [35]. The $\vec{A}$-field due to a given magnetization $\vec{M}$ is given by

$$\vec{A}(x) = \int_V \frac{\vec{\nabla}' \times \vec{M}(x')}{|x - x'|} dx' + \oint_S \frac{\vec{M}(x') \times \vec{n}'}{|x - x'|} da'. \quad (60)$$

Note that in the first term $\vec{\nabla}' \times \vec{M}(x')$ is an effective volume current density, and in the second term $\vec{M}(x') \times \vec{n}'$ is an effective surface current density. Therefore, the problem of a sphere of uniform $\vec{M}$ is equivalent to the problem of an infinitely thin spherical shell of current. We begin with this solution, and we integrate with respect to $r$ to obtain the solution to a finite thickness spherical shell of uniform current density. In spherical coordinates $\{r, \theta, \phi\}$, the current source is given by

$$J_r = 0$$
$$J_\theta = \begin{cases} r < a & 0 \\ a \le r \le b & M \sin(\phi) \\ r > b & 0 \end{cases}$$
$$J_\phi = 0$$

and the steady-state $\vec{A}$-field is given by

$$A_r = 0$$
$$A_\theta = \begin{cases} r < a & M\mu \sin(\phi) \frac{r(b-a)}{3} \\ a \le r \le b & M\mu \sin(\phi) \frac{r^3(4b-3r)-a^4}{12r^2} \\ r > b & M\mu \sin(\phi) \frac{b^4-a^4}{12r^2} \end{cases}$$
$$A_\phi = 0$$

where $a$ and $b$ denote the inner and outer radii of the spherical shell, respectively. A time-dependent solution can be created by constructing a current source that smoothly ramps up to the steady-state value. In this verification simulation, we use the ramp function

$$R(t) = \frac{1}{2}\left(\mathrm{Erf}\left[\eta\left(\frac{2t}{\tau} - 1\right)\right] + 1\right) \quad (61)$$

where Erf denotes the error function. If we let $J_\theta$ and $A_\theta$ denote the steady-state solution, then the full time-dependent solution is given by

$$A_\theta(t) = A_\theta R(t) \quad (62)$$
$$J_\theta(t) = J_\theta R(r) + \sigma A_\theta \frac{dR(t)}{dt}. \quad (63)$$

We use $\mu = 4\pi 10^{-7}$ and $\sigma = 10^7$ in the spherical shell. We choose $\eta = 4.0$ and $\tau = 1.0$ s for the ramp function. The initial condition is $\vec{A} = 0$, this is acceptable since the exact $\vec{A}$-field is essentially zero due to our choice of ramp function parameters. The spherical shell has $a = 0.8$ m and $b = 1.0$ m. The computational mesh had a total of 6000 elements, this is for the metal region only, as the fields in the air are accounted for by the Biot–Savart law. We used $\Delta t = 10^{-1}$ for 100 time steps. The GMRES algorithm was used to solve the linear system at each time step, and approximately 330 iterations were required for a tolerance of $10^{-8}$.

Fig. 4 shows the geometry and the computational mesh, with a section removed so that the interior can be seen. Figs. 5 and 6 show a slice of the steady-state $\vec{A}$-field and $\vec{B}$-field, respectively. Figs. 7 and 8 show a slice of the the steady-state $\vec{B}$-field vectors and $\vec{B}$-field $L_2$ error, respectively. The peak relative error (error energy per element divided by energy per element) was 0.001, and the total relative error (total error energy divided by total energy) was $10^{-6}$, which is excellent. Fig. 9 shows the source current, the eddy current, and the total current versus time for a particular mesh element. The chosen time step $\Delta t = 10^{-1}$ was significantly less than the diffusion time $\mu\sigma L^2 \approx 3.0$ s. This was not required for stability, as the method is unconditionally stable; rather this was required in order for the computed eddy currents to be time-accurate. Finally, Fig. 10 shows the time history of the computed $\vec{A}$-field versus the exact $\vec{A}$-field for a selected mesh element. Note that the agreement is excellent.

### B. Coaxial Rings

This computational experiment is of a set of three coaxial metal rings, this experiment verifies the performance of the proposed algorithm for multiple conductors. The middle ring is driven with a prescribed electric current density, which generates a magnetic field, which in turn induces eddy currents in the other conductors. This problem does not possess an exact analytical solution, instead we compare the Biot–Savart approach to a pure finite-element approach that employs meshing a large region of air surrounding the conductors. The air region extends out a distance of 4 radii. The air region is large enough so that we can assume $n \times \vec{A} = 0$ on the outer boundary. The computational mesh is shown in Fig. 11. We used $\mu = 4\pi 10^{-7}$ and $\sigma = 10^7$ for the metal rings. For the FEM with Air approach we solved (33)–(37) using a value of $\sigma = 1.0$ in the air.

For each case all the fields were initialized to zero. The problem was driven by a constant current density in the middle ring of

$$J_r = 0$$
$$J_\theta = 1.0$$
$$J_z = 0$$

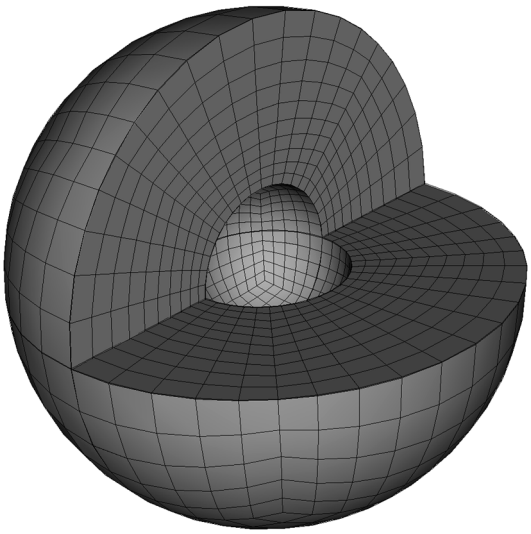

Fig. 4. Computational mesh of the spherical shell, with a portion removed for clarity.

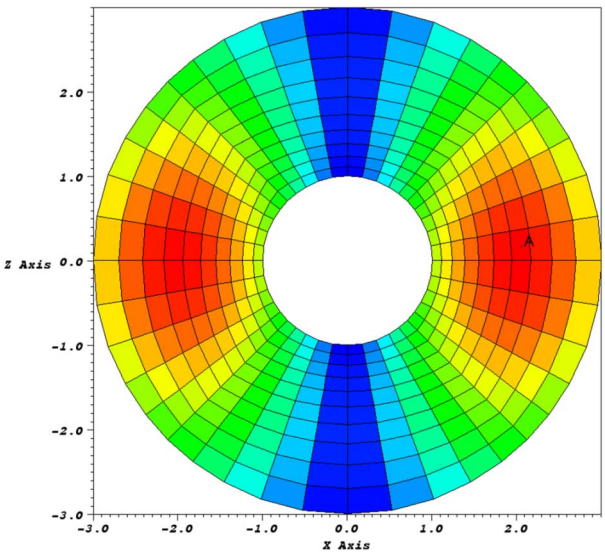


Fig. 5. Slice showing the steady-state $\vec{A}$-field magnitude.

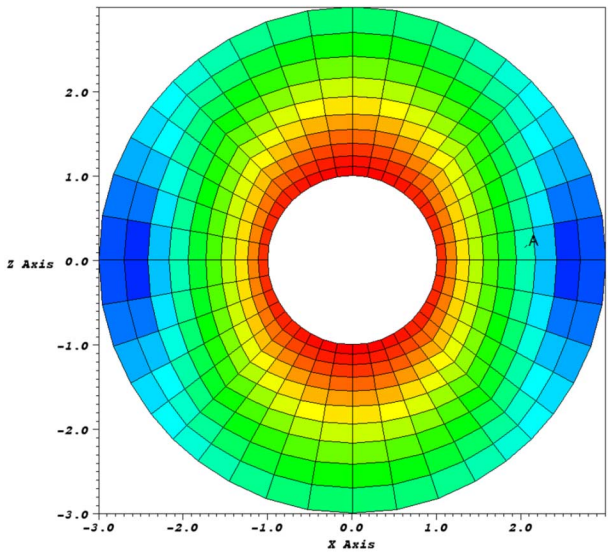


Fig. 6. Slice showing the steady-state $\vec{B}$-field magnitude.

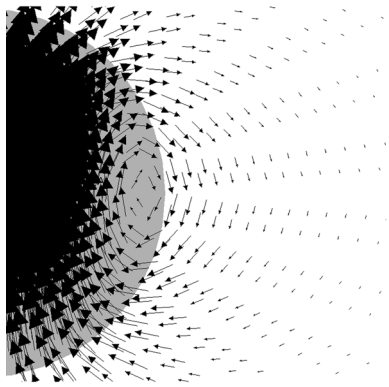

Fig. 7. Slice showing the steady-state $\vec{B}$ vectors.

that was turned on at $t = 0$. For each case a time step of $\Delta t = 0.1$ was used for 300 steps. The steady state $A$-field and $B$-field are shown for the FEM with Air model in Figs. 12 and 13. These results are compared to the steady-state results for the Biot–Savart approach. Examining the numerical value of the fields in the metal for the two cases, the worst case difference between the two cases was 1.6% which is very good. It is quite possible that the FEM with Biot–Savart law is the more accurate result, as this model does not force the fields go to zero after a distance of 4 radii.

At early times there are large eddy currents in the rings, at later times the eddy currents decay to zero, leaving no current in the top and bottom ring and constant current in the middle ring. The computed currents are shown in Fig. 14, with excellent agreement between the FEM with Air approach and the Biot–Savart approach. The $\vec{B}$-field increases with the expected $(1 - \exp(-\alpha t))$ rise time as shown in Fig. 15, again with excellent agreement between the two methods.

The FEM With Air mesh consisted of 2 064 384 elements, and the problem was solved using 64 parallel processors. For the FEM With Air approach there were over 7 million degrees of freedom; diagonally scaled Conjugate Gradient was used to solve the linear system with on average 650 iterations per time step. The Biot–Savart approach required meshing only the metal rings using 20 736 elements. There were fewer than 70 000 degrees of freedom; GMRES was used to solve the linear system with on average 390 iterations per time step. The Biot–Savart matrix was a dense $27648 \times 69120$ matrix that was compressed by a factor of $52\times$ using the hierarchical low-rank QR compression of Section V. A threshold of 1.0e-3 was used for the QR compression. It is interesting to note that the hierarchical QR decomposition resulted in 832 dense matrices (near interactions) and 3842 QR matrices (far interactions), these with an average rank of 15. The memory usage was balanced at approximately 30 MWords for both the dense and the QR matrices. This indicates that increasing the QR threshold, while compressing the QR matrices further, would not have a significant impact on the overall compression. It is difficult to directly compare the

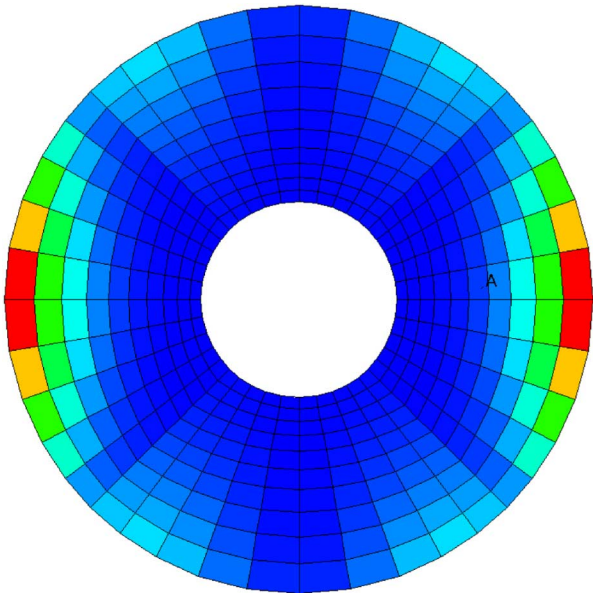


Fig. 8. Slice showing the steady-state $L_2$ error per element in the $\vec{B}$-field. The error is largest where the field varies most rapidly, as expected. The peak error was on the order of $10^{-3}$.

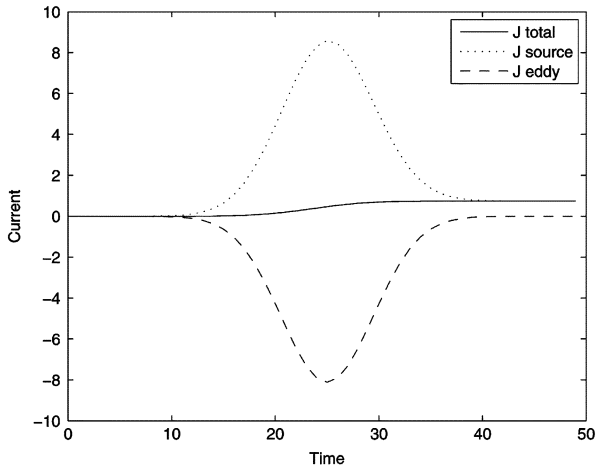


Fig. 9. Time-history plot of the $\theta$-components of the currents for a selected mesh element in the middle of the shell. As the independent current source ramps up, so does the induced eddy current but in the opposite direction.

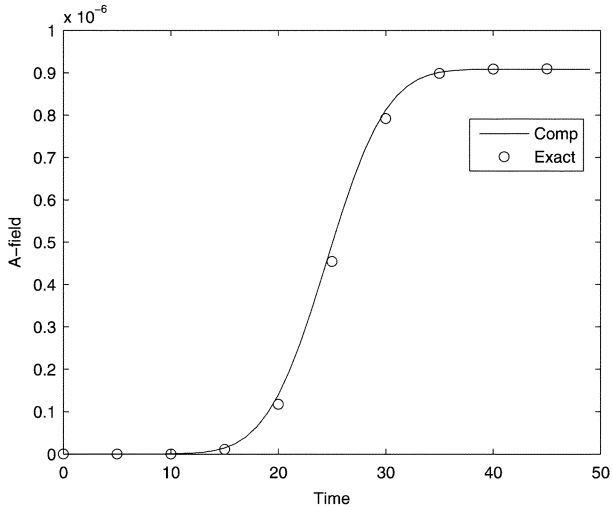


Fig. 10. Time-history plot of the computed $\vec{A}$-field and the exact $\vec{A}$-field for a selected mesh element in the middle of the shell, showing excellent agreement and verifying the validity of the approach.

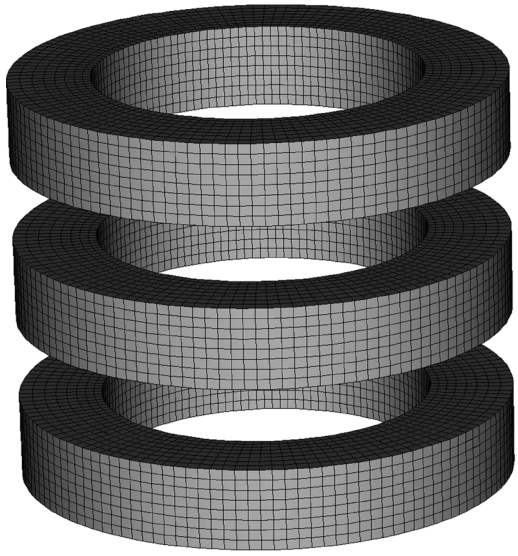

Fig. 11. Computational mesh of the coaxial rings problem. Note that only the metal is shown, for the FEM with Air model the computational mesh was extended out to a distance of 4 ring radii.

CPU times for the two different methods, since the FEM With Air problem was run on 64 processors while the Biot–Savart approach was run on 12 processors. Defining the total time as (wall clock time) $\times$ (number of processors) the Biot–Savart approach was better than 4 $\times$ faster. If we were to employ the Biot–Savart approach without the heirarchical low-rank QR compression, it would be significanlty more expensive then the FEM With Air approach due to the dense $27648 \times 69120$ matrix. Examining the computational cost of computing the heirarchical low-rank QR decomposition in more detail, we note that the cost of the LAPACK subroutines (Steps 2, 3, and 4 in Section V-A) represented only 3% of the CPU time, the bulk of the CPU time is spent in computing the sampled row and column matrices (Step 1 in Section V-A). Sampling the Biot–Savart matrix is expensive to compute partly due to the high accuracy quadrature rules we employ, and partly due to intrinsic inefficiencies of the C++ Standard Template Libary routines that are employed to provide mappings between degrees of freedom and elements.

### C. Additional QR Results

The results in Sections VI-A and VI-B demonstrate that the implicit hybrid finite-element Biot–Savart algorithm is an accurate method for solving transient eddy-current problems. In this section we summarize additional results on the performance of the hierarchical low-rank QR compression of the Biot–Savart matrix. In [20], [21], and [23] it is argued that a single-level QR approximation is $O(N^{1.5})$ and a multilevel QR is asymptotically $O(N\log(N))$. While we use a slightly different partitioning algorithm due to our desire for a simple parallelization, we should expect comparable performance. We ran 16 different simulations, ranging from 3000 to 450 000 volume unknowns. These simulations consisted of four problems (spherical shell,

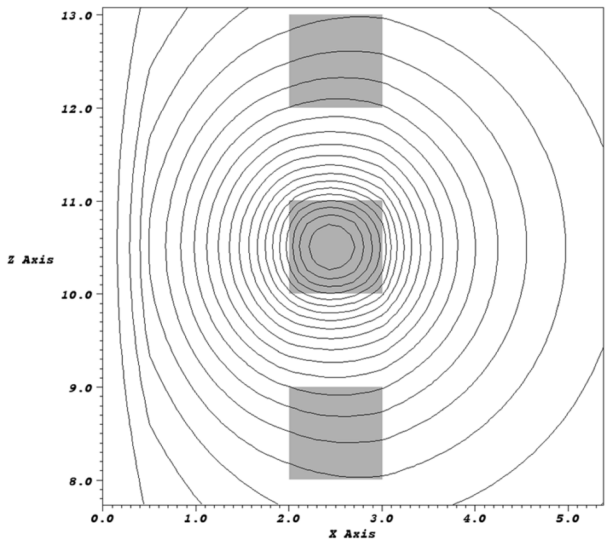


Fig. 12. $A$-field contour for the FEM with Air model. The plot is a slice through the middle of the geometry, only the right-half of the problem is shown.

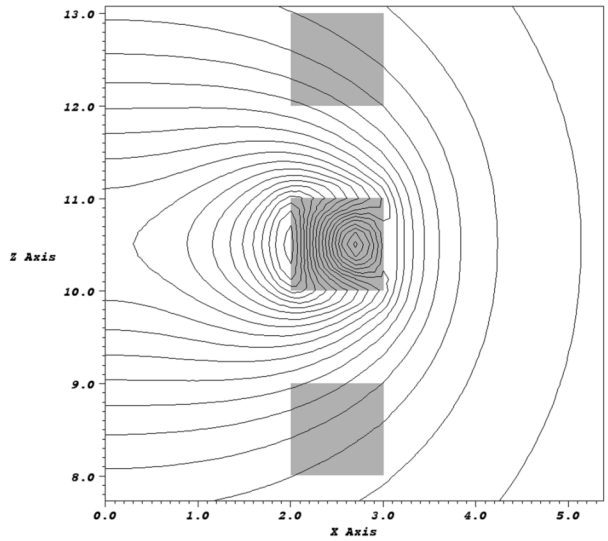


Fig. 13. $B$-field contour for the FEM with Air model. The plot is a slice through the middle of the geometry, only the right-half of the problem is shown.

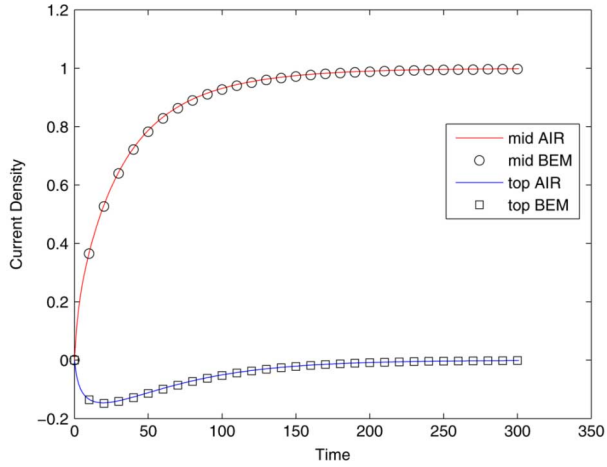


Fig. 14. Time-history plot of the currents for the coaxial ring problem. The solid line denotes the computed solution using the FEM With Air, the glyphs denote the computed solution using the Biot–Savart law. The top curves are for the total current at a point in the middle conductor, the bottom curves are for the total current at a point in the top conductor.

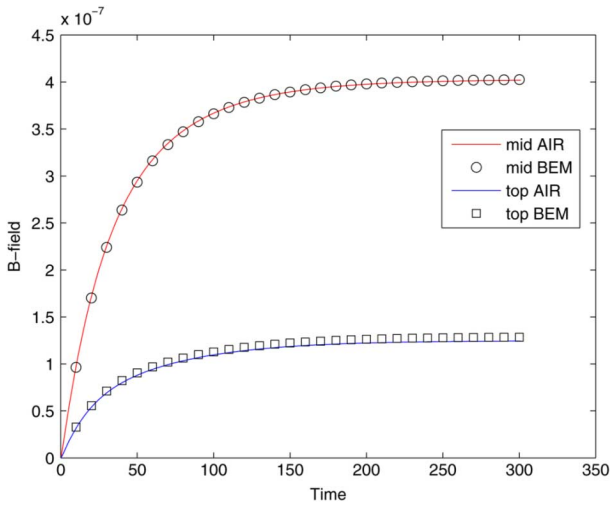


Fig. 15. Time-history plot of the $\vec{B}$-field for the coaxial ring problem. The solid line denotes the computed solution using the FEM With Air, the glyphs denote the computed solution using the Biot–Savart law. The top curves are for the $\vec{B}$-field at a point in the middle conductor, the bottom curves are for the $\vec{B}$-field at a point in the top conductor.

coaxial rings, flat sheet metal, generic railgun) at four mesh resolutions each. As each problem was refined the number of processors was increased to yield a fixed number of mesh elements per processor. The performance of the hierarchical QR algorithm for these problems is shown in Figs. 16 and 17. In Figs. 16 and 17, the QR compressed results are the actual measured memory usage and CPU time, respectively, whereas the uncompressed results are theoretical since the memory usage for the larger problems without QR compression was prohibitive. There is variation in the results due to differences in geometry and the number of parallel processors, but the overall conclusion is our algorithm is competitive with other $O(N\log(N))$ algorithms.

## VII. Conclusion

A novel numerical method for the multiply connected transient eddy-current problem has been derived, implemented, and verified. The algorithm consists of the $H(\mathrm{Curl})$-conforming finite-element discretization of the $\vec{A}-\phi$ eddy-current PDE, combined with a Biot–Savart law to specify the global boundary conditions on the conductors. The discrete Biot–Savart law is represented by a dense matrix that maps volume current to surface fields. This dense matrix is compressed using a hierarchical low-rank QR compression. The coupled system of equations is solved implicitly using the iterative GMRES method, and a scaling is proposed that significantly improves the conditioning of the system of equations. The numerical method was verified by comparing to an exact analytical solution for a simple spherical problem, and by comparing to a traditional finite-element method for a more complex problem. The method is very accurate and reasonably efficient. In our present implementation there is no motion, and hence the cost of computing the

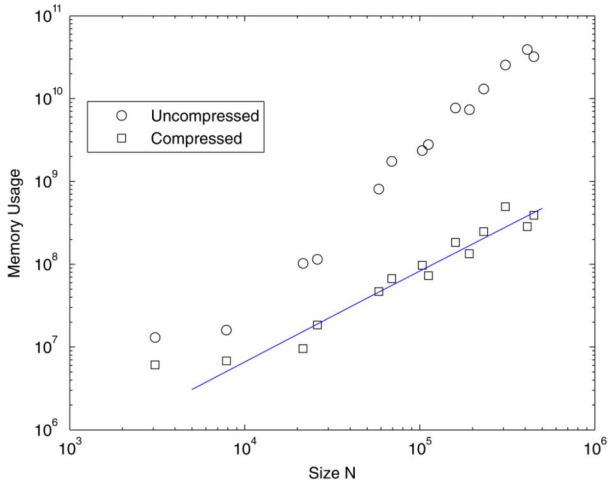


Fig. 16. Compression of the hierarchical low-rank QR algorithm. The $y$-axis is words, the $x$-axis is number of volume mesh elements. The axes are on a log scale. The solid line is is a plot of $50 \cdot N\log_2(N)$ for comparison.

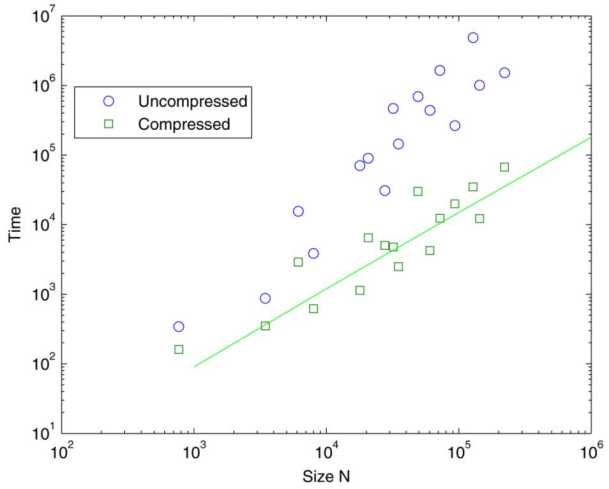


Fig. 17. Total CPU time for the hierarchical low-rank QR algorithm. This is the wall clock time (seconds) multiplied by the number of processors used. The axes are on a log scale. The solid line is is a plot of $0.3 \cdot N\log_2(N)$ for comparison.

low-rank QR approximation of the Biot–Savart law is amortized over many time steps. For applications involving moving conductors, further optimization would be required, in particular the cost of computing the sampled rows and columns of the Biot–Savart matrix.

## ACKNOWLEDGMENT

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

## REFERENCES

[1] Z. Ren, F. Bouillault, A. Razek, A. Bossavit, and J. C. Verite, "A new hybrid model using electric field formulation for 3d eddy current problems," *IEEE Trans. Magn.*, vol. 26, no. 2, pp. 470–473, Mar. 1990.

[2] Z. Ren and A. Razek, "A new technique for solving three-dimensional multiply connected eddy current problems," *IEE Proc. A—Sci. Meas. Technol.*, vol. 137, no. 3, pp. 135–140, 1990.

[3] P. Bochev, J. Hu, A. Robinson, and R. Tuminaro, "Towards robust Z-pinch simulations: Discretization and fast solvers for magnetic diffusion in heterogenous conductors," *Electron. Trans. Numer. Anal.*, vol. 15, pp. 186–210, 2003.

[4] A. Bossavit and J. C. Verite, "The "TRIFOU" code: Solving the 3d eddy current problems by using H as the state variable," *IEEE Trans. Magn.*, vol. MAG-19, no. 6, pp. 2465–2470, Nov. 1983.

[5] A. Bossavit, *Computational Electromagnetism: Variational Formulation, Complementarity, Edge Elements*. New York: Academic, 1998.

[6] O. Biro and K. Preis, "On the use of the magnetic vector potential in the finite element analysis of three-dimensional eddy currents," *IEEE Trans. Magn.*, vol. 25, no. 4, pp. 3145–3159, Jul. 1989.

[7] O. Biro, K. Preis, W. Renhart, K. Richter, and G. Vrisk, "Performance of different vector potential formulations in solving multiply connected eddy current problems," *IEEE Trans. Magn.*, vol. 26, no. 2, pp. 438–41, Mar. 1990.

[8] C. Bryant, C. Emson, and C. Trowbridge, "A comparison of Lorentz gauge formulations in eddy current problems," *IEEE Trans. Magn.*, vol. 26, no. 2, pp. 430–433, Mar. 1990.

[9] B. Weis and O. Biro, "On the convergence of transient eddy-curent problems.," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 957–960, Mar. 2004.

[10] R. Rieben and D. White, "Verification of high-order mixed finite element solution of transient magnetic diffusion problems," *IEEE Trans. Magn.*, vol. 42, no. l, pp. 25–39, Jan. 2006.

[11] K. Fujiwara, Y. Okada, T. Nakata, and N. Takahashi, "Improvements in the T-Ω method for 3–D eddy current analysis," *IEEE Trans. Magn.*, vol. 24, pp. 94–97, 1988.

[12] R. Merwa, K. Hollaus, B. Brandstatter, and H. Scharfetter, "Numerical solution o fthe general 3d eddy current problem for magnetic induction tomography," *Physiol. Meas.*, pp. 545–554, 2003.

[13] C. Daveau and M. Menad, "Mixed FEM and BEM coupling for the three-dimensional magnetostatic problem," *Numer. Meth. Partial Diff.*, 2002, Eq. 19.

[14] H. Whitney, *Geometric Integration Theory*. Princeton, NJ: Princeton University Press, 1957.

[15] J. C. Nédlec, "Mixed finite elements in R3.," *Numer. Math.*, vol. 35, pp. 315–341, 1980.

[16] J. C. Ńdlec, "A new family of mixed finite elements in R3.," *Numer. Math.*, vol. 50, pp. 57–81, 1986.

[17] P. A. Raviart and J. M. Thomas, "A mixed finite element method for 2nd order elliptic problems," in *Mathematical Aspects of the Finite Element Method*, I. Galligani and E. Mayera, Eds. New York: Springer-Verlag, 1977, vol. 606 of Lect. Notes on Mathematics, pp. 293–315.

[18] J. Jin, *The Finite Element Method in Electromagnetics*. New York: Wiley, 1993.

[19] M. S. Palma, T. K. Sarkar, L. Garcia-Castillo, T. Roy, and A. Djordjevic, *Iterative and Self Adaptive Finite-Elements in Electromagnetic Modeling*. Norwood, MA: Artech House, 1998.

[20] S. Kapur and D. Long, "IES3: Efficient electrostatic and electromagnetic solution," *IEEE Comput. Sci. Eng.*, vol. 5, no. 4, pp. 60–67, 1998.

[21] D. Gope and V. Jandhyala, "PILOT: A fast algorithm for enhanced 3d parasitic capacitance extraction," *Micro. Opt. Tech. Lett.*, vol. 41, no. 3, pp. 169–173, 2004.

[22] D. Gope and V. Jandhyala, "Efficient solution of EFIE via low-rank compression of multilevel predetermined interactions," *IEEE Trans. Antennas Propag.*, vol. 53, no. 10, pp. 3324–3333, Oct. 2005.

[23] T. Ozdemir and J.-F. Lee, "A low-rank qr algorithm for matrix compression in volume integral equations," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1017–1020, Mar. 2004.

[24] H. Ammari, A. Buffa, and J. C. Nédélec, "A justification of eddy currents for the Maxwell equations," *SIAM J. Appl. Math.*, vol. 60, no. 5, pp. 1805–1823, 2000.

[25] P. Castillo, R. Rieben, and D. White, "FEMSTER: An object oriented class library of discrete differential forms," in *Proc. 2003 IEEE Int. Antennas and Propagation Symp.*, Columbus, OH, Jun. 2003, vol. 2, pp. 972–975.

[26] P. Castillo, J. Koning, R. Rieben, and D. White, "A discrete differential forms framework for computational electromagnetics," *Comp. Meth. Eng. Sci.*, vol. 5, no. 4, pp. 331–346, 2004.

[27] P. Castillo, R. Rieben, and D. White, "FEMSTER: An object orienetd class library of higher-order discrete differential forms," *ACM Trans. Math. Soft.*, vol. 31, no. 4, pp. 425–57, 2005.

[28] G. Deschamps, "Electromagnetics and differential forms," *Proc. IEEE*, vol. 69, no. 6, pp. 676–687, Jun. 1981.

[29] D. Baldomir, "Differential forms and electromagnetism in 3-dimensional Euclidean space R3," *Proc. IEEE*, vol. 133, no. 3, pp. 139–143, Mar. 1986.
[30] K. Warnick, R. Selfridge, and D. Arnold, "Teaching electromagnetic field theory using differential forms," *IEEE Trans. Edu.*, vol. 40, no. l, pp. 53–68, 1997.
[31] M. Khayat and D. Wilton, "Numerical evaluation of singular and near-singular potential integrals," *IEEE Trans. Antennas Propag.*, vol. 53, no. 10, pp. 3180–3190, Oct. 2005.
[32] G. Karypis and V. Kumar, "A parallel algorithm for multilevel graph partitioning and sparse matrix ordering," *J. Parallel Distrib. Comp.*, vol. 48, no. 1, pp. 71–95, 1998.
[33] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1989.
[34] E. Anderson, LAPACK User's Guide 2.0. SIAM 1994.
[35] J. D. Jackson, *Classical Electrodynamics*, 1st, 2nd ed. New York: Wiley, 1962, 1975.

Manuscript received May 18, 2006; revised December 13, 2006. Corresponding author: D. White (e-mail: white37@llnl.gov).

**Daniel A. White** (M'88) received the B.S. and M.S. degrees in electrical and computer engineering, and the Ph.D. degree in applied science, from the University of California at Davis, in 1985, 1986, and 1997, respectively.

From 1986 to 1993 he was a Senior Engineer with Hughes Missile Systems Company (formerly General Dynamics Convair Division), San Diego, CA. At Hughes, he was involved with inverse synthetic aperture radar imaging, radar target identification, and low-observables technology. In 1993 he joined Lawrence Livermore National Laboratory, Livermore, CA, where he is currently Group Leader of the Computational Engineering Group. His current research interests include higher-order methods for computational electromagnetics, multi-physics simulations, and massively parallel computing.

**Ben Fasenfest** received the B.S. and M.S. degrees in electrical engineering from the University of Houston, Houston, TX, in 2002 and 2004, respectively.

He is currently an Engineer at Lawrence Livermore National Laboratory, Livermore, CA. His interests include computational electromagnetics, hybrid FEM–BEM problems, and large-scale electromagnetic modeling.

**Robert N. Rieben** (M'03) received the B.S. degree in physics from the University of California at Riverside in 1999 and the Ph.D. degree in engineering/applied science from the Department of Applied Science, University of California at Davis.

He is currently a Computational Physicist in the scientific B-Division code group at the Lawrence Livermore National Laboratory, Livermore, CA. His research interests include massively parallel physics simulations using high order numerical methods, time-domain vector finite element methods, the modeling and simulation of electromagnetic wave propagation, transient eddy-current phenomena, photonic crystals and photonic band-gap structures and magnetohydrodynamics.

**Mark Stowell** received the B.S. degree in physics in 1992 and the M.S. degree in applied mathematics in 1994 from the University of Massachusetts, Amherst.

Before joining Lawrence Livermore National Laboratory, Livermore, CA, he worked as a computer system administrator for the University of Massachusetts. His interests include computational physics, numerical analysis, and parallel computing.