A General Procedure for Introducing Structured Nonorthogonal Discretization Grids Into High-Order Finite-Difference Time-Domain Methods

Roberto B. Armenta, Member, IEEE, and Costas D. Sarris, Senior Member, IEEE

Abstract—This paper describes a general procedure for introducing structured nonorthogonal discretization grids into any high-order finite-difference time-domain method originally formulated on a uniform rectangular grid. The proposed procedure employs a coordinate transformation to map a conformal grid—a grid where all cells conform to the material boundaries of the given problem—onto a uniform rectangular grid where any high-order finite-difference scheme can be easily applied.

Index Terms—Finite-difference time-domain (FDTD) methods, waveguide components.

I. INTRODUCTION

The most appealing trait of high-order finite-difference time-domain (FDTD) methods [1]–[5] is that they can deliver accurate results using discretization grids that are much coarser than those required by the traditional second-order scheme proposed by Yee [6]. However, with only a few exceptions [7], [8], most high-order FDTD methods have been formulated on a uniform rectangular (or Cartesian) grid. As a result, their applicability has been unnecessarily restricted to problems where material boundaries are rectangular. This paper demonstrates that there is a general methodology for introducing a broader class of grid topologies—structured nonorthogonal grids—into any high-order FDTD method originally formulated on a uniform rectangular grid. To obtain a procedure that is independent of any particular discretization scheme, a coordinate transformation is used to map an arbitrary nonorthogonal structured grid onto a rectangular grid with uniformly spaced cells where any high-order FDTD method can be applied. The use of coordinate transformations is a defining characteristic of structured grid generation, and the reader not familiar with how grid topologies are defined and classified is referred to [9]–[11]. The proposed approach, which exploits differential geometry in a manner akin to transformation optics [12], is flexible enough to handle any type of curved material boundary.

Structured grid generation was first applied in a rigorous manner to Yee’s second-order FDTD scheme in [13]. This approach was built on earlier proposals [14]–[16], and it has since been applied to a number of waveguide problems [17]–[19].

One of the key advantages of using coordinate transformations is that they allow us to introduce the geometrical features of curved material boundaries prior to the discretization of Maxwell’s equations. As a result, this approach can be applied to any existing FDTD formulation. To implement the proposed mapping, two different coordinate systems—denoted hereby by \( x^i \) and \( u^j \) for all \( i, j = 1,2,3 \)—must be used. The first set \( x^i \) is simply the Cartesian coordinate system (i.e., \( x^1 = x, x^2 = y, \) and \( x^3 = z \)). The second one, \( u^j \), is a completely arbitrary set of curvilinear coordinates. An invertible coordinate transformation

\[
x^i = x^i(u^j) \quad \text{for all } i, j = 1,2,3
\]  

is created so that the \( u^j \) coordinate surfaces—which are drawn by keeping each of the three \( u^j \) coordinates constant one at a time—follow all of the material boundaries of the given problem in \( x^i \) coordinates. In this way, a uniform discretization of Maxwell’s equations in the \( u^j \) coordinate system will automatically map onto a conformal discretization in the \( x^i \) coordinate system. The \( u^j \) coordinate system will be sometimes referred to as the computational coordinate system since, under this approach, Maxwell’s equations are discretized and solved there. Having provided a basic sketch of the proposed idea, let us begin by expressing Maxwell’s equations in terms of the computational coordinate system. These equations are subsequently discretized using standard high-order finite differences.

II. MAXWELL’S EQUATIONS EXPRESSED IN A GENERAL CURVILINEAR VECTOR BASIS

To express Maxwell’s equations in computational coordinates, it is first necessary to define a set of basis vectors and a suitable metric tensor. Thus, as a first step, let us define two sets of basis vectors directly from (1).

A. Definition of the Basis Vectors and the Metric Tensor

The first set of basis vectors is defined by \( g_j = \partial r/\partial u^j \), where \( r = x^1 l_1 + x^2 l_2 + x^3 l_3 \) and \( l_j \) (or \( \mathbf{v}^j \)) stands for the \( j \)-th Cartesian unit vector. The \( q_j \) vector is known as the \( j \)-th covariant basis vector and its Cartesian components are \( g_{ij} = \partial x_i/\partial u^j \). The second set, denoted by \( q^i \), is known as the contravariant set, and the Cartesian components of the \( i \)-th contravariant vector are \( g^i \cdot i_j = \partial u^i/\partial x^j \). The geometrical role of the covariant and contravariant basis vectors is illustrated in Fig. 1. The \( j \)-th covariant basis vector \( g_j \) is tangential to the \( u^j \) coordinate line—that is, the line traced by varying \( u^j \) while keeping \( u^i \) and \( u^k \) constant—while the \( j \)-th contravariant basis vector \( g^j \) is normal to the \( u^j \) coordinate surface. From the basis
vectors, a metric tensor can be defined. Such tensor can be defined in either a covariant or a contravariant form. In covariant form, the metric tensor components are

$$g_{ij} = g_i \cdot g_j = \sum_{q=1}^{3} \frac{\partial x^i}{\partial u^q} \frac{\partial x^j}{\partial u^q}$$ (2)

whereas, in contravariant form, the metric components are

$$g^{ij} = g^i \cdot g^j = \sum_{q=1}^{3} \frac{\partial u^i}{\partial x^q} \frac{\partial u^j}{\partial x^q}.$$ (3)

Observe that the metric tensor is symmetric ($g_{ij} = g_{ji}$ and $g^{ij} = g^{ji}$), so it only contains six independent components. To avoid any potential confusion, a summary of the tensorial notation conventions used throughout this manuscript is given in Appendix A. With the necessary basis vector and metric definitions in place, let us carry on and project Maxwell’s equations using the covariant and contravariant basis vectors. To facilitate the introduction of material properties, a mixed covariant–contravariant representation was chosen.

B. Maxwell’s Equations Projected Onto the Basis Vectors

For our purposes, Maxwell’s equations are given by

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} - \mathbf{VB}$$ (4)

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{VD}$$ (5)

$$\frac{\partial \mathbf{H}}{\partial t} + \mathbf{GH} = \mathbf{M}^{-1} \left( \frac{\partial \mathbf{B}}{\partial t} + \mathbf{PB} \right)$$ (6)

$$\frac{\partial \mathbf{E}}{\partial t} + \mathbf{GE} = \mathbf{E}^{-1} \left( \frac{\partial \mathbf{D}}{\partial t} + \mathbf{PD} \right)$$ (7)

where $\mathbf{V}$, $\mathbf{P}$, and $\mathbf{M}$ introduce electric losses, $\mathbf{V}$, $\mathbf{P}$, and $\mathbf{M}$ introduce magnetic losses, and $\mathbf{E}$ and $\mathbf{E}$ represent the permeability and permittivity tensors of the material. Multiple loss tensors were included so that a perfectly matched layered (PML) absorber [20] can be introduced. By convention, Greek letters (all in italics) are used to represent second-rank material tensors whereas Latin letters are used to represent electric and magnetic field vectors. As a first step, let us carry out the projection of the curl equations. Multiplying (4) and (5) by $g^i$ leads to the component equations

$$\frac{1}{\sqrt{g}} \left( \frac{\partial \mathbf{E}_i}{\partial \mathbf{u}^j} - \frac{\partial \mathbf{E}_j}{\partial \mathbf{u}^i} \right) = -\frac{\partial \mathbf{H}^i}{\partial \mathbf{u}^j} - \sum_{s=1}^{3} \nu_s^i \mathbf{H}^s$$ (8)

$$\frac{1}{\sqrt{g}} \left( \frac{\partial \mathbf{H}_i}{\partial \mathbf{u}^j} - \frac{\partial \mathbf{H}_j}{\partial \mathbf{u}^i} \right) = -\frac{\partial \mathbf{E}^i}{\partial \mathbf{u}^j} + \sum_{s=1}^{3} \nu_s^i \mathbf{E}^s$$ (9)

where $g = \det(g_{ij})$. The set of indices $\{i,j,k\}$ must be cycled through the set $\{1,2,3\}$ in order to obtain each of the three components of the two curl equations. All quantities are assumed to be explicit functions of $\mathbf{u}^j$. As a second step, let us carry out the projection of the constitutive relations. Multiplying (6) and (7) with $g_{ij}$ yields

$$\frac{\partial \mathbf{H}^i}{\partial \mathbf{u}^j} + \sum_{s=1}^{3} \nu_s^i \mathbf{H}^s = \sum_{s=1}^{3} \left( \varepsilon^{-1} \varepsilon_{js}^s \mathbf{B}^s + \sum_{q=1}^{3} \mu q_j^q \mathbf{E}^s \right)$$ (10)

$$\frac{\partial \mathbf{E}^i}{\partial \mathbf{u}^j} + \sum_{s=1}^{3} \nu_s^i \mathbf{E}^s = \sum_{s=1}^{3} \left( \mu^{-1} \mu q_j^q \mathbf{H}^s + \sum_{q=1}^{3} \varepsilon q_j^q \mathbf{D}^s \right)$$ (11)

where, as before, all quantities are assumed to be explicit functions of $\mathbf{u}^j$. So far, the material in question has been assumed to be linear and nondispersive, but it can still be lossy, inhomogenous, and anisotropic. Under the proposed numerical procedure, (8)–(11) must be discretized using high-order finite differences; nevertheless, before their discretization is performed, let us elaborate on two important issues regarding the material tensors.

C. Material Tensors for a Simple Material

First, to avoid ending up with results that are difficult to check and interpret from a physical point of view, a lossless isotropic material with inhomogeneous $\varepsilon_r$ and $\mu_r$ is henceforth assumed. While this simplification facilitates the demonstration of the proposed procedure, keep in mind that lossy and anisotropic materials can be handled without any additional complications. Introducing anisotropic materials is simply a matter of choosing appropriate definitions of $\mathbf{M}$ and $\mathbf{E}$ while electric and magnetic losses can be added by making $\mathbf{V}$ and $\mathbf{P}$ nonzero outside of the absorber regions. Since only lossless materials are thus considered, the six loss tensors in (8)–(11) are used only for the purposes of introducing absorber losses. In such case, the electric loss tensors take the form

$$\nu_{\text{el}}^i = \left( \varepsilon_r \varepsilon_0 \right)^{-1} \text{diag}[\sigma^1, \sigma^2, \sigma^3]$$ (12)

$$\nu_{\text{el}}^i = \left( \varepsilon_r \varepsilon_0 \right)^{-1} \text{diag}[\sigma^3, \sigma^1, \sigma^2]$$ (13)

$$\nu_{\text{el}}^i = \left( \varepsilon_r \varepsilon_0 \right)^{-1} \text{diag}[\sigma^2, \sigma^3, \sigma^1]$$ (14)

where the electric conductivity function $\sigma^j$, which is nonzero only in the regions of the absorber, introduces attenuation along the $\mathbf{u}^j$ coordinate direction. Similarly, the magnetic loss tensors are of the form

$$\nu_{\text{ml}}^i = \left( \mu_r \mu_0 \right)^{-1} \text{diag}[\tilde{\sigma}^1, \tilde{\sigma}^2, \tilde{\sigma}^3]$$ (15)

$$\nu_{\text{ml}}^i = \left( \mu_r \mu_0 \right)^{-1} \text{diag}[\tilde{\sigma}^3, \tilde{\sigma}^1, \tilde{\sigma}^2]$$ (16)

$$\nu_{\text{ml}}^i = \left( \mu_r \mu_0 \right)^{-1} \text{diag}[\tilde{\sigma}^2, \tilde{\sigma}^3, \tilde{\sigma}^1]$$ (17)

where the magnetic conductivity $\tilde{\sigma}^j$ introduces attenuation along the $\mathbf{u}^j$ coordinate direction. For the absorber to be matched, the conductivities must satisfy

$$\sigma^j/\left( \varepsilon_r \varepsilon_0 \right) = \tilde{\sigma}^j/\left( \mu_r \mu_0 \right),$$ (18)

Guidelines for the selection of suitable conductivity profiles can be found in [20].

Second, the vocabulary used to describe the physical properties of materials is often inadvertently referenced to a Cartesian
coordinate system. For this reason, care must be exercised when physical interpretations are assigned to material properties.

\[
\mathcal{M}^{-1} = \frac{G}{(\mu_\mu \mu_0)} \quad \mathcal{E}^{-1} = \frac{G}{(\epsilon_\epsilon \epsilon_0)}
\]

(19)

where \(G\) is the metric tensor. In a covariant vector basis, the components of these two tensors are

\[
\mu_{\mu}^{-1} = g_{\mu \mu}/(\mu_\mu \mu_0) \quad \epsilon_{\epsilon}^{-1} = g_{\epsilon \epsilon}/(\epsilon_\epsilon \epsilon_0).
\]

(20)

If the mapping in (1) is to remain arbitrary, the entries of \(g_{\mu \mu}\) can have nonzero off-diagonal components. This does not imply that the material in question is anisotropic. The expressions in (20) simply state that, in a vector basis that is not Cartesian, the material tensors may have nonzero off-diagonal entries. Having introduced material tensor definitions for the assumed material properties, the next step is to discretize (8)–(11) using standard high-order finite differences.

III. DISCRETIZATION OF MAXWELL’S EQUATIONS

To facilitate the visualization of coordinate transformations and computed results, only 2-D structures are analyzed here. This does not represent any inherent limitation on the proposed procedure and all the concepts discussed hereby also apply to 3-D. Thus, let us start the discretization process by carrying out the 2-D reduction of (8)–(11). Assuming that \(x^3 = y^3, \sigma^3 = 0, \sigma^3 = 0, \) and that nothing varies along the \(x^3\) axis direction, the field components split into two independent sets. The first one is known as the TE\(^3\) set and it contains \(e_1^+, e_2^+, d^+, d^-, h_3^+, \) and \(b_3^+.\) The second one is known as the TM\(^3\) set and it contains the remaining field components. For conciseness, only the TE\(^3\) set is used here. For this set, (8)–(11) become

\[
\frac{1}{\sqrt{g}} \left( \frac{\partial e_2}{\partial x^3} - \frac{\partial e_1}{\partial x^1} \right) = -\frac{\partial b^3}{\partial t} - \nu_2^3 d^3
\]

(21)

\[
\frac{\partial h_3}{\partial t} + \nu_3^3 \frac{\partial b^3}{\partial t} = \frac{\partial e_1}{\partial x^1} - \nu_2^3 d^3
\]

(22)

\[
\frac{1}{\sqrt{g}} \frac{\partial b}{\partial x^1} = \frac{\partial e_2}{\partial t} + \nu_1^1 d^3
\]

(23)

\[
\frac{\partial e_1}{\partial t} = -\frac{\partial b^3}{\partial x^3} - \frac{\partial e_2}{\partial x^1} + \frac{\partial e_1^1}{\partial x^1} - \frac{\partial e_2^1}{\partial x^3} + \frac{\partial e_1^2}{\partial x^3} - \frac{\partial e_2^2}{\partial x^1}
\]

(24)

\[
\frac{1}{\sqrt{g}} \frac{\partial e_1^2}{\partial x^1} = -\frac{\partial b^3}{\partial x^3}
\]

(25)

\[
\frac{\partial e_2}{\partial t} + \gamma_2^2 e_2 = \frac{\partial e_1}{\partial t} + \frac{\partial e_2}{\partial t} + \frac{\partial e_1^1}{\partial x^1} - \frac{\partial e_2^1}{\partial x^3} + \frac{\partial e_1^2}{\partial x^3} - \frac{\partial e_2^2}{\partial x^1}
\]

(26)

Discretizing (21)–(26) using second-order accurate finite differences in time and standard high-order finite differences in space leads to the following time-stepping equations:

\[
\frac{\partial h_3^{n+1}}{\partial x^3} = \frac{2 - \Delta t \gamma_3^3}{2 + \Delta t \gamma_3^3} \frac{\partial h_3^{n-1}}{\partial x^3} + \left( \sum_{s=-L}^{L-1} c(s) \frac{\partial e_2^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^1} - \sum_{s=-L}^{L-1} c(s) \frac{e_1^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^2} \right)
\]

(27)

\[
h_3^{n+1,j+\frac{1}{2}} = \frac{2 - \Delta t \gamma_3^3}{2 + \Delta t \gamma_3^3} \frac{h_3^{n-1,j+\frac{1}{2}}}{2 + \Delta t \gamma_3^3} + \left( \sum_{s=-L}^{L-1} c(s) \frac{\partial e_2^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^1} - \sum_{s=-L}^{L-1} c(s) \frac{e_1^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^2} \right)
\]

(28)

\[
\frac{\partial e_1^{n+1}}{\partial x^1} = \frac{2 - \Delta t \gamma_1^1}{2 + \Delta t \gamma_1^1} \frac{e_1^{n-1}}{2 + \Delta t \gamma_1^1} + \left( \sum_{s=-L}^{L-1} c(s) \frac{\partial e_2^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^1} - \sum_{s=-L}^{L-1} c(s) \frac{e_1^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^2} \right)
\]

(29)

\[
ed_2^{n+1,j+\frac{1}{2}} = \frac{2 - \Delta t \gamma_2^2}{2 + \Delta t \gamma_2^2} \frac{e_2^{n}}{2 + \Delta t \gamma_2^2} + \left( \sum_{s=-L}^{L-1} c(s) \frac{\partial e_2^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^1} - \sum_{s=-L}^{L-1} c(s) \frac{e_1^{n+1,s+1,j+\frac{1}{2}}}{\Delta u^2} \right)
\]

(30)


TABLE I

<table>
<thead>
<tr>
<th>s</th>
<th>L = 1</th>
<th>L = 2</th>
<th>L = 3</th>
<th>L = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>s = 0</td>
<td>1/2</td>
<td>9/16</td>
<td>75/128</td>
<td>1225/2048</td>
</tr>
<tr>
<td>s = 1</td>
<td>0</td>
<td>-1/16</td>
<td>-25/256</td>
<td>-245/2048</td>
</tr>
<tr>
<td>s = 2</td>
<td>0</td>
<td>0</td>
<td>3/256</td>
<td>49/2048</td>
</tr>
<tr>
<td>s = 3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-5/2048</td>
</tr>
</tbody>
</table>

For negative values of s, use \( c(-s) = c(s-1) \).

An overbar over \( \hat{b} \) and \( \hat{d} \) indicates the normalizations \( \hat{b} = \sqrt{b} b \) and \( \hat{d} = \sqrt{d} d \), whereas an overbar over \( \varepsilon_{ij} \) and \( \mu_{ij} \) indicates the normalizations \( \varepsilon_{ij} = \varepsilon_{ij} / \sqrt{\varepsilon} \) and \( \mu_{ij} = \mu_{ij} / \sqrt{\mu} \). The weighting coefficients \( c(s) \) and \( c'(s) \) that appear in the finite-difference sums are given in Table I for the first four values of the stencil length \( L \). These coefficients were obtained by the method of undetermined coefficients described in [21]. The order of accuracy of the local truncation error of the finite-difference sums is \( 2L \); these sums are said to be of high-order only if \( L > 1 \). With a discretization of the TE model set now available, the 2-D waveguide in Fig. 2 will be used to illustrate how (27)–(32) can be used—together with a suitable coordinate transformation—to incorporate a conformal nonorthogonal structured grid. The structure in Fig. 2 is a 2-D reduction of a rectangular waveguide loaded with a cylindrical dielectric. This type of structure is often used as an unit cell in waveguide filters and it was previously analyzed numerically in [22]. To gain a better understanding of the properties of the proposed method, a slight variation of this problem where the cylindrical dielectric is replaced by a deformed square will also be considered. In both cases, the structure is excited by injecting an 80-GHz Gaussian pulse with a uniform front at the left end of the structure.

IV. INTRODUCING COORDINATE TRANSFORMATIONS

As stated earlier, a coordinate transformation must be constructed so that the \( u^j \) coordinate surfaces—or, in the 2-D case, the \( u^j \) coordinate lines—trace all the material boundaries of interest in \( x^i \) coordinates. Coordinate mappings that satisfy this requirement for the two problems under consideration are presented next. First, let us consider a mapping for the case of the circular dielectric inclusion depicted in Fig. 2.

A. Mapping for Gridding a Circular Dielectric

To avoid lengthy equations, let us split the coordinate transformation into three separate mappings: \( x^i = x^i(u^j) \), \( y^j = y^j(u^j) \), and \( x^i = x^i(y^j) \). The first of these coordinate transformations is given by

\[
v^j = 2 \left( u^j - x^i_c \right) / \ell_c \quad \text{for} \quad i = 1, 2
\]

where \( x^i_c \) represents the position of the center of the circular dielectric along the \( x^i \) axis (see Fig. 2) and \( \ell_c \) represents the side length of a square containing a circular grid deformation. The first equation of the second mapping is given by

\[
y^1 = \begin{cases} \frac{\text{sign}(v^1)}{C(\overline{\nu}) + \sqrt{R(\overline{\nu})^2 - \left( \frac{v^2 D(\overline{\nu})}{\overline{\nu}} \right)^2}}, & v^1 \geq v^2 \\ v^2, & v^1 < v^2 \end{cases}
\]

(34a)

\[
C(\overline{\nu}) = D(\overline{\nu}) - \sqrt{R(\overline{\nu})^2 - D(\overline{\nu})^2}
\]

(34b)

\[
D(\overline{\nu}) = \frac{\overline{\nu}}{\sqrt{2}} + \frac{(1 - 1/\sqrt{2})(\overline{\nu} - d/\ell_c)}{2(1 - d/\ell_c)} \cdot (1 + \tanh(8(\overline{\nu} - d/\ell_c)))
\]

(34c)

\[
R(\overline{\nu}) = \frac{d}{\ell_c} \cdot \left( 2 - 2 \left( \frac{1}{1 - \frac{d}{\ell_c}} \right) (1 + \tanh(8(\overline{\nu} - d/\ell_c))) \right)^{\frac{\overline{\nu}}{2} + \frac{1}{2}}
\]

(34d)

where \( d \) is given in Fig. 2. The second equation of the second mapping is given by

\[
y^2 = \begin{cases} \frac{v^1 D(\overline{\nu})}{\overline{\nu}}, & v^1 \geq v^2 \\ \text{sign}(v^2) \left( C(\overline{\nu}) + \sqrt{R(\overline{\nu})^2 - \left( \frac{v^1 D(\overline{\nu})}{\overline{\nu}} \right)^2} \right), & v^1 < v^2 \end{cases}
\]

(35)
Equations (34) and (35) are based on the coordinate transformation put forward in [23] with improvements to \( D(\gamma) \) and \( R(\gamma) \). Lastly, the third mapping is given by

\[
x^i = \ell_i y^j / 2 + x^i \quad \text{for} \quad i = 1, 2.
\]

Coordinate line plots for the overall transformation are given in Fig. 3. The area covered by the circular dielectric has been shaded in both plots of Fig. 3. Observe that the boundaries of the circular dielectric are indeed traced by the coordinate lines. Furthermore, the area of the dielectric gets mapped from a circle in Cartesian coordinates to a square in computational coordinates. Thanks to this property of the mapping, the boundary conditions of the dielectric can be enforced in computational coordinates [24].

### B. Mapping for Gridding a Deformed Square

With (33)–(36) now in place, the next step is to consider a coordinate transformation for the case where a dielectric with a deformed square shape is placed inside the waveguide. Such a shape could be introduced by using

\[
x^1 = u^1 \left(1 - \alpha \sin \left( \frac{\pi u^1}{\ell_1} \right) \sin^4 \left( \frac{\pi u^2}{\ell_2} \right) \psi(u^1) \right)
\]

\[
\psi(u^1) = \frac{1}{2} \left( \tanh \left( 20 \left( \frac{u^1}{\ell_1} - \frac{15}{100} \right) \right) - \tanh \left( 20 \left( \frac{u^1}{\ell_1} - \frac{85}{100} \right) \right) \right)
\]

\[
x^2 = u^2 \left(1 - \alpha \cos \left( \frac{\pi u^1}{\ell_1} \right) \sin^4 \left( \frac{\pi u^2}{\ell_2} \right) \psi(u^1) \right)
\]

where \( \ell_1 \) and \( \ell_2 \) were defined earlier in Fig. 2 and \( \alpha \) is a parameter that determines the deformation of the dielectric object. The coordinate transformation presented earlier in (33)–(36) maps a square of side length \( d \) in computational coordinates to a circle of diameter \( d \) in Cartesian coordinates. The coordinate transformation given in (37) and (38) maps a square of side length \( d \) in computational coordinates to a deformed square in Cartesian coordinates as illustrated in Fig. 4. Note that, if \( \alpha = 0 \), no deformation is introduced at all; however, as \( |\alpha| \) becomes greater than zero, the dielectric material inside the waveguide gets increasingly deformed and the coordinate transformation becomes increasingly nonorthogonal. The two coordinate transformations defined in (33)–(34) are nonorthogonal, and, for completeness, a precise definition of orthogonality is given in Appendix B. Having defined suitable coordinate transformations, (27)–(32) can now be solved assuming that the problem’s geometry is defined by the two coordinate transformations in (33)–(38). But, before any results are presented, let us briefly discuss the issue of how coordinate transformations are generated for more complex waveguide geometries.

### C. Mappings for Gridding Other Geometries

For the two relatively simple problem geometries considered here, it is easy to generate suitable coordinate transformations analytically. Examples of other 2-D coordinate transformations that can be used to model a waveguide bend and a joint between two waveguides of different widths are given in Figs. 5 and 6, respectively. The equations that define these mappings are given in Appendix B. When dealing with complex designs, one has to resort to general purpose structured grid generation algorithms such as those described in [9]–[11]. To show that these techniques are not limited to simple geometries, consider the more complex 2-D waveguide structure shown in Fig. 7. This structure is a 2-D simplification of a substrate integrated waveguide joint that connects two waveguide sections of different widths. Note that, in substrate integrated waveguide technology, closely spaced cylindrical holes filled with solder are used to create the sidewalls of the waveguides. A coordinate transformation to grid this structure was generated numerically using a commercial structured grid generation software package [25]. Coordinate line plots of the transformation are given in Figs. 8 and 9. When (1) is defined by a discrete set of points, keep in mind that
As a result, established techniques, such as those used in [25], can be exploited to solve the geometrical part of the problem.

V. SOLVING THE DISCRETE EQUATIONS

Now, going back to the issue of solving (27)–(32) for the two waveguide geometries under consideration, four different grids were generated using each of the two coordinate transformations in (33)–(38). The parameters that define each gridding are given in Table II. For all four grids, results were obtained using second-order \( (2L = 2) \), fourth-order \( (2L = 4) \), as well as sixth-order \( (2L = 6) \) accurate finite differences. The corresponding execution times are given in Table III. The execution times are the same for both coordinate transformations since the number of grid cells is the same in both cases. Snapshots of the computed results are given in Figs. 10 and 11. The coordinate transformation parameters were selected as follows: for the mapping in (37) and (38), \( \alpha = 0.16 \) and \( \alpha = 0.25 \) were used, and, for the mapping in (33)–(36), \( \ell_c = 0.8 \text{ cm} \) was used. Note that, prior to the time stepping, the excitation must be mapped from Cartesian coordinates to computational coordinates. Similarly, once the time stepping is completed, the obtained results must be mapped from computational coordinates to Cartesian coordinates. These mappings can be accomplished using the formulas described in Appendix A. The impact of using second-, fourth-, or sixth-order finite differences will be discussed shortly, but, before doing so, let us briefly outline how errors were defined and measured.

A. Local Truncation Error and the Global Error

In the FDTD literature, two types of errors are usually mentioned: the local truncation error and the global error. The local truncation error refers to the error introduced by interpolating a function or estimating its derivative based on a discrete sample of its values. For instance, one of the two approximations applied to (25) was derived from

\[
\frac{\partial h_3}{\partial u_1} = \sum_{s=-L}^{L-1} c'(s) \left[ \frac{h_3^{r} (u_1 + s + \frac{1}{2})}{\Delta u_1} + O \left( \Delta u_1^{2L} \right) \right].
\]

The local truncation error refers to the error introduced by dropping the error term on the right hand side, and its order of accuracy is \( 2L \). Observe that, as \( L \) becomes larger, the approximation converges faster to the true value of the derivative as \( \Delta u_1 \) becomes smaller. The global error, on the other hand, refers to the difference between the obtained numerical solution \( \hat{H}_3 \) and its exact value \( H_3 \). As a measure of the global error, an estimate of the global error norm

\[
\| \hat{H}_3 - H_3 \|_q = \left( \int_0^{\ell_1} \int_0^{\ell_2} |\hat{H}_3 - H_3|^q \, dx \, dv \right)^{1/q}
\]

is usually provided. The above \( q \)-norm can be estimated from a given numerical solution by carrying out the integrals numerically. Estimates of the 2-norm of the global error for all the simulations outlined in Tables II were obtained and are given in Figs. 12–14. To compute these error results, the dielectric materials had to be removed (i.e., \( \varepsilon_r \) was set to unity everywhere) while keeping the coordinate transformations in place. In this
Fig. 7. Diagram of a 2-D waveguide joint used to connect two waveguide sections of different widths. The sidewalls of the waveguide are created by closely spaced circles of a perfectly conducting material. This structure is a 2-D simplification of a substrate integrated waveguide joint where closely spaced cylindrical holes filled with solder are used to create the sidewalls of the waveguides.

Fig. 8. Coordinate line plots in Cartesian coordinates of a 2-D block structured coordinate transformation constructed to model the waveguide joint described in Fig. 7. The mapping was generated using a general-purpose block-structured grid generation software [25].

Fig. 9. Detailed view of the coordinate transformation in Fig. 8 for the area surrounding one of the circular conductors that make up the walls.

way, the effects of the grid deformations can be captured in a situation where the exact solution is known.

B. Convergence Properties of the Proposed Method

When finite differences with a local truncation error of order $2L$ are applied to a uniform rectangular grid, it is expected that $\|H_3 - H_3\|_2 \propto \Delta^p$, where $p \approx 2L$ and $\Delta = \Delta u^1 = \Delta u^2$. If nonorthogonal grids are used, the order of accuracy of the global error norm $p$ can be considerably lower than $2L$; nevertheless, it is still possible to make computational efficiency gains by exploiting the convergence properties of high-order finite differences. The grid parameters given in Table II were chosen so that this point could be illustrated. The slope of the lines given in the plots on Figs. 12–14 provide a measure of $p$ for a given value of $L$. Let us begin by looking at the results for the mapping in (37) and (38) with $\alpha = 0.16$, which are given in Fig. 12. These correspond to the scenario depicted in Fig. 4. A linear fit to the data shows that the slopes are 1.41 for the second-order case ($2L = 2$) and 3.40 for the fourth-order case ($2L = 4$). For the sixth-order case ($2L = 6$), a linear fit of the slope was not made because the two leftmost data points lie on the threshold at which making the grid finer will no longer produce a drop.
in the global error norm. If needed, this threshold can be lowered by reducing $\Delta t$. Observe that the coarsest grid (Grid #1 in Table II) with sixth-order accurate finite differences produces more accurate results than the finest grid (Grid #4 in Table II) with second-order accurate finite differences, yet the coarsest grid with sixth-order finite differences has a lower execution time than the finest grid with second-order finite differences.

This demonstrates that the computational efficiency gains normally associated with the use of high-order finite differences on uniform grids can also be obtained on nonorthogonal grids.

Unfortunately, the use of high-order finite differences does not always lead to an improved global error behavior. The results for the mapping in (33)–(36), which are given in Fig. 13, illustrate this point. For this particular coordinate transformation, increasing $L$ does not help at all. So, under what conditions can the convergence properties of the global error be improved by using high-order finite differences? In general, to obtain a global error with an order of accuracy $p$ that increases by increasing the local truncation error of the finite differences employed, two conditions must be met. The coordinate transformation must be constructed so that:

1) the metric tensor components are piecewise continuous;
2) any existing discontinuities must not cut through the interior of the grid cells.

The metric components produced by (37) and (38) are continuous everywhere, but the metric components produced by (33)–(36) contain a discontinuity that cuts diagonally through the grid cells at the points where the coordinate lines have a sudden change in direction (see Fig. 3). For this reason, the results shown in Fig. 13 do not show an improved global error.
is increased, it would be necessary to introduce significant modifications to the presented methodology.

C. Role of Grid Orthogonality

To observe the effect that grid orthogonality plays in the convergence properties of the global error, let us now turn our attention to the results for the mapping in (37) and (38) with $\alpha = 0.25$, which are given in Fig. 14. As stated earlier, as $|\alpha|$ is increased, the coordinate transformation in (37) and (38) becomes increasingly nonorthogonal. A linear fit to the data shows that the slopes are 0.98 for the second-order case ($2L = 2$), 2.84 for the fourth-order case ($2L = 4$), and 4.01 for the sixth-order case ($2L = 6$). If these results are compared to those in Fig. 12, it is easy to see that by making the coordinate transformation increasingly nonorthogonal the order of accuracy of the global error $p$ decreases for a given value of $L$. This effect suggests that the use of high-order finite differences should be strongly considered when highly nonorthogonal grids are employed since $p$ can be very low for the second order algorithm in such cases.

VI. STABILITY CONSIDERATIONS

As a final consideration, let us briefly discuss the stability properties of the proposed nonorthogonal gridding method. When (27)–(32) are leapfrogged in time, the time step must be small enough for the algorithm to be stable. A sufficient condition for stability is given by

$$
\Delta t \leq \left( \frac{\sum_{m=0}^{L-1} |\epsilon(m)|}{L \Delta x_j} \right)^2 \times \max_{i,j} \left| \frac{1}{N \Delta z_j} \right|
$$

This upper bound was derived assuming that the computational domain is infinite; however, as long as the chosen grid is not awkwardly deformed in the absorber regions, the upper bound in (41) guarantees that the proposed algorithm is stable. In the past, when the use of nonorthogonal structured grids has been explored, it has been pointed out that late-time instabilities can occur. To prevent this problem it was noted earlier that “... in the practical implementation it must be ensured that the metric expressions as well as the material coefficients ... are evaluated in a symmetrical way” [26]. The proposed method introduces symmetric material tensors which, as shown in (30) and (32), are evaluated symmetrically on the grid cells. Therefore, by following the exact same argument put forward in [26], the nonorthogonal gridding procedure proposed here can be shown to always be stable.

VII. CONCLUSION

A general method for introducing structured nonorthogonal grids into high-order FDTD schemes for solving Maxwell’s equations has been presented. It was demonstrated that the proposed method can significantly improve the global error convergence if the appropriate metric continuity conditions are satisfied. In addition, it was shown that all the geometrical information of a given problem can be introduced—through a coordinate transformation—prior to the discretization of Maxwell’s equations. As result, the proposed procedure can be applied to any high-order FDTD method originally formulated on a uniform Cartesian grid. For the relatively simple geometries analyzed here, it is was possible to find suitable coordinate transformations analytically; however, for more complex geometries, one has to resort to general purpose algorithms such as those described in [9]–[11]. While many structured [14]–[16] grid generation strategies have been proposed in the past, only the rigorous use of differential geometry proposed in [13] leads to an algorithm with a global error whose order of accuracy increases as the stencil size of the finite differences employed is increased.

APPENDIX A

SUMMARY OF THE EMPLOYED TENSORIAL NOTATION

Here, all of the notational conventions employed in this manuscript to express vector and second-rank tensor components are described. For completeness, the equations that relate all the different representations are provided. For a primer on differential geometry, the reader is referred to [27].
A. Vector Components Expressed in a Curvilinear Basis

Any vector field may be expressed in either a covariant or a contravariant basis. To avoid confusion among different field quantities, lowercase letters are used to express vector components in terms of a general curvilinear vector basis whereas uppercase letters are used to express vector components in terms of a Cartesian basis. Using this convention, the covariant and contravariant components of \( \mathbf{E} \) in a general curvilinear vector basis are given, respectively, by

\[
e_i = \mathbf{E} \cdot \mathbf{g}_i \quad e^i = \mathbf{E} \cdot \mathbf{g}^i.
\]  

Similarly, the components of \( \mathbf{E} \) in a Cartesian vector basis are given by

\[
E_i = \mathbf{E} \cdot \mathbf{i}_i \quad E^i = \mathbf{E} \cdot \mathbf{i}^i.
\]

For further clarity, subscripts are always used to identify covariant components and superscripts are used to identify contravariant components. Note that, while the placement of indices is immaterial in a Cartesian basis (i.e., \( E_i = E^i \) and \( \mathbf{i}_i = \mathbf{i}^i \)), this is not the case in a general curvilinear basis (i.e., \( e^i \neq e_i \) and \( g_i \neq g^i \)). In fact, to convert vector components from a covariant to a contravariant basis and vice versa, the following relations must be employed:

\[
e^i = \sum_{j=1}^{3} g^{ij} e_j \quad e_i = \sum_{j=1}^{3} g_{ij} e^j.
\]

Observe that the symmetry of the metric tensor allows us to swap indices arbitrarily in the above two relations. To employ the proposed method, it is necessary to know how to convert vector components from a Cartesian basis representation to a general curvilinear basis representation and vice versa. The forward mapping can be accomplished by using

\[
e_i = \sum_{j=1}^{3} \frac{\partial x^i}{\partial x^j} E_j \quad e^i = \sum_{j=1}^{3} \frac{\partial x^i}{\partial x^j} E^j
\]

while the reverse mapping can be accomplished by using

\[
E_i = \sum_{j=1}^{3} \frac{\partial x^j}{\partial x^i} e_j \quad E^i = \sum_{j=1}^{3} \frac{\partial x^j}{\partial x^i} e^j.
\]

The same notational conventions described here for vectors can be easily extended to second-rank tensors. Such extension, as well as the second-rank equivalents of (44)–(46), are discussed next.

B. Tensor Components Expressed in a Curvilinear Basis

As a first step, let us look at how second-rank tensors are projected onto a covariant or contravariant vector basis. To create a second-rank basis from the covariant or contravariant basis vectors, it is necessary to use the dyadic product between two vectors. This product, whose end result is a second-rank tensor, is denoted by the symbol \( \otimes \) and it is defined by the properties

\[
(\mathbf{A} \otimes \mathbf{B})\mathbf{C} = \mathbf{A}(\mathbf{B} \cdot \mathbf{C}) \quad \mathbf{A}((\mathbf{B} \otimes \mathbf{C}) = (\mathbf{A} \cdot \mathbf{B})\mathbf{C}
\]  

where \( \mathbf{A}, \mathbf{B}, \) and \( \mathbf{C} \) are arbitrary vectors. A second-rank tensor basis can be created from any of the four possible tensor product combinations of \( \mathbf{g}_i \) and \( \mathbf{g}^i \). Therefore, any second-rank tensor \( \mathbf{E} \) has four possible representations

\[
\mathbf{E} = \sum_{i=1}^{3} \sum_{j=1}^{3} \varepsilon_{ij} \mathbf{g}_i \otimes \mathbf{g}_j = \sum_{i=1}^{3} \sum_{j=1}^{3} \varepsilon_{ij} \mathbf{g}^i \otimes \mathbf{g}^j
\]

\[
= \sum_{i=1}^{3} \sum_{j=1}^{3} \varepsilon_{ij} \mathbf{g}_i \otimes \mathbf{g}^j = \sum_{i=1}^{3} \sum_{j=1}^{3} \varepsilon_{ij} \mathbf{g}^i \otimes \mathbf{g}_j.
\]

In the above, \( \varepsilon_{ij} = \mathbf{g}_i \mathbf{E} \mathbf{g}_j \) is referred to as the \( (i,j) \)-th covariant component and \( \varepsilon^{ij} = \mathbf{g}^i \mathbf{E} \mathbf{g}^j \) is the \( (i,j) \)-th contravariant component of \( \mathbf{E} \). Unlike vectors, second rank tensors can have mixed covariant–contravariant representations \( \varepsilon^i_j = \mathbf{g}^i \mathbf{E} \mathbf{g}_j \) and \( \varepsilon_i^j = \mathbf{g}_i \mathbf{E} \mathbf{g}^j \). As before, lowercase letters are used to express tensor components in terms of a general curvilinear basis while uppercase letters are used to express tensor components in a Cartesian basis. For further clarity, Greek letters (all in italics) are always used to represent second rank tensors while Latin letters are used to represent vectors. To express material properties, it is necessary to know how to map the components of \( \mathbf{E} \) from a general curvilinear basis to a Cartesian basis and vice versa. So, as a second step, let us describe how to accomplish these mappings. For covariant and contravariant representations, the forward mapping is accomplished by

\[
\varepsilon_{is} = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^q}{\partial x^i} \frac{\partial x^j}{\partial x^s} \varepsilon_{ij} \quad \varepsilon^is = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^i}{\partial x^q} \frac{\partial x^s}{\partial x^j} \varepsilon^{ij}
\]

while the reverse mapping is accomplished by

\[
\varepsilon_{is} = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^q}{\partial x^i} \frac{\partial x^j}{\partial x^s} \varepsilon_{ij} \quad \varepsilon^is = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^i}{\partial x^q} \frac{\partial x^s}{\partial x^j} \varepsilon^{ij}.
\]

When a mixed covariant and contravariant representation is used, the partial derivatives must be flipped accordingly in order to obtain the appropriate forward and reverse mappings, given as

\[
\varepsilon^i_s = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^i}{\partial x^q} \frac{\partial x^j}{\partial x^s} \varepsilon^{qj} \quad \varepsilon_s^i = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^q}{\partial x^i} \frac{\partial x^s}{\partial x^j} \varepsilon^{qj}
\]

\[
\varepsilon^i_s = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^i}{\partial x^q} \frac{\partial x^j}{\partial x^s} \varepsilon^{qj} \quad \varepsilon_s^i = \sum_{q=1}^{3} \sum_{j=1}^{3} \frac{\partial x^q}{\partial x^i} \frac{\partial x^s}{\partial x^j} \varepsilon^{qj}.
\]
As a third step, let us show how the metric tensor can be used to switch back and forth between covariant and contravariant representations. This operation is also known as the raising and lowering of indices. All four possible component representations of $\mathbf{E}$ are related to each other according to

$$
\varepsilon_{is} = \sum_{q=1}^{3} g_{iq} \varepsilon_q^j = \sum_{j=1}^{3} \varepsilon_i^j g_{js} = \sum_{q=1}^{3} \sum_{j=1}^{3} g_{iq} \varepsilon_q^j g_{js} \tag{57}
$$

$$
\varepsilon_{is} = \sum_{q=1}^{3} g^i_q \varepsilon_s^q = \sum_{j=1}^{3} \varepsilon_i^j g^j_s = \sum_{q=1}^{3} \sum_{j=1}^{3} g^i_q \varepsilon_s^q g^j_s. \tag{58}
$$

Finally, for (57) and (58) to be consistent, it is required that

$$
\sum_{s=1}^{3} g^i_j g_{js} = \delta^i_j \tag{59}
$$

where $\delta^i_j$ is the Kronecker delta symbol. This can be proved from (2) and (3) by using the chain rule. For convenience, it is assumed that the location of the indices in the Kronecker delta symbol is unimportant (i.e., $\delta_{ij} = \delta^i_j = \delta^j_i = \delta^{ji}$).

**APPENDIX B**

**DEFINITION OF ORTHOGONALITY**

Here, a definition of orthogonality—a term used throughout this manuscript to describe coordinate transformations—is given. A coordinate transformation and its associated grid is said to be orthogonal if the basis vectors remain perpendicular to each other in both the Cartesian and the computational reference frames. Using (2) and (3), this condition can be stated concisely as

$$
g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j = 0 \quad \text{or} \quad g^{ij} = \mathbf{g}^i \cdot \mathbf{g}^j = 0, \quad \text{for all, } i \neq j. \tag{60}
$$

When coordinate line plots (such as those in Figs. 3–6) are available, it is very easy to identify whether or not a coordinate transformation is orthogonal. Consider the covariant basis vectors $\mathbf{g}_i$ and $\mathbf{g}_j$. Since these two vectors are tangential to the $u^i$ and $u^j$ coordinate lines, respectively, then the condition in (60) implies that the $u^i$ coordinate lines always intersect the $u^j$ coordinate lines at 90° angles. An example of an orthogonal coordinate transformation that can be used to model a waveguide bend is [28]

$$
x^1 = \begin{cases} u^1, & \text{for } u^1 \text{ in } \mathcal{I}_1 \\ \ell_b + \frac{u_b + u^2 - u_b}{\csc \left( \frac{\ell_b}{\csc \varphi_b} \right)}, & \text{for } u^1 \text{ in } \mathcal{I}_2 \\ \ell_b + \frac{u_b + u^2 - u_b}{\csc \varphi_b} + \frac{u^1 - \ell_b - \ell_b \varphi_b}{\sec \varphi_b}, & \text{for } u^1 \text{ in } \mathcal{I}_3 \end{cases} \tag{61}
$$

where $\ell_b$ is the width of the waveguide, $\ell_b$ is the radius of curvature of the bend, $\varphi_b$ is the bending angle, and $\ell_b$ is the distance from the beginning of the waveguide to the beginning of the bend. A diagram illustrating these definitions was provided earlier in [29]. Coordinate line plots for this transformation are given in Fig. 5, and, as expected, the coordinate lines always intersect at right angles. An example of a nonorthogonal coordinate transformation that can be used to model a joint between two waveguide sections of different widths is given by

$$
x^1 = u^1 - \frac{u_b}{2} \left( 1 + \alpha_d \tanh \left( 4k \left( \frac{u^1}{\ell_d} - \frac{1}{2} \right) \right) \right) \tag{62}
$$

where $u_b$ is the width of the waveguide, $\alpha_d$ is a unitless factor that determines the change in width, $\beta_d$ is a unitless factor that determines the length of the joint and $\ell_d$ is the length of the waveguide. Coordinate line plots for this transformation are given in Fig. 6, and, as expected, the coordinate lines do not always intersect at right angles. As pointed out in Section V, the closer a coordinate transformation comes to being orthogonal, the better the convergence properties of the proposed method tend to be.

**REFERENCES**


