Collocated electrodynamic FDTD schemes using overlapping Yee grids and higher-order Hodge duals

C. Deimert, M.E. Potter*, M. Okoniewski

Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive N.W. Calgary, Alberta, T2N 1N4, Canada

A R T I C L E   I N F O

Article history:
Received 19 January 2016
Received in revised form 25 August 2016
Accepted 27 August 2016
Available online 5 September 2016

Keywords:
Discrete exterior calculus
FDTD methods

A B S T R A C T

The collocated Lebedev grid has previously been proposed as an alternative to the Yee grid for electromagnetic finite-difference time-domain (FDTD) simulations. While it performs better in anisotropic media, it performs poorly in isotropic media because it is equivalent to four overlapping, uncoupled Yee grids. We propose to couple the four Yee grids and fix the Lebedev method using discrete exterior calculus (DEC) with higher-order Hodge duals. We find that higher-order Hodge duals do improve the performance of the Lebedev grid, but they also improve the Yee grid by a similar amount. The effectiveness of coupling overlapping Yee grids with a higher-order Hodge dual is thus questionable. However, the theoretical foundations developed to derive these methods may be of interest in other problems.

© 2016 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The finite-difference time-domain (FDTD) method—particularly on the standard Yee grid [1]—is a popular technique for the numerical solution of Maxwell's equations. A key feature of the Yee grid is that it staggered the electromagnetic field components, so that they are not collocated in space. This configuration is advantageous when calculating the curl operators, but it is not always optimal when implementing materials. Anisotropic materials are one example, as they effectively require all three field components to be known at the same location [2]. Another example is that of a sharp dielectric interface, which, for a proper treatment, requires an effectively anisotropic permittivity at the interface [3].

The Lebedev grid has been proposed as an alternative to the standard Yee grid for handling these special cases [4]. Its collocated fields allow for improved performance whenever anisotropic (or effectively anisotropic) media arise. The downside is that it has poor performance in simpler cases like homogeneous, isotropic media. This is because, in isotropic media, the second-order Lebedev method is equivalent to four overlapping, uncoupled Yee methods running concurrently. Effectively, three redundant simulations are being run without any boost to the accuracy.

In this paper, we explore the use of discrete exterior calculus (DEC) [5] as a way to couple these overlapping Yee grids. By respecting the inherent space-time geometry of electrodynamics, DEC methods automatically obey a number of discrete conservation rules which mirror those of the continuous case [5]. It should be noted that the use of differential forms for the particular purpose of designing improved finite-difference, finite-volume, and finite-element discretizations for electromagnetism predates the coinage of the term ‘DEC’. Despite the more contemporary terminology, we use the term DEC

* Corresponding author.
E-mail address: mpotter@ucalgary.ca (M.E. Potter).

http://dx.doi.org/10.1016/j.jcp.2016.08.048
0021-9991/© 2016 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
to encompass all manner of methods which seek to design differential forms on discrete topologies for application to the solution of Maxwell’s equations. DEC has been proposed as a natural way to discretize Maxwell’s equations for numerical solution, with some representative examples in this field being [5–15].

When applying DEC to the problem of the Lebedev grids, it becomes clear that higher-order Hodge dual operators are necessary in order to couple overlapping Yee meshes. Though only second-order Hodge duals appear to have been developed in DEC, the necessary ideas have been developed in [14] and [16]. In this paper, we build on these, proposing a general method for deriving higher-order Hodge duals using basis expansions.

Note that the same general principles of ‘DEC’ can be found, in a different parlance, incorporated by the Finite Integration Technique (FIT), and in particular in [17] which, although differing in many details, also develops a higher-order FIT scheme (which on a rectangular mesh recovers the standard Yee FDTD grid) based on a factorization of the Hodge operator (dubbed “material matrices” in their context) and the exterior derivative (dubbed matrix-free curl and div matrices in their context). Also, the derivation in [17] is specific to a Yee grid; in contrast, this manuscript presents a general framework for designing higher-order Hodge duals on multiple variations of dual grids which may contain collocated or uncollocated fields.

We begin with a review of DEC and introduce some key definitions. Next, we develop higher-order Hodge duals in one dimension. As a proof-of-concept, we show that well-known fourth-order difference approximations can be reproduced under this framework. Next, we develop a general method for deriving Hodge duals in three dimensions. Using simple polynomial basis forms, we derive higher-order methods on the standard Yee grid, the collocated Lebedev grid, and a collocated cubic grid. The dispersion characteristics of these methods are then examined analytically.

A detailed comparison of the dispersion characteristics shows that the higher-order Hodge duals do indeed lead to a significant improvement over second order methods, as expected. Unfortunately, the results for the higher-order collocated methods are somewhat underwhelming compared with the higher-order Yee method in homogeneous, isotropic media. The collocated methods would likely fare better in anisotropic or inhomogeneous media (as in the second-order case), but there does not appear to be any special benefit from coupling overlapping grids. However, this analysis does show that the basis expansion approach to higher-order Hodge duals is a reasonable one. Since DEC brings the additional benefit of automatically obeying discrete conservation rules, these methods may be of interest in more complicated situations where the standard Yee method is suboptimal.

2. Computational electrodynamics using differential forms

To develop a method for coupling Yee grids, we turn to the machinery of differential forms. Unlike vector fields, differential forms explicitly distinguish between “field intensities” and “flux densities.” Doing this reveals that many of vector field concepts can actually be generalized and simplified [18,19]. (For example, the gradient, curl, and divergence operators all become special cases of the exterior derivative.) More importantly for this work, differential forms provide a great deal of insight into the development of numerical methods: particularly through discrete exterior calculus (DEC) [5].

We will now elaborate on this point further, beginning with a review of differential forms and discrete exterior calculus in computational electrodynamics. We will see how the popular Yee method arises naturally from this approach, and we will explore the important role played by the Hodge dual. These concepts are not new, but developing this deeper understanding of the Yee method will give us the tools we need to generalize it in a sensible way, laying a foundation for the collocated methods derived in later sections.

Differential forms are not reviewed here, but a short introduction may be found in [18].

2.1. Differential forms and Maxwell’s equations

We begin with Maxwell’s equations. In the language of differential forms, they can be written as

\[ d_t B = -d_s E - M \]  \hspace{1cm} (1a)
\[ d_t D = +d_s H - J \]  \hspace{1cm} (1b)
\[ d_s B = \rho^* \]  \hspace{1cm} (1c)
\[ d_s D = \rho \]  \hspace{1cm} (1d)

These equations are visibly similar to their familiar vector calculus form, but the fields have been replaced by forms, as summarized in Table 1. For the purposes of this paper, we will be working in $(3 + 1)$-dimensional space-time, which means that each quantity is, separately, a differential form on 3D space, $\{(x_1, x_2, x_3) \in \mathbb{R}^3\}$, and a differential form on 1D time, $t \in \mathbb{R}$. For clarity, we use the notation “$(m, n)$-form” to denote a field which is an $m$-form in space and an $n$-form in time. We also distinguish between the exterior derivative in time, $d_t$, and the exterior derivative in space, $d_s$. Effectively, when compared to the vector calculus approach, $d_t$ replaces the time derivative and $d_s$ replaces the gradient, curl, and divergence.

As an aside, it should be noted that the classification in Table 1 is somewhat non-standard. Typically, all the listed quantities would be treated as forms only in space, and not in time. However, in the full spacetime approach, this leads to $E, H, J,$ and $M$ always appearing in exterior products with $dt$ ($d$ being the 4D spacetime exterior derivative). For example, the key quantities are the spacetime two-forms.
Table 1
Differential form degrees of the electromagnetic quantities on \((3 + 1)\) space-time.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Degree (space)</th>
<th>Degree (time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field intensities</td>
<td>(E, H)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Flux densities</td>
<td>(D, B)</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Current densities</td>
<td>(J, M)</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Charge densities</td>
<td>(\rho, \rho^*)</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
F = E \wedge dt + B \\
G = H \wedge dt + D
\]  
(2a)  
(2b)

This “space-only” convention is fine, but it leads to unnecessarily complicated notation when dealing with \((3 + 1)\)-dimensional spacetime. So, for this paper, we are simply choosing to treat \(E \wedge dt\) as “the electric field,” and we are denoting it as \(E\). Then \(E\) becomes a 2-form in 4-dimensional spacetime, or a \((1, 1)\)-form in \((3 + 1)\)-dimensional spacetime. We treat \(H, J, \) and \(M\) similarly. This choice of convention is not strictly necessary, but it simplifies our notation and even adds some clarity when discretizing temporal derivatives.

Now, in addition to Maxwell’s equations, we require constitutive relations. In vector calculus, the constitutive relations are mostly used to describe materials, but with differential forms they play a much more important role. This is because the constitutive relations involve a conversion between different degrees of forms, even in a material-free environment. For example, since \(E\) is a \((1, 1)\)-form and \(D\) is a \((2, 0)\)-form, we cannot simply set one to be a scalar multiple of the other. In general, to make these conversions, we must introduce the Hodge dual operators on space and time: \(\star_s\) and \(\star_t\), respectively. The constitutive relations are then

\[
D = \star_t \star_s \varepsilon E \\
B = \star_s \mu H \\
J = \star_s \sigma E \\
M = \star_s \sigma^* H
\]  
(3a)  
(3b)  
(3c)  
(3d)

These relations and the Hodge dual operator will play an important role in the numerical methods that follow.

2.2. Discrete exterior calculus

Discrete exterior calculus arises naturally from the differential forms picture of electromagnetism. As noted above, the strength of DEC is its respect for the inherent (space-time) geometry of electromagnetism, leading to methods which automatically have a number of desirable conservation properties. For example, in DEC methods, the discrete analogs of \((d_B \rho)\) and \((d_D \rho)\) are automatically conserved, just as they are in the analytical case [5]. On a deeper level, DEC methods are automatically multi-symplectic [5], which is a geometric property of Lagrangian field theories [20].

Using DEC to discretize Maxwell’s equations essentially involves three steps.

1. Discretize the simulation domain by dividing it into cells.
2. Discretize the fields by integrating them over boundaries of those cells.
3. Discretize the operators by preserving as many properties of the continuous operators as possible.

This is similar to finite-volume methods (e.g., see [21]), but the process is quite natural and guided by the use of differential forms.

2.3. Discretizing the domain with overlapping meshes

DEC allows for a great deal of flexibility in discretizing the simulation domain. The usual approach (e.g., [15] or [5]) is to break the simulation domain into a set of space-time volumes. These could be simple tetrahedrons or rectangular boxes, for example, or they could be more complicated space-time volumes. The typical approach then uses this mesh to define a second, dual mesh, which allows for a convenient definition of the Hodge star. From there, a discrete set of points, lines, surfaces, and volumes are defined using the boundaries of these so-called “primary” and “dual” meshes. The fields are discretized by integrating them over lines, surfaces, or volumes (depending on what type of differential form they are).

By necessity, the above approach leads to fields which are not collocated: since the boundaries of a volume cell do not overlap, different field components will end up being stored at different locations in space. To remedy this, we introduce the notion of overlapping meshes. By using a number of overlapping meshes, we can create a set of overlapping lines, surfaces, and volumes, allowing the discretized fields to be collocated in space and/or time. For example, it is known that four staggered Yee (cubic) meshes can be used to create a collocated Lebedev grid [4].
To generalize slightly, we will define a discrete set of lines and surfaces in space-time.

- \( i \) is the spatial index corresponding to the location \((i_1 \Delta x, i_2 \Delta x, i_3 \Delta x)\).
- \( n \) is the temporal index corresponding to the time \( n \Delta t \).
- \( \Sigma_i[n] \) is the square of side-length \( 2 \Delta x \), centred at time \( n \) and position \( i \), with surface normal in the direction of the \( u \)-axis.
- \( \Gamma_i[n] \) is the line of length \( 2 \Delta x \), centred at time \( n \) and position \( i \), and parallel to the \( u \)-axis.
- \( \Gamma_i[n] \) is the time interval of length \( 2 \Delta t \), centred at position \( i \) and time \( n \).

By restricting the values of \( i \) and \( n \), we obtain different grids. For example, on one possible Yee grid, the surface \( \Sigma_1[i] \) would only exist when \( i_1 \) is even and \( i_2, i_3 \) is odd. On the other hand, if we allow \( \Sigma_i[n] \) and \( \Gamma_i[n] \) to exist for all integer triplets \( i \), then we obtain a fully-collocated grid which is equivalent to eight overlapping Yee grids.

2.4. Discretizing the fields

With the simulation domain discretized into lines and surfaces, we can easily discretize the fields. The degree of differential form immediately indicates which region(s) of space-time it should be integrated over. For example, since the \( E \) field is a \((1, 1)\) form, it can only be integrated over 1D regions in space \( (\Gamma_i[n]) \) and 1D regions in time \( (\Gamma_i[n]) \). With this in mind, we define the discretized fields as averages over the appropriate space-time regions.

\[
D_u[i, n] = \frac{1}{4 \Delta x^2} \int_{\Sigma_i[n]} D
\]
\[
B_u[i, n] = \frac{1}{4 \Delta x^2} \int_{\Sigma_i[n]} B
\]
\[
E_u[i, n] = \frac{1}{4 \Delta x \Delta t} \int_{\Gamma_i[n]} \int_{\Sigma_i[n]} E
\]
\[
H_u[i, n] = \frac{1}{4 \Delta x \Delta t} \int_{\Gamma_i[n]} \int_{\Sigma_i[n]} H
\]
\[
J_u[i, n] = \frac{1}{8 \Delta x^2 \Delta t} \int_{\Gamma_i[n]} \int_{\Sigma_i[n]} J
\]
\[
M_u[i, n] = \frac{1}{8 \Delta x^2 \Delta t} \int_{\Gamma_i[n]} \int_{\Sigma_i[n]} M
\]

To aid later discussions, let us denote this discretization process with a “discretization operator” \( \hat{\mathcal{D}} \), which maps a continuous \((n, m)\)-form to a discrete \((n, m)\)-form. So, for example, \( E_u[i, n] = \hat{\mathcal{D}} E \). Note that this discretization process is equivalent to finite volume methods, but because of our use of differential forms, it was trivial to choose integration regions.

2.5. Discretizing the exterior derivative

To discretize Maxwell’s equations, we need to discretize the exterior derivative operators. A key advantage of the DEC approach is that we can define a discrete exterior derivative which is exactly equivalent to the analytical. For example, let us define the discrete time derivative \( \hat{\mathcal{D}}_t \) on a discrete \((k, 0)\)-form \( f[i, n] \) as

\[
\hat{\mathcal{D}}_t f[i, n] = \frac{f[i, n + 1] - f[i, n - 1]}{2 \Delta t}
\]

Using the generalized Stokes’ theorem, it is simple to show that this definition is exactly consistent with

\[
\hat{\mathcal{D}}_t f[i, n] = \frac{1}{2 \Delta t} \int_{\Gamma_i[n]} \mathcal{D}_t f
\]

So, discretizing the field \( f \) and then applying the discrete operator \( \hat{\mathcal{D}}_t \) is exactly equivalent to applying the continuous operator \( \mathcal{D}_t \) and then discretizing the field \( \mathcal{D}_t f \). Or, in terms of the discretization operator, \( \mathcal{D}_t \mathcal{D} \hat{\mathcal{D}}_t = \hat{\mathcal{D}}_t \mathcal{D}_t \). This is an important property which we will explore later.

To write the derivative operators in a more convenient notation, let us define the operators
- \( \hat{T}_u \), which shifts the spatial index \( i_s \to i_u + 1 \), and

- \( \hat{T}_t \), which shifts the temporal index \( n \to n + 1 \).

In this notation, the exterior temporal derivative on a \((k, 0)\)-form \( f[i, n] \) is

\[
\hat{d}_t f[i, n] = \frac{\hat{T}_t - \hat{T}_t^{-1}}{2\Delta t} f[i, n]
\]

(7)

To discretize the spatial exterior derivative \( d_s \), we use Stokes’ theorem to pick definitions which are consistent with \( \hat{d}_s D = D d_s \). The exterior derivative on a discrete \((0, k)\)-form \( f[i, n] \) is

\[
(\hat{d}_s f)_u[i, n] = \frac{\hat{T}_u - \hat{T}_u^{-1}}{2\Delta x} f[i, n]
\]

(8)

The exterior derivative on a discrete \((1, k)\)-form \( f_u[i, n] \) is

\[
(\hat{d}_s f)_u[i, n] = \sum_{v,w=1}^{3} \epsilon_{uvw} \frac{\hat{T}_v - \hat{T}_v^{-1}}{2\Delta x} f_w[i, n]
\]

(9)

where \( \epsilon_{uvw} \) is the Levi-Civita symbol. Finally, the exterior derivative on a discrete \((2, k)\)-form \( f_u[i, n] \) is

\[
(\hat{d}_s f)[i, n] = \sum_{u=1}^{3} \frac{\hat{T}_u - \hat{T}_u^{-1}}{2\Delta x} f_u[i, n]
\]

(10)

These, of course, correspond to the discrete gradient, curl, and divergence respectively.

With these definitions, we can write the discrete Maxwell equations as simply

\[
\hat{d}_t B[i, n] = -\hat{d}_s E[i, n] - M[i, n]
\]

(11)

\[
\hat{d}_t D[i, n] = +\hat{d}_s H[i, n] - J[i, n]
\]

(12)

Comparing with the definitions of \( \hat{d}_t \) and \( \hat{d}_s \) above, we see that these are exactly equivalent to the standard Yee update equations. So, we have shown that DEC with a cubic discretization of space-time leads to the well-known Yee method. (This was shown in a similar manner by [5].)

2.6. Discretizing the constitutive relations

By discretizing the exterior derivatives, we have successfully discretized Maxwell’s equations and arrived at the Yee update equations. However, we are not finished: to discretize the constitutive relations (3), we need to discretize the Hodge dual operators \( \star \) and \( \star_s \). As it turns out, this is the primary challenge that needs to be solved when developing a numerical method with DEC [12,15,5]. Specifically, this is because the discrete Hodge dual operation is the most significant source of error in DEC methods [5,14].

That the discrete Hodge dual is the main source of error is somewhat surprising, considering that the Hodge dual on continuous fields is usually quite trivial. Take the \( E \) field, for example. Though \( \star_s E \) and \( E \) are different mathematical objects, their components have exactly the same numerical value. That is

\[
(\star_s E)_u(x, t) = E_u(x, t)
\]

(13)

Arguably, this is why vector calculus is so successful at hiding the existence of the Hodge dual (at least in flat space-time). For continuous fields, the Hodge dual is usually trivial enough that it can be ignored without consequence.

Discrete fields are a different story, however. Although continuous fields \( E \) and \( \star_s E \) have identical components, that feature is destroyed by the discretization process. Roughly speaking, \( D E \) involves averages of \( E \) over lines, while \( \hat{D}(\star_s E) \) involves averages of \( \star_s E \) over surfaces, and these averages need not be equal in general. This means that the discretized fields \( D E \) and \( \hat{D}(\star_s E) \) will have different numerical values in general! Thus, if we try to use a simple definition of the discrete Hodge dual, like

\[
(\star_s E)_u[i, n] = E_u[i, n]
\]

(14)

we will introduce error.

Mathematically, the root of the problem is that this definition does not satisfy \( \star_s \hat{D} = \hat{D} \star_s \). (This can be easily shown using the definition of \( \hat{D} \).) Unfortunately, defining a discrete Hodge dual which does satisfy that relationship would lead to complicated numerical methods (if it is even possible at all). As a result, the Hodge dual becomes the primary source of error (see Section 2.7), and the order of the numerical method is controlled by the order of the Hodge dual operator. In Section 3 we will present a generalized method for defining Hodge stars of a given order.
2.7. Where does error come from?

In [14], it was shown that for geometrical methods like DEC, the primary source of error is the constitutive relations rather than the derivative approximations. We will now explore this idea further to justify our claim that the Hodge dual should be the focus of more advanced methods. In addition, we will demonstrate why the discrete Hodge duals should ideally satisfy $\hat{*} \hat{*} \hat{D} = \hat{D} * \hat{*}$.

We begin with a basic process for solving Maxwell’s equations:

1. Discretize the initial continuous fields using $\hat{D}$.
2. Advance $B_a[i, n]$ in time using $\hat{d}_t B_a[i, n] = -\hat{d}_t E_a[i, n] - M_a[i, n]$.
3. Calculate $H_a[i, n]$ using $\hat{d}_t B_a[i, n] = \hat{*} \hat{*} \hat{D} H_a[i, n]$.
4. Advance $D_a[i, n]$ in time using $\hat{d}_t D_a[i, n] = +\hat{d}_t H_a[i, n] - J_a[i, n]$.
5. Calculate $E_a[i, n]$ using $D_a[i, n] = \hat{*} \hat{*} \hat{D} E_a[i, n]$.
6. Repeat from step 2 until the desired final time is reached.
7. Recover the final continuous fields using $\hat{D}^{-1}$.

An inevitable source of error in this process is the inverse discretization operator $\hat{D}^{-1}$. Applying $\hat{D}$ to a continuous field discards information about that field, and it is impossible to recover this information exactly. However, this is a one-time-only error: it appears once at the end, but does not accumulate with every time step.

Steps 2 and 4 do not introduce any error into the calculation. This is because, as discussed above, we can prove that $\hat{d}_t \hat{D} = \hat{D} \hat{d}_t$ and $\hat{d}_t \hat{D} = \hat{D} \hat{d}_t$. As a result, the equation we use to advance the $B$ field by one time-step is equivalent to

$$\hat{D}(\hat{d}_t B) = \hat{D}(-\hat{d}_t E - M)$$

(15)

So, discretizing and then advancing the fields using the discretized Ampere’s law is the same as advancing the continuous fields using Ampere’s law and then discretizing them. The only error comes from the original discretization error.

Unfortunately, the same is not true for the constitutive relations (steps 3 and 5). These steps do introduce error because $\hat{*} \hat{*} \hat{D} \neq \hat{D} * \hat{*}$. If we discretize the fields and then apply the discrete constitutive relations, we will obtain a different result than if we apply the continuous constitutive relations and then discretize the resulting fields. The constitutive relations introduce an extra error which accumulates at every time step.

It is interesting to note that the situation is somewhat reversed for finite difference methods: the exterior derivatives introduce error, but not the Hodge duals. This has to do with the discretization operator $\hat{D}$. In DEC, $\hat{D}$ essentially takes averages of the fields over the appropriate space-time surfaces. In finite difference methods, on the other hand, it samples the fields at a point. Under this definition of $\hat{D}$, we have $\hat{*} \hat{*} \hat{D} = \hat{D} * \hat{*}$, which means that the constitutive relations no longer introduce error. On the other hand, we also have $\hat{d}_t \hat{D} \neq \hat{D} \hat{d}_t$ and $\hat{d}_t \hat{D} \neq \hat{D} \hat{d}_t$, which means that the exterior derivative equations now do introduce error.

In light of this, what is the advantage of DEC methods over finite difference methods? One answer is that DEC methods automatically satisfy a number of conservation properties. First, note that the discrete exterior derivatives defined in Section 2.5 satisfy $\hat{d}_t \hat{d}_t = \hat{d}_t \hat{d}_t$ and $\hat{d}_t^2 = 0$, just like their continuous analogs. As a result, in a source-free environment we have

$$\hat{d}_t \left( \hat{d}_t B_a[i, n] \right) = 0$$

(16)

$$\hat{d}_t \left( \hat{d}_t D_a[i, n] \right) = 0$$

(17)

So the discrete divergences $\hat{d}_t B_a[i, n]$ and $\hat{d}_t D_a[i, n]$ are automatically conserved in DEC methods. This is desirable, since these quantities are conserved in the continuous case.

More importantly, these conservation rules (and others [5]) depend only on our definition of $\hat{d}_t$ and $\hat{d}_t$, and not on our definitions of $\hat{*}$ and $\hat{*}$. This is important because in DEC methods the definitions of $\hat{d}_t$ and $\hat{d}_t$ never need to change. For example, to create higher order methods we only change the definitions of $\hat{*}$ and $\hat{*}$, and so we can be sure that the conservation rules will still be satisfied. In finite difference methods, on the other hand, we deal with more challenging problems by altering $\hat{d}_t$ and $\hat{d}_t$, which has the potential to destroy conservation properties.

So DEC methods are, in some sense, “safer” with respect to conservation. Later, we will derive a method for coupling overlapping grids with a higher-order Hodge dual. Because we only alter the Hodge dual and not the derivatives, we can be confident that our new methods will still obey the desirable conservation properties of simpler methods. This can also be extended to a wide variety of other applications. For example, we could extend the method outlined in this paper to derive a new Hodge dual to deal with a sharp dielectric interface. Again, just by using the DEC framework we ensure that conservation properties are not broken. This would be difficult to guarantee when deriving a new FDTD derivative approximation for a sharp interface.
Fig. 1. The $\Sigma_1[i]$ surfaces for the standard Yee grid. For this particular choice of Yee grid, $\Sigma_1[i]$ only exists for $i_1$ odd and $i_2, i_3$ even.

Fig. 2. The $\Sigma_1[i]$ surfaces for the Lebedev collocated grid (four overlapping Yee grids). For this particular choice of Lebedev grid, $\Sigma_1[i]$ only exists when $(i_1 + i_2 + i_3)$ is odd.

2.8. Coupling overlapping grids

Before moving on to higher-order methods, let us briefly discuss why we are interested in higher-order methods in the first place. By using the general $[i, n]$ notation, we have somewhat distanced ourselves from the usual Yee mesh. This is desirable, of course, since our end goal is to generate methods with collocated fields. However, we cannot forget the Yee grid entirely. This is because our operators $\hat{d}_i$ and $\hat{d}_r$ couple points $[i, n]$ in a Yee grid structure. This implies two things.

1. We cannot pick an arbitrary grid. If we use a point $[i, n]$, then we are forced to use the Yee grid that includes that point.
2. Fields on different Yee grids will not be coupled to each other by $\hat{d}_i$ or $\hat{d}_r$. They can only be coupled by the discrete Hodge dual operators.

This second point is a known weakness of collocated methods such as the Lebedev grid [4]. With a simple, second-order Hodge dual, the four Yee grids which make up the Lebedev grid are not coupled together (at least in isotropic media). This essentially means that three redundant Yee simulations are being run without providing any extra accuracy. Our hope in the remaining sections is to “fix” collocated methods like the Lebedev method by coupling the four Yee grids with a more sophisticated Hodge dual.

The overlapping grids are visualized in Figs. 1, 2, and 3. Shown are the $\Sigma_1[i]$ surfaces, which would correspond to the $x_1$ component of some 2-form field (e.g., $B$ or $D$). Essentially, adjacent surfaces are coupled in a second order method, but overlapping surfaces are not. To couple the overlapping surfaces, we need to move to a higher-order Hodge dual.

3. Higher order methods

In the last section, we outlined the discrete exterior calculus (DEC) approach to solving Maxwell’s equations. In particular, we saw the key role played by the discrete Hodge dual operators, $\hat{\ast}_1$ and $\hat{\ast}_r$. We saw that these are the operators which primarily produce error in a numerical method, and they are the operators that we need to redefine if we are to produce higher-order methods. We will now develop a general approach for deriving higher-order Hodge stars, including some specific examples.
3.1. Practical numerical methods

The 7 step solution method outlined in the Section 2.7 is useful because it is transparent and easy to understand. However, it is not very efficient from an implementation standpoint, since it requires us to store all four fields, and perform four sets of calculations every time we advance the fields. We also did not account for conductive media, which any good method should do. Fortunately, we can come up with a much better implementation by just plugging the constitutive relations into Maxwell’s equations. The result is that we solve Maxwell’s equations as follows.

1. Discretize the initial continuous fields using $\bar{D}$.
2. Advance $H_u[i, n]$ in time using
   \[ \left( \hat{d}_i \hat{s}_j \mu + \hat{s}_i \sigma^* \right) H_u[i, n] = -\hat{d}_s E_u[i, n] \] (18)
3. Advance $E_u[i, n]$ in time using
   \[ \left( \hat{d}_i \hat{s}_j \varepsilon + \hat{s}_i \sigma^\ast \right) E_u[i, n] = +\hat{d}_s H_u[i, n] \] (19)
4. Repeat from step 2 until the desired final time is reached.
5. Recover the final continuous fields using $\bar{D}^{-1}$.

Now, there are a number of ways to go about discretizing the Hodge duals. The simplest, which we will focus on in this paper, is to move the spatial Hodge dual onto the right hand side of the update equations, yielding

\[ \left( \hat{d}_i \hat{s}_j \mu + \sigma^* \right) H_u[i, n] = -\hat{s}_s \hat{d}_s E_u[i, n] \] (20a)
\[ \left( \hat{d}_i \hat{s}_j \mu + \sigma^* \right) E_u[i, n] = +\hat{s}_s \hat{d}_s H_u[i, n] \] (20b)

Our goal, then, is to find

- a discrete $\hat{s}$ which turns a discrete $(2, 1)$-form into a discrete $(1, 1)$-form, and
- a discrete $\hat{t}$ which turns a discrete $(1, 1)$-form into a discrete $(1, 0)$-form.

Not surprisingly, the orders of accuracy of $\hat{s}$ and $\hat{t}$ will determine the orders of the resulting numerical method in space and time, respectively.

A potential downside of this approach is that it is not very “space-time friendly.” To develop a method which is compatible with a 4D space-time approach, we would need to discretize the entire operator $\hat{s} \hat{t}$, rather than discretizing $\hat{s}$ and $\hat{t}$ separately. Though it is not shown here, this space-time approach does not allow for an explicit method which is both 4th-order in space and 2nd-order in time. This is not surprising, since a space-time friendly method should treat space and time on the same footing, and we would expect it to be 4th order in both space and time.

3.2. Discrete temporal Hodge dual

We begin by deriving higher-order version of the discrete time Hodge dual. Though we are actually more interested in the spatial Hodge dual, it is helpful to see the approach in a simpler 1D setting. Here we extend upon the ideas of basis expansion outlined in [14].
Our goal, specifically, is to define the operator \( \hat{\star} \), which takes a discrete 1-form (in time), \( f[n] \) into a discrete 0-form (in time), \( \hat{\star} f[n] \). (To avoid unnecessary clutter, we drop any references to 3D space in this section, and just focus on time.) To begin with, we insist that \( \hat{\star} \) must take the form of a linear combination over neighbouring time steps. That is,

\[
(\hat{\star} f)[n] = \sum_p c_p \frac{\hat{t}^p}{t} f[n]
\]  

Which neighbours \( p \) are used will be a key factor in the accuracy of the Hodge dual. Roughly, the more neighbours we use, the more accurate the method. Of course, this comes at a cost of increased computational complexity.

Now, once we've selected a set of neighbours, \( p \), how do we determine what the \( c_p \) coefficients should be? The answer is, we want to get as close as possible to satisfying the condition

\[
(\hat{\star} \hat{\Delta} f)[n] = (\hat{\Delta} \hat{\star} f)[n]
\]  

for any \( f \). Or, writing out \( \hat{\star} \) and \( \hat{\Delta} \) more explicitly, our condition is

\[
\sum_p c_p \frac{1}{2\Delta t} \int_{(p-n+1)\Delta t}^{(p-n-1)\Delta t} f(t) = \hat{\star} f(n\Delta t)
\]  

We will not be able to meet this condition exactly, but let us expand \( f(t) \) about \( t = n\Delta t \) using a linear combination of basis 1-forms \( \phi^{(m)}(t)dt \). We will try to meet the condition for as many of these basis forms as we can. We write

\[
f(n\Delta t + t) = \sum_m a_m \phi^{(m)}(t) dt
\]  

which implies

\[
\hat{\star} f(n\Delta t + t) = \sum_m a_m \phi^{(m)}(t)
\]  

Then we can write our condition as

\[
\sum_m a_m \sum_p c_p \frac{1}{2\Delta t} \int_{(p-n+1)\Delta t}^{(p-n-1)\Delta t} \phi^{(m)}(t)dt = \sum_m a_m \phi^{(m)}(0)
\]  

We want this equation to hold for any combination of coefficients \( a_m \) (i.e., any function \( f(t) \)). So we can equate each term of the sum individually, giving us one equation for each \( m \).

\[
\sum_p c_p \frac{p+1}{2} \int_{p-1}^{p+1} \phi^{(m)}(\gamma \Delta t) d\gamma = \phi^{(m)}(0)
\]  

The dimensionless integration variable \( \gamma = \frac{t}{\Delta t} \) has been introduced here for simplicity.

This is a general result which works for any desired basis \( \phi^{(m)}dt \). But, suppose we use a polynomial basis, where

\[
\phi^{(m)}(t) = \left( \frac{t}{\Delta t} \right)^m
\]  

Then

\[
\phi^{(m)}(0) = \delta_{m,0}
\]  

where \( \delta_{m,m'} \) is the Kronecker delta. We also have

\[
\int_{p-1}^{p+1} \phi^{(m)}(\gamma \Delta t) d\gamma = \frac{(p+1)^{m+1} - (p-1)^{m-1}}{m+1}
\]  

So our conditions become

\[
\sum_p c_p \left( \frac{(p+1)^{m+1} - (p-1)^{m-1}}{2(m+1)} \right) = \delta_{m,0}
\]
In particular, the equations up to 4th order \((m = 0, 1, 2, 3)\) are
\[
\sum_p c_p = 1 \tag{32}
\]
\[
\sum_p c_p p = 0 \tag{33}
\]
\[
\sum_p c_p \left(p^2 + \frac{1}{3}\right) = 0 \tag{34}
\]
\[
\sum_p c_p \left(p^3 + p\right) = 0 \tag{35}
\]

Now, if we select a set of neighbours \(p\) to use, we can finally determine the discrete Hodge coefficients \(c_p\). Suppose first that we don’t use any neighbouring points at all. That is, we will only use \(p = 0\). Then we cannot solve all four equations, but we can solve the first two using \(c_0 = 1\). And so we obtain a simple discrete Hodge dual which is second-order accurate:
\[
\hat{*}t f[n] = f[n] \tag{36}
\]
This is exactly what we might have have naively guessed based on the continuous Hodge dual. So we see that the assumption of a trivial (identity) Hodge dual is actually only second-order accurate.

Now, let us try introducing neighbours. A simple choice is to introduce the field values one grid index forward and backward in time. That is, we use \(p = -1, 0, 1\). Then, because of the symmetry of the equations, we can solve the first- and third-order equations by setting \(c_{-1} = c_1\). The remaining equations are
\[
c_0 + 2c_1 = 1 \tag{37}
\]
\[
\frac{1}{3}c_0 + \frac{8}{3}c_1 = 0 \tag{38}
\]
Solving these, we obtain the coefficients for a fourth-order Hodge dual
\[
c_0 = \frac{4}{3} \tag{39}
\]
\[
c_{-1} = c_1 = \frac{-1}{6} \tag{40}
\]
Or, put another way,
\[
\hat{*}t f[n] = \frac{-f[n+1] + 8f[n] - f[n-1]}{6} \tag{41}
\]
This is an interesting expression: it looks somewhat like a weighted average, except that some of the weights are negative. The reason for this becomes apparent if we think about what the \(\hat{*}t\) operator is doing conceptually. The discrete Hodge dual tries to take the discrete 1-form \(\check{D}f\) into the discrete 0-form \(\check{D} \hat{*} \check{D} f\). But \(\check{D}f\) is essentially a time-averaged version of \(f\), while \(\check{D} \hat{*} \check{D} f\) is a sampled version of the same function. Thus, our discrete Hodge dual effectively needs to undo a time-averaging process. In light of this, the appearance of negative coefficient is not surprising.

Now, the above fourth-order Hodge dual is fine, but it will not work if we want to use a “leap-frog” time-stepping procedure, like we would in the standard Yee method. This is because in a leapfrogging method, the neighbours \(p = \pm 1\) do not actually exist. For example, \(E_y[i, n]\) might only be stored on even \(n\) while \(H\) is stored on odd \(n\). Thus, if we want to use nearest neighbours with a leap-frog time update structure, we must use \(p = 2, 0, 2\). Solving for the Hodge dual coefficients \(c_p\) in a similar manner to the last case, we obtain an alternative Hodge dual.
\[
\hat{*}t f[n] = \frac{-f[n+2] + 26f[n] - f[n-2]}{24} \tag{42}
\]
Like the last one, this Hodge dual is fourth-order accurate, though it uses different neighbours.

We would now like to make sure that the above results make sense. We can do this by looking at the operator \(\hat{d}_t \hat{*}_t\), where we recall that
\[
\hat{d}_t f[n] = \frac{f[n+1] - f[n-1]}{2\Delta t} \tag{43}
\]
Thus, with our second-order Hodge dual (using \(p = 0\)), we have
\[
\hat{d}_t \hat{*}_t f[n] = \frac{f[n+1] - f[n-1]}{2\Delta t} \tag{44}
\]
With our first fourth-order Hodge dual (using \( p = -1, 0, 1 \), we have
\[
\hat{d}_t \hat{\star} f[n] = \frac{-f[n + 2] + 8f[n + 1] - 8f[n - 1] + f[n - 2]}{12 \Delta t}
\] (45)

And, with our second fourth-order Hodge dual (using \( p = -2, 0, 2 \), we have
\[
\hat{d}_t \hat{\star} f[n] = \frac{-f[n + 3] + 27f[n + 1] - 27f[n - 1] + f[n - 3]}{48 \Delta t}
\] (46)

These expressions are all well-known second- and fourth-order finite-difference approximations for the time derivative. So, reassuringly, we see that DEC is able to reproduce higher-order 1D difference approximations. Our approach is somewhat advantageous though, because we only had to change the definition of \( \hat{\star} \), and not the definition of \( \hat{d}_t \). In some sense, we have separated out a portion of the time derivative which never changes, regardless of the order of accuracy.

3.3. Discrete spatial Hodge dual

We will now discretize the discrete spatial Hodge dual operator, \( \hat{\star} \). Fortunately, the process is similar to the discretization of \( \hat{\star} \) in the last section. Here, we seek an operator \( \hat{\star} \), which takes a discrete 2-form (in space), \( f_u[i] \) into a discrete 1-form (in space), \( (\hat{\star} f_u)[i] \).

As with the temporal Hodge dual, we insist that \( \hat{\star} \) take the form of a linear operator. However, let us generalize a little bit:
\[
\sum_q b_{q,u} \hat{\tau}^q (\hat{\star} f)_u[i] = \sum_p c_{p,u} \hat{\tau}^p f_u[i]
\] (47)

where \( q = (q_1, q_2, q_3) \), \( p = (p_1, p_2, p_3) \), and \( \hat{\tau}^q \) is short for \( \hat{T}^{q_1}_1 \hat{T}^{q_2}_2 \hat{T}^{q_3}_3 \). This generalization is interesting, because it allows us to see how we can generate either explicit or implicit numerical methods. If \( b_q \) is non-zero for any \( q \neq 0 \), then our equation for \( (\hat{\star} f)_u[i] \) is implicit, requiring us to solve a system of equations. In terms of solving Maxwell’s equations, this would mean that we would need to solve a system of equations at every time step. Implicit methods might be an interesting avenue to explore, but we will not discuss them further here, because of the additional computational complexity that they introduce. So, we will go back to the stricter form, which ensures that we end up with an explicit method:
\[
(\hat{\star} f_u[i] = \sum_p c_{p,u} \hat{\tau}^p f_u[i]
\] (48)

As with \( \hat{\star} \), our goal in defining \( \hat{\star} \) will be to satisfy
\[
(\hat{\star} \hat{\mathcal{D}} f)_u[i] = (\hat{\mathcal{D}} \hat{\star} f)_u[i]
\] (49)

We will use a similar notion of basis expansion, but we will use a basis of two-forms \( \phi_u^{(m)} dx_v \wedge dx_w \):
\[
f(n\Delta t + t) = \sum_m a_m \sum_{u=1}^3 \phi_u^{(m)} dx_v \wedge dx_w
\] (50)

where \( v, w \) are always chosen so that \( \{u, v, w\} \) is an even permutation of \( \{1, 2, 3\} \). (That is, \( \{u, v, w\} \) is either \( \{1, 2, 3\} \), \( \{2, 3, 1\} \), or \( \{3, 1, 2\} \).)

Using a similar approach to the temporal Hodge dual, we find that the equations we must solve are
\[
\sum_p \frac{c_{p,u}}{2\Delta x} \int_{\Sigma_u[i]} \phi_u^{(m)}(x)dx_v \wedge dx_w = \int_{\Gamma_u[i]} \phi_u^{(m)}(x)dx_u
\] (51)

where \( \Sigma_u[i] \) and \( \Gamma_u[i] \) are the same surfaces and lines defined in Section 2.3.

All that remains is to select a basis \( \phi_u^{(m)} \), select neighbours \( p \), and solve for \( c_p \). By letting \( m = (m_1, m_2, m_3) \), we can take \( \phi_u^{(m)} \) to be a polynomial basis
\[
\phi_u^{(m)}(x) = \left( \frac{x_1}{\Delta x} \right)^{m_1} \left( \frac{x_2}{\Delta x} \right)^{m_2} \left( \frac{x_3}{\Delta x} \right)^{m_3}
\] (52)

With some work, we can then write our conditions as
\[
\sum_p \frac{c_{p,u}}{p_u} \int_{\Delta x}^{p_u+1} \int_{\Delta x}^{p_u+1} \gamma_v^{m_v} \gamma_w^{m_w} \gamma_u^{m_u} dy_v dy_w = \delta_{m_v,0} \delta_{m_w,0} \int_{\Delta x}^{p_u+1} \gamma_u^{m_u} dy_u
\] (53)
Again, \( u = 1, 2, 3 \) and \( v, w \) are always chosen so that \( \{u, v, w\} \) is an even permutation of \( \{1, 2, 3\} \). The dimensionless integration variables \( y_u = x_u / \Delta x \) have been introduced for simplicity. To simplify further, we define

\[
\xi(p_u, m_u) = \int_{p_u - 1}^{p_u + 1} y_u^{m_u} dy_u = \frac{(p_u + 1)^{m_u+1} - (p_u - 1)^{m_u+1}}{m_u + 1}
\]

(54)

Then the equations we must solve are

\[
\sum_p c_{p,u} \frac{p_u^{m_u}}{2} \xi(p_v, m_v) \xi(p_w, m_w) = \delta_{m_v,0} \delta_{m_w,0} \xi(p_u, m_u)
\]

(55)

So, we now just need to pick neighbours \( p = (p_1, p_2, p_3) \) and solve for the coefficients \( c_{p,u} \). Fortunately, there is an important symmetry here which allows us to quickly calculate many of the \( c_{p,u} \) coefficients. For this to work, we will need to use symmetric sets of \( p \) values. Specifically, suppose we use some \( p \) neighbour when calculating the \( u \) component of the field. Then we also need to use every neighbour which can be generated by a combination of 90-degree rotations about the \( u \) axis and reflections along the \( u \) axis. Because of the symmetry, all the coefficients \( c_{p,u} \) of symmetric sets like this will be exactly equal. Furthermore, by using symmetric sets of \( p \) points, we automatically solve each equation (55) for which any of \((m_1, m_2, m_3)\) is odd. So symmetry simplifies our analysis considerably.

3.3.1. Second-order Yee

Now, let us start with the simplest case, where the only neighbour is \( p = (0, 0, 0) \). Our zeroth-order equation \( m = (0, 0, 0) \) gives us

\[
c_{(0,0,0),u} = 1
\]

(56)

So our discrete Hodge dual operator is just the identity operator.

\[
(\star_s f)_u[i] = f_u[i]
\]

(57)

This is, in fact, second-order accurate because of the symmetry discussion above. We essentially solve the \( m = (1, 0, 0), (0, 1, 0), (0, 0, 1) \) equations for free. This second-order spatial Hodge dual, combined with a second-order temporal Hodge dual, gives us the second-order Yee method.

3.3.2. Neighbour groups

It will be convenient at this point to define certain symmetric sets of neighbour points. Suppose we want to calculate \((\star_s f)_u[i] \) (i.e., the \( u \) component). Then define

- The 0 neighbour group as \( p_u, p_v, p_w = (0, 0, 0) \).
- The 1A neighbour group as \( p_u, p_v, p_w = (\pm 1, 0, 0) \).
- The 1B neighbour group as \( p_u, p_v, p_w = (0, \pm 1, 0) \).
- The 1A neighbour group as \( p_u, p_v, p_w = (0, \pm 1, \pm 1) \).
- The 1B neighbour group as \( p_u, p_v, p_w = (\pm 1, \pm 1, 0) \).
- The 2A neighbour group as \( p_u, p_v, p_w = (\pm 2, 0, 0) \).
- The 2B neighbour group as \( p_u, p_v, p_w = (0, \pm 2, 0) \).

A selection of these are visualized in Fig. 4. Again, \( u = 1, 2, 3 \) and \( v, w \) are always chosen so that \( \{u, v, w\} \) is an even permutation of \( \{1, 2, 3\} \). Each of these groups satisfies the symmetry requirements discussed above, so we know that all members in a given group will be multiplied by the same coefficient. By using different combinations of neighbour groups, we will be able to generate a number of higher-order methods.

3.3.3. Fourth-order Yee

Let us start with a fourth-order Yee grid. On the Yee grid, we are restricted in which neighbours we can use. For example, \( E_u \) might only be defined when \( i_u \) is even and \( i_v, i_w \) are odd. Of the neighbour groups defined above, the only ones we can use on the Yee grid are the 0, 2A and 2B groups. Because of symmetry, we only need to solve the equations (55) when all of \( m_u, m_v, m_w \) are even. The equation when \( m_u = m_v = m_w = 0 \) is

\[
c_{0,u} + 2c_{2A,u} + 4c_{2B,u} = 1
\]

(58)

The equation when \( m_u = 2 \) and \( m_v = m_w = 0 \) is

\[
24c_{2A,u} = 1
\]

(59)

And the equation when either \( m_v \) or \( m_w \) is 1 and the other two are 0 is
Fig. 4. Visualization of neighbour groups 0, 1A, 1B, 11A, and 11B for the u direction.

\[ c_{0,u} + 2c_{2A,u} + 28c_{2B,u} = 0 \]  
(60)

Solving these, we obtain

\[ c_{0,u} = \frac{26}{24} \]  
(61)
\[ c_{1A,u} = \frac{1}{24} \]  
(62)
\[ c_{1B,u} = -\frac{1}{24} \]  
(63)

Which means that our fourth order Yee Hodge dual is

\[ (\mathbf{\star}_s f)_u[i] = \frac{1}{24} \left[ 26 + \hat{T}_{u}^+ + \hat{T}_{u}^2 - \hat{T}_{v}^+ - \hat{T}_{v}^2 - \hat{T}_{w}^+ - \hat{T}_{w}^2 \right] f_u[i] \]  
(64)

### 3.3.4. Fourth-order Lebedev

As mentioned previously, the Lebedev method outlined in [4] is equivalent to four overlapping Yee grids. The exterior derivative operators and the second-order Hodge duals do not couple these grids together. However, we can couple them together using a fourth-order Hodge dual operator. On one version of the Lebedev grid, all three components of the \( E \) field are stored whenever \( i_1 + i_2 + i_3 \) is even, and all three components of the \( H \) field are stored whenever \( i_1 + i_2 + i_3 \) is odd. As a result, the “nearest neighbours” on the Lebedev grid are given by neighbour groups 11A and 11B. (There are 12 total neighbours: each at a distance of \( \sqrt{2}\Delta x \).) So, in calculating the fourth-order Lebedev Hodge dual, we use neighbour groups 0, 11A, and 11B. Solving equations (55) up to fourth-order is tedious, but not exceedingly hard because of symmetry. The resulting solution for the coefficients is

\[ c_{0,u} = \frac{28}{24} \]  
(65)
\[ c_{11A,u} = \frac{3}{24} \]  
(66)
\[ c_{11B,u} = \frac{1}{24} \]  
(67)

So the resulting Hodge dual operation is

\[ (\mathbf{\star}_s f)_u[i] = \frac{1}{24} \left[ 28f_u[i] - 3 \sum_{p \in 11A} \hat{T}_p f_u[i] + \sum_{p \in 11B} \hat{T}_p f_u[i] \right] \]  
(68)

Though more challenging to implement, this Hodge dual couples together the four Yee grids which make up the Lebedev grid. Thus, there should no longer be a four-fold waste of computational power in simple cases like homogeneous isotropic media.

### 3.3.5. Fourth-order cubic

Another collocated grid we can look at is a simple cubic grid, on which every component of both \( E \) and \( H \) is stored at every location \( i = (i_1, i_2, i_3) \). Though seemingly simple, this is actually equivalent to 8 overlapping Yee grids. As such, if we use a second-order Hodge dual, these 8 grids will not be coupled to each other by our update equations. So we would expect a second-order collocated cubic method to fail quite miserably (at very least in isotropic, homogeneous media). But, again, we can try to “fix” this method using a higher-order Hodge dual. Specifically, we will use neighbour groups 0, 1A, and 1B, to ensure that the eight Yee grids are coupled. Applying symmetry and solving (55) up to fourth order, we arrive at
\[
c_0 = \frac{8}{6}
\]
(69)
\[
c_{1A,u} = \frac{1}{6}
\]
(70)
\[
c_{1B,u} = -\frac{1}{6}
\]
(71)

so that our fourth-order Hodge dual operation is
\[
(\hat{\ast} f)_u[i] = \frac{1}{6} \left[ 8 f_u[i] + \sum_{p \in 1A} \hat{T}^p f_u[i] - \sum_{p \in 1B} \hat{T}^p f_u[i] \right] \tag{72}
\]

3.3.6. Pseudo-fourth-order methods

In the last two sections, we coupled overlapping Yee grids by moving to fourth-order methods. In all cases, this required the introduction of two neighbour groups. What happens if we only introduce one neighbour group? We will not be able to achieve fourth-order accuracy, but we will have a simpler Hodge dual operator, and we will still couple together the overlapping grids.

Finding the coefficients for these methods is quite similar to the cases above: we pick a set of neighbour points to use, and then we solve as many of the equations (55) as we can. In these cases, we will be able to solve some of the equations required for fourth-order accuracy, but not all of them. We will simply state the results.

For a pseudo-fourth-order Yee method, we only use neighbour groups 0 and 2B. The resulting Hodge dual is
\[
(\hat{\ast} f)_u[i] = \frac{1}{24} \left[ 28 f_u[i] - \sum_{p \in 2B} \hat{T}^p f_u[i] \right] \tag{73}
\]

For a pseudo-fourth-order Lebedev method, we only use neighbour groups 0 and 11B. The resulting Hodge dual is
\[
(\hat{\ast} f)_u[i] = \frac{1}{12} \left[ 16 f_u[i] - \sum_{p \in 11B} \hat{T}^p f_u[i] \right] \tag{74}
\]

There are other pseudo-fourth-order methods we could calculate, but initial testing showed them to have poorer performance and so they were discarded for simplicity.

4. Dispersion analysis

We have now outlined methods for generating higher-order Hodge duals, and we have presented a number of different methods on Yee, Lebedev, and cubic grids. To compare them, we will now examine their dispersion characteristics.

4.1. Dispersion relation derivation

For this section, we will assume sourceless media in which \( \mu = \varepsilon = 1 \). Then our update equations are simply
\[
\hat{\partial}_t \hat{\ast}_t \hat{H}_u[i, n] = -\hat{\ast}_s \hat{\partial}_s E_u[i, n] \tag{75}
\]
\[
\hat{\partial}_t \hat{\ast}_t E_u[i, n] = +\hat{\ast}_s \hat{\partial}_s \hat{H}_u[i, n] \tag{76}
\]

We can combine these to obtain
\[
\hat{\partial}_t \hat{\ast}_t \hat{\partial}_t \hat{\ast}_t E_u[i, n] = -\hat{\ast}_s \hat{\partial}_s \hat{\ast}_s \hat{\partial}_s E_u[i, n] \tag{77}
\]

Now, let us assume that the continuous field \( E \) takes the form of a plane wave
\[
E_u[i, n] = \hat{E}_u \exp \left\{ j ( q \cdot i + \Omega n ) \right\} \tag{78}
\]
where \( q \in \mathbb{R}^3, \Omega \in \mathbb{R} \), and \( q \cdot i = q_1 i_1 + q_2 i_2 + q_3 i_3 \). It is possible to show that the process of discretizing a continuous plane wave with \( D \) actually only affects the amplitude, and not the wave vector or phase. As a result, this discrete plane wave corresponds to a continuous field which is a plane wave with wave vector \( k = q/\Delta x \) and angular frequency \( \omega = \Omega/\Delta t \). The advantage of assuming plane waves is that plane waves are eigenstates of the translation operators
\[
\hat{T}_v^p E_u[i, n] = e^{jqv} E_u[i, n] \tag{79}
\]
\[
\hat{T}_t^p E_u[i, n] = e^{j\Omega v} E_u[i, n] \tag{80}
\]
Since all of the operators we have defined so far are linear combinations of the translation operators, we expect that these discrete plane waves will be eigenstates of the update equations. Our goal will be to find the conditions on $\Omega$ and $q$ which are necessary to make $E_u[i, n]$ an eigenstate.

First, let us examine our discrete Hodge dual operator under the plane wave assumption:

$$\hat{s}_t E_u[i, n] = \sum_p c_{p,u} e^{i q_p} E_u[i, n]$$  \hspace{1cm} (81)

Because of the symmetry of the neighbours used, this will always simplify into an expression involving cosines. For example, for the fourth-order Lebedev method, we have

$$\hat{s}_t E_u[i, n] = \left( c_0 + 4c_{\perp 1A} \cos(q_v) \cos(q_w) \\
+ 4c_{\perp 1B} \cos(q_v) \cos(q_w) + 4c_{\perp 1B} \cos(q_u) \cos(q_w) \right) E_u[i, n]$$  \hspace{1cm} (82)

Defining

$$S_u = \frac{\hat{s}_t E_u[i, n]}{E_u[i, n]} = \sum_p c_{p,u} e^{i q_p}$$  \hspace{1cm} (83)

we can write the spatial Hodge dual as a $3 \times 3$ matrix equation

$$\begin{bmatrix} \hat{s}_t E_1[i, n] \\ \hat{s}_t E_2[i, n] \\ \hat{s}_t E_3[i, n] \end{bmatrix} = \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix} \begin{bmatrix} E_1[i, n] \\ E_2[i, n] \\ E_3[i, n] \end{bmatrix}$$  \hspace{1cm} (84)

A similar analysis with the exterior derivative operator shows that

$$\begin{bmatrix} \hat{d}_t E_1[i, n] \\ \hat{d}_t E_2[i, n] \\ \hat{d}_t E_3[i, n] \end{bmatrix} = \begin{bmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{bmatrix} \begin{bmatrix} E_1[i, n] \\ E_2[i, n] \\ E_3[i, n] \end{bmatrix}$$  \hspace{1cm} (85)

where

$$\gamma_u = \frac{\sin(q_u)}{2Ax}$$  \hspace{1cm} (86)

As a result,

$$\begin{bmatrix} \hat{s}_t \hat{d}_t E_1[i, n] \\ \hat{s}_t \hat{d}_t E_2[i, n] \\ \hat{s}_t \hat{d}_t E_3[i, n] \end{bmatrix} = \begin{bmatrix} 0 & -\gamma_3 S_1 & \gamma_2 S_1 \\ \gamma_3 S_2 & 0 & -\gamma_1 S_2 \\ -\gamma_2 S_3 & \gamma_1 S_3 & 0 \end{bmatrix} \begin{bmatrix} E_1[i, n] \\ E_2[i, n] \\ E_3[i, n] \end{bmatrix}$$  \hspace{1cm} (87)

Finally, let us assume that we are using the second-order time Hodge dual, so that $\hat{s}_t$ is just the identity. Under this assumption, we have

$$\hat{d}_t \hat{s}_t E_u[i, n] = \gamma_t E_u[i, n]$$  \hspace{1cm} (88)

where

$$\gamma_t = \frac{\sin(\Omega)}{2Ax}$$  \hspace{1cm} (89)

Now that we have expressions for $\hat{d}_t \hat{s}_t$ and $\hat{s}_t \hat{d}_t$ under the plane wave assumption, we can write the update equation for $E$ as

$$\gamma_t^2 \begin{bmatrix} E_1[i, n] \\ E_2[i, n] \\ E_3[i, n] \end{bmatrix} = \begin{bmatrix} 0 & -\gamma_3 S_1 & \gamma_2 S_1 \\ \gamma_3 S_2 & 0 & -\gamma_1 S_2 \\ -\gamma_2 S_3 & \gamma_1 S_3 & 0 \end{bmatrix} \begin{bmatrix} E_1[i, n] \\ E_2[i, n] \\ E_3[i, n] \end{bmatrix}$$  \hspace{1cm} (90)

This is an eigenvalue equation, for which the eigenvalues are

$$\gamma_t^2 = 0$$  \hspace{1cm} (91)

and
\[ \gamma_1^2 = S_1 S_2 \gamma_2^2 + S_2 S_3 \gamma_1^2 + S_3 S_1 \gamma_2^2 \]  
(92)

These are the dispersion relationships. The first corresponds to DC fields (\( \omega = 0 \)), and the second corresponds to all other frequencies. We will therefore focus on the second.

To write it more explicitly, our dispersion relationship is

\[ \sin^2(\Omega) = \left( \frac{\Delta t}{\Delta x} \right)^2 \left[ \sum_{u,v,w} S_u(q) S_v(q) \sin^2(q_w) \right] \]  
(93)

where the sum is over all \( \{u, v, w\} \) triplets which are even permutations of \( \{1, 2, 3\} \). Also, to restate a definition above,

\[ S_u(q) = \sum_p c_{p,u} e^{i q_p} \]  
(94)

Note that, while we have restricted ourselves to methods which are second-order in time, the extension to fourth-order time methods is not difficult. All that changes is the left hand side of (93) becomes a sum of sine terms rather than a single sine term. For example, the fourth-order leapfrog method leads to \( (27 \sin(\Omega) - \sin(3\Omega))/24 \) on the left hand side rather than \( \sin(\Omega) \).

4.2. Stability

Under the plane wave assumption, \( E_u[i,n] \) is multiplied by \( \sin^2(\Omega) \) at each time step. (It is an eigenstate of the update equation.) Thus, our stability condition is

\[ \sin^2(\Omega) \leq 1 \]  
(95)

which is equivalent to saying that the frequency \( \Omega = \omega \Delta t \) must be real. Using (93), we obtain a stability condition for \( \Delta t \):

\[ \frac{\Delta t}{\Delta x} \leq \left[ \max_{u,v,w} \left\{ S_u(q) S_v(q) \sin^2(q_w) \right\} \right]^{-1/2} \]  
(96)

where the maximum is taken over all values \( q \in \mathbb{R}^3 \).

The maxima in (96) were determined numerically to evaluate the stability criteria for a number of different methods. The results are shown in Table 2. We see that the 2nd order Yee method is actually the best in terms of stability. As we move into higher-order methods, the stability criteria are more restrictive. Of course, this is not necessarily a problem, since higher-order methods are usually more accurate for the same grid spacing.

4.3. Comparison of dispersion characteristics

Using the dispersion characteristics derived above, we will now analyze the following eight methods in detail:

- Second-order Yee
- Pseudo-fourth-order Yee
- Fourth-order Yee
- Non-DEC fourth-order Yee
- Second-order Lebedev
- Pseudo-fourth-order Yee
- Fourth-order Yee
- Fourth-order cubic
A few notes:

- Each method is second-order in time, since we are interested in coupling collocated grids in space.
- The second-order Lebedev is identical to the second-order Yee in isotropic media, but it requires four times as much memory, so it is useful to include as a reference when we do equal-memory comparisons.
- Non-DEC fourth-order Yee refers to the typical fourth-order method which is derived by replacing the derivatives in the curl with fourth-order difference approximations. This fourth-order curl operator cannot be derived using the DEC formalism we have outlined, so it is included for comparison.
- Only one cubic method is included since it was found that the cubic methods perform quite poorly under any fair comparison.

To generate dispersion results, the numerical wave speed $v_p$ was calculated using (93) for a number of wave vectors with different magnitudes and directions. “Mean wave speed error” refers to the average of $|v_p - c|/c$ over all wave-vector directions (average over solid angles, to be more precise). Here $c$ is the actual speed of light. Note the absolute value: some of the methods involve both subluminal and superluminal wave speeds, which means that the average of $(v_p - c)/c$ can be zero, even when there is a significant amount of error. Using $|v_p - c|/c$ ensures a more fair comparison of dispersion error. “Standard deviation of wave speed error” refers to the standard deviation of $(v_p - c)/c$ over all wave-vector directions (solid angles). Here $(v_p - c)/c$ is used rather than $|v_p - c|/c$, since the absolute value tends to reduce the standard deviation for some methods and not others. We are interested in the total variation in error, so $(v_p - c)/c$ is a more reasonable value to use when calculating standard deviation.

The plots shown below were generated using the analytically-derived dispersion formulas, but they were confirmed numerically. Each method was implemented, and a plane wave with wave-vector $k$ was simulated using periodic boundary conditions. Choosing $k$ to be a reciprocal lattice vector, we expect the simulated plane wave to be very sharply-peak in the frequency domain at some frequency $\omega$. Taking advantage of this, the wave speed was then calculated as $\omega/|k|$. In each case, the wave speed of the simulated plane wave was found to match the analytically-derived results to arbitrary precision. In addition, these plane waves were indeed very sharply peaked in the frequency domain, confirming that they are eigenstates of the update equations.

### 4.3.1. Equal step comparison

As a baseline, let us examine the dispersion error when all methods use the same $\Delta x$ and $\Delta t$. We pick $\Delta t$ to be the biggest value for which all methods are stable (in this case, $\Delta t = 0.421 \Delta x/c$). The resulting dispersion error as a function of grid spaces per wavelength is plotted in Fig. 5. Since $\Delta t$ is the same for each method, the mean error depends strongly on the order of the method used, and weakly on the actual method used. In the standard deviation, we see that the order of the method is again the main determinant of error. However, in terms of isotropy of dispersion error, the 4th order Lebedev and 4th order cubic methods significantly outperform the 4th order Yee method.

### 4.3.2. Equal memory comparison

Our baseline comparison is not really fair, though, since the Lebedev and cubic methods have a much higher grid density than the Yee grid. To simulate the same volume using the same $\Delta x$ and $\Delta t$, the Lebedev and cubic methods would require significantly more memory and more calculations per time step. For a fairer comparison, we next perform an equal memory, equal $\Delta t$ comparison. This means that $\Delta x$ and $\Delta t$ are chosen so that all methods have the same number of stored values per unit volume and per unit time.

To ensure that there are the same number of stored values per unit volume, $\Delta x$ must be increased for the Lebedev and cubic methods. That is, if the Yee grid uses $\Delta x = \Delta_0$, then the Lebedev grid uses $\Delta x = 2^{2/3} \Delta_0$ and the cubic grid uses $\Delta x = 2 \Delta_0$. To ensure that there are the same number of stored values per unit time, we set $\Delta t$ to be the same for all methods. Again, we choose this to be the maximum $\Delta t$ such that all methods are stable ($\Delta t = 0.421 \Delta x/c$). There is a potential for unfairness here because some methods are not running near their stability criterion, but we will address this later. The results with $\Delta x$ and $\Delta t$ set this way are shown in Fig. 6.
First, let us look at the three 4th-order methods. In terms of mean error, they are roughly at the same level. While the cubic grid is best at higher grid densities, it is significantly worse at lower grid densities. Unfortunately, the 4th order cubic and Lebedev collocated methods perform quite poorly in terms of dispersion isotropy. The 4th order Yee method is almost 10 dB better than the 4th order Lebedev, and almost 20 dB better than the 4th order cubic in terms of error standard deviation. Considering that the 4th order Hodge dual for the Lebedev grid is more complex than that of the 4th order Yee method, this is not great news for the 4th order collocated methods. (The Lebedev Hodge dual uses 12 neighbours as opposed to 6.)

One interesting feature is that the 4th-order Yee and the non-DEC 4th-order Yee have nearly identical dispersion characteristics (though they are not quite identical). This is somewhat surprising, given that the two methods certainly do not have equivalent update equations. The DEC approach actually uses extra neighbours, which do not appear at all in the stencil of the non-DEC method. At this point, it is unclear why their dispersion relationships should be so similar. It is possible that this is an artefact of homogeneous, isotropic space, and that the DEC method would prove to be more useful in more complicated cases. However, a more thorough investigation into this feature would be interesting.

Next, let us look at the pseudo-4th-order methods. We see, reassuringly, that the pseudo-4th-order Lebedev method is significantly better than the 2nd-order Lebedev method. This is true both for average dispersion error and for dispersion isotropy. So it seems that there is some merit in coupling the overlapping Yee grids which form the collocated grid. However, the 2nd order Yee method still outperforms the pseudo-4th order Lebedev, and we see that the pseudo-4th-order Yee is even better still. That is, moving to a pseudo-4th order Hodge dual improves the Yee method just as much as it improves the Lebedev method, even though there is no grid-coupling effect.

4.3.3. Equal memory at stability criterion comparison

As a final check, the dispersion error was determined with $\Delta x$ set to give an equal density of stored values and $\Delta t$ set at each method’s individual stability criterion. The results can be seen in Fig. 7. We see that the collocated methods perform about the same as before in terms of dispersion isotropy but much worse in terms of mean dispersion error.

However, this is not really a fair comparison because some methods are using a larger time step than others: the specifics are shown in Table 3. Thus, while the error may be worse for the collocated methods, the computational time required for a
Fig. 6. Comparison of dispersion error with all methods using equal memory (equal number of grid spaces per unit volume) and equal $\Delta t$. Note that, in this case, “grid spaces per wavelength” means the number of standard Yee grid spaces ($\Delta y$) per wavelength.

simulation could potentially be smaller. For example, while the collocated method has worse error than the 2nd order Yee, it is also covering almost 50% more time per time step while using the same amount of memory. These competing factors make it difficult to make a meaningful comparison, which is why equal memory and equal $\Delta t$ were used for the primary comparison.

4.3.4. Discussion

From this analysis, it seems that using higher-order Hodge duals to couple overlapping grids is not as beneficial as one might hope. While these new methods are noticeably improved compared to the lower-order methods, they are still outperformed by higher-order Yee methods. It seems that there is no extra benefit coming from coupling grids together. Rather, the improvement appears to be comparable with any move to a higher-order method. So, when dealing with homogeneous, isotropic media, the Yee method is likely still the best choice.

On the other hand, this analysis confirms that our generalized approach to deriving Hodge dual operators is a reasonable one. We were successful in creating methods which are at least comparable to the Yee method. While they have worse dispersion characteristics in isotropic media, they retain the advantage of being collocated. Since our approach outlined in Section 3.3 is quite general, it would be interesting to see it applied in more complicated situations. For example, it may be possible to derive a robust way to deal with sharp dielectric interfaces under the 4th order Yee method. Alternatively, a collocated method could be used as it is probably a better option for sharp interfaces as it is easier to impose tangential and normal boundary conditions. Furthermore, a collocated method could be used at a material interface, which could be connected to standard Yee grids in homogeneous regions via appropriate Hodge operators; the intention here would be to save on computations in homogeneous regions via a hybrid method. Regardless of the application, because of the way our approach is structured, we would still automatically guarantee that discrete conservation laws are obeyed at the interface. This would be quite difficult to guarantee using standard FDTD approaches. Even if the higher-order collocated methods do not find a use, the method used to derive them certainly shows promise.
5. Conclusions

In this paper, we have used discrete exterior calculus (DEC) in an attempt to improve collocated methods. Ostensibly, our hope was that using higher-order Hodge duals to couple overlapping grids would make collocated methods comparable to the Yee method in isotropic media, while still maintaining advantages in anisotropic media and at interfaces. Given our dispersion error analysis, it is difficult to say whether this initial goal was met. We did see improvement in the collocated methods, but we also saw a similar improvement in the Yee methods. Thus moving to a higher-order collocated scheme is not necessarily the best option, since even better results can be obtained in isotropic media with a higher-order Yee method. The collocated grid methods may be useful, but they are not a “one size fits all” solution.

More interesting, though, are the theoretical methods developed along the way. The method used in Section 3.3 to derive higher-order Hodge duals is quite general, and could be applied to other problems. In particular, we used basis forms $\phi^{(m)}d\xi^\nu \wedge d\kappa_\nu$, which were simple polynomials, but they could be tailored to more interesting problems. For example, a spatial discontinuity could be added to the basis forms at a sharp material interface, allowing the boundary conditions to be properly enforced. In the second-order case this is equivalent to [3], but it could be extended to the fourth order for both the Yee and collocated methods. Certainly it would be challenging to solve for the coefficients, but the structure of our approach guarantees a number of desirable properties in the result. In general, carefully-selected basis forms could be used to account for behaviour at a sub-grid level, while still ensuring that discrete conservation laws are obeyed.

At very least, the DEC approach provides a clear and simple way of understanding existing methods for solving Maxwell’s equations. It confines error to the Hodge dual operation, and guarantees a number of desirable conservation properties: conservation properties which are not necessarily met in general finite-difference methods. The DEC approach is an elegant way to derive FDTD methods, and continues to show promise as it is developed further.

Acknowledgements

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant number RGPIN 2014-03984.
References


