



High-order FDTD methods via derivative matching for Maxwell's equations with material interfaces

Shan Zhao ^a, G.W. Wei ^{a,b,*}

^a *Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA*

^b *Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA*

Received 24 September 2003; received in revised form 29 March 2004; accepted 29 March 2004

Available online 27 April 2004

Abstract

This paper introduces a series of novel hierarchical implicit derivative matching methods to restore the accuracy of high-order finite-difference time-domain (FDTD) schemes of computational electromagnetics (CEM) with material interfaces in one (1D) and two spatial dimensions (2D). By making use of fictitious points, systematic approaches are proposed to locally enforce the physical jump conditions at material interfaces in a preprocessing stage, to arbitrarily high orders of accuracy in principle. While often limited by numerical instability, orders up to 16 and 12 are achieved, respectively, in 1D and 2D. Detailed stability analyses are presented for the present approach to examine the upper limit in constructing embedded FDTD methods. As natural generalizations of the high-order FDTD schemes, the proposed derivative matching methods automatically reduce to the standard FDTD schemes when the material interfaces are absent. An interesting feature of the present approach is that it encompasses a variety of schemes of different orders in a single code. Another feature of the present approach is that it can be robustly implemented with other high accuracy time-domain approaches, such as the multiresolution time-domain method and the local spectral time-domain method, to cope with material interfaces. Numerical experiments on both 1D and 2D problems are carried out to test the convergence, examine the stability, access the efficiency, and explore the limitation of the proposed methods. It is found that operating at their best capacity, the proposed high-order schemes could be over 2000 times more efficient than their fourth-order versions in 2D. In conclusion, the present work indicates that the proposed hierarchical derivative matching methods might lead to practical high-order schemes for numerical solution of time-domain Maxwell's equations with material interfaces.

© 2004 Elsevier Inc. All rights reserved.

Keywords: Maxwell's equations; Finite-difference; Time-domain methods; Staggered grids; Material interfaces; High-order methods; Stability analysis

* Corresponding author. Tel.: +1-517-353-4689; fax: +1-517-432-1562.

E-mail addresses: szhao@math.msu.edu (S. Zhao), wei@math.msu.edu (G.W. Wei).

1. Introduction

Recently, scientific and technological advancements in nano/micro devices, optical devices, microwave circuits, antennas, aircraft radar signature, and telecommunication chips, call for innovative computational methods for solving Maxwell's equations which govern the propagation and scattering of electromagnetic waves. The finite-difference time-domain (FDTD) method [1,2] has been a main workhorse of computational electromagnetics (CEM) in the time domain over the past few decades, due to its simplicity, lack of dissipative error, and having very low cost per grid node. However, the FDTD method suffers from relatively large dispersive error [3], (first order) staircased representation for complex geometries and boundaries, and reduction of accuracy to the first order at material interfaces [4]. Consequently, the FDTD is efficient only for regular geometries of small or moderate size, without dielectric interfaces. For broadband applications and problems including material interface, wave scattering and penetration over large and complex domains, the grid size required by using the FDTD method could become prohibitively expensive for modern computers. Much progress has been made in the past two decades in improving the FDTD method, including plentiful methods for removing the staircased approximation for boundaries and geometries [5–9], and numerous high-order FDTD methods [3,10–16]. Here, by high order we refer to orders being higher than three, which are essential for modern problems of moderately high frequency (short) waves and/or large domain in nature. However, at present, there is still a pressing need for a generalized FDTD method which is simple, robust, and high order for large scale computations involving complex geometries, boundary conditions, and material interfaces. In particular, very few high-order schemes are available for handling material interfaces.

Apart from the FDTD-based high-order methods, several other time-domain methods that are able to produce extremely small dispersive errors have also been proposed in the literature. Based on orthonormal wavelet expansions, a multiresolution time-domain (MRTD) method has been introduced for numerical analysis of microwave structures [17,18]. By using the perfectly matched layer (PML) [19] technique to bypass the limitation of the Fourier pseudospectral method on periodic boundary conditions, a Fourier pseudospectral time-domain (PSTD) method was successfully developed to solve wave propagation and radiation problems [20,21]. Recently, a wavelet collocation scheme, based on the mathematical framework of the discrete singular convolution (DSC) algorithm [22] for spatial derivative approximation has been proposed for scattering and guided wave problems described by Maxwell's equations on uniform grids [23] and on staggered grids [24–27]. This wavelet collocation scheme, entitled local spectral time-domain (LSTD) method, is capable of handling many boundary conditions commonly occurred in CEM. To achieve a satisfied accuracy, the required grid points per wavelength (PPW) of these three time-domain methods could be very low. Especially, the PSTD and LSTD can deliver extremely high accuracy by using a coarse grid of about two PPW, i.e., the Nyquist sampling rate limitation [20,24,25].

In general, the high-order FDTD schemes and the three other high accuracy time-domain methods all retain much of the simplicity of the original Yee algorithm [1]. Similar to the FDTD method, these time-domain approaches are also usually applied to a simple Cartesian grid or a single structured grid. However, it is well known that numerical schemes based on a single structured grid would be restricted to be used only for geometrically simple domains. Depending on the problems of interest, the FDTD methods can be directly extended to an orthogonal curvilinear grid by means of a body-conforming domain transformation to handle wave scattering and propagation in moderately complex geometries. The use of the fundamental FDTD scheme on nonorthogonal and unstructured grids has been extensively studied [2]. However, these efforts of constructing FDTD type of approaches based on complex grids might not be as efficient as expected. To handle truly complex geometries of practical interest, some sort of multidomain grid schemes, or unstructured grid schemes should be considered. Numerous finite volume time-domain methods [28–30] and finite element time-domain methods [31–34] have been put forward to systematically handle geometrically complex problems in CEM, since the most flexible grid, the unstructured grid, is best adapted to the

finite volume and finite-element approaches. Many vector elements, such as Nedelec elements [31], Whitney forms [35,36], and curl/div conforming vector elements [37] are constructed to provide a discrete analog to the continuous vector algebra and to enforce only minimal continuity across element boundaries. Recently, a promising high-order nodal method on fully unstructured grids has been developed by means of a discontinuous Galerkin formulation and a penalty method for boundary conditions [38]. As alternatives, various domain decomposition approaches have been proposed. Since the spectral methods which can deliver strikingly high accuracy are unfortunately quite inflexible to handle complex geometries, recently, the development of the domain decomposition pseudospectral or spectral methods has attracted much attentions [39–44]. A block pseudospectral (BPS) method was successfully developed for one- (1D) and two-dimensional (2D) electromagnetic problems by using overlapping or composite grids to achieve a highly accurate coupling of subdomains [39,40]. Remarkably, innovative ways to appropriately enforce the physical jump conditions at material interfaces are presented by using fictitious points (FPs) [39,40]. Based on a characteristic formulation of jump conditions, another interesting scheme for modeling material interfaces has been presented in a high-accuracy FDTD approach [44]. Often realized on subdomain grid systems, this high-accuracy approach numerically yields substantial error reductions to a typical fourth-order FDTD method [44]. At present, multidomain methods [41–44] are some of the best available approaches for handling complex geometries in CEM. The offset of a multidomain formulation is the requirement of correct connections between subdomains, which could be constricting and sometimes may be difficult to fulfill. Interpolation schemes are usually required. Also, properties such as conservation and monotonicity may have to be ensured across blocks. Moreover, the use of multidomain framework and unstructured grids inevitably introduces a need of automated grid generation and a severe restriction on stability. Even though considerable progress has been made in grid generation, the formation of a good quality block structured grid system or an unstructured grid system in geometrically complex domain remains a nontrivial and time-consuming task.

For moderately complex geometries, it is unnecessary to resort to unstructured grids and/or domain decompositions. Embedding interface FDTD methods [4,45–47] which make use of simple Cartesian grids, have been introduced to deal with moderately complex geometries. To overcome the staircasing problems and to impose the physically correct jump conditions, which are often encountered in the modeling of complex geometries, appropriate local modifications of the differentiation scheme close to boundaries and interfaces in a preprocessing stage are essential. While maintaining the simplicity and computational efficiency of the Yee scheme [1,2], these modified FDTD methods fully restore second-order accuracy, even in case of curved boundaries and interfaces, by using a simple Cartesian grid [4,45,46]. However, the extension of these embedding FDTD methods to higher-order accuracy remains a challenge [4].

Recently, some important contributions to the field have been made by Yefet and Turkel [48], Yefet and Petropoulos [49], Xie et al. [50,51], and Hesthaven [4] who generalize the embedding schemes to fully fourth-order FDTD methods on a simple structured grid with material interfaces. It is well known that the suitable and high-order interface schemes are crucial to high-order time-domain methods [4]. The development of such interface schemes can be considerably more complex than the derivation of high-order differential stencils. Therefore, the significant accomplishment of these fourth-order schemes [48–51] is that the physical jump conditions at material interfaces are correctly enforced up to high-order, so that fourth-order convergence is uniformly assured over the entire domain. Subtle interfaces techniques with one-sided difference approximations and extrapolations are employed in these methods. Similar to the embedding methods [4,45,46], the modeling of interfaces results in a local modification of the differential scheme in a preprocessing stage [4,48–51].

The studies of fourth-order embedding FDTD schemes [4,48–51] open up the opportunity for developing other more general and robust high-order interface schemes. There are still many important questions remaining unanswered. One interesting instance as noted in [4] is that whether it is possible to extend

the interface schemes [4,48–51] beyond the fourth-order accuracy, which is of greatly significance for large domain and high frequency wave computations. Since the local modifications of the differential schemes in [4,48–51] are explicitly implemented, the generalization to higher-order along this line may be theoretically admissible, however, would be quite empirical as well as mathematically complicated [4].

It is therefore of great interest to develop a systematic approach to correctly enforce the physical jump conditions at material interfaces to a variety of orders. To this end, we present herein a series of novel hierarchical implicit derivative matching methods for interface modeling, which can greatly exceed fourth-order accuracy. Our schemes make use of FPs, a technique extensively used in our previous work [22–26,52–55] to locally modify the differential stencils near the interfaces, which is an idea first proposed by Driscoll and Fornberg [39,40] in their BPS methods. However, only one single structured grid is assumed in the present work, similar to other high-order embedding FDTD methods [4,48–51]. Moreover, local modifications of the differential stencils at beginning, the so-called preprocessing stage, rather than subdomain coupling at each time step, are carried out in the present study. The stability issue of the proposed methods is examined. To enhance the stability and robustness of interface schemes, a novel hierarchical derivative matching method is proposed. Extensive numerical studies in both 1D and 2D are considered. We note that in 2D, our best high-order schemes can be over 2000 times more efficient than our fourth-order schemes.

The rest of this paper is organized as follows. Section 2 is devoted to a brief description of Maxwell's equations. Model equations in both 1D and 2D are considered. Commonly used boundary conditions in CEM are briefly introduced. Various issues associated with 1D modeling are dealt with in Section 3. The formalism of derivative matching (DM) is discussed in detail. Two new schemes, an implicit derivative matching method (IDM) and a hierarchical derivative matching method (HDM), are proposed. Comprehensive stability analysis is conducted. Numerical tests are carried out to validate the proposed methods. Section 4 presents the generalization of the proposed derivative matching methods to 2D electromagnetic problems. The difficulties of derivative matching approach in 2D are discussed. Two new schemes, a quasi-fourth-order 2D scheme and a 2D hierarchical derivative matching method are constructed. Stability issues and numerical tests in 2D schemes are examined. Finally, conclusions are presented in Section 5.

2. Governing equations

By assuming the absence of charge density and current source, the time-dependent Maxwell's equations for electromagnetic fields in free space read

$$\begin{aligned} \frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} &= 0, & \nabla \cdot \mathbf{B} &= 0, \\ \frac{\partial \mathbf{D}}{\partial t} - \nabla \times \mathbf{H} &= 0, & \nabla \cdot \mathbf{D} &= 0, \end{aligned} \quad (1)$$

where \mathbf{D} and \mathbf{B} are, respectively, the electric and magnetic flux densities, while \mathbf{E} and \mathbf{H} are, respectively, the electric and magnetic field intensities. In addition, the following linear isotropic constitutive relations are satisfied:

$$\mathbf{D} = \epsilon \mathbf{E}, \quad \mathbf{B} = \mu \mathbf{H},$$

where the electric permittivity ϵ and the magnetic permeability μ of material are piecewise constants. A nondimensional form of the equations is considered, i.e., $\epsilon = \mu = 1$ in free space.

Both 1D and 2D electromagnetic problems are considered in the present study. In 2D, Maxwell's equations can be decomposed into two independent sets of equations, the transverse magnetic (TM) modes and the transverse electric (TE) modes. In this paper, we concern ourselves with the TM case for 2D studies

$$\mu \frac{\partial H_x}{\partial t} = -\frac{\partial E_z}{\partial y}, \quad \mu \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x}, \quad \epsilon \frac{\partial E_z}{\partial t} = \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right). \quad (2)$$

The TM system (2) can be further simplified to 1D, resulting in an x -directed, z -polarized transverse electromagnetic (TEM) mode

$$\epsilon \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x}, \quad \mu \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x}. \quad (3)$$

As simple instances of Maxwell's equations, the TM mode (2) and TEM mode (3) still preserve the essential features of Maxwell's equations, e.g., the two-way wave propagation and the loss of smoothness across material interfaces. Thus, these two equations are ideally suitable for benchmarking novel numerical approaches. Meanwhile, we note that the numerical modeling of the TE polarization might be more complex than that of the TM polarization, especially when surface waves at interfaces are studied.

In order to obtain a complete description of an electromagnetic problem, besides the various forms of Maxwell's equations, we also need certain boundary conditions, which relate field components on either side of a boundary or interface. At an interface between two media, say medium 1 and medium 2, the boundary conditions can be expressed mathematically as:

$$\hat{\mathbf{n}} \times (\mathbf{E}_1 - \mathbf{E}_2) = \mathbf{0}, \quad \hat{\mathbf{n}} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \mathbf{0}, \quad (4)$$

$$\hat{\mathbf{n}} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{0}, \quad \hat{\mathbf{n}} \cdot (\mathbf{B}_1 - \mathbf{B}_2) = \mathbf{0} \quad (5)$$

for electric fields and magnetic fields, respectively. Here $\hat{\mathbf{n}}$ is the unit vector normal to the interface, pointing from medium 2 into medium 1. A special case with one of the media, say medium 2, being a perfect electric conductor (PEC), is practically important. Since a perfect conductor cannot sustain a field inside, PEC boundary conditions now reduce to

$$\hat{\mathbf{n}} \times \mathbf{E} = \mathbf{0}, \quad \hat{\mathbf{n}} \cdot \mathbf{B} = \mathbf{0}. \quad (6)$$

Another important type of boundary conditions for numerical simulation of Maxwell's equations is the absorbing boundary condition. Such conditions must be specified in practical computations at the outer boundary in order to obtain a unique solution for open or unbounded electromagnetic problems. Since the high-order time-domain methods considered in this paper are basically the extensions of the Yee algorithm, the widely used absorbing conditions, such as far-field radiation condition [56], PML boundary condition [19], and Lorentz material model absorber layers [57,58], could be used in these extended schemes for the simulation of radiation out of a domain, if required.

3. One-dimensional studies

We first consider in this section the method of implicit derivative matching in 1D cases. Some stability issues are considered. A novel hierarchical implicit derivative matching is discussed in detail. Several numerical experiments of time-domain electromagnetic problems are carried out.

3.1. Method of derivative matching with fictitious points

For convenience, we consider the vector form of the 1D Maxwell's equation (3) in present study,

$$\frac{\partial \mathbf{q}}{\partial t} = A \frac{\partial \mathbf{q}}{\partial x}, \quad \mathbf{q} = \begin{bmatrix} E_z \\ H_y \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1/\epsilon \\ 1/\mu & 0 \end{bmatrix}. \quad (7)$$

We suppose the domain under study consisting of two dielectric media, with the interface at $x = \xi$. Consequently, the coefficient matrix A attains different values in media 1 and 2, i.e., $A = A_1$ if $x < \xi$ or $A = A_2$ if $x > \xi$. For simplicity, we denote one-sided derivatives at the interface by

$$\mathbf{q}^{(p)}(\xi^-, t) = \left. \frac{\partial^p \mathbf{q}(x, t)}{\partial x^p} \right|_{x \rightarrow \xi^-}, \quad \mathbf{q}^{(p)}(\xi^+, t) = \left. \frac{\partial^p \mathbf{q}(x, t)}{\partial x^p} \right|_{x \rightarrow \xi^+}.$$

It can be derived from interface conditions (4) and (5) that the electromagnetic fields \mathbf{q} are continuous across the material interface, i.e.,

$$\mathbf{q}^{(0)}(\xi^-, t) = \mathbf{q}^{(0)}(\xi^+, t). \quad (8)$$

This condition is referred to as (zeroth-order) physical jump condition. Since the fields are continuous, their time derivatives are also continuous. Thus, by using the Maxwell's equation (7), we also have that

$$A_1 \mathbf{q}^{(1)}(\xi^-, t) = A_2 \mathbf{q}^{(1)}(\xi^+, t). \quad (9)$$

In principle, one can repeat this as often as needed to generate jump conditions of any order

$$A_1^p \mathbf{q}^{(p)}(\xi^-, t) = A_2^p \mathbf{q}^{(p)}(\xi^+, t), \quad p = 0, 1, 2, \dots \quad (10)$$

These jump conditions are physically exact and relate the electromagnetic fields and their derivatives as taken from the two sides of the interface.

Driscoll and Fornberg [39,40] developed their BPS method by matching the interface with the derivative condition (10) and fictitious grid values. This novel technique for coupling subdomains is referred to as the method of derivative matching (DM) [39,40]. The use of FPs significantly enhances the flexibility of the DM modeling. Similar fictitious domain approaches have also been studied in different occasions [59,60,52,53]. In the present study, it is of great interest to generalize the idea of the DM to single structured grid and to the framework of FDTD methods. In fact, it is clear that the DM modeling near interfaces is much in the spirit of those in the embedding FDTD methods and their extensions [4,45–46,48–51]. The embedding FDTD schemes are usually constructed based on the zeroth-order jump conditions, while the BPS method repeatedly utilizes these conditions. The distinct difference between the BPS method [39,40] and the embedding FDTD schemes [4,45–46,48–51] is that jump conditions are enforced via solving fictitious values at each time step in the former, while in the latter, this is achieved by locally altering the finite-difference (FD) weights in a preprocessing stage. It is noted that it might be more efficient if the DM of the BPS method could be carried out once for all at the beginning of the computation. Meanwhile, this is computationally admissible, since the algebraic systems at different time steps attain the identical organization, albeit have different values.

In order to study such an implicit DM modeling, we first establish some notations. Referring to Fig. 1, we denote function values under consideration (either E_z or H_y) at original and fictitious points as g_i and f_i (for $i = 1, 2, \dots, 2m$), respectively. The FP f_i is located at the same position as g_i . The total number of FPs is $2m$. For simplicity, a uniform grid is assumed in the present study. It is noted that interface $x = \xi$ may not need to be laid on the grid in the present modeling. In the BPS method [39,40], the values of f_i are solved

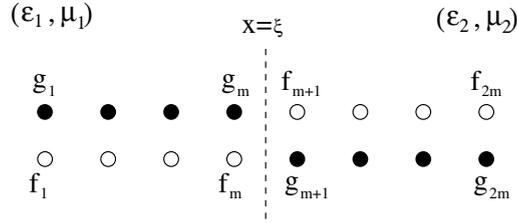


Fig. 1. Illustration of FPs and notation used in the proposed IDM method.

based on the values of g_i at each time step. However, in the present paper, a preprocessing scheme is sought. To this end, we assume the following representation:

$$f_i = \sum_{j=1}^{2m} r_{i,j} g_j \quad \text{for } i = 1, 2, \dots, 2m, \tag{11}$$

where $r_{i,j}$ are representation coefficients. With this representation, it is not necessary to solve f_i at each time step. One needs only to determine $r_{i,j}$, $i, j = 1, 2, \dots, 2m$, once.

We consider the matrix form of representation (11)

$$F = RG,$$

where $F = (f_1, f_2, \dots, f_{2m})^T$, $G = (g_1, g_2, \dots, g_{2m})^T$, and $R = (r_{i,j})_{i,j=1}^{2m}$. We shall determine the unknown coefficient matrix R . For convenience, we consider the rows of R as new variables, $R_j = (r_{j,1}, r_{j,2}, \dots, r_{j,2m})$ for $j = 1, 2, \dots, 2m$. To determine the coefficients/elements of R , it is equivalent to solve a vector \hat{R} from an algebraic system, where $\hat{R} = (R_1, R_2, \dots, R_{2m})^T = (r_{1,1}, r_{1,2}, \dots, r_{1,2m}, r_{2,1}, r_{2,2}, \dots, r_{2,2m}, \dots, r_{2m,1}, r_{2m,2}, \dots, r_{2m,2m})^T$. It is noted that the dimension of the matrix system for determining \hat{R} is $4m^2 \times 4m^2$. Similarly, we define $2m$ vectors I_j as the rows of a $2m \times 2m$ identity matrix I , i.e., $I_1 = (1, 0, 0, \dots, 0)$, $I_2 = (0, 1, 0, \dots, 0)$, \dots , $I_{2m} = (0, 0, 0, \dots, 0, 1)$. Symbolically, we have

$$f_j = R_j G \quad \text{and} \quad g_j = I_j G. \tag{12}$$

Similar to the BPS method [39,40], up to $(2m - 1)$ th-order physical jump conditions are employed for determining the representation weights of $2m$ FPs. Standard central FD approximations are utilized to discretize these $2m$ jump conditions. Throughout the study, the FD weights are calculated via a fast algorithm introduced in [61]. The discretized equations are evaluated at each grid point to form $2m \times 2m$ algebraic equations. To illustrate the idea, we consider one particular jump condition as an example

$$\frac{1}{\epsilon_1} E_z^{(2)}(\xi^-, t) = \frac{1}{\epsilon_2} E_z^{(2)}(\xi^+, t). \tag{13}$$

We consider the following FD approximation of Eq. (13):

$$\frac{1}{\epsilon_1} \left[\sum_{i=1}^m w_{2,i} g_i + \sum_{i=m+1}^{2m} w_{2,i} f_i \right] = \frac{1}{\epsilon_2} \left[\sum_{i=1}^m w_{2,i} f_i + \sum_{i=m+1}^{2m} w_{2,i} g_i \right], \tag{14}$$

where $w_{2,i}$, $i = 1, 2, \dots, 2m$, are the FD weights for the second-order derivative approximation. It is obvious from Eq. (14) that although a simple computational domain is assumed, the domain decomposition point of view is adopted in the present study. The approximation style of Eq. (14) is very close to that used in the BPS method.

To form the desired algebraic equations, we rewrite Eq. (14) as

$$\frac{1}{\epsilon_1} \sum_{i=m+1}^{2m} w_{2,i} f_i - \frac{1}{\epsilon_2} \sum_{i=1}^m w_{2,i} f_i = \frac{1}{\epsilon_2} \sum_{i=m+1}^{2m} w_{2,i} g_i - \frac{1}{\epsilon_1} \sum_{i=1}^m w_{2,i} g_i. \quad (15)$$

The right-hand side of Eq. (15) are known values. We then substitute two relations in Eq. (12) into Eq. (15), to obtain

$$\frac{1}{\epsilon_1} \sum_{i=m+1}^{2m} w_{2,i} R_i G - \frac{1}{\epsilon_2} \sum_{i=1}^m w_{2,i} R_i G = \frac{1}{\epsilon_2} \sum_{i=m+1}^{2m} w_{2,i} I_i G - \frac{1}{\epsilon_1} \sum_{i=1}^m w_{2,i} I_i G. \quad (16)$$

By eliminating the common abstract variable G , we finally obtain $2m$ algebraic equations from the jump condition (13)

$$\frac{1}{\epsilon_1} \sum_{i=m+1}^{2m} w_{2,i} R_i^T - \frac{1}{\epsilon_2} \sum_{i=1}^m w_{2,i} R_i^T = \frac{1}{\epsilon_2} \sum_{i=m+1}^{2m} w_{2,i} I_i^T - \frac{1}{\epsilon_1} \sum_{i=1}^m w_{2,i} I_i^T. \quad (17)$$

These equations are independent of field values. Algebraic equations for other jump conditions can be similarly derived.

Unlike the BPS method [39,40], the function values on FPs, i.e., $\{f_i\}_{i=1}^{2m}$, are never evaluated in the present study. In contrary, by means of representation coefficients $r_{i,j}$, we locally modify the differential stencil near the material interface at the beginning of the computation, as in the embedding FDTD schemes [4,45,46,48–51]. Thus, the present numerical modeling can be regarded as an implicit version of DM. One possible advantage of the implicit derivative matching (IDM) is that it might be more efficient for numerical simulation of long time wave propagation. It is noted that although the IDM is formulated on a uniform grid, it actually works on more general grid settings. For example, either staggered or nonstaggered grid systems can be employed together with a time-domain solver. As well, the IDM can be easily adopted in a block structured grid like the original DM method [39]. Moreover, a nonuniform grid could be applied near the interface to enhance adaptivity with appropriate Lagrange coefficients of differentiation. In fact, the IDM can even be extended to based on unstructured grids. Furthermore, this kind of representation could also be utilized in an implicit time-stepping scheme or a resolution of boundary value problems or eigenvalue problems, due to its implicit nature.

The major difference between the proposed scheme and the embedding FDTD schemes [4,45,46,48–51] is that the present local modification of differential stencil is modeled in a systematic way, such that it can be made up to arbitrarily high-order in principle, while the previous schemes were manually constructed one by one. Thus, the IDM yields an excellent way to generalize the embedding FDTD scheme to high orders. Furthermore, it could be applied together with various different high-order time-domain methods, such as high-order FDTD methods [3,10–16], the MRTD method [17,18], and the LSTD method [24–26]. In these applications, the number of FPs (m) could be simply specified according to the length of stencil involved in these time-domain methods, and significant improvement of accuracy is expected. It is noted that, however, the IDM cannot be directly incorporated into a global formulation, e.g., the PSTD method [20,21], especially when at least two interfaces are present. One possible drawback of the proposed scheme is that the size of algebraic system to be solved for the interface is much larger than that in the BPS method [39]. This might imply a substantial memory requirement and requires longer CPU time in computation.

3.2. Numerical tests of the IDM method

We consider two model problems in this subsection to investigate the numerical performance of the IDM method. Both problems under consideration are for TEM waves, governed by the 1D Maxwell's equation (7), but with possibly different jump conditions. These two model problems have the same electromagnetic structure as following. The electromagnetic fields are defined on a standard interval $x \in [-1, 1]$, with PEC walls located at both ends. The interior of the resonator is assumed to be nonmagnetic with $\mu = 1$ and filled with two dielectric media with ϵ_1 and ϵ_2 . The material interface is at $\xi = 0$.

In the present study, a uniform staggered grid system is used for E_z and H_y , with either an E node or an H node on the boundaries $x = \pm 1$. The standard high-order central FDTD approximations are employed for the spatial discretization

$$E_z^{(1)}(x_{i+1/2}, t) \approx \frac{1}{\Delta x} \sum_{j=-M}^{-1} w_j E_z(x_{i+1+j}, t) + \frac{1}{\Delta x} \sum_{j=1}^M w_j E_z(x_{i+j}, t), \quad (18)$$

$$H_y^{(1)}(x_i, t) \approx \frac{1}{\Delta x} \sum_{j=-M}^{-1} w_j H_y(x_{i+1/2+j}, t) + \frac{1}{\Delta x} \sum_{j=1}^M w_j H_y(x_{i-1/2+j}, t), \quad (19)$$

where Δx is the grid spacing and M is the half computational bandwidth. Here w_j , for $j = \pm 1, \dots, \pm M$, are the normalized FD weights. For example, when $M = 1$, we simply have $w_{-1} = -1$ and $w_1 = 1$ as in the Yee algorithm [1,2]. When $M = 2$, the weights for the fourth-order accurate stencil are given as $(w_{-2}, w_{-1}, w_1, w_2) = \frac{1}{24}(1, -27, 27, -1)$ [2,4,49–51]. The weights for higher-order stencils can be computed as discussed in [61].

Throughout the study, we always maintain the translation invariance property of the differential stencil, i.e., only one basic approximation kernel (w_j) is required for the entire computational domain $[-1, 1]$. Obviously, this property has distinct computational efficiency. To keep this property near boundary and/or interface, it is clearly necessary to create a fictitious domain outside the boundary or interface and correspondingly generate fictitious values. At material interface, this is equivalent to carry out the IDM with $m = M$ to locally modify the basic differential stencil. An LU decomposition is utilized to solve the algebraic system of the IDM modeling. At the PEC walls, it can be derived from Eq. (6) that the fields satisfy extra conditions at two ends [39,45]

$$\frac{\partial^p E_z}{\partial x^p} \Big|_{x=\pm 1} = 0, \quad p = 0, 2, 4, \dots, \quad (20)$$

$$\frac{\partial^p H_y}{\partial x^p} \Big|_{x=\pm 1} = 0, \quad p = 1, 3, 5, \dots \quad (21)$$

Since we have a grid node on the boundary, following the BPS method [39], the symmetric or anti-symmetric properties of electric and magnetic fields are directly exploited in numerical boundary modeling. This results in the so-called anti-symmetric and symmetric boundary extensions, respectively, for E_z and H_y [24–26]. The similar boundary modeling is referred to as the image principle in the MRTD method [17,18]. In case that there is no grid point on the boundary as considered in [45], an IDM for imposing boundary conditions can be similarly formulated, as suggested in [39].

After completing the spatial discretization, the resulting semidiscrete form of Maxwell's equation (7) at time $t = t_n$ can be expressed as

$$\frac{\partial \mathbf{Q}^n}{\partial t} = \frac{1}{\Delta x} \mathbf{S} \mathbf{Q}^n, \quad (22)$$

where $\mathbf{Q}^n = (E_z(x_1, t_n), \dots, E_z(x_N, t_n), H_y(x_{1/2+1}, t_n), \dots, H_y(x_{1/2+N}, t_n))^T$ with N being the total number of grid points for either E_z or H_y . Here, the matrix \mathbf{S} represents the complete spatial approximation to the right-hand side of Eq. (7). In the present study, the classical fourth-order Runge–Kutta (RK4) method is employed for the temporal integration, i.e.,

$$\begin{aligned}\mathbf{Q}^{n+1} &= \mathbf{Q}^n + \frac{1}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4), \\ \mathbf{K}_1 &= \frac{\Delta t}{\Delta x} \mathbf{S} \mathbf{Q}^n, \quad \mathbf{K}_2 = \frac{\Delta t}{\Delta x} \mathbf{S}(\mathbf{Q}^n + \frac{1}{2}\mathbf{K}_1), \\ \mathbf{K}_3 &= \frac{\Delta t}{\Delta x} \mathbf{S}(\mathbf{Q}^n + \frac{1}{2}\mathbf{K}_2), \quad \mathbf{K}_4 = \frac{\Delta t}{\Delta x} \mathbf{S}(\mathbf{Q}^n + \mathbf{K}_3).\end{aligned}\tag{23}$$

Throughout the paper, the time step Δt is chosen to be sufficiently small so that the error of time discretization is negligible comparing to that of spatial discretization.

We first consider a 1D model problem with the usual jump conditions discussed above. The exact solution of this model problem can be given as [45]

$$E_z = \begin{cases} [a_1 \exp(i\sqrt{\epsilon_1}\omega x) - b_1 \exp(-i\sqrt{\epsilon_1}\omega x)] \exp(i\omega t), & -1 \leq x \leq 0, \\ [a_2 \exp(i\sqrt{\epsilon_2}\omega x) - b_2 \exp(-i\sqrt{\epsilon_2}\omega x)] \exp(i\omega t), & 0 \leq x \leq 1, \end{cases}\tag{24}$$

$$H_y = \begin{cases} \sqrt{\epsilon_1} [a_1 \exp(i\sqrt{\epsilon_1}\omega x) + b_1 \exp(-i\sqrt{\epsilon_1}\omega x)] \exp(i\omega t), & -1 \leq x \leq 0, \\ \sqrt{\epsilon_2} [a_2 \exp(i\sqrt{\epsilon_2}\omega x) + b_2 \exp(-i\sqrt{\epsilon_2}\omega x)] \exp(i\omega t), & 0 \leq x \leq 1, \end{cases}\tag{25}$$

where $i = \sqrt{-1}$ and $a_1 = (\sqrt{\epsilon_2} \cos(\sqrt{\epsilon_2}\omega)) / (\sqrt{\epsilon_1} \cos(\sqrt{\epsilon_1}\omega))$, $a_2 = \exp(-i\omega(\sqrt{\epsilon_1} + \sqrt{\epsilon_2}))$, $b_1 = a_1 \exp(-i2\sqrt{\epsilon_1}\omega)$, and $b_2 = a_2 \exp(i2\sqrt{\epsilon_2}\omega)$. For homogeneous medium with $\epsilon = \epsilon_1 = \epsilon_2$, the wavenumber ω simply takes the value of $\omega = 2\pi/\sqrt{\epsilon}$, while when the waveguide is filled with two different media, the value of ω is obtained as the solution to equation

$$-\sqrt{\epsilon_2} \tan(\sqrt{\epsilon_1}\omega) = \sqrt{\epsilon_1} \tan(\sqrt{\epsilon_2}\omega).$$

An example plot of fields in Eqs. (24) and (25) at $t = 0$ with $\epsilon_1 = 1$, $\epsilon_2 = 2.25$, and $\omega \approx 5.07218116182516$ is given in Fig. 2. It is noted that the exact solution loses its regularity at the material interface, although it is continuous.

The numerical results of this 1D problem with $\omega \approx 5.07218116182516$ by using the high-order FDTD approximation with the IDM scheme for several different M values are listed in Table 1. For a comparison, results obtained by solely employing the standard FDTD method are also given in Table 1. It is clear from the table that the standard FDTD results are all essentially first-order accurate, no matter how large M is. In other words, the full accuracy of high-order FDTD approximations is deteriorated due to the discontinuous nature of the media. However, it is obvious from Table 1 that after the IDM is applied, the full accuracy of the FDTD approximations is recovered. The numerical results clearly display the theoretical rate of convergence, i.e., $(2M)$ th-order for the FDTD scheme with a given M , except when the precision limit is reached, see also Fig. 3. Consequently, extremely high accuracy can be achieved in our numerical simulations for this 1D electromagnetic problem with inhomogeneous media. Finally, by using $M = 8$, we have already attained the highest possible accuracy on the coarsest grid with $N = 50$, so that the numerical result of $M = 8$ seems not converging any further when the grid is refined in Fig. 3. In fact, the spatial discretization error herein should be restricted by either the temporal discretization error or the machine limit.

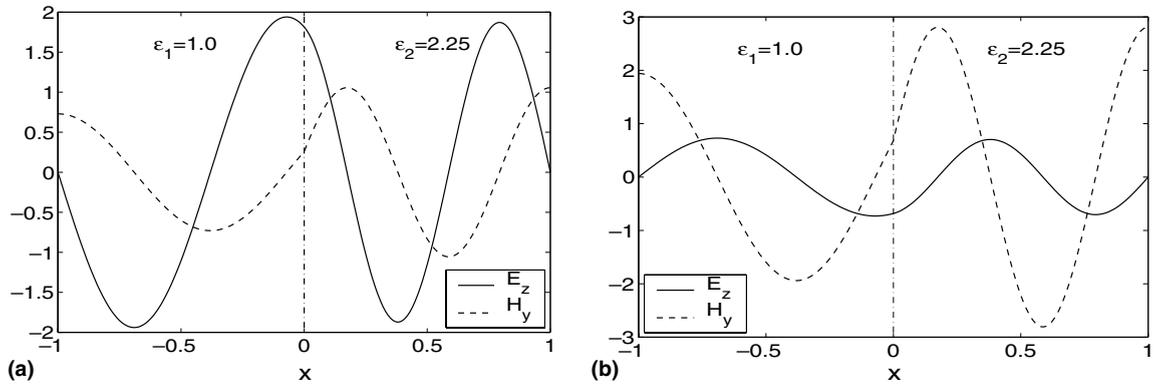


Fig. 2. Solution of the 1D model problem at $t = 0$ with $\omega \approx 5.07218116182516$. (a) Real part; (b) imaginary part.

Table 1

The L_2 errors in E_z and H_y at $t = \pi$ by using N grid points for each field intensity for the 1D model problem with $\omega \approx 5.07218116182516$

M	N	FDTD				FDTD with IDM ($m = M$)			
		E_z		H_y		E_z		H_y	
		Error	Rate	Error	Rate	Error	Rate	Error	Rate
1	50	1.56(-1)		1.99(-1)		7.85(-2)		9.81(-2)	
	100	5.98(-2)	1.379	7.63(-2)	1.386	1.94(-2)	2.014	2.41(-2)	2.025
	200	2.56(-2)	1.226	3.26(-2)	1.226	4.83(-3)	2.008	5.97(-3)	2.013
2	50	8.45(-2)		1.10(-1)		7.90(-4)		9.74(-4)	
	100	4.23(-2)	0.998	5.45(-2)	1.011	4.83(-5)	4.032	5.92(-5)	4.040
	200	2.12(-2)	0.999	2.72(-2)	1.004	2.98(-6)	4.018	3.65(-6)	4.022
3	50	8.39(-2)		1.09(-1)		1.08(-5)		1.33(-5)	
	100	4.22(-2)	0.992	5.44(-2)	1.004	1.64(-7)	6.045	2.00(-7)	6.052
	200	2.11(-2)	0.996	2.72(-2)	1.001	2.52(-9)	6.023	3.06(-9)	6.027
4	50	8.40(-2)		1.09(-1)		1.70(-7)		2.09(-7)	
	100	4.22(-2)	0.993	5.44(-2)	1.005	6.45(-10)	8.044	7.86(-10)	8.051
	200	2.11(-2)	0.995	2.72(-2)	1.000	6.72(-12)	6.585	8.36(-12)	6.555
8	50	8.39(-2)		1.09(-1)		4.34(-12)		5.52(-12)	
	100	4.22(-2)	0.993	5.44(-2)	1.003	4.36(-12)	-0.005	5.50(-12)	0.004
	200	2.11(-2)	0.996	2.72(-2)	1.001	4.32(-12)	0.012	5.44(-12)	0.018

The number of FPs in the IDM is always set to $m = M$, and sufficiently small time increment is used in all cases ($\Delta t = \pi \times 10^{-4}$). Here 1.56(-1) denotes 1.56×10^{-1} .

In order to examine the full order of accuracy for $M = 8$, we consider the same problem with a large $\omega \approx 36.48810769772309$. The numerical results are given in Table 2. It is obvious that for this highly under-sampling case, all schemes produce large errors due to the numerical dispersion. When the grid resolution is refined, the designed theoretical order of convergence of each scheme can be reached. In particular, we achieve 16th order of accuracy by using $M = 8$ for this high-frequency problem with inhomogeneous media.

It is interesting to compare the performance of present method at different orders, which are obtained by simply changing the parameter m (and M) in the present approach. As shown in Table 1, with about 3.6

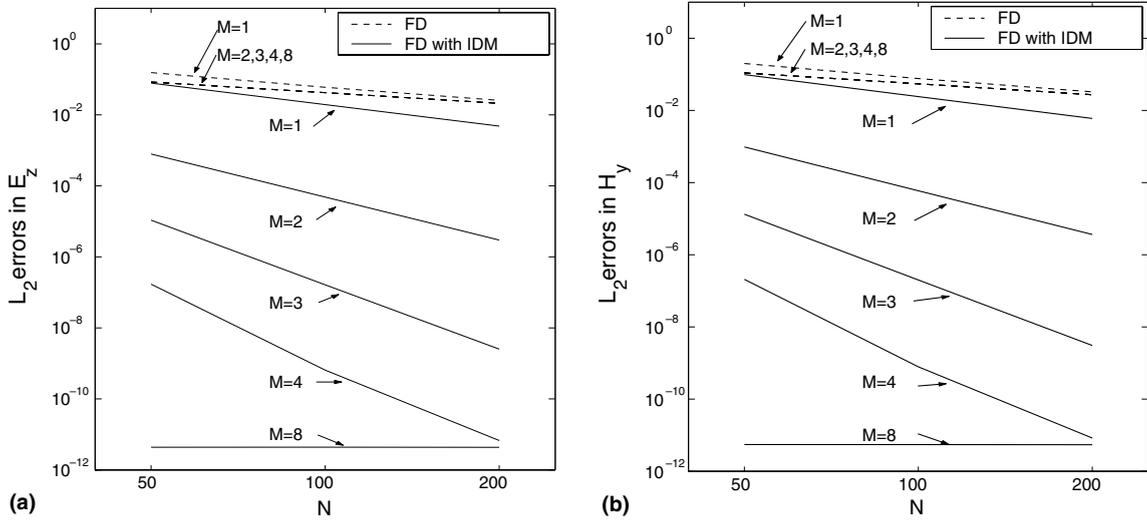


Fig. 3. Log–log plots of the L_2 errors by using different M values for the 1D model problem. Numerical results of the high-order FDTD with or without the IDM are shown. (a) Errors in E_z ; (b) errors in H_y .

Table 2

The L_2 errors in E_z and H_y at $t = \pi/2$ for the 1D model problem with high frequency parameter setting ($\omega \approx 36.48810769772309$)

M	N	E_z		H_y	
		Error	Rate	Error	Rate
1	50	2.00(+0)		2.55(+0)	
	100	2.51(+0)	-0.327	3.32(+0)	-0.381
	200	8.26(-1)	1.605	1.09(+0)	1.600
2	50	2.27(+0)		2.81(+0)	
	100	4.01(-1)	2.499	5.40(-1)	2.376
	200	2.61(-2)	3.939	3.50(-2)	3.949
3	50	2.27(+0)		3.08(+0)	
	100	6.56(-2)	5.116	8.87(-2)	5.121
	200	1.12(-3)	5.872	1.51(-3)	5.879
4	50	1.57(+0)		2.15(+0)	
	100	1.23(-2)	6.999	1.66(-2)	7.015
	200	5.56(-5)	7.787	7.49(-5)	7.794
8	50	3.07(-1)		4.22(-1)	
	100	2.64(-5)	13.505	3.59(-5)	13.522
	200	6.06(-10)	15.412	8.19(-10)	15.419

The FDTD method with the IDM modeling ($m = M$) is used, and $\Delta t = \pi/5 \times 10^{-4}$.

times more CPU time, our 16th order scheme can be over 10^7 times more accurate than our 4th order scheme, on the grid of $N = 50$. For 1D computation, when mesh size is double for the fourth-order scheme, its accuracy is 16 times better, while its CPU time is two times larger. Consequently, our 16th order scheme could be about 32 times more efficient than the fourth-order one under this circumstance. The similar pattern could also be observed in the high-frequency setting when $N = 200$ (Table 2). It is also important to

note that how sensitive a computational scheme is to the change in frequency parameter. Had the frequency parameter ω been set at a 20 times higher in Table 2, it could have been only the highest order scheme that delivers anything meaningful at the present grid spacing. These findings reveal the benefit of using a higher-order scheme whenever it is possible.

We then consider a quasi-1D model problem studied in [45] with $\epsilon_1 = 1$ and $\epsilon_2 = 2.25$. This quasi-1D problem is degenerated from a 2D problem in which the material interface is at some angle with respect to the Cartesian grid. The important feature of this quasi-1D problem is that the individual electromagnetic field component may become discontinuous at the material interface [45]. Such a phenomenon usually appears only in higher dimensional cases. Following [45], we denote θ as the angle between the unit vector normal to the interface, $\hat{\mathbf{n}}$, and the x -axis. Now, the E_z field is discontinuous across the interface, satisfying the relation [45]

$$E_z^{(0)}(\zeta^-, t) = \frac{\epsilon_2}{1 + (\epsilon_2 - 1) \cos^2 \theta} E_z^{(0)}(\zeta^+, t). \quad (26)$$

In other words, the zeroth-order jump condition for the quasi-1D model problem changes to Eq. (26). Consequently, the general p th-order jump condition becomes

$$A_1^p \mathbf{q}^{(p)}(\zeta^-, t) = A' A_2^p \mathbf{q}^{(p)}(\zeta^+, t), \quad A' = \begin{bmatrix} \epsilon_2 / (1 + (\epsilon_2 - 1) \cos^2 \theta) & 0 \\ 0 & 1 \end{bmatrix}$$

for $p = 0, 1, 2, \dots$. The exact solution of this model problem is also given by Eqs. (24) and (25) [45]. While ω is found as the solution to equation

$$\tan(\omega) = \frac{-\sqrt{\epsilon_2} \tan(\sqrt{\epsilon_2} \omega)}{1 + (\epsilon_2 - 1) \cos^2 \theta}.$$

An example plot of the solution at $t = 0$ with $\omega \approx 5.05589071456588$ is shown in Fig. 4. A small jump in the field E_z is clearly displayed in the figure, as suggested in Eq. (26). It is well known that the approximation of discontinuous variables is numerically challenging. For this difficulty problem, it has been found in [45] that the Yee scheme exhibits local divergence and losses of global convergence, due to its incapability of correctly modeling the discontinuous field components.

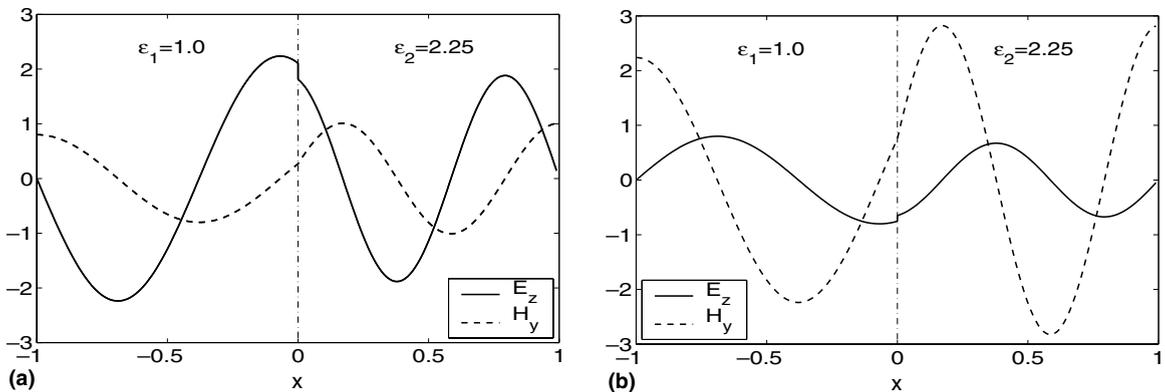


Fig. 4. Solution of the quasi-1D model problem at $t = 0$. (a) Real part; (b) imaginary part.

The numerical results of the quasi-1D model problem with $\omega \approx 5.05589071456588$ are listed in Table 3. It can be observed from Table 3 that the standard FDTD approximation for this problem is globally nonconvergent, no matter how large M is (see also Fig. 5). The loss of global convergence is due to the lack of properly enforced jump conditions on the field components [45]. On the other hand, it can be seen clearly

Table 3
The L_2 errors in E_z and H_y at $t = \pi$ for the quasi-1D model problem with $\omega \approx 5.05589071456588$ by using $\Delta t = \pi \times 10^{-4}$

M	N	FDTD				FDTD with IDM ($m = M$)			
		E_z		H_y		E_z		H_y	
		Error	Rate	Error	Rate	Error	Rate	Error	Rate
1	50	2.32(-1)		2.67(-1)		8.23(-2)		9.97(-2)	
	100	1.99(-1)	0.220	2.29(-1)	0.222	2.04(-2)	2.013	2.45(-2)	2.024
	200	1.98(-1)	0.010	2.31(-1)	-0.014	5.07(-3)	2.007	6.08(-3)	2.012
2	50	2.04(-1)		2.36(-1)		8.15(-4)		9.74(-4)	
	100	1.97(-1)	0.048	2.30(-1)	0.038	4.99(-5)	4.030	5.93(-5)	4.038
	200	1.99(-1)	-0.013	2.32(-1)	-0.015	3.08(-6)	4.017	3.65(-6)	4.021
3	50	2.03(-1)		2.36(-1)		1.10(-5)		1.31(-5)	
	100	1.97(-1)	0.040	2.30(-1)	0.038	1.67(-7)	6.043	1.98(-7)	6.050
	200	1.99(-1)	-0.011	2.32(-1)	-0.015	2.58(-9)	6.022	3.04(-9)	6.026
4	50	2.03(-1)		2.36(-1)		1.73(-7)		2.05(-7)	
	100	1.98(-1)	0.036	2.30(-1)	0.038	6.55(-10)	8.041	7.73(-10)	8.048
	200	1.99(-1)	-0.010	2.32(-1)	-0.015	7.07(-12)	6.535	8.52(-12)	6.504
8	50	2.03(-1)		2.36(-1)		4.65(-12)		5.73(-12)	
	100	1.98(-1)	0.036	2.30(-1)	0.038	4.67(-12)	-0.006	5.72(-12)	0.003
	200	1.99(-1)	-0.010	2.32(-1)	-0.015	4.63(-12)	0.013	5.65(-12)	0.017

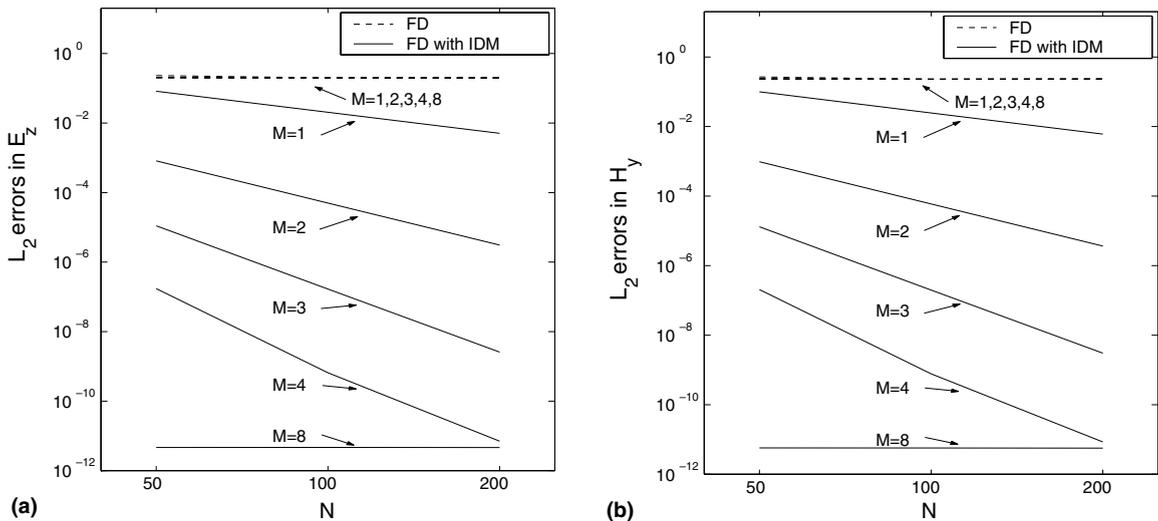


Fig. 5. Log-log plots of the L_2 errors by using different M values for the quasi-1D model problem. Numerical results of the high-order FDTD with or without the IDM are shown. (a) Errors in E_z ; (b) errors in H_y .

from Table 3 and Fig. 5 that the numerical results of the FDTD approach with the IDM for the quasi-1D problem are as good as those for the 1D model problem given in Table 1 and Fig. 3. Both expected convergence rates and extremely high accuracy are restored after using the IDM method to correctly model the discontinuous jump conditions.

Since, in Table 3, the numerical accuracy of $M = 8$ is also limited by the machine precision, we similarly consider a high-frequency wave study for the present quasi-1D model problem. By setting $\omega \approx 36.47181725046381$, the numerical results are listed in Table 4. Similar findings as in Table 2 are obtained. Moreover, a comparison between different orders in this case indicates the same pattern: the 16th order scheme can be over 10^7 times more accurate and about 35 times more efficient than the fourth-order scheme.

Although the IDM scheme performs very well together with the FDTD methods for small M , it is found in our numerical experiments that the IDM scheme suffers from certain problems when M becomes larger, see Table 5. Quite large $m = M$ values are considered in Table 5 for the FDTD schemes

Table 4

The L_2 errors in E_z and H_y at $t = \pi$ for the quasi-1D model problem with high frequency parameter setting ($\omega \approx 36.47181725046381$)

M	N	E_z		H_y	
		Error	Rate	Error	Rate
1	50	2.12(+0)		2.65(+0)	
	100	2.66(+0)	-0.325	3.42(+0)	-0.369
	200	8.63(-1)	1.619	1.12(+0)	1.612
2	50	2.46(+0)		2.98(+0)	
	100	4.13(-1)	2.572	5.47(-1)	2.446
	200	2.70(-2)	3.939	3.54(-2)	3.949
3	50	2.35(+0)		3.13E(+0)	
	100	6.72(-2)	5.130	8.93(-2)	5.133
	200	1.15(-3)	5.871	1.52(-3)	5.879
4	50	1.61(+0)		2.17(+0)	
	100	1.25(-2)	7.004	1.67(-2)	7.020
	200	5.68(-5)	7.787	7.53(-5)	7.794
8	50	3.12(-1)		4.22(-1)	
	100	2.68(-5)	13.505	3.59(-5)	13.522
	200	6.16(-10)	15.411	8.19(-10)	15.419

The FDTD method with the IDM modeling ($m = M$) is used, and $\Delta t = \pi/5 \times 10^{-4}$.

Table 5

The L_2 errors at $t = \pi$ for two model problems when the number of FPs m is large

Problem	M	IDM			HDM			
		E_z	H_y	CPU	l	E_z	H_y	CPU
1D	16	4.24(-12)	5.35(-12)	0.0240	8	3.93(-12)	4.91(-12)	0.0029
	32	2.16(-5)	2.77(-5)	2.8867	8	7.12(-11)	7.85(-11)	0.0166
Quasi-1D	16	4.61(-12)	5.65(-12)	0.0245	8	5.37(-12)	6.59(-12)	0.0028
	32	4.72(-6)	6.52(-6)	2.9822	8	3.45(-12)	3.52(-12)	0.0167

Here, $\omega \approx 5.07218116182516$ and $\omega \approx 5.05589071456588$ for the 1D and quasi-1D model problem, respectively. The FDTD methods are employed in both the IDM and HDM with $N = 100$, $\Delta t = \pi \times 10^{-4}$, and $m = M$. CPU time in hours is reported.

together with the IDM. It can be observed from Table 5 that the results of both 1D and quasi-1D problems are quite similar. In particular, when $M = 16$, the L_2 errors in E_z and H_y are still very low. However, when $M = 32$, the accuracy level quickly goes down. This is actually due to the Lagrange weights used in the IDM increase exponentially for high-order derivative approximations. It is well known that under such an occasion, the resolution of representation coefficients in the IDM becomes an unstable procedure as m goes to infinity. On the other hand, for a finite but quite large m , like 32, the calculation of exponentially increased coefficients usually introduces huge round-off errors in numerical computations, although the FD weights are properly scaled as suggested in [39] in our computation. By noted this point, it is recommended in [39] that the number of FPs m should be quite small. However, for some special applications involving high-frequency waves, it might be desired to generate as many as hundreds of grid stencils in practice. The extension of the IDM method to these applications is clearly questionable. Meanwhile, the CPU time of the IDM with large m could be quite large, e.g., it could be a few hours for $m = 32$ as shown in Table 5.

3.3. Stability analysis

It is of great interest to explore the use of the IDM method together with other high-order time-domain approaches, due to its excellent performance with the FDTD scheme. Computationally, we just need to simply change the basic differential stencil used in the approximations (18) and (19). However, it is found that for the simple 1D model problem, when an MRTD kernel, which is constructed based on the Battle–Lemarie scaling function [17,18], is used, the entire procedure could be unconditionally unstable, while the MRTD method is free of this instability issue without adopting the IDM. This means that the use of the IDM method may introduce certain instability problems. Therefore, the stability issue of the IDM method demands careful studies. The lack of necessary analysis of stability for the DM methods has also been noted by Hesthaven [4].

For simplicity, the stability issues are dealt with in this section primarily for the 1D model problems. The results can be similarly extended to other 1D problems. We first consider the stability of the present discretization procedure without using the IDM method. It is well known that the general stability condition for solving Maxwell's equations can be given as [4]

$$\Delta t \leq C' \sqrt{\min\{\epsilon\mu\}} \Delta x.$$

The value of the constant C' is determined by both the spatial and temporal discretizations. We can further separate these two sources, by considering a more detailed condition [49]

$$\Delta t \leq \sqrt{\min\{\epsilon\mu\}} \frac{C}{\rho^\infty} \Delta x, \quad (27)$$

where the constant C is related to the time-stepping scheme, while ρ^∞ is induced by the spatial approximation. In particular, ρ^∞ is the limit value of the spectral radius, ρ^S , of the spatial discretization matrix S given in (22), and

$$\rho^S = \max_{j=1,\dots,2N} |\lambda_j^S|,$$

where λ_j^S are eigenvalues of ρ^S . Numerically, it has been found [49] that, when $\Delta x \rightarrow 0$, $\rho^S \rightarrow \rho^\infty$ from below. Usually, ρ^∞ is a product of a factor with \sqrt{d} [49], where $d = 1, 2, 3$ is the dimension number.

We first consider the values of ρ^∞ for different spatial schemes. For the FDTD methods, it is known that we have $\rho_{\text{FDTD},M=1}^\infty = 2\sqrt{d}$ and $\rho_{\text{FDTD},M=2}^\infty = \frac{7}{3}\sqrt{d}$ [49]. Here two subscripts are added to ρ^∞ , in order to identify different spatial discretization schemes. The constants containing in ρ^∞ are clearly determined by

the differential kernel being used. Following the stability analysis considered in the MRTD method [18], it can be easily verified that we actually have

$$\rho^\infty = \sqrt{d} \sum_{j=-M, j \neq 0}^M |w_j|. \quad (28)$$

The analyses of the FDTD methods in [49] are in consistent with Eq. (28), i.e., when $M = 1$, $\sum_{j=-1, j \neq 0}^1 |w_j| = |-1| + |1| = 2$ and when $M = 2$, $\sum_{j=-2, j \neq 0}^2 |w_j| = \frac{1}{24}(|1| + |-27| + |27| + |-1|) = \frac{7}{3}$.

By using the MRTD method, $\sum_{j=-M, j \neq 0}^M |w_j|$ converges to a limit, which is reported to be about $2/0.6371 = 3.1392$, after $M \geq 16$ [18]; see Fig. 6. The stability analysis of the DSC spatial discretization can be similarly analyzed for the LSTD method [24–26]. In Fig. 6, we also plot the stability factor of the LSTD method. It is clear that the factor of the LSTD converges faster than that of the MRTD. The limit value of the LSTD is found to be π , which is the same as that of the Fourier pseudospectral method [60]. It is well known that the Battle–Lemarie scaling function has closed form expression only in the spectral domain. For time-domain computations, the MRTD kernel is constructed via a truncation to an accuracy level of 0.1% [17,18]. We thus believe that if a more accurate truncation is considered in the MRTD method, its stability factor should be closer to π .

We then consider the stability factor C for time-stepping schemes. It is known that we have $C = 2$ for usual staggered Leapfrog time integration [49]. We are interesting to find such a C for the RK4 time integration. To this end, we rewrite Eq. (23) into the form

$$\mathbf{Q}^{n+1} = \left[\mathbf{I} + \frac{\Delta t}{\Delta x} \mathbf{S} + \frac{1}{2} \frac{\Delta t^2}{\Delta x^2} \mathbf{S}^2 + \frac{1}{6} \frac{\Delta t^3}{\Delta x^3} \mathbf{S}^3 + \frac{1}{24} \frac{\Delta t^4}{\Delta x^4} \mathbf{S}^4 \right] \mathbf{Q}^n. \quad (29)$$

It is clear that this procedure is stable provided that

$$\left| 1 + \frac{\Delta t}{\Delta x} \lambda_j^S + \frac{1}{2} \frac{\Delta t^2}{\Delta x^2} (\lambda_j^S)^2 + \frac{1}{6} \frac{\Delta t^3}{\Delta x^3} (\lambda_j^S)^3 + \frac{1}{24} \frac{\Delta t^4}{\Delta x^4} (\lambda_j^S)^4 \right| \leq 1 \quad (30)$$

for any $j = 1, \dots, 2N$ and all Δx . Moreover, since the differential matrix of central approximations to linear hyperbolic systems is antisymmetric, eigenvalues λ_j^S of the differential matrix \mathbf{S} would be purely imaginary numbers [60]. The similar observation has been reported in [49] that the real parts of all eigenvalues of \mathbf{S} are

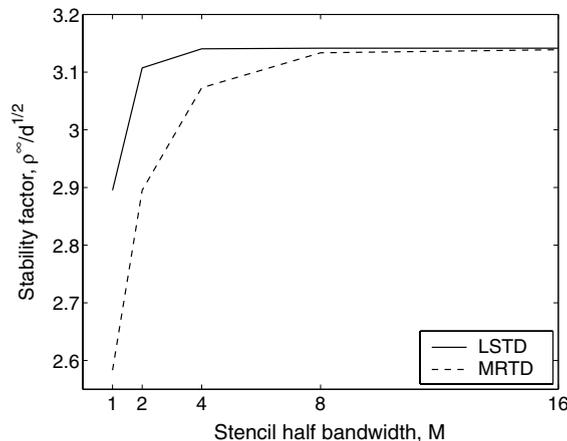


Fig. 6. The plot of the stability parameter ρ^∞ / \sqrt{d} versus the half computational bandwidth M for both LSTD and MRTD methods.

of $O(10^{-16})$ for all Δx . Consequently, it can be easily derived from (30) that when λ_j^S lie on the imaginary axis, the stability factor of RK4 is $C = 2\sqrt{2}$ [45]. This is in consistent with the fact that the stability region of RK4 extends from the origin a distance of $2\sqrt{2}$ up and down the imaginary axis [60]. Therefore, when the IDM is not carried out, the entire discretization procedure (29) is stable when

$$\Delta t \leq \sqrt{\min\{\epsilon\mu\}} \frac{2\sqrt{2}}{\rho^\infty} \Delta x. \quad (31)$$

In our numerical experiments, the stability constraint (31) has been found to be in excellent agreement with numerical results.

Following the same procedure, we then consider the possible influence of the IDM method to the stability of the entire discretization. For the FDTD approximations, the stability conditions, after the IDM is carried out, are found to be almost the same as those without the IDM modeling, as shown in Fig. 7. In particular, for all tested M values (1, 2, 4, 8, 12, and 16) and all tested Δx , the real parts of the eigenvalues of the spatial discretization matrix \mathbf{S} are all smaller than 1.0×10^{-16} . After applying the IDM method, the spectral radius of \mathbf{S} , i.e., ρ^S , is usually a little bit smaller than that of the standard FDTD; see Fig. 7. However, such differences are of extremely small magnitude so that they essentially do not alter the numerical stability constraint. Moreover, when Δx approaches to zero, such differences quickly vanish, and both spectral radii converge to the same ρ^∞ . Therefore, the stability condition (31) holds for the FDTD approximation with the IDM modeling, which has also been verified numerically.

The situation is different for the MRTD method [17,18]. Without the IDM, the MRTD method is conditionally stable with the stability constraint being Eq. (31). However, when the IDM is conducted, the MRTD approximation is found to be unconditionally unstable by employing a commonly used half stencil length M , i.e., $8 \leq M \leq 16$. By taking a more detailed investigation, it is found that many eigenvalues of the spatial discretization matrix \mathbf{S} have non-zero real parts now. Furthermore, although the maximum value of imaginary parts of λ_j^S is still very close to $\sum_{j=-M, j \neq 0}^M |w_j|$, the spectral radius of \mathbf{S} , ρ^S , now is solely fixed by the real parts of λ_j^S . In particular, the dominant eigenvalue of \mathbf{S} takes the form $\pm \rho^S + 0i$, with $\rho^S \gg \sum_{j=-M, j \neq 0}^M |w_j|$. It is clear from Fig. 8 that ρ^S becomes a very large number after the IDM is carried out, and it also converges to certain limit as $\Delta x \rightarrow 0$. It is noted that the stability factor $C = 2\sqrt{2}$ for the RK4 temporal integration is derived under the assumption that the real parts of λ_j^S are all zero. Thus, for the MRTD with the IDM, the stability factor C is no longer $2\sqrt{2}$. For the specific form of the dominant eigenvalue, $\rho^S + 0i$, to satisfy the requirement (30) for all $j = 1, \dots, 2N$ and all Δx , it is equivalent to force

$$\left(1 + \frac{\Delta t}{\Delta x} \rho^S + \frac{1}{2} \frac{\Delta t^2}{\Delta x^2} (\rho^S)^2 + \frac{1}{6} \frac{\Delta t^3}{\Delta x^3} (\rho^S)^3 + \frac{1}{24} \frac{\Delta t^4}{\Delta x^4} (\rho^S)^4 \right) \leq 1$$

for all Δx . Obviously, this is true only if $(\Delta t/\Delta x)\rho^S = 0$. For a concrete discretization, we have $\Delta x > 0$ and $\rho^S > 0$. Then, the MRTD method with the IDM is stable if $\Delta t = 0$, i.e., it is unconditionally unstable.

We further analyze the stability of the LSTD method [24–26] with the IDM. In the DSC spatial approximation of the LSTD method, apart from the different half computational bandwidth M can be freely chosen, there is a parameter r in the DSC algorithm which can be adjusted to deliver higher accuracy for the same M [22]. In practice, one can select the desired DSC parameters M and r according to the nature of the problem under consideration by means of the discrete Fourier analysis [24,54]. In the present study, it is found that by using a quite small r , the stability constraint of the LSTD method could be the same after the IDM is carried out. In Fig. 9, we consider the DSC approximation with $M = 16$. When $r = 2.1$, similar to the FDTD method, the stability condition of the LSTD method essentially remains unchanged after considering the IDM modeling. When we consider a larger $r = 2.2$, the spectral radius ρ^S with the IDM attains a larger value than that without the IDM. Nevertheless, all eigenvalues of \mathbf{S} still have only imaginary parts, so that the stability condition (31) still holds with $C = 2\sqrt{2}$ for the LSTD method with the

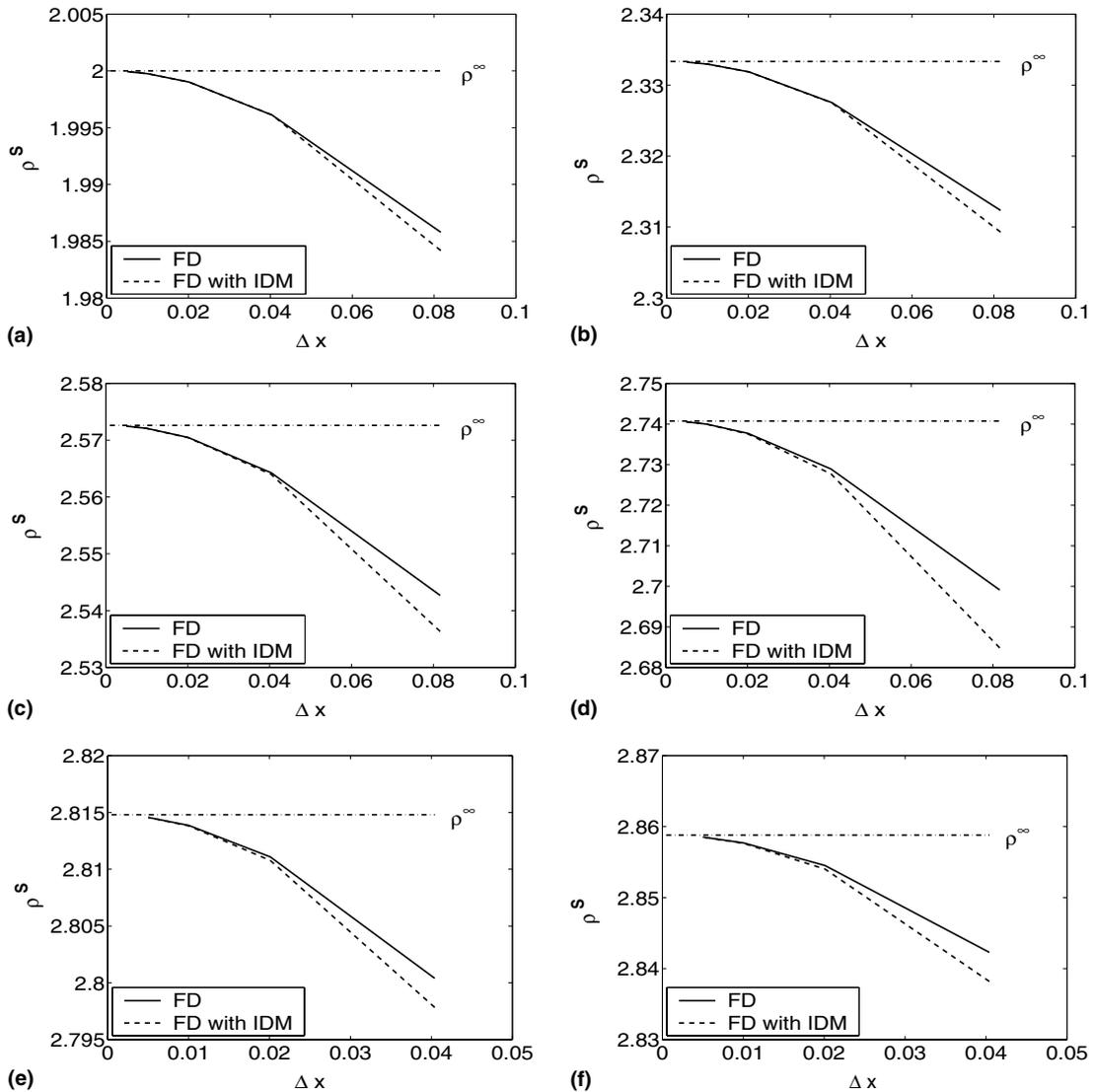


Fig. 7. The spectral radius ρ^S as a function of mesh size for the FDTD schemes. (a) $M = 1$; (b) $M = 2$; (c) $M = 4$; (d) $M = 8$; (e) $M = 12$; (f) $M = 16$. For $M = 1, 2, 4, 8, 12,$ and 16 , we have $\rho^\infty = 2, 2.3333, 2.5726, 2.7408, 2.8148,$ and 2.8588 , respectively.

IDM, except now the analytical value of ρ^∞ is unknown. As can be seen from Fig. 9(b), ρ^∞ becomes larger in the present case. Thus, the stability constraint of the LSTD method becomes more severe after using the IDM. When an even larger r , e.g., $r \geq 2.3$ for $M = 16$, is used, the LSTD method becomes unconditionally unstable. The similar difficult of the MRTD method is observed again herein. In particular, the dominant eigenvalue of \mathbf{S} also takes the form $\pm \rho^S + 0i$, with $\rho^S > \sum_{j=-M, j \neq 0}^M |w_j|$, so that $C \neq 2\sqrt{2}$ but $C = 0$. These stability analyses are in excellent agreement with our numerical experiments. Although a small r could always be used in the LSTD method to avoid the stability problem in electromagnetic applications, these r values are actually smaller than the usual optimal r values of the DSC approximation for a fixed M [24,54]. Thus, the high accuracy and applicability of the IDM method with LSTD are reduced.

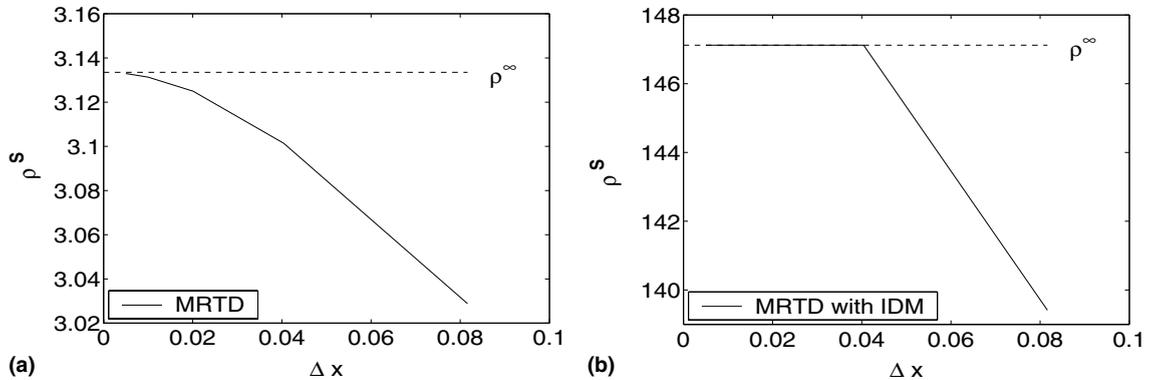


Fig. 8. The spectral radius ρ^S as a function of mesh size for the MRTD scheme with $M = 8$. (a) Without the IDM ($\rho^\infty = 3.1335 = \sum_{j=-8, j \neq 0}^8 |w_j|$); (b) with the IDM ($\rho^\infty \approx 147.1205$).

In summary, the use of the IDM method may introduce additional problems on stability for high-order time-domain Maxwell solvers. The exception of the FDTD method in the present investigation may be due to the fact that the FD weights are harmoniously used in both the IDM modeling and the FDTD approximation. Essentially the instability associated with other high-order time-domain schemes is believed to be linked with the fact that the amplitudes of the Lagrange FD weights used in the IDM increase exponentially as the differentiation order increases. However, the detailed insight of the mechanism of the instability observed in the MRTD and LSTD approaches is still unavailable. The stability issue of the IDM method calls for further studies.

It is noted that the stability analyses in the present study could be also useful to the explicit DM considered in the BPS method [39,40]. The interface modeling in the BPS method is carried out in a post-processing manner, so that the stability of the explicit DM could not be analyzed in a systematic way, as in the present study. Alternatively, by means of the IDM modeling, the present results on stability can be easily extended to the BPS method. In particular, it is expected that the stability constraint in the BPS method would be more severe, because the global discretizations in the BPS method [39,40] are essentially one-sided approximations. Only global FD weights are employed in the BPS method [39,40]. Based on the present analysis, it is also supposed that the problems of instability may occur if other popular basis functions, such as Chebyshev and Legendre polynomials, are utilized. Moreover, the nonuniform grid in these spectral methods introduces additional stability problems.

3.4. Hierarchical derivative matching

It is found in our stability analysis that the direct application of the IDM method might introduce stability problems for the MRTD and LSTD approaches. Moreover, we have shown that the IDM modeling with a large number of FPs suffers from the loss of precision, and could be time-consuming. Furthermore, the difficult of handling mixed derivatives is encountered when the present IDM scheme is generalized to higher dimensions, an issue for which is fully accounted in Section 4. Thus, the IDM method is not robust enough for general electromagnetic computations. Therefore, it is of interest to seek after alternative ways. For this purpose, a hierarchical derivative matching (HDM) method is developed in the present study.

The essential motivation of the HDM method is to bypass the large size algebraic system introduced in high-order jump conditions in the IDM method. Nevertheless, the main philosophy of the IDM method is still inherited in the HDM method. In fact, an IDM with a small number of FPs is carried out

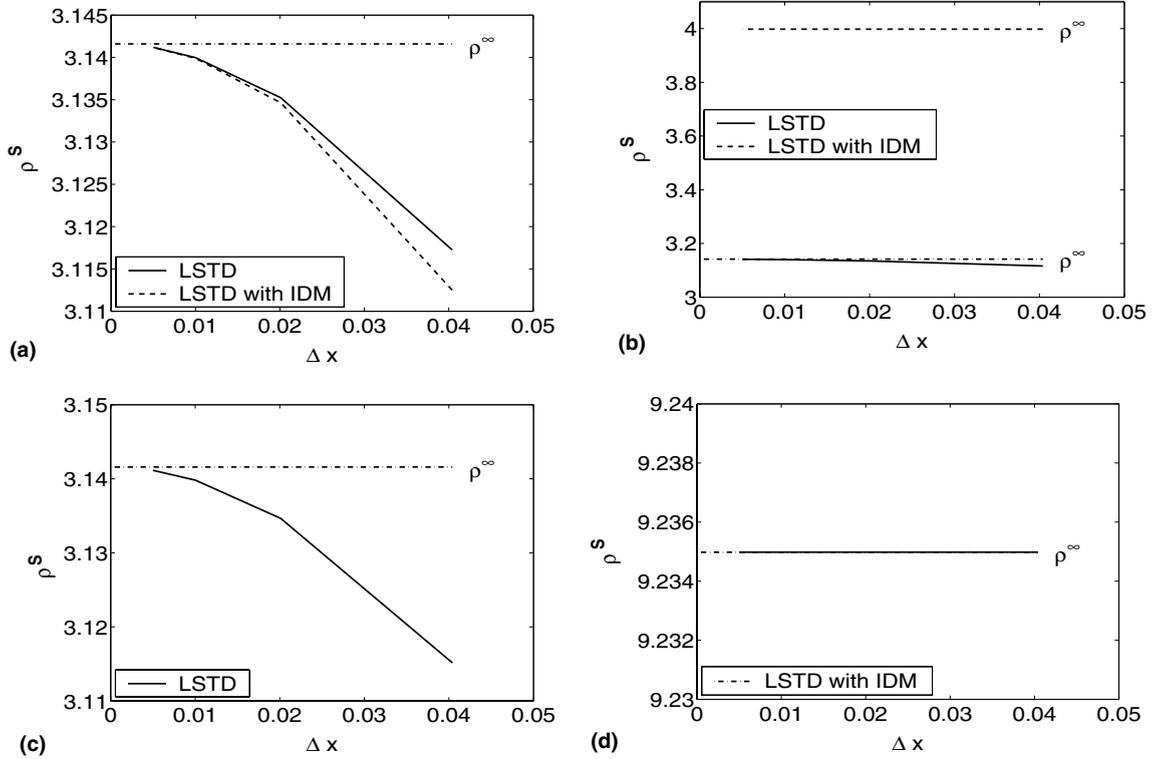


Fig. 9. The spectral radius ρ^s as a function of mesh size for the LSTD scheme with $M = 16$. (a) $r = 2.1$; (b) $r = 2.2$; (c) $r = 2.3$ without the IDM; (d) $r = 2.3$ with the IDM. Without the IDM, the spectral radius converges to $\rho^\infty = \pi$ for all r values. With the IDM, ρ^∞ may attain different values. Here $\rho^\infty = 3.9979$ and 9.2350 , for $r = 2.2$ and $r = 2.3$, respectively.

in the first stage of the HDM method. The basic consideration of the HDM method is that we dissociate the matrix size involved in the IDM method and the total number of FPs. The latter is usually determined by the applications, such as the stencil length $2M$ being used in a specific time-domain solver. To achieve more general applicability, we always maintain the translation invariance property of the basic differential stencil, so that the total number of FPs, $2m$, is still be equal to $2M$. In the HDM method, we introduce a new integer parameter l , which is usually small and satisfies $1 \leq l \leq m$, and require the dimensional size involved in the IDM method at the first stage to be $2l$, see Fig. 10. In other words, we first establish the representation coefficients via solving an algebraic system of the IDM method for $2l$ nearest neighboring FPs, i.e., $f_{m-l+1}, \dots, f_{m+l}$ in Fig. 10. We then seek for a different procedure to generate the representation coefficients for the rest $2m - 2l$ FPs, f_1, \dots, f_{m-l} and f_{m+l+1}, \dots, f_{2m} . It is obviously crucial that the high accuracy of the IDM method should not be destroyed in the present HDM modeling.

In the HDM method, the determination of representation coefficients for the rest $2m - 2l$ FPs is carried out in a recursive manner. In each recursive step, we determine only one FP at each side of the material interface. For example, we first consider the determination for f_{m-l} and f_{m+l+1} , then for f_{m-l-1} and f_{m+l+2} , and so on. Totally $m - l$ recursive steps are taken in this dynamic procedure. At each recursive step, the FPs with known representation coefficients are referred to as master FPs, while the two new FPs are called subordinate FPs. A pseudoprocedure of the HDM modeling is given as follows:

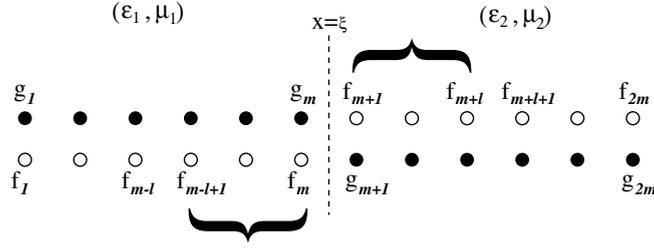


Fig. 10. Illustration of FPs and notation used in the HDM method. The $2l$ FPs inside the braces are involved in the first step of the HDM method, i.e., the ordinary IDM modeling.

A pseudo-procedure of the HDM method

Stage 1. Carry out the IDM modeling for the interior $2l$ FPs, $f_{m-l+1}, \dots, f_{m+l}$.

Stage 2. The recursion

DO $k = l + 1, m$

Known values: representation coefficients of $2(k - 1)$ master FPs, $f_{m-k+2}, \dots, f_{m+k-1}$

Unknowns: representation coefficients of 2 subordinate FPs, f_{m-k+1} and f_{m+k}

Use known values and low-order jump conditions to determine unknowns

END DO

The remaining issue of the HDM method is how to determine the representation coefficients of two subordinate FPs. Without loss of generality, we take a simple case with $m = l + 1$ as example. Suppose that the coefficients $r_{i,j}$ for $i, j = 2, \dots, 2m - 1$ in Eq. (11) are known from the previous IDM or HDM modeling. We need to determine the rest coefficients in matrix $R = (r_{i,j})$, for $i, j = 1, \dots, 2m$. We first set $r_{i,1} = r_{i,2m} = 0$, for $i = 2, \dots, 2m - 1$. In other words, the $2l$ master FPs f_2, \dots, f_{2m-1} are represented by the interior $2l$ grid points g_2, \dots, g_{2m-1} only. By doing this, the accuracy of the representation coefficients for the master FPs remains the same and no additional complexity involves.

We then consider using all grid points g_1, \dots, g_{2m} to represent two subordinate FPs f_1 and f_{2m} . Since we have only two unknown FPs now, we can only consider two low-order jump conditions, such as zero- and first-order ones. Take H_y as an example, these two jump conditions are given as

$$H_y(\zeta^-) = H_y(\zeta^+), \quad \frac{1}{\epsilon_1} H_y^{(1)}(\zeta^-) = \frac{1}{\epsilon_2} H_y^{(1)}(\zeta^+). \tag{32}$$

The FD approximations to Eqs. (32) on both fictitious and grid points are

$$\sum_{i=1}^m w_{0,i} g_i + \sum_{i=m+1}^{2m} w_{0,i} f_i = \sum_{i=1}^m w_{0,i} f_i + \sum_{i=m+1}^{2m} w_{0,i} g_i, \tag{33}$$

$$\frac{1}{\epsilon_1} \left[\sum_{i=1}^m w_{1,i} g_i + \sum_{i=m+1}^{2m} w_{1,i} f_i \right] = \frac{1}{\epsilon_2} \left[\sum_{i=1}^m w_{1,i} f_i + \sum_{i=m+1}^{2m} w_{1,i} g_i \right]. \tag{34}$$

Similarly, we symbolically substitute $f_i = R_i G$ and $g_i = I_i G$ into Eqs. (33) and (34) to attain

$$\sum_{i=1}^m w_{0,i} I_i^T + \sum_{i=m+1}^{2m} w_{0,i} R_i^T = \sum_{i=1}^m w_{0,i} R_i^T + \sum_{i=m+1}^{2m} w_{0,i} I_i^T, \tag{35}$$

$$\frac{1}{\epsilon_1} \left[\sum_{i=1}^m w_{1,i} I_i^T + \sum_{i=m+1}^{2m} w_{1,i} R_i^T \right] = \frac{1}{\epsilon_2} \left[\sum_{i=1}^m w_{1,i} R_i^T + \sum_{i=m+1}^{2m} w_{1,i} I_i^T \right]. \quad (36)$$

Unlike the previous case, now R_i of $2l$ master FPs f_2, \dots, f_{2m-1} are known. Thus, there are only two unknowns in Eqs. (35) and (36): R_1 and R_{2m} . Therefore, without resorting to a numerical algebraic solver, R_1^T and R_{2m}^T can be directly resolved

$$R_1^T = \frac{\frac{1}{\epsilon_2} w_{1,2m} V_1 - w_{0,2m} V_2}{\frac{1}{\epsilon_2} w_{1,2m} w_{0,1} - \frac{1}{\epsilon_1} w_{1,1} w_{0,2m}} \quad R_{2m}^T = \frac{w_{0,1} R_1^T - V_1}{w_{0,2m}},$$

$$V_1 = \sum_{i=1}^m w_{0,i} I_i^T + \sum_{i=m+1}^{2m-1} w_{0,i} R_i^T - \sum_{i=2}^m w_{0,i} R_i^T - \sum_{i=m+1}^{2m} w_{0,i} I_i^T,$$

$$V_2 = \frac{1}{\epsilon_1} \left[\sum_{i=1}^m w_{1,i} I_i^T + \sum_{i=m+1}^{2m-1} w_{1,i} R_i^T \right] - \frac{1}{\epsilon_2} \left[\sum_{i=2}^m w_{1,i} R_i^T + \sum_{i=m+1}^{2m} w_{1,i} I_i^T \right].$$

For cases with $m > l + 1$, we can generate coefficients R_i of two subordinate FPs each time, and repeat this procedure as often as needed by means of an index replacement.

It is interesting to investigate the accuracy level of the HDM method. In the first stage of the HDM method, the accuracy of the IDM modeling is clearly determined by the parameter l , i.e., it is of $(2l)$ th-order. In the following HDM recursion, at the first step, the accuracy of the FD approximations considered in Eqs. (33) and (34) is of $(2l + 2)$ th-order for the two new subordinate FPs. Moreover, since the representation coefficients for the interior $2l$ master FPs are unchangingly shifted. The representation accuracy of these $2l$ master FPs is maintained as $(2l)$ th-order, so that the final accuracy of entire HDM modeling remains as $(2l)$ th-order. The same conclusion can be drawn for the rest recursive steps. Therefore, the numerical accuracy of the HDM method is solely determined by the parameter l . In other words, the HDM method is capable of maintaining the high accuracy of the IDM method when introducing more FPs.

Apart from the formal order of accuracy, the numerical computations are usually subject to many other error sources, such as the round-off error due to the finite precision limit. It has been found that when m is large, the IDM method may lose the precision, owing to the round-off error. However, for the HDM method, the FD approximations to very high-order derivatives are avoided so that the problem of round-off error accumulations is well under controlled. Consequently, the HDM method is well suited for applying to problems requiring a large m . Meanwhile, since the resolution of a big algebraic system is not required in the HDM method, the HDM method usually is much faster than the IDM method to generate a large number FPs. For example, a few hours are required for the IDM method when $m = 32$, while for the HDM method by using $m = 32$ and $l = 8$, 1 or 2 min CPU time would be enough. Therefore, the HDM method is an efficient means for enforcing jump conditions to high-order accuracy.

As the stability of IDM method was one of concerns for the motivation of the HDM method, we examine the stability of the HDM and see whether there is any improvement in the stability. Similar to the IDM method, the HDM method is found to be free of additional stability problem when applying to the high-order FDTD methods. It is of great interest to investigate the stability of the MRTD and LSTD approaches when the HDM modeling is considered. It is found from our analysis that the MRTD method with the HDM could be conditionally stable when l is small, see Fig. 11. In particular, after the HDM modeling, the MRTD method can still have the same stability condition when $l \leq 3$, for $8 \leq M \leq 16$. However, when $l \geq 4$, the scheme becomes unconditionally unstable again.

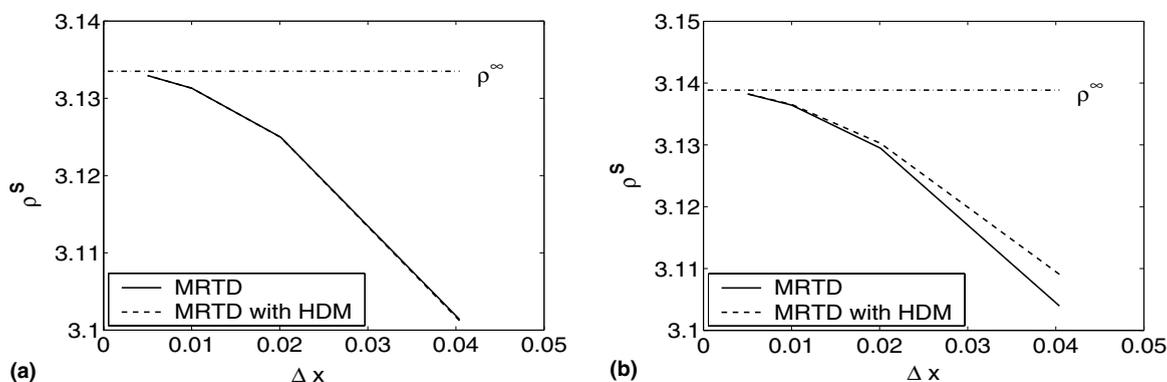


Fig. 11. The spectral radius ρ^S as a function of mesh size for the MRTD scheme. (a) $M = 8$ and $l = 3$ ($\rho^\infty = 3.1335 = \sum_{j=-8, j \neq 0}^8 |w_j|$); (b) $M = 16$ and $l = 3$ ($\rho^\infty = 3.1386 = \sum_{j=-16, j \neq 0}^{16} |w_j|$).

Similarly, the LSTD method becomes more stable with the HDM than with the IDM. In Fig. 12, the spectral radius for the LSTD with $M = 16$ and different r values is shown. For each parameter r , the maximum l of the HDM method which produces stable approximation is considered. Recall that in Fig. 9, when a larger r is used for a fixed M , the chance of the LSTD method with the IDM to be unstable could be larger. The similar finding is still valid for the LSTD method with the HDM, if we consider r being increased with fixed M and l . On the other hand, similar to the MRTD method with the HDM, it is found in our studies that the possibility of the LSTD method with the HDM to be stable is increased if a smaller l is used for given M and r . Therefore, for a fixed M , when r increases, in order to maintain the stability of the entire approximation, a smaller l has to be employed, as can be seen clearly from Fig. 12. It should be emphasized that for $M = m = 16$, the parameter r could be even larger than 2.8, if a smaller l is used. In fact, the range $2.3 \leq r \leq 2.8$ is sufficiently large for the purpose of selecting optimal r values of the DSC algorithm with $M = 16$ for scientific computing [24,54]. Similar findings also hold for different M values. It is also noted that the situation in which the approximation is still conditionally stable, but with a more severe constraint, does not appear in the LSTD method with the HDM. In general, one can draw the conclusion that the time-domain solvers of Maxwell's equations with the HDM method are usually more stable than those with the IDM method.

The studies on the accuracy and stability of the HDM method also reveal rules of selecting the parameter l in the HDM modeling. The general guideline is that, when a higher accuracy is expected, a larger l should be chosen, while when a more stable scheme is required, a smaller l could be a better choice. In real world applications, one should properly choose l in the HDM modeling according to the nature of the problem under consideration, and the time-domain solver being employed.

In comparing with the IDM, the HDM method is more efficient and stable, thus more robust for applications in CEM. Nevertheless, due to a small l is typically employed, the order of accuracy of the HDM method is usually lower than that of the IDM method. A recursive procedure is introduced in the HDM method to increase the applicability of the IDM method. Alternatively, we note that using different iteration schemes characterized by the matrix sizes is another possible way to resolve function values at subordinate FPs. As well, an iteration procedure might even improve the accuracy of the HDM method. However, this has not been tested yet.

In summary, both the IDM and HDM modelings have their own merits and limitations. Nevertheless, the IDM method can be regarded as a special case of the HDM method with $m = l$. In fact, by selecting the parameter l , one can freely choose between two modelings without any need to change one's computer code, while the selection of accuracy and stability should be guided by the nature of the problem under

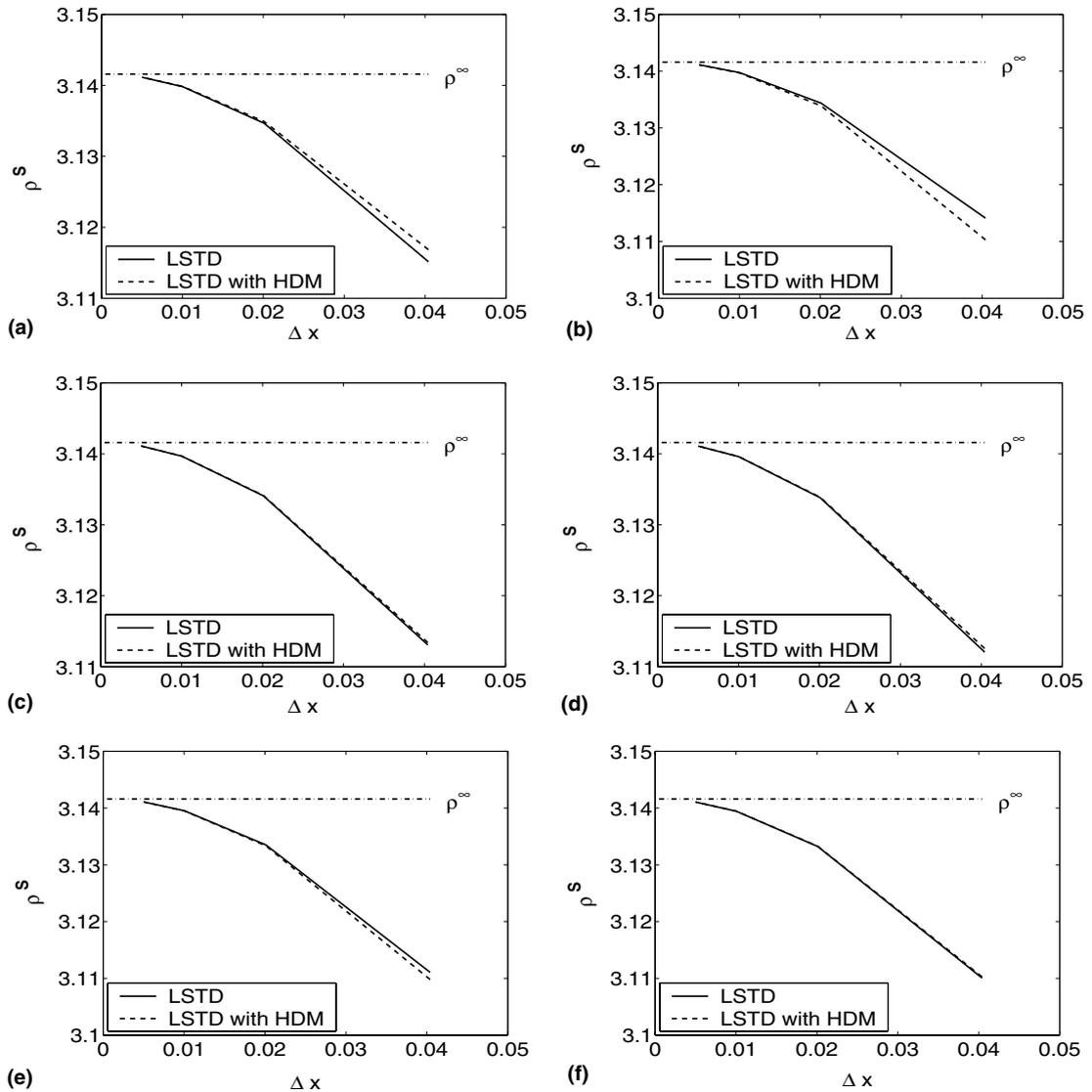


Fig. 12. The spectral radius ρ^S as a function of mesh size for the LSTD scheme with $M = 16$. (a) $r = 2.3$ and $l = 10$; (b) $r = 2.4$ and $l = 9$; (c) $r = 2.5$ and $l = 7$; (d) $r = 2.6$ and $l = 7$; (e) $r = 2.7$ and $l = 6$; (f) $r = 2.8$ and $l = 5$. In all cases, we have $\rho^\infty = \pi$.

study. Thus, without loss of the generality, we could refer to both HDM and IDM modelings as the HDM method.

3.5. Further numerical tests

We explore the numerical performance of the HDM method for CEM in this subsection. Two model problems studied in Section 3.2 are investigated by using the HDM method with a large number of FPs. The results of the HDM method for cases with large $m = M$ are given in Table 5. It is clear from table that

the HDM method performs very well for cases with a large number of FPs, in terms of both efficiency and accuracy, for both the 1D and quasi-1D model problems, as expected.

Our stability analyses demonstrate that the MRTD with the IDM is always an unstable procedure, while the MRTD with the HDM could be a stable scheme. Thus, by using the HDM modeling, the MRTD method could be applicable to inhomogeneous electromagnetic computations to achieve high accuracy. We thus consider two model problems by using both the MRTD and LSTD methods together with the HDM, see Table 6. For the 1D model problem, it can be seen from the table that after using the HDM method, the MRTD results generally become more accurate. Significant improvement in accuracy is also observed for the LSTD method. Furthermore, for both cases with and without the HDM method, the LSTD results are much more accurate than those of the MRTD. This finding is consistent with outcomes presented in [26].

For the quasi-1D model problem, however, it is found that the stability condition of both time-domain approaches is generally more severe than that for the 1D model problem. This is probably owing to the discontinuous nature of the solution. By using the MRTD method with the HDM, the entire approximation is stable only when $l = 1$. Consequently, the accuracy of the MRTD method with the HDM is quite low. Hence, a fine grid resolution with $N = 200$ is employed in the MRTD method for the quasi-1D model problem in Table 6. It can be seen from Table 6 that by using $M = 16$, the MRTD results become much better when the HDM modeling is carried out. However, the MRTD results with $M = 8$ are all quite inaccurate no matter whether the HDM is applied or not. For the same quasi-1D model problem, by using the LSTD method with the HDM, extremely accurate results can be obtained again with $N = 100$. In fact, the results of the LSTD method with the HDM for the quasi-1D model problem are quite close to those for the 1D model problem, albeit now the scheme has a slightly more severe stability constraint. In general, the numerical studies considered in Table 6 well demonstrate the robustness of the HDM method for electromagnetic applications.

Before considering the extension of higher dimensions, it is commented that there is a family of DM schemes that can be developed to generalize the FDTD for handling material interfaces. Essentially, this family of schemes makes use of different numbers of FPs to achieve a given order of accuracy, and for a given number of FPs, complements by appropriate one-sided FD approximation of derivatives near the interface. Therefore, the first scheme of the family uses only the zeroth order DM condition, fully one-sided approximations and no FP, whereas the last scheme is the present HDM method which uses the largest number of FPs as well as largest number of DM conditions and no one-sided approximation. It is expected

Table 6
The L_2 errors at $t = \pi$ by using both the MRTD and LSTD approaches together with the HDM method

Scheme	Problem	M	r	Without HDM		With HDM		
				E_z	H_y	l	E_z	H_y
MRTD	1D	8	–	2.84(–1)	3.62(–1)	3	2.43(–1)	3.08(–1)
		16	–	4.50(–2)	5.80(–2)	3	2.87(–3)	3.74(–3)
	Quasi-1D	8	–	3.17(–1)	3.79(–1)	1	3.65(–1)	4.41(–1)
		16	–	1.97(–1)	2.29(–1)	1	7.67(–2)	9.12(–2)
LSTD	1D	8	1.7	4.21(–2)	5.43(–2)	3	2.87(–5)	3.48(–5)
		16	2.4	4.22(–2)	5.44(–2)	6	4.97(–10)	6.57(–10)
	Quasi-1D	8	1.7	1.97(–1)	2.30(–1)	3	2.91(–5)	3.41(–5)
		16	2.4	1.98(–1)	2.30(–1)	6	5.90(–10)	7.54(–10)

Here, $\omega \approx 5.07218116182516$ and $\omega \approx 5.05589071456588$ for the 1D and quasi-1D model problem, respectively. For the LSTD method, conventionally used parameter values of r are employed. Here $\Delta t = \pi \times 10^{-4}$ and $m = M$. A fine resolution $N = 200$ is employed in the MRTD method for the quasi-1D model problem. In other cases, a normal resolution $N = 100$ is used.

that the numerical stability and robustness of this family of schemes vary from scheme to scheme. The flexibility of being implemented in 3D modeling is another criterion for accessing these schemes. Obviously, it is both interesting and important to systematically investigate this family of methods in the future.

4. Beyond one-dimension

We explore the use of the HDM modeling for 2D electromagnetic applications in this section. First, the complexity and difficulty of DM modeling in 2D are analyzed in detail. The use of central FD approximations for 2D DM is explored and a simple quasi-fourth-order scheme is then introduced. A general 2D HDM method is presented. For simplicity, the basic time-domain solver is fixed to be the standard FDTD method in this section, although the proposed DM methods may also be applied to other high accuracy time-domain approaches. The idea underlying the present 2D studies in this section could be similarly extended to a general 3D scenario.

4.1. 2D considerations

In this section, we consider the 2D TM equation (2) for (H_x, H_y, E_z) in the vector form

$$\frac{\partial \mathbf{q}}{\partial t} = A \frac{\partial \mathbf{q}}{\partial x} + B \frac{\partial \mathbf{q}}{\partial y}, \quad (37)$$

$$\mathbf{q} = \begin{bmatrix} H_x \\ H_y \\ E_z \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1/\mu \\ 0 & 1/\epsilon & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & -1/\mu \\ 0 & 0 & 0 \\ -1/\epsilon & 0 & 0 \end{bmatrix}. \quad (38)$$

It is supposed that the media are non-magnetic with $\mu = 1$ and are homogeneous in the y -direction. Similar to the 1D cases, the electric permittivity ϵ is a piecewise constant with two values ϵ_1 and ϵ_2 in the x -direction. Again, we assume that the interface is at $x = \xi$. The case that there are multiple interfaces can be similarly treated. The notations for A_1, A_2, B_1 and B_2 can be similarly defined. Following the convention of CEM, a 2D staggered grid is used for the fields E_z, H_x and H_y [24–26].

We first establish physical jump conditions at $x = \xi$. It is worthwhile to note that we have three conditions for three field components in each order of jump conditions. Among them, only two conditions are employed in the DM modeling, because such a modeling is carried out for E_z and H_y only. For the field component H_x , since derivative of H_x with respect to x is not required to be evaluated in the resolution of the TM equation (2), it is not necessary to conduct a DM modeling for H_x .

The zeroth-order jump condition states that the fields are continuous across the interface

$$\mathbf{q}(\xi^-, t) = \mathbf{q}(\xi^+, t). \quad (39)$$

Similar to the 1D case, the first-order time derivative of \mathbf{q} is also continuous across the interface. Thus, we have the first-order jump condition

$$\left(A_1 \frac{\partial}{\partial x} + B_1 \frac{\partial}{\partial y} \right) \mathbf{q}(\xi^-, t) = \left(A_2 \frac{\partial}{\partial x} + B_2 \frac{\partial}{\partial y} \right) \mathbf{q}(\xi^+, t). \quad (40)$$

More insights can be gained if we rewrite condition (40) into its complete form:

$$\frac{\partial}{\partial y} E_z(\xi^-, t) = \frac{\partial}{\partial y} E_z(\xi^+, t), \quad (41)$$

$$\frac{\partial}{\partial x} E_z(\xi^-, t) = \frac{\partial}{\partial x} E_z(\xi^+, t), \quad (42)$$

$$\frac{1}{\epsilon_1} \left(\frac{\partial}{\partial x} H_y(\xi^-, t) - \frac{\partial}{\partial y} H_x(\xi^-, t) \right) = \frac{1}{\epsilon_2} \left(\frac{\partial}{\partial x} H_y(\xi^+, t) - \frac{\partial}{\partial y} H_x(\xi^+, t) \right). \quad (43)$$

Since the media are homogeneous in the y -direction, the derivatives of fields with respect to y are continuous throughout the domain. Therefore, condition (41) is trivially valid. The condition (42) is also trivial, since it is the same as its counterpart in the 1D case. However, the third condition (43) is genuinely nontrivial, because of the term $(\partial/\partial y)H_x$. This condition is identical to the corresponding 1D jump condition only when $(\partial/\partial y)H_x = 0$, which, however, is not true for general 2D applications. Therefore, one has to consider the derivative approximation in the y -direction along the interface $x = \xi$ in the 2D DM modeling. This is dramatically different from the 1D studies and introduces considerable difficulties, even though $(\partial/\partial y)H_x$ is continuous along x -direction at $x = \xi$.

Similarly, the second-order and the third-order jump conditions can be given as

$$\left(A_1 \frac{\partial}{\partial x} + B_1 \frac{\partial}{\partial y} \right)^2 \mathbf{q}(\xi^-, t) = \left(A_2 \frac{\partial}{\partial x} + B_2 \frac{\partial}{\partial y} \right)^2 \mathbf{q}(\xi^+, t), \quad (44)$$

$$\left(A_1 \frac{\partial}{\partial x} + B_1 \frac{\partial}{\partial y} \right)^3 \mathbf{q}(\xi^-, t) = \left(A_2 \frac{\partial}{\partial x} + B_2 \frac{\partial}{\partial y} \right)^3 \mathbf{q}(\xi^+, t). \quad (45)$$

In the explicit form, we have

$$\frac{1}{\epsilon_1} \frac{\partial^2 H_x(\xi^-, t)}{\partial y^2} - \frac{1}{\epsilon_1} \frac{\partial^2 H_y(\xi^-, t)}{\partial x \partial y} = \frac{1}{\epsilon_2} \frac{\partial^2 H_x(\xi^+, t)}{\partial y^2} - \frac{1}{\epsilon_2} \frac{\partial^2 H_y(\xi^+, t)}{\partial x \partial y}, \quad (46)$$

$$\frac{1}{\epsilon_1} \frac{\partial^2 H_y(\xi^-, t)}{\partial x^2} - \frac{1}{\epsilon_1} \frac{\partial^2 H_x(\xi^-, t)}{\partial x \partial y} = \frac{1}{\epsilon_2} \frac{\partial^2 H_y(\xi^+, t)}{\partial x^2} - \frac{1}{\epsilon_2} \frac{\partial^2 H_x(\xi^+, t)}{\partial x \partial y}, \quad (47)$$

$$\frac{1}{\epsilon_1} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) E_z(\xi^-, t) = \frac{1}{\epsilon_2} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) E_z(\xi^+, t), \quad (48)$$

$$\frac{1}{\epsilon_1} \left(\frac{\partial^3}{\partial x^2 \partial y} + \frac{\partial^3}{\partial y^3} \right) E_z(\xi^-, t) = \frac{1}{\epsilon_2} \left(\frac{\partial^3}{\partial x^2 \partial y} + \frac{\partial^3}{\partial y^3} \right) E_z(\xi^+, t), \quad (49)$$

$$\frac{1}{\epsilon_1} \left(\frac{\partial^3}{\partial x^3} + \frac{\partial^3}{\partial x \partial y^2} \right) E_z(\xi^-, t) = \frac{1}{\epsilon_2} \left(\frac{\partial^3}{\partial x^3} + \frac{\partial^3}{\partial x \partial y^2} \right) E_z(\xi^+, t), \quad (50)$$

$$\begin{aligned} & \frac{1}{\epsilon_1^2} \left(\frac{\partial^3}{\partial x^3} + \frac{\partial^3}{\partial x \partial y^2} \right) H_y(\xi^-, t) - \frac{1}{\epsilon_1^2} \left(\frac{\partial^3}{\partial y^3} + \frac{\partial^3}{\partial x^2 \partial y} \right) H_x(\xi^-, t) \\ &= \frac{1}{\epsilon_2^2} \left(\frac{\partial^3}{\partial x^3} + \frac{\partial^3}{\partial x \partial y^2} \right) H_y(\xi^+, t) - \frac{1}{\epsilon_2^2} \left(\frac{\partial^3}{\partial y^3} + \frac{\partial^3}{\partial x^2 \partial y} \right) H_x(\xi^+, t). \end{aligned} \quad (51)$$

It is noted that in order to numerically deal with these conditions, cross derivatives, such as $(\partial^2/\partial x \partial y)H_y$, are required to be discretized. To approximate a cross derivative by using x and y differential kernels with

length being, respectively, $2M_x$ and $2M_y$, a total number of $4M_xM_y$ grid points is involved. Obviously, the computational cost is then extremely expensive, especially when M_x and M_y are quite large. In viewing this, in the 2D BPS method [40], only one FP at each side of the interface was considered for each field component. Consequently, only the zeroth and first-order jump conditions were used so that the approximation of cross derivatives involved in second or higher-order jump conditions is simply avoided. In the present study, we tentatively explore the discretization of the cross derivatives in low-order jump conditions in the next subsection, before a different consideration is taken.

4.2. A quasi-fourth-order scheme

In the embedded FDTD methods [4,45,46,48–51], one-sided approximations and extrapolations are typically employed to uniformly achieve up to fourth-order of accuracy. It is well known that compared to one-sided derivative approximations, the central approximations are usually more accurate as well as more stable. In the 1D case, we have demonstrated that by using central approximations in the HDM/IDM modeling, one can easily generate embedding FDTD methods systematically to high-order accuracy. It is thus of great interest to explore if we can achieve higher-order accuracy by considering only central approximations for the 2D DM modeling in this subsection. In particular, we consider the use of up to third-order jump conditions to construct a fourth-order DM scheme in 2D.

We first consider the DM modeling for E_z . Similar to the 1D case, we do not require that the interface be laid on the grid nodes. For a fourth-order scheme, two FPs at each side of the interface are needed. The x derivatives involved in up to third-order jump conditions can be similarly dealt with. However, two terms involving y derivatives are also presented in these jump conditions, i.e., $\partial^2 E_z / \partial y^2$ and $\partial^3 E_z / \partial x \partial y^2$, which demand additional attention. It can be derived that these two derivatives are continuous across the interface $x = \xi$. For simplicity, we consider the discretization of $\partial^2 E_z / \partial y^2$ here. The modeling of $\partial^3 E_z / \partial x \partial y^2$ can be similarly carried out.

At the interface $x = \xi$, we have,

$$\frac{\partial^2}{\partial y^2} E_z(\xi^-, t) = \frac{\partial^2}{\partial y^2} E_z(\xi^+, t).$$

Thus, Eq. (48) can be rewritten as

$$\frac{1}{\epsilon_1} \frac{\partial^2}{\partial x^2} E_z(\xi^-, t) = \frac{1}{\epsilon_2} \frac{\partial^2}{\partial x^2} E_z(\xi^+, t) + \left(\frac{1}{\epsilon_2} - \frac{1}{\epsilon_1} \right) \frac{\partial^2}{\partial y^2} E_z(\xi, t). \quad (52)$$

Since the second term on the right-hand side of Eq. (52) is not defined by one-sided limitation, the simplest way to discretize this term is to consider it in numerical approximation of the TM equations, rather than in the DM modeling. In other words, in the DM modeling we regard the whole term $(\partial^2 / \partial y^2) E_z(\xi, t)$ as one independent variable. The representation of f_i by g_i is then modified by adding a new term in the summation

$$f_i = \sum_{j=1}^4 r_{i,j} g_j + r_{i,5} \frac{\partial^2}{\partial y^2} E_z(\xi, t) \quad \text{for } i = 1, 2, 3, 4,$$

where $r_{i,j}$ are representation coefficients. The extra coefficients $r_{i,5}$ can be similarly determined as $r_{i,j}$ in the DM modeling. It is noted that this DM procedure is computationally efficient, since it is equivalent to increase one more grid point and maintain the same number of FPs in comparing with the 1D DM modeling. Furthermore, this procedure is numerically attainable. However, this procedure calls for the need of evaluation of $\partial^2 E_z(\xi, t) / \partial y^2$ at each time step in the time-domain computation. In general cases

that the interface $x = \xi$ is not on the E_z nodes, an interpolation along the x -direction across the interface has to be resorted to estimate $\partial^2 E_z / \partial y^2$. Unfortunately, such an approximation is at most second-order accurate for the given set of stencils, just like the direct discretization of Maxwell's equations. Therefore, in order to achieve higher-order accuracy by using a Cartesian grid, this simple procedure is not applicable. Alternatively, we consider the discretization of the y derivatives, $\partial^2 E_z / \partial y^2$ and $\partial^3 E_z / \partial x \partial y^2$, in the DM modeling.

To discretize two derivatives, $\partial^2 E_z / \partial y^2$ and $\partial^3 E_z / \partial x \partial y^2$, by using central FD approximations up to the fourth-order of accuracy, at least 20 grid points are required for computing E_z , see Fig. 13. In our 1D DM method, each FP is assumed to be on the same position as one corresponding grid node. However, it is impossible to do so in the 2D DM modeling. The calculation of 20 FPs is obviously quite expensive. Furthermore, there are only four jump conditions, so that the algebraic equations for the rest 16 unknowns are absent unless one-sided approximations are used, which is explored in the next subsection. Note that making use of higher-order DM condition places a requirement for an even larger computational support, i.e., more FPs, in order to maintain the accuracy. Hence, there is always an inconsistency between the order of accuracy and the number of FPs in 2D central approximations. Therefore, in the present 2D study, we make use of only four FPs, together with 20 grid points, see Fig. 13. As a consequence, we have to choose Eq. (52) rather than Eq. (48) to be discretized. In Eq. (52), the y derivative $\partial^2 E_z / \partial y^2$ is not defined by one-sided limitation, so that it is natural to discretize this term by the original grid points only. On the other hand, if we consider Eq. (48), we must use both FPs and the original grid points to approximate $\partial^2 / \partial y^2 E_z(\xi^-, t)$ and $\partial^2 / \partial y^2 E_z(\xi^+, t)$. For the same reason, since the differentiation over y in $\partial^3 E_z / \partial x \partial y^2$ is continuous across the interface, we also rewrite the jump condition for it in the form of Eq. (52), and discretize the new condition. In summary, only four FPs are utilized to approximate all terms that involve purely x derivatives in the jump conditions, as in the 1D DM modeling, while 20 original grid points are used for approximations of the rest terms involving y derivatives. As a result, an 80×80 algebraic system is constructed, whose solutions are desired representation coefficients.

The DM modeling for H_y can be similarly carried out. The number of FPs is still 4, however, at least 36 grid points are required now since we also need to discretize derivatives of H_x on staggered nodes. Similarly, four FPs are employed in the approximation of x derivatives in the jump conditions. Jump

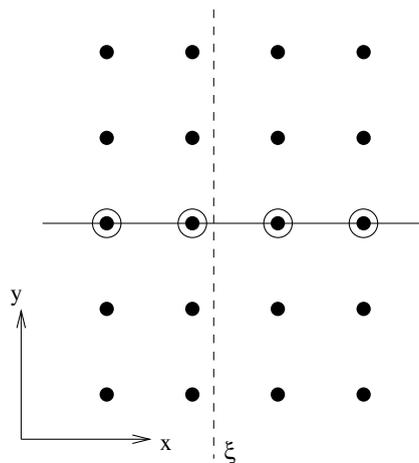


Fig. 13. Illustration of fictitious and original grid points used in the quasi-fourth-order DM scheme for E_z . Along the solid line, the DM is modeled. The FPs are shown as open circles, while the original grid points are shown as filled circles. The FPs are located at the same position as some grid points.

conditions involving y derivatives and derivatives for H_x are rewritten in the form of Eq. (52). We then discretize these derivatives by using the original grid nodes only. The size of the resulting algebraic system is 144×144 . However, it is noted that now some derivatives, such as $\partial^3 H_x / \partial x^2 \partial y$, are actually discontinuous at the interface, so that certain reductions of accuracy are encountered in the approximation of these terms by using the original grid points. We numerically investigate this issue later. Surely, as an academic exercise on data structure and coding, the treatment of mixed derivatives can still be improved in many different ways. However, as complicated as it is, it is somewhat impractical to further pursue these alternatives.

By combining the DM modelings for both E_z and H_y , and by considering a standard fourth-order FDTD scheme, a new high-order central FDTD scheme is obtained for Maxwell's Eq. (2). Although the development of this scheme is aimed to fourth-order of accuracy, certain reductions in accuracy are introduced in the DM modeling for H_y . Therefore, this scheme is referred to as a quasi-fourth-order DM (Q4DM) method. Compared to the fourth-order embedding schemes [4,48–51], the proposed Q4DM scheme might be technically more complicated and numerically less accurate. However, the proposed method has promising to be applied to more complex geometries, since it is designed on a simple Cartesian grid and does not require the interface to be on the grid.

The present study provide us a deep appreciation of difficulties and remaining problems in the 2D DM modeling. In addition, the 2D DM modeling is also found to be quite complex from the viewpoint of programming. In the 1D case, the high-order jump conditions can be easily generated. Moreover, these conditions only involve derivatives with respect to x . Thus, it is quite easy to write a code to construct these jump conditions. However, in the 2D case, various different cross derivatives appear each time when we consider a new jump condition. Computationally, these cross derivatives are better to be represented via a symbolic language, such as MAPLE. If the construction of these jump conditions are handled by a FORTRAN code, well-designed data structure is indispensable, which is quite difficult for programming and would be extremely time-consuming. This difficulty restricts the use of high-order jump conditions in the 2D DM modeling.

Similar to 1D studies, it is also interesting to examine if the present Q4DM scheme can be extended to high-order systematically. However, besides the associated complexity, such a generalization by considering high-order jump conditions might not be able to achieve high-order of convergence eventually, simply because the more and more cross derivatives being discretized on the original grid are actually discontinuous. Therefore, we do not consider the generalization of this Q4DM scheme to higher order along this line. Alternatively, in order to achieve high-order of accuracy, one-sided approximation is considered in the 2D DM modeling in the next subsection, similar to the embedding FDTD schemes [4,45,46,48–51] and the 2D BPS method [40].

4.3. 2D HDM method

A 2D HDM method based on the use of one-sided FD approximations is investigated in this subsection. Due to the complexity of using high-order jump conditions, only low-order jump conditions are considered for simplicity. The resulting HDM method could be very similar to the counterpart of the explicit DM method consider in [40], i.e., only two FPs and two jump conditions are considered for each field component. However, in order to maintain the general applicability of adopting the scheme in various time-domain methods and the simplicity of using structured grid, a uniform staggered grid system, with the FPs located at the same positions as the corresponding grid nodes, is still employed in the present 2D DM method. Obviously, certain instability issues would correspondingly arise. We also explore this issue later.

We first select low-order jump conditions. For H_y , the zeroth and first-order jump conditions, Eqs. (39) and (43), are considered. For E_z , to avoid trivial conditions, we choose first and second-order jump conditions, Eqs. (42) and (48). These four jump conditions are employed throughout the 2D HDM modeling.

Two y -direction derivatives involve in these four conditions, i.e., $(\partial/\partial y)H_x(\xi)$ and $(\partial^2/\partial y^2)E_z(\xi)$. Note that these two terms are continuous at the interface $x = \xi$. Thus, we rewrite the jump conditions (43) and (48) in the form

$$\frac{1}{\epsilon_1} \frac{\partial}{\partial x} H_y(\xi^-, t) = \frac{1}{\epsilon_2} \frac{\partial}{\partial x} H_y(\xi^+, t) + \left(\frac{1}{\epsilon_1} - \frac{1}{\epsilon_2} \right) \frac{\partial}{\partial y} H_x(\xi, t), \quad (53)$$

$$\frac{1}{\epsilon_1} \frac{\partial^2}{\partial x^2} E_z(\xi^-, t) = \frac{1}{\epsilon_2} \frac{\partial^2}{\partial x^2} E_z(\xi^+, t) + \left(\frac{1}{\epsilon_2} - \frac{1}{\epsilon_1} \right) \frac{\partial^2}{\partial y^2} E_z(\xi, t). \quad (54)$$

For simplicity, in approximating the two y derivatives in the present 2D HDM method, we assume the interface $x = \xi$ laid on some E_z nodes. It is noted that, in a 2D staggered grid system [24–26], if the interface $x = \xi$ is on some E_z nodes, it is also on some H_x nodes. Therefore, by using a body-conformed structured grid with some E_z (H_x) nodes on the interface, the evaluation of two y derivatives $(\partial/\partial y)H_x(\xi)$ and $(\partial^2/\partial y^2)E_z(\xi)$ at each time step could be quite straightforward and extremely accurate. However, we also note that the present HDM scheme is then not suitable for general Cartesian grids, but is just applicable to body-conformed structured grids, similar to the fourth-order embedding FDTD schemes [48–51].

In numerical computation, we regard two derivatives $(\partial/\partial y)H_x(\xi)$ and $(\partial^2/\partial y^2)E_z(\xi)$ as independent variables, and simply calculate their representation coefficients in the DM modeling. For instance, we consider the IDM modeling for E_z here. Apart from two FPs, we need $2l$ grid points to form the one-sided approximation, see Fig. 14. We denote the y derivative term $(\partial^2/\partial y^2)E_z(\xi)$ as g_e . Visually, g_e can be just thought as an extra grid point, as shown in Fig. 14. Then, the jump conditions Eqs. (42) and (54) are approximated as

$$\sum_{i=1}^l w_{1,i}^{(1)} g_i + w_{1,l+1}^{(1)} f_2 = w_{1,1}^{(2)} f_1 + \sum_{i=2}^{l+1} w_{1,i}^{(2)} g_{l+i-1}, \quad (55)$$

$$\frac{1}{\epsilon_1} \left(\sum_{i=1}^l w_{2,i}^{(1)} g_i + w_{2,l+1}^{(1)} f_2 \right) = \frac{1}{\epsilon_2} \left(w_{2,1}^{(2)} f_1 + \sum_{i=2}^{l+1} w_{2,i}^{(2)} g_{l+i-1} \right) + \left(\frac{1}{\epsilon_2} - \frac{1}{\epsilon_1} \right) g_e, \quad (56)$$

where one-sided FD weights $w_{j,i}^{(k)}$ for $i = 1, \dots, l+1$, $j = 1, 2$, and $k = 1, 2$ can be generated by using the fast algorithm presented in [61]. Here the subscript j represents the first or second order derivative approximation, and i is for grid index. The superscripts (1) and (2) are followed from the permittivities ϵ_1 and ϵ_2 . The representation coefficients for two FPs can be simply solved from Eqs. (55) and (56). The similar IDM modeling can also be carried out for H_y . The IDM modeling of E_z and H_y constitutes the first stage of the proposed 2D HDM method, see also Fig. 14.

Similar to the 1D HDM method, in stage two, the representation coefficients of total $2m$ FPs are calculated via a hierarchical procedure, and usually we have $m = M_x$. The procedure is recursive, with only two new representation coefficients being sought at each step. The one-sided FD approximations used in each step can be similarly formulated as in Eqs. (55) and (56). Again, these FD approximations are of higher order accuracy than those in stage one. It is noted that during the hierarchical procedure, the number of grid points is kept the same as in stage one, i.e., $2l$, see Fig. 14. Thus, $l > m$ is admissible in the 2D HDM method. This is different from the 1D HDM method. The 2D HDM modeling stops if $l \geq m$. If $M_x = m > l$, one additional stage, the stage 3 of the HDM method, is invoked, in which both numbers of FPs and grid points are increased and central FD approximations are employed, see Fig. 14. In such a case, we have $l = m$ as in the 1D HDM method. It is obvious that the computation in the present HDM is in much spirit of that in the 1D HDM method. Thus, it is also quite simple as well as efficient.

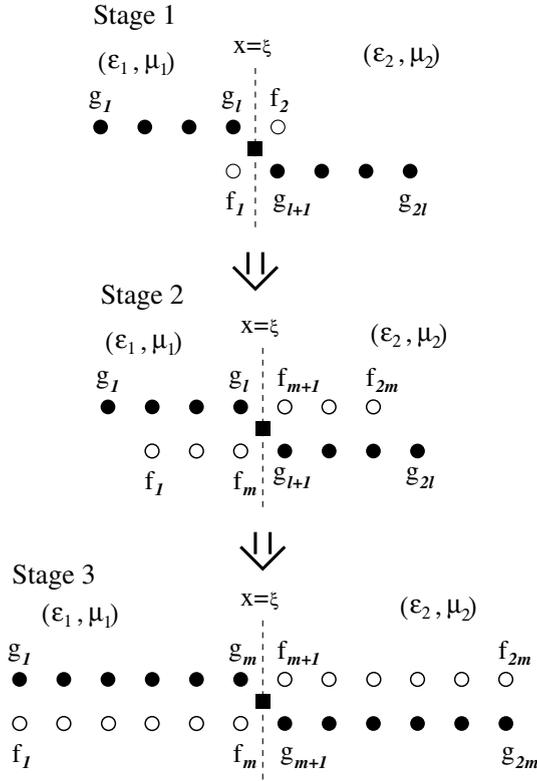


Fig. 14. Illustration of fictitious and original grid points used in the 2D HDM modeling for H_y . The FPs are shown as open circles, while the original grid points are shown as filled circles. The filled square is for $(\partial/\partial y)H_x(\xi)$. The stage 3 is invoked only when $m > l$.

However, it is well known that one-sided approximation becomes unstable very soon when l is large. It is thus interesting to carry out a stability analysis of the present 2D HDM modeling. We first consider the stability issue of the FDTD methods without the HDM method. Similar to 1D studies, we represent the complete spatial approximation to the right-hand side of Eq. (37) by a matrix \mathbf{S} , then the semidiscrete form of Maxwell's Eq. (37) can be given as

$$\frac{\partial \mathbf{Q}^n}{\partial t} = \frac{1}{\Delta} \mathbf{S} \mathbf{Q}^n, \tag{57}$$

where we assume $\Delta = \Delta x = \Delta y$ for simplicity. The vector \mathbf{Q} contains all unknowns in E_z , H_x , and H_y , and its size is determined by the size of computational domain $N_x \times N_y$. Similarly, the full discretization form combining both the RK4 temporal integration and the spatial approximation in Eq. (57) can be rewritten as

$$\mathbf{Q}^{n+1} = \left[\mathbf{I} + \frac{\Delta t}{\Delta} \mathbf{S} + \frac{1}{2} \frac{\Delta t^2}{\Delta^2} \mathbf{S}^2 + \frac{1}{6} \frac{\Delta t^3}{\Delta^3} \mathbf{S}^3 + \frac{1}{24} \frac{\Delta t^4}{\Delta^4} \mathbf{S}^4 \right] \mathbf{Q}^n = \mathbf{S}' \mathbf{Q}^n.$$

The time-domain computation is stable if the spectral radius of \mathbf{S}' satisfying $\rho^{\mathbf{S}'} \leq 1$ for all Δ . As in 1D studies, we calculate the spectral radius of \mathbf{S} to see if $\rho^{\mathbf{S}} \rightarrow \rho^\infty$, where $\rho^\infty = \sqrt{2} \sum_{j=-M_x, j \neq 0}^{M_x} |w_j|$. The plots of $\rho^{\mathbf{S}}$ for different resolutions are given in Fig. 15. These results demonstrate that the stability condition for the

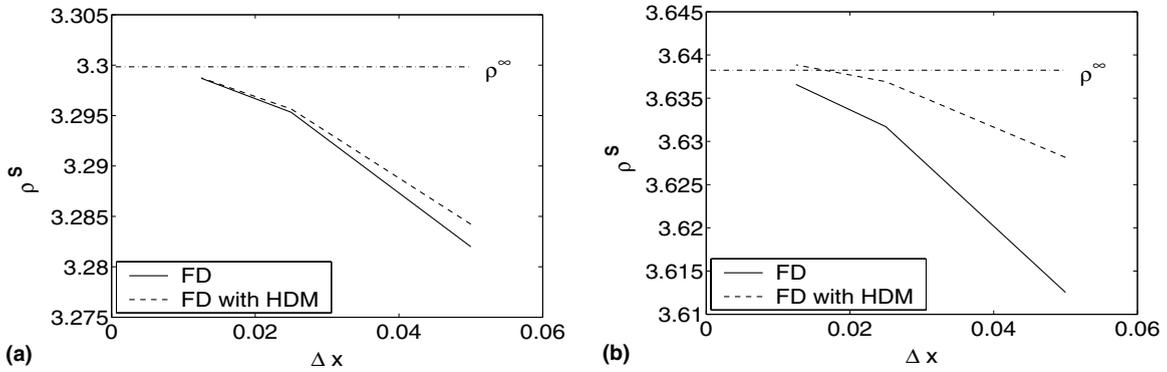


Fig. 15. The spectral radius ρ^S as a function of mesh size for the FDTD method with the 2D HDM. (a) $M = m = l = 2$ ($\rho^\infty = 7\sqrt{2}/3 = 3.29983$); (b) $M = m = l = 4$ ($\rho^\infty = 3.63823$).

2D FDTD method without the HDM is still that for 1D, i.e., Eq. (31). This has also been numerically verified.

We then consider the stability condition of the FDTD method with the 2D HDM modeling. The plots of ρ^S with $m = l = 2$ and $m = l = 4$ are also shown in Fig. 15. It can be observed that when $l = 2$, ρ^S also converges to ρ^∞ as $\Delta \rightarrow 0$. But when $l = 4$, it converges to a value which is slightly larger than ρ^∞ . In fact, when l is increased further, the ρ^S can be much larger than ρ^∞ , see Fig. 16. A fixed grid resolution with $\Delta = 0.025$ is used for all cases in Fig. 16. It is found in our analysis that when $l \leq 8$, the FDTD approximations can be conditionally stable. For these stable schemes, the stability constraint usually becomes more severe when l is increased, and we have roughly,

$$\Delta t \lesssim \sqrt{\min\{\epsilon\mu\}} \frac{2\sqrt{2}}{\rho^S} \Delta x. \tag{58}$$

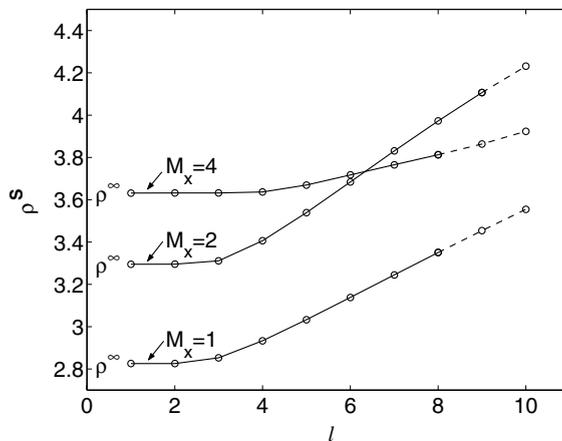


Fig. 16. The spectral radius ρ^S as a function of parameter l for the FDTD method with the 2D HDM. A fixed $\Delta = \Delta x = \Delta y = 0.025$ is used. For $M_x = m = 1, 2$, and 4 , $\rho^\infty = 2.82843, 3.29983$, and 3.63823 , respectively. When l is large, the scheme is unconditionally unstable. The corresponding spectral radius values are connected by dashed lines.

In other words, the stability condition by using the 2D HDM with $l \leq 8$ can be simply estimated by calculating corresponding ρ^S . This finding holds not only for $M_x = 1, 2$, and 4 but also for a larger M_x .

However, when $l > 8$, the FDTD approximation with the 2D HDM could be unconditionally unstable, even though no abrupt changes appear in ρ^S curves in Fig. 16. Instead, there are some significant changes in eigenvalues of \mathbf{S} , i.e., λ_j^S . By considering a resolution with $\Delta = 0.05$, we have examined the real parts of λ_j^S for different l . When $l \leq 8$, except at most a few λ_j^S , most of eigenvalues have zero real parts. Moreover, the real parts of those exceptional λ_j^S are of small magnitudes. However, when $l > 8$, the real parts of at least half eigenvalues are not zero, but of magnitude ranging from $O(10^{-13})$ to $O(10^{-15})$. Furthermore, quite a few eigenvalues even have real parts which are significantly larger (or smaller) than zero. Consequently, the corresponding spectral radius of \mathbf{S}' , $\rho^{S'}$, is always larger than unit no matter how small Δt is. Thus, the numerical approximation becomes unconditionally unstable, as confirmed in our numerical experiments. The same results are also valid for a larger M_x . Therefore, $l = 8$ might be the highest value which still produces a stable discretization for the 2D HDM method. We thus choose up to $l = 8$ in our following numerical studies for achieving the best accuracy. It is believed that the present stability analysis has significantly enhanced our understanding of the potential and limitation in constructing high order embedded FDTD methods.

The advantages of the HDM method, compared to the Q4DM scheme, are its simplicity in programming and capability to achieve high-order accuracy. However, the requirement of a body fitting structured grid system may weaken its applicability to general electromagnetic problems. Furthermore, since the one-sided approximation becomes unstable as l is quite large, it is impossible to systematically generate arbitrarily high-order schemes within the framework of the proposed HDM method, at least in principle. Finally, it is noted that in 1D cases, the HDM method can be regarded as a supplement to the IDM method in some sense. However, in 2D cases, the hierarchical method plays a fundamental role for generating high-order embedding methods in the DM modeling. For 3D real applications, the HDM method is expected to be more promising.

4.4. Numerical studies

We numerically investigate the performance of proposed two schemes for 2D electromagnetic applications in this subsection. The standard high-order FDTD methods are employed for basic spatial discretization, although other highly accurate time-domain approaches, like the MRTD and LSTD methods, may also be employed. The time stepping method is still the RK4 scheme. Similar to 1D studies, boundary extensions are used to impose the PEC conditions.

The first problem being considered has a PEC-bounded domain $\Omega = \{(x, y) | 0 \leq x \leq 5/4, 0 \leq y \leq 1\}$. The permittivity is defined by $\epsilon = \epsilon_2$ if $0 \leq x \leq \frac{1}{2}$ and $0 \leq y \leq 1$, and $\epsilon = \epsilon_1$ if $\frac{1}{2} \leq x \leq \frac{5}{4}$ and $0 \leq y \leq 1$. The exact solution for time-varying electromagnetic fields is [49]

$$E_z = \begin{cases} \sin(a_1x) \sin(by) \sin(\omega t), & 0 \leq x \leq \frac{1}{2}, 0 \leq y \leq 1, \\ \cos(a_2x) \sin(by) \sin(\omega t), & \frac{1}{2} \leq x \leq \frac{5}{4}, 0 \leq y \leq 1, \end{cases}$$

$$H_y = \begin{cases} -\frac{a_1}{\omega} \cos(a_1x) \sin(by) \cos(\omega t), & 0 \leq x \leq \frac{1}{2}, 0 \leq y \leq 1, \\ \frac{a_2}{\omega} \sin(a_2x) \sin(by) \cos(\omega t), & \frac{1}{2} \leq x \leq \frac{5}{4}, 0 \leq y \leq 1, \end{cases}$$

$$H_x = \begin{cases} \frac{b}{\omega} \sin(a_1x) \cos(by) \cos(\omega t), & 0 \leq x \leq \frac{1}{2}, 0 \leq y \leq 1, \\ \frac{b}{\omega} \cos(a_2x) \cos(by) \cos(\omega t), & \frac{1}{2} \leq x \leq \frac{5}{4}, 0 \leq y \leq 1, \end{cases}$$

where $a_1^2 + b^2 = \epsilon_2 \omega^2$, $a_2^2 + b^2 = \epsilon_1 \omega^2$, $\sin(\frac{a_1}{2}) = \cos(\frac{a_2}{2})$, $\cos(\frac{5a_2}{4}) = 0$. As in [49], the first set of parameters to be tested is chosen as $\epsilon_1 = 1$, $\epsilon_2 = 2$, $a_1 = 3\pi$, $a_2 = 2\pi$, $b = \pi$, and $\omega = \sqrt{5}\pi$.

We first examine the convergence rates of two x derivative approximations in the Q4DM method. The errors of spatial discretizations of $\partial E_z / \partial x$ and $\partial H_y / \partial x$ are computed by using the proposed Q4DM scheme with exact field values of E_z and H_y at a fixed time, say $t = 0.1$. The numerically tested convergence rates in the approximation of $\partial E_z / \partial x$ and $\partial H_y / \partial x$ are listed in Table 7. Here, a quite long differential stencil in y direction with $M_y = 8$ is used such that the spatial discretization errors are introduced mainly by the approximation of x derivatives. It is clear from Table 7 that the approximation of $\partial E_z / \partial x$ is around fourth-order. However, the order of accuracy is only around 2.5 for the approximations to $\partial H_y / \partial x$. This confirms our previous discussions that the approximation of cross derivatives which are discontinuous at the interface $x = \xi$ by using solely the original grid nodes introduces certain reductions in accuracy. Fortunately, such reductions are not too grave, so that the overall convergence rates of the interested field components E_z , H_x and H_y are still quite close to fourth-order, see Table 8. Thus, numerical results of the Q4DM method in Table 8 justify its quasi-fourth-order accuracy. Meanwhile, it is noted that the Q4DM method performs well for both cases where the interface is either on some E_z nodes (i.e., cases whose N_x take odd integers) or not (i.e., cases whose N_x take even integers).

We next study the same 2D problem by considering the 2D HDM method. Following the above discussions, a maximum value of parameter l is chosen, i.e., $l = 8$. The numerical results for different $M_x = m$ are given in Table 9. It is clear that for $M_x = 1, 2$, and 4 , the numerically tested convergence rates are almost identical to the theoretical ones, see also Fig. 17. For $M_x = 6$ and $M_x = 8$, it is found that both numerical accuracy and numerically tested rates for these two cases are exactly identical, except when the limit of

Table 7
The L_2 errors in x derivative approximations of the FDTD methods with the Q4DM

(N_x, N_y)	$\partial E_z / \partial x$		$\partial H_y / \partial x$	
	Error	Rate	Error	Rate
(50, 41)	7.69(-5)		9.92(-5)	
(100, 81)	6.37(-6)	3.541	1.72(-5)	2.490
(200, 161)	5.44(-7)	3.524	3.03(-6)	2.489
(51, 41)	2.88(-5)		7.12(-5)	
(101, 81)	1.88(-6)	3.936	1.15(-5)	2.635
(201, 161)	1.26(-7)	3.901	1.99(-6)	2.525

Here $M_x = 2$, and a longer differential kernel is used in y direction with $M_y = 8$. Two types of grid resolutions, i.e., E_z nodes being coincided with the interface $x = \xi$ (N_x take odd integers) or not (N_x take even integers), are considered.

Table 8
The L_2 errors of the FDTD methods with the Q4DM at time $t = 1$ with $\Delta t = 5.0 \times 10^{-4}$

(N_x, N_y)	E_z		H_x		H_y	
	Error	Rate	Error	Rate	Error	Rate
(50, 41)	2.76(-5)		1.07(-5)		3.21(-5)	
(100, 81)	1.93(-6)	3.780	7.44(-7)	3.791	2.16(-6)	3.841
(200, 161)	1.50(-7)	3.656	5.80(-8)	3.653	1.59(-7)	3.732
(51, 41)	2.93(-5)		1.12(-5)		3.10(-5)	
(101, 81)	2.30(-6)	3.671	8.80(-7)	3.670	2.34(-6)	3.728
(201, 161)	2.03(-7)	3.499	7.89(-8)	3.479	1.98(-7)	3.558

The rest parameter setting is the same as that in Table 7.

Table 9

The L_2 errors of the FDTD methods with the 2D HDM at time $t = 1$ with $\Delta t = 2.5 \times 10^{-4}$

M_x	(N_x, N_y)	E_z		H_x		H_y	
		Error	Rate	Error	Rate	Error	Rate
1	(51, 41)	3.97(-3)		1.53(-3)		4.23(-3)	
	(101, 81)	1.00(-3)	1.990	3.84(-4)	1.990	1.07(-3)	1.986
	(201, 161)	2.51(-4)	1.994	9.63(-5)	1.996	2.68(-4)	1.995
2	(51, 41)	2.25(-5)		8.67(-6)		2.48(-5)	
	(101, 81)	1.42(-6)	3.985	5.44(-7)	3.993	1.56(-6)	3.990
	(201, 161)	8.93(-8)	3.993	3.41(-8)	3.998	9.79(-8)	3.996
4	(51, 41)	2.42(-9)		9.64(-10)		2.60(-9)	
	(101, 81)	7.05(-12)	8.484	2.67(-12)	8.499	7.73(-12)	8.394
	(201, 161)	2.90(-14)	7.924	8.22(-15)	8.341	2.35(-14)	8.303
6	(26, 21)	1.29(-6)		5.60(-7)		1.20(-6)	
	(51, 41)	1.09(-9)	10.214	4.59(-10)	10.214	1.00(-9)	10.224
	(101, 81)	6.00(-13)	10.821	2.52(-13)	10.833	5.57(-13)	10.817
	(201, 161)	1.33(-14)	5.499	5.06(-15)	5.636	1.29(-14)	5.430
8	(26, 21)	1.29(-6)		5.62(-7)		1.21(-6)	
	(51, 41)	1.09(-9)	10.208	4.61(-10)	10.251	1.01(-9)	10.224
	(101, 81)	6.00(-13)	10.821	2.52(-13)	10.838	5.56(-13)	10.825
	(201, 161)	1.95(-14)	4.940	5.80(-15)	5.441	1.56(-14)	5.159

Here $M_y = 8$ is used and the maximal $l = 8$ is employed for all M_x values.

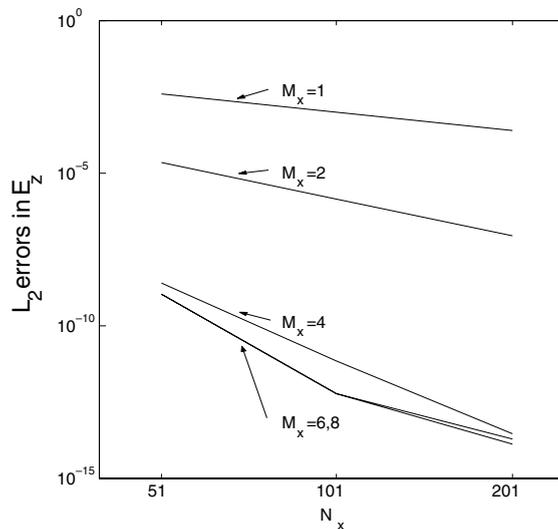


Fig. 17. Log-log plots of the L_2 errors in E_z at time $t = 1$ by using the 2D HDM method with $l = 8$ for different M_x values. Here $\Delta t = 2.5 \times 10^{-4}$ and $M_y = 8$.

double precision is reached. This suggests that the accuracy of both schemes is limited by that of the 2D HDM with $l = 8$. The theoretical order of accuracy of $M_x = 6$ and $M_x = 8$ should be, respectively, 12 and 16, which, however, are higher than the numerical convergence rate of the HDM method. Therefore, the numerically tested order of accuracy for both schemes is about from 10.2 to 10.8 for the present problem.

Since a maximum $l = 8$ is employed in the 2D HDM, the order of 10.8 might be the highest that the present HDM scheme can possibly achieve for this problem.

Similar to the 1D case, we note that in Table 9 when $(N_x, N_y) = (201, 161)$, our 12th order HDM ($M_x = 6$) could be about 10^7 times more accurate than our fourth order one. It is interested herein to compare the efficiency of these two schemes. For simplicity, we focus only on the L_2 errors of E_z , since those of H_x and H_y follow the same pattern. In Table 9, by using $\Delta t = 2.5 \times 10^{-4}$ and $(N_x, N_y) = (201, 161)$, the error of the 4th and 12th order schemes is $8.93(-8)$ and $1.33(-14)$, respectively, while the CPU time is 6.7 and 8.5 min, respectively. The CPU time is recorded on the same Sun workstation (Blade 2000) and two schemes are generated by the same code. In an ideal case, it takes slightly more than 5 mesh refinements in order for the 4th order scheme to reduce the error from $8.93(-8)$ to $1.33(-14)$, and finally mesh size (N_x, N_y) should be about $(10231, 8185)$, see Table 10. Basically, for refinement 1–5, when the mesh size is doubled, the error will be divided by 16 (4th order scheme, $2^4 = 16$) and CPU time will be multiplied by 4 (2D computation, $2^2 = 4$). The last refinement is targeted to an error of $1.30(-14)$, and thus the CPU time is estimated to be 17416.4, which is about 2048 times of the CPU time used by the 12th order scheme, 8.5 min. Therefore, an efficiency gain factor about 2048 could be achieved by using the 12th order HDM instead of the 4th order one for this particular problem.

In the following, we will show based on numerical experiments that this efficiency estimate is very conservative. Conducted on the same machine, numerically tested accuracy and CPU time are reported in Table 10. It is seen from Table 10 that, for refinements 1–4 the estimated errors are in excellent agreement with the experimental ones. Moreover, the general pattern of the experimental CPU time also follows our estimation. Although the experimental CPU time is slightly larger than the estimated one, it increases by a factor slightly larger than 4, when the mesh is doubled in each dimension. The same CPU pattern has been observed when different workstations (Sun Blade 1000) are used.

Furthermore, the estimated CPU time should be even larger, if the stability constraint is taken into account. In the previous estimation, a fixed $\Delta t = 2.5 \times 10^{-4}$ is assumed, while Δt should satisfy the stability condition (58). In the present case, $\sqrt{\min\{\epsilon\mu\}} = 1$, and ρ^S can be observed from Fig. 16 to be around 4 for $M_x = 2$ and $l = 8$. Thus, we have approximately $\Delta t < 0.7\Delta x$, which has been numerically validated. Then, it can be easily shown that one has to reduce the Δt to about 1.3×10^{-4} for the refinement 5, i.e., $(N_x, N_y) = (6401, 5121)$, otherwise the scheme will be unstable. Similarly, the time step for refinement 6 has to be reduced to about $\Delta t = 8.5 \times 10^{-5}$ as the scheme is limited by the stability. As a consequence, about 2.94 times of the originally estimated CPU time are required for our 4th order scheme to reach the accuracy attained by the 12th order scheme. Therefore, the efficiency gain factor of the 12th order scheme over the 4th order scheme could be about 6024, if the stability constraint is taken into account.

Table 10

Theoretically estimated and experimentally tested CPU time and numerical errors for successive refinements of the 4th order scheme

Refinement	(N_x, N_y)	Estimated		Tested	
		Error	CPU	Error	CPU
1	(401, 321)	5.58(-9)	26.6	5.59(-9)	29.7
2	(801, 641)	3.49(-10)	106.6	3.50(-10)	136.8
3	(1601, 1281)	2.18(-11)	426.0	2.19(-11)	752.1
4	(3201, 2561)	1.36(-12)	1704.1	1.34(-12)	3211.2
5	(6401, 5121)	8.52(-14)	6816.6	–	–
6	(10231, 8185)	1.30(-14)	17416.4	–	–

CPU time in minutes is reported.

The numerical demonstration of refinements 5 and 6 in Table 10 has not been carried out because their memory requirement is larger than 2 Gbytes, which is beyond the limit of our workstation. However, it is straightforward to verify our estimation when a better computer is available.

A comparison of the performance between the Q4DM and the 4th order 2D HDM is in order, as the accuracy of the former is questionable due to its approximation nature. This can be done by comparing Tables 8 and 9. It turns out that in fact, the accuracy of the Q4DM is very close to that of the 2D HDM at the corresponding approximation order. This might be due to the central schemes used in Q4DM which are in general more accurate than one-sided approximations. Therefore, we believe that Q4DM is also a successful scheme whose accuracy is compatible with other 4th order schemes in the literature [4,48–51].

It is interesting to consider a high frequency wave study for the present problem to further explore the numerical performance of the 2D HDM method. The parameters of the problem are chosen as $\epsilon_1 = 1$, $\epsilon_2 = \frac{261}{136}$, $a_1 = 15\pi$, $a_2 = 10\pi$, $b = 6\pi$, and $\omega = 2\sqrt{34}\pi$. The numerical results are given in Table 11. For the short wave problem, we also find that the 2D HDM method with $M_x = 6$ and $M_x = 8$ has an identical convergent rate, which is around 9.7.

We further examine the numerical performance of two proposed schemes by considering another 2D electromagnetic problem. In this problem, a lossless dielectric with a relative permittivity of ϵ_2 is enclosed by air in x direction, and the media are nonmagnetic and homogeneous along y direction. The computational domain $\Omega = \{(x, y) | |x| \leq 1, |y| \leq 1\}$ is enveloped by PEC walls. The permittivity is given as $\epsilon = \epsilon_2$ if $\frac{1}{2} \leq |x| \leq 1$ and $|y| \leq 1$, and $\epsilon = \epsilon_1$ if $|x| \leq \frac{1}{2}$ and $|y| \leq 1$, where $\epsilon_1 = 1$ and $\epsilon_2 = 2.25$. An exact solution for time-varying electromagnetic fields can be given as

$$E_z = \begin{cases} \sin(\frac{\omega_2}{2}) \sin(\omega_1(x+1)) \sin(\omega_y y) \cos(\omega t), & -1 \leq x < -\frac{1}{2} |y| \leq 1, \\ -\sin(\frac{\omega_1}{2}) \sin(\omega_2 x) \sin(\omega_y y) \cos(\omega t), & -\frac{1}{2} \leq x \leq \frac{1}{2} |y| \leq 1, \\ \sin(\frac{\omega_2}{2}) \sin(\omega_1(x-1)) \sin(\omega_y y) \cos(\omega t), & \frac{1}{2} < x \leq 1 |y| \leq 1, \end{cases}$$

Table 11

The L_2 errors of the FDTD methods with the 2D HDM at time $t = 0.5$ with $\Delta t = 1.0 \times 10^{-4}$

M_x	(N_x, N_y)	E_z		H_x		H_y	
		Error	Rate	Error	Rate	Error	Rate
1	(51, 41)	1.88(-1)		1.29(-1)		2.79(-1)	
	(101, 81)	6.44(-2)	1.550	2.56(-2)	2.333	5.58(-2)	2.320
	(201, 161)	1.70(-2)	1.922	5.83(-3)	2.132	1.27(-2)	2.136
2	(51, 41)	3.63(-2)		1.39(-2)		3.02(-2)	
	(101, 81)	2.37(-3)	3.935	8.35(-4)	4.058	1.83(-3)	4.046
	(201, 161)	1.52(-4)	3.967	5.29(-5)	3.980	1.16(-4)	3.981
4	(51, 41)	4.45(-3)		2.71(-3)		5.32(-3)	
	(101, 81)	1.09(-5)	8.675	4.55(-6)	9.217	8.99(-6)	9.210
	(201, 161)	3.18(-8)	8.420	1.20(-8)	8.571	2.48(-8)	8.502
6	(51, 41)	4.91(-3)		2.66(-3)		5.15(-3)	
	(101, 81)	6.12(-6)	9.647	3.30(-6)	9.650	6.48(-6)	9.632
	(201, 161)	7.19(-9)	9.732	3.85(-9)	9.745	7.65(-9)	9.727
8	(51, 41)	4.97(-3)		2.67(-3)		5.17(-3)	
	(101, 81)	6.13(-6)	9.664	3.31(-6)	9.658	6.49(-6)	9.638
	(201, 161)	7.22(-9)	9.729	3.86(-9)	9.742	7.67(-9)	9.726

High frequency parameter setting is used. Here $M_y = 8$ is used and the maximal $l = 8$ is employed for all M_x values.

$$H_x = \begin{cases} -\frac{\omega_y}{\omega} \sin(\frac{\omega_2}{2}) \sin(\omega_1(x+1)) \cos(\omega_y y) \sin(\omega t), & -1 \leq x < -\frac{1}{2}|y| \leq 1, \\ \frac{\omega_y}{\omega} \sin(\frac{\omega_1}{2}) \sin(\omega_2 x) \cos(\omega_y y) \sin(\omega t), & -\frac{1}{2} \leq x \leq \frac{1}{2}|y| \leq 1, \\ -\frac{\omega_y}{\omega} \sin(\frac{\omega_2}{2}) \sin(\omega_1(x-1)) \cos(\omega_y y) \sin(\omega t), & \frac{1}{2} < x \leq 1|y| \leq 1, \end{cases}$$

$$H_y = \begin{cases} \frac{\omega_1}{\omega} \sin(\frac{\omega_2}{2}) \cos(\omega_1(x+1)) \sin(\omega_y y) \sin(\omega t), & -1 \leq x < -\frac{1}{2}|y| \leq 1, \\ -\frac{\omega_2}{\omega} \sin(\frac{\omega_1}{2}) \cos(\omega_2 x) \sin(\omega_y y) \sin(\omega t), & -\frac{1}{2} \leq x \leq \frac{1}{2}|y| \leq 1, \\ \frac{\omega_1}{\omega} \sin(\frac{\omega_2}{2}) \cos(\omega_1(x-1)) \sin(\omega_y y) \sin(\omega t), & \frac{1}{2} < x \leq 1|y| \leq 1, \end{cases}$$

where $\omega_1^2 + \omega_y^2 = \epsilon_1 \omega^2$ and $\omega_2^2 + \omega_y^2 = \epsilon_2 \omega^2$. The value of ω can be determined according to the relation

$$\sqrt{\epsilon_2 \omega^2 - \omega_y^2} \tan\left(\frac{\sqrt{\epsilon_1 \omega^2 - \omega_y^2}}{2}\right) = \sqrt{\epsilon_1 \omega^2 - \omega_y^2} \tan\left(-\frac{\sqrt{\epsilon_2 \omega^2 - \omega_y^2}}{2}\right).$$

In the first test case, we choose $\omega_y = 2\pi$ to satisfy the PEC conditions on $y = \pm 1$. Correspondingly, $\omega \approx 9.07716175885174$. An example plot of analytical solutions is shown in Fig. 18.

Numerical results of two proposed schemes for this case are listed in Table 12. Again, the HDM schemes with both $M_x = 6$ and $M_x = 8$ yield almost the same accuracy and convergence rate. It is also noted that now the HDM method with $l = 8$ produces a higher order accuracy than in the previous example, i.e., about 12th-order. It is also clear that in Table 12, the HDM method is much more accurate than the Q4DM scheme.

We then study two proposed schemes for a high frequency setting. By using $\omega_y = 5\pi$, we have $\omega \approx 16.81412105455500$. The numerical results of this case are given in Table 13. Essentially, the results of both schemes are similar to those in Table 12. CPU time of two methods is also given in Tables 12 and 13. The CPU time is smaller than the HDM method. However, if the same accuracy level is required to be achieved, the 2D HDM is significantly more cost-efficient than the Q4DM. On the other hand, it is noted that, although the Q4DM method is slightly less accurate, it possesses better applicability for general 2D electromagnetic structures. This suggests that there exists a tradeoff between the robustness and accuracy (thus cost-efficiency) for selecting these 2D time-domain approaches. The 2D HDM method can be applied to quite general electromagnetic problems, and it can always achieve very high accuracy, as shown in Tables

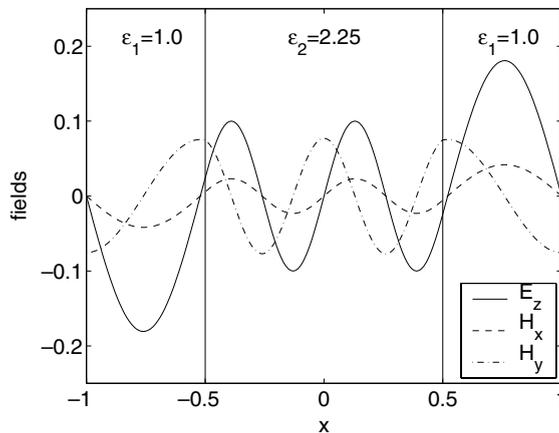


Fig. 18. Plots of electromagnetic fields along the line $y = 1/3$ at time $t = 0.75$.

Table 12
The L_2 errors of the FDTD method at time $t = 1$ with $\Delta t = 2.0 \times 10^{-4}$

Scheme	M_x	l	(N_x, N_y)	E_z		H_x		H_y		CPU
				Error	Rate	Error	Rate	Error	Rate	
Q4DM	2	–	(21, 21)	1.15(–4)		1.12(–4)		1.86(–4)		6.86
			(41, 41)	7.53(–6)	3.932	7.34(–6)	3.937	1.18(–5)	3.973	23.16
HDM	6	8	(21, 21)	1.83(–6)		9.92(–7)		1.50(–6)		11.28
			(41, 41)	5.50(–10)	11.697	2.48(–10)	11.967	4.00(–10)	11.868	34.73
	8	8	(21, 21)	1.84(–6)		9.98(–7)		1.51(–6)		15.05
			(41, 41)	5.50(–10)	11.706	2.48(–10)	11.976	4.00(–10)	11.880	41.56

For all two DM schemes, $M_y = 8$ and $M_x = m$. CPU time in seconds is reported.

Table 13
The L_2 errors of the FDTD method at time $t = 1$ with $\Delta t = 2.0 \times 10^{-4}$

Scheme	M_x	l	(N_x, N_y)	E_z		H_x		H_y		CPU
				Error	Rate	Error	Rate	Error	Rate	
Q4DM	2	–	(21, 41)	2.25(–3)		8.55(–4)		7.56(–4)		15.62
			(41, 81)	1.66(–4)	3.759	6.11(–5)	3.808	4.67(–5)	4.017	56.50
HDM	6	8	(21, 41)	2.10(–4)		2.25(–4)		1.97(–4)		24.18
			(41, 81)	5.16(–8)	11.991	4.97(–8)	12.146	4.41(–8)	12.123	75.29
	8	8	(21, 41)	2.11(–4)		2.26(–4)		1.97(–4)		31.74
			(41, 81)	5.11(–8)	12.015	4.96(–8)	12.157	4.40(–8)	12.129	91.61

High frequency parameter setting is used. For all two DM schemes, $M_y = 12$ and $M_x = m$. CPU time in seconds is reported.

12 and 13. Therefore, the HDM method well balances the tradeoff between robustness and accuracy, and is a better time domain approach in terms of both properties.

5. Conclusion

In the present work, the well-known difficulty [4] of constructing Maxwell's equation solvers that are of higher order than previous fourth-order embedding finite difference time-domain (FDTD) methods [4,48–51] at material interfaces has been overcome. A systematic procedure via a series of novel hierarchical derivative matching (HDM) is proposed to construct numerical schemes that greatly exceed the convergence rate of the previous fourth-order methods. Our studies start from one-dimensional (1D) electromagnetic applications. The fundamentals of the derivative matching (DM) are discussed in detail. An implicit derivative matching (IDM) method is proposed as a general high accuracy time-domain approach, based on a simple structured or Cartesian grid. To address the potential stability problems arising in numerical studies, comprehensive stability analyses are considered for time-domain computations and the IDM modeling. A novel HDM is then proposed to achieve better stability and robustness. Numerical tests are carried out to validate new formulations and schemes. Extensions of the DM to two-dimensional (2D) electromagnetic problems are studied in detail. Two new schemes, a quasi-fourth-order derivative matching (Q4DM) scheme and a 2D HDM method are presented. The rationale, stability, merits, and disadvantages of each method are analyzed in detail. Such analysis has significantly extended our understanding of availability and limitation of

high order embedded FDTD methods. Similar to the previous fourth-order embedding FDTD methods, a single staggered, structured grid is typically employed for the present 2D HDM method. However, since the central finite difference scheme is utilized in the whole domain, the proposed schemes reduce to the standard high-order FDTD methods without material interfaces. Several inhomogeneous interface problems are considered to demonstrate the numerical performances of the two 2D schemes.

It is well known that the direct time-domain electromagnetic computation in inhomogeneous media exhibits only first order of accuracy, even though a high-order spatial discretization scheme is used [4]. To restore the high accuracy of a high-order spatial approximation scheme, subtle numerical modelings near material interfaces are indispensable for addressing the loss of regularity of field components across the interfaces [4]. In computational electromagnetics (CEM), several interesting interface schemes have been manually constructed [4,39,40,45,46,48–51]. However, the extension of these promising schemes to arbitrarily high order would be quite empirical and mathematically complicated [4] if it is not impossible. Motivated by a novel explicit DM method presented in [39,40], an IDM method is introduced in the present study. By correctly enforcing the physical jump condition at material interfaces, the IDM method can be applied to general time-domain approaches, such as the multiresolution time-domain method and the local spectral time domain method to restore high-order accuracy. The modeling of the IDM results in local modifications of the differential stencils near the interfaces, similar to the embedding FDTD schemes [4,45,46,48–51]. Therefore, the IDM method yields an excellent way to systematically generalize the embedding FDTD scheme to higher orders. Compared with the explicit DM method in [39,40], the IDM method could be more efficient for long time wave simulations, and can be applied to more general electromagnetic applications, e.g., it can be used in an implicit time-stepping scheme or numerical solution of frequency-domain CEM problems. Furthermore, the analysis of stability issues of the IDM method can be easier than that of the explicit DM, due to the implicit nature of the IDM. A comprehensive stability analysis of the IDM method is carried out in the present work. However, a direct IDM modeling is found to be quite difficult in 2D studies, due to the presence of mixed derivatives. As a result, only a Q4DM scheme is constructed in 2D by using the IDM approach.

A novel HDM method is also introduced in the present paper. In 1D cases, the HDM method can be viewed as a good supplement to the IDM method in some sense. In particular, the HDM method is computationally more efficient and robust than the IDM method for problems requiring a large number of fictitious points. Thus, it is well suited for large scale electromagnetic problems. The HDM method is also found to be more stable than the IDM method so that it can be easily applied to various high accuracy time-domain Maxwell solvers. The importance of the HDM method becomes more significant in 2D simulations. The 2D HDM is constructed by using high order one-sided approximations and the first two sets of DM conditions. In 2D cases, the hierarchical modeling is found to be the best approach for the purpose of designing ultra high order DM schemes for general time-domain Maxwell's solver. By employing one-sided approximations, the proposed 2D HDM method can achieve up to 12th order of accuracy in numerical computations and is based on a simple structured grid. In terms of accuracy, cost-efficiency, stability and applicability, the HDM method is the best DM method for generating ultra high order 2D embedding FDTD methods. For 3D real applications, the use of HDM method is expected to be more promising.

The main purpose of this paper is to introduce the hierarchical DM methods for the purpose of achieving high-order accuracy in numerical solution of time-domain Maxwell's equations with material interfaces in the framework of FDTD. Many important issues have not been touched, although possible extensions of the present work are briefly discussed all over the text. A number of promising generalizations of the present work, including 2D TE solvers, 3D solvers, frequency domain solvers, complex-domain solvers, and curved boundary and interface solvers, are under our consideration.

Acknowledgements

This work was supported by Michigan State University. The authors thank Jan Hesthaven and Wei Cai for providing preprint [4, 46], respectively. Special thanks go to Gang Bao and Bengt Fornberg for comments and discussions.

References

- [1] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antennas Propagat.* 14 (1966) 302.
- [2] A. Taflove, S.C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, Boston, London, 2000.
- [3] D.W. Zingg, Comparison of high-accuracy finite difference methods for linear wave propagation, *SIAM J. Sci. Comput.* 22 (2000) 476.
- [4] J.S. Hesthaven, High-order accurate methods in time-domain computational electromagnetics. A review, *Adv. Imag. Electron Phys.* 127 (2003) 59.
- [5] R. Holland, Finite difference solutions of Maxwell's equations in generalized nonorthogonal coordinates, *IEEE Trans. Nucl. Sci.* 30 (1983) 4589.
- [6] K.S. Yee, J.S. Chen, A.H. Chang, Conformal finite difference time-domain (FDTD) with overlapping grids, *IEEE Trans. Antennas Propagat.* 40 (1992) 1068.
- [7] T.G. Jurgens, A. Taflove, K. Umashaankar, T.G. Moore, Finite difference time-domain modeling of curved surfaces, *IEEE Trans. Antennas Propagat.* 40 (1992) 357.
- [8] P. Monk, E. Suli, Error Estimates of Yee's method on non-uniform grid, *IEEE Trans. Magn.* 30 (1994) 393.
- [9] A. Taflove, *Computational Electrodynamics. The Finite Difference Time-Domain Method*, Aztech House, Boston, 1995.
- [10] J. Fang, Time domain finite difference computation for Maxwell's equations, Ph.D. dissertation, Department of Electrical Engineering, University of California, Berkeley, CA, 1989.
- [11] P.G. Petropoulos, Phase error control for FD-TD methods of second and fourth order accuracy, *IEEE Trans. Antennas Propagat.* 42 (1994) 859.
- [12] D.W. Zingg, H. Lomax, H. Jurgens, High-accuracy finite-difference schemes for linear wave propagation, *SIAM J. Sci. Comput.* 17 (1996) 328.
- [13] J.L. Young, D. Gaitonde, J.S. Shang, Towards the construction of a fourth order difference scheme for transient EM wave simulation: Staggered grid approach, *IEEE Trans. Antennas Propagat.* 45 (1997) 1573.
- [14] J.S. Shang, High order compact-difference schemes for time-dependent Maxwell equations, *J. Comput. Phys.* 153 (1998) 312.
- [15] E. Turkel, High order methods, in: A. Taflove (Ed.), *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, Boston, MA, 1998, Chapter 2.
- [16] E. Turkel, A. Yefet, On the construction of a high order difference scheme for complex domains in a Cartesian grid, *Appl. Numer. Math.* 33 (2000) 113.
- [17] M. Krumpolz, L.P.B. Katehi, MRTD: New time-domain schemes based on multiresolution analysis, *IEEE Trans. Microwave Theory Tech.* 44 (1996) 555.
- [18] L.P.B. Katehi, J.F. Harvey, E. Tentzeris, Time-domain analysis using multiresolution expansions, in: A. Taflove (Ed.), *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, Boston, MA, 1998, Chapter 3.
- [19] J.P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, *J. Comput. Phys.* 114 (1994) 185.
- [20] Q.H. Liu, The PSTD algorithm: A time-domain method requiring only two cells per wavelength, *Microwave Opt. Techn. Lett.* 15 (1997) 158.
- [21] Q.H. Liu, Large-scale simulations of electromagnetic and acoustic measurements using the pseudospectral time-domain (PSTD) algorithm, *IEEE Trans. Geosci. Remote Sensing* 37 (1999) 917.
- [22] G.W. Wei, Discrete singular convolution for the solution of the Fokker–Planck equation, *J. Chem. Phys.* 110 (1999) 8930.
- [23] G.W. Wei, A unified method for solving Maxwell's equation, in: *Proceedings of 1999 Asia-Pacific Microwave Conference*, Singapore, 1999, pp. 562–565.
- [24] S. Zhao, Aspects of discrete singular convolution for scientific and engineering computing, Ph.D. dissertation, Department of Computational Science, National University of Singapore, Singapore, 2003.
- [25] G. Bao, G.W. Wei, S. Zhao, Local spectral time-domain method for electromagnetic wave propagation, *Opt. Lett.* 28 (2003) 513.
- [26] Z.H. Shao, G.W. Wei, S. Zhao, DSC time-domain solution of Maxwell's equations, *J. Comput. Phys.* 189 (2003) 427.
- [27] Z.H. Shao, Z. Shen, Q. He, G.W. Wei, A generalized higher-order finite difference time domain method and its application in guided-wave problems, *IEEE T. Microwave Theory Tech.* 51 (2003) 856.

- [28] A.H. Mohammadian, V. Shankar, W.F. Hall, Computation of electromagnetic scattering and radiation using a time-domain finite-volume discretization procedure, *Comput. Phys. Commun.* 68 (1991) 175.
- [29] J.S. Shang, D. Gaitonde, Characteristic-based time-dependent Maxwell equation solvers on a general curvilinear frame, *AIAA J.* 33 (1995) 491.
- [30] K.S. Yee, J.S. Chen, The finite-difference time-domain (FDTD) and the finite-volume time-domain (FVTD) methods in solving Maxwell's equations, *IEEE Trans. Antennas Propagat.* 45 (1997) 354.
- [31] J.C. Nedelec, Mixed finite elements in R^3 , *Numer. Math.* 35 (1980) 315.
- [32] A.C. Cangellaris, C.C. Lin, K.K. Mei, Point-matched time-domain finite element methods for electromagnetic radiation and scattering, *IEEE Trans. Antennas Propagat.* 35 (1987) 1160.
- [33] J.F. Lee, R. Lee, A. Cangellaris, Time-domain finite-element methods, *IEEE Trans. Antennas Propagat.* 45 (1997) 430.
- [34] J.M. Jin, M. Zunoubi, K.C. Donepudi, W.C. Chew, Frequency-domain and time-domain finite-element solution of Maxwell's equations using spectral Lanczos decomposition method, *Comput. Meth. Appl. Mech. Eng.* 169 (1999) 279.
- [35] A. Bossavit, A rationale for "Edge-Elements" in 3D fields computations, *IEEE Trans. Magn.* 24 (1998) 74.
- [36] R. Hiptmair, Higher order Whitney forms, *Progr. Electromagn. Res. PIER* 32 (2001) 271.
- [37] D.R. Wilton, A.F. Peterson, R.G. Graglia, High order interpolatory vector bases for computational electromagnetics, *IEEE Trans. Antennas Propagat.* 45 (1997) 329.
- [38] J.S. Hesthaven, T. Warburton, Nodal high order methods on unstructured grids. I. Time-domain solution of Maxwell's equations, *J. Comput. Phys.* 181 (2002) 186.
- [39] T.A. Driscoll, B. Fornberg, A block pseudospectral method for Maxwell's equations, I. one-dimensional case, *J. Comput. Phys.* 140 (1998) 47.
- [40] T.A. Driscoll, B. Fornberg, Block pseudospectral methods for Maxwell's equations, II. two-dimensional, discontinuous-coefficient case, *SIAM J. Sci. Comput.* 21 (1999) 1146.
- [41] B. Yang, D. Gottlieb, J.S. Hesthaven, Spectral simulations of electromagnetics, *J. Comput. Phys.* 134 (1997) 216.
- [42] B. Yang, J.S. Hesthaven, A pseudospectral method for time-domain computation of electromagnetic scattering by bodies of revolution, *IEEE Trans. Antennas Propagat.* 47 (1999) 132.
- [43] B. Yang, J.S. Hesthaven, Multidomain pseudospectral computation of Maxwell's equations in 3-D general curvilinear coordinates, *Appl. Numer. Math.* 33 (2000) 281.
- [44] H.M. Jurgens, D.W. Zingg, Numerical solution of the time-domain Maxwell equations using high-accuracy finite-difference methods, *SIAM J. Sci. Comput.* 22 (2000) 1675.
- [45] A. Ditkowski, K. Dridi, J.S. Hesthaven, Convergent Cartesian grid methods for Maxwell's equations in complex geometries, *J. Comput. Phys.* 170 (2001) 39.
- [46] W. Cai, S.Z. Deng, An upwinding embedded boundary method for Maxwell's equations in media with material interfaces: 2D case, *J. Comput. Phys.* 190 (2003) 159.
- [47] C.M. Zhang, R.J. LeVeque, The immersed interface method for acoustic wave equations with discontinuous coefficients, *Wave Motion* 27 (1997) 237.
- [48] A. Yefet, E. Turkel, Fourth order compact implicit method for the Maxwell equations with discontinuous coefficients, *Appl. Numer. Math.* 33 (2000) 125.
- [49] A. Yefet, P.G. Petropoulos, A non-dissipative staggered fourth order accurate explicit finite difference scheme for the time-domain Maxwell's equations, *J. Comput. Phys.* 168 (2001) 286.
- [50] Z.Q. Xie, C.-H. Chan, B. Zhang, An explicit fourth order staggered finite-difference time-domain method for Maxwell's equations, *J. Comput. Appl. Math.* 147 (2002) 75.
- [51] Z.Q. Xie, C.-H. Chan, B. Zhang, An explicit fourth order orthogonal curvilinear staggered-grid FDTD method for Maxwell's equations, *J. Comput. Phys.* 175 (2002) 739.
- [52] Y. Wang, Y.B. Zhao, G.W. Wei, A note on the numerical solution of high order differential equations, *J. Comput. Appl. Math.* 159 (2003) 387.
- [53] G.W. Wei, Y.B. Zhao, Y. Xiang, Discrete singular convolution and its application to the analysis of plates with internal supports. I Theory and algorithm, *Int. J. Numer. Meth. Eng.* 55 (2002) 913.
- [54] G. Bao, G.W. Wei, S. Zhao, Numerical solution of the Helmholtz equation with high wavenumbers, *Int. J. Numer. Methods Eng.* 59 (2004) 389.
- [55] G.W. Wei, D.S. Zhang, D.J. Kouri, D.K. Hoffman, Lagrange distributed approximating functionals, *Phys. Rev. Lett.* 79 (1997) 775.
- [56] A. Bayliss, E. Turkel, Radiation boundary conditions for wave-like equations, *Comm. Pure Appl. Math.* 33 (1980) 707.
- [57] R. Ziokowski, Time-derivative Lorentz-material model-based absorbing boundary condition, *IEEE Trans. Antennas Propagat.* 45 (1997) 1530.
- [58] S. Abarbanel, D. Gottlieb, On the construction and analysis of absorbing layers in CEM, *Appl. Numer. Math.* 27 (1998) 331.
- [59] B. Fornberg, An improved pseudospectral method for initial-boundary value problems, *J. Comput. Phys.* 91 (1990) 381.
- [60] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, 1996.
- [61] B. Fornberg, Calculation of weights in finite difference formulas, *SIAM Rev.* 40 (1998) 685.