A GPU-Accelerated Finite-Difference Time-Domain Scheme for Electromagnetic Wave Interaction With Plasma

Patrick D. Cannon and Farideh Honary

***Abstract*—A graphical processing unit (GPU)-accelerated finite-difference time-domain (FDTD) scheme for the simulation of radio-frequency (RF) wave propagation in a dynamic, magnetized plasma is presented. This work builds on well-established FDTD techniques with the inclusion of new time advancement equations for the plasma fluid density and temperature. The resulting FDTD formulation is suitable for the simulation of the time-dependent behavior of an ionospheric plasma due to interaction with an RF wave and the excitation of plasma waves and instabilities. The stability criteria and the dependence of accuracy on the choice of simulation parameters are analyzed and found to depend on the choice of simulation grid parameters. It is demonstrated that accelerating the FDTD code using GPU technology yields significantly higher performance, with a dual-GPU implementation achieving a rate of node update almost two orders of magnitude faster than a serial implementation. Optimization techniques such as memory coalescence are demonstrated to have a significant effect on code performance. The results of numerical tests performed to validate the FDTD scheme are presented, with a good agreement achieved when the simulation results are compared to both the predictions of plasma theory and to the results of the Tech-X VORPAL 4.2.2 software that was used as a benchmark.**

***Index Terms*—Electromagnetic propagation, finite-difference time-domain (FDTD) methods, graphical processing unit (GPU) computing, ionosphere, magnetized plasma.**

I. Introduction

SINCE the early 1990s and the explosion in interest around computational physics, the finite-difference time-domain (FDTD) method has become an increasingly popular and powerful technique for modeling the propagation of electromagnetic (EM) waves through a variety of media. Using the methodology first proposed by Yee in 1966 [1], an FDTD model simulates the fundamental EM and plasma wave interactions at a spatially discrete series of nodes in computational space. Approximate expressions for the fundamental governing equations are used to advance the field magnitude at each node in quantized time steps, following a leapfrog-style algorithm [2]. This method of simulation leads to a natural update of the simulated fields that is explicit in time.

The FDTD method has an advantage over many other numerical simulation techniques as it deals with complex, nonlinear, and impulsive interactions in a natural way, avoiding complex and computationally intensive linear algebra calculations. FDTD proves to be a particularly useful tool for geophysical modeling due to the fact that the grid-based structure allows material properties to be defined separately at each point, enabling variations in properties such as topography, composition, geomagnetic field, and plasma profile to be accurately represented [3]. The versatility of the FDTD technique has led to its application to a diverse range of geophysical problems, from the study of lightning [4] to the modeling of space weather effects on power grids [5], and advances in computing technology will serve only to further enhance the potential of this method. A detailed review of FDTD techniques with emphasis on the computational cost of each can be found in [6].

One of the most appealing applications of the FDTD method is in the study of the interaction of EM waves with plasmas, and the wide range of instabilities and nonlinear phenomena that may result. Many of these processes are not fully understood and would benefit from detailed study via numerical simulation. As FDTD is well suited to computationally modeling these scenarios, the original Yee FDTD scheme has often been adapted to describe the propagation of EM waves through dispersive media or simple plasmas using a variety of discretization and time-integration schemes [7]–[12]. Many FDTD schemes extend these formulations further to incorporate the anisotropic effect of a magnetic field on a plasma fluid, including the three-dimensional (3-D) scheme described by Young [13], which collocates plasma fluid velocity vector nodes with the E-field vector nodes in the computational grid for ease of coupling between the EM wave equations and the Lorentz equation of motion. This implementation includes a scalar pressure node located at the corner of each basic computational unit cell (Yee cell) to allow a warm plasma medium to be modeled. Lee and Kalluri [14] locate the plasma current nodes in the center of the Yee cell to avoid spatial averaging between nodes and introduce a dynamic plasma medium through inclusion of a time-dependent plasma frequency that is updated between simulation steps. Yu and Simpson [15] collocate the plasma current nodes with the E-field nodes and include individual

Manuscript received July 24, 2014; revised December 22, 2014; accepted April 03, 2015. Date of publication April 17, 2015; date of current version July 02, 2015. This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/K007971/1. The work of P. D. Cannon is supported by an STFC studentship.

The authors are with the Space Plasma Environment and Radio Science Group, Lancaster University, Lancaster LA1 4YB, U.K. (e-mail: p.cannon@lancaster.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAP.2015.2423710

coupled equations to describe the current due to electrons and both positive and negative ion species. This scheme has been used to develop a 3-D global Earth-ionosphere FDTD model which has been used to study low-frequency wave propagation [16], and has the potential to be coupled to other atmospheric models as part of a multiphysics simulator. Numerical simulation of the time-dependent behavior of the plasma medium in response to interaction with an EM wave has been performed by Gondarenko *et al*. [17] who use a one-dimensional (1-D) or two-dimensional (2-D) alternating direction implicit (ADI) finite-difference algorithm to simulate linear mode conversion processes due to a radio-frequency (RF) pump wave incident on an idealized ionospheric plasma. The numerical scheme considers slow timescales that are comparable to that of the plasma density evolution. This formulation is extended in [18] to include updates to the plasma temperature and density as part of the simulation algorithm to study the growth of density structures.

This paper presents the formulation and implementation of a 3-D FDTD model which has been developed for use in modeling the propagation of high-power radio waves through a dynamic, magnetized, and collisional plasma, and can be used as a virtual laboratory in which this scenario can be investigated in detail. This formulation extends previously established schemes with new finite-difference equations for the variation in plasma temperature and density with time, which are incorporated into the simulation update algorithm and as such allow the investigation of nonlinear perturbations of the plasma medium. This FDTD scheme has been developed to study the interaction between ionospheric plasmas and RF EM waves; however, the formulation is equally applicable to waves of any frequency interacting with laboratory-based plasma, astrophysical plasma, or any medium where a fluid plasma description holds, provided that sufficient computing resources are available and that stability criteria can be met.

In Section II, formation of the model's update algorithm is described, and factors contributing to the stability and accuracy of the scheme are discussed. The possibility of accelerating the FDTD code using graphical processing unit (GPU) technology is explored and found to be of great benefit to code performance.

Section III presents a series of numerical tests performed to validate the performance of the FDTD scheme. Simulation results are compared to the predictions of plasma theory and benchmarked against results provided by the VORPAL 4.2.2 software [19].

II. Methodology

A. Governing Equations

The formation of the FDTD algorithm assumes a multifluid description of a dynamic, anisotropic, collisional plasma, in which electron or charged ion species are treated as individual fluids of continuous mass and charge. The effect on wave propagation due to the presence of plasma is introduced through the coupling of Maxwell's wave equations with the Lorentz equations of motion for each constituent plasma species, with anisotropy introduced through inclusion of a static

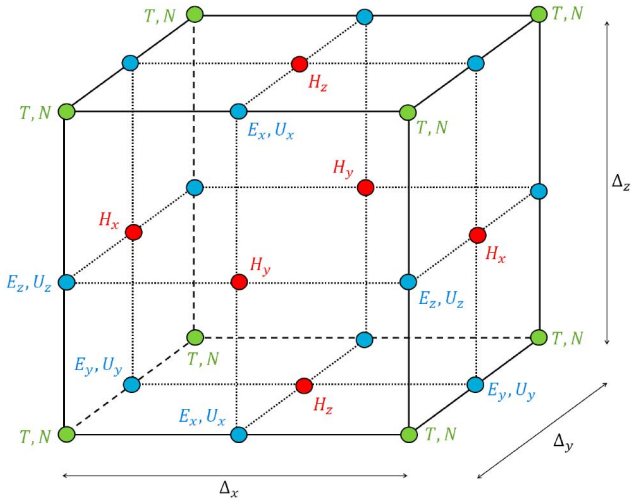

Fig. 1. Basic computational grid unit cell, with the positions of field nodes indicated.

externally applied magnetic field. The time-dependent variation in plasma temperature and density is treated by the inclusion of expressions for the dynamic behaviors of small perturbations of the plasma fluid temperature and density [20]. Together, these form a set of coupled first-order partial differential equations which govern the time-dependent behavior of the EM wave and plasma medium

$$\boldsymbol{\nabla} \times \mathbf{E} = -\mu_0 \frac{\partial \mathbf{H}}{\partial t} \tag{1}$$

$$\nabla \times \mathbf{H} - \sum_a N_a e_a \mathbf{U}_a = \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \tag{2}$$

$$N_a m_a \frac{\partial \mathbf{U}_a}{\partial t} = N_a e_a (\mathbf{E} + \mathbf{U}_a \times \mathbf{B}) - N_a m_a \nu_a \mathbf{U}_a - \boldsymbol{\nabla}(k_B N_a T_a) \tag{3}$$

$$\frac{\partial N_a}{\partial t} + \nabla \cdot (N_a \mathbf{U}_a) = \triangle N_a \tag{4}$$

$$\frac{3}{2} k_B \frac{\partial}{\partial t}(N_a T_a) + \nabla \cdot \mathbf{Q}_a - N_a e_a \mathbf{E} \cdot \mathbf{U}_a - \triangle \varepsilon_a = 0. \tag{5}$$

In these expressions, subscript a refers to plasma component species. $\mathbf{U}$ is the time-varying fluid bulk velocity vector, $\mathbf{B} = B\hat{\mathbf{b}}$ is the static background magnetic flux density, T and N are the plasma temperature and number density, ν is the effective collision frequency, $\triangle N$ is a plasma species loss rate term, $\nabla \cdot \mathbf{Q}$ describes the heat flux transport, and $\triangle \varepsilon$ is a collisional heating term. These expressions reduce to that of an unmagnetized plasma on removal of the $\mathbf{B}$ term, or a collisionless plasma on removal of the ν term.

B. Discretization Scheme

The set of governing equations (1)–(5) is amenable to a full finite-difference treatment. The discretization scheme used in this model is shown in Fig. 1, which shows the spatial location of the electric field, magnetic field, and fluid velocity vector components, along with the temperature and density scalar nodes, in a grid unit cell. The $\mathbf{E}$ and $\mathbf{U}$ component nodes

are taken to lie on the same points of the grid to facilitate efficient coupling between the electric field and the plasma, with E_x and U_x located at $(m+\frac{1}{2}, n, p)$, E_y and U_y located at $(m, n+\frac{1}{2}, p)$, and E_z and U_z located at $(m, n, p+\frac{1}{2})$, where m, n, and p are integer coordinates describing the x, y, and z Cartesian position of a node in the computational grid in units of the discrete special steps $\triangle_x$, $\triangle_y$, and, $\triangle_z$, respectively. To provide a finite differencing scheme consistent with the curl operators in Maxwell's (1) and (2), the $\mathbf{H}$ nodes are spatially offset from the $\mathbf{E}$ nodes, with H_x located at $(m, n+\frac{1}{2}, p+\frac{1}{2})$, H_y located at $(m+\frac{1}{2}, n, p+\frac{1}{2})$, and H_z located at $(m+\frac{1}{2}, n+\frac{1}{2}, p)$. Temperature and density nodes are located at (m, n, p) in each cell to allow accurate finite-difference evaluation of the grad and div operations in (3)–(5). To facilitate consistent evaluation of the temporal partial derivatives, the $\mathbf{E}$, T, and N nodes are calculated at integer values of discrete time step $q\triangle_t$ (where q is an integer), while the $\mathbf{H}$ and $\mathbf{U}$ nodes are calculated at half-integer time steps $(q+\frac{1}{2})\triangle_t$.

1) Update Equation for Magnetic Field: The formation of the FDTD update equations for the magnetic field is well established and proceeds as described in [1] and others. For the calculations below, equal spatial step sizes will be assumed in all grid directions ($\triangle_x = \triangle_y = \triangle_z$). The temporal partial derivative in (1) is expanded as a second-order finite-difference approximation around integer time step q and the equation rearranged to give an expression for $\mathbf{H}$ at time step $q+\frac{1}{2}$. The spatial derivatives in the curl term of (1) are expanded about the Yee cell points $(m, n+\frac{1}{2}, p+\frac{1}{2})$, $(m+\frac{1}{2}, n, p+\frac{1}{2})$, and $(m+\frac{1}{2}, n+\frac{1}{2}, p)$ to give independent update equations for H_x, H_y, and H_z, respectively, shown in (6) at the bottom of the page, where the notation $H_x^{q+\frac{1}{2}}[m, n+\frac{1}{2}, p+\frac{1}{2}]$ represents the value of the x component of $\mathbf{H}$ that exists at point $(m, n+\frac{1}{2}, p+\frac{1}{2})$ in the FDTD grid at time step $q+\frac{1}{2}$.

2) Update Equation for Electric Field: The classic Yee method from [1] is likewise followed to form update equations for the E nodes, with the temporal partial derivative in (2) expanded around half-integer time step $q+\frac{1}{2}$ and the spatial partial derivatives in the curl term expanded about Yee cell points $(m+\frac{1}{2}, n, p)$, $(m, n+\frac{1}{2}, p)$, and $(m, n, p+\frac{1}{2})$ for the E_x, E_y, and E_z update equations, respectively. The location of the $E_{x,y,z}$ and $U_{x,y,z}$ nodes at the same point of the fundamental grid cell allows the Yee formulation to be easily extended to include the term coupling the E-field to the particle velocity in a similar method to [16] with no loss of accuracy due to spatial or temporal averaging. The complete $\mathbf{E}$ node update equations are shown in (7) (as shown at the bottom of the page).

$$\mathbf{U}_a^{q+\frac{1}{2}} = \mathbf{R}_a \cdot \mathbf{U}_a^{q-\frac{1}{2}} + \left(\frac{e_a\triangle_t}{m_a}\right)\mathbf{S}_a^{-1}\cdot\mathbf{E}^q + \left(\frac{k_B\triangle_t}{m_a N_a^q}\right)\mathbf{S}_a^{-1}\cdot(\boldsymbol{\nabla} N_a^q T_a^q) \tag{8}$$

$$\mathbf{S}_a = \left\{\mathbf{I} - \frac{1}{2}\triangle_t\begin{pmatrix} -\nu_a & -\omega_{ca}b_z & \omega_{ca}b_y \\ \omega_{ca}b_z & -\nu_a & -\omega_{ca}b_x \\ -\omega_{ca}b_y & \omega_{ca}b_x & -\nu_a \end{pmatrix}\right\} \tag{9}$$

$$\mathbf{R}_a = \mathbf{S}_a^{-1}\left\{\mathbf{I} + \frac{1}{2}\triangle_t\begin{pmatrix} -\nu_a & -\omega_{ca}b_z & \omega_{ca}b_y \\ \omega_{ca}b_z & -\nu_a & -\omega_{ca}b_x \\ -\omega_{ca}b_y & \omega_{ca}b_x & -\nu_a \end{pmatrix}\right\} \tag{10}$$

3) Update Equation for Fluid Velocity: FDTD treatment of plasma fluid velocity is well established and here it proceeds following the methods of [13]. The temporal partial differential in (3) is expanded as a finite-difference approximation about integer time step q. As the $\mathbf{U}$ nodes are only known at half-integer time steps, the undifferentiated $\mathbf{U}$ terms on the right-hand side of (3) must then be temporally averaged around q as shown by

$$\mathbf{U}^q = \frac{\mathbf{U}^{q+\frac{1}{2}} + \mathbf{U}^{q-\frac{1}{2}}}{2}. \tag{14}$$

$$\begin{Bmatrix} H_x^{q+\frac{1}{2}}[m,n+\frac{1}{2},p+\frac{1}{2}] \\ H_y^{q+\frac{1}{2}}[m+\frac{1}{2},n,p+\frac{1}{2}] \\ H_z^{q+\frac{1}{2}}[m+\frac{1}{2},n+\frac{1}{2},p] \end{Bmatrix} = \begin{Bmatrix} H_x^{q-\frac{1}{2}}[m,n+\frac{1}{2},p+\frac{1}{2}] \\ H_y^{q-\frac{1}{2}}[m+\frac{1}{2},n,p+\frac{1}{2}] \\ H_z^{q-\frac{1}{2}}[m+\frac{1}{2},n+\frac{1}{2},p] \end{Bmatrix} - \frac{\triangle_t}{\mu_0\triangle_x}\begin{Bmatrix} E_z^q[m,n+1,p+\frac{1}{2}] - E_z^q[m,n,p+\frac{1}{2}] - E_y^q[m,n+\frac{1}{2},p+1] + E_y^q[m,n+\frac{1}{2},p] \\ E_x^q[m+\frac{1}{2},n,p+1] - E_x^q[m+\frac{1}{2},n,p] - E_z^q[m+1,n,p+\frac{1}{2}] + E_y^q[m,n,p+\frac{1}{2}] \\ E_y^q[m+1,n+\frac{1}{2},p] - E_z^q[m,n+\frac{1}{2},p] - E_x^q[m+\frac{1}{2},n+1,p] + E_x^q[m+\frac{1}{2},n,p] \end{Bmatrix}. \tag{6}$$

$$\begin{Bmatrix} E_x^{q+1}[m+\frac{1}{2},n,p] \\ E_y^{q+1}[m,n+\frac{1}{2},p] \\ E_z^{q+1}[m,n,p+\frac{1}{2}] \end{Bmatrix} = \begin{Bmatrix} E_x^q[m+\frac{1}{2},n,p] \\ E_y^q[m,n+\frac{1}{2},p] \\ E_z^q[m,n,p+\frac{1}{2}] \end{Bmatrix} - \left(\frac{\triangle_t}{\varepsilon_0}\right)\sum_a e_a \begin{Bmatrix} N_a^{q+\frac{1}{2}}[m+\frac{1}{2},n,p]U_{x,a}^{q+\frac{1}{2}}[m+\frac{1}{2},n,p] \\ N_a^{q+\frac{1}{2}}[m,n+\frac{1}{2},p]U_{y,a}^{q+\frac{1}{2}}[m,n+\frac{1}{2},p] \\ N_a^{q+\frac{1}{2}}[m,n,p+\frac{1}{2}]U_{z,a}^{q+\frac{1}{2}}[m,n,p+\frac{1}{2}] \end{Bmatrix} + \frac{\triangle_t}{\varepsilon_0\triangle_x}\begin{Bmatrix} H_z^{q+\frac{1}{2}}[m+\frac{1}{2},n+\frac{1}{2},p] - H_z^{q+\frac{1}{2}}[m+\frac{1}{2},n-\frac{1}{2},p] - H_y^{q+\frac{1}{2}}[m+\frac{1}{2},n,p+\frac{1}{2}] + H_y^{q+\frac{1}{2}}[m+\frac{1}{2},n,p-\frac{1}{2}] \\ H_x^{q+\frac{1}{2}}[m,n+\frac{1}{2},p+\frac{1}{2}] - H_x^{q+\frac{1}{2}}[m,n+\frac{1}{2},p-\frac{1}{2}] - H_z^{q+\frac{1}{2}}[m+\frac{1}{2},n+\frac{1}{2},p] + H_z^{q+\frac{1}{2}}[m-\frac{1}{2},n+\frac{1}{2},p] \\ H_y^{q+\frac{1}{2}}[m+\frac{1}{2},n,p+\frac{1}{2}] - H_y^{q+\frac{1}{2}}[m-\frac{1}{2},n,p+\frac{1}{2}] - H_x^{q+\frac{1}{2}}[m,n+\frac{1}{2},p+\frac{1}{2}] + H_x^{q+\frac{1}{2}}[m,n-\frac{1}{2},p+\frac{1}{2}] \end{Bmatrix}. \tag{7}$$

The $\mathbf{U} \times \mathbf{B}$ cross-product in (3) introduces coupling between $\mathbf{U}$ components; thus to maintain consistency, the equation must be rearranged using matrices $\mathbf{R}$ (10) and $\mathbf{S}$ (9) to give an update equation for $\mathbf{U}^{q+\frac{1}{2}}$ in terms of past values $\mathbf{U}^{q}$, shown in (8). In this expression, ω_{ca} refers to the cyclotron frequency of plasma species a.

Four-node spatial averaging is required when no node of a particular type exists at the desired location of the Yee cell. For example, the $E_x[m,\, n+\frac{1}{2},\, p]$ value required in (8) does not exist as a predefined node, but can be formed by spatial averaging of the four surrounding E_x nodes equidistant from the desired reference point, as shown below

$$E_x\left[m,\, n+\frac{1}{2},\, p\right] = \frac{1}{4}\left(E_x\left[m+\frac{1}{2},\, n,\, p\right] + E_x\left[m+\frac{1}{2},\, n+1,\, p\right] + E_x\left[m-\frac{1}{2},\, n,\, p\right] + E_x\left[m-\frac{1}{2},\, n+1,\, p\right]\right). \tag{15}$$

This treatment is applied to all cases where the value of a variable is required at a grid position where no update node exists. Collisions are treated using an effective collision frequency term $\nu_a = \nu_{an} + \sum_b \nu_{ab}$ for each species, which can be set independently at each node. This expression encompasses collisions with neutral particles through ν_{an} and collisions with the other plasma species present in the simulation through the ν_{ab} terms. In reality, the effective collision frequency will vary with time due to changes in particle temperature and density; however, for the purposes of this work, it is kept constant after initialization. The grad term coupling the particle fluid velocity to the temperature and density $\boldsymbol{\nabla}(k_B N_a T_a)$ in (3) is evaluated by finite difference approximation about the points $(m+\frac{1}{2},\, n,\, p)$, $(m,\, n+\frac{1}{2},\, p)$, and $(m,\, n,\, p+\frac{1}{2})$ for the U_x, U_y, and U_z update equations, respectively. Further spatial averaging is required to perform this operation. The complete $\mathbf{U}$ node update equations are shown in (11) (as shown at the bottom of the page). In this expression, the notation $[R]^{ij}$ refers to the value located at the ith row and jth column of matrix $\mathbf{R}$.

4) Update Equations for Density and Temperature: To make the scheme suitable for studying the response of a magnetized plasma to an incident EM wave, e.g., in the simulation of an ionospheric modification experiment, the FDTD techniques

$$\begin{aligned}
\begin{Bmatrix} U_x^{q+1/2}[m+\frac{1}{2},n,p] \\ U_y^{q+1/2}[m,n+\frac{1}{2},p] \\ U_z^{q+1/2}[m,n,p+\frac{1}{2}] \end{Bmatrix} &= \begin{Bmatrix} [R_a]^{00}\, U_x^{q-1/2}[m+\frac{1}{2},n,p] + [R_a]^{01}\, U_y^{q-1/2}[m+\frac{1}{2},n,p] + [R_a]^{02}\, U_z^{q-1/2}[m+\frac{1}{2},n,p] \\ [R_a]^{10}\, U_x^{q-1/2}[m,n+\frac{1}{2},p] + [R_a]^{11}\, U_y^{q-1/2}[m,n+\frac{1}{2},p] + [R_a]^{12}\, U_z^{q-1/2}[m,n+\frac{1}{2},p] \\ [R_a]^{20}\, U_x^{q-1/2}[m,n,p+\frac{1}{2}] + [R_a]^{21}\, U_y^{q-1/2}[m,n,p+\frac{1}{2}] + [R_a]^{22}\, U_z^{q-1/2}[m,n,p+\frac{1}{2}] \end{Bmatrix} + \dots \\
&+ \left(\frac{k_B \triangle_t}{m_a \triangle_x}\right)\left[\begin{Bmatrix} \left[S_a^{-1}\right]^{00}\left(T_a^q[m+1,n,p] - T_a^q[m,n,p]\right) + \left[S_a^{-1}\right]^{01}\left(T_a^q[m,n+\frac{1}{2},p] - T_a^q[m,n-\frac{1}{2},p]\right) + \dots \\ \left[S_a^{-1}\right]^{10}\left(T_a^q[m+\frac{1}{2},n,p] - T_a^q[m-\frac{1}{2},n,p]\right) + \left[S_a^{-1}\right]^{11}\left(T_a^q[m,n+1,p] - T_a^q[m,n,p]\right) + \dots \\ \left[S_a^{-1}\right]^{20}\left(T_a^q[m+\frac{1}{2},n,p] - T_a^q[m-\frac{1}{2},n,p]\right) + \left[S_a^{-1}\right]^{21}\left(T_a^q[m,n+\frac{1}{2},p] - T_a^q[m,n,p-\frac{1}{2}]\right) + \dots \\ + \left[S_a^{-1}\right]^{02}\left(T_a^q[m,n,p+\frac{1}{2}] - T_a^q[m,n,p-\frac{1}{2}]\right) \\ + \left[S_a^{-1}\right]^{12}\left(T_a^q[m,n,p+\frac{1}{2}] - T_a^q[m,n,p-\frac{1}{2}]\right) \\ + \left[S_a^{-1}\right]^{22}\left(T_a^q[m,n,p+1] - T_a^q[m,n,p]\right) \end{Bmatrix} + \begin{Bmatrix} \frac{T_a^q[m+1/2,n,p]}{N_a^q[m+1/2,n,p]} \\ \frac{T_a^q[m,n+1/2,p]}{N_a^q[m,n+1/2,p]} \\ \frac{T_a^q[m,n,p+1/2]}{N_a^q[m,n,p+1/2]} \end{Bmatrix}^T \cdot \begin{Bmatrix} \left[S_a^{-1}\right]^{00}\left(N_a^q[m+1,n,p] - N_a^q[m,n,p]\right) + \dots \\ \left[S_a^{-1}\right]^{10}\left(N_a^q[m+\frac{1}{2},n,p] - N_a^q[m-\frac{1}{2},n,p]\right) + \dots \\ \left[S_a^{-1}\right]^{20}\left(N_a^q[m+\frac{1}{2},n,p] - N_a^q[m-\frac{1}{2},n,p]\right) + \dots \\ + \left[S_a^{-1}\right]^{01}\left(N_a^q[m,n+\frac{1}{2},p] - N_a^q[m,n-\frac{1}{2},p]\right) + \left[S_a^{-1}\right]^{02}\left(N_a^q[m,n,p+\frac{1}{2}] - N_a^q[m,n,p-\frac{1}{2}]\right) \\ + \left[S_a^{-1}\right]^{11}\left(N_a^q[m,n+1,p] - N_a^q[m,n,p]\right) + \left[S_a^{-1}\right]^{12}\left(N_a^q[m,n,p+\frac{1}{2}] - N_a^q[m,n,p-\frac{1}{2}]\right) \\ + \left[S_a^{-1}\right]^{21}\left(N_a^q[m,n+\frac{1}{2},p] - N_a^q[m,n,p-\frac{1}{2}]\right) + \left[S_a^{-1}\right]^{22}\left(N_a^q[m,n,p+1] - N_a^q[m,n,p]\right) \end{Bmatrix}\right] \\
&+ \left(\frac{e_a \triangle_t}{m}\right)\begin{Bmatrix} \left[S_a^{-1}\right]^{00} E_x^q[m+\frac{1}{2},n,p] + \left[S_a^{-1}\right]^{01} E_y^q[m+\frac{1}{2},n,p] + \left[S_a^{-1}\right]^{02} E_z^q[m+\frac{1}{2},n,p] \\ \left[S_a^{-1}\right]^{10} E_x^q[m,n+\frac{1}{2},p] + \left[S_a^{-1}\right]^{11} E_y^q[m,n+\frac{1}{2},p] + \left[S_a^{-1}\right]^{12} E_z^q[m,n+\frac{1}{2},p] \\ \left[S_a^{-1}\right]^{20} E_x^q[m,n,p+\frac{1}{2}] + \left[S_a^{-1}\right]^{21} E_y^q[m,n,p+\frac{1}{2}] + \left[S_a^{-1}\right]^{22} E_z^q[m,n,p+\frac{1}{2}] \end{Bmatrix}
\end{aligned} \tag{11}$$

$$\tilde{N}_a^{q+1}[m,n,p] = \left(\frac{1 - \frac{\triangle_t}{2}\nabla\cdot\mathbf{U}_a^{q+\frac{1}{2}}}{1 + \frac{\triangle_t}{2}\nabla\cdot\mathbf{U}_a^{q+\frac{1}{2}}}\right)\tilde{N}_a^q[m,n,p] - \frac{\triangle_t}{1 + \frac{\triangle_t}{2}\nabla\cdot\mathbf{U}_a^{q+\frac{1}{2}}}\left(N_{a0}[m,n,p]\nabla\cdot\mathbf{U}_a^{q+\frac{1}{2}} + \mathbf{U}_a^{q+\frac{1}{2}}\cdot\nabla\left(N_{a0}\right) - \triangle N_a\right) \tag{12}$$

where

$$\begin{aligned}
\nabla\cdot\mathbf{U}_a^{q+\frac{1}{2}} &= \frac{1}{\triangle_x}\left(U_x^{q+\frac{1}{2}}[m+\frac{1}{2},n,p] - U_x^{q+\frac{1}{2}}[m-\frac{1}{2},n,p] + U_y^{q+\frac{1}{2}}[m,n+\frac{1}{2},p] - U_y^{q+\frac{1}{2}}[m,n-\frac{1}{2},p]\right. \\
&\qquad \left. + U_z^{q+\frac{1}{2}}[m,n,p+\frac{1}{2}] - U_z^{q+\frac{1}{2}}[m,n,p-\frac{1}{2}]\right) \\
\mathbf{U}_a^{q+\frac{1}{2}}\cdot\nabla\left(N_{a0}\right) &= \frac{1}{2\triangle_x}\left\{U_x^{q+\frac{1}{2}}[m,n,p]\left(N_{a0}[m+1,n,p] - N_{a0}[m-1,n,p]\right)\right. \\
&\left. + U_y^{q+\frac{1}{2}}[m,n,p]\left(N_{a0}[m,n+1,p] - N_{a0}[m,n-1,p]\right) + U_z^{q+\frac{1}{2}}[m,n,p]\left(N_{a0}[m,n,p+1] - N_{a0}[m,n,p-1]\right)\right\}.
\end{aligned}$$

described in Sections II-B1–II-B3 above must be augmented with new time-explicit update equations to simulate changes to the plasma density and temperature. Including on-grid updates for perturbations to the plasma medium in the time-stepping algorithm allows nonlinear plasma processes such as the self-focusing instability to be simulated. A standard first-order linearization technique such as that described in [21] is used to express N and T as combinations of a constant background part and a small time-dependent perturbation part such that $N(t) = N_0 + \tilde{N}(t)$ and $T(t) = T_0 + \tilde{T}(t)$. Equations (4) and (5) are reformed using the linearized fields, such that they become expressions describing the time-dependent behavior of the perturbed parts only. Update equation formation then proceeds via finite difference approximation about time step $q + \frac{1}{2}$ and point $(m,\, n,\, p)$. Averaging between nodes is required to ensure consistent evaluation of the grad and div terms in each expression.

A variety of forms can be used for the heat flux transport $\nabla \cdot \mathbf{Q}_a$ and collisional heating $\triangle\varepsilon_a$ terms in (5); however, care must be taken as not all valid expressions are amenable to the finite-differencing scheme used here. As this model has been developed for use in simulation of wave–plasma interactions in the ionosphere, an elastic expression [20] is used for the collisional term, with $\triangle\varepsilon_a = -\frac{N_a m_a \nu_a}{m_a + m_0}[3(T_0 - T_a) + m_0 U_a^2]$, where m_0 and T_0 represent the mass and temperature of background species. A further term taking into account in elastic collisions $\frac{3}{2} N_e R(T_e - T_0)$ is included in the electron temperature update equation, where R is the heat loss per electron to the background species. A simple heat flux density expression of $\mathbf{Q}_a = \kappa_a k_B \nabla T_a$ is used, with thermal conductivity κ_a taken to be $\frac{N_a k_B T_a}{m_a \nu_a}$. As both collisional and heat transport terms are functions of T, these must be temporally averaged and the update equation rearranged to give an expression for T^{q+1} only.

The final form of the update equations for the density and temperature is shown by (12) at the bottom of the previous page and (13) at the bottom of the page, respectively.

5) Full Update Algorithm: The complete set of update (6), (7), and (11)–(13) naturally lend themselves to a leapfrog time-stepping scheme, following the cyclical update pattern: $\mathbf{E}^q \to T^q \to \mathbf{H}^{q+\frac{1}{2}} \to \mathbf{U}^{q+\frac{1}{2}} \to N^{q+1} \to \mathbf{E}^{q+1} \to \cdots$. Special processes such as source injectors or boundary conditions can be added into this cycle at the appropriate points to complete the update algorithm.

C. Stability and Accuracy

An important consideration in any FDTD scheme is the stability of the model. In an unstable simulation, small numerical artifacts may grow rapidly with time to the extent that they may alter or obscure the simulation results. Stability is enforced by the Courant condition, which limits the ratio between the temporal and spatial discrete steps in the simulation. For a 3-D grid simulating free space, this condition is given by [2]

$$c\triangle_t \leq \frac{1}{\sqrt{\triangle_x^2 + \triangle_y^2 + \triangle_z^2}}. \tag{16}$$

This condition effectively ensures that energy in the simulation is not able to propagate through more than one grid cell per time step. It has been shown previously that in similar explicit calculation systems, with $\mathbf{E}$ and $\mathbf{U}$ nodes located at the same spatial points on the Yee grid and $\mathbf{H}$ and $\mathbf{U}$ nodes collocated in time, that the Courant condition is dependent not only on the grid parameters but also on the plasma medium contained within the simulation domain [13]. For an unmagnetized, collisionless plasma, the stability condition is given by (17) [13], which implies a further restriction on the discrete time step $\omega_p \triangle_t < 2$

$$c\triangle_t \leq \sqrt{\frac{1 - \left(\frac{\omega_p \triangle_t}{2}\right)^2}{\triangle_x^2 + \triangle_y^2 + \triangle_z^2}}. \tag{17}$$

$$\tilde{T}_a^{q+1}[m,n,p] = \frac{1}{1 - \frac{A}{2}} \left\{ \left(1 + \frac{A}{2}\right) \tilde{T}_a^q[m,n,p] + AT_{a0}[m,n,p] + \frac{2\triangle_t}{3k_B} \left(e_a \mathbf{E}^{q+\frac{1}{2}} \cdot \mathbf{U}_a^{q+\frac{1}{2}} + \frac{\triangle\varepsilon_a^{q+\frac{1}{2}} - \nabla \cdot \mathbf{Q}_a^{q+\frac{1}{2}}}{N_{a0}[m,n,p] + \tilde{N}_a^{q+\frac{1}{2}}[m,n,p]} \right) \right\} \tag{13}$$

where

$$A = \triangle_t \nabla \cdot \mathbf{U}_a^{q+\frac{1}{2}} + \triangle_t \left\{ \frac{\mathbf{U}_a^{q+\frac{1}{2}} \cdot \nabla \left(N_{a0} + \tilde{N}_a^{q+\frac{1}{2}} \right) - \triangle N_a}{N_{a0}[m,n,p] + \tilde{N}_a^{q+\frac{1}{2}}[m,n,p]} \right\}$$

$$\mathbf{E}^{q+\frac{1}{2}} \cdot \mathbf{U}_a^{q+\frac{1}{2}} = E_x^{q+\frac{1}{2}}[m,n,p] U_x^{q+\frac{1}{2}}[m,n,p] + E_y^{q+\frac{1}{2}}[m,n,p] U_y^{q+\frac{1}{2}}[m,n,p] + E_z^{q+\frac{1}{2}}[m,n,p] U_z^{q+\frac{1}{2}}[m,n,p]$$

$$\mathbf{U}_a \cdot \nabla \left(N_{a0} + \tilde{N}_a^{q+\frac{1}{2}} \right) = \frac{1}{2\triangle_x} \left\{ U_x^{q+\frac{1}{2}}[m,n,p] \left(N_{a0}[m+1,n,p] - N_{0a}[m-1,n,p] + \tilde{N}_a^{q+\frac{1}{2}}[m+1,n,p] - \tilde{N}_a^{q+\frac{1}{2}}[m-1,n,p] \right) \right.$$

$$+ U_y^{q+\frac{1}{2}}[m,n,p] \left(N_{a0}[m,n+1,p] - N_{0a}[m,n-1,p] + \tilde{N}_a^{q+\frac{1}{2}}[m,n+1,p] - \tilde{N}_a^{q+\frac{1}{2}}[m,n-1,p] \right)$$

$$\left. + U_z^{q+\frac{1}{2}}[m,n,p] \left(N_{a0}[m,n,p+1] - N_{0a}[m,n,p-1] + \tilde{N}_a^{q+\frac{1}{2}}[m,n,p+1] - \tilde{N}_a^{q+\frac{1}{2}}[m,n,p-1] \right) \right\}.$$

The effect of collisions or magnetically introduced anisotropy on stability is more difficult to quantify; however, it has been shown by [6] that direct integration FDTD schemes are stable at the unmagnetized Courant limit for all nonzero values of $\nu_c \triangle_t$. The applicability of the nonmagnetized limit to collisional and anisotropic situations has been tested numerically by [4] for a time-implicit implementation, and is further verified for the time-explicit scheme presented here by the validation tests described in Section III.

The FDTD technique is inherently approximate, so it is important to be able to assess the accuracy of a particular scheme. The discrete nature of the FDTD grid introduces both a numerical phase error to signals traveling through the computational domain (numerical dispersion) and an energy dissipation error (numerical dissipation) which must be accounted for. Both quantities are heavily dependent on the choice of discretization parameters used in the FDTD grid. The accuracy of the FDTD scheme for a particular set of grid parameters can be gauged using a dispersion analysis of the type presented in [6]. This approach is equivalent to a standard dispersion calculation for a wave propagating through plasma in continuous space, but with the assumed plane-wave form of the constituent wave fields substituted for a numerical world equivalent that takes into account the discrete nature of the grid

$$e^{-i(\omega t - k_x x - k_y y - k_z z)} \rightarrow e^{-i(\omega q \triangle_t - k_x m \triangle_x - k_y n \triangle_y - k_z p \triangle_z)}. \tag{20}$$

This has the effect of transforming the continuous-space partial differential operator Fourier pairs $\frac{\partial}{\partial t} \rightarrow i\omega$ and $\nabla \rightarrow -i\mathbf{k}$ into their discrete-space analogues

$$\frac{\partial}{\partial t} \rightarrow i \left(\frac{2}{\triangle_t} \right) \sin \left(\frac{\omega \triangle_t}{2} \right) \tag{21}$$

$$\nabla \rightarrow -i \left(\frac{2}{\triangle_j} \right) \sin \left(\frac{k_j \triangle_j}{2} \right) \hat{\mathbf{j}} \quad \text{for } j = x, y, z. \tag{22}$$

It can be seen from (21) and (22) that as the discrete steps approach zero, the continuous-space expressions are recovered. By applying relations (21) and (22) to (1) and (2), time-harmonic versions of Maxwell's wave equations can be formed for plane waves propagating in the numerical domain

$$-\left(\frac{2}{\triangle_j} \right) \sin \left(\frac{k_j \triangle_j}{2} \right) \hat{\mathbf{j}} \times \mathbf{E} = -\mu_0 \left(\frac{2}{\triangle_t} \right) \sin \left(\frac{\omega \triangle_t}{2} \right) \mathbf{H} \tag{23}$$

$$-i \left(\frac{2}{\triangle_j} \right) \sin \left(\frac{k_j \triangle_j}{2} \right) \hat{\mathbf{j}} \times \mathbf{H} - \sum_a N_a e_a \mathbf{U}_a = i\varepsilon_0 \left(\frac{2}{\triangle_t} \right) \sin \left(\frac{\omega \triangle_t}{2} \right) \mathbf{E}. \tag{24}$$

An expression for the frequency-dependent refractive index in the numerical domain can then be derived starting from (23) and (24), with the $\mathbf{U}$ term in (24) eliminated via substitution of a numerical time-harmonic version of (3)

$$i \left(\frac{2}{\triangle_t} \right) \sin \left(\frac{\omega \triangle_t}{2} \right) \mathbf{U}_a = \frac{e_a}{m_a} (\mathbf{E} + \mathbf{U}_a \times \mathbf{B}) - \nu_a \mathbf{U}_a. \tag{25}$$

Note that for this calculation, homogeneous plasma density and temperature has been assumed as the particle sound speed typically found in an ionospheric plasma is significantly less than the EM wave propagation velocity ($\frac{c_s}{c} << 1$), and thus the inclusion of finite temperature or density inhomogeneities adds a negligibly small perturbation to the EM wave dispersion relation. A static magnetic field in the z-direction has bee assumed, with wave propagation constrained to occur in the xz-plane. From here, the derivation follows the standard refractive index calculation for oblique plane wave propagation in a magnetized, collisional plasma as can be found in [22] and others, and leads to the expression shown in (18) at the bottom of the page. This is similar to the familiar Appelton–Hartree equation for oblique wave propagation, but with the continuous-space frequency replaced by the numerical equivalent Ω, given in (19) (as shown at the bottom of the page). In this expression, ω_p represents the fundamental plasma frequency, ω_c is the cyclotron frequency, and θ is the angle between the external B-field and the direction of wave propagation.

To assess the accuracy of the FDTD scheme, the frequency dependence of the discrete-space refractive index was calculated for a range of discretization regimes. Medium parameters were fixed, such that $\omega_p = 1.78 \times 10^7$ rad s^{-1}, $\omega_c = 8.18 \times 10^6$ rad s^{-1}, and $\nu_c = 500$ s^{-1} for an electron-only plasma. The magnetic field direction was set to be $\theta = \frac{\pi}{4}$ from the z-axis in the xz-plane, with wave propagation taken to be along the z-direction. The Courant number was chosen to be $S_c = \frac{c \triangle_t}{\triangle_x} = 0.5$ to ensure stability, with spatial step sizes set to be equal in each grid direction ($\triangle_x = \triangle_y = \triangle_z$). Discretization was varied by changing the size of discrete time step $\triangle_t$ (changing this parameter also scaled the discrete spatial step size through the relationship with fixed S_c). The equivalent refractive index for nondiscrete continuous space was calculated from (18) and (19) in the limit $\triangle_t, \triangle_{x,y,z} \rightarrow 0$. In this limit, $\Omega \rightarrow \omega$ and (18) reverts to the familiar Appelton–Hartree equation. The error between the numerical refractive index curve and its continuous-space counterpart is then calculated using

$$error(\omega) = \left| \frac{n_{continuous}(\omega) - n_{numerical}(\omega)}{n_{continuous}(\omega)} \right|. \tag{26}$$

By considering the real part of the refractive index, the numerical dispersion error introduced by the discrete nature

$$n^2 = \frac{c^2 K^2}{\Omega^2} = 1 - \frac{\left(\frac{\omega_p}{\Omega} \right)^2}{1 - i \left(\frac{\nu_c}{\Omega} \right) + \frac{\frac{\omega_c^2 \sin^2 \theta}{2\left(\Omega^2 - \omega_p^2 - i\nu_c \Omega\right)}}{\pm \left[\left\{ \frac{\omega_c^2 \sin^2 \theta}{2\left(\Omega^2 - \omega_p^2 - i\nu_c \Omega\right)} \right\}^2 + \left(\frac{\omega_p}{\Omega} \right)^2 \cos^2 \theta \right]^{\frac{1}{2}}}} \tag{18}$$

$$\Omega = \frac{2}{\delta_t} \sin \left(\frac{\omega \delta_t}{2} \right). \tag{19}$$

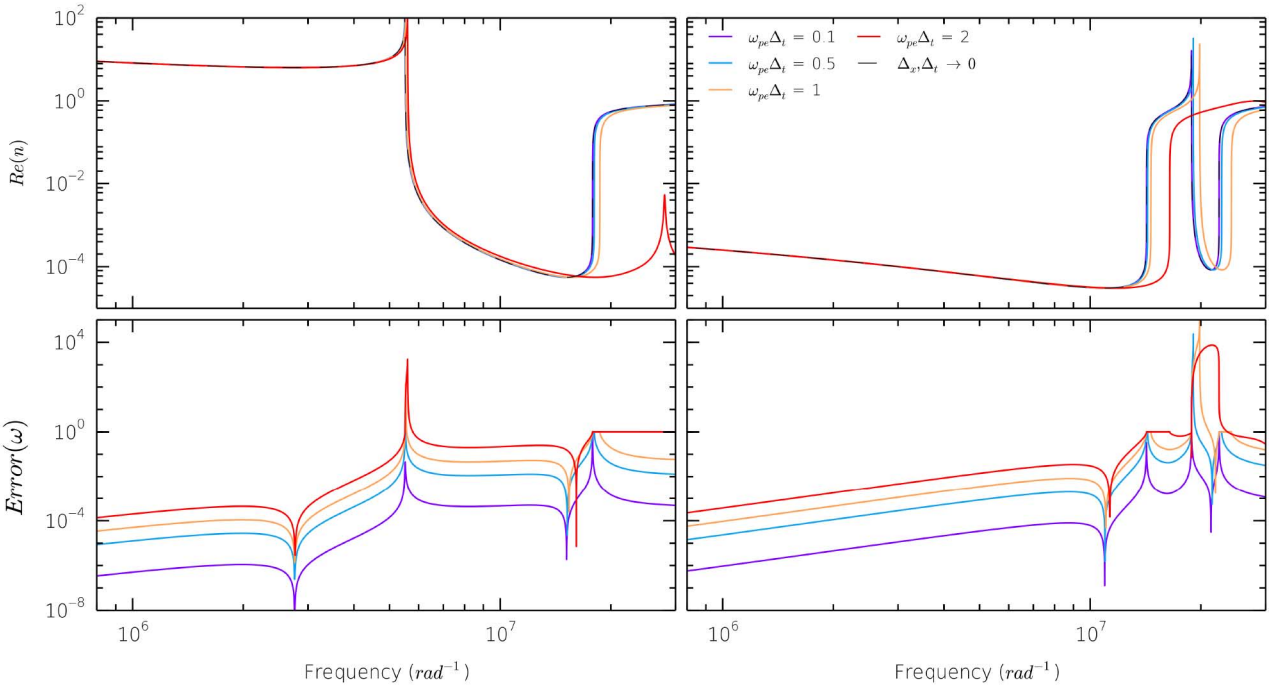

Fig. 2. Dispersion curves (upper) and relative errors when compared to the continuous-world regime (lower) for a range of dimensionless parameter $\omega_p \triangle_t$ shown for the ordinary mode (*left*) and extraordinary mode (*right*) branches of (18). Positive root shown only.

of the grid can be evaluated. Considering the imaginary part allows the energy dissipation error to be evaluated. Fig. 2 shows the continuous-space dispersion curves for the ordinary-mode (O-mode; left panels) and extraordinary-mode (X-mode; right panels) branches of (18) compared to curves calculated for different values of dimensionless parameter $\omega_p \triangle_t$. The lower panels show the relative errors calculated using (26). Fig. 3 shows the equivalent dissipation curves. As would be expected, deviation from the continuous-space case decreases as the discrete step sizes are reduced. The greatest source of error in all cases is in the frequency at which cutoffs and resonances occur, which shift to higher frequencies as the discretization becomes more coarse. As $\omega_p \triangle_t$ is increased to 2, the numerical dispersion and dissipation relationships break down completely and bear very little resemblance to the desired curves, as predicted by the stability requirement $\omega_p \triangle_t < 2$ introduced above.

Grid parameter choice for this FDTD scheme is ultimately constrained by the need for stability and the the requirement for a simulation to sample features of a desired spatial or temporal resolution. Beyond this, discrete step size choice must be a compromise between the accuracy needs of a given simulation and the computational resource required to run it, which would be expected to increase drastically with accuracy. For example, the simulation of an RF wave interaction with a kilometer-scale plasma feature requires the simulation time step to be small enough (of the order 10^{-8} s), such that the wave period can be well resolved; however, this automatically scales the spatial step size to be small (of the order 10 m) to maintain stability, meaning that a large computational grid is required to model the desired feature. A larger grid necessarily means that more memory is required and more calculations must be performed, thus increasing runtime.

D. Computational Performance

Among the prime considerations when developing the FDTD code was that simulation runtimes be kept to a minimum. As has been explored by [6], the direct integration method of FDTD used here is under most circumstances the most computationally efficient of finite-difference schemes, with smaller equivalent-simulation runtimes than the recursive convolution and exponential fitting methods it was tested against.

As the FDTD method scales particularly well with the single-instruction multiple-data (SIMD) concept of parallel processing, huge benefits in performance (often several orders of magnitude) were encountered when the code was accelerated using GPU technology as has been reported by [23]. The multiprocessor architecture of the GPU allows similar-type nodes to be updated simultaneously while still following the overall update algorithm described in Section II, vastly reducing processing time when compared to a serial implementation. A major advantage of the use of GPU technology was that almost all calculations and memory operations occur on-chip, with the only (comparatively slow) communication with the host machine occurring at initialization when memory arrays and input data are loaded onto the chip, and at the points where the code performs a data dump.

The FDTD code was written using the OpenCL parallel processing language, allowing it to be easily ported between heterogeneous devices of different hardware configurations. In the OpenCL implementation, the GPU receives computation instructions from the host CPU in the form of precompiled kernel functions. Before execution of a kernel, the computational grid is divided into work items (equivalent to CUDA threads), with each work item handling the update of an individual node in the grid. Due to the data-parallel properties of

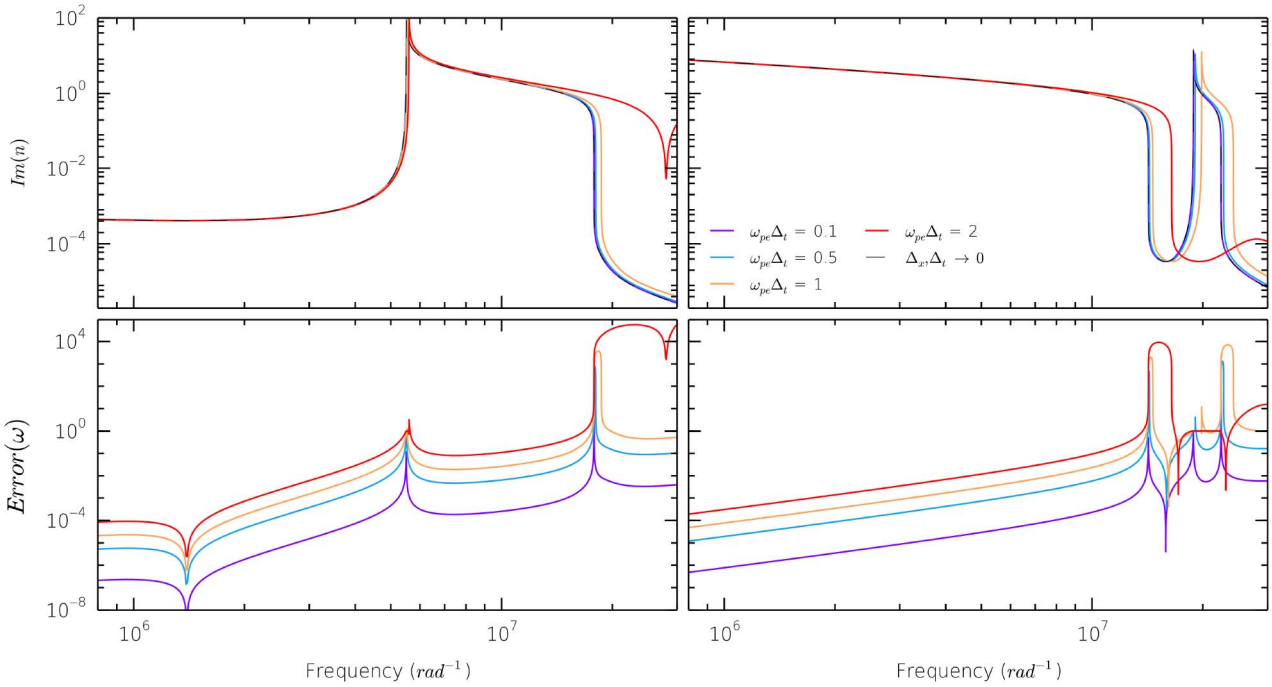

Fig. 3. Dissipation curves (upper) and relative errors when compared to the continuous-world regime (lower) for a range of dimensionless parameter $\omega_p \triangle_t$ shown for the ordinary mode (*left*) and extraordinary mode (*right*) branches of (18). Positive root shown only.

OpenCL, each work item can simultaneously execute the same arithmetic instruction on different parts of the data stored in the GPU memory. The grid can also be subdivided into discrete blocks of work items with user-defined dimensions, known as work groups. On kernel execution, work groups to be processed are distributed among the available device compute units by an on-chip instruction unit. Each compute unit is a multicore processor made up of many discrete processing elements capable of performing simple arithmetic operations. Once a work group has been assigned to a compute unit, the processing elements simultaneously execute the same kernel instruction on wavefronts of 32 consecutively indexed work items. For the FDTD code described here, synchronous kernel execution was used to maintain consistency between nodes, meaning that all compute units on the device must finish executing a particular kernel before moving on to the next set of instructions in the queue.

The choice of work group dimensions is therefore of critical importance to code performance; after division of the grid, there must be a sufficient number of work groups to fully occupy all compute units, and each must be large enough to ensure that all processing elements are in operation as close to 100% of the time as possible. Wavefronts are executed concurrently, and thus a high number of active wavefronts are required to mask delays due to thread stalling or memory latency. Additionally, members of a work group have shared access to the fast-access scratch memory (OpenCL local memory) available on the GPU chip, which can significantly reduce memory transaction times and thus speed up performance, but is limited in size compared to chip global memory.

To test the impact of changing the number of work groups on FDTD code performance, a simple simulation was repeated for varying ratio of the total number of work groups to device

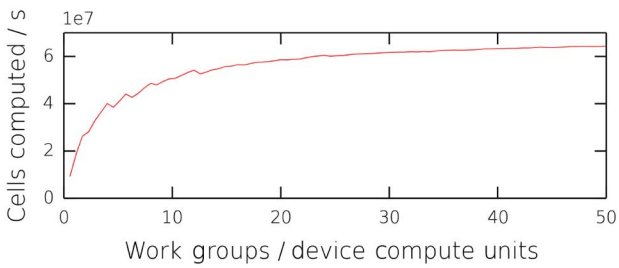

Fig. 4. Code performance variation with the number of work groups per device compute unit, for a constant work group size.

compute units, with work group size kept constant. The tests were performed on a Nvidia Tesla M2075 GPU containing a total of 14 compute units, with the number of FDTD grid cells fully advanced by one time step per wall-clock second used as a measure of performance. The results are shown in Fig. 4, which demonstrates that code performance increases significantly as the ratio of work groups to compute units is increased. Performance can be seen to continue to increase far beyond a ratio of one work group per compute unit due to the fact that each compute unit can handle many work groups simultaneously, and increasing the workload of each compute unit ensures that the processing elements are always occupied. Saturation in this case likely occurs due to the limit on the number of active work items imposed by the limited number of private memory registers available to each compute unit.

The compute unit executes work groups in units of 32-work-item wavefronts; therefore, a work group size of less than 32 will leave processing units idle and reduce performance. Likewise, a work group that is not made up of an integer number of warps will reduce processing core occupancy. Fig. 5 shows

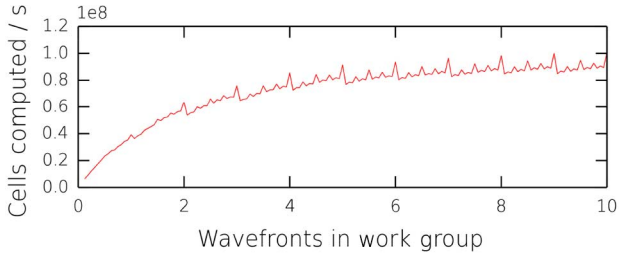

Fig. 5. Code performance variation with the number of wavefronts in a work group, for a constant total number of work groups.

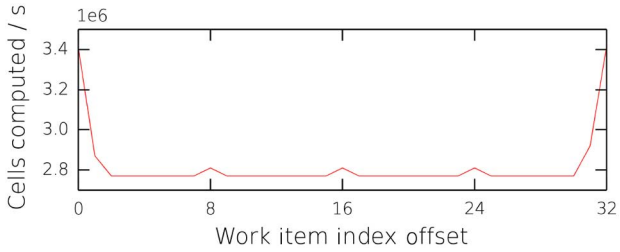

Fig. 6. Code performance for varying offsets of the z-direction work item index. Offset multiples of 32 correspond to coalesced memory access.

how the performance of the FDTD code varies with the number of wavefronts in a work group, for a constant total number of work groups. Performance peaks are seen at integer numbers of wavefronts per work group, as in these scenarios, there are no partially complete wavefronts. There is a general increase in performance due to increased occupancy of the compute units with greater block sizes.

Further performance gains were achieved by the use of a memory coalescence model such as that described by [24]. In this structure, kernel execution is organized such that all work items in a half-wavefront simultaneously access data from a contiguous block of memory, allowing the GPU to combine multiple memory calls into a single transaction and greatly reducing the net memory access time per work item. Without memory coalescence, multiple memory fetches may be required to access the same data, resulting in a significantly lower effective memory bandwidth. This is shown by Fig. 6, which shows the FDTD code performance for varying offsets of the z-direction work item index. Adding an offset to the index causes the work item grid to be out of alignment with the data stored in page-aligned global memory, resulting in uncoalesced memory access. Increasing the offset to 32 (one whole wavefront out of alignment) brings each wavefront back into phase with the global memory and performance returns to coalesced-memory levels.

Using several GPU nodes together in a networked cluster as described by [25] or [26] was found to increase the performance by a factor approximately equal to the number of GPUs available, with the only major bottleneck being the communication times between nodes. In this scenario, the computational domain was split between each GPU, such that each device was responsible for updating a different region of the grid and an OpenMPI framework used to initialize and pass commands to each GPU on separate threads. To ensure consistency, the regions of the grid updated by each GPU were set to overlap slightly, and the overlap regions passed between chips as boundary conditions. Increasing the size of these shared overlap regions meant that data transfer between chips did not have to occur every time step, but did introduce an additional overhead as it increased the number of redundant calculations (since the overlap regions of the grid are effectively updated twice—once by each chip that shares this region). It was found that an overlap region depth of 32 cells produced the optimal results; however, this is heavily dependent on the hardware available, and will likely vary between systems. Table I shows how the performance is increased when a serial implementation of the FDTD code running on an Intel Core i5-3750 CPU is compared to a parallel implementation running on a Nvidia Tesla M2075 GPU. The number of nodes fully advanced by one time step per wall-clock second is used as a measure of performance. Also shown is the performance of a dual-GPU implementation, which shows a gain factor of approximately 2 over the single-GPU implementation. Due to hardware limitations, an implementation using more than two GPUs could not be tested; however, performance would be expected to scale approximately linearly with the number of GPUs used, with the only additional overheads arising from the transfer of overlap regions between devices.

TABLE I
TABLE COMPARING THE PERFORMANCE OF A SERIAL IMPLEMENTATION OF THE FDTD CODE RUNNING ON A SINGLE CPU WITH PARALLEL IMPLEMENTATIONS RUNNING ON A SINGLE GPU AND ON TWO NETWORKED GPUS

	Computed nodes / second		
Grid cells	CPU	Single GPU	Dual GPU
1024	1.37×10^6	2.04×10^6	1.06×10^6
4096	1.64×10^6	8.38×10^6	4.21×10^6
16 384	2.00×10^6	3.34×10^7	1.72×10^7
65 536	1.63×10^6	6.62×10^7	6.83×10^7
102 400	1.80×10^6	8.00×10^7	1.00×10^8
262 144	2.13×10^6	7.99×10^7	1.27×10^8
589 824	2.36×10^6	9.29×10^7	1.40×10^8
1 048 576	2.38×10^6	8.19×10^7	1.59×10^8

III. VALIDATION

A. Wave Propagation Through Homogeneous Plasma

To test the validity of the FDTD algorithm, the model was used to simulate the propagation of a plane-polarized EM wave through a homogeneous block of plasma, in a similar manner to the numerical experiments used to validate the FDTD algorithm in [16]. Several test cases involving different homogeneous plasma regimes were considered. In each case, the results were benchmarked against those obtained using Tech-X's VORPAL 4.2.2 software package [19]. The grid was initialized with discrete spatial steps of $\triangle_x = \triangle_y = \triangle_z = 11.626\,\mathrm{m}$ and a discrete step size of $\triangle_t = 1.939 \times 10^{-8}\,\mathrm{s}$ chosen to give a Courant number of 0.5, ensuring stability. Discretization parameters were chosen to represent those used in a typical ionospheric simulation, with accuracy balanced by a need for computational efficiency. The simulation boundaries were terminated using second-order absorbing boundary conditions of

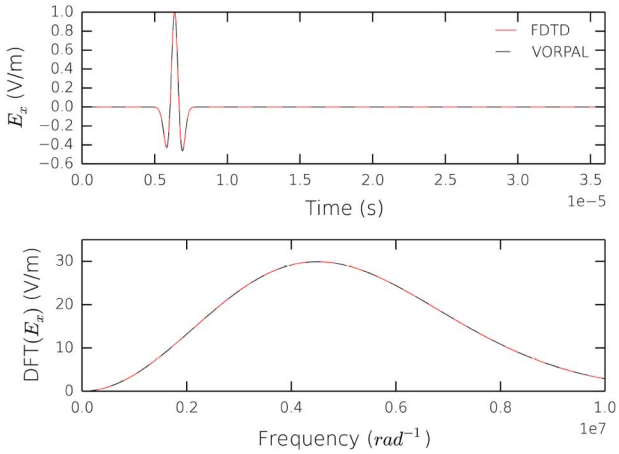

Fig. 7. E_x signal for an EM pulse of form (27) propagating through a free-space simulation, recorded at a point 128 cells from the $z = 0$ launch plane. Upper panel shows time-domain comparison of signals measured using the FDTD algorithm described in this work (red) and an equivalent VORPAL simulation (black). Lower panel shows the frequency domain form of the signals, which peak at $\omega_{peak} = 4.55 \times 10^6$ rad s^{-1}.

the type described by Mur [27], which allowed outgoing waves to be absorbed with minimal numerical reflection. The boundaries of the VORPAL comparison simulation were terminated using absorbing boundary conditions, and a particle-in-cell approach implementing a Boris push update to particle velocities [28] used to mediate any interaction between the simulated EM field and plasma. A linearly polarized input pulse in the form of a twice-differentiated Gaussian described by (27) was introduced into the computational domain from the lower z-boundary, and allowed to propagate through the domain in the positive z-direction

$$E_x(q) = \left[1 - \frac{1}{2}\left(\omega_{peak} q \triangle_t - 1\right)^2\right] exp\left[-\left(\frac{\omega_{peak} q \triangle_t}{2\pi} - 1\right)^2\right]. \tag{27}$$

This form of pulse was particularly suitable as the frequency of peak amplitude could be set easily using the w_{peak} parameter, and it avoided the DC frequency component found in undifferentiated Gaussian signals. For the following tests, the peak frequency of the input pulse was set to be $\omega_{peak} = 4.55 \times 10^6$ rad s^{-1}, corresponding to a wavelength of 36 cells. The free space time-domain and frequency-domain signals from this pulse measured at a point 128 cells from the source plane are shown in Fig. 7. In this simple free-space example, the FDTD model was in almost perfect agreement with the benchmark, with the greatest error between signals of order 10^{-4} V/m.

The pulse was then introduced into a spatially homogeneous unmagnetized electron-only plasma medium of electron number density 2×10^9 m^{-3}. Figs. 8 and 9 show the time- and frequency-domain signals, respectively, again recorded at a point 128 cells from the source plane. Signals from both the FDTD code and VORPAL are shown, alongside the expected result from plasma theory. The time-domain waveform shows a distorted pulse with a long tail oscillating at the electron plasma frequency. This agrees with expected behavior, as in a dispersive plasma, different frequency components should propagate

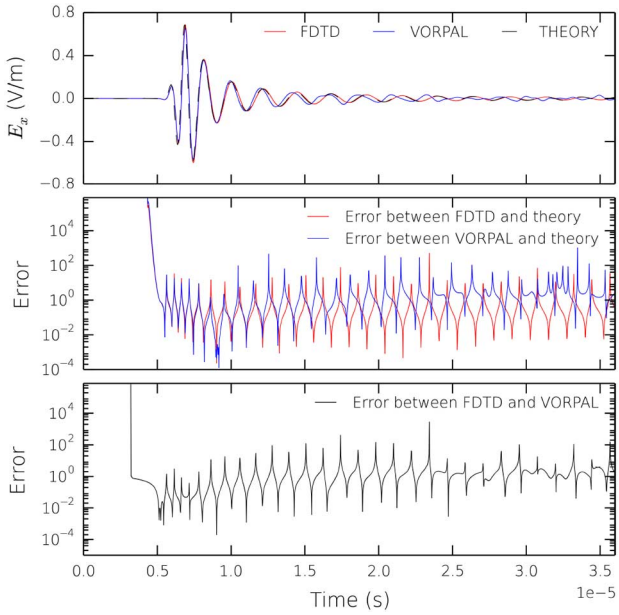

Fig. 8. Time-domain E_x signal for pulse propagating through an unmagnetized plasma simulation. Upper panel shows FDTD signal (red), VORPAL signal (blue), and the expected result from plasma theory (black). Central panel shows the error between the simulated signals and the predictions of plasma theory. Lower panel shows the error between FDTD and VORPAL signals.

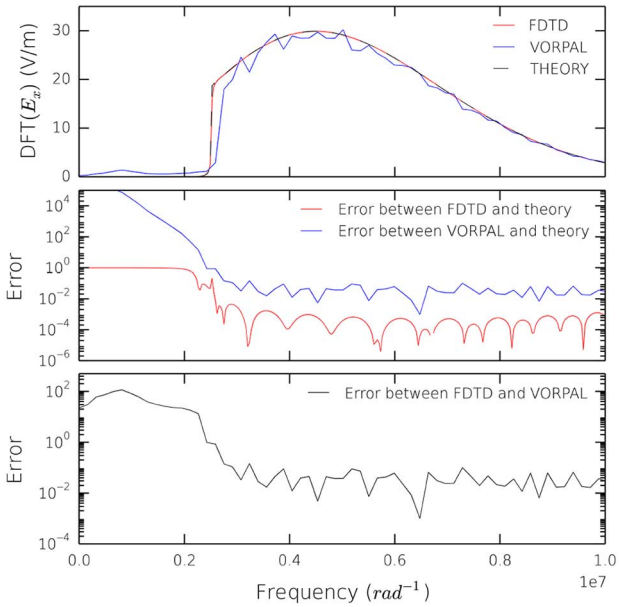

Fig. 9. Frequency domain E_x signal for pulse propagating through an unmagnetized plasma simulation. Upper panel shows discrete Fourier transform of FDTD signal (red), VORPAL signal (blue), and the expected result from plasma theory (black). A clear cutoff can be seen at simulation plasma frequency $\omega_p = 2.5 \times 10^6$ rad s^{-1}. Central panel shows the error between the simulated signals and the predictions of plasma theory. Lower panel shows the error between FDTD and VORPAL signals.

through the plasma at different velocities, causing the observed distortion. The slow tail is caused by frequency components on or close to the electron plasma frequency which propagate with very low group velocities. The FDTD algorithm results

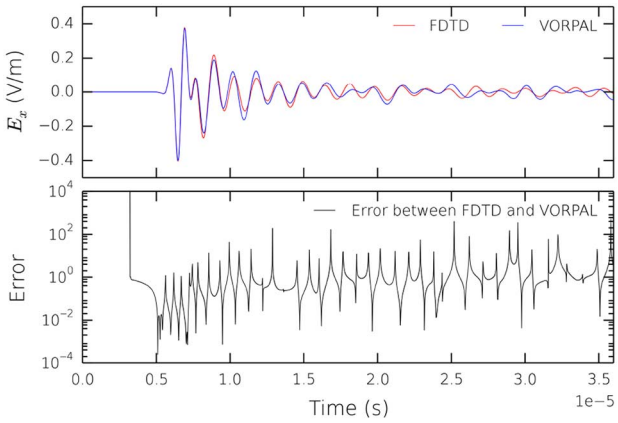

Fig. 10. Time-domain E_x signal for pulse propagating through a magnetized plasma simulation. Upper panel shows FDTD signal (red) and VORPAL signal (blue). Lower panel shows the error between FDTD and VORPAL signals.

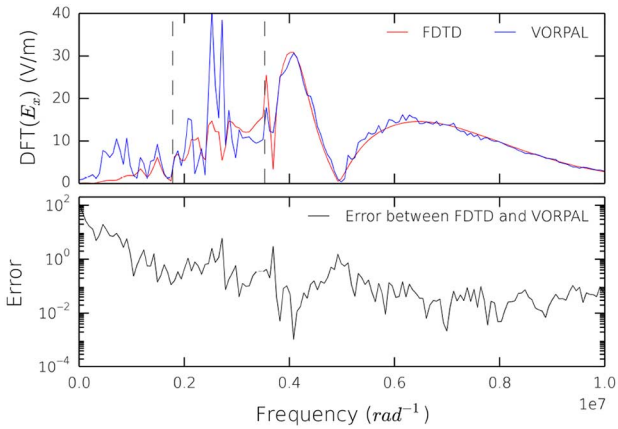

Fig. 11. Frequency domain E_x signal for pulse propagating through a magnetized plasma simulation. Upper panel shows discrete Fourier transform of FDTD signal (red) and VORPAL signal (blue). Clear cutoffs can be seen due to the different propagation characteristics of the right- and left-hand circularly polarized components. The expected positions of the cutoffs are indicated in black. Lower panel shows the error between FDTD and VORPAL signals.

demonstrate good agreement with both the benchmark and theory, with substantial error between the FDTD and benchmark signals only creeping into the trace during later time tail. This difference is likely due to the different levels of numerical dispersion present in each simulation. The FDTD result was found to match very closely to the theoretical prediction at all times sampled. The frequency-domain waveform clearly shows a cutoff at the electron plasma frequency $\omega_p = 2.52 \times 10^6$ rad s^{-1}. This agrees well with expectations, as no frequency components below the fundamental plasma frequency should be able to propagate through a plasma of this type. Note that this frequency is the numerical plasma frequency, calculated using the methods of Section II-C above, and depends not only on the plasma density but also on the choice of grid discretization parameters.

A static magnetic field of magnitude $\mathbf{B} = 10\hat{\mathbf{z}}\,\mu T$ was then applied to the medium, fixing the electron cyclotron frequency in the simulation to be $\omega_c = 1.76 \times 10^6$ rad s^{-1}. The magnetic field was directed parallel to the direction of pulse propagation. In this situation, the initially linearly polarized pulse should decompose into left-handed and right-handed circularly polarized components as it propagates through the magnetized plasma. The different components propagate at different phase velocities and experience different cutoff frequencies below which they will not propagate, given by

$$\omega_{LHC} = \frac{1}{2}\left(\left[\omega_c^2 + 4\omega_p^2\right]^{\frac{1}{2}} - \omega_c\right) = 1.79 \times 10^6 \text{ rad s}^{-1} \tag{28}$$

$$\omega_{RHC} = \frac{1}{2}\left(\left[\omega_c^2 + 4\omega_p^2\right]^{\frac{1}{2}} + \omega_c\right) = 3.55 \times 10^6 \text{ rad s}^{-1}. \tag{29}$$

The time- and frequency-domain signals for this case are shown in Figs. 10 and 11. The time-domain signal agrees well with the benchmark until, as encountered in the previous case, a small error creeps into the low-amplitude tail. Again, this is likely due to the different numerical dispersion regimes. The predicted cutoffs can clearly be seen in the frequency-domain signal. There are substantial differences between the FDTD frequency-domain signal and the benchmark at lower frequencies; however, this is mostly due to a resonance encountered in the kinetic PIC code that is not developed in the fluid-only FDTD scheme. Despite this, the qualitative features and position of the cutoffs are present and in good agreement in each trace. In all the above tests, the FDTD code was found to complete the simulation considerably more quickly than the benchmark software, with the single-core CPU version of VORPAL found to perform on average 4.7×10^5 node updates per second compared to a rate of 2.4×10^6 node updates per second achieved by the FDTD code on the same hardware. The GPU accelerated version of the FDTD code was able to run the simulations almost two orders of magnitude faster than this with a node update rate of 1.1×10^8 nodes s^{-1} achieved using a single-GPU implementation. A GPU enabled version of VORPAL was not available for comparison.

B. Wave Propagation Through Inhomogeneous Plasma

This section describes further validation tests that are more representative of the intended end-use of the model. Simple numerical experiments were performed demonstrating the ability of the FDTD algorithm to replicate some of the key features that are required for realistic simulation of RF wave propagation in the ionosphere. The model was set up to simulate a simple ionospheric heating experiment, with a continuous harmonic input wave at a frequency of $\omega_0 = 2.7 \times 10^7$ rad s^{-1} introduced into a domain containing a linear electron density profile given by (30), where N_{crit} is the critical density at which the plasma frequency equals the frequency of the incident wave, z_{crit} is the height at which this density occurs, and L_z is the scale size of the gradient. The variable z refers to the distance from the lower edge of the simulation domain. For these tests, z_{crit} was set to be at 29.8 km from the lower edge of the simulation and L_z set to be 60 km

$$N_0(z) = N_{crit}\left(1 + \frac{z - z_{crit}}{L_z}\right). \tag{30}$$

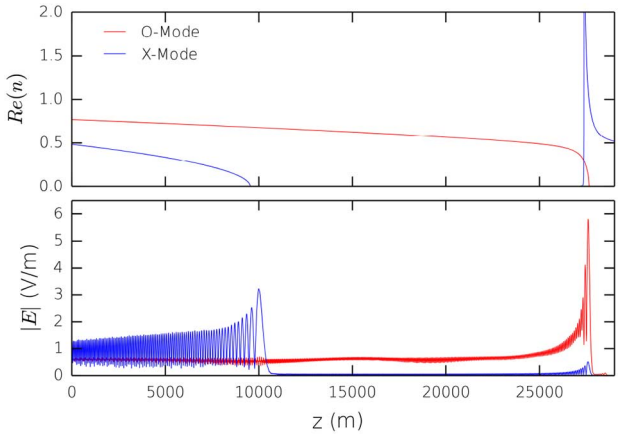

Fig. 12. Upper panel shows the numerical refractive index curves for a magnetized plasma of density profile (30). Lower panel shows a comparison of the time-averaged O- and X-mode E-field amplitudes measured along the central axis of the computational domain.

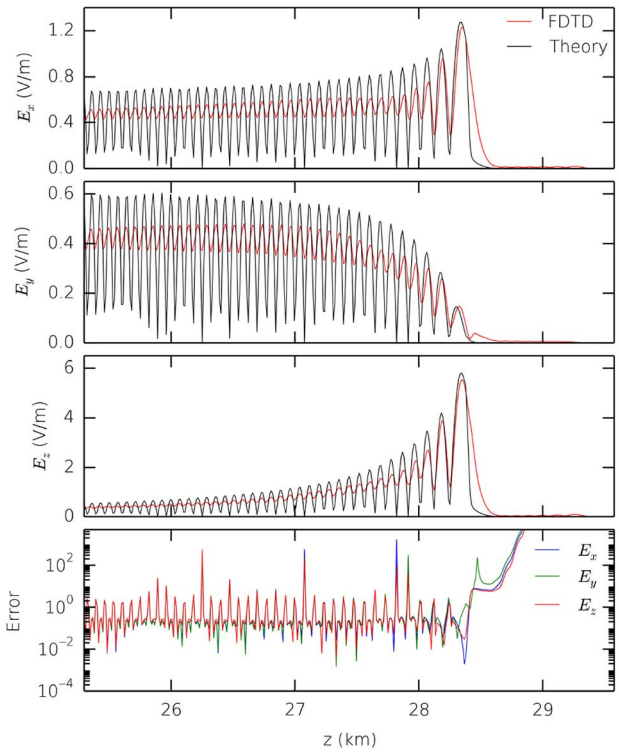

Fig. 13. Comparison between standing wave pattern developed in FDTD simulation and theoretical calculation following the method of [31].

The simulation was initialized using the discretization parameters used in Section III-A and run in a 2-D configuration in which all y-direction gradients were assumed to be zero. A static background magnetic field of magnitude $4.65 \times 10^{-5}\,T$ at an angle of 12° to the z-axis was applied to mimic the geomagnetic field. Background electron temperature was set to 1500 K and electron-neutral collision frequency to $500\,s^{-1}$. Domain boundaries were terminated by 64-cell complex-frequency-shifted perfectly matched layers [29], which allowed outgoing waves at a range of oblique angles to be absorbed effectively with minimal numerical reflection. The input wave was given an angular spread to appear as if it originated form a point located 220 km below the $z = 0$ launch edge. The polarization of the wave could be set at launch to represent either the O-mode or X-mode using the polarization relations given in [30] for a cold, collisional, magnetized plasma. The FDTD model was used to check that the propagation of these modes proceeded in the expected manner. The upper panel of Fig. 12 shows how the refractive index for O- and X-mode waves in this simulation varies with altitude. The curves shown are the numerical forms of the refractive index curves specific to the chosen discretization parameters, calculated as described in Section II-C. The traces in the lower panel of Fig. 12 show the time-averaged electric field amplitude measured along the central axis of the computational domain after 2×10^6 time steps for both the O- and X-mode input waves. The results demonstrate that the simulation is able to replicate the propagation characteristics of each mode successfully. The O-mode wave was able to propagate almost the whole length of the simulation domain before being reflected at z_{crit} as expected from the dispersion curve. The X-mode wave was attenuated at a distance of around 10 km from the lower edge of the simulation, the point at which the theory predicts that this wave should become evanescent. A small amount of X-mode amplitude appears to have leaked beyond this point and can be seen to have been amplified at the resonance just above z_{crit}. The standing wave pattern that develops below the reflection height in the O-mode case is a well-understood phenomenon which has been characterized by [31]. Fig. 13 shows a comparison between the wave pattern developed in the simulation and that predicted by theoretical calculation. Note that the numerical form of the the refractive index has been used in calculating the theoretical result. The results from the FDTD simulation are in good agreement with those from theory and successfully reproduce the expected wave pattern for all E-field components. The wave amplitudes in the simulated case are lower than the predicted values by a factor of ~ 0.8; however, this is likely due to energy leaking out of the grid through absorption at the grid boundaries.

IV. Conclusion

This paper describes the formulation of a 3-D FDTD scheme suitable for the simulation of RF wave propagation in a dynamic, magnetized plasma. This work builds on the well-established FDTD technique by extending it to include a time-explicit update of the plasma fluid temperature and density, allowing the time-dependent behavior of the plasma medium to be studied. The stability criteria and accuracy of the FDTD scheme were analyzed and found to be heavily dependent on the choice of spatial and temporal discrete step parameters. Computational efficiency is crucial for a numerical simulation of this type and it was found that accelerating the FDTD code using GPU technology yielded significantly better performance, with a dual-GPU implementation achieving a rate of node update almost two orders of magnitude faster than a serial implementation. Optimization techniques such as memory coalescence were shown to have a significant effect on code performance, and it was demonstrated that large performance gains could be achieved through careful choice of the

GPU work group dimensions. Numerical validation tests simulating EM propagation through a range of plasma regimes demonstrate that this model agrees well with plasma theory and the benchmark software results. In particular, the propagation characteristics of waves of ordinary- and extraordinary-mode polarizations in an inhomogeneous, anisotropic plasma were accurately replicated. In the case of the ordinary-mode wave, the amplitude swelling effects produced around the critical density demonstrate that the FDTD scheme is able to successfully simulate the mode-conversion process responsible for this phenomenon. Both these effects are crucial to the ongoing and planned work involving this model, which concern the numerical simulation of an artificial ionospheric modification experiment and the study of the resulting nonlinear processes. The FDTD code was found to run the validation test simulations considerably more quickly than the benchmark software using equivalent hardware.

ACKNOWLEDGMENT

The authors would like to thank the staff at the Lancaster University High-End Cluster and J. Smith at Tech-X U.K. Ltd. for their technical assistance.

REFERENCES

[1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propag.*, vol. 14, no. 3, pp. 302–307, May 1966.
[2] A. Taflove and S. C. Hagness, *Computational Electrodynamics*, 2nd ed. Norwood, MA, USA: Artech House, 2000.
[3] J. J. Simpson, "An established numerical method applied to geophysics," *Eos Trans. Amer. Geophys. Union*, vol. 93, no. 29, pp. 265–266, 1996.
[4] W. Hu and S. Cummer, "An FDTD model for low and high altitude lightning-generated EM fields," *IEEE Trans. Antennas Propag.*, vol. 54, no. 5, pp. 1513–1522, May 2006.
[5] J. J. Simpson, "On the possibility of high level transient coronal mass ejection-induced ionospheric current coupling to electric power grids," *J. Geophys. Res.*, vol. 116, no. 2, pp. 105–130, 2009.
[6] S. A. Cummer, "An analysis of new and existing FDTD methods for isotropic cold plasma and a method for improving their accuracy," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 392–400, Mar. 1997.
[7] R. M. Joseph, S. C. Hagness, and A. Taflove, "Direct time integration of Maxwell's equations in linear dispersive media with absorption for scattering and propagation of femtosecond electromagnetic pulses," *Opt. Lett.*, vol. 16, no. 18, pp. 1412–1414, 1991.
[8] R. J. Luebbers, F. Hunsberger, and K. S. Kunz, "A frequency-dependent finite-difference time-domain formulation for transient propagation in plasma," *IEEE Trans. Antennas Propag.*, vol. 39, no. 1, pp. 29–34, Jan. 1991.
[9] R. J. Luebbers and F. Hunsberger, "FDTD for Nth-order dispersive media," *IEEE Trans. Antennas Propag.*, vol. 40, no. 11, pp. 1297–1301, Nov. 1992.
[10] R. J. Hawkins and J. S. Kallman, "Linear electronic dispersion and finite-difference time-domain calculations: A simple approach," *J. Lightw. Technol.*, vol. 11, no. 11, pp. 1872–1874, Nov. 1993.
[11] J. L. Young and F. P. Brueckner, "A time domain numerical model of a warm plasma," *Radio Sci.*, vol. 29, no. 2, pp. 451–463, 1994.
[12] Q. Chen, M. Katsurai, and P. H. Aoyagi, "An FDTD formulation for dispersive media using a current density," *IEEE Trans. Antennas Propag.*, vol. 46, no. 11, pp. 1739–1746, Nov. 1998.
[13] J. L. Young, "A full finite difference time domain implementation for radio wave propagation in a plasma," *Radio Sci.*, vol. 29, no. 6, pp. 1513–1522, 1994.
[14] J. H. Lee and D. K. Kalluri, "Three-dimensional FDTD simulation of electromagnetic wave transformation in a dynamic inhomogeneous magnetized plasma," *IEEE Trans. Antennas Propag.*, vol. 47, no. 7, pp. 1146–1151, Jul. 1999.
[15] Y. Yu and J. J. Simpson, "An E-J collocated 3-D FDTD model of electromagnetic wave propagation in magnetized cold plasma," *IEEE Trans. Antennas Propag.*, vol. 58, no. 2, pp. 469–478, Feb. 2010.
[16] Y. Yu, J. Niu, and J. J. Simpson, "A 3-D global earth-ionosphere FDTD model including an anisotropic magnetized plasma ionosphere," *IEEE Trans. Antennas Propag.*, vol. 60, no. 7, pp. 3246–3256, Jul. 2012.
[17] N. A. Gondarenko, P. N. Guzdar, S. L. Ossakow, and P. A. Bernhardt, "Linear mode conversion in inhomogeneous magnetized plasmas during ionospheric modification by HF radio waves," *J. Geophys. Res.*, vol. 108, no. A12, 2003.
[18] N. A. Gondarenko, S. L. Ossakow, and G. M. Milikh, "Generation and evolution of density irregularities due to self-focusing in ionospheric modifications," *J. Geophys. Res.*, vol. 110, no. A09304, 2005.
[19] C. Nieter and J. R. Cary, "VORPAL: A versatile plasma simulation code," *J. Comput. Phys.*, vol. 196, no. 2, pp. 448–473, 2004.
[20] T. R. Robinson, "The heating of the high-latitude ionosphere by high-power radio waves," *Phys. Rep.*, vol. 79, no. 2–3, pp. 79–209, 1989.
[21] U. S. Inan and M. Golkowski, *Principles of Plasma Physics for Engineers and Scientists*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[22] V. L. Ginzburg, *The Propagation of Electromagnetic Waves in Plasmas*, 2nd ed. New York, NY, USA: Pergamon, 1970.
[23] P. Sypekand, A. Dziekonski, and M. Mrozowski, "How to render FDTD computations more effective using a graphics accelerator," *IEEE Trans. Magn.*, vol. 45, no. 3, pp. 1324–1327, Mar. 2009.
[24] M. Livesey *et al.*, "Development of a CUDA implementation of the 3D FDTD method," *IEEE Antennas Propag. Mag.*, vol. 54, no. 5, pp. 186–195, Oct. 2012.
[25] T. Stefaski, S. Benkler, N. Chavannes, and N. Kuster, "OpenCL-based acceleration of the FDTD method in computational electromagnetics," *Int. J. Numer. Model.*, vol. 26, pp. 355–365, 2012.
[26] K.-H. Kim and Q.-H. Park, "Overlapping computation and communication of three-dimensional FDTD on a GPU cluster," *Comput. Phys. Commun.*, vol. 183, pp. 2364–2369, 2012.
[27] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations," *IEEE Trans. Electromagn. Compat.*, vol. 23, no. 4, pp. 377–382, Nov. 1981.
[28] J. P. Boris, "The acceleration calculation from a scalar potential," Plasma Phys. Lab., Princeton Univ., Princeton, NJ, USA, Tech. Rep. MATT-152, 1970.
[29] J. P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 114, pp. 185–200, 1994.
[30] K. G. Budden, *Radio Waves in the Ionosphere*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 1961.
[31] B. Lundberg and B. Thid, "Standing wave pattern of HF radio waves in the ionospheric reflection region (2)," *Radio Sci.*, vol. 21, no. 3, pp. 486–500, 1986.

Patrick D. Cannon received the M.S. degree in physics from the University of Oxford, St. Catherine's College, Oxford, U.K., in 2010. He is currently pursuing the Ph.D. degree in physics at Lancaster University, Lancaster, U.K.

His research interests include development of a high-performance FDTD code to investigate the nonlinear plasma processes that may occur when a radio-frequency electromagnetic wave interacts with the ionosphere.

Farideh Honary received the B.Sc. degree from the National University of Iran, Tehran, in 1978 and the Ph.D. degree in high-energy physics from the University of Kent, Canterbury, UK, in 1983.

She is a Professor of Radio Science and Space Plasma and Head of Space Plasma Environment and Radio Science (SPEARS) Research Group, Department of Physics, Lancaster University, Lancaster, U.K. Her research interests include investigating the complex behavior of energetic particles during magnetic storms and substorms with ground- and space-based instruments, utilizing the ionosphere as a natural plasma laboratory to study a variety of important nonlinear plasma processes associated with the interaction of high-power HF waves and the ionospheric plasma. This is done by employing the EISCAT high-power HF transmitter, and investigation of lunar dust charging and dynamics.