Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0079

Vol. **21**, No. 4, pp. 1039-1064
April 2017

Simulation of Maxwell's Equations on GPU Using a High-Order Error-Minimized Scheme

Tony W. H. Sheu[1,2,3,*], S. Z. Wang[1], J. H. Li[1] and Matthew R. Smith[4]

[1] *Department of Engineering Science and Ocean Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan.*
[2] *Center for Advanced Study in Theoretical Sciences, National Taiwan University, Taipei, Taiwan.*
[3] *Institute of Applied Mathematical Sciences, National Taiwan University, Taipei, Taiwan.*
[4] *Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan.*

Communicated by Weng Cho Chew

Received 26 May 2016; Accepted (in revised version) 8 August 2016

Abstract. In this study an explicit Finite Difference Method (FDM) based scheme is developed to solve the Maxwell's equations in time domain for a lossless medium. This manuscript focuses on two unique aspects – the three dimensional time-accurate discretization of the hyperbolic system of Maxwell equations in three-point non-staggered grid stencil and it's application to parallel computing through the use of Graphics Processing Units (GPU). The proposed temporal scheme is symplectic, thus permitting conservation of all Hamiltonians in the Maxwell equation. Moreover, to enable accurate predictions over large time frames, a phase velocity preserving scheme is developed for treatment of the spatial derivative terms. As a result, the chosen time increment and grid spacing can be optimally coupled. An additional theoretical investigation into this pairing is also shown. Finally, the application of the proposed scheme to parallel computing using one Nvidia K20 Tesla GPU card is demonstrated. For the benchmarks performed, the parallel speedup when compared to a single core of an Intel i7-4820K CPU is approximately 190x.

AMS subject classifications: 35Q61, 65M06, 65Y05

Key words: Hamiltonian systems, symplecticity, GPU, 3-D Maxwell's equations, dispersion relation equation.

[*]Corresponding author. *Email addresses:* `twhsheu@ntu.edu.tw` (T. W. H. Sheu), `r01221024@ntu.edu.tw` (S. Z. Wang), `jiahan@ntu.edu.tw` (J. H. Li), `msmith@mail.ncku.edu.tw` (M. R. Smith)

1 Introduction

Gauss's law is known to serve as the constraint equation on the Maxwell's equations consisting of the Faraday's and Ampère's laws. The two divergence-free equations in Gauss's law are not always discretely satisfied when solving the electromagnetic (EM) wave solutions solely from the Faraday's and Ampère's equations. The computed nonzero-divergence errors in magnetic and electric flux densities can introduce undesired instability into the calculation process. Circumvention of this type of unphysical oscillations is therefore crucial in a successful simulation of Maxwell's equation [1]. The two constraint equations in Gauss's law can be numerically satisfied at all times when solving Maxwell's equations using Yee's staggered grid system [2]. For efficiently calculating EM wave solutions in parallel, in this study an explicit scheme capable of rendering a set of divergence-free electric and magnetic solutions is adopted using non-staggered (or co-located) grids.

While approximating the spatial and temporal derivative terms in Maxwell's equations, the key measure of the prediction accuracy is the dispersion error introduced into the solution [3]. In the worst cases, dispersion error tends to hinder simulation of problems involving narrow pulses and large time spans [4]. Therefore, any attempts to numerically solve the Maxwell equations should endeavor to simultaneously reduce dispersion and dissipation errors – this is particularly important for approximation of the first-order spatial derivative terms in the EM wave equations. Owing to this reason, extensive effort has been put toward the development of higher order FDTD schemes. One can refer to the detailed overview of higher order time-domain methods in [5]. An alternative to reduce numerical dispersion error, which constitutes a major source of error in the FDTD method, is to design schemes based on optimization criteria, other than to maximize the order of accuracy. Minimization of dispersion error can be achieved through the angle-optimized FDTD algorithm [6] and the parameter-dependent FDTD scheme [3]. Over the past two decades, there have been several FDTD schemes developed with the objective of satisfying the dispersion-relation equation to reduce dispersion errors [7]. Shlager and Schneider compared dispersion properties of several low-dispersion FDTD algorithms [8].

When attempting long-term simulations – meaning simulations which require a very large number of discrete time steps – of the electromagnetic wave equations, the solution quality may be deteriorated substantially due to the accumulation of numerical error resulting from the application of a non-symplectic time-stepping scheme. These accumulated errors, while initially quite small, may build to values large enough to physically alter the properties of the solution. In order to avoid this problem, it is important to numerically preserve the symplectic property existing in Maxwell's equations when treating the time derivative terms in Faraday's and Ampère's equations. When simulating Maxwell's equations, the quality of the solution predicted by the Finite Difference Time Domain (FDTD) method can be deteriorated as well by the introduced anisotropy error. Dispersion and anisotropy errors are both accumulative with time and can seriously contaminate the true propagation characteristics. One of the goals of this study is to reduce the magnitude of the above mentioned error types through different underlying theories applied to a three dimensional domain.

Performing large-scale electromagnetic wave simulations requires significant computational expense. The application of single core architectures is often not feasible or practical when using error-minimizing or optimized models alone. When examining ways to increase the computational efficiency, the hardware architecture chosen to carry out FDTD simulation of equations often needs to be taken into account. The recent resurgence in popularity of vector-type computational architectures, such as Advanced Vector eXtensions (AVX), the 512-bit wide registers in the Intel Xeon Phi coprocessor and various Graphics Processing Units (GPU's) means that careful consideration to SIMD-type parallelization must be considered when performing parallel computation on such devices. With the invent of such tools, in addition to the development of CPU sockets supporting increasing number of CPU cores, we are able to dramatically increase the compute density of computer systems, allowing us access to the so-called "green" computing solutions which require lower power than conventional HPC systems at a fraction of the capital expense.

With the recent increase in the computational capacity of GPU devices, multiple authors have been drawn to the numerical simulation of Maxwell's equations on such devices. These implementations have provided evidence which demonstrates the reduction of computing time [9–15] to these equations. While the reported speedups vary from author to author, all agree that the application of GPU computing is a worthwhile undertaking. Owing to this and the continued increase in GPU computational power, this study will also focus on the application of our proposed explicit FDTD scheme capable of yielding dispersive error-minimization to parallel computation using a single GPU device.

This paper is organized as follows. In Section 2, some of the distinguished physical and fruitful mathematical features in the ideal (or lossless) Maxwell's equations related to the development of our proposed scheme and code verification are presented together with two indispensable divergence-free constraint equations. In Section 3, the first-order spatial derivative terms in Faraday's and Ampère's equations are discretized using a non-staggered grid rather than the conventional staggered grid approach proposed by Yee [2]. In this paper, the difference between the exact and numerical phase velocities is minimized to achieve a higher dispersive accuracy. Maxwell's equations belong to the class of integrable equations [16] which may be solved in a symplectic manner. Hence, a symplectic structure-preserving time integrator shall be applied to conserve numerical symplecticity. Here, we propose the application of the Symplectic Partitioned Runge-Kutta (SPRK) scheme. The associated derivation of the corresponding stability condition for the proposed explicit scheme is also given in Section 3. In Section 4, the Nvidia K-20 GPU and the CPU/GPU hybrid architecture are briefly reviewed. Following this, the GPU parallelization of the method described in Section 3 is shown with a special focus on the proper arrangement of the global and shared memories which make up the GPU architecture. In Section 5, the proposed second-order accurate temporal scheme and the fourth-order accurate spatial scheme are verified and validated through the application of multiple benchmark problems. In addition, the performance of the current GPU implementation is detailed and compared with its CPU counterpart. Finally, we will draw conclusions in Section 6 based on the solutions computed in parallel on non-staggered grids using a single Nvidia K-20 GPU card.

2 Maxwell's equations

Maxwell's equations in lossless media are represented below in terms of the dependent variables $\underline{E}=\left(E_x,E_y,E_z\right)^T$ and $\underline{H}=\left(H_x,H_y,H_z\right)^T$

$$\frac{\partial \underline{H}}{\partial t}=-\frac{1}{\mu}\,\nabla\times\underline{E}, \tag{2.1}$$

$$\frac{\partial \underline{E}}{\partial t}=\frac{1}{\epsilon}\,\nabla\times\underline{H}. \tag{2.2}$$

The above equations are coupled with Gauss's law which consists of two divergence-free equations $\nabla\cdot\underline{B}=0$ and $\nabla\cdot\underline{D}=0$. These two divergence-free constraint equations can be derived directly from Faraday's law and Ampère's law for a linear, isotropic and lossless material provided that the electric current density and electric charge density terms are neglected. Within the context of differential equations, Gauss's law is unconditionally satisfied in case the vectors $\underline{B}$ and $\underline{D}$ are both initially divergence-free [17]. The differential set of Maxwell's equations becomes over-determined in this case. Two divergence-free equations need to be neglected so that the number of unknowns is equal to the number of field equations. Eqs. (2.1)-(2.2) are derived under the conditions of $\underline{D}=\epsilon\underline{E}$ and $\underline{B}=\mu\underline{H}$, where $\underline{D}$ denotes the electric flux density and $\underline{E}$ is the electric field density. In the proportional constants, ϵ is known as the electric permittivity and μ is the magnetic permeability. The values of ϵ and μ determine the light speed c $(\equiv(\varepsilon\,\mu)^{-1/2})$.

The first Hamiltonian in the bi-Hamiltonian differential system of equations (2.1)-(2.2) has association with the helicity Hamiltonian H_1 given below [18]

$$H_1=\frac{1}{2}\int\frac{1}{\epsilon}\underline{H}\cdot\nabla\times\underline{H}+\frac{1}{\mu}\underline{E}\cdot\nabla\times\underline{E}\;d\Omega. \tag{2.3}$$

The second quadratic Hamiltonian (or energy density) is expressed as follows [19]

$$H_2=\frac{1}{2}\int\mu\underline{H}\cdot\underline{H}+\epsilon\underline{E}\cdot\underline{E}\;d\Omega. \tag{2.4}$$

The two Hamiltonians given above will be used in this study to indirectly justify the proposed numerical scheme. Numerical errors computed solely from Faraday's and Ampère's equations may cause the resulting solutions computed from the magnetic and electric equations to cease being divergence free. To overcome the computational difficulty owing to the omission of Gauss's law, two gradient terms for the scalar variables Φ_1

and Φ_2 are introduced into Eqs. (2.1) and (2.2). The resulting modified Maxwell's equations to be applied are expressed as $\frac{\partial \underline{E}}{\partial t}-\frac{1}{\epsilon}\nabla\times\underline{H}+\nabla\Phi_1=0$ and $\frac{\partial \underline{H}}{\partial t}+\frac{1}{\mu}\nabla\times\underline{E}+\nabla\Phi_2=0$. The equations responsible for the two introduced correction potentials are shown in [20]. It is noted that Φ_1 and Φ_2 in divergence cleaning technique are governed by $\nabla^2\Phi_1=-\frac{\partial}{\partial t}(\nabla\cdot\underline{E})$ and $\nabla^2\Phi_2=-\frac{\partial}{\partial t}(\nabla\cdot\underline{H})$, respectively. In our calculation of the Φ_1 and Φ_2 values, we computed $-\frac{\partial}{\partial t}(\nabla\cdot\underline{E})$ and $-\frac{\partial}{\partial t}(\nabla\cdot\underline{H})$ based on the previously computed values of $\underline{E}$ and $\underline{H}$ at $(n-1)\Delta t$ and $(n-2)\Delta t$ so that the scheme can be retained to be explicit, which is essential in the current parallel computation in GPU described in Section 4. As a result, the introduced correction potentials Φ_1 and Φ_2 are not functions of $\underline{E}$ and $\underline{H}$, thus facilitating the derivation of the weighting coefficients in Section 3.2.

While simulating EM wave propagation in an open domain, we need to truncate the potentially infinitely large domain to make simulation feasible due to limitation in computational resources. To practically resolve this issue, either a radiation boundary condition on the truncated boundary or an artificial boundary of finite length shall be added to the chosen physical domain so as to absorb all possible outgoing waves. In this study the computational domain is divided into two parts – in free space, or in a vacuum, the modified Maxwell's equations are employed. In the absorbing layer, the following equations as defined in Berenger's Perfectly Matched Layer (PML) approach are employed [21,22]:

$$\begin{aligned}
\frac{\partial E_x}{\partial t}&=\frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y}-\frac{\partial H_y}{\partial z}-\frac{\sigma_y}{\varepsilon_0}D_x+\frac{\sigma_x}{\varepsilon_0}D_x\right)-\frac{\sigma_z}{\varepsilon_0}E_x,\\
\frac{\partial E_y}{\partial t}&=\frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z}-\frac{\partial H_z}{\partial x}-\frac{\sigma_z}{\varepsilon_0}D_y+\frac{\sigma_y}{\varepsilon_0}D_y\right)-\frac{\sigma_x}{\varepsilon_0}E_y,\\
\frac{\partial E_z}{\partial t}&=\frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x}-\frac{\partial H_x}{\partial y}-\frac{\sigma_x}{\varepsilon_0}D_z+\frac{\sigma_z}{\varepsilon_0}D_z\right)-\frac{\sigma_y}{\varepsilon_0}E_z,\\
\frac{\partial H_x}{\partial t}&=\frac{1}{\mu}\left(\frac{\partial E_y}{\partial z}-\frac{\partial E_z}{\partial y}-\frac{\sigma_y}{\varepsilon_0}B_x+\frac{\sigma_x}{\varepsilon_0}B_x\right)-\frac{\sigma_z}{\varepsilon_0}H_x,\\
\frac{\partial H_y}{\partial t}&=\frac{1}{\mu}\left(\frac{\partial E_z}{\partial x}-\frac{\partial E_x}{\partial z}-\frac{\sigma_z}{\varepsilon_0}B_y+\frac{\sigma_y}{\varepsilon_0}B_y\right)-\frac{\sigma_x}{\varepsilon_0}H_y,\\
\frac{\partial H_z}{\partial t}&=\frac{1}{\mu}\left(\frac{\partial E_x}{\partial y}-\frac{\partial E_y}{\partial x}-\frac{\sigma_x}{\varepsilon_0}B_z+\frac{\sigma_z}{\varepsilon_0}B_z\right)-\frac{\sigma_y}{\varepsilon_0}H_z.
\end{aligned}\tag{2.5}$$

In the above equations, $\sigma_i(i=x,y,z)$ denote the electric conductivity along the x,y,z directions. The notation ε_0 represents the electric permittivity in a vacuum. For additional information regarding the PML approach, the expressions such as the constitutive relations are described in additional detail in [21].

3 Numerical method

Unlike most EM wave numerical solution approaches, which have conventionally employed staggered computational grids, this study focuses on the solution to the Maxwell's equations employing non-staggered grids. This makes the implementation of the solver on various computational architectures – in serial and in parallel – a relatively simpler prospect. To avoid the so-called pattern of even-odd (or checkerboard) oscillatory solutions often observed in non-staggered (or collocated) grids, the center-type compact difference scheme in [20] has been successfully employed. The reason this approach successfully avoids the production of oscillatory solutions is because all field solutions at the reference point of the employed central scheme have their role to play. It is worthy to note that the tridiagonal system of equations involved in the compact difference scheme [20] can be effectively solved in parallel on a GPU through the methods of cyclic reduction (CR) [23], recursive doubling (RD) [24] and the hybrid CR-RD method [25].

3.1 Explicit symplectic partitioned Runge-Kutta temporal scheme

Maxwell's equations are mathematically separable. This makes a symplectic approach possible when attempting the numerical solution of said equations in separated form. The explicit symplectic partitioned Runge-Kutta time-stepping scheme is applied in this study to integrate Faraday's and Ampère's equations [26]. Calculation of $\underline{E}^{n+1}$ and $\underline{H}^{n+1}$ solutions at $t=(n+1)\triangle t$ from the stably computed solutions at time nΔt is split into the following steps by using the second-order accurate explicit partitioned Runge-Kutta scheme as presented in [27]

$$\underline{H}^{n+\frac{1}{2}}=\underline{H}^{n}-\frac{dt}{2\mu}\nabla\times\underline{E}^{n}, \tag{3.1}$$

$$\underline{E}^{n+1}=\underline{E}^{n}+\frac{dt}{\epsilon}\nabla\times\underline{H}^{n+\frac{1}{2}}, \tag{3.2}$$

$$\underline{H}^{n+1}=\underline{H}^{n+\frac{1}{2}}-\frac{dt}{2\mu}\nabla\times\underline{E}^{n+1}. \tag{3.3}$$

3.2 Numerical treatment of spatial derivative terms

In addition to the necessity of developing a symplecticity-preserving scheme with the goal of preserving accuracy in time, this study also aims to reduce the dispersion error in space. To this end, the difference between the numerical and exact phase velocities shall be minimized in wavenumber space [28]. To achieve this goal, the spatial derivative terms shown in (3.1)-(3.3) are approximated by the methods of modified equation analysis given in Section 3.2.1, dispersion analysis given in Section 3.2.2, and the grid-anisotropy analysis presented in Section 3.2.3.

At $t=n\Delta t$, we can get $\underline{H}^{n}=\underline{H}^{n-\frac{1}{2}}-\frac{dt}{2\mu}\nabla\times\underline{E}^{n}$ and, then, $\underline{H}^{n+\frac{1}{2}}=\underline{H}^{n-\frac{1}{2}}-\frac{dt}{2\mu}\nabla\times\underline{E}^{n}$ according to the following equations

$$E_z^{n+\frac{1}{2}}=E_z^{n-\frac{1}{2}}+\frac{dt}{\epsilon}\left(\frac{\partial H_y^n}{\partial x}-\frac{\partial H_x^n}{\partial y}\right), \tag{3.4}$$

$$E_x^{n+\frac{1}{2}} = E_x^{n-\frac{1}{2}} + \frac{dt}{\epsilon}\left(\frac{\partial H_z^n}{\partial y} - \frac{\partial H_y^n}{\partial z}\right), \tag{3.5}$$

$$E_y^{n+\frac{1}{2}} = E_y^{n-\frac{1}{2}} + \frac{dt}{\epsilon}\left(\frac{\partial H_x^n}{\partial z} - \frac{\partial H_z^n}{\partial x}\right). \tag{3.6}$$

To get a higher accuracy at a reasonable computational cost, either a compact or a combined compact difference scheme can be applied to effectively reduce numerical errors at a small wavelength [29]. In this study, we aim to not only reduce the phase error but also the amplitude error [30]. We mean to achieve this goal of reducing dispersive error by minimizing errors in the numerical dispersion relation equation [31, 32]. Therefore, we must first derive the explicit form of the numerical dispersion relation equation in order to reach this goal.

The first-order derivative terms at any given interior node (i,j,k) are approximated on non-staggered grids, by the scheme given below:

$$\begin{aligned}\frac{\partial H_y}{\partial x}\Big|_{i,j,k}^n = \frac{1}{h}\Big[& a_1\left(H_y|_{i+3,j,k}^n - H_y|_{i-3,j,k}^n\right) + a_2\left(H_y|_{i+2,j,k}^n - H_y|_{i-2,j,k}^n\right) \\ & + a_3\left(H_y|_{i+1,j,k}^n - H_y|_{i-1,j,k}^n\right)\Big],\end{aligned} \tag{3.7}$$

$$\begin{aligned}\frac{\partial H_x}{\partial y}\Big|_{i,j,k}^n = \frac{1}{h}\Big[& a_1\left(H_x|_{i,j+3,k}^n - H_x\Big|_{i,j-3,k}^n\right) + a_2\left(H_x|_{i,j+2,k}^n - H_y|_{i,j-2,k}^n\right) \\ & + a_3\left(H_x|_{i,j+1,k}^n - H_x|_{i,j-1,k}^n\right)\Big].\end{aligned} \tag{3.8}$$

After substituting (3.7), (3.8) into (3.4) and then expanding the resulting terms using a Taylor series with respect to E_z, the following equation at an interior point (i,j,k) can be derived

$$\begin{aligned}& \frac{\partial E_z}{\partial t}\Big|_{i,j,k}^n + \frac{dt^2}{24}\frac{\partial^3 E_z}{\partial t^3}\Big|_{i,j,k}^n + \frac{dt^4}{1920}\frac{\partial^5 E_z}{\partial t^5}\Big|_{i,j,k}^n + \frac{dt^6}{322560}\frac{\partial^7 E_z}{\partial t^7}\Big|_{i,j,k}^n + \cdots \\ = & \frac{1}{\epsilon}\Big\{\Big[\left(6a_1 + 4a_2 + 2a_3\right)\frac{\partial H_y}{\partial x}\Big|_{i,j,k}^n + \left(9a_1 + \frac{8}{3}a_2 + \frac{1}{3}a_3\right)dx^2\frac{\partial^3 H_y}{\partial x^3}\Big|_{i,j,k}^n \\ & + \left(\frac{81}{20}a_1 + \frac{8}{15}a_2 + \frac{1}{60}a_3\right)dx^4\frac{\partial^5 H_y}{\partial x^5}\Big|_{i,j,k}^n + \left(\frac{243}{280}a_1 + \frac{16}{315}a_2 + \frac{1}{2520}a_3\right)dx^6\frac{\partial^7 H_y}{\partial x^7}\Big|_{i,j,k}^n \\ & + \cdots\Big] - \Big[\left(6a_1 + 4a_2 + 2a_3\right)\frac{\partial H_y}{\partial x}\Big|_{i,j,k}^n + \left(9a_1 + \frac{8}{3}a_2 + \frac{1}{3}a_3\right)dx^2\frac{\partial^3 H_y}{\partial x^3}\Big|_{i,j,k}^n \\ & + \left(\frac{81}{20}a_1 + \frac{8}{15}a_2 + \frac{1}{60}a_3\right)dx^4\frac{\partial^5 H_y}{\partial x^5}\Big|_{i,j,k}^n + \left(\frac{243}{280}a_1 + \frac{16}{315}a_2 + \frac{1}{2520}a_3\right)dx^6\frac{\partial^7 H_y}{\partial x^7}\Big|_{i,j,k}^n + \cdots\Big]\Big\}.\end{aligned} \tag{3.9}$$

The three introduced weighting coefficients a_1, a_2 and a_3 are determined by performing the modified equation analysis and the dispersion analysis described herein. All time derivative terms $\frac{\partial^3 E_z}{\partial t^3}$, $\frac{\partial^5 E_z}{\partial t^5}$, $\frac{\partial^7 E_z}{\partial t^7}\cdots$ shown in (3.9) are replaced first by their equivalent spatial derivative terms through Ampère's equations to get the corresponding equations for $\frac{\partial^i E_j}{\partial t^i}$ ($i=3$ and 5, $j=x,y,z$). After replacing the high-order temporal derivative terms $\frac{\partial^3 E_z}{\partial t^3}$ and $\frac{\partial^5 E_z}{\partial t^5}$ with the corresponding spatial derivative terms, an equation equivalent to (3.9) is derived. By comparing the resulting equation with the equation $\frac{\partial E_z}{\partial t}=\frac{1}{\epsilon}\big(\frac{\partial H_y}{\partial x}-\frac{\partial H_x}{\partial y}\big)$, the equations for a_1, a_2 and a_3 are derived as

$$3a_1+2a_2+a_3=\frac{1}{2} \tag{3.10}$$

and

$$9a_1+\frac{8}{3}a_2+\frac{1}{3}a_3-\frac{Cr^2}{12}(3a_1+2a_2+a_3)=0, \tag{3.11}$$

where $Cr=\frac{c\Delta t}{h}$ denotes the Courant number and h denotes the grid spacing.

Determination of the three undetermined coefficients shown above requires the derivation of the third algebraic equation. Substitution of the plane wave solution

$$\underline{E}=\underline{E}_0\exp\big(I\left(k_x i\Delta x+k_y j\Delta y+k_z k\Delta z-\omega n\Delta t\right)\big),$$

where $I=\sqrt{-1}$, into the equation given by

$$\left.\frac{\partial \underline{E}}{\partial t}\right|^n_{i,j,k}=\frac{\underline{E}^{n+\frac{1}{2}}|_{i,j,k}-\underline{E}^{n-\frac{1}{2}}|_{i,j,k}}{\Delta t}$$

and the equations given by

$$\begin{aligned}
\left.\frac{\partial \underline{E}}{\partial x}\right|^n_{i,j,k}&=\frac{1}{h}\left[a_1\left(\underline{E}|^n_{i+3,j,k}-\underline{E}|^n_{i-3,j,k}\right)+a_2\left(\underline{E}|^n_{i+2,j,k}-\underline{E}|^n_{i-2,j,k}\right)+a_3\left(\underline{E}|^n_{i+1,j,k}-\underline{E}|^n_{i-1,j,k}\right)\right],\\
\left.\frac{\partial \underline{E}}{\partial y}\right|^n_{i,j,k}&=\frac{1}{h}\left[a_1\left(\underline{E}|^n_{i,j+3,k}-\underline{E}|^n_{i,j-3,k}\right)+a_2\left(\underline{E}|^n_{i,j+2,k}-\underline{E}|^n_{i,j-2,k}\right)+a_3\left(\underline{E}|^n_{i,j+1,k}-\underline{E}|^n_{i,j-1,k}\right)\right],\\
\left.\frac{\partial \underline{E}}{\partial z}\right|^n_{i,j,k}&=\frac{1}{h}\left[a_1\left(\underline{E}|^n_{i,j,k+3}-\underline{E}|^n_{i,j,k-3}\right)+a_2\left(\underline{E}|^n_{i,j,k+2}-\underline{E}|^n_{i,j,k-2}\right)+a_3\left(\underline{E}|^n_{i,j,k+1}-\underline{E}|^n_{i,j,k-1}\right)\right]
\end{aligned}$$

allows us to obtain $\frac{\partial \underline{E}}{\partial t}$, $\frac{\partial \underline{E}}{\partial x}$, $\frac{\partial \underline{E}}{\partial y}$, and$\frac{\partial \underline{E}}{\partial z}$ and then the equations for $\frac{\partial^2 \underline{E}}{\partial t^2}(=c^2(\frac{\nabla^2 \underline{E}}{\partial x^2}+\frac{\nabla^2 \underline{E}}{\partial y^2}+\frac{\nabla^2 \underline{E}}{\partial z^2}))$, $\frac{\partial^2 \underline{E}}{\partial x^2}$, $\frac{\partial^2 \underline{E}}{\partial y^2}$ and $\frac{\partial^2 \underline{E}}{\partial z^2}$. The numerical dispersion relation equation can be finally derived as

$$\begin{aligned}
\frac{1}{c^2}\frac{\omega^2}{4}\left(\frac{\sin(\omega\Delta t/2)}{\omega\Delta t}\right)^2=&k_x^2\left(3a_1\frac{\sin(3k_x\Delta x)}{3k_x\Delta x}+2a_2\frac{\sin(2k_x\Delta x)}{2k_x\Delta x}+a_3\frac{\sin(k_x\Delta x)}{k_x\Delta x}\right)^2\\
&+k_y^2\left(3a_1\frac{\sin(3k_y\Delta y)}{3k_y\Delta y}+2a_2\frac{\sin(2k_y\Delta y)}{2k_y\Delta y}+a_3\frac{\sin(k_y\Delta y)}{k_y\Delta y}\right)^2\\
&+k_z^2\left(3a_1\frac{\sin(3k_z\Delta z)}{3k_z\Delta z}+2a_2\frac{\sin(2k_z\Delta z)}{2k_z\Delta z}+a_3\frac{\sin(k_z\Delta z)}{k_z\Delta z}\right)^2.
\end{aligned} \tag{3.12}$$

The wavenumber vector is defined as $\underline{k}=\left(k_x,k_y,k_z\right)$. The exact dispersion relation equation can be similarly derived as $\left(\frac{\omega}{c}\right)^2=k_x^2+k_y^2+k_z^2$ by substituting the plane wave solution into the second-order wave equation $\frac{\partial^2\underline{E}}{\partial t^2}=c^2\nabla^2\underline{E}$.

To get accurate propagation characteristics while solving Maxwell's equations in the time domain we need to develop a scheme whose numerical phase velocity $\underline{v}_p$ $(\equiv\frac{\omega_{num}}{\underline{k}})$ matches well with its exact counterpart. Following this line of thought, the error function defined as $\left[|\frac{\omega_{num}}{\underline{k}}|-|\frac{\omega_{exact}}{\underline{k}}|\right]^2$ is minimized in a weak sense within the integral range given below

$$E_p=\int_{-m_p\pi}^{m_p\pi}\left[\frac{\omega_{num}}{\underline{|k|}}-\frac{\omega_{exact}}{\underline{|k|}}(\equiv c)\right]^2 W_p\;d(k_x\Delta x)\;d(k_y\Delta y)\;d(k_z\Delta z). \tag{3.13}$$

In the above, $k_x\Delta x$, $k_y\Delta y$ and $k_z\Delta z$ denote the scaled (or modified) wavenumbers along the x, y and z directions, respectively. Application of the above weighting function W_p enables us to integrate E_p analytically for the value of m_p in between 0 and $\frac{1}{2}$.

By enforcing the limiting condition given by $\frac{\partial E_p}{\partial a_3}=0$, the third algebraic equation for the coefficients a_1, a_2 and a_3 can be written as:

$$\begin{aligned}&-0.00946472\;a_1-0.00787899\;a_2+0.224744\;a_1^3+0.0948775\;a_2^3+0.367829\;a_2^2a_1\\&\quad+0.0166091\;a_3^3+0.107206\;a_3^2a_1+0.261056\;a_1^2a_3+0.156637\;a_2^2a_3-0.00453852\;a_3\\&\quad+0.492672\;a_1^2a_2+0.395351\;a_3a_2a_1+0.0875208\;a_3^2a_2=0.\end{aligned} \tag{3.14}$$

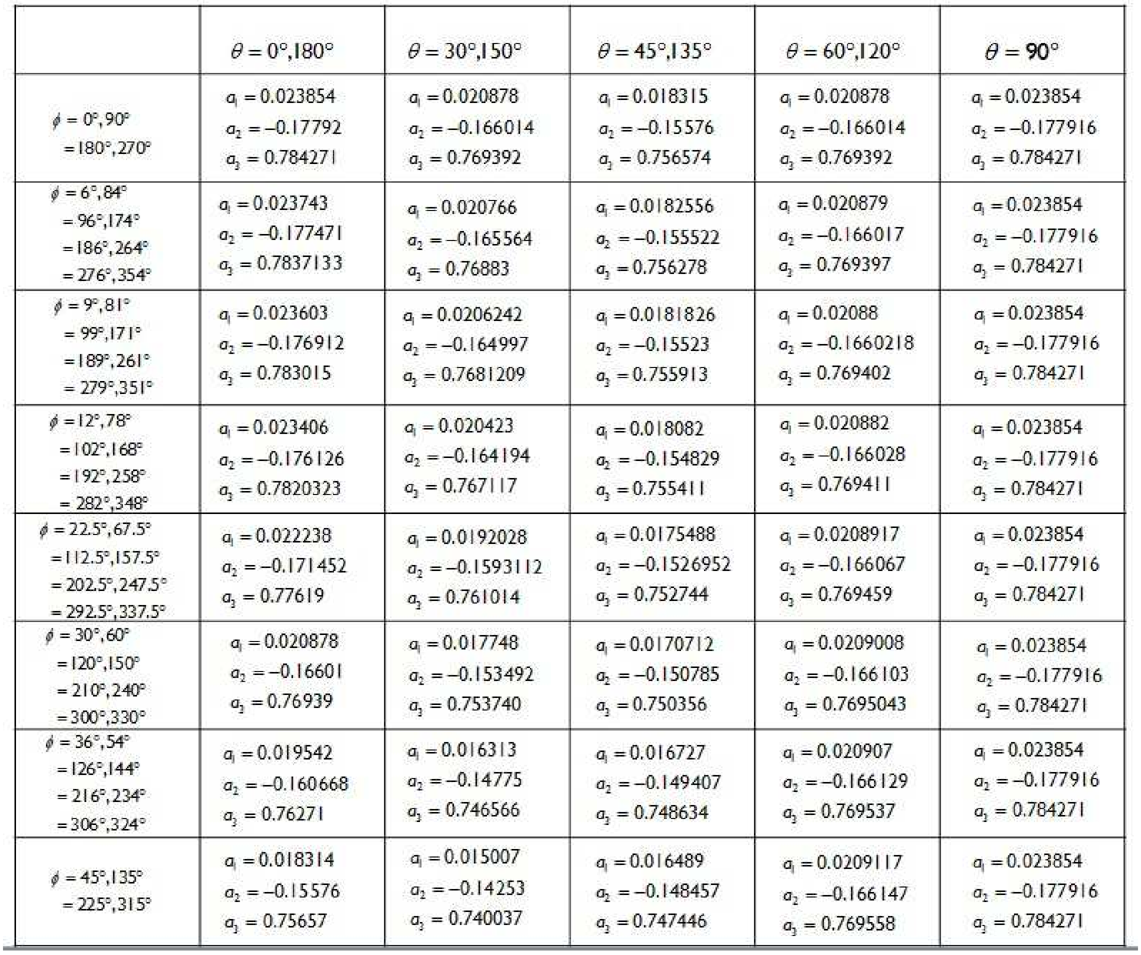
Eq. (3.14) derived through the minimization of the dispersive error will be used together with the other two algebraic equations (3.10)-(3.11) derived from the underlying modified equation analysis of second kind. From the simulation results tabulated in Table 1, $m_p=\frac{1}{2}$ is chosen and the three coefficients shown in (3.7)-(3.8) can be computed as those shown in Table 2.

Table 1: Comparison of the computed L_2-error norms at $t=10$ for the component E_z in a domain of 201^3 mesh points at different values of m in (3.11).

parameter m	L_2-**error norm of** E_z
1/2	2.2340E-04
3/7	5.5748E-04
2/5	3.5673E-04
1/3	6.1732E-04

Owing to the successful computation of the theoretically derived weighting coefficients a_1, a_2 and a_3 which are plotted with respect to the wavenumber angle $\phi\equiv\tan^{-1}\left(\frac{k_y}{k_x}\right)$ in Fig. 1, the proposed space centered error-optimized scheme has an order of spatial accuracy of four since the derived modified equation can be written as $\frac{\partial H_x}{\partial x}=\frac{\partial H_x}{\partial x}|_{exact}-0.018459h^4\frac{\partial^5 H_x}{\partial x^5}+\mathcal{O}(h^6)+\cdots$.

Table 2: The coefficients derived in the proposed three dimensional scheme. Note that $\sin(\frac{\pi}{2}-\phi)=\cos\phi$, $\cos(\frac{\pi}{2}-\phi)=\sin\phi$, $\sin(\frac{3\pi}{2}-\phi)=-\cos\phi$, $\cos(\frac{3\pi}{2}-\phi)=\sin\phi$, $\sin(\pi-\phi)=\sin\phi$, $\cos(\pi-\phi)=-\cos\phi$, $\sin(-\phi)=-\sin\phi$, $\cos(-\phi)=\cos\phi$.



	$\theta=0°,180°$	$\theta=30°,150°$	$\theta=45°,135°$	$\theta=60°,120°$	$\theta=90°$
$\phi=0°,90°$ $=180°,270°$	$a_1=0.023854$ $a_2=-0.17792$ $a_3=0.784271$	$a_1=0.020878$ $a_2=-0.166014$ $a_3=0.769392$	$a_1=0.018315$ $a_2=-0.15576$ $a_3=0.756574$	$a_1=0.020878$ $a_2=-0.166014$ $a_3=0.769392$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$
$\phi=6°,84°$ $=96°,174°$ $=186°,264°$ $=276°,354°$	$a_1=0.023743$ $a_2=-0.177471$ $a_3=0.7837133$	$a_1=0.020766$ $a_2=-0.165564$ $a_3=0.76883$	$a_1=0.0182556$ $a_2=-0.155522$ $a_3=0.756278$	$a_1=0.020879$ $a_2=-0.166017$ $a_3=0.769397$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$
$\phi=9°,81°$ $=99°,171°$ $=189°,261°$ $=279°,351°$	$a_1=0.023603$ $a_2=-0.176912$ $a_3=0.783015$	$a_1=0.0206242$ $a_2=-0.164997$ $a_3=0.7681209$	$a_1=0.0181826$ $a_2=-0.15523$ $a_3=0.755913$	$a_1=0.02088$ $a_2=-0.1660218$ $a_3=0.769402$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$
$\phi=12°,78°$ $=102°,168°$ $=192°,258°$ $=282°,348°$	$a_1=0.023406$ $a_2=-0.176126$ $a_3=0.7820323$	$a_1=0.020423$ $a_2=-0.164194$ $a_3=0.767117$	$a_1=0.018082$ $a_2=-0.154829$ $a_3=0.755411$	$a_1=0.020882$ $a_2=-0.166028$ $a_3=0.769411$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$
$\phi=22.5°,67.5°$ $=112.5°,157.5°$ $=202.5°,247.5°$ $=292.5°,337.5°$	$a_1=0.022238$ $a_2=-0.171452$ $a_3=0.77619$	$a_1=0.0192028$ $a_2=-0.1593112$ $a_3=0.761014$	$a_1=0.0175488$ $a_2=-0.1526952$ $a_3=0.752744$	$a_1=0.0208917$ $a_2=-0.166067$ $a_3=0.769459$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$
$\phi=30°,60°$ $=120°,150°$ $=210°,240°$ $=300°,330°$	$a_1=0.020878$ $a_2=-0.16601$ $a_3=0.76939$	$a_1=0.017748$ $a_2=-0.153492$ $a_3=0.753740$	$a_1=0.0170712$ $a_2=-0.150785$ $a_3=0.750356$	$a_1=0.0209008$ $a_2=-0.166103$ $a_3=0.7695043$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$
$\phi=36°,54°$ $=126°,144°$ $=216°,234°$ $=306°,324°$	$a_1=0.019542$ $a_2=-0.160668$ $a_3=0.76271$	$a_1=0.016313$ $a_2=-0.14775$ $a_3=0.746566$	$a_1=0.016727$ $a_2=-0.149407$ $a_3=0.748634$	$a_1=0.020907$ $a_2=-0.166129$ $a_3=0.769537$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$
$\phi=45°,135°$ $=225°,315°$	$a_1=0.018314$ $a_2=-0.15576$ $a_3=0.75657$	$a_1=0.015007$ $a_2=-0.14253$ $a_3=0.740037$	$a_1=0.016489$ $a_2=-0.148457$ $a_3=0.747446$	$a_1=0.0209117$ $a_2=-0.166147$ $a_3=0.769558$	$a_1=0.023854$ $a_2=-0.177916$ $a_3=0.784271$

The weighting coefficients at each grid point can then be calculated by interpolation of the data shown in Table 2. At each grid location we can compute the local Courant numbers Cr_x and Cr_y and their corresponding interpolated weighting coefficients as plotted in Fig. 2. It is worth addressing here that the derived nodally-satisfying DRE (Dispersion Relation Equation) scheme is the main contribution of the present study. One can follow the theoretical guideline to determine the time increment Δt according to the user's chosen grid spacing.

The stability condition of the proposed explicit scheme, which maintains precision in time and space through symplecticity and application of the dispersion relation equation respectively, is derived by considering the equivalent eigenvalue equations. The

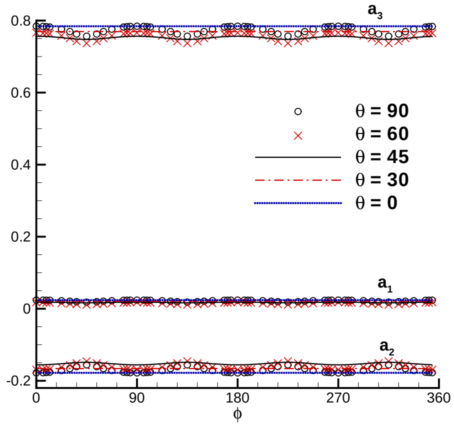


Figure 1: The theoretically derived coefficients a_1, a_2 and a_3 in (3.7) and (3.8) are plotted with respect to $\phi \equiv tan^{-1}(\frac{k_y}{k_x})$ at different zenith angles θ at $Cr = (Cr_x^2 + Cr_y^2 + Cr_z^2)^{1/2} = 0.2$.

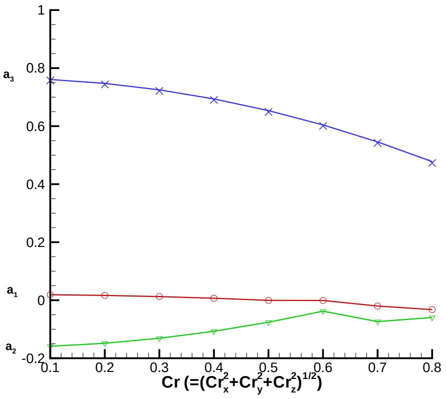


Figure 2: The values of a_1, a_2 and a_3 are plotted with respect to $Cr = (Cr_x^2 + Cr_y^2 + Cr_z^2)^{1/2}$ for the Courant number ranging from 0.1 to 0.8 using the proposed phase velocity optimized scheme at, for example, $\theta = \phi = \frac{\pi}{4}$.

proposed conditionally stable explicit scheme is subject to the constraint

$$\Delta t \leq \frac{1}{c} \left(\frac{\max(F_x^2)}{\Delta x^2} + \frac{\max(F_y^2)}{\Delta y^2} + \frac{\max(F_z^2)}{\Delta z^2} \right)^{-\frac{1}{2}},$$

where

$$F_x = a_1 \sin(3k_x \triangle x) + a_2 \sin(2k_x \triangle x) + a_3 \sin(k_x \triangle x),$$
$$F_y = a_1 \sin(3k_y \triangle y) + a_2 \sin(2k_y \triangle y) + a_3 \sin(k_y \triangle y),$$
$$F_z = a_1 \sin(3k_z \triangle z) + a_2 \sin(2k_z \triangle z) + a_3 \sin(k_z \triangle z).$$

By substituting the previously derived coefficients a_1, a_2 and a_3 into the above inequality equation, the stability condition for the current scheme developed for the calculation of three dimensional Maxwell's equations is $\Delta t \le 0.673844\frac{h}{c}$.

4 GPU parallelization of Maxwell's equations

The explicit dispersion-relation-equation preserving scheme developed in Section 3 for solving the Maxwell's equations is suitable for parallel implementation. The reason is that the update of electric field components requires only the available magnetic field values, and vice versa. The explicit and relatively local nature of the difference scheme makes it ideal for application on vector computing platforms such as Graphics Processing Units (GPUs). With this in mind, we have implemented the Maxwell equation solver described here for execution on the Nvidia Tesla K20 GPU device with the target of reducing the time required for the numerical solution of the three-dimensional equations. The implementation contained herein was performed using Nividia's Compute Unified Device Architecture (CUDA).

4.1 Description of GPU K-20 architecture

Nvidia's Tesla K20 Computing GPU device contains thirteen streaming multiprocessor (SMX units), each of which contains a large number of cores with the clock frequency of 700M Hz. Specifically, each SMX contains 192 single precision scalar processors (SPs) and 64 double precision scalar processors (DPs) capable of performing arithmetic computations across varying precisions. In addition, each SMX contains 32 SFUs (Special Function Units) and 48 KB of on-chip shared memory. This shared memory makes it possible for threads running simultaneously within the SMX to access the same local cache. This is possible in CUDA since threads – which serve as the basic execution unit on the device – are grouped into blocks of threads (i.e. groups of threads) which are executed on SMX's. The number and structure of threads within each block may vary depending on the user needs and the number of registers required by the kernels which contain the information executed by each thread. The typical number of threads is between 64 to 1024 per block. Seeing as there are 13 SMXs contained within a single K20 GPU device, an efficient parallel computation on the device will ensure that each SMX is computationally occupied during runtime. It is meant that we may have thousands of threads running concurrently – distributed across the SMX's – if the computational scale is large enough and each thread is kept busy long enough. The scale of this parallelization contrasts with conventional shared memory parallelization where a single workstation may only execute 8-32 threads simultaneously. As mentioned earlier, each block of threads is executed in a single SMX, meaning that each of the threads in this block has access to the same shared memory cache, which is located on-chip as opposed to the GPU's standard form of data storage – global memory – which is off-chip. The lifetime of this local cache is the same as the lifetime of the block, since it is cleared in preparation for the next block of threads to be executed upon it.

Unlike the various layers of memory and cache available to conventional CPU architectures – which is several MBs in size – computation on the GPU device uses its global memory as the main storage medium due to the low size and volatility of the smaller on-chip caches. The lowest level memory on the GPU device is the local memory – which is on-chip – which is available to single threads, very small in size and has the same lifetime as a thread. The only permanent means of storing data on the GPU device is within global memory, or through the application of texture memory (which will not be investigated here). Unfortunately, the high latency of global memory may have negative impact on the parallel performance of the GPU device. The access pattern of computations involving global memory also plays a role in determining the efficiency and speed of global memory use. Hence, in order to increase performance on the GPU, we aim to increase the number of computations performed using data contained within the on-chip caches, such as shared memory.

4.2 Solution algorithm implemented in CPU/GPU

The solver proposed in this study is implemented through a parallelized Fortran code. At the highest level, the resulting application is executed through calculation across a large number of thread blocks by the streaming multiprocessor units contained within the GPU device. Within each block, the threads are further subdivided into smaller groups of threads known as ”warps” during the execution on the SMX. Hence, the number of threads contained within each block is usually a multiple of the number of threads contained within each warp. In the computations employed within this study, which are performed on the Nvidia Tesla K20 GPU device, we employ the maximum number of threads per block, which is 1024. Each warp contains 32 threads.

Considering the 3D EM wave simulation employing 121^3 grid points as an example, the arrangement of blocks and threads for the calculation of the first order derivative terms is schematically shown in Fig. 3. A grid is organized as a 3D array of blocks and each block is organized as a 2D array of threads. As discussed earlier, the performance of the parallel CUDA implementation on the GPU depends highly on the size of blocks and the number of threads per block as this is related to the total number of blocks to be distributed across the SMXs during computation. Generally speaking, in the work contained in this study, a good strategy was for determining the target number of threads N_t and warps N_w was to follow the rule of $\frac{N_t}{N_w}=32n$, where n is an integer. Therefore, assuming our simulation of EM wave propagation employed 121^3 mesh points, the number of blocks would be 8^2*121 with each block containing a $16*16$ two-dimensional array of threads. As part of the study into the efficiency of our parallel implementation on GPU, performance data for various thread and block arrangements for our currently proposed solver can be seen in Table 3.

With regard to the Nvidia Tesla K20c and our specific application, each SMX accom-

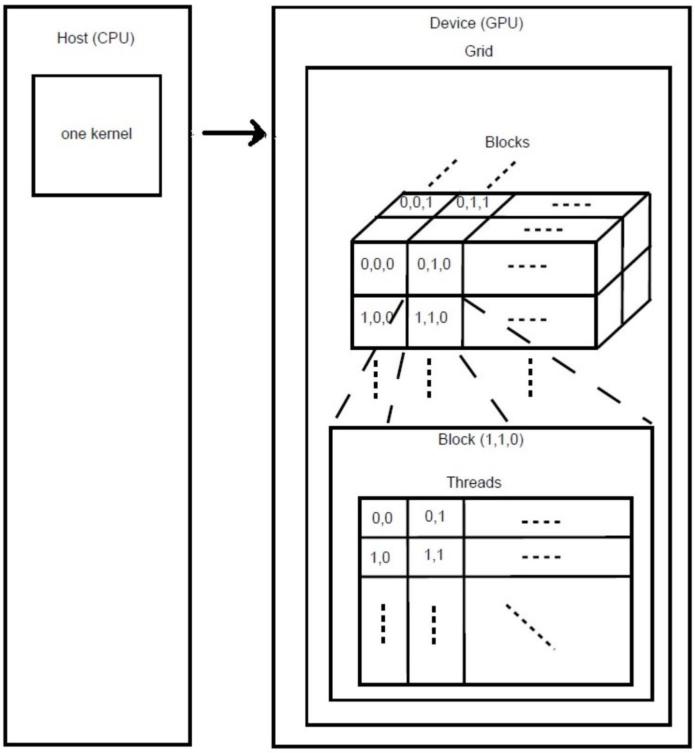


Figure 3: Schematic of the employed CPU-GPU architecture. A grid is organized as a 3D array of blocks while in each block it is organized as a 2D array of threads.

Table 3: Comparison of the computing times at $t = 1000\ \Delta t$ in four different arrangements of the blocks per grid and the threads per block.

blocks of a grid	**threads of a block**	**Time(s)**
$8\times8\times121$	$16\times16\times1$	117.7910
$8\times8\times61$	$16\times16\times2$	127.5680
$8\times4\times121$	$16\times32\times1$	128.6950
$4\times4\times121$	$32\times32\times1$	127.2910

modates up to 596 blocks, thereby permitting the total simultaneously executing thread number to be 152576. In Fig. 4 the correspondence between the hardware structure and CUDA programming is illustrated. Each block carrying out thread execution has 256 threads marked with the thread id number. As mentioned above, there is a difference in latency for the global (off-chip) and shared (on-chip) memory – for the global memory, the Load/Store time (LD/S) is 800 cycles which is much slower than shared memory (80 cycles). Such a large discrepancy in the needed LD/ST times sheds light on the necessity of exploiting shared memory. Effective utilization of shared memory, together with a proper determination of the number of blocks and the number of threads per block, are

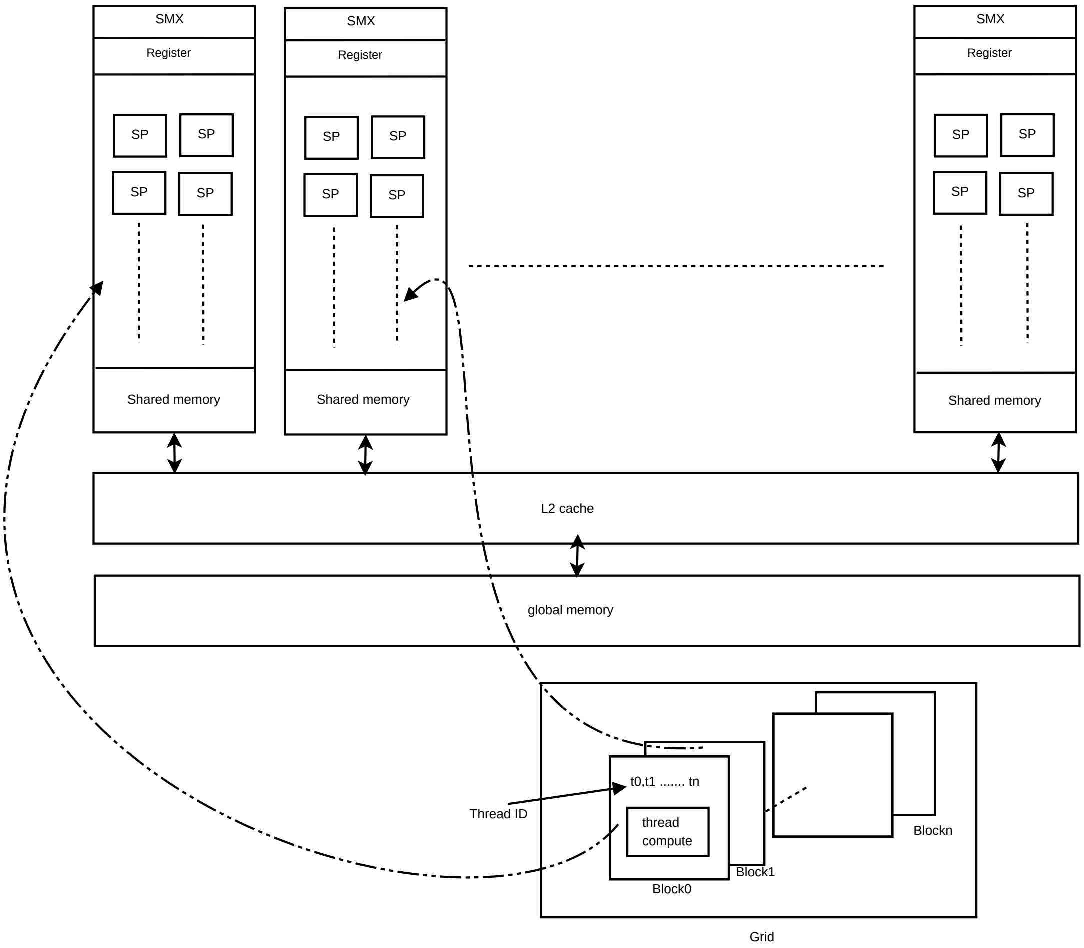


Figure 4: Schematic of the relation of K20c (hardware) with the CUDA program. Note that $t_0,\cdots,t_n$ denote the thread Id numbers. The thread blocks #1, 2, $\cdots$ constitute a block grid. CUDA contains block grid, thread block, and threads.

the keys to increasing the overall performance of algorithms such as the one proposed on Tesla K20c.

5 Numerical results

To demonstrate the applicability of the proposed solver, we first need to verify the proposed error-minimized high-order parallel scheme implemented on the GPU while solving Maxwell's equations and then compare the computed results against the analytical solutions available. Following comparison against analytical solutions – which are often too simplistic for practical demonstration – the computer code is further validated through the Mie scattering benchmark problem. Finally, an application will be demonstrated – in this case, a three-dimensional waveguide problem – and the results reported together with the parallel computational performance.

5.1 Verification studies

The proposed explicit symplectic PRK scheme developed in non-staggered grids is verified by solving the three dimensional Maxwell's equations amenable to the exact solution in a cube of $-\pi \leq x \leq \pi$, $-\pi \leq y \leq \pi$ and $-\pi \leq z \leq \pi$. The solution sought at $\mu = 1$ and $\epsilon = 1$ is subject to the initial solenoidal solutions

$$\begin{aligned}
&E_x(x,y,z,0) = E_y(x,y,z,0) = E_z(x,y,z,0) = 0,\\
&H_x(x,y,z,0) = \cos(x+y+z),\\
&H_y(x,y,z,0) = \frac{1}{2}(-1+\sqrt{3})\cos(x+y+z),\\
&H_z(x,y,z,0) = -\frac{1}{2}(1+\sqrt{3})\cos(x+y+z).
\end{aligned}$$

The exact electric and magnetic field solutions to Eqs. (2.1)-(2.2) are given by

$$\begin{aligned}
&E_x(x,y,z,t) = \sin(\sqrt{3}t)\sin(x+y+z),\\
&E_y(x,y,z,t) = -\frac{1}{2}(1+\sqrt{3})\sin(\sqrt{3}t)\sin(x+y+z),\\
&E_z(x,y,z,t) = \frac{1}{2}(-1+\sqrt{3})\sin(\sqrt{3}t)\sin(x+y+z),\\
&H_x(x,y,z,t) = \cos(\sqrt{3}t)\cos(x+y+z),\\
&H_y(x,y,z,t) = \frac{1}{2}(-1+\sqrt{3})\cos(\sqrt{3}t)\cos(x+y+z),\\
&H_z(x,y,z,t) = -\frac{1}{2}(1+\sqrt{3})\cos(\sqrt{3}t)\cos(x+y+z).
\end{aligned}$$

The spatial rate of convergence is computed first at $\Delta t = 10^{-5}$, which is much smaller than the grid sizes chosen as $\Delta x = \Delta y = \Delta z = \pi/5, \pi/10, \pi/15$ and $\pi/20$ in this study. From the predicted L_2-error norms tabulated in Table 4(a) one can see that there is only a very

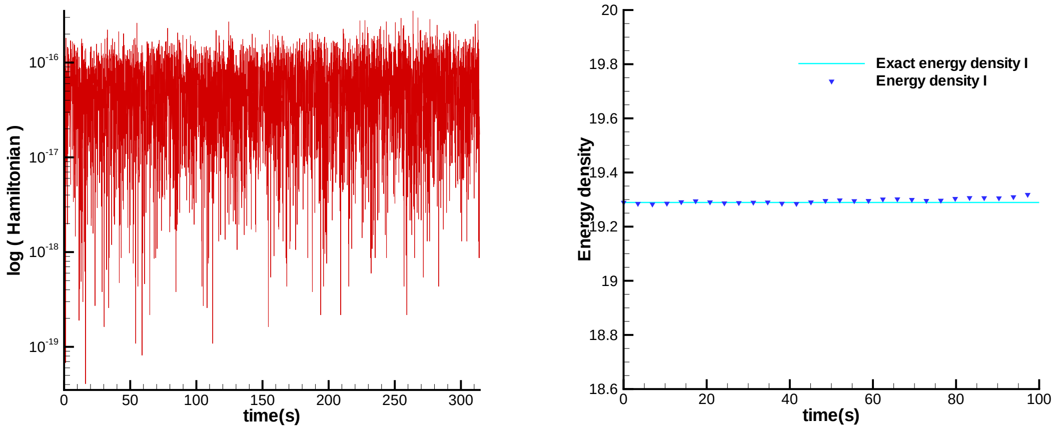
Table 4: The predicted L_2 errors and the corresponding for the analytical test problem investigated in Section 5.1. (a) spatial rates of convergence (sroc); (b) temporal rates of convergence (troc).

(a)

meshes	L_2- **error norm of** E_z	**sroc**
$10\times10\times10$	1.8366E-05	-
$20\times20\times20$	1.2339E-06	3.8957
$30\times30\times30$	8.5169E-08	3.8567
$40\times40\times40$	5.3609E-09	3.9897

(b)

Δt	L_2- **error norm of** E_z	**troc**
1/1000	2.04514E-02	-
1/2000	5.17849E-03	1.9816
1/3000	1.32417E-03	1.9674
1/4000	3.79144E-04	1.8043



(a)

(b)

Figure 5: The computed and exact energy densities, shown in (2.3)-(2.4), are plotted with respect to time for the analytical problem in Section 5.1 using the proposed phase velocity optimized compact difference scheme. (a) Hamiltonian function; (b) Energy density.

Table 5: Comparison of the computing times and speedups in CPU and GPU when performing computations in different meshes at the time $t = 1000\ \Delta t$.

meshes	CPU time(s)	with GPU time(s)	Speedup
$81\times81\times81$	1270.21	67.03	18.95
$121\times121\times121$	5900.42	117.79	50.55
$161\times161\times161$	20994.74	215.26	97.53
$201\times201\times201$	94568.75	486.70	194.31

small difference between the predicted spatial rate of convergence and the theoretically derived fourth-order accuracy. The temporal rate of convergence tabulated in Table 4(b) is also seen to be very closed to the theoretical order, which is second order.

The Hamiltonian defined in (2.3) and the energy density given in (2.4) are computed from the predicted solutions of $\underline{E}$ and $\underline{H}$ for additional theoretical justification of the proposed scheme. One can conclude from the results shown in Fig. 5 that the computed Hamiltonian and energy density are deemed to change little with time. The predicted norms of $\nabla\cdot\underline{H}$ and $\nabla\cdot\underline{E}$ are also plotted with respect to time to ensure that Gauss's law is indeed satisfied discretely. In Fig. 6, the predicted magnetic field predicted by the proposed scheme is essentially divergence-free. We also assess the proposed scheme, which has equally accurate solutions computed at each spatial point, with four other commonly referenced schemes in terms of the required CPU times for the same predicted L_2-error norm. The results shown in Tables 5 and 6 justify the choice of the proposed scheme for the solution to the Maxwell equations.

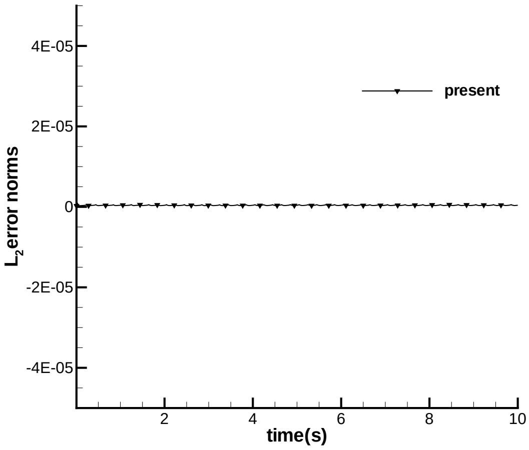


Figure 6: The computed L_2-norm of $\nabla\cdot\underline{H}$ is plotted with respect to time using the present explicit partitioned Runge-Kutta symplectic scheme.

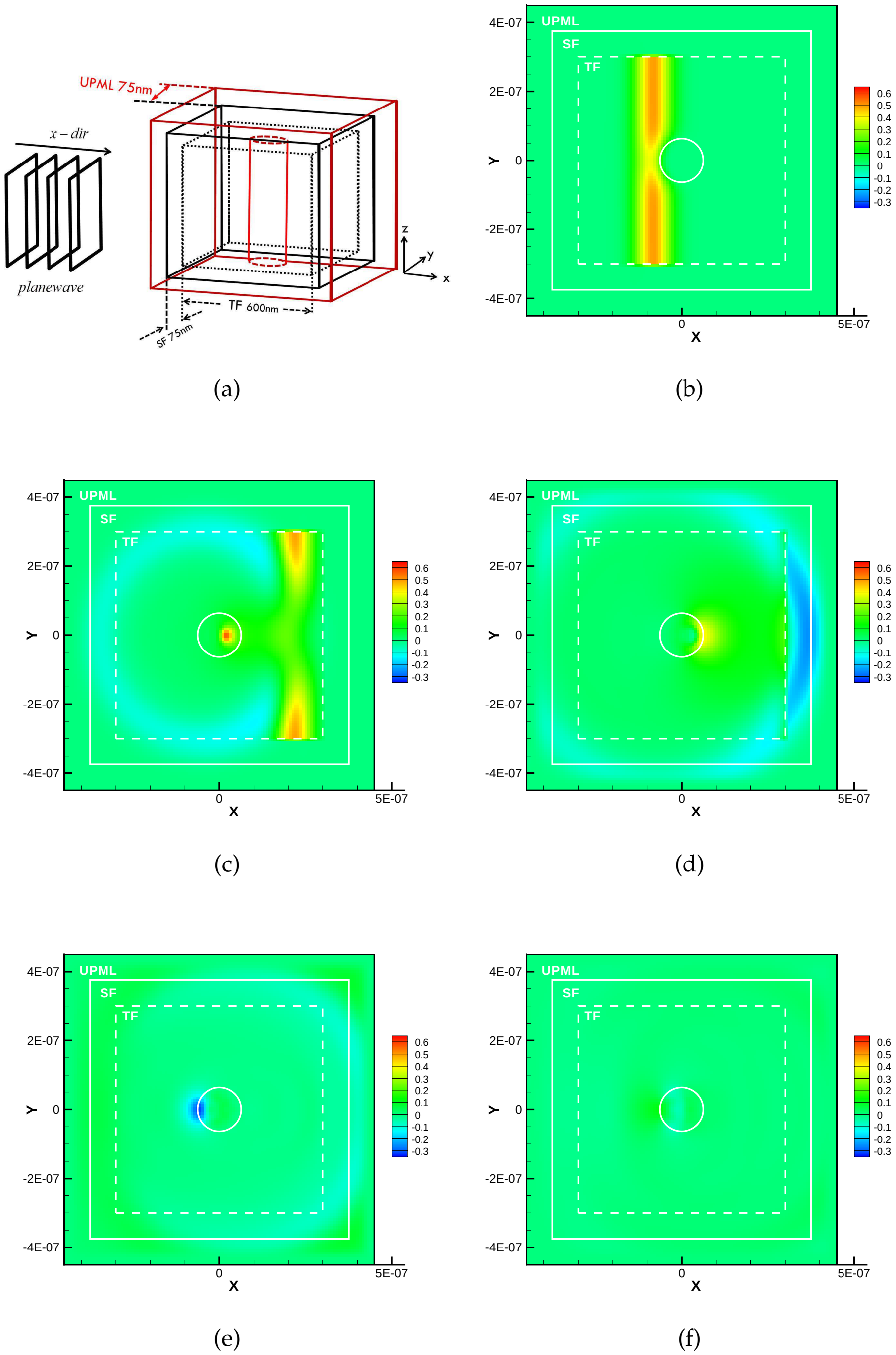
Table 6: Comparison of the required CPU times using the present scheme with the optimized phase velocity and the 2D Yee's scheme given in [2]. Note that these schemes yield approximately the same L_2-error norm, at time $t=10$, at completely different grid resolutions.

present			**Yee [2]**		
L_2-error norm	grid	CPU time (s)	L_2-error norm	grid	CPU time (s)
8.7962E-03	41×41	0.1092	8.3483E-03	181×181	4.9608
4.0657E-03	51×51	0.2028	4.8228E-03	251×251	13.0261
2.0981E-03	61×61	0.3276	3.0141E-03	341×341	33.0092

5.2 Mie scattering problem

The diameter of a dielectric cylinder under current study is 126.56 nm. As shown in Fig. 7(a), this isotropic cylinder located at the center of a cube with volume of 720^3 nm^3 has $\epsilon_r=12.1104$. The cross-sectional area in this case is 760×760 nm^2. For this benchmark problem, an incident x-polarized plane wave with amplitude of 0.5 $\frac{v}{m}$ and angular frequency of 13.263 $\frac{rad}{s}$ propagates rightward according to the one-dimensional Maxwell equations $\frac{\partial E_z}{\partial t}=\frac{1}{\epsilon}\nabla\times H$, $\frac{\partial \underline{H}}{\partial t}=-\frac{1}{\mu}\nabla\times\underline{E}$.

In the presence of a single dielectric cylinder, the incident wave is scattered so that the total field/scattered field formulation is adopted. The physical domain is divided into the regions known as the total field, scattered field, and the uni-axial perfectly matched layer to absorb waves. The results are calculated at the same Courant number $Cr=0.2$, which corresponds to the specified time increment $\Delta t=0.0026685$ fs. The three-dimensional results of E_z are plotted in Figs. 7(b)-(f) at the cutting plane $z=0$ nm.



(a) (b)

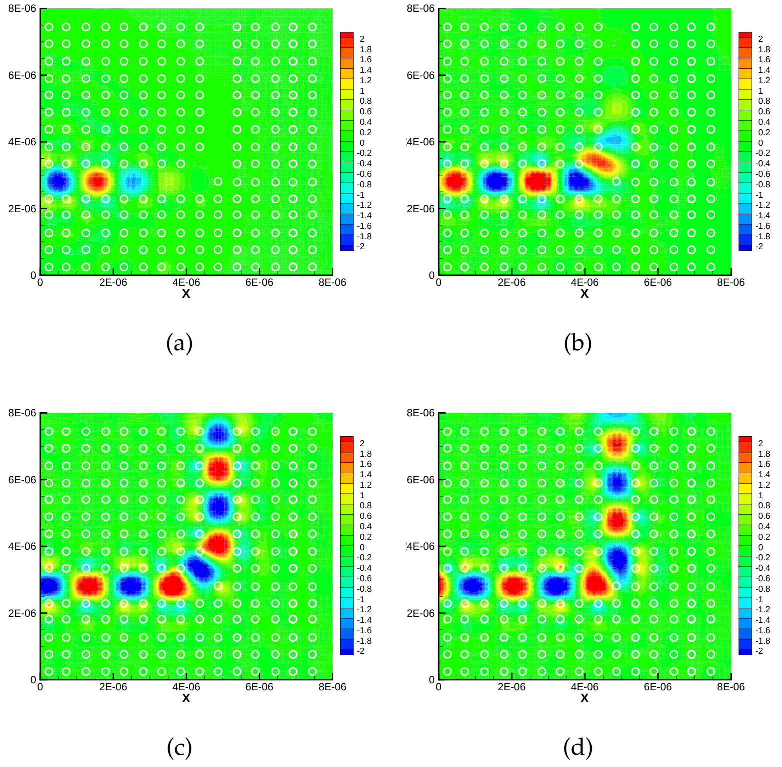
(c) (d)

(e) (f)

Figure 7: (a) Schematic of the 3D Mie scattering problem; The predicted time-varying contours E_z $(z=0)$ at the cutting plane containing a cylindrical scatter. (b) time step$=560$ (2.8 fs); (c) time step$=760$ (3.8 fs); (d) time step$=1350$ (4.25 fs); (e) time step$=1600$ (5.8 fs); (f) time step$=1900$ (9 fs).

5.3 Photonic crystal L-shaped bent waveguide problem

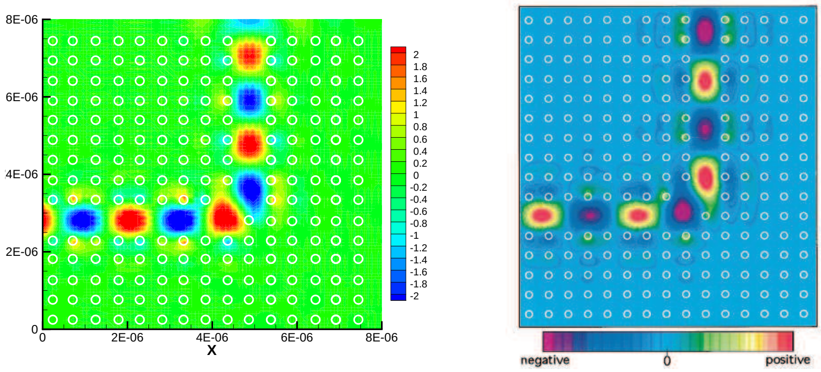
To demonstrate the practicality of the scheme, a waveguide application problem is demonstrated. In this problem, a group of dielectric pillars are placed within a domain containing a L-shaped defect channel. The relative permittivity of the medium is set at 1 and the dielectric constant of the pillars is assumed to be $\epsilon_r = 11.56$. The waveguide with the width $\frac{w}{a} = 2$ has been defected by taking one vertical pillar and one row pillar away from the otherwise uniformly distributed lattice. Light with a frequency belonging to the photonic band gap is confined to the three-dimensional L-shaped defect channel. At the start of the problem (t=0), a transverse magnetic wave incident into the domain propagates rightward. In this case, the pillar radius is chosen to be 0.2a, where the lattice constant a $(=515\ nm)$ denotes the length between the centroids of two adjacent pillars. The L-shaped waveguide problem is simulated at $\Delta t = 0.05337\ fs$ in a domain discretized into a uniform grid with spacing of 57.1429 nm. The uni-axial perfectly matched layer enclosing the scatter field is applied to avoid unphysical wave re-entering into the domain. Fig. 8 shows the time-evolution of E_z contours computed at the normalized frequency



(a) (b)

(c) (d)

Figure 8: The contours of E_z $(z=0)$ predicted at the cutting plane and at different times. (a) time step=575 (30.68775 fs); (b) time step=1075 (57.37275 fs); (c) time step=1750 (93.3975 fs); (d) time step=2325 (124.08525 fs).



(a) (b)

Figure 9: The computed E_z field values at the time $t = 124.08525 fs$ in the PC-based L-shape waveguide at the normalized frequency $0.353(c/a)$. (a) Present; (b) Mekis et al. [33].

$= 0.353$ $(\frac{c}{a})$ (or wavelength = 1458.92 nm), where c denotes the speed of light.

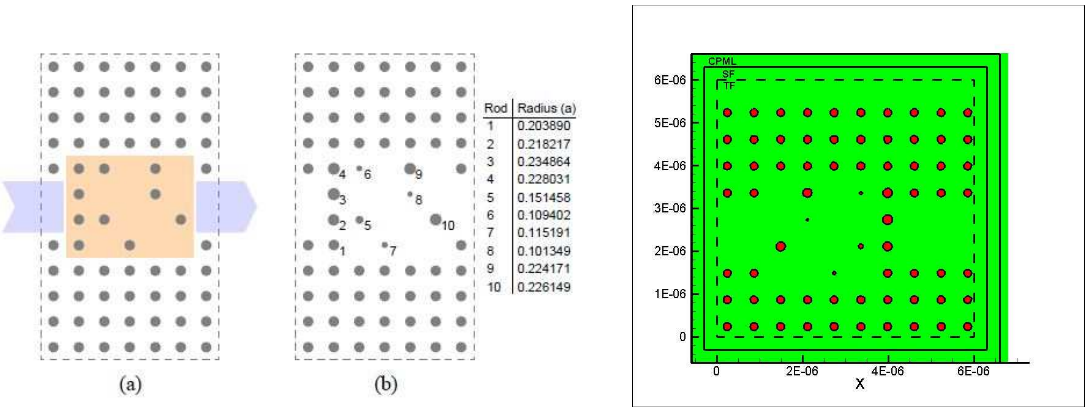
The optical wave is seen to propagate through the ninety degree bend and the electric field is concentrated mostly in the defect channel. Comparison of the current three-dimensional result with the two-dimensional result of Mekis et al. [33] is illustrated in Fig. 9.

5.4 Simulation of a photonic crystal waveguide spatial mode converter

To accelerate the speed of information processing in an optical system, there is a growing interest in employing ever-improving multiple spatial modes on, for example, on-chip applications [34]. For these optical communication systems, their basic building blocks are to perform spatial mode manipulation effectively in a sense that one waveguide mode can be transferred to another with a high conversion efficiency.

In this study we considered the extremely compact photonic crystal waveguide mode converter employed before in [34], aiming at converting the fundamental even-symmetric waveguide mode into the second odd-symmetric mode. The structure under current investigation, which is described schematically in Fig. 10, consists of photonic crystal of silicon rods with $\epsilon_r = 12$ of radius 0.2a in air with $\epsilon_r = 1$. To enable mode conversion, two lines of rods are purposely removed, thereby leading to input and output waveguide subject to out-of-plane electric fields. The aim of this study is to numerically investigate how and whether this mode-converter device can convert fundamentally even modes on a square lattice with lattice constant a=623.875 nm.

In this numerical study, simulations were carried out in a cubic domain of length 6000 nm containing 191 uniformly distributed nodal points along the x, y and z directions. For avoiding non-physical wave reflection from the inflow boundary, both layers to account for the scatter field and CPML are attached.

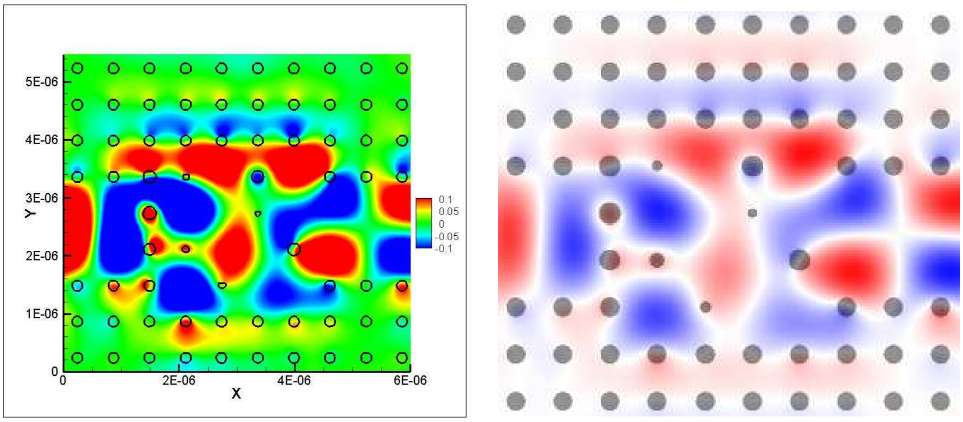


Rod	Radius (a)
1	0.203890
2	0.218217
3	0.234864
4	0.228031
5	0.151458
6	0.109402
7	0.115191
8	0.101349
9	0.224171
10	0.226149

(a) (b)

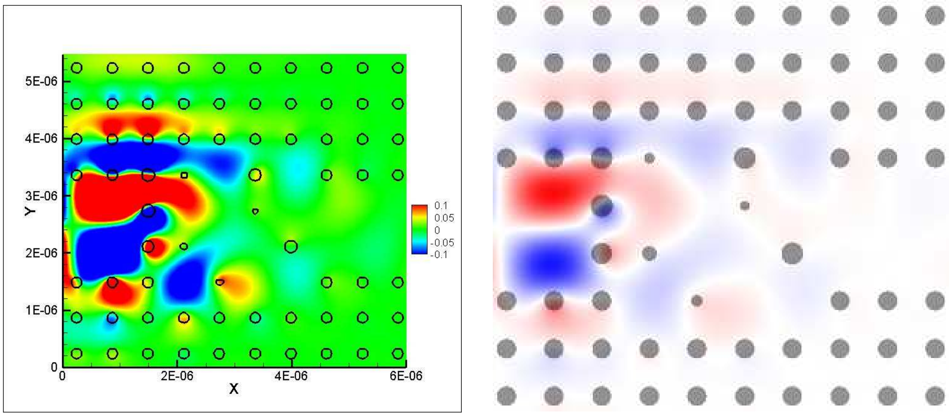
Figure 10: (a) Schematic of the problem in Section 5.4; (b) Schematic of the physical field of interest and the surrounding scatter field (SF) and absorbing field (CPML).

As shown in Fig. 10, the incident right-running wave has been electrically polarized with a normalized center of 0.4025 c/a. This input mode passes through the defect region (or coupler regime) marked in grey and converts its mode to the output mode. The simulated CPU/GPU results using the proposed three-dimensional GPU parallelized EM wave code are compared with several results of Liu et al. [24] at the catting plane $z=85$. One can observe from Figs. 11-14 good agreement with the reference solution in [24], confirming that the scheme proposed here may be applied to solve the solution of the Maxwell equations for the design of a waveguide converter.



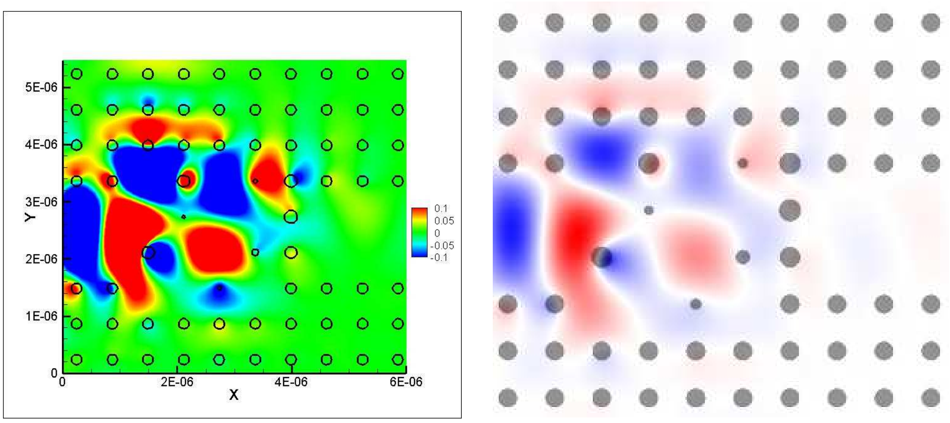
(a) (b)

Figure 11: The computed E_z field values at the time $t=135.38677fs$ in the Victor's design waveguide at the normalized frequency $0.4025(c/a)$. (a) Present; (b) Victor Li's. [34].



(a) (b)

Figure 12: The computed E_z field values at the time $t=133.92fs$ in the Victor's design waveguide at the normalized frequency $0.4025(c/a)$. (a) Present; (b) Victor Li's. [34].

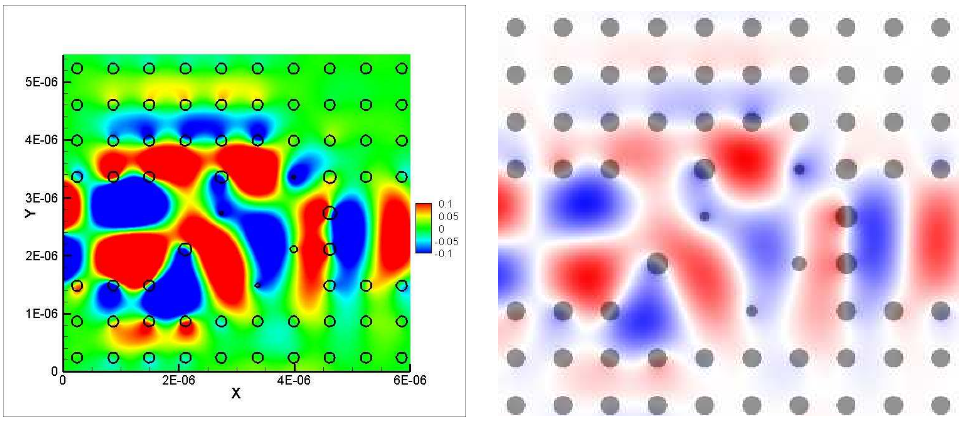


(a) (b)

Figure 13: The computed E_z field values at the time $t=138.346677fs$ in the Victor's design waveguide at the normalized frequency $0.4025(c/a)$. (a) Present; (b) Victor Li's. [34].

6 Conclusions

A high-order FDTD scheme has been developed using a three-point grid stencil for the numerical solution to the three-dimensional Maxwell's equations using computations applied on a non-staggered Cartesian grid. In this research, our first aim was to numerically preserve symplecticity and conserve Hamiltonians as required for increased temporal and spatial accuracy. To retain these theoretical properties in our solution to Maxwell's equations for large-scale transient computations, the explicit partitioned Runge-Kutta symplectic time integrator is applied together with the space-centered scheme. To in-



(a) (b)

Figure 14: The computed E_z field values at the time $t = 131.86677fs$ in the Victor's design waveguide at the normalized frequency $0.4025(c/a)$. (a) Present; (b) Victor Li's. [34].

crease the dispersive accuracy essential to a correct prediction of wave propagation, the discrepancy between the numerical and exact phase velocities is minimized. The numerically verified temporally second-order and spatially fourth-order accurate compact finite difference scheme developed here is then also shown to satisfy the discrete Gauss' law (or divergence-free magnetic and electric fields). The solutions computed from the analytical and benchmark problems for the verification and validation purposes have been shown to agree very well with the exact solutions (where available) and reference benchmark numerical solutions.

The explicit scheme proposed here features satisfaction of the dispersion relation equation for Maxwell's equations and has been implemented in parallel using Graphics Processing Units (GPU) using Nvidia's CUDA. In this study, the parallel implementation is described and simulations are executed on a single Nvidia K-20 card, demonstrating a considerable gain in speedup when compared to the serial CPU calculation using a single CPU core. The results from this study demonstrate that one may solve the three dimensional Maxwell's equations in reasonable computational time due to the combined exploitation of a high-order scheme with minimal error on phase velocity and parallelization on the GPU, reducing the simulation time considerably and thus allowing the proposed solver to be applied to practical design-related tasks.

Acknowledgments

This work was supported by the Ministry of Science and Technology (MOST) of the Republic of China under the Grants NSC96-2221-E-002-293-MY2, NSC96-2221-E-002-004, and CQSE97R0066-69.

References

[1] J. X. Cai, Y. S. Wang, B. Wang, and B. Jiang, New multisymplectic self-adjoint scheme and its composition scheme for the time-domain Maxwell's equations, J. Math. Phys. 47, 1-18, 2006.

[2] K. S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE T. Antenn. Propag. 14, 302-307, 1966.

[3] A. H. Panaretos and R. E. Diaz, A simple and accurate methodology to optimize parameter-dependent finite-difference time-domain schemes, IEEE Trans. Microw. Theory. Tech. Vol. 56(5), 1125-1136, 2008.

[4] B. Finkelstein and R. Kastner, Finite difference time domain dispersion reduction schemes, J. Comput. Phys. Vol. 221(1), 422-438, 2007.

[5] P. Hawke, Ed., Higher-order accurate method in time domain computational electromagnetics: A review in Advances in Imaging and Electron Physics, New York. Academic, Vol. 127, 59-123, 2003.

[6] S. Wang and F. L. Teixeira, A three-dimensional angle-optimized finite-difference time-domain algorithm, IEEE Trans. Microw. Theory. Tech. Vol. 31(3), 811-817, 2003.

[7] S. Wang and F. L. Teixeira, Dispersion-relation-preserving FDTD algorithms for large-scale three-dimensional problems, IEEE Trans. Antennas Propagat. Vol. 51(8), 1818-1828, 2003.

[8] K. L. Shlager and J. B. Schneider, Comparison of the dispersion properties of several low-dispersion finite-difference time-domain algorithms, IEEE Trans. Antennas Propagat. Vol. 31(3), 642-653, 2003.

[9] J. Chi, F. Liu, E. Weber, Y. Li, and S. Crozier, GPU-accelerated FDTD modeling of radio-frequency field-tissue interactions in high-field MRI, IEEE Trans. Biomed. Eng. Vol. 58(6), 1789-96, 2011.

[10] M. R. Zunoubi, J. Payne, and W. P. Roach, CUDA implementation of TE-FDTD solution of Maxwell's equations in dispersive media, IEEE Antennas and Propagation Society, Vol. 9, 756-759, 2010.

[11] K. H. Lee, I. Ahmed, R. S. M. Goh, E. H. Khoo, E. P. Li, and T. G. G. Hung, Implementation of the FDTD method based on Lorentz-Drude dispersive model on GPU for plasmonics applications, Progr. Electromagn. Res. 116, 441-456, 2011.

[12] T. T. Zygiridis, High-order error-optimized FDTD algorithm with GPU implementation, IEEE Trans. Magnetics, Vol. 49(5), 1809-1813, 2013.

[13] P. Micikevicius, 3D Finite Difference Computation on GPUs Using CUDA, ACM New York 79-84, 2009.

[14] B. Zhang, Z.H. Xue, W. Ren, W. M. Li, and X. Q. Sheng, Accelerating FDTD algorithm using GPU computing, IEEE (ICMTCE) 410-413, 2011.

[15] R. Shams and P. Sadeghi, On optimization of finite-difference time-domain (FDTD) computation on heterogeneous and GPU clusters, J. Parallel Distrib. Comput. 71, 584-593, 2011.

[16] T. J. Bridges and S. Reich, Multi-symplectic integration numerical scheme for Hamiltonian PDEs that conserves symplecticity, Phys. Lett. A 284, 184-193, 2001.

[17] B. Cockburn, F. Li, and C.-W. Shu, Locally divergence-free discontinuous Galerkin methods for the Maxwell equations, J. Comput. Phys. 194, 588-610, 2004.

[18] N. Anderson, A. M. Arthurs, Helicity and variational principles for Maxwell's equations, Int. J. Electron. 54, 861-864, 1983.

[19] J. E. Marsden, A. Weinstein, The Hamiltonian structure of Maxwell-Vlasov equations, Physical D, 4, 394-406, 1982.

[20] T. W. H. Sheu, Y. W. Hung, M. H. Tsai, and J. H. Li, On the development of a triple-preserving

Maxwell's equations solver in non-staggered grids, Int. J. Numer. Methods Fluids. 63, 1328-1346, 2010.

[21] J. P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys 114, 185-200, 1994.

[22] S. D. Gedney, An anisotropic perfectly matched layer absorbing media for the truncation of FDTD lattices, IEEE T. Antenn. Propag. 44, 1630-1639, 1996.

[23] R. W. Hockney, A fast solution of Poisson's equation using Fourier analysis, J. ACM 12(1), 95-113, 1965.

[24] H. S. Stone, An efficient parallel algonithm for the solution of a tridiagonal linear system of equations, J. ACM 20(1), 27-38, 1973.

[25] S. Sengupta, A. E. Lefohn, and J. D. Owens, A work-efficient step-efficient prefix sum algorithm, In Proceedings of the Workshop on Edge Computing using New Commodity Architectures pp. D-26-27, 2006.

[26] J. M. Sanz-Serna, Symplectic Runge-Kutta and related methods: recent results, Physica D 293-302, 1992.

[27] L. L. Jiang, J. F. Mao, and X. L. Wu, Symplectic finite-difference time-domain method for Maxwell equations, IEEE Trans. Magn. 42(8), 1991-1995, 2006.

[28] W. Sha, Z. X. Huang, M. S. Chen, and X. L. Wu, Survey on symplectic finite-difference time-domain schemes for Maxwell's equations, IEEE T. Antenn. Propag. 56, 493-510, 2008.

[29] S. K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 17, 328-346, 1996.

[30] D. W. Zingy, H. Lomax, and H. Jurgens, High-accuracy finite-difference schemes for linear wave propagation, SIAM J. Sci. Comput. 17, 328-346, 1996.

[31] H. Spachmann, R. Schuhmann, T. Weiland, High order spatial operators for the finite integration theory, ACES Journal 17(1), 11-22, 2002.

[32] T. Kashiwa, Y. Sendo, K. Taguchi, T. Ohtani, and Y. Kanai, Phase velocity errors of the non-standard FDTD method and comparison with other high-accuracy FDTD methods, IEEE Transactions on Magnetics 39(4), 2125-2128, 2003.

[33] A. Mekis, J. C. Chen, I. Kurland, S. Fan, P. R. Villeneuve, and J. D. Joannopoulos, High transmission through sharp bends in photonic crystal waveguides, Phys. Rev. Lett. 77, 3787-3790, 1996.

[34] Victor Liu, David A. B. Miller, Shanhui Fan, Ultra-compact photonic crystal waveguide spatial mode converter and its connection to the optical diode effect, Optics Express Vol. 20 (2012) 28388-28397.