Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods

Tan, Eng Leong

2008

Tan, E. L. (2008). Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods. IEEE Transactions on Antennas and Propagation, 56(1), 170-177. doi:10.1109/TAP.2007.913089

https://hdl.handle.net/10356/138249

https://doi.org/10.1109/TAP.2007.913089

Fundamental Schemes for Efficient Unconditionally Stable Implicit Finite-Difference Time-Domain Methods

Eng Leong Tan, *Senior Member, IEEE*

***Abstract*—Generalized formulations of fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain (FDTD) methods are presented. The fundamental schemes constitute a family of implicit schemes that feature similar fundamental updating structures, which are in simplest forms with most efficient right-hand sides. The formulations of fundamental schemes are presented in terms of generalized matrix operator equations pertaining to some classical splitting formulae, including those of alternating direction implicit, locally one-dimensional and split-step schemes. To provide further insights into the implications and significance of fundamental schemes, the analyses are also extended to many other schemes with distinctive splitting formulae. Detailed algorithms are described for new efficient implementations of the unconditionally stable implicit FDTD methods based on the fundamental schemes. A comparative study of various implicit schemes in their original and new implementations is carried out, which includes comparisons of their computation costs and efficiency gains.**

***Index Terms*—Finite-difference time-domain (FDTD) methods, unconditionally stable methods, implicit schemes, alternating direction implicit (ADI) scheme, locally one-dimensional (LOD) scheme, split-step approach, computational electromagnetics.**

I. Introduction

The finite-difference time-domain (FDTD) method has been widely used to obtain the numerical solutions of Maxwell's equations for investigating electromagnetic wave radiation, propagation and scattering problems [1]. For the conventional explicit FDTD method [2], the computational efficiency is restricted by the Courant-Friedrichs-Lewy (CFL) stability condition, which imposes a maximum constraint on the time step size depending on the spatial mesh sizes. To remove the CFL condition, the unconditionally stable FDTD method based on the alternating direction implicit (ADI) technique has been developed [3], [4]. The success of ADI-FDTD method has brought about a resurgence of interest in the unconditionally stable schemes not only within the electromagnetics community, but also among other scientists and mathematicians at large. With its unconditional stability advantage, the ADI-FDTD method has been extensively analyzed, improved and extended for many applications. Since the literature in these aspects is so voluminous, we could have easily missed out many references that are not directly relevant. For more comprehensive survey, we refer the readers to the bibliography in [1], [5], [6] (see also those cited in the author's previous works).

The author is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: eeltan@ntu.edu.sg).

The idea of ADI scheme that has been adapted in the recent celebrated ADI-FDTD, can be traced back to the early classic works by Peaceman and Rachford [7]. Apart from the ADI scheme, many alternative implicit schemes have also been introduced by researchers in applied mathematics to deal with various parabolic, elliptic and hyperbolic partial differential systems. Some of such schemes that are more common and closely related to the present scope include locally one-dimensional (LOD) and Crank-Nicolson schemes [8]-[11]. These schemes have been adapted as well for FDTD solutions of Maxwell's equations, leading to unconditionally stable LOD-FDTD method [12]-[13], split-step FDTD approach [14]-[15] and other Crank-Nicolson-based approximation methods [16]. Most of these unconditionally stable FDTD methods have been built upon the classical implicit schemes by adopting their respective splitting formulae directly.

Despite the successful adaptation of various classical schemes, continuing efforts are underway to devise new stable methods that are more efficient and simpler to implement. Recently, a new efficient algorithm has been presented for the ADI-FDTD method [17]. The algorithm involves updating equations whose right-hand sides are much simpler and more concise than the conventional implementation [3], [4]. This leads to substantial reduction in the number of arithmetic operations required for their computations. While the underlying principle of the new algorithm has helped make the ADI-FDTD simpler and more efficient, it actually has greater significance in its own right. This paper aims to clarify and extend such principle to arrive at a series of new efficient algorithms for various other unconditionally stable implicit FDTD methods, including some of those mentioned above. Moreover, it will be found that the resultant algorithms constitute a family of implicit schemes, all of which feature similar fundamental updating structures that are in simplest forms with most efficient right-hand sides.

The organization of this paper is as follows. Section II presents the formulations of fundamental schemes in terms of generalized matrix operator equations pertaining to some classical splitting formulae, including those of ADI, LOD and split-step schemes. To provide further insights into the implications and significance of fundamental schemes, the analyses are also extended to many other schemes with distinctive splitting formulae. In Section III, detailed algorithms are described based on the fundamental schemes for new efficient

E. L. Tan, "Fundamental Schemes for Efficient Unconditionally Stable Implicit Finite-Difference Time-Domain Methods," IEEE Transactions on Antennas and Propagation, Vol. 56, No. 1, pp. 170-177, January 2008. DOI: 10.1109/TAP.2007.913089

implementations of the unconditionally stable implicit FDTD methods, e.g. ADI-FDTD and LOD-FDTD. In Section IV, a comparative study of various implicit schemes in their original and new implementations is carried out, which includes comparisons of their computation costs and efficiency gains. The fundamental nature of new implementations will also become evident through the comparisons and discussions.

II. Generalized Formulations

In this section, we present the generalized formulations of fundamental schemes for implicit finite-difference methods. Starting from some classical implicit schemes, their generalized matrix operator equations are revisited and reformulated in the simplest and most efficient forms. These new forms feature convenient matrix-operator-free right-hand sides with least number of terms, which will lead to coding simplification in their algorithm implementations. For simplicity and clarity, we omit the nonhomogeneous terms that appear merely as vectors (without involving matrices).

A. ADI

The ADI scheme, originated by Peaceman and Rachford [7], is one of the most popular implicit finite-difference schemes in use today. This scheme calls for generalized splitting formulae in the form

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n} \tag{1a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+1}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}}. \tag{1b}$$

For many decades, such splitting formulae form the basis of many other numerical methods, which include the recent unconditionally stable ADI-FDTD method [3], [4]. Note that the specific matrix operators $\mathbf{A}$ and $\mathbf{B}$ of [3], [4] (for 3-D Maxwell) are different from those of [7] (for 2-D parabolic), even though they conform to the same generalized form (1). In this section, we shall let these matrices be general but would caution that one must choose their operators properly for a particular scheme to stay unconditionally stable [16]. To implement the ADI algorithm, it is more convenient to introduce auxiliary variables for denoting the right-hand sides of implicit equations. This allows us to rewrite the original algorithm as

$$\mathbf{v}^{n}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n} \tag{2a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}}=\mathbf{v}^{n} \tag{2b}$$

$$\mathbf{v}^{n+\frac{1}{2}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}} \tag{2c}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+1}=\mathbf{v}^{n+\frac{1}{2}} \tag{2d}$$

where the $\mathbf{v}$'s serve as the auxiliary variables.

If one exploits these auxiliary variables, the original algorithm above can be modified into one of the more efficient scheme. In particular, based on (2d) at one time step backward:

$$\mathbf{v}^{n-\frac{1}{2}}=\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n}, \tag{3}$$

it follows that $\mathbf{v}^n$ of (2a) is reducible to

$$\mathbf{v}^{n}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n} \tag{4a}$$

$$=2\mathbf{u}^{n}-\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n} \tag{4b}$$

$$=2\mathbf{u}^{n}-\mathbf{v}^{n-\frac{1}{2}}. \tag{4c}$$

Furthermore, upon recognizing (2b), $\mathbf{v}^{n+\frac{1}{2}}$ of (2c) is also reducible to

$$\mathbf{v}^{n+\frac{1}{2}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}} \tag{5a}$$

$$=2\mathbf{u}^{n+\frac{1}{2}}-\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}} \tag{5b}$$

$$=2\mathbf{u}^{n+\frac{1}{2}}-\mathbf{v}^{n}. \tag{5c}$$

With (4c) and (5c), algorithm (2) becomes more efficient as

$$\mathbf{v}^{n}=2\mathbf{u}^{n}-\mathbf{v}^{n-\frac{1}{2}} \tag{6a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}}=\mathbf{v}^{n} \tag{6b}$$

$$\mathbf{v}^{n+\frac{1}{2}}=2\mathbf{u}^{n+\frac{1}{2}}-\mathbf{v}^{n} \tag{6c}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+1}=\mathbf{v}^{n+\frac{1}{2}}. \tag{6d}$$

Through a simple re-definition of field variables

$$\tilde{\mathbf{u}}^{n}=2\mathbf{u}^{n},\quad \tilde{\mathbf{u}}^{n+\frac{1}{2}}=2\mathbf{u}^{n+\frac{1}{2}}, \tag{7}$$

we may reduce the algorithm further into [17]

$$\mathbf{v}^{n}=\tilde{\mathbf{u}}^{n}-\mathbf{v}^{n-\frac{1}{2}} \tag{8a}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{A}\Big)\tilde{\mathbf{u}}^{n+\frac{1}{2}}=\mathbf{v}^{n} \tag{8b}$$

$$\mathbf{v}^{n+\frac{1}{2}}=\tilde{\mathbf{u}}^{n+\frac{1}{2}}-\mathbf{v}^{n} \tag{8c}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\tilde{\mathbf{u}}^{n+1}=\mathbf{v}^{n+\frac{1}{2}}. \tag{8d}$$

This algorithm proceeds in terms of $\tilde{\mathbf{u}}$'s for its main iterations, and only when the field output is required, one may retrieve the data from

$$\mathbf{u}^{n+1}=\frac{1}{2}\tilde{\mathbf{u}}^{n+1}. \tag{9}$$

Equations (8a)-(8d) constitute the most efficient ADI scheme that has the simplest right-hand sides without involving any explicit matrix operator. For nonzero initial fields, the algorithm takes the input initialization

$$\mathbf{v}^{-\frac{1}{2}}=\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{0}. \tag{10}$$

Note that such initialization will not degrade the accuracy of ADI scheme since it simply corresponds to (3) with $n=0$ or (2d) with $n=-1$. Furthermore, (3) along with (7)-(8b) are just (2a)-(2b) or (1a). Hence the equivalence of both present and original (generalized) Peaceman-Rachford schemes becomes evident here.

B. LOD1/SS1

The LOD scheme, introduced in the early Russian literature (cf. [8], [10]), is another classical scheme that has been used extensively. This scheme calls for generalized splitting formulae in the form

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n} \tag{11a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+1}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+\frac{1}{2}}. \tag{11b}$$

Such splitting formulae have been adopted to develop the recent unconditionally stable LOD-FDTD method [12] and split-step approach [14]. When $\mathbf{A}$ and $\mathbf{B}$ do not commute, the scheme is accurate to first order in time. Thus, the resultant LOD-FDTD method and split-step approach may be denoted by LOD1 and SS1 respectively.

For most efficiency, the original LOD scheme can be modified into simplest form with matrix-operator-free right-hand sides. From (11a), we have

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n} \tag{12a}$$

$$=2\mathbf{u}^{n}-\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n}. \tag{12b}$$

This can be manipulated readily to give

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\Big(\mathbf{u}^{n+\frac{1}{2}}+\mathbf{u}^{n}\Big)=2\mathbf{u}^{n} \tag{13}$$

where the vector terms in bracket may be denoted by auxiliary variable

$$\mathbf{v}^{n+\frac{1}{2}}=\mathbf{u}^{n+\frac{1}{2}}+\mathbf{u}^{n}. \tag{14}$$

Similar manipulation applies to (11b) which leads to auxiliary variable $\mathbf{v}^{n+1}$. Combining all auxiliary and field variables, we arrive at

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{A}\Big)\mathbf{v}^{n+\frac{1}{2}}=\mathbf{u}^{n} \tag{15a}$$

$$\mathbf{u}^{n+\frac{1}{2}}=\mathbf{v}^{n+\frac{1}{2}}-\mathbf{u}^{n} \tag{15b}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{v}^{n+1}=\mathbf{u}^{n+\frac{1}{2}} \tag{15c}$$

$$\mathbf{u}^{n+1}=\mathbf{v}^{n+1}-\mathbf{u}^{n+\frac{1}{2}}. \tag{15d}$$

In these equations, all their right-hand sides are seen to be in the most convenient matrix-operator-free form. Furthermore, there is no special input initialization required and the output field solution is directly available from (15d).

C. SS2

While the former SS1 approach is only first-order accurate in time, the general split-step approach actually permits simple extensions to achieve higher-order temporal accuracy [14]. Consider the split-step approach of second-order accuracy denoted by SS2. It is based on the Strang splitting formulae [9] and involves three updating procedures

$$\Big(\mathbf{I}-\frac{\Delta t}{4}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{4}}=\Big(\mathbf{I}+\frac{\Delta t}{4}\mathbf{A}\Big)\mathbf{u}^{n} \tag{16a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+\frac{3}{4}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+\frac{1}{4}} \tag{16b}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{4}\mathbf{A}\Big)\mathbf{u}^{n+1}=\Big(\mathbf{I}+\frac{\Delta t}{4}\mathbf{A}\Big)\mathbf{u}^{n+\frac{3}{4}}. \tag{16c}$$

In line with the previous subsection, the algorithm implementation can be made much simpler along with improved efficiency by resorting to

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{8}\mathbf{A}\Big)\mathbf{v}^{n+\frac{1}{4}}=\mathbf{u}^{n} \tag{17a}$$

$$\mathbf{u}^{n+\frac{1}{4}}=\mathbf{v}^{n+\frac{1}{4}}-\mathbf{u}^{n} \tag{17b}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{v}^{n+\frac{3}{4}}=\mathbf{u}^{n+\frac{1}{4}} \tag{17c}$$

$$\mathbf{u}^{n+\frac{3}{4}}=\mathbf{v}^{n+\frac{3}{4}}-\mathbf{u}^{n+\frac{1}{4}} \tag{17d}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{8}\mathbf{A}\Big)\mathbf{v}^{n+1}=\mathbf{u}^{n+\frac{3}{4}} \tag{17e}$$

$$\mathbf{u}^{n+1}=\mathbf{v}^{n+1}-\mathbf{u}^{n+\frac{3}{4}}. \tag{17f}$$

Again, we have the right-hand sides of these equations cast in the most convenient matrix-operator-free form. However, since there are three implicit equations to be dealt with, the scheme still involves more arithmetic operations than those having merely two implicit equations with matrix-operator-free right-hand sides.

D. LOD2

An alternative LOD-FDTD method denoted by LOD2 has been presented, which does not exhibit the non-commutativity error and preserves the second-order temporal accuracy [13]. In essence, the main iterations proceed with two updating procedures in the same way like LOD1, cf. (11a)-(11b). However, the time indices are associated with advancement of time steps from $n+1/4$ to $n+3/4$ and from $n+3/4$ to $n+1(1/4)$ as

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{3}{4}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{4}} \tag{18a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+1\frac{1}{4}}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+\frac{3}{4}}. \tag{18b}$$

To relate the fields to those normally at integer time steps, such association will prompt for additional processing for the input field data

$$\Big(\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{u}^{\frac{1}{4}}=\Big(\mathbf{I}+\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{u}^{0} \tag{19}$$

as well as for the output field data

$$\Big(\mathbf{I}+\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{u}^{n+1}=\Big(\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{u}^{n+1\frac{1}{4}}. \tag{20}$$

Note that the input processing is to be invoked only once at the beginning, while for the output processing at $n+1$, it may be performed separately and independently in parallel without disrupting the main iterations. Furthermore, one often does not need to frequently output all field components, except probably one or two of interest at few desired observation points after certain (fairly long) interval periodically.

The LOD2 algorithm above, with its facilitation for parallel/reduced/infrequent output processing, has been found to be more efficient than those of ADI and SS2 methods in their original implementations [13]. To improve the efficiency further, we perform simplifications like previous subsections to arrive at matrix-operator-free right-hand sides for the main iterations as

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{A}\Big)\mathbf{v}^{n+\frac{3}{4}}=\mathbf{u}^{n+\frac{1}{4}} \tag{21a}$$

$$\mathbf{u}^{n+\frac{3}{4}}=\mathbf{v}^{n+\frac{3}{4}}-\mathbf{u}^{n+\frac{1}{4}} \tag{21b}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{v}^{n+1\frac{1}{4}}=\mathbf{u}^{n+\frac{3}{4}} \tag{21c}$$

$$\mathbf{u}^{n+1\frac{1}{4}}=\mathbf{v}^{n+1\frac{1}{4}}-\mathbf{u}^{n+\frac{3}{4}}. \tag{21d}$$

If desired, similar simplifications may be performed for the input processing

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{8}\mathbf{B}\Big)\mathbf{v}^{\frac{1}{4}}=\mathbf{u}^{0} \tag{22a}$$

$$\mathbf{u}^{\frac{1}{4}}=\mathbf{v}^{\frac{1}{4}}-\mathbf{u}^{0} \tag{22b}$$

and output processing

$$\Big(\frac{1}{2}\mathbf{I}+\frac{\Delta t}{8}\mathbf{B}\Big)\mathbf{v}^{n+1}=\mathbf{u}^{n+1\frac{1}{4}} \tag{23a}$$

$$\mathbf{u}^{n+1}=\mathbf{v}^{n+1}-\mathbf{u}^{n+1\frac{1}{4}}. \tag{23b}$$

Equations (21a)-(21d) constitute the most efficient LOD2 scheme which is comparable to that of (8a)-(8d).

E. Fundamental Implications and Significance

It turns out that the new algorithm implementations presented above represent some very fundamental ones for implicit finite-difference methods. As is evident from (8) and (15) or (21), these schemes merely involve similar updating structures, namely implicit updating with (simplest) matrix-operator-free right-hand sides and explicit updating via subtraction of two vectors. Their main slight difference might just be the sequence of implicit and explicit updatings. Moreover, one may convert from one scheme to another simply by proper initialization and swapping the roles of field and auxiliary variables. For instance, treating $\mathbf{v}^n$ as the field variables (uncontaminated to second order) in (15) and using

$$\mathbf{u}^{n}=\Big(\frac{1}{2}\mathbf{I}+\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{v}^{n} \tag{24}$$

to initialize its iterations, one may obtain the $\mathbf{v}^{n+1}$ field solutions from (15c) that correspond to the $\mathbf{u}^{n+1}$ field solutions from (8)-(9). Such interesting but subtle link between LOD and ADI schemes has been pointed out earlier [18]. Through the fundamental schemes herein, the link becomes particularly obvious with their similar updating structures and with the aid of variables that are otherwise not defined in the original schemes, e.g. $\mathbf{v}$ is not directly seen in (11).

The implications of the fundamental schemes above are actually more far-reaching. In particular, we note that many other classical and recent non-dissipative splitting schemes, such as D'Yakonov scheme, delta formulation, Crank-Nicolson direct-splitting method etc. may be cast into the same simplest form of (8) featuring matrix-operator-free right-hand sides. Such analyses when extended to these schemes with distinctive splitting formulae, will give further insights into the significance of fundamental schemes. Let us illustrate this point using the scheme with splitting formulae (cf. generalized matricization of D'Yakonov or Beam-Warming scheme)

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\mathbf{u}^{*}=\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A}\Big)\Big(\mathbf{I}+\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n} \tag{25a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\mathbf{u}^{n+1}=\mathbf{u}^{*}. \tag{25b}$$

Such scheme may be reduced to an efficient one with the iterations algorithm read as

$$\mathbf{v}^{n}=\mathbf{u}^{n}-\mathbf{v}^{n-\frac{1}{2}} \tag{26a}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{A}\Big)\mathbf{u}^{n+\frac{1}{2}}=\mathbf{v}^{n} \tag{26b}$$

$$\mathbf{v}^{n+\frac{1}{2}}=\mathbf{u}^{n+\frac{1}{2}}-\mathbf{v}^{n} \tag{26c}$$

$$\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{u}^{n+1}=\mathbf{v}^{n+\frac{1}{2}} \tag{26d}$$

along with input initialization

$$\mathbf{v}^{-\frac{1}{2}}=\Big(\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\mathbf{B}\Big)\mathbf{u}^{0}. \tag{27}$$

Notice that the form of (26) is just that of (8), while noting also the relation

$$\mathbf{u}^{*}=2\mathbf{v}^{n+\frac{1}{2}}. \tag{28}$$

Another illustration is the splitting formulae (cf. generalized matricization of 2D Douglas-Gunn scheme or delta formulation)

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A}\Big)\Delta\mathbf{u}^{*}=\Delta t\Big(\mathbf{A}+\mathbf{B}\Big)\mathbf{u}^{n} \tag{29a}$$

$$\Big(\mathbf{I}-\frac{\Delta t}{2}\mathbf{B}\Big)\Delta\mathbf{u}=\Delta\mathbf{u}^{*} \tag{29b}$$

$$\mathbf{u}^{n+1}=\mathbf{u}^{n}+\Delta\mathbf{u}. \tag{29c}$$

Again, this can be turned into the previous fundamental scheme, specifically (7)-(10) and with

$$\Delta\mathbf{u}^{*}=\tilde{\mathbf{u}}^{n+\frac{1}{2}}-\tilde{\mathbf{u}}^{n} \tag{30a}$$

$$\Delta\mathbf{u}=\frac{1}{2}\Big(\tilde{\mathbf{u}}^{n+1}-\tilde{\mathbf{u}}^{n}\Big). \tag{30b}$$

Meanwhile, the fundamental scheme of (15a)-(15b) forms the basis of simplification for many split-step methods of higher order accuracy in [14]. This will lead to their convenient algorithms all with matrix-operator-free right-hand sides following the same way that underpins (15c)-(15d), (17a)-(17f) and (21a)-(21d), etc. Even for the classical Crank-Nicolson scheme having the original unfactorized form

$$\Big[\mathbf{I}-\frac{\Delta t}{2}\Big(\mathbf{A}+\mathbf{B}\Big)\Big]\mathbf{u}^{n+1}=\Big[\mathbf{I}+\frac{\Delta t}{2}\Big(\mathbf{A}+\mathbf{B}\Big)\Big]\mathbf{u}^{n}, \tag{31}$$

one may based on similar manipulation to devise

$$\Big[\frac{1}{2}\mathbf{I}-\frac{\Delta t}{4}\Big(\mathbf{A}+\mathbf{B}\Big)\Big]\mathbf{u}^{n+\frac{1}{2}}=\mathbf{u}^{n} \tag{32a}$$

$$\mathbf{u}^{n+1}=\mathbf{u}^{n+\frac{1}{2}}-\mathbf{u}^{n}. \tag{32b}$$

This obviously constitutes the simpler algorithm than that of the original scheme.

It should be mentioned that the extent of simplification and actual improvement of computation efficiency for various fundamental schemes depend much on the particular matrix operators and algorithm implementation details (to be described next). Although the scope of this paper is mainly about the unconditionally stable implicit FDTD methods in electromagnetics, it is evident that all fundamental schemes discussed can be extended readily to many other finite-difference schemes in computational physics, even conditionally stable or parabolic/elliptic ones, etc. Some numerical methods that bear resemblance to the schemes above such as implicit integrations or iterative solutions of certain control problems may be made simpler and more efficient in the similar manner.

III. Efficient Implementations

In this section, we describe the detailed algorithms for new efficient implementations of the unconditionally stable implicit FDTD methods based on the fundamental schemes. For simplicity, consider the lossless isotropic medium with permittivity ϵ and permeability μ. The splitting matrix operators of Maxwell's equations may be selected as

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{\epsilon}\partial_y \\ 0 & 0 & 0 & \frac{1}{\epsilon}\partial_z & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\epsilon}\partial_x & 0 \\ 0 & \frac{1}{\mu}\partial_z & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\mu}\partial_x & 0 & 0 & 0 \\ \frac{1}{\mu}\partial_y & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{33}$$

$$\mathbf{B}=\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{-1}{\epsilon}\partial_z & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\epsilon}\partial_x \\ 0 & 0 & 0 & \frac{-1}{\epsilon}\partial_y & 0 & 0 \\ 0 & 0 & \frac{-1}{\mu}\partial_y & 0 & 0 & 0 \\ \frac{-1}{\mu}\partial_z & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{\mu}\partial_x & 0 & 0 & 0 & 0 \end{bmatrix} \tag{34}$$

where ∂_x, ∂_y, ∂_z are the spatial difference operators for the first derivatives along x, y, z directions respectively. The vector components of field and auxiliary variables can be written as

$$\mathbf{u}=\begin{bmatrix} E_x \\ E_y \\ E_z \\ H_x \\ H_y \\ H_z \end{bmatrix}, \quad \tilde{\mathbf{u}}=\begin{bmatrix} \tilde{E}_x \\ \tilde{E}_y \\ \tilde{E}_z \\ \tilde{H}_x \\ \tilde{H}_y \\ \tilde{H}_z \end{bmatrix}, \quad \mathbf{v}=\begin{bmatrix} e_x \\ e_y \\ e_z \\ h_x \\ h_y \\ h_z \end{bmatrix}. \tag{35}$$

For convenience, we also introduce the notations

$$b=\frac{\Delta t}{2\epsilon}, \quad d=\frac{\Delta t}{2\mu}. \tag{36}$$

A. ADI

Let us illustrate the implementation of fundamental ADI scheme (8). While equation (8a) can be implemented directly as

$$e_\xi^n=\tilde{E}_\xi^n-e_\xi^{n-\frac{1}{2}}, \quad \xi=x,y,z \tag{37a}$$
$$h_\xi^n=\tilde{H}_\xi^n-h_\xi^{n-\frac{1}{2}}, \quad \xi=x,y,z, \tag{37b}$$

equation (8b) may find more convenience through implicit updating

$$\frac{1}{2}\tilde{E}_x^{n+\frac{1}{2}}-\frac{bd}{2}\partial_y^2\tilde{E}_x^{n+\frac{1}{2}}=e_x^n+b\partial_y h_z^n \tag{38a}$$
$$\frac{1}{2}\tilde{E}_y^{n+\frac{1}{2}}-\frac{bd}{2}\partial_z^2\tilde{E}_y^{n+\frac{1}{2}}=e_y^n+b\partial_z h_x^n \tag{38b}$$
$$\frac{1}{2}\tilde{E}_z^{n+\frac{1}{2}}-\frac{bd}{2}\partial_x^2\tilde{E}_z^{n+\frac{1}{2}}=e_z^n+b\partial_x h_y^n, \tag{38c}$$

followed by explicit updating

$$\tilde{H}_x^{n+\frac{1}{2}}=2h_x^n+d\partial_z\tilde{E}_y^{n+\frac{1}{2}} \tag{39a}$$
$$\tilde{H}_y^{n+\frac{1}{2}}=2h_y^n+d\partial_x\tilde{E}_z^{n+\frac{1}{2}} \tag{39b}$$
$$\tilde{H}_z^{n+\frac{1}{2}}=2h_z^n+d\partial_y\tilde{E}_x^{n+\frac{1}{2}}. \tag{39c}$$

Similar arguments apply to equations (8c) and (8d), where the former is implemented directly as

$$e_\xi^{n+\frac{1}{2}}=\tilde{E}_\xi^{n+\frac{1}{2}}-e_\xi^n, \quad \xi=x,y,z \tag{40a}$$
$$h_\xi^{n+\frac{1}{2}}=\tilde{H}_\xi^{n+\frac{1}{2}}-h_\xi^n, \quad \xi=x,y,z, \tag{40b}$$

while the latter comprises implicit updating

$$\frac{1}{2}\tilde{E}_x^{n+1}-\frac{bd}{2}\partial_z^2\tilde{E}_x^{n+1}=e_x^{n+\frac{1}{2}}-b\partial_z h_y^{n+\frac{1}{2}} \tag{41a}$$
$$\frac{1}{2}\tilde{E}_y^{n+1}-\frac{bd}{2}\partial_x^2\tilde{E}_y^{n+1}=e_y^{n+\frac{1}{2}}-b\partial_x h_z^{n+\frac{1}{2}} \tag{41b}$$
$$\frac{1}{2}\tilde{E}_z^{n+1}-\frac{bd}{2}\partial_y^2\tilde{E}_z^{n+1}=e_z^{n+\frac{1}{2}}-b\partial_y h_x^{n+\frac{1}{2}}, \tag{41c}$$

and explicit updating

$$\tilde{H}_x^{n+1}=2h_x^{n+\frac{1}{2}}-d\partial_y\tilde{E}_z^{n+1} \tag{42a}$$
$$\tilde{H}_y^{n+1}=2h_y^{n+\frac{1}{2}}-d\partial_z\tilde{E}_x^{n+1} \tag{42b}$$
$$\tilde{H}_z^{n+1}=2h_z^{n+\frac{1}{2}}-d\partial_x\tilde{E}_y^{n+1}. \tag{42c}$$

When the difference operators above represent specifically the second-order central-differencing operators on Yee cells, equations (37)-(42) correspond to the efficient algorithm delineated in [17]. In particular, the solution $\tilde{E}_\xi^{n+1}$ from (41) coincides with $E_\xi^*|^{n+1}$ from [17, eqn. (11)], while the solution $\tilde{H}_\xi^{n+1}$ from (42) is half of $H_\xi^*|^{n+1}$ from [17, eqn. (12)]. The reason for such magnetic field relation is that certain variables and coefficients in (37)-(42) have been re-scaled in the corresponding auxiliary and explicit updating equations of [17]. This helps save the scaling of final magnetic field solution that is otherwise necessary, cf. (9). Moreover, additional savings of operations can be achieved by combining (39) and (40b) as

$$h_x^{n+\frac{1}{2}}=h_x^n+d\partial_z\tilde{E}_y^{n+\frac{1}{2}} \tag{43a}$$
$$h_y^{n+\frac{1}{2}}=h_y^n+d\partial_x\tilde{E}_z^{n+\frac{1}{2}} \tag{43b}$$
$$h_z^{n+\frac{1}{2}}=h_z^n+d\partial_y\tilde{E}_x^{n+\frac{1}{2}}. \tag{43c}$$

When the final magnetic field data is not needed frequently, (42) and (37b) (at $n+1$ time step) may also be combined for higher efficiency:

$$h_x^{n+1} = h_x^{n+\frac{1}{2}} - d\partial_y \tilde{E}_z^{n+1} \tag{44a}$$

$$h_y^{n+1} = h_y^{n+\frac{1}{2}} - d\partial_z \tilde{E}_x^{n+1} \tag{44b}$$

$$h_z^{n+1} = h_z^{n+\frac{1}{2}} - d\partial_x \tilde{E}_y^{n+1}. \tag{44c}$$

Other implementation details may be referred to [17] including for-looping, tridiagonal system solving, memory reuse etc. In actuality, there are many possibilities as far as detailed implementations are concerned, such as different definition and scaling of variables, different implicit and explicit updatings, different difference operators, e.g. higher order, parameter optimized, compact etc. [19]-[22]. The updating equations presented above still leave much room and generality for exploring into these possibilities.

B. LOD2

For subsequent discussions and comparisons, we also describe the algorithm implementation of fundamental LOD2 scheme (21) in more detail and yet general. Adopting the same notations of (33)-(36), equation (21a) may be split into implicit updating

$$\frac{1}{2}e_x^{n+\frac{3}{4}} - \frac{bd}{2}\partial_y^2 e_x^{n+\frac{3}{4}} = E_x^{n+\frac{1}{4}} + b\partial_y H_z^{n+\frac{1}{4}} \tag{45a}$$

$$\frac{1}{2}e_y^{n+\frac{3}{4}} - \frac{bd}{2}\partial_z^2 e_y^{n+\frac{3}{4}} = E_y^{n+\frac{1}{4}} + b\partial_z H_x^{n+\frac{1}{4}} \tag{45b}$$

$$\frac{1}{2}e_z^{n+\frac{3}{4}} - \frac{bd}{2}\partial_x^2 e_z^{n+\frac{3}{4}} = E_z^{n+\frac{1}{4}} + b\partial_x H_y^{n+\frac{1}{4}} \tag{45c}$$

and explicit updating

$$h_x^{n+\frac{3}{4}} = 2H_x^{n+\frac{1}{4}} + d\partial_z e_y^{n+\frac{3}{4}} \tag{46a}$$

$$h_y^{n+\frac{3}{4}} = 2H_y^{n+\frac{1}{4}} + d\partial_x e_z^{n+\frac{3}{4}} \tag{46b}$$

$$h_z^{n+\frac{3}{4}} = 2H_z^{n+\frac{1}{4}} + d\partial_y e_x^{n+\frac{3}{4}}, \tag{46c}$$

while (21b) is directly

$$E_\xi^{n+\frac{3}{4}} = e_\xi^{n+\frac{3}{4}} - E_\xi^{n+\frac{1}{4}}, \quad \xi = x, y, z \tag{47a}$$

$$H_\xi^{n+\frac{3}{4}} = h_\xi^{n+\frac{3}{4}} - H_\xi^{n+\frac{1}{4}}, \quad \xi = x, y, z. \tag{47b}$$

Likewise, equation (21c) consists of implicit updating

$$\frac{1}{2}e_x^{n+1\frac{1}{4}} - \frac{bd}{2}\partial_z^2 e_x^{n+1\frac{1}{4}} = E_x^{n+\frac{3}{4}} - b\partial_z H_y^{n+\frac{3}{4}} \tag{48a}$$

$$\frac{1}{2}e_y^{n+1\frac{1}{4}} - \frac{bd}{2}\partial_x^2 e_y^{n+1\frac{1}{4}} = E_y^{n+\frac{3}{4}} - b\partial_x H_z^{n+\frac{3}{4}} \tag{48b}$$

$$\frac{1}{2}e_z^{n+1\frac{1}{4}} - \frac{bd}{2}\partial_y^2 e_z^{n+1\frac{1}{4}} = E_z^{n+\frac{3}{4}} - b\partial_y H_x^{n+\frac{3}{4}} \tag{48c}$$

and explicit updating

$$h_x^{n+1\frac{1}{4}} = 2H_x^{n+\frac{3}{4}} - d\partial_y e_z^{n+1\frac{1}{4}} \tag{49a}$$

$$h_y^{n+1\frac{1}{4}} = 2H_y^{n+\frac{3}{4}} - d\partial_z e_x^{n+1\frac{1}{4}} \tag{49b}$$

$$h_z^{n+1\frac{1}{4}} = 2H_z^{n+\frac{3}{4}} - d\partial_x e_y^{n+1\frac{1}{4}}, \tag{49c}$$

whereas (21d) is simply

$$E_\xi^{n+1\frac{1}{4}} = e_\xi^{n+1\frac{1}{4}} - E_\xi^{n+\frac{3}{4}}, \quad \xi = x, y, z \tag{50a}$$

$$H_\xi^{n+1\frac{1}{4}} = h_\xi^{n+1\frac{1}{4}} - H_\xi^{n+\frac{3}{4}}, \quad \xi = x, y, z. \tag{50b}$$

As before, there are many choices for the difference operators in equations (45)-(50). For being more specific in the next section, we select the second-order central-differencing operators on Yee cells. To minimize the number of for-loops in the previous algorithm, some of their updating equations may be incorporated in the same loops. Moreover, when the auxiliary magnetic variables h_ξ are not to be output, the explicit updating equations (46) and (47b) can be combined to read

$$H_x^{n+\frac{3}{4}} = H_x^{n+\frac{1}{4}} + d\partial_z e_y^{n+\frac{3}{4}} \tag{51a}$$

$$H_y^{n+\frac{3}{4}} = H_y^{n+\frac{1}{4}} + d\partial_x e_z^{n+\frac{3}{4}} \tag{51b}$$

$$H_z^{n+\frac{3}{4}} = H_z^{n+\frac{1}{4}} + d\partial_y e_x^{n+\frac{3}{4}}, \tag{51c}$$

while (49) and (50b) become

$$H_x^{n+1\frac{1}{4}} = H_x^{n+\frac{3}{4}} - d\partial_y e_z^{n+1\frac{1}{4}} \tag{52a}$$

$$H_y^{n+1\frac{1}{4}} = H_y^{n+\frac{3}{4}} - d\partial_z e_x^{n+1\frac{1}{4}} \tag{52b}$$

$$H_z^{n+1\frac{1}{4}} = H_z^{n+\frac{3}{4}} - d\partial_x e_y^{n+1\frac{1}{4}}. \tag{52c}$$

Meanwhile, to minimize the memory storage requirement, some of the variables may occupy the same memory spaces. For instance, one can choose to reuse the spaces of those variables grouped within curly braces as ($\xi = x, y, z$):

$$\{E_\xi^{n+\frac{1}{4}}, e_\xi^{n+1\frac{1}{4}}, E_\xi^{n+1\frac{1}{4}}\}; \quad \{e_\xi^{n+\frac{3}{4}}, E_\xi^{n+\frac{3}{4}}\};$$
$$\{H_\xi^{n+\frac{1}{4}}, h_\xi^{n+1\frac{1}{4}}, H_\xi^{n+1\frac{1}{4}}\}; \quad \{h_\xi^{n+\frac{3}{4}}, H_\xi^{n+\frac{3}{4}}\}.$$

In a computer program, these variables could just share the same names to be assigned with new values successively. Note that in conjunction with (51)-(52), one may absorb h_ξ altogether and reuse the H_ξ spaces.

Following the similar arguments as above, the detailed algorithms for the fundamental LOD1/SS1 and SS2 schemes may be written and coded with reference to (15) and (17) accordingly.

IV. Discussions and Comparisons

Having systematically addressed various implicit schemes and their new algorithm implementations, we are now ready to make a comparative study. Of particular interest are the computation efficiency gains as compared to the original schemes, or more specifically over the prevailing ADI-FDTD implementation [3], [4]. Furthermore, the fundamental nature of new algorithm implementations, as has been pointed out in Section II-E, will become evident once again through the comparisons and discussions.

In what follows, we acquire the floating-point operations (flops) count for the main iterations of ADI, LOD1/SS1, SS2 and LOD2 schemes. Based on the right-hand sides (RHS) of their respective updating equations using second-order central-differencing operators, cf. Sections II and III, the number of multiplications/divisions (M/D) and additions/subtractions (A/S) required for one complete time step are determined. Each complete step comprises two/three procedures whose implicit and explicit updating codes are assumed to have been arranged in order to achieve minimum number of for-loops.

TABLE I
COMPARISONS OF UNCONDITIONALLY STABLE FDTD METHODS WITH SECOND-ORDER SPATIAL CENTRAL DIFFERENCE

Scheme		ADI		LOD1/SS1		SS2		LOD2	
Algorithm		Original	New	Original	New	Original	New	Original	New
Equations	Main Iterations	(1)	(8)	(11)	(15)	(16)	(17)	(18)	(21)
	Input	–	(7), (10)	–	–	–	–	(19)	(22)
	Output	–	(9)	–	–	–	–	(20)	(23)
Memory	Field Array 1	$\mathbf{u}^{n+1}$ $\mathbf{u}^{n}$	$\tilde{\mathbf{u}}^{n+1}$ $\mathbf{v}^{n}$ $\tilde{\mathbf{u}}^{n}$	$\mathbf{u}^{n+1}$ $\mathbf{u}^{n}$	$\mathbf{u}^{n+1}$ $\mathbf{v}^{n+1}$ $\mathbf{u}^{n}$	$\mathbf{u}^{n+1}$ $\mathbf{u}^{n+\frac{1}{4}}$	$\mathbf{u}^{n+1}$ $\mathbf{v}^{n+1}$ $\mathbf{u}^{n+\frac{1}{4}}$ $\mathbf{v}^{n+\frac{1}{4}}$	$\mathbf{u}^{n+1\frac{1}{4}}$ $\mathbf{u}^{n+\frac{1}{4}}$	$\mathbf{u}^{n+1\frac{1}{4}}$ $\mathbf{v}^{n+1\frac{1}{4}}$ $\mathbf{u}^{n+\frac{1}{4}}$
	Field Array 2	$\mathbf{u}^{n+\frac{1}{2}}$	$\mathbf{v}^{n+\frac{1}{2}}$ $\tilde{\mathbf{u}}^{n+\frac{1}{2}}$ $\mathbf{v}^{n-\frac{1}{2}}$	$\mathbf{u}^{n+\frac{1}{2}}$	$\mathbf{u}^{n+\frac{1}{2}}$ $\mathbf{v}^{n+\frac{1}{2}}$	$\mathbf{u}^{n+\frac{3}{4}}$ $\mathbf{u}^{n}$	$\mathbf{u}^{n+\frac{3}{4}}$ $\mathbf{v}^{n+\frac{3}{4}}$ $\mathbf{u}^{n}$	$\mathbf{u}^{n+\frac{3}{4}}$	$\mathbf{u}^{n+\frac{3}{4}}$ $\mathbf{v}^{n+\frac{3}{4}}$
Implicit	M/D	18	6	18	6	27	9	18	6
	A/S	48	18	24	18	36	27	24	18
Explicit	M/D	12	6	6	6	9	9	6	6
	A/S	24	12	24	12	36	18	24	12
Total	M/D	30	12	24	12	36	18	24	12
	A/S	72	30	48	30	72	45	48	30
	M/D+A/S	102	42	72	42	108	63	72	42
Efficiency gain	RHS	1	2.43	1.42	2.43	0.94	1.62	1.42	2.43
	Overall	1	1.83	1.29	1.83	0.86	1.22	1.29	1.83
For-Loops		12		12		18		12	
Temporal Accuracy		Second-Order		First-Order		Second-Order		Second-Order	

It is also assumed that all multiplicative factors have been precomputed and stored, while the number of electric and magnetic field components in all directions have been taken to be the same. Our results are summarized in Table I, which lists the flops count for both original and new algorithm implementations of each scheme. For convenience, the pertaining equations involved in the main iterations are also indicated, along with those that may be needed for input initialization and output processing. When there is no input or output equation shown, it means that no special input or output treatment is required for the particular implementation. For the main iterations variables, their memory storage requirements are also specified in Table I. By properly reusing the spaces of those variables grouped within the same table entity, only two field arrays need to be allocated for all. Note that judicious reuse of memory space is important so that the iterations do not often invoke virtual memory (e.g. via hard disk). Otherwise, they may lead to slower computations than those iterations with more flops but do not need virtual memory.

From Table I, it is clear that the total flops count (M/D+A/S) for all implicit schemes has been reduced considerably using the new implementations as compared to the original counterparts. Most notably, the conventional ADI-FDTD method based on (1) takes 102 flops [3], [4], whereas the new one based on (8) merely takes 42 flops, cf. Section III-A (assuming no need for frequent output of magnetic field data). Notice that the LOD1/SS1 and LOD2 schemes, which take 72 flops in their original implementations [13], also take the same 42 flops using the new algorithms, cf. Section III-B. Moreover, the number and type of flops for respective implicit and explicit updatings are seen to be identical for all these new implicit schemes. This is simply a manifestation of their aforementioned fundamental updating structures, which can be found to take 21 flops per updating procedure. Applying such argument for the SS2 scheme with three updating procedures, it is easy to deduce that its new algorithm will take 63 flops, in contrast to the 108 flops required when using its original implementation.

Based on the flops count reduction for the right-hand sides of updating equations, Table I has listed the efficiency gains over the conventional ADI-FDTD method. The new fundamental ADI, LOD1/SS1 and LOD2 schemes all feature the same efficiency gain of 2.43 with their same reduced total flops count. Since there is also the cost for solving tridiagonal systems, we estimate about $5N$ flops for a system of order N using precomputed bidiagonally factorized elements. Taking this cost into account, all these new implementations still achieve an overall efficiency gain of 1.83 in flops count reduction over the conventional ADI-FDTD. It is interesting to find that even for the SS2 scheme with three updating procedures, the overall efficiency may still be higher (gain $\sim$ 1.22) by using the new algorithm implementation. Note that although we have carefully taken into account various flops count, there still exist other factors that may affect the actual computation efficiency. These factors are often dependent on the particular computer platform, hardware configuration, operating system, compiler software, program code arrangement, etc. [23].

Besides the flops count, the for-loops count is also given in Table I. Each for-loop is to perform the whole sweep of i, j, k indices along x, y, z directions for one field component. For both original and new implementations of ADI, LOD1/SS1 and LOD2 schemes, there are 12 for-loops for all updating equations in both procedures. For the SS2 scheme, there are 18 for-loops altogether corresponding to its three updating procedures. Meanwhile, the order of temporal accuracy is also summarized for various implicit schemes in Table I. As mentioned in Section II-E, even though the LOD1/SS1 scheme is only first-order accurate in time, it may be converted to other schemes of second-order accuracy by exploiting their similar fundamental updating structures.

Although not included in Table I, many other implicit schemes discussed in Section II-E can be found to take the same reduced flops count (42) and the same for-loops count (12) when using the new implementations with (33)-(36). This is anticipated because they all simply converge to the same fundamental schemes with matrix-operator-free right-hand sides just like those addressed in the table, even though their original splitting formulae are distinctive, cf. (25)-(30). Here again the significance of these fundamental schemes accrues when one is equipped with a general methodical approach for improving the computation efficiency of their original counterparts. Moreover, their updating structures that are of fundamental nature make them well-suited to serve as the basis and benchmark for construction and development of future implicit schemes, e.g. with new $\mathbf{A}$ and $\mathbf{B}$ or different time-stepping procedures, etc.

V. Conclusion

This paper has presented the generalized formulations of fundamental schemes for efficient unconditionally stable implicit FDTD methods. The formulations have been presented in terms of generalized matrix operator equations pertaining to classical splitting formulae of ADI, LOD and split-step schemes. To provide further insights into the implications and significance of fundamental schemes, the analyses have also been extended to many other schemes with distinctive splitting formulae. It has been noted that all the fundamental schemes feature similar fundamental updating structures that are in simplest forms with most efficient right-hand sides. Detailed algorithms have been described for new efficient implementations of the unconditionally stable implicit FDTD methods based on the fundamental schemes. A comparative study of various implicit schemes in their original and new implementations has been carried out, which includes comparisons of their computation costs and efficiency gains.

The fundamental schemes presented in this paper will be of much usefulness and significance not only in electromagnetics, but also in many other areas that may be adopting various classical implicit schemes. With their simplest forms featuring the most efficient right-hand sides, these fundamental schemes will lead to coding simplification and efficiency improvement in algorithm implementations. Furthermore, their fundamental updating structures invite further investigations into their properties and subsequent extensions for their applications. They will also serve aptly as the basis and benchmark for construction and development of future implicit schemes.

References

[1] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method* (Boston, M. A.: Artech House, 2005).

[2] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, no. 3, pp. 302-307, May 1966.

[3] F. Zheng, Z. Chen and J. Zhang, "Toward the Development of a Three-Dimensional Unconditionally Stable Finite-Difference Time-Domain Method," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1550-1558, Sep. 2000.

[4] T. Namiki, "3-D ADI-FDTD Method – Unconditionally Stable Time-Domain Algorithm for Solving Full Vector Maxwell's Equations," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 10, pp. 1743-1748, Oct. 2000.

[5] D. Poljak, *Time Domain Techniques in Computational Electromagnetics* (Boston: WIT Press, 2004).

[6] N. V. Kantartzis and T. D. Tsiboukis., *Higher-Order FDTD Schemes for Waveguide and Antenna Structures* (San Rafael: Morgan and Claypool, 2006).

[7] D. Peaceman and H. Rachford Jr., "The numerical solution of parabolic and elliptic differential equations," *J. Soc. Ind. Appl. Math.*, vol. 3, no. 1, pp. 28-41, 1955.

[8] Ye. G. D'Yakonov, "Some difference schemes for solving boundary problems," *U.S.S.R. Comput. Math. and Math. Phys.*, vol. 3, pp. 55-77, 1963.

[9] G. Strang, "On construction and comparison of difference schemes," *SIAM J. Numer. Anal.*, vol. 5, pp. 506-516, 1968.

[10] A. R. Mitchell and D. F. Griffiths, *The Finite-Difference Method in Partial Differential Equations* (New York: Wiley, 1980).

[11] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods* (New York: Springer-Verlag, 1998).

[12] J. Shibayama, M. Muraki, J. Yamauchi and H. Nakano, "Efficient implicit FDTD algorithm based on locally one-dimensional scheme," *Electron. Lett.*, vol. 41, no. 19, pp. 1046-1047, Sep. 2005.

[13] E. L. Tan, "Unconditionally Stable LOD-FDTD Method for 3-D Maxwell's Equations," *IEEE Microw. Wireless Comp. Lett.*, vol. 17, no. 2, pp. 85-87, Feb. 2007.

[14] J. Lee and B. Fornberg, "A split step approach for the 3-D Maxwell's equations", *J. Comput. Appl. Math.*, vol. 158, pp. 485-505, 2003.

[15] W. Fu and E. L. Tan, "Development of split-step FDTD method with higher order spatial accuracy," *Electron. Lett.*, vol. 40, no. 20, pp. 1252-1254, Sep. 2004.

[16] G. Sun and C. W. Trueman, "Efficient Implementations of the Crank-Nicolson Scheme for the Finite-Difference Time-Domain Method," *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 5, pp. 2275-2284, May 2006.

[17] E. L. Tan, "Efficient Algorithm for the Unconditionally Stable 3-D ADI-FDTD Method," *IEEE Microw. Wireless Comp. Lett.*, vol. 17, no. 1, pp. 7-9, Jan. 2007.

[18] A. R. Gourlay and A. R. Mitchell, "The equivalence of certain alternating direction and locally one-dimensional difference methods,"" *SIAM J. Numer. Anal.*, vol. 6, pp. 37-46, 1969.

[19] W. Fu and E. L. Tan, "Stability and Dispersion Analysis for Higher Order 3-D ADI-FDTD Method," *IEEE Trans. Antennas Propagat.*, vol. 53, no. 11, pp. 3691-3696, Nov. 2005.

[20] W. Fu and E. L. Tan, "A Parameter Optimized ADI-FDTD Method Based on the (2,4) Stencil," *IEEE Trans. Antennas Propagat.*, vol. 54, no. 6, pp. 1836-1842, Jun. 2006.

[21] W. Fu and E. L. Tan, "A compact higher-order ADI-FDTD method," *Microwave Opt. Technol. Lett.*, vol. 44, no. 3, pp. 273-275, Feb. 2005.

[22] M. Wang, Z. Wang and J. Chen, "A parameter optimized ADI-FDTD method," *IEEE Antennas. Wireless Propagat. Lett.*, vol. 2, pp. 118-121, 2003.

[23] G. Sun and C. W. Trueman, "Suppression of Numerical Anisotropy and Dispersion With Optimized Finite-Difference Time-Domain Methods," *IEEE Trans. Antennas Propagat.*, vol. 53, no. 12, pp. 4121-4128, Dec. 2005.