



HAL
open science

A new family of exponential-based high order DGTD methods for modelling 3D transient multiscale electromagnetic problems

Hao Wang, Li Xu, Bin Li, Stéphane Descombes, Stéphane Lantéri

► **To cite this version:**

Hao Wang, Li Xu, Bin Li, Stéphane Descombes, Stéphane Lantéri. A new family of exponential-based high order DGTD methods for modelling 3D transient multiscale electromagnetic problems. 2017. hal-01523749

HAL Id: hal-01523749

<https://hal.science/hal-01523749>

Preprint submitted on 16 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A new family of exponential-based high order DGTD methods for modelling 3D transient multiscale electromagnetic problems

Hao Wang, Li Xu, *Member, IEEE*, Bin Li, *Member, IEEE*, Stéphane Descombes, and Stéphane Lantéri

Abstract—The accurate and efficient simulation of 3D transient multiscale electromagnetic problems is extremely challenging for conventional numerical methods. Assuming a splitting of the underlying tetrahedral mesh in coarse and fine parts and using the Lawson procedure, we derive a family of exponential-based time integration methods for the time-domain Maxwell’s equations discretized by a high order discontinuous Galerkin (DG) scheme formulated on locally refined unstructured meshes. These methods remove the stiffness on the time explicit integration of the semi-discrete operator associated to the fine part of the mesh, and allow for the use of high order time explicit scheme for the coarse part operator. They combine excellent stability properties with the ability to obtain very accurate solutions even for very large time step sizes. Here, the explicit time integration of the Lawson-transformed semi-discrete system relies on a Low-Storage Runge-Kutta (LSRK) scheme, leading to a combined Lawson-LSRK scheme. In addition, efficient techniques are also presented to further improve the efficiency of this exponential-based time integration. For the efficient calculation of matrix exponential, we employ the Krylov subspace method. Numerical experiments are presented to assess the stability, verify the accuracy and numerical convergence of the Lawson-LSRK scheme. They also demonstrate that the DGTD methods based on the proposed time integration scheme can be much faster than those based on classical fully explicit time stepping schemes, with the same accuracy and moderate memory usage increase on locally refined unstructured meshes, and are thus very promising for modelling three-dimensional multiscale electromagnetic problems.

Index Terms—Multiscale problems, time-domain Maxwell’s equations, Discontinuous Galerkin Time-Domain method, locally refined unstructured meshes, Lawson procedure, exponential time integration, Krylov subspace method.

I. INTRODUCTION

REALISTIC electromagnetic wave propagation problems are often multiscale. They represent a major research emphasis for computational electromagnetics [1]-[2]-[3]. Consider the problem of the scattering of plane wave by an aircraft as an example: the aircraft frame is very large, which is tens of meters in the long and wing spans, and comprises various small geometric structures such as stabilizers, trailing edges of wings as well as antennas. The numerical simulation of

such multiscale problems is extremely challenging for conventional numerical methods. The Finite Difference Time-Domain (FDTD) methods [4]-[5] are very often used in this context because of their simplicity and computational efficiency, but they are based on an orthogonal Cartesian grid. Therefore, a large number of grid points is required to capture the geometric details, and leads to an amount of unnecessary unknowns in the electrically coarse region. Moreover, for complex practical problems with oblique or curved boundaries or interfaces, FDTD methods suffer from the inaccurate representation of the solution (so called, staircase approximation). Even if the subgridding technique [6] can alleviate this drawback, the simplicity of the standard FDTD method is lost. Another widely used kinds of methods, the Finite Element (FE) methods [7]-[8], can handle unstructured grids, complex geometries, and heterogeneous media, regardless of the order of approximation. However, FE methods require solving a linear system of equations. For multiscale problems, the system matrix is usually very large, and the solution of this system can be very expensive, especially in a time-domain setting where this has to be performed at each time step. Discontinuous Galerkin Time-Domain (DGTD) methods [9]-[10]-[11] are now popular for the solution of electromagnetic problems. DGTD methods accommodate elements of various types and shapes, irregular non-matching grids, and even locally varying polynomial order, and hence offer great flexibility for modeling complex problems. Moreover, DGTD methods lead to a block-diagonal mass matrix, thus when a spatial DG discretization is combined with explicit time integration, the resulting time marching scheme will be truly explicit and inherently parallel. Additionally, DGTD methods allow domain decomposition, either element-wise or not, they can easily handle the problems too large to be solved for FETD methods by splitting them into a number of smaller problems. In this work, we consider such a DGTD method as a starting point.

For transient multiscale electromagnetic problems with complex geometries or heterogeneous media, adaptive mesh refinement is an attractive technique for the efficient numerical solution of time-domain Maxwell’s equations. Local mesh refinement, however, also imposes a severe stability constraint on explicit time integration since the maximal time step size is determined by the smallest elements in the mesh. A first natural way to limit the impact of this problem is to use explicit local time stepping (LTS) approaches like that proposed in [12]-[13]-[14]-[15]. Explicit LTS methods adopt smaller time step sizes inside the refined part of mesh, while remaining fully

This work was supported by the National Natural Science Foundation of China under Grant 61301054.

H. Wang, L. Xu and B. Li are with the National Key Laboratory of Science and Technology on Vacuum Electronics, School of Physical Electronics, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054 China e-mail: (haowang_uestc@outlook.com; lixu@uestc.edu.cn; libin@uestc.edu.cn).

S. Descombes and S. Lantéri are with Université Côte d’Azur, Inria, CNRS, LJAD, Sophia Antipolis, 06902 France e-mail: (stephane.descombes@inria.fr; stephane.lanteri@inria.fr).

explicitness in the entire computational domain. Additionally, these methods can be used recursively and easily parallelized. On the other hand, since they are conditionally stable, small time step sizes are still necessary for the refined part of the mesh. A second way is to adopt unconditional implicit time integration [16]-[17]. However, this approach is very expensive and even infeasible, especially for 3D problems, since a large global matrix system needs to be solved at each time step. To overcome this expensive resource consumption, various implicit-explicit (IMEX) schemes (or so-called locally implicit schemes) have been proposed by Piperno [12] or Descombes, Lanteri and Moya [18] (based on the work of Verwer [19]), where a time implicit scheme is used only for the refined part of the mesh, and a time explicit scheme is used for the remaining part. Note that highly disparate mesh element sizes in the refined part of the mesh can lead to ill-conditioned problems since time implicit scheme is used there. Additionally, Hochbruck and Sturm [20] have provided a rigorous analysis for the second order IMEX scheme proposed by Verwer [19] based on a variational formulation and energy techniques. Using a variational formulation of the evolution equation, the analysis shows that it is necessary to implicitly treat not only the fine elements but also their direct neighbors to obtain a scheme with a CFL condition independent of the fine part. IMEX schemes combine enhanced stability properties with ability to produce a very accurate solution even for relatively large time step sizes but are limited so far to second order accuracy in time. With the increasing accuracy requirement of modelling multiscale electromagnetic problems, which is possibly beyond the capabilities of those methods, drives the quest for more accurate methods.

Recently, Demirel *et al.* [21] have proposed efficient high order multiple time-stepping (MTS) methods for ordinary differential equations with stiff terms (either Algebraic stiffness or Grid-induced stiffness, which mandate a small time step). Like an IMEX method, this approach allows to employ different time stepping strategies for the inner (stiff) and the outer (non-stiff) integration. The derived predictor-corrector MTS (PCMTS) scheme for the outer integration allows significantly larger time step sizes when compared to previously known explicit MTS schemes and yields significant performance enhancement. For the inner integration, both the implicit and the explicit schemes can be adopted. Note that the time step sizes in the inner integration still have to be small enough if an explicit time stepping scheme is used. Based on the exponential time integration which starts from the so-called variation-of-constants formula, Botchev [22] has proposed exponential Krylov subspace time integration methods for time-domain Maxwells equations discretized by FDTD scheme. This method combines excellent (unconditional) stability properties and enables the ability to produce a very accurate solution even for relatively large time step sizes. In addition, the adopted efficient techniques such as the Krylov shift-and-invert method and residual-based stopping criteria further improved its efficiency. However, actions of matrix functions of a large sparse global matrix on a vector are required every time step if the source term is non-zero or non-constant. Moreover, the first order exponentially fitted Euler scheme [23] and the second

order exponential Krylov (EK2) scheme [17] are adopted and thus the obtained accuracy is limited to second order. Based on the Adams-Bashforth multi-step schemes [24], Grote and Mitkova [25] derived the Adams-Bashforth based LTS scheme of arbitrarily high order of accuracy, while remaining fully explicitness, for damped wave equations. This method splits the unknown vector into two parts associated with the locally refined region or not, and treats them differently during the time iteration. In this way, arbitrarily small time step sizes are allowed where small elements in the spatial mesh are located. In addition, numerical experiments are presented to demonstrate the stability properties and the usefulness of this method in 1D and 2D.

The Lawson method [26] has been successfully applied for solving non-linear problems [27]-[28], by splitting the problems into linear part and non-linear part and solving the linear part with high accuracy. Through a transformation, this method removes the explicit dependence in the differential equation on the operator of linear part. This has inspired us to develop a new family of exponential-based high order DGTD methods to so solve multiscale electromagnetic wave problems accurately and efficiently. In section II, we present the initial boundary value problem considered in this work. We then introduce the DG discretization in space in section III. In section IV, we propose a time integration strategy that combines excellent stability properties with efficient and accurate time explicit schemes, to overcome the severe stability restrictions caused by the local mesh refinement of multiscale electromagnetic problems. We first formulate the semi-discrete DG-based equations in the form of a global system of ordinary differential equations. Assuming a splitting of the underlying tetrahedral mesh in coarse and fine parts and using the Lawson procedure, we then derive a family of exponential-based time integration methods that remove the stiffness on the time explicit integration of the semi-discrete operator associated to the fine part of the mesh, and allow for the use of high order time explicit scheme for the coarse part operator. The developed exponential time integration can be time advancing by a variety of explicit time stepping schemes; we adopt here a Low-Storage Runge-Kutta (LSRK) scheme [29]. Thus the so-called combined Lawson-LSRK time integration is constructed. In addition, efficient techniques are also presented to further improve the efficiency of this exponential-based time integration in Section V, such as the transformation of the combined Lawson-LSRK scheme and the model reduction for exponential time integration which significantly reduce the dimension of the matrix required to calculate exponential. Finally, to investigate the stability, accuracy, convergence order, and demonstrate the computational performance of the proposed exponential-based time integration strategy, we present in Section VI several numerical experiments for 3D transient multiscale electromagnetic problems .

II. INITIAL AND BOUNDARY VALUE PROBLEM

In this paper, apart from some exceptions that will be made clear in the text, we adopt the following notations: v denotes a scalar quantity, \mathcal{V} , \mathbf{v} or \mathbf{V} a vector, and \mathbb{A} a matrix. We

consider the system of 3D time-domain Maxwell's equations on a bounded polyhedral domain $\Omega \subset \mathbb{R}^3$

$$\begin{cases} \varepsilon \partial_t \mathcal{E} - \mathbf{curl} \mathcal{H} = -\mathcal{J}, & \text{in } \Omega \times [0, T], \\ \mu \partial_t \mathcal{H} + \mathbf{curl} \mathcal{E} = 0, & \text{in } \Omega \times [0, T], \end{cases} \quad (1)$$

where the symbol ∂_t denotes a time derivative, \mathcal{J} the current density, T a final time, $\mathcal{E}(\mathbf{x}, t)$ and $\mathcal{H}(\mathbf{x}, t)$ are the electric and magnetic fields. The dielectric permittivity ε and the magnetic permeability μ are varying in space, time-invariant and both positive functions. The boundary of Ω is defined as $\partial\Omega = \Gamma_m \cup \Gamma_a$ with $\Gamma_m \cap \Gamma_a = \emptyset$. The boundary conditions are chosen as

$$\begin{cases} \mathbf{n} \times \mathcal{E} = 0, & \text{on } \Gamma_m \times [0, T], \\ \mathbf{n} \times \mathcal{E} + \sqrt{\frac{\mu}{\varepsilon}} \mathbf{n} \times (\mathbf{n} \times \mathcal{H}) = \mathbf{g}^{\text{inc}}, & \text{on } \Gamma_a \times [0, T], \end{cases} \quad (2)$$

where \mathbf{n} denotes the unit outward normal to $\partial\Omega$ and

$$\mathbf{g}^{\text{inc}} = \mathbf{n} \times \mathcal{E}^{\text{inc}} + \sqrt{\frac{\mu}{\varepsilon}} \mathbf{n} \times (\mathbf{n} \times \mathcal{H}^{\text{inc}}). \quad (3)$$

Here $(\mathcal{E}^{\text{inc}}, \mathcal{H}^{\text{inc}})$ denotes a given incident field. The first boundary condition is often referred as a metallic boundary condition and is applied on a perfectly conducting surface. The second relation is an absorbing boundary condition (ABC) and takes here the form of the first-order Silver-Müller condition. It is applied on a surface corresponding to an artificial truncation of a theoretically unbounded propagation domain. Finally, the system is supplemented with initial conditions: $\mathcal{E}_0(\mathbf{x}) = \mathcal{E}(\mathbf{x}, 0)$ and $\mathcal{H}_0(\mathbf{x}) = \mathcal{H}(\mathbf{x}, 0)$.

III. THE DG DISCRETIZATION IN SPACE

We consider a partition \mathcal{T}_h of $\Omega \subset \mathbb{R}^3$ into a set of tetrahedra. Each non-empty intersection of two elements K^+ and K^- is called an interface. We denote by \mathcal{F}_h^I the union of all interior interfaces of \mathcal{T}_h , by \mathcal{F}_h^B the union of all boundary interfaces of \mathcal{T}_h , and $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^B$. Note that $\partial\mathcal{T}_h$ represents all the interfaces ∂K for all $K \in \mathcal{T}_h$. As a result, an interior interface shared by two elements appears twice in $\partial\mathcal{T}_h$, unlike in \mathcal{F}_h where this interface is evaluated once. For an interface $F \in \mathcal{F}_h^I$, $F = K^+ \cap K^-$, let \mathbf{v}^\pm be the traces of \mathbf{v} on F from the interior of K^\pm . We define the tangential trace and projection operators $\gamma(v) = \mathbf{n} \times v|_{\partial K}$ and $\pi(v) = \mathbf{n} \times (v \times \mathbf{n})|_{\partial K}$, where \mathbf{n} defines the outward normal vector to the face ∂K . On this interior face, we further define the following mean values $\{\cdot\}$ and jumps $[\![\cdot]\!]$ as

$$\begin{cases} \{\mathbf{v}\}_F = \frac{1}{2}[\pi(\mathbf{v}^+) + \pi(\mathbf{v}^-)], \\ [\![\mathbf{v}]\!]_F = \gamma(\mathbf{v}^+) + \gamma(\mathbf{v}^-). \end{cases}$$

For the boundary faces, these expressions are modified as

$$\begin{cases} \{\mathbf{v}\}_F = \pi(\mathbf{v}^+), \\ [\![\mathbf{v}]\!]_F = \gamma(\mathbf{v}^+), \end{cases}$$

since we assume \mathbf{v} is single-valued on the boundaries. In the following, we introduce the discontinuous finite element spaces and some basic operations on these spaces for later use. Let $\mathbb{P}_{p_K}(K)$ denotes the space of polynomial functions of

degree at most p_K on the element $K \in \mathcal{T}_h$. The discontinuous finite element space is introduced as

$$\mathbf{V}_h = \left\{ \mathbf{v} \in [L^2(\Omega)]^3 \mid \mathbf{v}_K \in [\mathbb{P}_{p_K}(K)]^3, \forall K \in \mathcal{T}_h \right\}, \quad (4)$$

where $L^2(\Omega)$ is the space of square integrable functions on the domain Ω . The functions in \mathbf{V}_h are continuous inside each element and discontinuous across the interfaces between elements. By following the interior penalty approach described in [11], the DGTD formulation using a central flux can be derived as follows: find $(\mathcal{H}, \mathcal{E}) \in \mathbf{V}_h \times \mathbf{V}_h$, so that for $\forall \Phi \in \mathbf{V}_h$

$$\begin{aligned} & \int_{\Omega} (\mathbf{curl} \mathcal{E} + \mu \partial_t \mathcal{H}) \cdot \Phi d\Omega - \\ & \int_{\Omega} (\mathbf{curl} \mathcal{H} - \mu \partial_t \mathcal{E}) \cdot \Phi d\Omega + \\ & \int_{\mathcal{F}_h} \{\Phi\}_F^T [\mathcal{H}]_F ds - \int_{\mathcal{F}_h} \{\Phi\}_F^T [\mathcal{E}]_F ds = 0. \end{aligned} \quad (5)$$

This is a conservative formulation, but with a suboptimal $O(h^p)$ rate of convergence as derived in [9]. Considering the Silver-Müller boundary condition (the second equation in (2)) on boundary Γ_a . For each face on this boundary, the trace of a fictitious neighboring element is required for the computation of the numerical flux. We treat the Silver-Müller boundary conditions in a weak sense by defining the following appropriate values of the electric and magnetic fields in the fictitious element, such that for $\forall a_{ik} \in \Gamma_a$

$$\begin{cases} \mathcal{E}_k = -\sqrt{\frac{\mu_i}{\varepsilon_i}} \mathbf{n} \times \mathcal{H}_i + \mathcal{E}_i^{\text{inc}} + \sqrt{\frac{\mu_i}{\varepsilon_i}} \mathbf{n} \times \mathcal{H}_i^{\text{inc}} \\ \mathcal{H}_k = \sqrt{\frac{\varepsilon_i}{\mu_i}} \mathbf{n} \times \mathcal{E}_i + \mathcal{H}_i^{\text{inc}} - \sqrt{\frac{\varepsilon_i}{\mu_i}} \mathbf{n} \times \mathcal{E}_i^{\text{inc}}. \end{cases} \quad (6)$$

Inside each finite element K_i , the local electric and magnetic fields $(\mathbf{E}_h|_{K_i}, \mathbf{H}_h|_{K_i}) = (\mathbf{E}_i, \mathbf{H}_i)$ are expanded as combinations of linearly independent vector basis functions $\Phi_{il} \in \mathbf{V}_h$ presented in [7]-[30]-[31] as follows

$$\begin{cases} \mathbf{E}_i(\mathbf{x}, t) = \mathbf{E}_h(\mathbf{x}, t)|_{\tau_i} = \sum_l^{d_i} E_{il}(t) \Phi_{il}(\mathbf{x}), \\ \mathbf{H}_i(\mathbf{x}, t) = \mathbf{H}_h(\mathbf{x}, t)|_{\tau_i} = \sum_l^{d_i} H_{il}(t) \Phi_{il}(\mathbf{x}) \end{cases} \quad (7)$$

where d_i denotes the local number of degrees of freedom associated to the basis function degree p_i in K_i , \mathbf{x} denotes the position vector, and E_{il}, H_{il} reflect the coefficient of the vector basis Φ_{il} . After imposing the Silver-Müller boundary condition on boundary Γ_a and considering the volume source term \mathcal{J} , we separate the basis function Φ in (5). Then the following local semi-discrete systems for element K_i in matrix

form can be obtained

$$\left\{ \begin{array}{l} \mathbb{M}_i^\varepsilon \partial_t \mathbf{E}_i = \mathbb{K}_i \mathbf{H}_i - \sum_{k \in V_i^i} \mathbb{S}_{ik} \mathbf{H}_i - \sum_{k \in V_i^i} \mathbb{S}_{ik}^+ \mathbf{H}_k \\ \quad - \sum_{k \in V_i^a} \mathbb{S}_{ik} \mathbf{H}_i + \sum_{k \in V_i^a} \mathbb{S}_{ik}^E \mathbf{E}_i + f_i^{inc,E} \\ \quad - j_i^E, \\ \mathbb{M}_i^\mu \partial_t \mathbf{H}_i = -\mathbb{K}_i \mathbf{E}_i + \sum_{k \in V_i^i} \mathbb{S}_{ik} \mathbf{E}_i + \sum_{k \in V_i^i} \mathbb{S}_{ik}^+ \mathbf{E}_k \\ \quad + \sum_{k \in V_i^a} \mathbb{S}_{ik} \mathbf{E}_i + \sum_{k \in V_i^a} \mathbb{S}_{ik}^H \mathbf{H}_i + f_i^{inc,H}, \end{array} \right. \quad (8)$$

with

$$\begin{aligned} (\mathbb{M}_i^\varepsilon)_{jl} &= \int_{K_i} \Phi_{ij}^T \cdot \varepsilon_i \Phi_{il} dv, \\ (\mathbb{M}_i^\mu)_{jl} &= \int_{K_i} \Phi_{ij}^T \cdot \mu_i \Phi_{il} dv, \\ (\mathbb{K}_i)_{jl} &= \int_{K_i} \Phi_{ij}^T \cdot \nabla \times \Phi_{il} dv, \\ (\mathbb{S}_{ik})_{jl} &= \frac{1}{2} \int_{a_{ik}} \pi(\Phi_{ij}^T) \cdot \gamma(\Phi_{il}) ds, \\ (\mathbb{S}_{ik}^+)_{jlb} &= \frac{1}{2} \int_{a_{ik}} \pi(\Phi_{ij}^T) \cdot \gamma(\Phi_{kl}) ds, \\ (\mathbb{S}_{ik}^E)_{jl} &= \frac{1}{2} \sqrt{\frac{\varepsilon_i}{\mu_i}} \int_{a_{ik}} \pi(\Phi_{ij}^T) \cdot \pi(\Phi_{il}) ds, \\ (\mathbb{S}_{ik}^H)_{jl} &= \frac{1}{2} \sqrt{\frac{\mu_i}{\varepsilon_i}} \int_{a_{ik}} \pi(\Phi_{ij}^T) \cdot \pi(\Phi_{il}) ds, \\ f_i^{inc,E} &= \sum_{k \in V_i^a} \int_{a_{ik}} [\pi(\Phi_{ij}^T) \cdot \gamma(\mathbf{H}_i^{inc}) \\ &\quad - \frac{1}{2} \sqrt{\frac{\varepsilon_i}{\mu_i}} \pi(\Phi_{ij}^T) \cdot \pi(\mathbf{E}_i^{inc})] ds, \\ f_i^{inc,H} &= - \sum_{k \in V_i^a} \int_{a_{ik}} [\pi(\Phi_{ij}^T) \cdot \gamma(\mathbf{E}_i^{inc}) \\ &\quad + \frac{1}{2} \sqrt{\frac{\mu_i}{\varepsilon_i}} \pi(\Phi_{ij}^T) \cdot \pi(\mathbf{H}_i^{inc})] ds, \\ j_i^E &= \int_{K_i} \Phi_{ij}^T \cdot \mathcal{J}_i dv, \end{aligned}$$

where V_i^i denotes the set of indices of the elements which have a common interface with element K_i , V_i^a denotes the set of indices of fictitious neighboring elements introduced for imposing Silver-Müller ABC on element i , a_{ik} denotes the common face between element K_i and K_k , \mathbf{n}_{ik} is the unitary normal vector of the interface a_{ik} oriented from K_i to K_k . Moreover, \mathbb{M}_i^* (* stands for ε or μ) is the symmetric positive definite mass matrix, \mathbb{K}_i is the stiffness matrix, \mathbb{S}_{ik} is the local (in the sense of element K_i) square interface matrix and \mathbb{S}_{ik}^+ is the square (or rectangular if different interpolation orders are adopted in K_i and K_k) interface matrix merging basis functions of K_i and K_k , \mathbb{S}_{ik}^E and \mathbb{S}_{ik}^H is the local square matrix on the Silver-Müller boundary faces, $f_i^{inc,E}$ and $f_i^{inc,H}$ are the local face integration associated with the given incident field, and j_i^E is the local volume integration associated with current density.

IV. EXPONENTIAL TIME INTEGRATION FOR DGTD

To overcome the restriction on the time step when using a fully explicit time integration scheme in combination with a high order spatial discretization method formulated on a possibly locally refined mesh, we propose a new family of exponential-based integration methods based on the Lawson procedure. Exponential time integrators are usually applied to semi-linear systems of ordinary differential equations (ODEs). Thus, to construct the form of exponential time integrator that we will consider, we first derive a global version of the semi-discrete system of ODEs (8). By gathering the electric and magnetic unknowns of each element in column vectors of size $d = \sum_{i=1}^{N_t} d_i$, denoted by \mathbf{E} and \mathbf{H} , respectively, the local semi-discrete systems (8) for each element of the mesh can be transformed into the following global semi-discrete systems

$$\left\{ \begin{array}{l} \mathbb{M}^\varepsilon \partial_t \mathbf{E} = \mathbb{K} \mathbf{H} + \mathbb{S}^E \mathbf{E} + f^{inc,E} - j^E, \\ \mathbb{M}^\mu \partial_t \mathbf{H} = -\mathbb{K} \mathbf{E} + \mathbb{S}^H \mathbf{H} + f^{inc,H}, \end{array} \right. \quad (9)$$

where \mathbb{M}^* is a $d \times d$ block diagonal matrix with i th diagonal block be \mathbb{M}_i^* ; \mathbb{K} is a $d \times d$ block sparse matrix with diagonal blocks be $\mathbb{K}_i - \sum_{k \in V_i^i} \mathbb{S}_{ik}$ and the remaining nonzero blocks be $-\mathbb{S}_{ik}^+$; \mathbb{S}^E and \mathbb{S}^H are $d \times d$ block diagonal matrices with i th diagonal block be \mathbb{S}_i^E and \mathbb{S}_i^H respectively; $f^{inc,E}$, $f^{inc,H}$, and j^E are column vectors with i th element be $f_i^{inc,E}$, $f_i^{inc,H}$, and j_i^E respectively.

In practice, we pre-compute and store the inverse mass matrices of each element, then reuse them in time advancing for efficiency. Thus, the global semi-discrete systems can be rewritten as

$$\left\{ \begin{array}{l} \partial_t \mathbf{E} = \mathbb{M}^{-\varepsilon} \mathbb{K} \mathbf{H} + \mathbb{M}^{-\varepsilon} \mathbb{S}^E \mathbf{E} \\ \quad + \mathbb{M}^{-\varepsilon} (f^{inc,E} - j^E), \\ \partial_t \mathbf{H} = -\mathbb{M}^{-\mu} \mathbb{K} \mathbf{E} + \mathbb{M}^{-\mu} \mathbb{S}^H \mathbf{H} \\ \quad + \mathbb{M}^{-\mu} f^{inc,H}. \end{array} \right. \quad (10)$$

By gathering electric and magnetic unknowns into a single vector, we first rewrite the global semi-discrete systems (10) in form of ODEs as follows

$$\partial_t \mathbf{U} = \mathbb{A} \mathbf{B} \mathbf{U} + \mathbb{A} f_{inc}(t) - \mathbb{A} j(t), \quad (11)$$

where

$$\begin{aligned} \mathbf{U} &= \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix}, \\ \mathbb{A} &= \begin{pmatrix} \mathbb{M}^{-\varepsilon} & 0 \\ 0 & \mathbb{M}^{-\mu} \end{pmatrix}, \quad \mathbb{B} = \begin{pmatrix} \mathbb{S}^E & \mathbb{K} \\ -\mathbb{K} & \mathbb{S}^H \end{pmatrix}, \\ f_{inc}(t) &= \begin{pmatrix} f_{inc}^E(t) \\ f_{inc}^H(t) \end{pmatrix}, \quad j(t) = \begin{pmatrix} j^E(t) \\ 0 \end{pmatrix}. \end{aligned}$$

Let $\mathbb{C} = \mathbb{A} \mathbb{B}$, we have

$$\partial_t \mathbf{U} = \mathbb{C} \mathbf{U} + \mathbb{A} f_{inc}(t) - \mathbb{A} j(t). \quad (12)$$

A. Lawson exponential time integration for DGTD

We proceed as with the application of the Lawson procedure [26] to non-linear problems [27]-[28], which splits the problems into linear part and non-linear part and solve the linear part with high precision. We treat the unknowns corresponding to the locally refined part of the mesh as the linear part and calculate it with high accuracy, and treat the remaining unknowns as the non-linear part. Then we derive the Lawson exponential time integration for DGTD method formulated on a locally refined unstructured mesh. We first decompose the mesh \mathcal{T}_h into two parts: $\mathcal{T}_{h,f}$ is the subset containing the smallest elements of \mathcal{T}_h (according to an appropriate geometric criterion) and $\mathcal{T}_{h,c}$ is the remaining subset and $\mathcal{T}_{h,c} = \mathcal{T}_h \setminus \mathcal{T}_{h,f}$. We further assume that the number of elements in the subset $\mathcal{T}_{h,f}$ is far less than that of $\mathcal{T}_{h,c}$. We then split the unknown vector according to the decomposition of the mesh

$$\mathbf{U} = \mathbb{P}\mathbf{U} + (\mathbb{I} - \mathbb{P})\mathbf{U},$$

where \mathbb{P} is a diagonal matrix with diagonal entries equal to zero or one, to identify the unknowns associated with the locally refined part; \mathbb{I} is the identity matrix. Thus, according to (12) we have

$$\partial_t \mathbf{U} = \mathbb{C}\mathbb{P}\mathbf{U} + \mathbb{C}(\mathbb{I} - \mathbb{P})\mathbf{U} + \mathbb{A}f_{inc}(t) - \mathbb{A}j(t).$$

Setting

$$\mathbb{C}_f = \mathbb{C}\mathbb{P} \text{ and } \mathbb{C}_c = \mathbb{C}(\mathbb{I} - \mathbb{P}),$$

we have

$$\partial_t \mathbf{U} = \mathbb{C}_f \mathbf{U} + \mathbb{C}_c \mathbf{U} + \mathbb{A}f_{inc}(t) - \mathbb{A}j(t). \quad (13)$$

Now we introduce a new vector

$$\mathbf{V}(t) = e^{-t\mathbb{C}_f} \mathbf{U}(t).$$

After replacing the original state vector $\mathbf{U}(t)$ by this new vector in (13), the left-hand side becomes

$$\partial_t (e^{t\mathbb{C}_f} \mathbf{V}) = \mathbb{C}_f e^{t\mathbb{C}_f} \mathbf{V} + e^{t\mathbb{C}_f} \partial_t \mathbf{V}, \quad (14)$$

and the right-hand side becomes

$$\mathbb{C}_f e^{t\mathbb{C}_f} \mathbf{V} + \mathbb{C}_c e^{t\mathbb{C}_f} \mathbf{V} + \mathbb{A}f_{inc}(t) - \mathbb{A}j(t). \quad (15)$$

Comparing the above two new terms, we have

$$\partial_t \mathbf{V} = e^{-t\mathbb{C}_f} \mathbb{C}_c e^{t\mathbb{C}_f} \mathbf{V} + e^{-t\mathbb{C}_f} [\mathbb{A}f_{inc}(t) - \mathbb{A}j(t)] \quad (16)$$

Note that the transformed system (16) can be time integrated using an explicit Runge-Kutta scheme or another high order accurate explicit time integration technique. The result is then back transformed to provide an approximation in the variable \mathbf{U} . The purpose of transforming the differential equation in the above-described way is to remove the explicit dependence in the differential equation on the operator \mathbb{C}_f , except inside the exponential. Notice that, in (16), the spectrum of $e^{-t\mathbb{C}_f} \mathbb{C}_c e^{t\mathbb{C}_f}$ is the same as that of \mathbb{C}_c , since the product of $e^{-t\mathbb{C}_f}$ and $e^{t\mathbb{C}_f}$ is an identity matrix and thus $e^{-t\mathbb{C}_f} \mathbb{C}_c e^{t\mathbb{C}_f}$ is a similarity transformation of \mathbb{C}_c . In other words, the characteristic of the locally refined part has no impact on the stability of the system (16), which is only dominated by the coarse part. Thus the stability of the Lawson exponential time integration

method for the locally refined part of the mesh is excellent or even unconditional. Therefore, the particular exponential time integration method considered here not only removes the stiffness due to the refined part of the mesh on the allowable time step size, but also reverses the global explicitness when time integrated using an explicit time integration technique. Moreover, the global explicitness of the obtained method and the data locality of DG formulation make it easily parallelized. These properties of Lawson method enable the possibility to design a solution strategy which is much more efficient than the one obtained by applying an explicit time integration scheme to the original system.

B. Combined Lawson-LSRK time scheme

As mentioned in the previous subsection, the Lawson exponential time integration method can be further time integrated using various explicit time integration techniques. To obtain an efficient high order method, we focus on the widely used Low-Storage Runge-Kutta (LSRK) scheme [29] in this paper. Runge-Kutta schemes are a class of multi-stage algorithms that rely on multiple evaluations of the RHS of (16) to evolve the system in time. Unlike leap-frog schemes, they do not combine different time levels to eliminate terms in Taylor expansions of the unknown function, which leads to dispersive and dissipative schemes. Suppose that one formally integrates the system

$$\partial_t \mathbf{V} = f(t, \mathbf{V}(t)),$$

between time stations t and $t + \Delta t$

$$\begin{aligned} & \mathbf{V}(t + \Delta t) \\ &= \mathbf{V}(t) + \int_t^{t+\Delta t} f(\tau, \mathbf{V}(\tau)) d\tau \\ &\simeq \mathbf{V}(t) + \Delta t \sum_{j=1}^s b_j f(t + c_j \Delta t, \mathbf{V}(t + c_j \Delta t)). \end{aligned} \quad (17)$$

LSRK is a widely used efficient Runge-Kutta scheme, in which the required memory is limited compared with classical Runge-Kutta schemes. It can be expressed as in Algorithm 1. In this algorithm, ϕ_1 and ϕ_2 are the two unknown vectors required to be stored during time advancing; a_k , b_k and c_k are the coefficients of the LSRK scheme; s defines the number of stages of the LSRK scheme. To time advance the

Algorithm 1 Low-storage Runge-Kutta (LSRK) scheme.

- 1: $\phi_1^0 = \mathbf{V}_n$
 - 2: **for** $k = 1 : s$ **do**
 - 3: $\phi_2^k = a_k \phi_2^{k-1} + \Delta t f(t + c_k \Delta t, \phi_1^{k-1})$
 - 4: $\phi_1^k = \phi_1^{k-1} + b_k \phi_2^k$
 - 5: **end for**
 - 6: $\mathbf{V}_{n+1} = \phi_1^s$
-

ODEs (16) of Lawson exponential time integration with the efficient LSRK scheme, the right-hand side of (16) is used to replace $f(t + c_k \Delta t, \phi_1^{k-1})$ in Algorithm 1. Then we obtain the algorithm of the combined Lawson-LSRK scheme, which is described in Algorithm 2.

Algorithm 2 Combined Lawson-LSRK scheme.

```

1:  $\phi_1^0 = \mathbf{V}_n$ 
2: for  $k = 1 : s$  do
    $\phi_2^k = a_k \phi_2^{k-1} + \Delta t e^{-(t_n + c_k \Delta t) \mathbb{C}_f} \mathbb{C}_c e^{(t_n + c_k \Delta t) \mathbb{C}_f} \phi_1^{k-1}$ 
3:    $+ e^{-(t_n + c_k \Delta t) \mathbb{C}_f} \mathbb{A} f_{inc}(t_n + c_k \Delta t)$ 
    $- e^{-(t_n + c_k \Delta t) \mathbb{C}_f} \mathbb{A} j(t_n + c_k \Delta t)$ 
4:    $\phi_1^k = \phi_1^{k-1} + b_k \phi_2^k$ 
5: end for
6:  $\mathbf{V}_{n+1} = \phi_1^s$ 

```

V. COMPUTER IMPLEMENTATION ASPECTS

In this section, we present some techniques for an efficient implementation of the proposed exponential time integration method.

A. Transformation of the combined Lawson-LSRK scheme

Obviously, Algorithm 2 is not in a form suitable for a practical implementation due to the presence of matrices $e^{\pm(t_n + c_k \Delta t) \mathbb{C}_f}$. One has to evaluate the matrix exponential with coefficients $t_n + c_k \Delta t$ vary from stage (or time step) to stage (or time step), which is very expensive. Moreover, to obtain the original unknown vector \mathbf{U}_{n+1} , one should compute another action of matrix exponential $\mathbf{U}_{n+1} = e^{t_{n+1} \mathbb{C}_f} \mathbf{V}_{n+1}$. We note that $\mathbf{V}(t) = e^{-t \mathbb{C}_f} \mathbf{U}(t)$, thus matrix-vector products of the form $e^{t_n \mathbb{C}_f} \mathbf{V}_n$ are approximations to the solution of the original problem $\mathbf{U}(t_n)$. To improve the efficiency of the combined Lawson-LSRK scheme, we introduce new variable vectors

$$\begin{cases} \mathbf{U}_n = e^{t_n \mathbb{C}_f} \mathbf{V}_n, \\ \phi_2^* = e^{(t_n + c_k \Delta t) \mathbb{C}_f} \phi_2, \\ \phi_1^* = e^{(t_n + c_k \Delta t) \mathbb{C}_f} \phi_1. \end{cases} \quad (18)$$

To implement this transformation, we multiply both sides of the equations from Algorithm 2 by the exponential term $e^{(t_n + c_k \Delta t) \mathbb{C}_f}$. Thus, for each stage we have

$$\begin{cases} \phi_2^{(k)*} = e^{(t_n + c_k \Delta t) \mathbb{C}_f} \phi_2^{(k)} \\ = a_k e^{(c_k - c_{k-1}) \Delta t \mathbb{C}_f} \phi_2^{(k-1)*} \\ + \Delta t \mathbb{C}_c e^{(c_k - c_{k-1}) \Delta t \mathbb{C}_f} \phi_1^{(k-1)*}, \\ \phi_1^{(k)*} = e^{(t_n + c_k \Delta t) \mathbb{C}_f} \phi_1^{(k)} \\ = e^{(c_k - c_{k-1}) \Delta t \mathbb{C}_f} \phi_1^{(k-1)*} + b_k \phi_2^{(k)*}. \end{cases} \quad (19)$$

Let

$$\phi_1^{(k-1)**} = e^{(c_k - c_{k-1}) \Delta t \mathbb{C}_f} \phi_1^{(k-1)*}.$$

Then the algorithm of the combined Lawson-LSRK scheme can be transformed to Algorithm 3. Algorithm 3 is more appropriate for the practical implementation than Algorithm 2 since the coefficients of the matrix exponentials do not change while time advancing, which allows for a more efficient evaluation of matrix exponentials. In addition, we do not have to compute matrix exponentials associated with the incident and volume source terms. Moreover, we can pre-compute $s+1$ matrix exponentials directly and store them before the time iteration if the matrix \mathbb{C}_f is very small.

Algorithm 3 Transformed Lawson-LSRK scheme.

```

1:  $\phi_1^{(0)*} = \mathbf{U}_n$ 
2: for  $k = 1 : s$  do
    $\phi_2^{(k)*} = a_k e^{(c_k - c_{k-1}) \Delta t \mathbb{C}_f} \phi_2^{(k-1)*} + \Delta t \mathbb{C}_c \phi_1^{(k-1)**}$ 
3:    $+ \mathbb{A} f_{inc}(t_n + c_k \Delta t) - \mathbb{A} j(t_n + c_k \Delta t)$ 
4:    $\phi_1^{(k)*} = \phi_1^{(k-1)**} + b_k \phi_2^{(k)*}$ 
5: end for
6:  $\mathbf{U}_{n+1} = e^{(1 - c_s) \Delta t \mathbb{C}_f} \phi_1^{(s)*}$ 

```

B. Model reduction for exponential time integration

From Algorithm 3, matrix exponentials in form of $e^{-\alpha \mathbb{C}_f}$ (where α is a constant) must be computed. However, \mathbb{C}_f is a global block sparse matrix with dimension $2d \times 2d$. The computation of these matrix exponentials can thus be an expensive task. Moreover, the overhead of computing matrix exponentials will increase exponentially with the increase of the dimension of the matrix. For practical problems, the global matrices often have large dimensions. Therefore, for the computation of those matrix exponentials, the scaling and squaring method based on Padé approximation [32], an efficient method for computing the exponential of a small dense matrix, is not feasible. Alternatively, efficient Krylov subspace methods can be used, but they still require many operations. In our setting, the number of very small elements in the mesh is assumed to be much less than that of large elements. If one can only compute the exponential of a small matrix linked to the degrees of freedom in the smallest mesh elements, the overall computational work will decrease significantly. Since the unknowns associated to the elements in the coarse part of the mesh are zero in \mathbb{C}_f and the number of very small mesh elements is assumed to be much less than that of large elements, \mathbb{C}_f is a large and very sparse matrix. Assuming the matrix \mathbb{C} can be constructed as a 2-by-2 block matrix

$$\mathbb{C} = \begin{pmatrix} \mathbb{C}_{ff} & \mathbb{C}_{fc} \\ \mathbb{C}_{cf} & \mathbb{C}_{cc} \end{pmatrix}, \quad (20)$$

where \mathbb{C}_{ff} (respectively \mathbb{C}_{cc}) consists of the degrees of freedom associated to the elements inside the fine (respectively coarse) part. The remaining blocks, \mathbb{C}_{fc} and \mathbb{C}_{cf} , correspond to the coupling faces between the coarse and fine parts. Thus, the corresponding diagonal projection matrix \mathbb{P} for splitting the unknown vector must has the following form

$$\mathbb{P} = \begin{pmatrix} \mathbb{I}_{ff} & 0 \\ 0 & 0 \end{pmatrix}.$$

Then we have

$$\mathbb{C}_f = \mathbb{C} \mathbb{P} = \begin{pmatrix} \mathbb{C}_{ff} & 0 \\ \mathbb{C}_{cf} & 0 \end{pmatrix}. \quad (21)$$

We recall the φ -functions [33]

$$\begin{cases} \varphi_k(-t\mathbb{A}) = \int_0^t e^{-(t-\tau)\mathbb{A}} \frac{\tau^{k-1}}{(k-1)!} d\tau & k \geq 1, \\ \varphi_0(-t\mathbb{A}) = e^{-t\mathbb{A}} \end{cases} \quad (22)$$

which satisfy the recurrence relation

$$\begin{cases} \varphi_{k+1}(-t\mathbb{A}) = \frac{\varphi_k(-t\mathbb{A}) - \frac{1}{k!}}{-t\mathbb{A}} \\ \varphi_k(0) = \frac{1}{k!}. \end{cases} \quad (23)$$

It can be easily proved that

$$\mathbb{C}_f^n = \begin{pmatrix} \mathbb{C}_{ff} & 0 \\ \mathbb{C}_{cf} & 0 \end{pmatrix}^n = \begin{pmatrix} \mathbb{C}_{ff}^n & 0 \\ \mathbb{C}_{cf}\mathbb{C}_{ff}^{n-1} & 0 \end{pmatrix}, n \geq 1. \quad (24)$$

Let $\varphi_0(x) = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$ be the Taylor expansion of $\varphi_0(x)$, we have

$$\begin{aligned} e^{-\alpha\mathbb{C}_f} &= \varphi_0(-\alpha\mathbb{C}_f) \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} (-\alpha\mathbb{C}_f)^n = \mathbb{I} + \sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha\mathbb{C}_f)^n \\ &= \begin{pmatrix} \mathbb{I}_{ff} & 0 \\ 0 & \mathbb{I}_{cc} \end{pmatrix} + \sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha)^n \begin{pmatrix} \mathbb{C}_{ff}^n & 0 \\ \mathbb{C}_{cf}\mathbb{C}_{ff}^{n-1} & 0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{I}_{ff} + \sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha)^n \mathbb{C}_{ff}^n & 0 \\ \sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha)^n \mathbb{C}_{cf}\mathbb{C}_{ff}^{n-1} & \mathbb{I}_{cc} \end{pmatrix}, \end{aligned} \quad (25)$$

where

$$\mathbb{I}_{ff} + \sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha)^n \mathbb{C}_{ff}^n = \varphi_0(-\alpha\mathbb{C}_{ff})$$

and

$$\begin{aligned} &\sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha)^n \mathbb{C}_{cf}\mathbb{C}_{ff}^{n-1} \\ &= \mathbb{C}_{cf} \left[\mathbb{C}_{ff}^{-1} - \mathbb{C}_{ff}^{-1} + \sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha)^n \mathbb{C}_{ff}^n \mathbb{C}_{ff}^{-1} \right] \\ &= \mathbb{C}_{cf} \left[\mathbb{I}_{ff} - \mathbb{I}_{ff} + \sum_{n=1}^{\infty} \frac{1}{n!} (-\alpha)^n \mathbb{C}_{ff}^n \right] \mathbb{C}_{ff}^{-1} \\ &= \mathbb{C}_{cf} [\varphi(-\alpha\mathbb{C}_{ff}) - \mathbb{I}_{ff}] \mathbb{C}_{ff}^{-1} \\ &= -\alpha\mathbb{C}_{cf}\varphi_1(-\alpha\mathbb{C}_{ff}). \end{aligned}$$

Then, we have

$$e^{-\alpha\mathbb{C}_f} = \varphi_0(-\alpha\mathbb{C}_f) = \begin{pmatrix} \varphi(-\alpha\mathbb{C}_{ff}) & 0 \\ -\alpha\mathbb{C}_{cf}\varphi_1(-\alpha\mathbb{C}_{ff}) & \mathbb{I}_{cc} \end{pmatrix}. \quad (26)$$

Therefore, rather than evaluating the exponential $e^{-\alpha\mathbb{C}_f}$ of a large sparse matrix directly, one can compute the exponential $e^{-\alpha\mathbb{C}_{ff}}$ instead. Since the number of elements in the subset $\mathcal{T}_{h,f}$ is assumed to be much less than that of $\mathcal{T}_{h,c}$, the order of the matrix required to compute exponential reduces significantly. Thus, the computational overheads will be reduced significantly.

To construct the block matrix form (20), we have to reorder the rows and columns of the matrix \mathbb{C} . This reordering can be expressed as a permutation matrix \mathbb{Q} and

$$\widehat{\mathbb{C}} = \mathbb{Q}\mathbb{C}\mathbb{Q}^T.$$

To obtain the form of the transformed system (16) of Lawson exponential time integration, we first use \mathbb{Q} to reorder the unknown vector \mathbf{U} and the rows of \mathbb{C} as follows

$$\partial_t(\mathbb{Q}\mathbf{U}) = \mathbb{Q}\mathbb{C}\mathbf{U}.$$

Since \mathbb{Q} is a permutation matrix, we have that

$$\mathbb{Q}\mathbb{Q}^T = \mathbb{Q}^T\mathbb{Q} = \mathbb{I},$$

therefore

$$\partial_t(\mathbb{Q}\mathbf{U}) = \mathbb{Q}\mathbb{C}\mathbb{Q}^T\mathbb{Q}\mathbf{U} = \widehat{\mathbb{C}}\mathbb{Q}\mathbf{U}. \quad (27)$$

Let $\mathbf{W}(t) = \mathbb{Q}\mathbf{U}(t)$, we have

$$\partial_t(\mathbf{W}) = \widehat{\mathbb{C}}\mathbf{W}, \quad (28)$$

which has the same form as (16). Therefore, we can easily obtain a more efficient algorithm by following the procedures and algorithms described in subsections IV-A and IV-B.

C. Efficient computation of matrix exponential

There are several ways to compute the matrix exponential $e^{-t\mathbb{A}}$ (or the related matrix functions $\varphi(-t\mathbb{A})$) for a given square matrix \mathbb{A} that can be classified into two kinds. The first kind of methods comprise methods based on Taylor developments [34], Padé approximates [32], scaling and squaring methods [35] with Padé or Taylor approximations, and so on. These methods are designed for computing the matrix exponential $e^{-t\mathbb{A}}$ of a small dense matrix \mathbb{A} directly. Among these methods, scaling and squaring methods are generally competitive, which are efficient for small dense matrices. However, for the computation of large matrix exponential, these methods are inefficient or even infeasible. The second kind of methods include Krylov subspace methods [23]-[36]-[22] and Chebyshev polynomials based methods [37]. They allow to compute the product $e^{-t\mathbb{A}}\mathbf{v}$ for a given vector \mathbf{v} and various values of t . These methods are very effective for large sparse problem. The Chebyshev polynomials based methods are mostly used for computing the matrix exponentials of symmetric or skew-symmetric matrices. Computing matrix exponentials with these methods for general matrices is possible but not trivial [38]. Krylov subspace based methods seem to combine versatility and efficiency, and are more effective for general problems [32].

Note that in Algorithm 3, the coefficients c_k of the LSRK scheme are known. If the matrix \mathbb{C}_f is very small and the first kind of methods are adopted, we can pre-compute $s+1$ matrix exponentials and store them before starting the time-stepping loop instead of computing them at each time step. This decreases the computational overheads significantly at the expense of an increase in the memory requirement.

For generalization and efficiency, we adopt the more efficient Krylov subspace type methods. In these methods the matrix-vector product $e^{\mathbb{A}}\mathbf{v}$ is approximated as following

$$e^{\mathbb{A}}\mathbf{v} \approx [\widehat{\mathbf{v}}, \mathbb{A}\widehat{\mathbf{v}}, \dots, \mathbb{A}^{m-1}\widehat{\mathbf{v}}] e^{\mathbb{H}_m} \mathbf{e}_1 = \mathbb{V}_m e^{\mathbb{H}_m} \mathbf{e}_1, \quad (29)$$

where \mathbb{A} is a $n \times n$ matrix, $\widehat{\mathbf{v}}$ is a Krylov subspace basis vector with dimension n , which is usually obtained by normalizing the given vector \mathbf{v} to $\widehat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$, m is the dimension of

Krylov subspace which is typically smaller than the dimension n of \mathbb{A} , \mathbb{H}_m is a $m \times m$ upper Hessenberg matrix, \mathbf{e}_1 is the first column of \mathbb{I}_m . Accurate enough approximations are often obtained with relatively small m , and computable error bounds exist for the approximation. Thus, the large sparse $e^{\mathbb{A}}$ problem is replaced with a small dense $e^{\mathbb{H}_m}$ problem which can be computed by the scaling and squaring methods based on Padé approximations efficiently. The cost of computing the expression $\nabla_m e^{\mathbb{H}_m} \mathbf{e}_1$ is usually much smaller than the cost needed to compute $e^{\mathbb{A}} \mathbf{v}$ [27]. In this paper, the routines EXPV and PHIV in Expokit [39], a software package that provides matrix exponential routines for very large sparse matrices based on Krylov subspace approximations, are used to compute $e^{t\mathbb{A}} \mathbf{v}$ and $\varphi_1(t\mathbb{A}) \mathbf{v}$, respectively.

VI. NUMERICAL EXPERIMENTS

In this section, several numerical experiments are presented to illustrate the stability, investigate the accuracy and expected order of convergence, and demonstrate the computational performance of the proposed exponential-based Lawson-LSRK time integration method. Firstly, we consider the problem of the propagation of a plane wave in vacuum for which a simple analytical solution is available. We then present the problem of the radiation of a localized source in a PEC composite structure. Note that the fourth order five stages Low-Storage Runge-Kutta (LSRK(5,4)) scheme is adopted in the following numerical experiments. The numerical experiments are performed on a workstation equipped with an Intel Xeon CPU running at 3.70 GHz with 32 GB of RAM memory, the codes are implemented serially.

A. Plane wave propagation in vacuum

We consider a cube of edge 1 m ($\Omega = [0, 1]^3$) filled with vacuum. We apply the Silver-Müller ABC condition on each side of the cube, which is illuminated by a plane wave $\mathcal{E}_{inc} = \mathcal{E}_0 \cos\{w_0[t - \mathbf{k}(\mathbf{x} - \mathbf{x}_0)/c_0]\}$ with frequency 300 MHz, where $\mathcal{E}_0 = (1, 0, 0)$, $\mathbf{k} = (0, 0, 1)$, $\mathbf{x}_0 = (0, 0, 0)$; w_0 indicates angular frequency, and c_0 indicates the speed of light in vacuum. In order to introduce a grid-induced stiffness, several meshes with local refinement in the center of the cube are used to assess the accuracy and perform a numerical convergence study of the combined Lawson-LSRK scheme. The characteristics of these meshes are given in Table I, where N_v indicates the number of vertices, N_t the number of tetrahedra, and h_{max} (relatively h_{min}) the maximal (relatively minimal) edge length in the meshes. Figure 1 gives a 3D view of the mesh \mathbf{M}_4 , where the mesh elements in the fine part are marked by red. In this subsection, the Expokit routines are used with a maximal dimension m of the Krylov subspace set to 4, while the threshold tol for monitoring the accuracy of the matrix exponential calculation is set to be 0.5 (relative tolerance). As a general rule, keeping m small allows to minimize the computational work and memory usage. However, the convergence of the Krylov method may deteriorate when decreasing m too much, and a large number iterations may be required to achieve the required accuracy.

TABLE I: Plane wave propagation in vacuum: characteristics of locally refined meshes.

	\mathbf{M}_1	\mathbf{M}_2	\mathbf{M}_3	\mathbf{M}_4
N_v	635	1 887	4 203	7 759
N_t	2 968	9 336	21 496	40 616
h_{max}	2.89E-01	1.96E-01	1.40E-01	1.13E-01
h_{min}	2.60E-03	1.25E-03	9.20E-04	8.68E-04
h_{max}/h_{min}	110.9	156.7	152.0	130.4

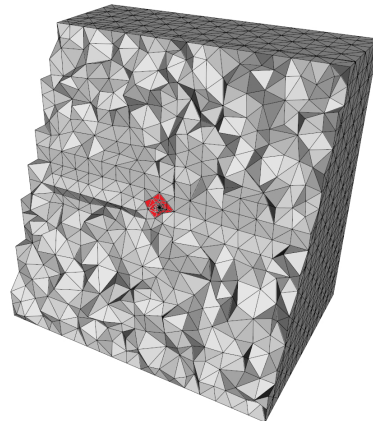


Fig. 1: Plane wave propagation in vacuum: partial view of mesh \mathbf{M}_4 .

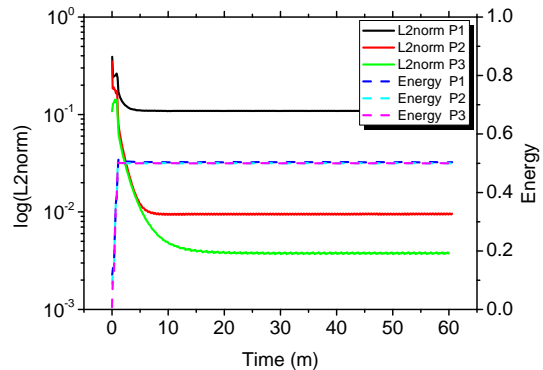


Fig. 2: Plane wave propagation in vacuum: time evolution of the global L^2 norm of the error and the time evolution of the energy obtained by the DGTD methods based on the combined Lawson-LSRK scheme.

We first investigate the stability of the combined Lawson-LSRK scheme. Figure 2 shows the time evolution of the global L^2 -norm of the error and the energy for the DGTD method based on the combined Lawson-LSRK scheme using mesh \mathbf{M}_1 . The simulation time is set to 60 periods of the incident plane wave. It is seen that the global L^2 -norm of the error decreases rapidly and stabilizes to a limit value, which illustrates the stability of the proposed Lawson-LSRK scheme. To investigate the convergence order of the combined Lawson-LSRK scheme, we then measure the global L^2 -norm of the error for the sequence of four successively locally refined tetrahedral meshes given in Table I. The error is plotted as a

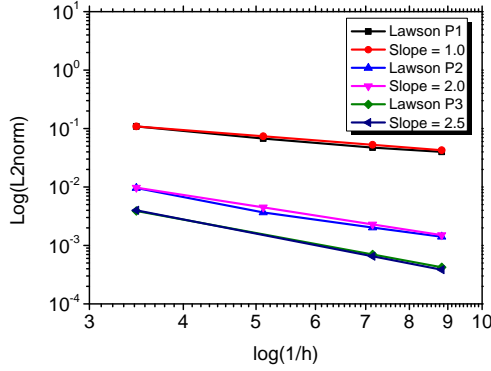
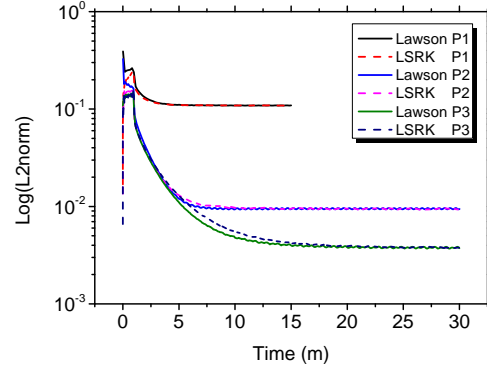
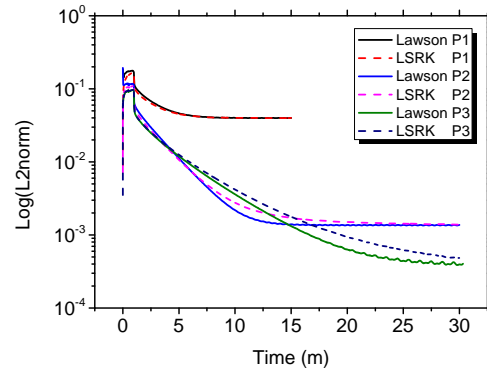


Fig. 3: Plane wave propagation in vacuum: numerical convergence order obtained by the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme.

function of $1/h$, in logarithmic scale. The DGTD- \mathbb{P}_k methods with $k = 1, 2$ or 3 are considered (note that in this paper, \mathbb{P}_3 indicates the incomplete order curl-conforming vector basis presented in [7], while for the other lower interpolation orders, the complete order curl-conforming vector basis presented in [30] is used). The measured convergence orders in Figure 3 are compatible with the expected theoretical behavior. The order of convergence for the DGTD- \mathbb{P}_1 and DGTD- \mathbb{P}_2 methods are 1.03 and 2.02, respectively. As for the DGTD- \mathbb{P}_3 method, the order of convergence is 2.36, which is slightly weaker than the theoretical order 2.5. These results also confirm that the subdivision into coarse and fine elements is not detrimental to the convergence order of the combined Lawson-LSRK scheme. Figure 4 shows that the time evolution of the global L^2 -norm of the error obtained with the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme and the fully explicit LSRK scheme converge to the same accuracy. Note that the DGTD- \mathbb{P}_k methods based on the fully explicit LSRK scheme take more physical time to converge than the methods based on the combined Lawson-LSRK scheme. We observe that this phenomenon is more apparent when the interpolation order increases. Moreover, by comparing the two sub-figures in Figure 4, we also observe that this phenomenon becomes more apparent with the refinement of the mesh. In addition, from Figure 5, we observe that the time evolution of the electric fields at selected locations in the coarse part and fine part of the mesh match the exact solution very well. This confirms that the proposed Lawson-LSRK scheme is accurate for locally refined meshes. Now we investigate the computational performance of the combined Lawson-LSRK scheme in comparison with that of the fully explicit LSRK scheme. The physical simulation time is set to $T = 1$ m which corresponds to one period of the incident plane wave. The locally refined mesh \mathbf{M}_4 is used. The characteristics of the matrices for DGTD- \mathbb{P}_k methods are summarized in Table II, where Dim and Nz indicate the matrix order and the number of nonzero terms respectively. The obtained results are summarized in Table III. We note that the proposed Lawson-LSRK scheme allows to reduce noticeably the required number



(a)



(b)

Fig. 4: Plane wave propagation in vacuum: time evolution of the global L^2 -norm of the error obtained with the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme versus the fully explicit LSRK scheme using (a) mesh \mathbf{M}_1 and (b) mesh \mathbf{M}_4 .

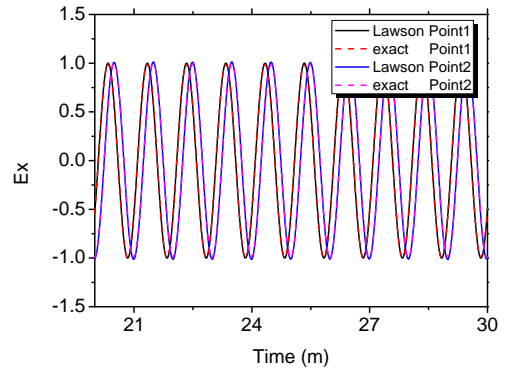


Fig. 5: Plane wave propagation in vacuum: time evolution of the electric fields at the points with coordinates $(0.35, 0.35, 0.35)$, in the coarse part, $(0.5, 0.5, 0.5)$ in the fine part, obtained with the DGTD- \mathbb{P}_2 method based on the combined Lawson-LSRK scheme using mesh \mathbf{M}_4 .

of time iterations to reach the final physical time due to the increase of the allowable time step size. As a result,

TABLE II: Plane wave propagation in vacuum: characteristics of matrices for the DGTD- \mathbb{P}_k methods.

\mathbb{P}_k	Dim	Dim C_{ff}	Nz C_{ff}
\mathbb{P}_1	974 784	12 288	437 760
\mathbb{P}_2	2 436 960	30 720	2 905 979
\mathbb{P}_3	3 655 440	46 080	4 795 192

despite the overhead of computing matrix exponentials, the new scheme outperforms the fully explicit LSRK scheme in terms of CPU time. Herein, the time step size of the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme is chosen to be 32 times larger than that of the fully explicit LSRK scheme. We can observe that the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme is much faster than the methods based on the classical fully explicit LSRK scheme, which yields 24.3, 20.5, and 20.3 speedups for the DGTD- \mathbb{P}_1 to DGTD- \mathbb{P}_3 methods respectively. Note that we obtain a significant CPU gain only with the price of slight memory usage increase, which is around 2.5% here. Figure 6 indicates that the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme outperform the methods based on the fully explicit LSRK scheme for efficiency, where the former is able to converge to a given accuracy much faster than the later. For DGTD- \mathbb{P}_k methods based on the fully explicit LSRK scheme, the number of arithmetic operations is almost constant during time advancing and thus the CPU time for each time step is the same. However, the situation is different for the Lawson-LSRK scheme since a Krylov method is adopted to compute the action of matrix exponential $e^{t\mathbb{A}}\mathbf{v}$, in which the vector \mathbf{v} changes with the time advancing. Therefore, the number of Krylov iterations is not constant and thus the CPU time for each time step changes with time advancing. The number of Krylov iterations for the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme is plotted in Figure 7. We observe that the number of Krylov iteration of each time step for DGTD- \mathbb{P}_2 and DGTD- \mathbb{P}_3 methods fluctuate at the beginning of time advancing and tend to be stable with the time advancing. The average number of Krylov iterations at each time step for the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme are respectively 10.7, 11.2, and 11.6 for $k = 1, 2, 3$. For the DGTD- \mathbb{P}_2 and DGTD- \mathbb{P}_3 methods, the number of Krylov iterations are always larger than the final stable number. That is to say, the CPU time for each time step is more likely larger at the beginning of the time advancing. Moreover, as mentioned previously, the DGTD- \mathbb{P}_k method based on the combined Lawson-LSRK scheme requires less physical time to converge to a periodic regime than when using a fully explicit LSRK scheme, especially for the more refined meshes or for higher values of the interpolation order k . Thus, in these conditions, the advantage in terms of efficiency of the combined Lawson-LSRK scheme against the fully explicit LSRK scheme will be more apparent.

Table IV summarizes performance figures of the DGTD- \mathbb{P}_1 method based on the two time schemes. Here the time step size of the combined Lawson-LSRK scheme is 32 times larger than that of the fully explicit LSRK scheme. We observe that

TABLE III: Plane wave propagation in vacuum: performance figures of the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme versus the fully explicit LSRK scheme using mesh \mathbf{M}_4 .

\mathbb{P}_k	$\frac{\Delta t_{\text{Lawson}}}{\Delta t_{\text{LSRK}}}$	CPU (mn)			Peak Mem (MB)	
		LSRK	Lawson	Gain	LSRK	Lawson
\mathbb{P}_1	32	73	3	24.3	511	522
\mathbb{P}_2	32	205	10	20.5	1 804	1 850
\mathbb{P}_3	32	507	25	20.3	3 466	3 547

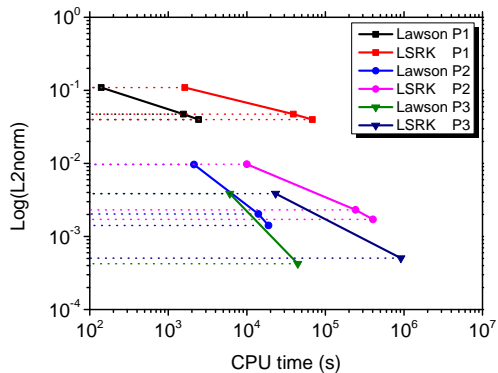


Fig. 6: Plane wave propagation in vacuum: maximal L^2 -norm of the error of electric fields in function of CPU time obtained for DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme versus the fully explicit LSRK scheme.

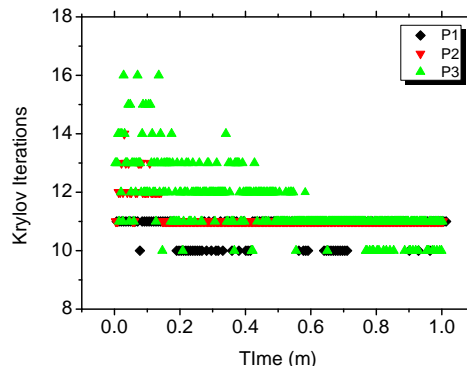


Fig. 7: Plane wave propagation in vacuum: number of Krylov iterations for DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme.

the speedup between the two time schemes increases when reducing the percentage of mesh elements in the fine part (R_f is the proportion of refined elements in the entire mesh). It is shown in Figure 8 that the speedup of the DGTD- \mathbb{P}_3 method based on the combined Lawson-LSRK scheme versus the fully explicit LSRK scheme increases steadily with the ratio of the time step sizes of the former to that of the later for a given mesh. We also notice that the growth rate of the speedup gradually reduces with an increasing R_f . The reason is that the dimension of the matrix C_{ff} becomes larger and

TABLE IV: Plane wave propagation in vacuum: performance figures of the DGTD- \mathbb{P}_1 method based on the combined Lawson-LSRK scheme versus the fully explicit LSRK scheme using meshes with different refinement ratio.

Mesh	R_f (%)	CPU (sec)			Peak Mem (MB)	
		LSRK	Lawson	Gain	LSRK	Lawson
\mathbf{M}_1	18.1	104	9	11.6	51	59
\mathbf{M}_3	2.6	2 510	100	25.1	269	286
\mathbf{M}_4	1.3	4 389	156	28.1	511	522

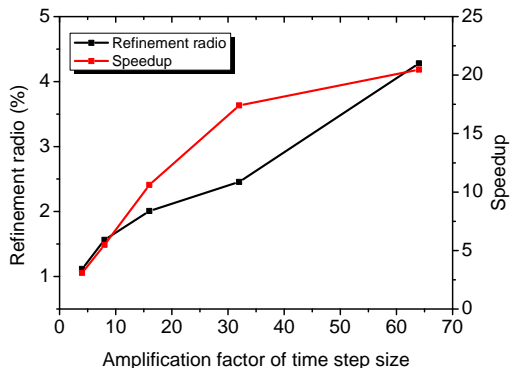


Fig. 8: Plane wave propagation in vacuum: percentage of refined elements and speedup of the DGTD- \mathbb{P}_3 method based on the combined Lawson-LSRK scheme versus the fully explicit LSRK scheme as a function of the time step size ratio of the former to the later using mesh \mathbf{M}_3 .

thus the overhead for exponential calculation increases.

B. Composite structure with a localized radiation source

We now consider a composite structure made of a PEC sphere and a small PEC cylinder, where the sphere is of radius 1 m and the cylinder of radius 0.02 m and height 0.1 m as shown in Figure 9. A localized radiation source is placed in the gap between these two structures, where the size of the gap is 0.1 m. The computational domain is truncated by a Silver-Müller ABC. The localized source $\mathcal{J}_z(\mathbf{x}, t) = \sin(\omega t)e^{-(\mathbf{x}-\mathbf{x}_s)^2}$, with frequency $f = 300$ MHz is oriented along the z -axis, and localized at the midpoint $\mathbf{x}_s = (-1.05, 0, 0)$ of the gap. To capture the localized source accurately, the elements around the source point are locally refined and marked in red in Figure 9 (b). The characteristics of this mesh are given in Table V. In this case, the settings of the Krylov subspace method are $m = 4$ and $tol = 10^{-1}$.

Note that there is no analytical solution for this case. The numerical solutions obtained with the DGTD- \mathbb{P}_k ($k = 1, 2, 3$) methods based on the combined Lawson-LSRK scheme are compared with those resulting from the methods based on the fully explicit LSRK scheme to verify the accuracy of the proposed time scheme. The time evolution of the electric components at a given point obtained by the DGTD- \mathbb{P}_1 method based on the combined Lawson-LSRK scheme and the fully explicit LSRK scheme is plotted in Figure 10. We

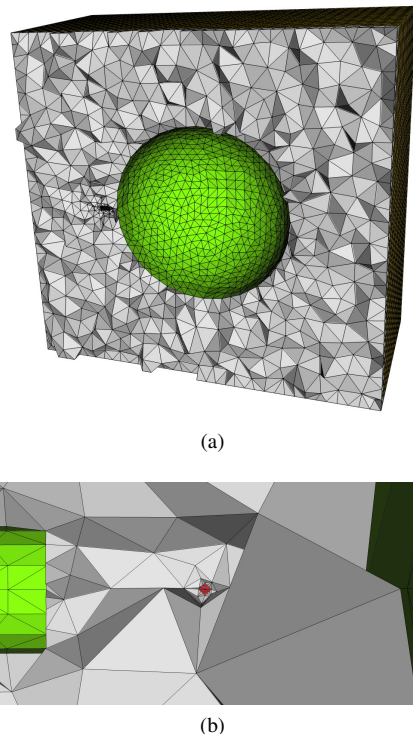
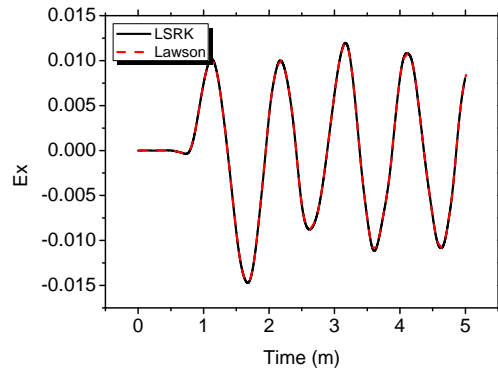


Fig. 9: Composite structure with a localized radiation source: View of the composite structure mesh.

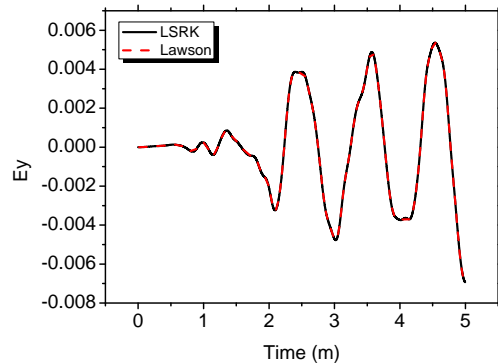
TABLE V: Composite structure with a localized radiation source: characteristics of the composite structure mesh.

N_v	N_t	h_{max}	h_{min}	h_{max}/h_{min}
16 805	76 755	3.96E-01	6.73E-04	188

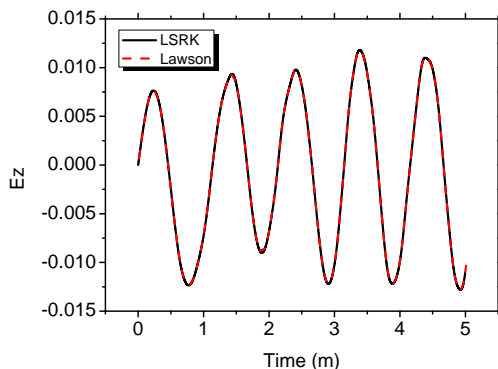
observe that the solution obtained by the methods based on those two time schemes agree very well. Figure 11 gives the electric distribution, in logarithmic scale, of three cross sections obtained by the DGTD- \mathbb{P}_3 method based on the combined Lawson-LSRK scheme at a given time in the last period of simulation. To further investigate the efficiency of the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme, performance figures are again compared with those obtained by the methods based on the fully explicit LSRK scheme. The physical simulation time is set to $T = 1$ m. Results are summarized in Table VI. The characteristics of the corresponding matrices are given in Table VII. It is shown that the DGTD- \mathbb{P}_1 method based on the fully explicit LSRK scheme requires 170 minutes to complete the simulation of 1 period. In contrast, the method based on the combined Lawson-LSRK scheme only needs 4 minutes, which is 42.5 times faster than that method based on the LSRK scheme. Likewise, the DGTD- \mathbb{P}_2 and DGTD- \mathbb{P}_3 methods based on the combined Lawson-LSRK scheme yields 24.2 and 23.8 CPU speedups, respectively. Although the efficiency of the DGTD- \mathbb{P}_k methods improved significantly when using the proposed time scheme, the peak memory usage is moderately increased, 6.15%, 5.5% and 5% for the \mathbb{P}_1 to \mathbb{P}_3 methods respectively.



(a)



(b)

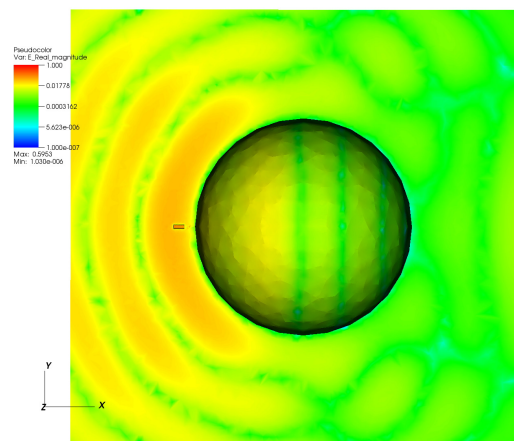


(c)

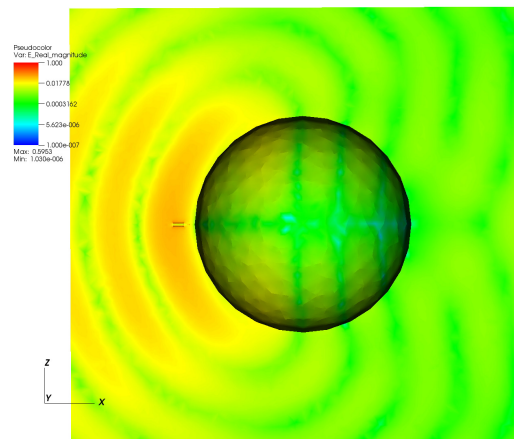
Fig. 10: Composite structure with a localized radiation source: time evolution of the electric field at a given point obtained by the DGTD- \mathbb{P}_1 method based on the combined Lawson-LSRK scheme versus fully explicit LSRK scheme.

VII. CONCLUSION

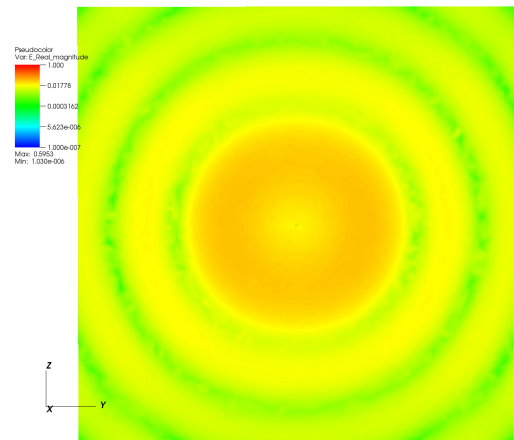
In this paper, a new family of exponential time integration methods for solving 3D time-domain Maxwell's equations discretized by a high order DG scheme formulated on locally refined unstructured meshes, has been proposed. These methods have excellent or even unconditional stability for the locally refined part of the mesh, which gives rise to much larger time step size than that of existing explicit time stepping



(a)



(b)



(c)

Fig. 11: Composite structure with a localized radiation source: contour lines of the amplitude of the electric field in the planes (a)XOY (b)XOZ (c)YOZ obtained by the DGTD- \mathbb{P}_3 method based on the combined Lawson-LSRK scheme.

schemes. This greatly decreases the time iterations for a given physical simulation time. Moreover, time integration can rely on a variety of explicit time stepping schemes and hence lead to efficient and accurate global explicit time integration methods. Additionally, the global explicitness of the obtained

TABLE VI: Composite structure with a localized radiation source: performance figures of the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme versus the fully explicit LSRK scheme.

\mathbb{P}_k	$\frac{\Delta t_{Lawson}}{\Delta t_{LSRK}}$	CPU (mn)			Peak Mem (MB)	
		LSRK	Lawson	Gain	LSRK	Lawson
\mathbb{P}_1	64	170	4	42.5	927	984
\mathbb{P}_2	54	460	19	24.2	3 322	3 505
\mathbb{P}_3	64	1 072	45	23.8	6 419	6 743

TABLE VII: Composite structure with a localized radiation source: characteristics of matrices for DGTD- \mathbb{P}_k methods.

\mathbb{P}_k	Dim	Dim $_{C_{ff}}$	Nz $_{C_{ff}}$
\mathbb{P}_1	1 819 988	35 578	1 153 108
\mathbb{P}_2	4 563 270	90 462	6 450 202
\mathbb{P}_3	6 857 088	137 097	13 303 005

method and the data locality of DG formulation makes it easily parallelized. For instance, a low-storage Runge-Kutta scheme is employed here, and a combined Lawson-LSRK scheme is constructed. Additionally, several efficient techniques are presented to further improve the proposed exponential time integration methods. The accuracy and numerical convergence of the proposed exponential time integration methods have been verified through numerical experiments. These numerical experiments demonstrate that the DGTD- \mathbb{P}_k methods based on the combined Lawson-LSRK scheme outperforms the fully explicit DGTD- \mathbb{P}_k methods with the same accuracy and negligible memory usage increase. Therefore, the DGTD- \mathbb{P}_k methods based on the proposed high order exponential time integration methods are very promising for the numerical simulation of 3D transient multiscale electromagnetic problems.

REFERENCES

- [1] F. Vipiana, M. Francavilla, and G. Vecchi, "Efie modeling of high-definition multiscale structures," *IEEE Trans. Antennas Propag.*, vol. 58, no. 7, pp. 2362–2374, 2010.
- [2] J. Chen and Q. Liu, "Discontinuous galerkin time-domain methods for multiscale electromagnetic simulations: a review," *Proc. IEEE*, vol. 101, no. 2, pp. 242–254, 2013.
- [3] Z. Peng, K. Lim, and J. Lee, "Nonconformal domain decomposition methods for solving large multiscale electromagnetic scattering problems," *Proc. IEEE*, vol. 101, no. 2, pp. 298–319, 2013.
- [4] K. Yee, "Numerical solution of initial boundary value problems involving Maxwells equations in isotropic media," *IEEE Trans. Antennas Propag.*, vol. 14, no. 3, pp. 302–307, 1966.
- [5] A. Taove and S. Hagness, *Computational electrodynamics: the finite-difference time-domain method*, 2nd ed. Artech House: Norwood, 1995.
- [6] M. Okoniewski, E. Okoniewska, and M. Stuchly, "Three-dimensional subgridding algorithm for fdtd," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 422–429, 1997.
- [7] J. Nédélec, "Mixed finite elements in \mathbb{R}^3 ," *Numerische Mathematik*, vol. 35, no. 3, pp. 315–341, 1980.
- [8] J. Jin, *The finite element method in electromagnetics*, 2nd ed. New York: John Wiley & Sons, 2002.
- [9] L. Fezoui, S. Lanteri, S. Lohrengel, and S. Piperno, "Convergence and stability of a discontinuous Galerkin time-domain method for the 3d heterogeneous maxwell equations on unstructured meshes," *ESAIM: Math. Model. and Numer. Anal.*, vol. 39, no. 6, pp. 1149–1176, 2005.
- [10] J. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. New York: Springer Science & Business Media, 2007.
- [11] S. Dosopoulos and J. Lee, "Interconnect and lumped elements modeling in interior penalty discontinuous galerkin time-domain methods," *J. Comput. Phys.*, vol. 22, pp. 8521–8536, 2010.
- [12] S. Piperno, "Symplectic local time-stepping in non-dissipative dgtd methods applied to wave propagation problems," *ESAIM: Math. Model. and Numer. Anal.*, vol. 40, no. 05, pp. 815–841, 2006.
- [13] E. Montseny, S. Pernet, X. Ferrieres, and G. Cohen, "Explicit local time-stepping methods for maxwell's equations," *J. Comput. Phys.*, vol. 227, no. 14, pp. 6795–6820, 2008.
- [14] M. Grote and T. Mitkova, "Explicit local time-stepping methods for Maxwell's equations," *J. Comput. Appl. Math.*, vol. 234, no. 12, pp. 3283–3302, 2010.
- [15] J. Diaz and M. Grote, "Multi-level explicit local time-stepping methods for second-order wave equations," *Comput. Meth. Appl. Mech. Engng.*, vol. 291, p. 240265, Jul. 2015.
- [16] A. Catella, V. Dolean, and S. Lanteri, "An unconditionally stable discontinuous galerkin method for solving the 2-D time-domain Maxwell equations on unstructured triangular meshes," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1250–1253, 2008.
- [17] J. Verwer and M. Botchev, "Unconditionally stable integration of Maxwells equations," *IEEE Trans. Magn.*, vol. 431, no. 3, pp. 300–317, 2009.
- [18] S. Descombes, S. Lanteri, and L. Moya, "Locally implicit time integration strategies in a discontinuous Galerkin method for Maxwells equations," *J. Sci. Comput.*, vol. 56, no. 1, pp. 190–218, 2013.
- [19] J. Verwer, "Component splitting for semi-discrete Maxwell equations," *J. Sci. Comput.*, vol. 11, p. 427445, Jun. 2011.
- [20] M. Hochbruck and A. Sturm, "Error analysis of a second-order locally implicit method for linear maxwell's equations," *SIAM Journal on Numerical Analysis*, vol. 54, no. 5, pp. 3167–3191, 2016.
- [21] A. Demirel, J. Niegemann, K. Busch, and M. Hochbruck, "Efficient multiple time-stepping algorithms of higher order," *J. Comput. Phys.*, vol. 285, p. 133148, Mar. 2015.
- [22] M. Botchev, "Krylov subspace exponential time domain solution of maxwells equations in photonic crystal modeling," *J. Comput. Appl. Math.*, vol. 293, p. 2034, Feb. 2016.
- [23] M. Hochbruck, C. Lubich, and H. Selhofer, "Exponential integrators for large systems of differential equations," *SIAM J. Sci. Comput.*, vol. 19, no. 5, pp. 1552–1574, 1988.
- [24] E. Haier, G. Wanner, and S. Norsett, *Solving ordinary differential equations, I: nonstiff problems*. Berlin Heidelberg: Springer, 1993.
- [25] M. Grote and T. Mitkova, "High-order explicit local time-stepping methods for damped wave equations," *J. Comput. Appl. Math.*, vol. 239, pp. 270–289, Feb. 2013.
- [26] J. Lawson, "Generalized Runge-Kutta processes for stable systems with large Lipschitz constants," *SIAM J. Numer. Anal.*, vol. 4, no. 3, pp. 372–380, 1967.
- [27] B. Minchev and W. Wright, "A review of exponential integrators for first order semi-linear problems," Norwegian University of Science and Technology, Tech. Rep. 2/2005, 2005.
- [28] C. Besse, G. Dujardin, and I. Lacroix-Violet, "High order exponential integrators for nonlinear schrödinger equations with application to rotating bose-einstein condensates," arXiv preprint, Tech. Rep. arXiv:1507.00550, Jun. 2015.
- [29] J. Williamson, "Low-storage Runge-Kutta schemes," *J. Comput. Phys.*, vol. 35, no. 1, pp. 48–5, 1980.
- [30] J. Nédélec, "A new family of mixed finite elements in \mathbb{R}^3 ," *Numerische Mathematik*, vol. 50, no. 1, pp. 57–81, 1986.
- [31] D. Sun, J. Lee, and Z. Cendes, "Construction of nearly orthogonal Nedelec bases for rapid convergence with multilevel preconditioned solvers," *SIAM J. Sci. Comput.*, vol. 23, no. 4, pp. 1053–1076, 2001.
- [32] C. Moler and C. Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Rev.*, vol. 45, no. 1, pp. 3–49, 2003.
- [33] M. Hochbruck and A. Ostermann, "Exponential integrators," *Acta Numerica*, vol. 19, pp. 209–286, May 2010.
- [34] M. Liou, "A novel method of evaluating transient response," *Proceedings of the IEEE*, vol. 54, no. 1, pp. 20–23, 1966.
- [35] R. Ward, "Numerical computation of the matrix exponential with accuracy estimate," *SIAM J. Numer. Anal.*, vol. 14, no. 4, pp. 600–610, 1977.
- [36] M. Hochbruck and C. Lubich, "On Krylov subspace approximations to the matrix exponential operator," *SIAM J. Numer. Anal.*, vol. 34, no. 5, pp. 1911–1925, 1997.
- [37] L. Bergamaschi, M. Caliarì, and M. Vianello, "Efficient approximation of the exponential operator for discrete 2D advection-diffusion problems," *Numer. Lin. Alg. Appl.*, vol. 10, no. 3, pp. 271–289, 2003.

- [38] V. Lebedev, "Explicit difference schemes for solving stiff systems of ODEs and PDEs with complex spectrum," *Russian Journal of Numerical Analysis and Mathematical Modelling*, vol. 13, no. 2, pp. 107–116, 1998.
- [39] R. Sidje, "Expokit: a software package for computing matrix exponentials," *ACM Trans. Math. Softw.*, vol. 24, no. 1, pp. 130–156, 1998.