Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 215 (2006) 321–347

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A finite integration method for conformal, structured-grid, electromagnetic simulation

S.J. Cooke [a,*], R. Shtokhamer [b], A.A. Mondelli [b], B. Levush [a]

[a] Naval Research Laboratory, Code 6841, 4555 Overlook Avenue S.W., Washington, DC 20375, USA
[b] SAIC, McLean, VA 22102, USA

Received 25 April 2005; received in revised form 26 September 2005; accepted 28 October 2005

Abstract

We describe a numerical scheme for solving Maxwell's equations in the frequency domain on a conformal, structured, non-orthogonal, multi-block mesh. By considering Maxwell's equations in a volume parameterized by dimensionless curvilinear coordinates, we obtain a set of tensor equations that are a continuum analogue of common circuit equations, and that separate the metrical and metric-free parts of Maxwell's equations and the material constitutive relations. We discretize these equations using a new formulation that treats the electric field and magnetic induction using simple basis-function representations to obtain a discrete form of Faraday's law of induction, but that uses finite integral representations for the displacement current and magnetic field to obtain a discrete form of Ampere's law, as in the finite integration technique [T. Weiland, A discretization method for the solution of Maxwell's equations for six-component fields, Electron. Commun. (AE U) 31 (1977) 116; T. Weiland, Time domain electromagnetic field computation with finite difference methods, Int. J. Numer. Model: Electron. Netw. Dev. Field 9 (1996) 295–319]. We thereby derive new projection operators for the discrete tensor material equations and obtain a compact numerical scheme for the discrete differential operators. This scheme is shown to exhibit significantly reduced numerical dispersion when compared to the standard linear finite element method. We take advantage of the mesh structure on a block-by-block basis to implement these numerical operators efficiently, and achieve computational speed with modest memory requirements when compared to explicit sparse matrix storage. Using the Jacobi–Davidson [G.L.G. Sleijpen, H.A. van der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, SIAM J. Matrix Anal. Appl. 17 (2) (1996) 401–425; S.J. Cooke, B. Levush, Eigenmode solution of 2-D and 3-D electromagnetic cavities containing absorbing materials using the Jacobi–Davidson algorithm, J. Comput. Phys. 157 (1) (2000) 350–370] and quasi-minimal residual [R.W. Freund, N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339] iterative matrix solution algorithms, we solve the resulting discrete matrix eigenvalue equations and demonstrate the convergence characteristics of the algorithm. We validate the model for three-dimensional electromagnetic problems, both cavity eigenvalue solutions and a waveguide scattering matrix calculation.

Keywords: Eigenvalue problems; Finite element methods; Finite integration technique

---

[*] Corresponding author. Tel.: +1 202 404 4511; fax: +1 202 767 1280.
E-mail address: simon.cooke@nrl.navy.mil (S.J. Cooke).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.10.032

1. Introduction

The numerical solution of Maxwell's equations encompasses a variety of methods and their implementations that address specific trade-offs with respect to CPU and memory resources [6]. Amongst three-dimensional (3-D), mesh-based, volumetric techniques, the finite integration technique (FIT) [1,2] is a general method particularly suited to fast evaluation of the discrete differential operators arising in the formulation, and achieves a minimal memory overhead due to the tightly localized coupling defined between field components stored on the mesh. The classical FIT formulation applies to an orthogonal, tensor-product grid, and represents each material parameter (permittivity or permeability) using a single stored numerical value for each degree of freedom of the electric or magnetic field, respectively. The method has been extended to allow structured, non-orthogonal grids [7], including eigenvalue problems [8], as well as triangular [9,10] and tetrahedral [11] meshes.

In this paper, we consider the case of a conformal, structured, non-orthogonal, multi-block mesh, using a new approach to compute the off-diagonal terms in the material operators that arise in the general non-orthogonal case. In contrast to the FIT approach, we introduce a vector basis-function representation for the electric fields as commonly used in the vector finite element method [12], and derive weighted Gaussian integration points to obtain the diagonal and (symmetric) off-diagonal material matrix components via the FIT method. This retains the compact storage of the FIT approach while improving the mesh-induced numerical dispersion. The product of this research is a new 3-D frequency-domain electromagnetic simulation code, CTLSS-2, that solves both cavity eigenvalue problems and driven waveguide scattering matrix problems in general geometries, using the Jacobi-Davidson [3] and quasi-minimal residual [5] iterative matrix algorithms, respectively. The CTLSS (cold-test, large-signal simulation) series of codes is being developed to address the problem of providing realistic, three-dimensional design tools for the vacuum electron device industry [13]. A need for electromagnetic simulation of 3-D device geometries – typically complex cavities and waveguides structures, and often incorporating highly absorbing materials – provided the motivation for the development of the original CTLSS code [14]. That research focussed on iterative methods for solving the highly non-Hermitian matrix eigenvalue problem resulting from the discretization procedure [4], and used a relatively simple electromagnetic model based on an orthogonal 3-D grid, in either Cartesian or cylindrical geometry. Here, we extend the electromagnetic model to significantly enhance accuracy through use of a conformal mesh representation of the problem geometry, using a novel hybrid of the finite integration and finite element techniques.

In Section 2, the basic frequency-domain electromagnetic equations are cast using a tensor formulation appropriate for the non-orthogonal case. A metric-free form of the differential equations is obtained, that is a continuum analogue of circuit equations. Section 3 describes the discrete representation of each kind of field in our scheme, and derives the form of the discrete differential and material operators that result. Section 4 describes the issues relating to the multi-block representation of fields and operators to complete the multi-block formulation used in the code. These include the representation of fields at block interfaces and preconditioning of the matrix equations to accelerate their solution by iterative methods. Finally, Section 5 presents a number of numerical examples to verify and validate the electromagnetic model, both for simple test problems and for real world examples from vacuum electronics device design. Conclusions are formulated in Section 6.

2. Electromagnetic theory

The basis of our discretization procedure will be a representation of the fundamental frequency-domain electromagnetic equations expressed in coordinate-independent form, using a tensor formulation appropriate for the general non-orthogonal case. Using the notion of tensor densities, a metric-free form of the differential equations is obtained. By assuming that our grid coordinates are dimensionless quantities, we demonstrate that these equations are the continuum analogue of circuit equations.

2.1. Notation

Field equations may be expressed using a variety of notations, including the commonly used vector notation of Gibbs, tensor notation, and differential forms. In this paper, we find it appropriate to use tensor notation to represent our field equations in a curvilinear coordinate system defined by a non-orthogonal numerical mesh. The symbols used in this section are summarized in Table 1. We restrict our treatment to orientable domains, and begin by reviewing the definitions necessary for our mathematical formulation.

2.1.1. Tensor representation of fields

We aim to solve Maxwell's equations on a solution domain that is specified in terms of global Cartesian coordinates, $\mathbf{r} \equiv (x, y, z)$. To facilitate the development of our formulation on a numerical mesh, we consider a localized region of this physical domain to be parameterized using a dimensionless reference coordinate system $\bar{x}^i \equiv (\bar{x}^1, \bar{x}^2, \bar{x}^3)$, and define a mapping from the reference system to the physical coordinates, i.e., $\mathbf{r} = \mathbf{r}(\bar{x}^i)$. It will prove useful to formulate our problem entirely in the reference coordinate system, from which solutions may be transformed back to a Cartesian coordinate representation. In practice, the transformation may be interpolated from a set of stored mesh coordinates, or defined using analytic expressions. Both methods are supported in our implementation.

The metrical properties of equations in the reference coordinate system are derived entirely from the coordinate transformation, which is assumed to be non-singular. We define first the local basis vectors for vector fields, and express the (symmetric) covariant metric tensor, g_{ij}, according to

$$\mathbf{e}_k = \frac{\partial \mathbf{r}}{\partial \bar{x}^k}, \quad g_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j. \tag{1}$$

These definitions are sufficient to represent an arbitrary vector field locally as a linear combination of the basis vectors, $\mathbf{A} = A^k \mathbf{e}_k$, where summation is implied over repeated indices.

A second local basis, defined by

$$\mathbf{e}^k = \nabla \bar{x}^k, \quad g^{ij} = \mathbf{e}^i \cdot \mathbf{e}^j \tag{2}$$

is a *reciprocal* basis in the sense that $\mathbf{e}_i \cdot \mathbf{e}^j = \delta_i^j$ and therefore $g_{ij} g^{jk} = \delta_i^k$.

Table 1
Notation for continuum fields and their dimensionality, using base dimensions of length $[l]$, time $[t]$, mass $[m]$ and charge $[q]$, and derived dimensions for voltage $[V] = [ml^2t^{-2}q^{-1}]$, current $[I] = [qt^{-1}]$, magnetic flux $[\Phi] = [Vt]$, capacitance $[C] = [qV^{-1}]$, inductance $[L] = [\Phi I^{-1}]$ and conductance $[G] = [IV^{-1}]$

Symbol	Dimension	Description
$i,j,k,\ldots$		Tensor component indices, $\in(1,2,3)$
$\mathbf{r}$	$[l]$	Euclidean coordinate vector
$\bar{x}^i$	$[1]$	Reference coordinates
$\mathbf{e}_i$	$[l]$	Basis vectors
$\mathbf{e}^i$	$[l^{-1}]$	Reciprocal basis vectors
ϵ^{ijk}	$[l^{-3}]$	Permutation tensor
e^{ijk}	$[1]$	Permutation symbol
g_{ij}	$[l^2]$	Metric
g^{ij}	$[l^{-2}]$	Inverse metric
$g^{1/2}$	$[l^3]$	Volume measure
$\mathbf{E}$, $\mathbf{H}$	$[Vl^{-1}]$, $[Il^{-1}]$	Electric and magnetic field vectors
E_j, H_j	$[V]$, $[I]$	Electric and magnetic field components
$\mathbf{B}$, $\mathbf{D}$, $\mathbf{J}$	$[\Phi l^{-2}]$, $[ql^{-2}]$, $[Il^{-2}]$	Magnetic induction, electric displacement and current density vectors
B^i, D^i, J^i	$[\Phi l^{-3}]$, $[ql^{-3}]$, $[Il^{-3}]$	Magnetic induction, electric displacement and current density components
$\mathfrak{B}^i$, $\mathfrak{D}^i$, $\mathfrak{J}^i$	$[\Phi]$, $[q]$, $[I]$	Tensor densities derived from B^i, D^i, J^i, respectively
ε, μ, σ	$[Cl^{-1}]$, $[Ll^{-1}]$, $[Gl^{-1}]$	Material permittivity, permeability and conductivity tensors
ε^{ij}, μ^{ij}, σ^{ij}	$[Cl^{-3}]$, $[Ll^{-3}]$, $[Gl^{-3}]$	Material permittivity, permeability and conductivity tensor components
$\mathfrak{C}^{ij}$, $\mathfrak{L}^{ij}$, $\mathfrak{G}^{ij}$	$[C]$, $[L]$, $[G]$	Tensor density components derived from ϵ^{ij}, μ^{ij}, σ^{ij}, respectively
$\bar{w}_{\mathrm{E}}$, $\bar{w}_{\mathrm{M}}$	$[qV] \equiv [\Phi I] \equiv [ml^2t^{-2}]$	Integrated electric and magnetic energies
Ω, $\Omega_{[k]}$, Γ		Domain, subdomains and boundary
$\hat{n}_j$	$[l]$	Surface normal components

Any vector field, **A**, may be represented using either the basis vectors or the reciprocal basis vectors. The components are termed, respectively, the contravariant and covariant components, and are defined as follows:

$$\mathbf{A} = A^k\mathbf{e}_k = A_k\mathbf{e}^k, \quad A^k = \mathbf{e}^k \cdot \mathbf{A}, \quad A_k = \mathbf{e}_k \cdot \mathbf{A}, \tag{3}$$

where the components are related by $A^i = g^{ij}A_j$ or $A_i = g_{ij}A^j$. Higher rank tensors quantities are derived from products of vector components. Under a change of coordinates, components transform according to the rules of tensor algebra.

2.1.2. Vector operations

The scalar product of vectors expressed using tensor components is obtained using either the metric tensor or its reciprocal to lower or raise indices as necessary,

$$\mathbf{a} \cdot \mathbf{b} = a_i b^i = a^i b_i = g_{ij}a^i b^j = g^{ij}a_i b_j. \tag{4}$$

To translate other common vector operations to tensor notation, we first define the vector triple product as a linear function of the covariant components of three vectors,

$$\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \epsilon^{ijk}a_i b_j c_k, \tag{5}$$

where the components of the tensor ϵ are defined using the reciprocal basis vectors,

$$\epsilon^{ijk} = \mathbf{e}^i \cdot \left(\mathbf{e}^j \times \mathbf{e}^k\right). \tag{6}$$

From the properties of the triple product, ϵ is cyclic in its three indices, and antisymmetric under the exchange of any pair of indices. The components therefore simplify to

$$\epsilon^{ijk} = \frac{e^{ijk}}{g^{1/2}}, \tag{7}$$

where the permutation symbol e^{ijk} is 1 for cyclic permutations of indices, -1 for anticyclic permutations and zero otherwise. The factor $g^{1/2}$, where $g = |\det g_{ij}|$, is associated with the magnitude of a volume element, and is necessary for ϵ to transform properly under coordinate transformations.

These definitions lead to the definition of the vector cross product,

$$(\mathbf{a} \times \mathbf{b})^i = \mathbf{e}^i \cdot (\mathbf{a} \times \mathbf{b}) = \epsilon^{ijk}a_j b_k, \tag{8}$$

and consequently the differential *curl* operator in tensor notation is obtained,

$$(\nabla \times \mathbf{a})^i = \epsilon^{ijk}\frac{\partial}{\partial \bar{x}^j}a_k. \tag{9}$$

2.2. Maxwell's equations

Electromagnetic fields in 3-D in the presence of material media are described by Maxwell's phenomenological equations. To develop our numerical scheme for inhomogeneous materials represented on a general curvilinear mesh, we employ tensor densities to separate the representation of the coordinate system and material parameters from the differential structure of the equations.

2.2.1. Frequency domain equations

In the frequency domain, we use the convention $\mathbf{E} \propto \mathrm{e}^{-\mathrm{i}\omega t}$, etc., for time-harmonic fields of angular frequency ω, and write Maxwell's electromagnetic equations using tensor notation:

$$\epsilon^{ijk}\frac{\partial E_k}{\partial \bar{x}^j} = \mathrm{i}\omega B^i, \tag{10}$$

$$\epsilon^{ijk}\frac{\partial H_k}{\partial \bar{x}^j} = -\mathrm{i}\omega D^i + J^i. \tag{11}$$

where E_k and H_k are the covariant components of the electric and magnetic fields, and B^i, D^i and J^i are the contravariant components of the magnetic induction, electric displacement current density and electric current density fields, respectively.

As tensor equations, the form of these equations is independent of the particular coordinate system chosen, however, the equations depend implicitly upon the metric via the tensor ϵ^{ijk}. A more convenient form can be obtained that removes any dependence on the metric, and that will therefore be more useful for numerical computation. This follows the approach of [15].

2.2.2. Metric-free equations

To remove the implicit metric-dependence in Maxwell's equations we replace some of the tensor field quantities by tensor densities. Tensor densities are closely related to tensors, but implicitly include a factor $g^{1/2}$. The transformation rules for such quantities under coordinate transformations are modified accordingly from those for regular tensors [16].

To transform Maxwell's equations to the metric-free form, we will make use of the following contravariant tensor densities,

$$\mathfrak{D}^i = g^{1/2}D^i, \quad \mathfrak{B}^i = g^{1/2}B^i, \quad \mathfrak{J}^i = g^{1/2}J^i, \tag{12}$$

where we have employed the standard notation for densities to distinguish them from regular tensors. Substituting these definitions into Maxwell's equations gives, after appropriate cancellation:

$$\mathfrak{d}^{ij}E_j = \mathrm{i}\omega\mathfrak{B}^i, \tag{13}$$

$$\mathfrak{d}^{ij}H_j = -\mathrm{i}\omega\mathfrak{D}^i + \mathfrak{J}^i, \tag{14}$$

where we define the differential operator density

$$\mathfrak{d}^{ik} = e^{ijk}\frac{\partial}{\partial\bar{x}^j}. \tag{15}$$

There is no dependence in this form of the equations, either direct or indirect, upon metrical quantities. The tensor density e^{ijk} is purely numerical, taking the same value in each coordinate system, and the equations are a metric-free expression of Maxwell's equations.

2.2.3. Boundary conditions

While Maxwell's equations hold interior to our solution region, which we denote Ω, we must specify boundary conditions on the surface of this region. The boundary is assumed to consist of regions that are either perfect electric conductor (PEC), denoted Γ_{PEC}, or perfect magnetic conductor (PMC), denoted Γ_{PMC}. On PEC surfaces, the tangential electric field is zero, while on PMC surfaces, the tangential magnetic field is zero.

The normal vector $\hat{n}$ at a surface is well defined without reference to the metric as a covariant vector $\hat{n}_j$, using the coordinate basis interior to the surface. Using this normal vector, the boundary conditions may be expressed in metric-free form as:

$$e^{ijk}\hat{n}_jE_k = 0 \quad \text{on } \Gamma_{\mathrm{PEC}}, \tag{16}$$

$$e^{ijk}\hat{n}_jH_k = 0 \quad \text{on } \Gamma_{\mathrm{PMC}}. \tag{17}$$

2.2.4. Constitutive relations

The system of electromagnetic equations may be closed by specifying material constitutive relations and Ohm's law. We treat the case of materials whose response to fields is linear, local and instantaneous, though possibly inhomogeneous and anisotropic. In terms of the original fields, these relations may be written

$$D^i = \varepsilon^{ij}E_j, \quad B^i = \mu^{ij}H_j, \quad J^i_\Omega = \sigma^{ij}E_j, \tag{18}$$

where ε^{ij} is the permittivity tensor, μ^{ij} the permeability tensor and σ^{ij} the conductivity tensor. The current density $J^i = J^i_{\mathrm{ext}} + J^i_\Omega$ has been decomposed into external and Ohmic current contributions. The components of the material tensors may be calculated from their Cartesian representations by projection on the basis vectors, for example,

$$\mu^{ij} = \mathbf{e}^i \cdot \mu \cdot \mathbf{e}^j. \tag{19}$$

Therefore *scalar*-valued material quantities are represented as tensors via $\mu^{ij} = \mu g^{ij}$, etc.

Using instead the tensor density fields, the constitutive relations become

$$\mathfrak{D}^i = \mathfrak{E}^{ij} E_j, \quad \mathfrak{B}^i = \mathfrak{L}^{ij} H_j, \quad \mathfrak{J}^i_\Omega = \mathfrak{G}^{ij} E_j, \tag{20}$$

where we have defined the new material tensor densities

$$\mathfrak{E}^{ij} = g^{1/2}\varepsilon^{ij}, \quad \mathfrak{L}^{ij} = g^{1/2}\mu^{ij}, \quad \mathfrak{G}^{ij} = g^{1/2}\sigma^{ij}, \tag{21}$$

that combine metric and material information. Therefore, the geometrical content of Maxwell's equations is incorporated entirely into the continuum constitutive relations.

2.2.5. Notation and dimensions

In defining our local coordinate system, we choose to use coordinates that are dimensionless quantities, in contrast to the physical coordinates that have dimensions of length, [*l*]. Therefore, our basis vectors $\mathbf{e}_k$ have dimension [*l*] also, while reciprocal basis vectors $\mathbf{e}^k$ have dimension $[l^{-1}]$. We must therefore be careful to translate the units associated with physical quantities when they are represented using basis coefficients.

Table 1 summarizes the field quantities and notations introduced in this section, and shows the dimensions of the components for the field tensors and tensor densities. Due to our choice of dimensionless reference coordinates, the dimensions of the components are not those of the actual quantities. By using dimensionless local coordinates and appropriate tensor densities, we have obtained essentially a continuum representation in terms of fields that relate to electrical circuit quantities. Similar parallels have been used previously for the discrete representations of fields [17,18].

In the following section, we describe the procedure used to discretize the partial differential equations, (13) and (14), and the tensor density constitutive relations (20).

3. Discretization

The problem domain may be complex in both shape and topology, including possibly curved boundaries and multiply connected cavities. Therefore, to represent an arbitrary domain, we choose to discretize the mathematical formulation on a structured, conformal, multi-block grid. This type of grid permits accurate representation of typical cavity structures and therefore accurate solutions to the equations, while retaining some of the speed and simplicity of the single-block, orthogonal, structured grid algorithm [14]. In common with that approach, we implement the differential operators independently from the material operators. Taking advantage of the grid structure, the two *curl* operations are implemented as optimized functions rather than matrices. Within each block, therefore, the algorithm is similar to the original algorithm, except that the diagonal-matrix material operators of the orthogonal-grid algorithm are replaced by more general operators that include off-diagonal terms to describe the generalized metric and material tensor density relations.

We describe first the grid model, and then the discrete representation of each type of field and operator. We will delay treatment of the interfaces that extends the model from individual blocks to the global multi-block grid implementation until the following section. We will introduce a number of discrete quantities in this section, which we summarize in Table 2. Discrete indices, usually enumerating a large number of degrees of freedom, are written in parentheses to distinguish them from tensor component indices. We will use the same summation convention for repeated discrete index variables as for tensor indices.

3.1. Multi-block grid

We decompose the physical domain into non-overlapping, hexahedral subdomains that in physical space conform to the exact boundary of the problem geometry, but that correspond in the reference coordinate system to simple rectangular bricks. As a simple example, Fig. 1 shows such a decomposition for a two-dimensional disk into five blocks; this decomposition avoids the singular points that would occur in a transformation mapping from a single square to the disk, or the coordinate singularity that must be introduced to represent the same domain using cylindrical coordinates. The logical representation consists of five blocks that are subdivided into regular arrays of cells, and connected at the surfaces, as indicated by the arrows. The mapping

Table 2
Notation for discrete variables

Symbol	Discrete quantity
$v_i^{(m)}$, $w_i^{(m)}$	Vector and dual vector field basis functions
$\hat{v}^i_{(m)}$, $\hat{w}^i_{(m)}$	Pseudo-vector and dual pseudo-vector field basis functions
$V_{(m)}$, $I_{(m)}$	Voltage and current coefficients representing electric and magnetic fields
$\Phi^{(m)}$, $Q^{(m)}$, $I^{(m)}$	Flux, charge and current coefficients representing magnetic induction, and electric charge and current density fields
$C^{(mn)}$, $L^{-1}_{(mn)}$, $G^{(mn)}$	Capacitance, reciprocal inductance and conductance operators
$K^{(mn)}$, $M^{(mn)}$	Matrix operators for the electromagnetic eigenvalue problem
$d^{(mn)}$, $\bar{d}^{(mn)}$	Discrete *curl* operators

For circuit quantities, dimensions are implied by the symbols used.

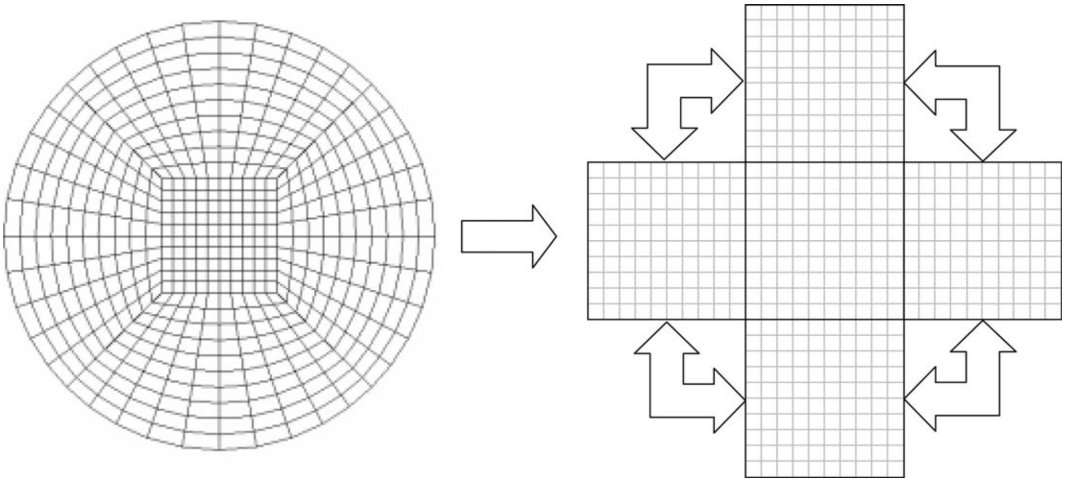

Fig. 1. Two-dimensional example demonstrating logical block connectivity for a conformal, structured grid. The arrows indicate interfaces that are logically connected in the model.

from the array of logical cells to the physical mesh is usually defined via multi-linear interpolation between coordinates stored for each node.

This kind of decomposition can in principle be generalized to arbitrarily complex geometries. The blocking and subsequent grid generation for a complex three-dimensional structure may be performed either using a commercial grid-generation tool, starting from a 3-D CAD representation, or by employing facilities provided by the code to create and interconnect blocks for simpler geometries. The first method provides a general capability for modeling complex structures, while the second is implemented in the code to allow specific, commonly used geometries to be encapsulated and parameterized as automated templates.

Discretization of the model equations is simplified by first considering the fields inside a block, and then treating the interface conditions between blocks as boundary conditions. This will require a particular interpretation of the different types of fields stored on the grid that distinguishes between local and distributed field quantities. Introducing this distinction has the advantage that the algorithm for each block requires no knowledge of the neighboring blocks.

To represent the tensor equations in each block of the mesh, it is useful to use the simplest coordinate system for each block, matching between adjacent block coordinate systems only when treating the interface conditions. Since tensor equations are valid for an arbitrary coordinate system, we are at liberty to use the logical structure of the mesh to impose a specific local coordinate system.

Each logical block is subdivided into a regular grid of unit cubic cells that map to curvilinear hexahedral cells in physical space. This subdivision defines a three-dimensional block coordinate system consisting of orthogonal, dimensionless coordinate ranges $\bar{x} \in [0, N^1] \otimes [0, N^2] \otimes [0, N^3]$, where N^i is the number of cells in the i-direction for that block. The mapping, $\mathbf{r}(\bar{x})$, from block coordinates to global Cartesian coordinates is defined separately for each block, either by interpolation of coordinates, explicitly stored at the grid nodes, or using one of a set of predefined functional forms. Such functions include the common orthogonal coordinate systems and transfinite interpolation schemes, and allow tailored versions of the algorithm to be implemented that are faster, and that require reduced storage.

The block coordinates also serve to specify the orientation of the edges and faces of the mesh cells to which specific field components will be associated. Within each block, these orientations are well defined, but additional transformations will be necessary across block boundaries.

The actual discretization procedure may be defined for a single grid cell. Since only coordinate differentials arise in the physical equations, via the metric g_{ij}, we may describe the discretization procedure for a single cell of the mesh using either local cell coordinates, $\bar{x}^i \in [0,1]$, or cell-centered coordinates $\hat{x}^i \in [-\frac{1}{2},\frac{1}{2}]$, where $\hat{x}^i = \bar{x}^i - \frac{1}{2}$. These logical coordinate systems will simplify the transformation of the physical model to a discrete set of equations.

3.2. Discrete representation of fields

Discrete equations can be obtained from the continuum formulation either by introducing a finite basis to represent fields in place of the continuum coordinate basis, as in the finite element approach, or by introducing integral quantities as the discrete variables, as in the FIT. We will combine these two approaches, using the former to discretize Maxwell's homogeneous equation in Section 3.3 and the latter for the inhomogeneous equation in Section 3.4. The two methods are used together to define the discrete material constitutive relations in Section 3.5.

To support our choice of discretization, we distinguish two types of fields. The primary fields **E** and **B** are the fundamental, microscopic electromagnetic fields of the vacuum that can be defined unambiguously at any point. Therefore, the basis representation is appropriate to represent these fields throughout the solution domain. The secondary fields **D** and **H** are the product of the averaging procedures that are necessary in transitioning to the phenomenological electromagnetic equations [19]. These fields are therefore inherently integral quantities.

There is a sense in which each of these secondary fields is dual to the corresponding primary field. For some volume Ω, we can write time-averaged electric and magnetic stored energies as the integrals:

$$\bar{w}_{\mathrm{E}} = \frac{1}{4} Re \int_{\Omega} E_i \mathfrak{D}^{i*} \, \mathrm{d}^3\bar{x} \tag{22}$$

and

$$\bar{w}_{\mathrm{M}} = \frac{1}{4} Re \int_{\Omega} \mathfrak{B}^i H_i^* \, \mathrm{d}^3\bar{x}, \tag{23}$$

respectively. These are well-defined, physical, scalar quantities irrespective of the particular coordinate system used. Therefore $\mathfrak{D}^i$ occupies a vector space that is dual to that of E_i, and similarly the vector space of magnetic fields H_i is dual to that of $\mathfrak{B}^i$ over any chosen domain Ω.

3.3. Basis functions

Basis fields are chosen to be simple polynomial functions in the reference coordinate system, and fields are represented by a linear combination of basis fields in each cell. The fields that may be represented form a finite-dimensional subset of the fields in the continuum description, and the discrete equations obtained represent a projection of the continuum equations onto this subspace.

To obtain field values in the global Cartesian coordinate system from the finite basis representation, both the coordinate and the field values are transformed, using the coordinate mapping and basis vectors used to define the metric. For example, for the vector field, **A**, the general transformation is of the form

$$\mathbf{A}(\mathbf{r}(\bar{x})) = A_{(m)} v_k^{(m)}(\bar{x}) \mathbf{e}^k(\bar{x}), \tag{24}$$

where $A_{(m)}$ is the associated coefficient and $v_k^{(m)}(\bar{x})$ is a vector-field basis function. An index in parentheses denumerates a discrete degree of freedom, and summation over repeated indices is implied. The spatial dependence of continuum fields has been displayed explicitly here to show the mapping from local to global coordinates. The possible values of the set of coefficients $A_{(m)}$ defines a finite-dimensional linear vector space. Choice of the form of the basis functions $v_k^{(m)}$ defines a particular discrete representation of the vector field.

To represent pseudo-vector fields, such as the magnetic induction, it is important to choose the discrete representation such that differential identities satisfied by continuum fields carry through to the discrete fields. Such differential identities relate to the differential properties of the underlying manifold, and do not involve the metric. Failure to satisfy these identities in the discrete operators can lead to mixing of curl-free and divergence-free solutions to Maxwell's equations, causing both spurious solutions and global errors in determined solution fields. We therefore choose to use a vector-basis that satisfies these conditions; for a finite-element mesh of hexahedral cells, the simplest robust basis set for vector fields is the set of mixed-order linear basis functions based on the work of Nédélec [20]. It is useful to define basis functions local to each cell, so that their form is independent of the cell location within a block. Continuity conditions then determine which local basis functions combine to give the global basis function associated with each degree of freedom.

3.3.1. Vector field basis functions

For vector fields, a degree of freedom is associated with each edge of the mesh, as in the Yee scheme [21]. This scheme is shown in Fig. 2, and associates each electromagnetic field component with a particular mesh or dual mesh element.

Each primary mesh cell has a local basis function associated with each of its 12 edges; the local basis functions are chosen so as to satisfy constraints of tangential continuity with the corresponding functions in adjacent cells that share a common edge. In cell-centered coordinates, the set of 12 vector basis functions inside a cell can be written,

$$\{v_i\} = \{\delta_i^l W(\pm\hat{x}^m)W(\pm\hat{x}^n) \quad \text{for } e^{lmn} = 1\}, \tag{25}$$

taking each combination of the signed terms in combination with the three cyclic permutations of the superscript indices, where δ_i^l is the Kronecker delta symbol and

$$W(x) = \frac{1}{2} + x \tag{26}$$

defines a triangular *wedge* function on the interval $x \in [-\frac{1}{2}, \frac{1}{2}]$. Each vector basis function is associated with one edge of the cell: the parallel component has constant unit amplitude along the edge, and varies linearly in the transverse direction to zero at each of the opposite faces. Tangential continuity of the global vector field is ensured if each basis function is associated with a degree of freedom associated with that edge.

The electric field inside the cell represented using this basis has the form

$$E_i = V_{(m)}v_i^{(m)} \Rightarrow \mathbf{E} = V_{(m)}v_i^{(m)}\mathbf{e}^i, \tag{27}$$

where the coefficients $V_{(m)}$ are voltages associated with cell edges, also indicated in Fig. 2.

3.3.2. Pseudo-vector field basis functions

A pseudo-vector field, such as **B**, is represented in our formulation as a contravariant vector density. In cell-centered coordinates, the compatible set of 6 pseudo-vector basis functions is

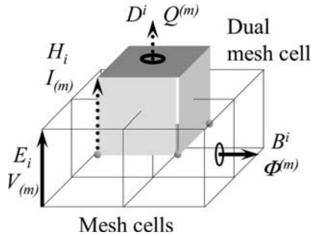

Fig. 2. Sketch showing the locations of discrete components for each type of field quantity in the Yee scheme, related to either the primary or dual mesh cells.

$$\{\hat{v}^i\} = \{\delta^i_l W(\pm\hat{x}^l) \quad \text{for } l = 1,2,3\}. \tag{28}$$

Each pseudo-vector basis function is associated with one face of the cell: the component normal to that face has constant unit amplitude over the face, and varies linearly to zero at the opposite face. Normal continuity of the global pseudo-vector field is therefore ensured if each basis function is associated with a degree of freedom associated with that face.

A magnetic induction field represented using this basis has the form

$$\mathfrak{B}^i = \Phi^{(m)}\hat{v}^i_{(m)} \Rightarrow \mathbf{B} = \frac{1}{g^{1/2}}\Phi^{(m)}\hat{v}^i_{(m)}\mathbf{e}_i, \tag{29}$$

where the coefficients $\Phi^{(m)}$ are magnetic fluxes associated with the primary mesh cell faces.

3.3.3. Maxwell's homogeneous equation

To discretize Maxwell's homogeneous equation,

$$\mathfrak{d}^{ij}E_j = \mathrm{i}\omega\mathfrak{B}^i, \tag{30}$$

we must obtain a discrete form of the differential operator $\mathfrak{d}^{ij}$. Using the above choice for basis functions, it is possible to represent the discrete form of this operator without further approximation [22]. This is possible since the *curl* of a vector basis function may be represented exactly as a linear combination of pseudo-vector basis functions, and in a manner that is compatible across cell interfaces. We will therefore write the matrix form, $d^{(mn)}$, of the differential operator $\mathfrak{d}^{ij}$ interior to a block in the form

$$\mathfrak{d}^{ij}v^{(n)}_j = d^{(mn)}\hat{v}^i_{(m)}. \tag{31}$$

To show that (31) is valid for any cell, we take for example the vector basis function associated with the edge connecting cell coordinates (0,0,0) and (1,0,0),

$$\begin{aligned}\mathfrak{d}^{ik}(\delta^1_k W(-\hat{x}^2)W(-\hat{x}^3)) &= \left[0, \frac{\partial}{\partial\bar{x}^3}W(-\hat{x}^2)W(-\hat{x}^3), -\frac{\partial}{\partial\bar{x}^2}W(-\hat{x}^2)W(-\hat{x}^3)\right]^i = [0, -W(-\hat{x}^2), W(-\hat{x}^3)]^i \\ &= -\delta^i_2 W(-\hat{x}^2) + \delta^i_3 W(-\hat{x}^3).\end{aligned} \tag{32}$$

This is the signed sum of two pseudo-vector basis functions – namely those associated with the cell faces adjacent to the chosen edge. Therefore the pseudo-vector degree of freedom associated with a face is exactly equal to a signed sum of the vector degrees of freedom associated with the four adjacent edges, exactly as in the Yee scheme [21]. By extension, the tensor *curl* of any field represented in the vector basis may be exactly represented in the pseudo-vector basis, in a strong sense.

Substituting the basis representations (27) and (29) for the fields into (30) and assuming the form (31) for the differential operator, we obtain the discrete representation,

$$(\mathfrak{d}^{ij}v^{(n)}_j)V_{(n)} = \mathrm{i}\omega\Phi^{(m)}\hat{v}^i_{(m)} \Rightarrow d^{(mn)}V_{(n)} = \mathrm{i}\omega\Phi^{(m)}. \tag{33}$$

It is highly efficient to implement this operator over a block of the structured grid as an optimized function in the code, since the adjacencies of cell components are straightforwardly computed and no reference is made to material or geometric information.

3.4. Discrete representation of dual fields

A discrete representation of a dual field is obtained by projecting the dual field onto a discrete set of test functions. We choose to use the lowest order test functions that are compatible with the integration procedure.

To describe the test functions, it is convenient to introduce a dual mesh for each block, as in the FIT method. Fig. 2 shows the relationship between the primary and dual mesh cells. In the interior mesh, dual mesh cells are shifted from the primary mesh by a half cell in each direction, so that dual mesh vertices coincide with the primary mesh cell centers, and vice versa. At the boundaries of a block, however, the block surfaces will truncate the dual mesh cells.

Note that each edge of the original mesh is uniquely associated with the dual mesh face that it intersects, while each face similarly is uniquely associated with a dual edge. Since the vector field representation associates degrees of freedom with the primary mesh edges, then by associating the dual vector field test functions with the faces of the dual mesh, the primary and dual quantities will be collocated, and we will obtain an equal number of equations and unknowns in the matrix problem.

3.4.1. Dual vector fields

The discrete representation of a dual vector field is a projection of the field onto a set of test vector field functions. In cell-centered coordinates, we choose the following set of 12 vector test functions in each cell,

$$\{w_i\} = \{\delta^l_i \delta[\hat{x}^l] H(\pm\hat{x}^m) H(\pm\hat{x}^n) \quad \text{for } e^{lmn} = 1\}, \tag{34}$$

taking each combination of the signed terms, where δ^l_i is the Kronecker delta symbol, $\delta[x]$ is the Dirac delta functional and $H(x)$ is the Heaviside step function. These 12 test functions correspond in each cell to the twelve quarter-faces of the dual mesh that intersect the cell and that all meet at the cell center. It is valid to use the Dirac functional in the definition of the test functions, since they will always be used in conjunction with integration.

Using these test functions, we can define the following projections in each cell:

$$Q^{(m)} = \int_{\text{cell}} w_i^{(m)} \mathfrak{D}^i \,\mathrm{d}^3\bar{x}, \tag{35}$$

where (m) identifies a specific face of the dual mesh. This is equivalent to a surface integral over one quarter of a dual mesh face, and evaluates the cell's contribution to the discrete charge associated with that face. The total charge may be summed from contributions from each contributing cell. In the interior of a block there will be four such contributions, but only two or one at the block surfaces and edges, respectively.

3.4.2. Dual pseudo-vector fields

In cell-centered coordinates, the set of 6 dual pseudo-vector test functions are

$$\{\hat{w}^i\} = \{\delta^i_l H(\pm\hat{x}^l) \delta[\hat{x}^m] \delta[\hat{x}^n] \quad \text{for } e^{lmn} = 1\}. \tag{36}$$

These 6 functions correspond to the six half-edges of the dual mesh that intersect a primary mesh cell and meet at the cell center.

Using these test functions, we can define the following projections in each cell:

$$I_{(m)} = \int_{\text{cell}} \hat{w}^i_{(m)} H_i \,\mathrm{d}^3\bar{x} \tag{37}$$

Each integral is equivalent to a line integral along one half of a dual mesh edge, identified by the index (m), and evaluates the cell's contribution to the discrete circulating current value associated with that edge. The total current may be summed from contributions from each contributing cell. In the interior of a block there will be two such contributions, but only one at the block surfaces from each block.

3.4.3. Maxwell's inhomogeneous equation

To discretize Maxwell's inhomogeneous equation,

$$\mathfrak{d}^{ik} H_k = -\mathrm{i}\omega \mathfrak{D}^i + \mathfrak{J}^i, \tag{38}$$

we must again obtain a discrete form of the differential operator $\mathfrak{d}^{ik}$, however, since the field is now a dual field, this will not be the same discrete operator as found earlier.

We begin the discretization by projecting (38) in a cell onto a dual vector test function

$$e^{ijk} \int w_i^{(m)} \frac{\partial H_k}{\partial \bar{x}^j} \,\mathrm{d}^3\bar{x} = -\mathrm{i}\omega \int w_i^{(m)} \mathfrak{D}^i \,\mathrm{d}^3\bar{x} + \int w_i^{(m)} \mathfrak{J}^i \,\mathrm{d}^3\bar{x}, \tag{39}$$

where the integral may be considered as a sum of integrals taken over each cell. Integrating the left hand side by parts, we obtain

$$e^{ijk}\int_{\Gamma} w_i^{(m)}\hat{n}_j H_k \,\mathrm{d}^2\bar{x} + \int (\mathfrak{d}^{ki} w_i^{(m)}) H_k \,\mathrm{d}^3\bar{x} = -\mathrm{i}\omega \int w_i^{(m)} \mathfrak{D}^i \,\mathrm{d}^3\bar{x} + \int w_i^{(m)} \mathfrak{J}^i \,\mathrm{d}^3\bar{x}. \tag{40}$$

Assuming for the present that we have either PEC or PMC conditions specified on the boundary, the surface term evaluates to zero.

Once again, it is possible to represent the discrete form of the differential operator without further approximation, since in a distributional sense the *curl* of a dual vector test function may be represented exactly as a linear combination of dual pseudo-vector test functions, in a manner that is compatible across cell interfaces. We therefore again write the matrix form, now $\bar{d}^{(mn)}$, of the differential operator $\mathfrak{d}^{ij}$ interior to a block,

$$\mathfrak{d}^{ki} w_i^{(m)} = \bar{d}^{(mn)} \hat{w}^k_{(n)}. \tag{41}$$

Substituting (41) into (40) and substituting the projected components (35) and (37) we obtain the discrete form of Maxwell's inhomogeneous equation,

$$\bar{d}^{(mn)} I_{(n)} = -\mathrm{i}\omega Q^{(m)} + I^{(m)}, \tag{42}$$

where we have projected the source current,

$$I^{(m)} = \int_{\mathrm{cell}} w_i^{(m)} \mathfrak{J}^i \,\mathrm{d}^3\bar{x}. \tag{43}$$

Once again, $\bar{d}^{(mn)}$ corresponds to the Yee scheme, now on the dual mesh, and the operator can again be implemented over a block of the structured grid as an optimized function.

From the known properties of the discrete difference operators of the Yee scheme, the matrix $\bar{d}^{(mn)}$ is in fact simply the transpose of the matrix $d^{(mn)}$, i.e., $\bar{d}^{(mn)} = d^{(nm)}$. This is an important symmetry property that a discretization method must possess to ensure that the computed eigenfrequencies of a closed, lossless cavity will be real.

3.5. Material operators

The discrete constitutive relations for the block are now obtained by substituting the above basis and test fields into the continuum constitutive relations and summing all contributions to each component. Here the basis functions are enumerated by the degree of freedom with which they are associated, and are assumed to be defined piecewise by the corresponding cell basis functions in each adjacent cell.

3.5.1. Capacitance operator

To obtain the capacitance operator, we can make the following sequence of transformations:

$$Q^{(p)} = \int_{\Omega} w_i^{(p)} \mathfrak{D}^i \,\mathrm{d}^3\bar{x} = \int_{\Omega} w_i^{(p)} \mathfrak{C}^{ij} E_j \,\mathrm{d}^3\bar{x} = V_{(q)} \int_{\Omega} w_i^{(p)} \mathfrak{C}^{ij} v_j^{(q)} \,\mathrm{d}^3\bar{x} = C^{(pq)} V_{(q)}, \tag{44}$$

where

$$C^{(pq)} = \int_{\Omega} w_i^{(p)} \mathfrak{C}^{ij} v_j^{(q)} \,\mathrm{d}^3\bar{x}. \tag{45}$$

This is a generalization of the familiar circuit relation, and defines the generalized capacitance between grid edges that share a common adjacent cell. A similar derivation applies to obtain components of the conductance operator.

3.5.2. Inductance operator

We discretize the inverse inductance operator similarly, but now using the pseudo-vector basis and dual pseudo-vector test functions,

$$I_{(m)} = \int_{\Omega} \hat{w}^i_{(m)} H_i \,\mathrm{d}^3\bar{x} = \int_{\Omega} \hat{w}^i_{(m)} \mathfrak{L}^{-1}_{ij} \mathfrak{B}^j \,\mathrm{d}^3\bar{x} = \Phi^{(n)} \int_{\Omega} \hat{w}^i_{(m)} \mathfrak{L}^{-1}_{ij} \hat{v}^j_{(m)} \,\mathrm{d}^3\bar{x} = L^{-1}_{(mn)} \Phi^{(n)}, \tag{46}$$

where

$$L^{-1}_{(mn)} = \int_{\Omega} \hat{w}^{i}_{(m)} \mathfrak{L}^{-1}_{ij} \hat{v}^{j}_{(n)} \, \mathrm{d}^3\bar{x}. \tag{47}$$

Again, this generalizes the familiar circuit relation, and defines the generalized reciprocal inductance between grid faces that share an adjacent cell.

3.6. Numerical integration

Each matrix element of the discrete material operators can be obtained by evaluating the integrals on each cell and summing the contributions. In order to reduce the memory requirements, we use a one-point Gauss–Legendre integration scheme in each cell, using the predetermined basis functions as weighting functions in the derivation of each scheme. Our treatment allows the capacitance and inductance operators to vary inside the cell, for example due to curved mesh elements at the boundary. The values are sampled at specific integration points inside the cell, appropriate to the integral being performed.

Different schemes are used for the diagonal matrix terms and the off-diagonal terms. Typically many cells contribute to the diagonal (self-coupling) terms, so that performing accurate integrations and storing the accumulated term is worthwhile. For off-diagonal terms, that represent coupling between near-neighbors, it is computationally advantageous to store the value of the tensor density at each cell center and apply a set of constant weights to compute the off-diagonal terms as they are needed. This may be done at the expense of some additional approximation in the numerical integrations.

Each integral may be performed as a sum of local cell integrations, for the cells that contribute, and each cell integration may be factorized into a product of three separate integrals, one for each orthogonal direction.

The contributions to the diagonal matrix terms are computed using a higher-order integration scheme and the contributions from each cell are accumulated and stored. For the diagonal terms, each cell integration requires evaluation of the following type of integral, where the weight and integration point are derived to perform exact integration up to and including linear variation of the material parameter, denoted $u(\hat{x})$,

$$\int_{-1/2}^{1/2} H(\hat{x}) u(\hat{x}) W(\hat{x}) \, \mathrm{d}\hat{x} \approx \alpha u(\hat{x}_0). \tag{48}$$

Assuming first a constant function for $u(\hat{x})$, then a linearly varying function, and solving the resulting equations for the values of α and $\hat{x}_0$, we obtain the following coefficients:

$$u(\hat{x}) = 1 \Rightarrow \alpha = \frac{3}{8} \tag{49}$$

$$u(\hat{x}) = \hat{x} \Rightarrow \hat{x}_0 = \frac{5}{18}. \tag{50}$$

In this scheme, therefore, the integration point lies closer to the edge or face with which the degree of freedom is associated, reflecting the weighting induced by the basis function. Fig. 3 shows the location of this integration point used for the diagonal terms.

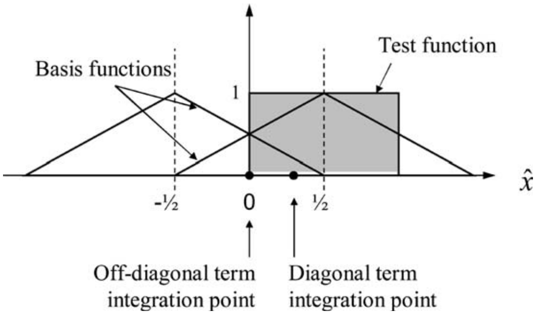

Fig. 3. Overlap of test and basis functions in cell-centered coordinates, from which weight functions are computed. The integration points are shown for both off-diagonal and diagonal term integration schemes.

The integrals for the off-diagonal terms (in cell-centered coordinates) are approximated using a single-point numerical integration scheme in each case, where the function $u(\hat{x})$ represents the continuum material parameter in the integrand:

$$\int_{-1/2}^{1/2} \delta[\hat{x}]u(\hat{x})\,\mathrm{d}\hat{x} = u(0), \tag{51}$$

$$\int_{-1/2}^{1/2} H(\hat{x})u(\hat{x})\,\mathrm{d}\hat{x} \approx \frac{1}{2}u(0), \tag{52}$$

$$\int_{-1/2}^{1/2} \delta[\hat{x}]u(\hat{x})W(\pm\hat{x})\,\mathrm{d}\hat{x} = \frac{1}{2}u(0), \tag{53}$$

$$\int_{-1/2}^{1/2} H(\hat{x})u(\hat{x})W(\hat{x})\,\mathrm{d}\hat{x} \approx \frac{3}{8}u(0), \tag{54}$$

$$\int_{-1/2}^{1/2} H(\hat{x})u(\hat{x})W(-\hat{x})\,\mathrm{d}\hat{x} \approx \frac{1}{8}u(0). \tag{55}$$

In each case, we assume that the function $u(\hat{x})$ is smooth and take the integration point to be at the mid-point, so that only a single evaluation of the material tensor density is stored for each cell. Such a scheme enforces the symmetry of the operator. Fig. 3 depicts the overlap of test and basis functions for the last two integrals, and shows the location of the integration point for the off-diagonal terms.

The result of this scheme is an efficient implementation in terms of both speed and memory overhead. We will show in Section 5.1 that we also achieve significantly reduced discretization error when compared to the standard linear finite element method on an identical mesh. This corresponds to lower numerical dispersion of the discrete operators.

3.7. Matrix storage

A significant limitation for practical computation is the memory required for storage of the matrix elements. It is therefore useful to compare the number of such element values that must be stored with the finite integration and finite element methods. We will compare the storage for the capacitance matrix using each method, since the calculation is less involved but still representative of the total storage needed.

To estimate the storage for a typical mesh of N cells, we first estimate the number of edges and faces in the mesh. If we consider a general region of a structured hexahedral mesh, then every cell has 12 edges, each shared on average by 4 cells, so that there are approximately $3N$ edges. Similarly, every cell has 6 faces, each shared by 2 cells, so that there are approximately $3N$ faces also. These values may be used to compute expressions that estimate the necessary matrix storage.

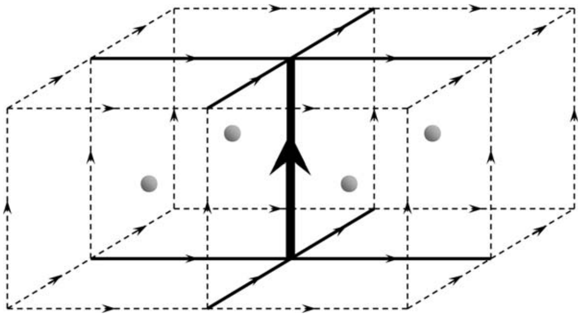

Fig. 4. Coupling between field components in a structured mesh: the finite element method couples the central edge to each other edge (solid and broken lines). The finite integration method couples only to the immediately adjacent edges (solid lines). The present method couples to each other edge, but uses a cell-centered representation (spheres) to reduce storage for the off-diagonal terms.

Fig. 4 shows four cells of the primary mesh adjacent to an edge, represented in the reference coordinate system. Each edge represents an electric field component (voltage), and therefore each row of the matrix represents the coupling terms between edges due to the mutual capacitance. For all three methods, the diagonal matrix elements represent coupling of each edge to itself, and require storage of one value for each edge, so there are $3N$ values to store for a mesh of N cells. To compute the number of off-diagonal terms needed to represent a region of a general non-orthogonal mesh we will treat each method separately.

For the finite element method using linear basis functions applied to a general region of a curvilinear mesh, the central edge may couple to each of the other 32 edges in the block of 4 cells shown in Fig. 4. There will therefore be 32 non-zero off-diagonal matrix elements per row of the matrix. With matrix symmetry storage is needed for 16 values per row, for a total of $51N$ values for a mesh of N cells including the diagonal terms.

The finite integration method for a non-orthogonal mesh reduces this number by coupling only to the 8 edges adjacent to the central edge, each with a unique coupling coefficient [23]. These are shown in Fig. 4 as solid edges radiating from the central edge. Matrix symmetry reduces the number to 4 independent off-diagonal elements per row, or a total of $15N$ values for a mesh of N cells. This represents a substantial saving over the finite element method, and reduces further if the mesh is in fact orthogonal, as all of the off-diagonal terms vanish.

In the present method, rather than store matrix elements explicitly, we take advantage of the fact that we have only one integration point per cell for the off-diagonal terms, at the cell center. We need therefore only store the symmetric 3×3 capacitance tensor density at this point, from which we can derive the off-diagonal matrix elements during application of the capacitance operator. We therefore need to store only 6 values per mesh cell, or $6N$ values to represent the off-diagonal elements, giving a total of $9N$ stored values to represent the capacitance matrix. In the present implementation we do not take advantage of the symmetry, but even so we store on average only $12N$ values. Our method therefore provides a reduction in the storage requirements for the matrix representation over both the finite element and finite integration methods.

We will now describe some details of our present implementation before demonstrating the numerical properties of our discretization scheme with some examples.

4. Implementation

CTLSS-2 implements the discrete representations of Maxwell's equations both as a system of linear equations driven by a source term, either a current source or external port, and as an eigenvalue system of equations. The implementation details are described in this section.

4.1. Second-order equations

We have derived discrete forms both for Maxwell's equations, in their metric-free form:

$$d^{(mn)}V_{(n)} = \mathrm{i}\omega\Phi^{(m)}, \tag{56}$$

$$\bar{d}^{(mn)}I_{(n)} = -\mathrm{i}\omega Q^{(m)} + I^{(m)}, \tag{57}$$

where $I^{(m)} = I^{(m)}_{\Omega} + I^{(m)}_{\text{ext}}$, and for the constitutive relations:

$$Q^{(m)} = C^{(mn)}V_{(n)}, \tag{58}$$

$$I_{(m)} = L^{-1}_{(mn)}\Phi^{(n)}, \tag{59}$$

$$I^{(m)}_{\Omega} = G^{(mn)}V_{(n)}. \tag{60}$$

We can write these equations as a single, second order discrete differential equation

$$\left[\bar{d}^{(mp)}L^{-1}_{(pq)}d^{(qn)} - \mathrm{i}\omega G^{(mn)} - \omega^2 C^{(mn)}\right]V_{(n)} = \mathrm{i}\omega I^{(m)}_{\text{ext}}, \tag{61}$$

where the right hand side is an imposed current source having angular frequency ω.

When there is no external current source we obtain a quadratic matrix eigenvalue equation,

$$\left[\bar{d}^{(mp)}L^{-1}_{(pq)}d^{(qn)} - \mathrm{i}\omega G^{(mn)} - \omega^2 C^{(mn)}\right]V_{(n)} = 0. \tag{62}$$

This can be simplified to an approximate linear eigenvalue equation,

$$[K^{(mn)} - \omega^2 M^{(mn)}]V_{(n)} = 0, \tag{63}$$

where we define:

$$K^{(mn)} \equiv \bar{d}^{(mp)} L^{-1}_{(pq)} d^{(qn)}, \tag{64}$$

$$M^{(mn)} \equiv C^{(mn)} + \frac{\mathrm{i}G^{(mn)}}{\omega}. \tag{65}$$

This is the form of the equation solved by CTLSS-2, where the eigenvalue ω^2 gives the frequency $f = \omega/2\pi$ of a resonant mode, and the eigenvector $V_{(n)}$ gives the basis coefficients for the electric field of that mode.

The matrices $K^{(mn)}$ and $M^{(mn)}$ have properties similar to those exhibited by the equivalent matrices found in the finite element or finite integration methods. The symmetry of $K^{(mn)}$ derives from the symmetry between the discrete *curl* operators, each being the transpose of the other, and the symmetry of the inverse inductance matrix, enforced during the integration procedure. Similarly, the symmetry of $M^{(mn)}$ derives from the integration procedure. As in the FIT method, and the FEM method using edge-elements, the matrix $K^{(mn)}$ has a large nullspace corresponding to the nullspace of the discrete *curl* operator, $d^{(qn)}$ above. This nullspace contains discrete fields that derive from the discrete gradient of a scalar function, and leads to many zero eigenvalue solutions to the equations. The system is therefore only positive semi-definite, the strictly positive solutions typically being the ones of interest. It will be necessary, therefore, when solving the system of equations to filter out these zero frequency solutions.

4.2. Multi-block decomposition of fields and operators

The electromagnetic model implemented in CTLSS-2 performs the following operation on a vector field, for use in iterative solution schemes,

$$y^{(m)} = [K^{(mn)} - \omega^2 M^{(mn)}]x_{(n)}, \tag{66}$$

where $x_{(n)}$ is the vector field, $y^{(m)}$ is a dual vector field defined on the solution domain, Ω. We can take the operators here to apply to the entire multi-block mesh, assuming appropriate care is taken at the block interfaces. Using the multi-block subdivision scheme, the solution domain is represented formally as a union of blocks, each a subdomain $\Omega_{[k]}$, as

$$\Omega = \bigcup_k \Omega_{[k]}. \tag{67}$$

Accordingly, the tensor density operators may be subdivided into block operators, and the dual fields into individual block fields as follows:

$$y^{(m)} = \sum_k y^{(m)}_{[k]} \tag{68}$$

$$K^{(mn)} = \sum_k K^{(mn)}_{[k]}, \text{ etc.,} \tag{69}$$

where the domain of integration in the defining functionals for each term is restricted to the corresponding block subdomain. Therefore the block operators may be written independently,

$$y^{(m)}_{[k]} = \left[K^{(mn)}_{[k]} - \omega^2 M^{(mn)}_{[k]}\right]x_{(n)}. \tag{70}$$

This leads to a *distributed* representation of operators and dual fields: since the components are integral quantities, the fields at the block interfaces are represented partially in each of the adjoining blocks. The vector fields, on the other hand, may be conveniently stored as a set of blocks fields, but these must be consistent across block interfaces so that they represent parts of the same global field $x_{(n)}$. This distinction is the motivation for the careful classification of fields according to their geometric properties.

In the context of a multi-block mesh, fields must satisfy the same conditions of continuity across block boundaries as was required between adjacent cells within each block. Care must be taken when field coefficients are compared between adjacent blocks since the orientation of blocks with respect to one another at the interface will require an appropriate transformation of the orientations of the edges, and therefore the signs of the associated field components. Therefore, a valid multi-block field requires that all coincident components in adjacent blocks are transformed compatibly.

Dual fields are somewhat different, in that they are represented in a distributed fashion; specifically, the surface components of each dual field block represent a contribution to the dual field associated with the volume of the dual mesh cell in that block. Unlike the components interior to the blocks, the contributions from each adjacent cell are not accumulated across block boundaries. The advantage of this scheme is that it is possible to compute the block operators *independently*. Provided the operators are applied to a valid multi-block vector field, i.e., compatible at the interfaces, the resulting dual vector field is correctly computed in its distributed form, with no further communication between blocks.

This scheme is self-consistent, in that global scalar products between vector fields and dual vector fields may be performed by first computing each block's contribution independently, then computing the global inner product by summation. Provided the vector field satisfies the compatibility conditions, this will produce the correct result independent of the presence of block interfaces.

4.3. Inter-block communication

Clearly, some communication between blocks is necessary during the solution procedure, but it is performed as an independent operation. It turns out that this separation can be included naturally within iterative solution algorithms.

The second order matrix operators in (66) are applied to a vector field to produce a (distributed) dual vector field. In an iterative scheme, the result is combined, according to the particular method, to update the vector field for the next iteration. Since the field and dual field lie in different (dual) vector spaces, it is necessary to introduce a preconditioning operator to transform from the dual vector space back to the vector space before any such operations can be performed. Therefore, for the eigenvalue case, we apply a preconditioned iterative scheme to the equivalent equation,

$$P^{-1}_{(qm)}\left[K^{(mn)} - \omega^2 M^{(mn)}\right]V_{(n)} = 0, \tag{71}$$

where $P^{-1}_{(qm)}$ is usually referred to as a *preconditioning* matrix, though we never construct the matrix explicitly. Preconditioned variants of many iterative algorithms are well known, and these correctly distinguish between vectors and dual vectors when performing mathematical operations.

The purpose of preconditioning is usually simply to accelerate the convergence of the iterative solver, however, here we rely on the preconditioning operation for two other important purposes. The first is to gather the distributed dual field components distributed across blocks and to form a vector field that is in some sense equivalent, enforcing the constraint that the resulting block representations of the vector field are compatible across block interfaces. We choose a weighting that uses the inverse of the diagonal part of the capacitance operator to perform the transformation from the accumulated dual vector field to the vector field, corresponding to a standard Jacobi preconditioner. The second purpose is to enforce additional constraints on the electric field solution due to the boundary conditions, and this will be described in the following section.

This scheme would lend itself naturally to parallel processing implementations. Individual block operators could be computed in parallel, with a single communication step during the preconditioning phase. In principle the communication may be performed in parallel with the computation of the block operators, by first computing the block surface field components to send to adjacent blocks, then performing the interior field computations in parallel with communication, before finally receiving the surface terms from adjacent blocks necessary to apply the preconditioner.

4.4. Boundary conditions

CTLSS-2 includes a number of boundary conditions useful to electromagnetic simulations. We outline here the implementation for each type of boundary condition.

The boundary of the solution domain may be subdivided into regions according to the type of boundary condition that is to be applied, and each region considered independently. Therefore, in the descriptions below, we refer only to that part of the surface to which that boundary condition applies. The mesh model in CTLSS-2 supports control of the boundary condition on each cell face, though frequently each block face has the same condition applied uniformly.

4.4.1. Perfect conductors

The case of a PEC boundary is an *essential* boundary condition that is imposed directly as a constraint on the field degrees of freedom that lie on the boundary. The contributions to the tangential components of the electric field on the surface are located on the mesh edges that lie on the surface. These are forced to zero during the preconditioning phase. The PMC, or *symmetry*, boundary condition is the *natural* boundary condition in this formulation and no further action is therefore necessary on this portion of the boundary.

4.4.2. Finite conductivity metal

For high-Q cavities, the dominant losses are typically due to Ohmic currents in cavity walls that have finite conductivity. Due to the inherent frequency-dependence of the skin depth in this case, this effect is best computed using perturbation theory [24], after the cavity eigensolution is complete and the resonant frequency is known. CTLSS-2 computes the cavity-Q and frequency shift due to this effect.

4.4.3. Floquet condition

The Floquet boundary condition models the phase advance of a wave traveling in a periodic system. The simplest form of the condition is

$$E_i(\mathbf{r}_0 + \delta\mathbf{r}) = E_i(\mathbf{r}_0)\mathrm{e}^{\mathrm{i}\beta\cdot\delta\mathbf{r}}, \tag{72}$$

where β is the wavevector of propagation and $\delta\mathbf{r}$ defines the translation necessary for periodicity. This form may be applied when the two faces are congruent under the translation $\delta\mathbf{r}$.

A more general form of the Floquet boundary condition is supported in CTLSS-2, and requires only that the two faces be congruent under any rigid-body transformation. That is, the second face maps exactly to the first face under a combination of rigid translation, rotation, and reflection. To implement this condition, electric field degrees of freedom at one face of the solution domain are set equal to those at the opposite face, scaled by the complex exponential phase factor, and taking account of the relative orientation of the block faces when matching field degrees of freedom. Again this is an essential boundary condition, applied during the preconditioning phase.

4.4.4. Waveguide ports

To model waveguide ports, and also radiation boundaries, it is necessary to absorb all propagating radiation that reaches the boundary. For this purpose, we use a perfectly matched layer (PML) [25,26] method. This method introduces a continuation of the solution region beyond the regular boundary, with special absorbing properties. In our formulation, we replace the real coordinate transformation $\mathbf{r}(\bar{x}^i)$ in this region with a complex coordinate transformation, $\mathbf{r} = \mathbf{r}'(\bar{x}^i) + \mathbf{r}''(\bar{x}^i)$, where the imaginary part is a monotonically increasing function of the distance into the absorbing region. The metric derived from this transformation corresponds to that obtained using the PML method, and is valid for a general curvilinear mesh. The particular parameters for this function are determined based upon the wave frequency. The PML is essentially a volumetric region of matched impedance, and is therefore not strictly a boundary condition.

To excite an incoming wave in a port, we introduce an appropriate source current on the original boundary in front of the extended PML region. The source current profile is derived from a 2-D eigenmode solution of the Helmholtz equation for each port cross-section and can therefore selectively excite specific waveguide eigenmodes. The amplitudes and phases of a number of port modes can be varied to create different drive scenarios.

In a simulation containing one or more driven ports, it is usually of interest to compute the scattering matrix components, or S-parameters. They may be represented as a matrix of complex amplitude ratios, relating the amplitude of each outgoing mode to that of each input mode. CTLSS-2 computes these ratios for selected modes, and converts to power ratios and phase information.

5. Verification and validation

This section describes tests performed with CTLSS-2 that compare its performance against analytical results, other codes and experimental data. The first two examples compare with analytic solutions in order to demonstrate the high accuracy achieved using our discretization method, even on relatively coarse meshes, when compared with other methods. The remaining examples demonstrate the capabilities of the code for modeling a range of geometries appropriate to vacuum electronic device design.

5.1. Spherical cavity

CTLSS-2 is compared here with analytical results [27] for the mode frequencies of a spherical cavity with perfect metallic walls, having a nominal radius $a = 1$ m. This test problem is used to compare the scaling of the relative frequency error with average cell size and run time with finite element and finite integration methods, for a few low order modes of the cavity.

A typical grid for the cavity is shown in Fig. 5. It consists of seven blocks, as shown, and has 4-fold symmetry in the azimuthal angle as shown in the center frame. A simple orthogonal discretization based on a cylindrical coordinate system would lead to a coordinate singularity at the origin, whereas here the center of the sphere is handled using an O-grid topology, in which the block containing the center of the sphere is treated with a Cartesian grid, while the outer blocks map to the spherical surface.

To compare the effectiveness of the new algorithm, this problem was solved using three methods: the present method, an implementation of the linear finite element method, and the commercial electromagnetics code Microwave Studio (from CST). The commercial code provides an absolute measure of the performance of our code with respect to a state-of-the-art commercial electromagnetics code that uses the original FIT algorithm. Microwave Studio, in addition, has the capability to handle curved boundaries conformally, making this problem well suited even on the orthogonal grid.

For the first two methods we used the same meshes to perform the discretization, and employed similar iterative solvers. Both methods used a higher-order representation of the spherical surface so that the geometry was well resolved. Four different grids with varying grid density were employed in this study, summarized in Table 3. The average cell size is computed as $\Delta x = (N_{\text{cells}})^{-1/3}$, where Δx is the average cell size (in arbitrary units) and N_{cells} is the number of cells. For Microwave Studio, we used simple uniform orthogonal Cartesian meshes of $20 \times 20 \times 20$ and $40 \times 40 \times 40$ cells, using the conformal boundary approximation for our comparison. We solved for only the TM_{11} mode for each mesh in this case.

To demonstrate the convergence scaling with respect to the mesh density for the present method and the finite element method, the frequency errors for each solved mode were studied as a function of the average cell size using the grids described earlier. The relative frequency error is defined as

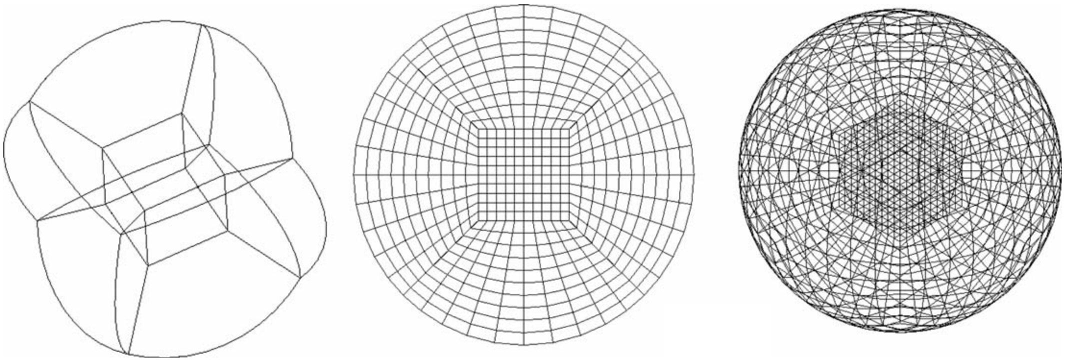

Fig. 5. Multi-block O-grid for spherical cavity. The left-hand frame shows the seven-block grid structure. The block containing the center of the sphere uses a Cartesian grid, while curved grid segments are used outside this central block. The center frame shows a cut plane through the center of the sphere, aligned with the Cartesian grid in the center block, while the frame on the right shows an isometric view of the grid edges.

Table 3
Grid sizes used for discretization of sphere

Grid	7×6^3	7×10^3	7×15^3	7×20^3
Number of cells	1512	7000	23,625	56,000
Average cell size	0.0871	0.0522	0.0348	0.0261

$$\Delta f = \frac{f_{\text{numerical}} - f_{\text{analytical}}}{f_{\text{analytical}}}, \tag{73}$$

where $f_{\text{numerical}}$ is the converged mode frequency computed by each method and $f_{\text{analytical}}$ is the analytical mode frequency.

The normal mode frequencies for this cavity are given by solutions to the equations:

$$j_l(k_l a) = 0 \quad \text{(TE modes)}, \tag{74}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[x j_l(k_l x)]_{x=a} = 0 \quad \text{(TM modes)}, \tag{75}$$

$$k_l = \frac{\omega}{c}, \tag{76}$$

where $j_l(x)$ is the spherical Bessel function of order l. The modes are labeled by $\{l,n,m\}$ where l is the polar angle mode number, n is the radial mode number (i.e., the nth zero of the appropriate equation above for TE or TM modes, respectively), and m is the azimuthal angle mode number. There are $2l+1$ degenerate modes, corresponding to $-l \leqslant m \leqslant l$, for each $l = 1, 2, \ldots$ In the following, the m number has been suppressed from the mode notation, and modes are labeled as TE_{ln} or TM_{ln}.

We compare the numerical eigensolutions using each method for the TM_{11}, TM_{21}, TE_{11} and TE_{21} modes. These modes were chosen as they are well resolved and converge uniformly with the grid densities used. Fig. 6 shows the frequency error plotted against average cell size, while the slope of the triangle indicates the relationship $\Delta f \propto (\Delta x)^2$. Both the present method (open markers) and the finite-element method (solid markers) can be seen to follow this relationship (i.e., second-order convergence) as Δx becomes small, however, the present method clearly has a significantly lower absolute error in each case for the same mesh density. This

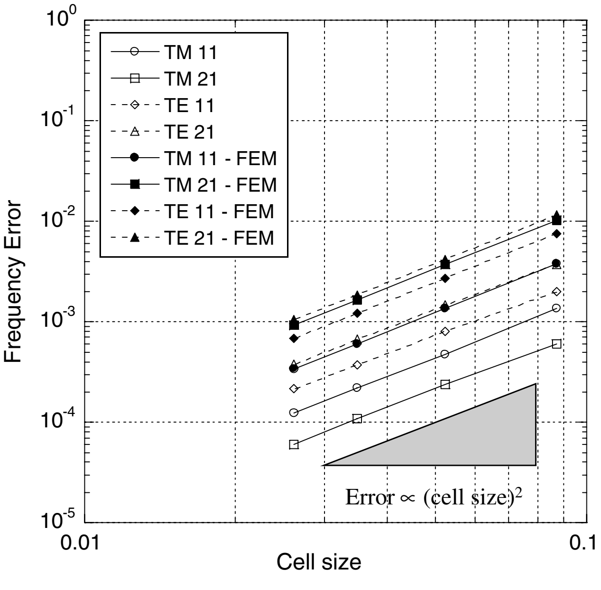

Fig. 6. Frequency error plotted against average cell size for the low order modes of a spherical cavity. This plot demonstrates second-order scaling, in both the present method (open markers) and the finite element method (closed markers), but with significantly lower errors achieved using the new algorithm.

indicates a significantly improved numerical dispersion of our operators for any given mesh, when compared to the finite element method, that is independent of the field structure in the cavity.

An explanation for this fact may lie in the hybrid nature of our method. Typically for the finite element method, the solution eigenfrequencies converge to the true solution from above, while the finite integration method typically converges from below. It appears that by introducing an intermediate level of coupling between adjacent field components our method achieves some degree of cancellation of these positive and negative error terms. The precise mechanism will require further investigation.

The scaling of solution time with frequency error is shown in Fig. 7, with times adjusted for a 2.1 GHz PC. For the first two methods above, the run time scales as the product of the number of cells, N_{cells}, and the square root of the spectral condition number, κ, for the matrix that represents the operator to be inverted or diagonalized. Since $N_{\text{cells}} \propto (\Delta x)^{-3}$ and $\kappa \propto (\Delta x)^{-2}$ assuming a uniform Cartesian grid, the running time, τ, will scale as,

$$\tau \propto N_{\text{cells}}\kappa^{1/2} \propto (\Delta x)^{-4}. \tag{77}$$

For a second-order solver, the frequency error scales as $(\Delta x)^2$, and hence the frequency error would be expected to scale as $\tau^{-1/2}$, as seen in Fig. 7.

Since the curves for the finite element method and our present method are almost collinear for most of the modes shown, it seems that in our present implementation the gains achieved in numerical dispersion on the mesh is offset by the increased run time necessary to compute the matrix elements from the stored data during application of the operators. However for the same solution time we clearly can use a much coarser mesh, with a corresponding additional saving in memory. Alternatively, by using the same explicit matrix storage scheme as used for the finite element method, it should be possible to significantly enhance speed at the expense of memory storage, while keeping the improved numerical characteristics.

Also shown are two points found using Microwave Studio to solve for the TE_{11} mode. These are close to the corresponding curves for the other two methods, and suggests that our present code is providing acceptable performance, at least for certain types of geometries.

5.2. Cylindrical cavity having a conducting wall

In this test, a cylindrical cavity whose wall has a finite conductivity typical of real metals is analyzed using the perturbative approach. The wall is assigned a skin depth, $\delta = 1$ μm, and the cavity a radius $r = 1$ cm. The frequency and cavity Q for the TE_{111} and TM_{010} modes are then compared with analytical expressions [24].

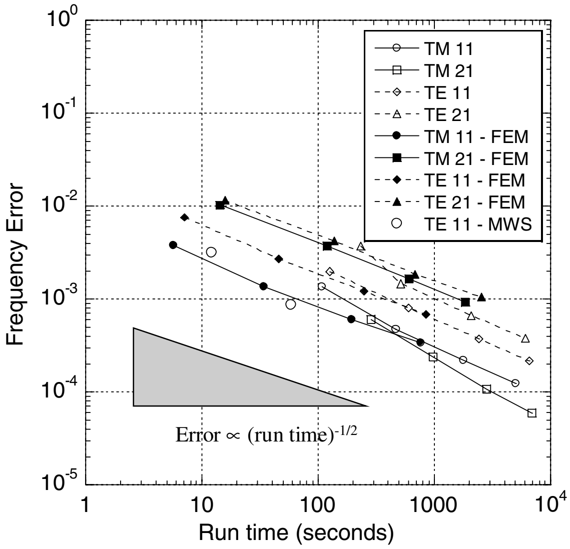

Fig. 7. Frequency error vs. run time for the three methods. Discrete points are shown for simulations using Microwave Studio (MWS). This plot shows the expected scaling for the present method, in which the frequency error scales as the inverse square root of the run time.

TE_{111} mode:

$$f = \frac{1.841c}{2\pi r}[1 + 2.912(r/h)^2]^{1/2}, \tag{78}$$

$$Q = \frac{1}{2}\left(\frac{h}{\delta}\right)\frac{1 + 0.344(h/r)^2}{1 + 0.209(h/r) + 0.242(h/r)^3}. \tag{79}$$

TM_{010} mode:

$$f = \frac{2.405c}{2\pi r}, \tag{80}$$

$$Q = \left(\frac{h}{\delta}\right)\frac{1}{1 + h/r}. \tag{81}$$

The results for the TE_{111} and TM_{010} modes are given in Tables 4 and 5, respectively. The same mesh topology was used for each simulation, consisting of 5 blocks, arranged as shown in Fig. 1. For the TE_{111} mode, each block was $10 \times 10 \times 20$ cells (10,000 cells in total), while for the TM_{010} mode, each block was $10 \times 10 \times 10$ cells (5000 cells in total). Run times on a 2.1 GHz PC were typically 30 s for the TM_{010} solutions and 90 s for the TE_{111} solutions, but reached over 300 s for the finer mesh with 0.5 cm cavity height, for which the cells have a very high aspect ratio and the iterative solution method is poorly conditioned.

The frequency error and Q error from CTLSS-2 relative to the analytical expressions are shown graphically in Figs. 8 and 9. For each mode, the frequency error $\Delta f/f$ is 0.1% or less, while the Q error, $\Delta Q/Q$, is less than 0.5% for all but a single case. For the TE_{111} mode, the perturbative calculation appears to diverge as the cavity becomes shorter and the mode frequency increases, reaching $\Delta Q/Q \approx 3\%$ for the 0.1 cm cavity.

5.3. Coupled-cavity TWT

CTLSS-2 has been tested on a coupled-cavity TWT slow-wave structure design that was developed at Communications and Power Industries. This calculation involved only perfectly conducting materials, and used only electric wall or magnetic wall boundary conditions at the ends of the unit cell to select the π- and 2π-modes. The fields and mode frequencies are therefore real. Using symmetries in the transverse plane,

Table 4
TE_{111} mode frequency and Ohmic Q

Cavity height (cm)	Analytical		CTLSS-2	
	Frequency (GHz)	Q	Frequency (GHz)	Q
0.1	150.15	491	150.46	495
0.5	31.240	2393	31.29	2381
1.0	17.374	4631	17.38	4631
2.0	11.548	7084	11.55	7081
3.0	10.106	7528	10.11	7538
5.0	9.2824	7432	9.288	7431

Table 5
TM_{010} mode frequency and Ohmic Q

Cavity height (cm)	Analytical		CTLSS-2	
	Frequency (GHz)	Q	Frequency (GHz)	Q
0.1	11.475	909.1	11.470	906.5
0.5	11.475	3333	11.475	3327
1.0	11.475	5000	11.476	4992
2.0	11.475	6666	11.476	6658
3.0	11.475	7500	11.476	7491
5.0	11.475	8333	11.476	8326

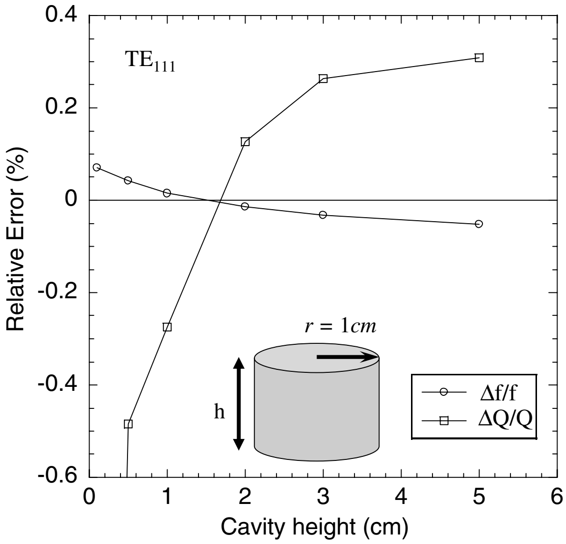

Fig. 8. Frequency and Q error in CTLSS-2 relative to analytical expressions for the TE_{111} mode in a cylindrical cavity with a conducting wall. The cavity geometry is shown.

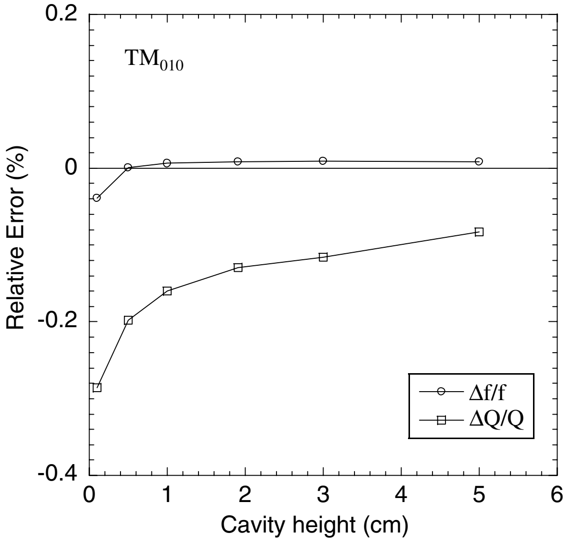

Fig. 9. Frequency and Q error in CTLSS-2 relative to analytical expressions for the TM_{010} mode in a cylindrical cavity with a conducting wall.

CTLSS-2 was run for just one quadrant of a unit cell. The unit cell is a single axial period of the structure, assuming a 90° rotation between consecutive periods. Placing electric wall boundary conditions at the end faces of the unit cell yields the 2π mode (having phase advance per cell of 2π). With magnetic wall boundary conditions imposed at the end faces, the solution is the π mode.

This problem was analyzed using a low-resolution grid with 3328 cells, and also using a high-resolution grid with 36,855 cells. Table 6 summarizes the results and timing for these simulations.

Using the high-resolution grid, the mode frequencies computed with CTLSS-2 agree with experimental data to within 0.27%. With the low-resolution grid, the frequency error relative to experimental data is within 0.60%, with a running time that is reduced almost 20-fold. The field structure for the 2π mode on the high-resolution grid is shown in Fig. 10.

Table 6
Comparison between simulation and experimental measurement for the coupled-cavity structure

Mode	Frequency (GHz)			Error	
	3328 Cells	36,855 Cells	Experiment	3328 Cells	36,855 Cells
π-mode	4.8519 (110 s)	4.8325 (2098 s)	4.8440	0.16%	−0.24%
2π-mode	6.6524 (88 s)	6.6747 (1632 s)	6.6925	−0.60%	−0.27%

Runtime is shown for each solution for a 2.1 GHz PC.

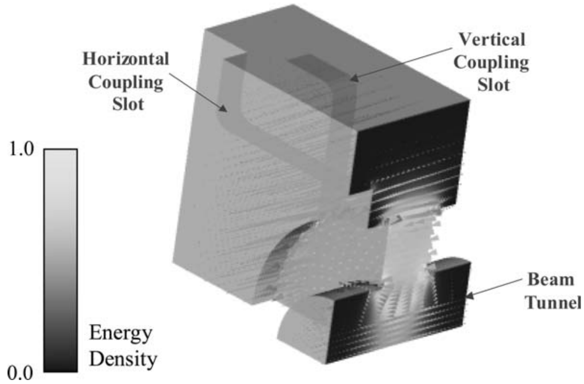

Fig. 10. CCTWT result with CTLSS-2. The gray-scale represents the electric field energy density on a cross-section, and electric field vectors are superimposed. This result corresponds to the 2π mode on a grid with 36,855 cells. One-quarter cell of the double-staggered ladder CCTWT geometry is visible in the plot.

5.4. Klystron cavity with loss buttons

To examine a vacuum electron device that incorporates lossy materials, CTLSS-2 has been tested on a cavity of a klystron that incorporates buttons of lossy material on the cavity walls to control the cavity Q. These buttons have a complex permittivity, $\varepsilon/\varepsilon_0 = 51.0 + 34.6\mathrm{i}$, where ε_0 is the permittivity of free space, and therefore a loss tangent, $\tan\delta = 0.68$.

This particular klystron cavity uses five loss buttons distributed evenly around the axis. CTLSS-2 therefore needs to model only 1/10th of this cavity, consisting of the pie-shaped slice that extends from the symmetry plane between two loss buttons to the mid-plane of one of them. The commercial grid generator ICEM-CFD was used to create a multi-block, hexahedral, structured mesh for this geometry, consisting of 20,736 cells. Fig. 11 shows this type of klystron cavity geometry, and the mesh geometry used in the simulations.

The presence of loss buttons having such large loss tangent and large real part of ε causes the operator matrix to become very poorly conditioned, leading to rather long execution due to slow convergence. The testing procedure therefore consisted of two steps. First, the cavity was modeled without loss buttons, using the same mesh. Then, the cavity was modeled again with the loss button structure included with its complex permittivity. The surfaces of the grid defining the wedge are defined to have magnetic wall symmetry boundary conditions, while the cavity walls and beam tunnel termination are treated as PEC boundaries.

Without loss buttons CTLSS-2 obtained 3.453 GHz for the lowest mode, while the measured mode frequency (without loss buttons) is (3.452 ± 0.001) GHz. The runtime for this simulation was approximately 19 min on a 2.1 GHz PC. The result with loss buttons gave a mode frequency of 3.4138 GHz and a mode Q of 51.75, compared with the experimental result, frequency of 3.414 GHz and Q of 97. The discrepancy in Q can be removed by increasing the imaginary part of ε 2-fold, which is within the uncertainty in the loss tangent of the loss button material at RF frequencies. For the same structure and material values, CTLSS-2 and the finite element code HFSS compute the same mode frequencies to within 0.1% and the same Q values to within 2%, with comparable run times in the lossless case.

The field structure in this mode, obtained by CTLSS-2, is shown in Fig. 12.

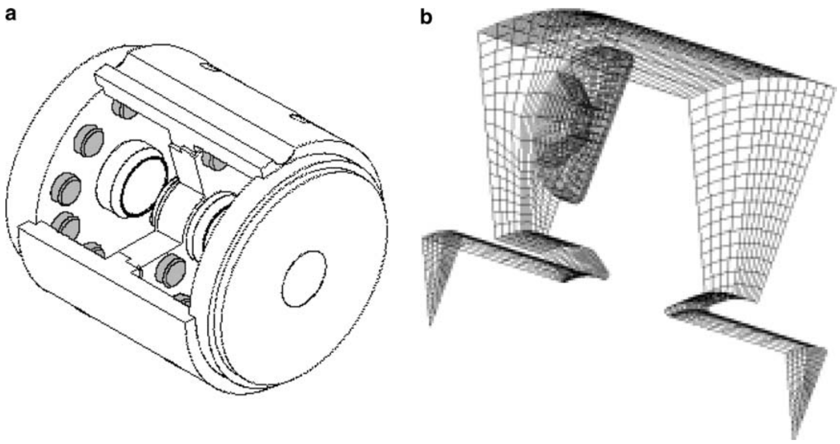

Fig. 11. (a) Cut-away view of a pair of button-loaded klystron cavities. (b) Detail of the surface mesh for the 20,736-cell multi-block model of a 1/10th segment of the five-button cavity. The button surface is represented conformally.

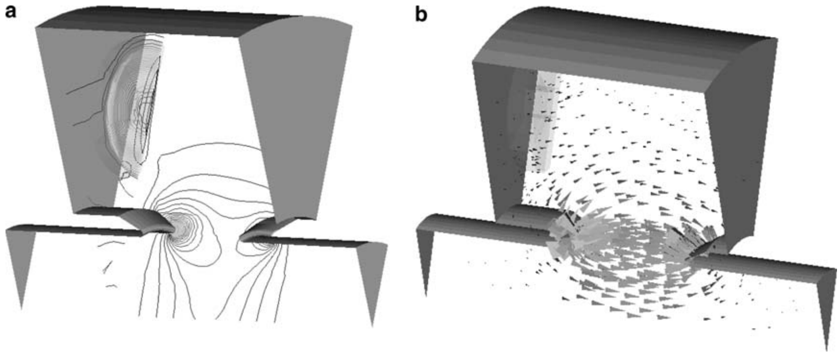

Fig. 12. Field profiles from a CTLSS-2 eigenmode solution: (a) contours of electric field energy density; (b) the electric field vectors.

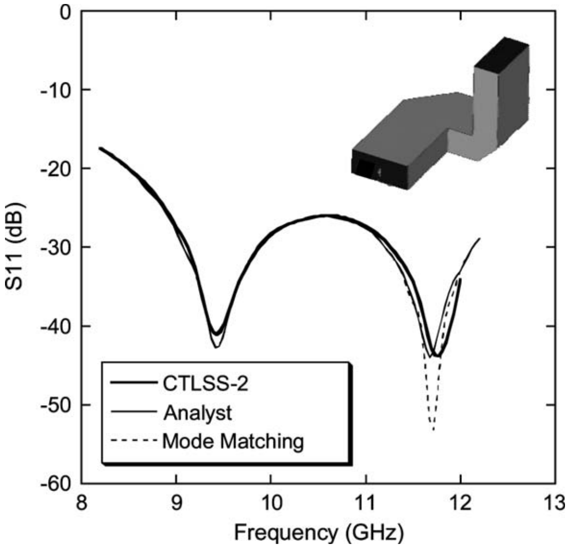

Fig. 13. Reflected power (S_{11}) from a double miter bend in a waveguide, computed using CTLSS-2, analyst and a mode-matching method [28]. The problem geometry is shown.

5.5. Waveguide with double miter bends

To test the port model, the final example consists of a waveguide with consecutive 90° H-plane and E-plane bends [28]. We computed the *S*-parameters for this problem over a range or frequencies using CTLSS-2, for a mesh of roughly 105,000 cells. Fig. 13 shows the structure and computed power reflection, compared to results from both the commercial finite-element electromagnetics code Analyst (Simulation Technology & Applied Research, Inc.), and the original mode-matching calculation [28].

Agreement is excellent in the non-resonant regions, while a small spread in the frequencies of resonances can be seen. This is most likely due to residual numerical dispersion caused by the mesh, to which the resonance frequencies will be particularly sensitive.

6. Conclusions

We have presented results of a new hybrid discretization method applied to the solution of Maxwell's equations. By using a tensor density formulation in general curvilinear coordinates, we obtained a metric-free form of the electromagnetic equations on a non-orthogonal, multi-block grid. By introducing new projection operators for the material matrix elements, derived using a combination of finite element basis functions with finite integrals on a dual mesh, we have demonstrated significantly improved eigenfrequency accuracy when compared to discretization using the standard linear finite element method using the same mesh. We were also able to reduce the storage necessary to represent the discrete matrix operators by implementing the discrete difference operators efficiently in functional form and using an efficient representation for the off-diagonal material matrix elements. The new model achieves second-order solution accuracy with respect to the average mesh cell size as expected for a conformal geometry representation.

We have shown using our present implementation that the method is effective for accurate simulation of large, realistic, electromagnetic problems with complex geometries and material distributions. Examples of cold-test simulations for typical vacuum electronic device components have been presented to validate the algorithm for both cavity eigenvalue and port-driven frequency domain electromagnetic problems.

Acknowledgments

This work was supported by the US Office of Naval Research. We thank Dr. Carol Kory for performing the Microwave Studio simulations.

References

[1] T. Weiland, A discretization method for the solution of Maxwell's equations for six-component fields, Electron. Commun. (AE U) 31 (1977) 116.

[2] T. Weiland, Time domain electromagnetic field computation with finite difference methods, Int. J. Numer. Model: Electron. Netw. Dev. Field 9 (1996) 295–319.

[3] G.L.G. Sleijpen, H.A. van der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, SIAM J. Matrix Anal. Appl. 17 (2) (1996) 401–425.

[4] S.J. Cooke, B. Levush, Eigenmode solution of 2-D and 3-D electromagnetic cavities containing absorbing materials using the Jacobi–Davidson algorithm, J. Comput. Phys. 157 (1) (2000) 350–370.

[5] R.W. Freund, N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339.

[6] W.C. Chew, J.-M. Jin, C.-C. Lu, E. Michielssen, J.M. Song, Fast solution methods in electromagnetics, IEEE Trans. Antenn. Propag. 45 (3) (1997) 533–543.

[7] R. Schuhmann, T. Weiland, A stable interpolation technique for FDTD on nonorthogonal grids, Int. J. Numer. Model 11 (6) (1998) 299–306.

[8] R. Schuhmann, T. Weiland, The non-orthogonal finite integration technique applied to 2D- and 3D-eigenvalue problems, IEEE Trans. Mag. 36 (4) (2000) 897–901.

[9] U. van Rienen, T. Weiland, Triangular discretization method for the evaluation of RF-fields in cylindrically symmetric cavities, IEEE Trans. Mag. 21 (6) (1985) 2317–2320.

[10] R. Schuhmann, P. Schmidt, T. Weiland, A new Whitney-based material operator for the finite integration technique on triangular grids, IEEE Trans. Mag. 38 (2) (2002) 409–412.

[11] M. Cinalli, F. Edelvik, R. Schuhmann, T. Weiland, Consistent material operators for tetrahedral grids based on geometric principles, Int. J. Numer. Model 17 (2004) 487–507.

[12] J. Jin, The Finite Element Method in Electromagnetics, Wiley-IEEE Press, New York, 2002.

[13] T.M. Antonsen Jr., A.A. Mondelli, B. Levush, J.P. Verboncoeur, C.K. Birdsall, Advances in modeling and simulation of vacuum electronic devices, Proc. IEEE 87 (1999) 804–839.

[14] S.J. Cooke, A.A. Mondelli, B. Levush, T.M. Antonsen Jr., D.P. Chernin, T.H. McClure, D.R. Whaley, M. Basten, CTLSS – An advanced electromagnetic simulation tool for designing high-power microwave sources, IEEE Trans. Plasma Sci. 28 (3) (2000) 841–866.

[15] J. Eastwood, W. Arter, N. Brealey, R. Hockney, Body-fitted electromagnetic PIC software for use on parallel computers, Comput. Phys. Commun. 87 (1995) 155–178.

[16] J.A. Schouten, Ricci-calculus; An Introduction to Tensor Analysis and Its Applications, Springer-Verlag, Berlin, 1954.

[17] M. Bartsch, M. Dehler, M. Dohlus, F. Ebeling, P. Hahne, R. Klatt, F. Krawczyk, M. Marx, Z. Min, T. Pröpper, D. Schmitt, P. Schütt, B. Steffen, B. Wagner, T. Weiland, S. Wipf, H. Wolter, Solution of Maxwell's equations, Comput. Phys. Commun. 73 (1992) 22–39.

[18] J.P. Verboncoeur, A.B. Langdon, N.T. Gladd, An object-oriented electromagnetic PIC code, Comput. Phys. Commun. 87 (1995) 199–211.

[19] G. Scharf, From Electrostatics to Optics, Texts and Monographs in Physics, Springer-Verlag, Berlin, 1994.

[20] J.C. Nédélec, Mixed finite elements in R^3, Numer. Math. 35 (3) (1980) 315–341.

[21] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antenn. Propag. 14 (1966) 302–307.

[22] A. Bossavit, I. Mayergoyz, Edge-elements for scattering problems, IEEE Trans. Mag. 25 (4) (1989) 2816–2821.

[23] R. Schuhmann, T. Weiland, Stability of the FDTD algorithm on nonorthogonal grids related to the spatial interpolation scheme, IEEE Trans. Mag. 34 (5) (1998) 2751–2754.

[24] J.D. Jackson, Classical Electrodynamics, Wiley, New York, 1962.

[25] J. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (1994) 185–200.

[26] J.-P. Berenger, Three-dimensional perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 127 (1996) 363–379.

[27] E.M. Nelson, High accuracy electromagnetic field solvers for cylindrical waveguides and axisymmetric structures using the finite element method, Ph.D. Thesis, SLAC-0431, Stanford Linear Accelerator Center, Stanford University, 1993.

[28] J.M. Reiter, F. Arndt, Rigorous analysis of arbitrarily shaped H- and E-plane discontinuities in rectangular waveguides by a full-wave boundary contour mode-matching method, IEEE Trans. Microwave Theory Tech. 43 (4) (1995) 796–801.