Progress In Electromagnetics Research, Vol. 137, 487–512, 2013

ELIMINATION OF NUMERICAL DISPERSION FROM ELECTROMAGNETIC TIME DOMAIN ANALYSIS BY USING RESOURCE EFFICIENT FINITE ELEMENT TECHNIQUE

S. M. Raiyan Kabir*, B. M. A. Rahman, Arti Agrawal, and Ken T. V. Grattan

City University London, London EC1V 0HB, United Kingdom

Abstract—Time domain analysis of electromagnetic wave propagation is required for design and characterization of many optical and microwave devices. The FDTD method is one of the most widely used time domain methods for analysing electromagnetic scattering and radiation problems. However, due to the use of the Finite Difference grid, this method suffers from higher numerical dispersion and inaccurate discretisation due to staircasing at slanted and curve edges. The Finite Element (FE)-based meshing technique can discretize the computational domain offering a better approximation even when using a small number of elements. Some of the FE-based approaches have considered either an implicit solution, higher order elements, the solution of a large matrix or matrix lumping, all of which require more time and memory to solve the same problem or reduce the accuracy. This paper presents a new FE-based method which uses a perforated mesh system to solve Maxwell's equations with linear elements. The perforated mesh reduces the requirement on memory and computational time to less than half of that compared to other FE-based methods. This paper also shows a very large improvement in the numerical dispersion over the FDTD method when the proposed method is used with an equilateral triangular mesh.

1. INTRODUCTION

The use of time domain analysis for electromagnetic radiation and scattering problems was first introduced by Yee in 1966 [1]. He proposed a finite difference algorithm which runs the initial value

Received 23 January 2013, Accepted 17 February 2013, Scheduled 4 March 2013

* Corresponding author: S. M. Raiyan Kabir (Raiyan.Kabir.1@city.ac.uk).

problem on a lattice of staggered field components known as Yee's lattice. This method, known as the Finite Difference Time Domain (FDTD) approach, is the most popular and widely available numerical method for electromagnetic time domain analysis [2–5]. The reasons for its popularity are, firstly, the simplicity of the algorithm which makes the development of a three-dimensional code easy. Secondly, the algorithm is data parallel, so a parallel implementation of the algorithm can be carried out on all computing platforms [6–9]. Thirdly, for each computational cell the FDTD uses the least possible computational resources to obtain the evolving fields.

The disadvantages of the FDTD algorithm are also well known. Firstly, with the use of a rectangular finite difference grid, it is difficult to model slanted or curved structures. Although the general perception is to decrease the cell size to achieve more accurate representation, decreasing the grid size does not always give predictable improvements. To minimize the error, a sub-pixel smoothing scheme can be added [10], which require modification of the real boundary conditions and use of an anisotropic technique to model devices when the actual device contains only isotropic materials. Secondly, the use of a rectangular grid with the FDTD introduces numerical dispersion in the computational domain.

The Finite Element (FE)-based approaches are better alternatives for the effective representation of an arbitrary shaped structure, such as one with slanted or curvilinear interfaces because it uses an unstructured polygonal mesh to represent the structure. The FEM was introduced to the electromagnetic analysis during the 1980's to solve frequency domain problems [11–13]. To represent the structure more accurately, researchers have considered the FEM for time domain analysis [5, 14–18]. Although these methods are sometime more accurate in structural representation, some of them may require an implicit solution of the computational domain for each time step [19], some require the solution of large matrices [5] and some require higher order solutions of Maxwell's equations [14, 17, 18].

However, among all the FE methods reported, the point-matched method [14] has features which may make it the more suitable. Firstly, it solves Maxwell's equations directly in the same manner as the FDTD. Secondly, it does not generate mass matrices and all calculations are local to each element. Lastly, the formulation is data parallel and suitable for parallel computing implementation. However, the downsides of this method presented in the work of Cangellaris [14] are the use of the rectangular grid. So, there is no significant advantage offered in the numerical dispersion. As the method uses four node rectangles as computational elements, it requires the solution of second

order shape functions. Hence, the method takes more memory and CPU time to analyze the same structure.

This paper presents a novel two-dimensional point matched technique using two linear triangle meshes. This paper reports the development of an FE based time domain technique which uses minimum possible computational resources (memory and CPU time) and reduce the numerical dispersion to a negligible value which makes the time domain simulation almost dispersion free.

2. DERIVATION OF THE METHOD

In this section, the derivation of the simplest possible governing equations from Maxwell's equations using FE discretization and linear shape functions is presented. The method will use FE meshes to discretize the computational domain and thus a more accurate representation of the computational structure can be made.

2.1. Maxwell's Equations

Maxwell's Equations for a source-free and isotropic region can be written as

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

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon}\nabla \times \mathbf{H} \tag{1b}$$

where, $\mathbf{H} = \hat{x}H_x + \hat{y}H_y + \hat{z}H_z$, $\mathbf{E} = \hat{x}E_x + \hat{y}E_y + \hat{z}E_z$, μ and ϵ are the vector magnetic field, the vector electric field, the permeability and the permittivity of the medium respectively with $\hat{x}$, $\hat{y}$ and $\hat{z}$ being the unit vectors in the x, y and z directions, respectively.

For the two-dimensional (2D) propagation in x-y plane, it can be assumed that $\partial/\partial z = 0$. In this case, two sets of equations (Eq. (2) and Eq. (3)) can be obtained for the propagation of Transverse Electric (TE) and Transverse Magnetic (TM) modes, respectively.

For TE Propagation

$$\frac{\partial H_x}{\partial t} = -\frac{1}{\mu}\frac{\partial E_z}{\partial y} \tag{2a}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\frac{\partial E_z}{\partial x} \tag{2b}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right) \tag{2c}$$

For TM Propagation

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\frac{\partial H_z}{\partial y} \tag{3a}$$

$$\frac{\partial E_y}{\partial t} = -\frac{1}{\epsilon}\frac{\partial H_z}{\partial x} \tag{3b}$$

$$\frac{\partial H_z}{\partial t} = -\frac{1}{\mu}\left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\right) \tag{3c}$$

2.2. Discretization

To apply the governing equations of Eq. (2) and Eq. (3), the computational domain has to be discretized. All the field components in these equations are functions of both space (x, y) and time (t). For all equations the left hand side of the equations calculate only the time evolution of the field and the right hand side of the equations calculate evolution in space separately. Therefore, the time evolution can be calculated with a time shape function (t is the variable of the function) at a fixed space node and the space evolution can be calculated at a fixed time node with a spatial shape function (x and y are the variable of the function).

2.2.1. Space Discretization

To discretize the computational domain both in space and time, nodal elements can be used. Linear shape functions can be used to describe the variation of the field inside an element. The distribution shape functions of all the field components can be written as

$$\Phi = \sum_{i=1}^{M} N_i \phi_i \tag{4}$$

where Φ can be any field component (any one of H_x, H_y, H_z, E_x, E_y, E_z) inside the element, ϕ_i is the field component at the ith node of the element (any one of h_x, h_y, h_z, e_x, e_y, e_z), M is the number of nodes. For a linear element (i.e., three node triangular elements) $M = 3$ and N_i is the shape function for the ith node. Linear shape functions can be expressed as

$$N_i = a_i + b_i x + c_i y \tag{5}$$

where, a_i, b_i and c_i are the coefficients of the equation of plane going through the nodes of the element. It should be mentioned here that, by using Eq. (4), shape function of any order can be incorporated with the proposed method. However, to store each higher order element

more computer memory space is required than a linear element. Each linear element also takes least computation time. So, linear elements were chosen for the space discretization.

2.2.2. Time Discretization

Similarly, the field components along the time axis can be discretized as

$$\Psi = \sum_{j=1}^{P} Q_j \psi^{(j)} \tag{6}$$

where Ψ can be any field component (any one of the H_x, H_y, H_z, E_x, E_y, E_z) inside the element, $\psi^{(j)}$ is the field component at jth time node, P is the number of the time node in the time element and for linear elements, $P = 2$. Here, Q_j is the shape function for the jth time node and for the linear shape function it can be expressed as

$$Q_j = p_j t + q_j \tag{7}$$

where p_j and q_j are the coefficients of the line passing through the nodes of the time element. Similar to the space discretization, Eq. (6) allows higher order time elements.

Both Eq. (4) and Eq. (6) can be applied to Eq. (2) and Eq. (3). For example, Eq. (2a) can be written as

$$\frac{\partial}{\partial t}\sum_{j=1}^{2} Q_j h_x^{(j)} = -\frac{1}{\mu}\frac{\partial}{\partial y}\sum_{i=1}^{3} N_i e_{zi} \Rightarrow \sum_{j=1}^{2}\frac{\partial Q_j}{\partial t} h_x^{(j)} = -\frac{1}{\mu}\sum_{i=1}^{3}\frac{\partial N_i}{\partial y} e_{zi}$$

$$\Rightarrow h_x^{(2)} = \frac{1}{\frac{\partial Q_2}{\partial t}}\left[-\frac{1}{\mu}\sum_{i=1}^{3}\frac{\partial N_i}{\partial y} e_{zi} - \frac{\partial Q_1}{\partial t} h_x^{(1)}\right]$$

Similarly, all the equations from Eq. (2) and Eq. (3) can be derived. Discretized versions of Eq. (2) and Eq. (3) are given in Eq. (8) and Eq. (9), respectively.

For TE Propagation

$$h_x^{(n+1)} = \frac{1}{\frac{\partial Q_2}{\partial t}}\left[-\frac{1}{\mu}\sum_{i=1}^{3}\frac{\partial N_i}{\partial y} e_{zi}^{(n)} - \frac{\partial Q_1}{\partial t} h_x^{(n-1)}\right] \tag{8a}$$

$$h_y^{(n+1)} = \frac{1}{\frac{\partial Q_2}{\partial t}}\left[\frac{1}{\mu}\sum_{i=1}^{3}\frac{\partial N_i}{\partial x} e_{zi}^{(n)} - \frac{\partial Q_1}{\partial t} h_y^{(n-1)}\right] \tag{8b}$$

$$e_z^{(n+1)} = \frac{1}{\frac{\partial Q_2}{\partial t}}\left[\frac{1}{\epsilon}\left(\sum_{i=1}^{3}\frac{\partial N_i}{\partial x} h_{yi}^{(n)} - \sum_{i=1}^{3}\frac{\partial N_i}{\partial y} h_{xi}^{(n)}\right) - \frac{\partial Q_1}{\partial t} e_z^{(n-1)}\right] \tag{8c}$$

For TM Propagation

$$e_x^{(n+1)} = \frac{1}{\frac{\partial Q_2}{\partial t}}\left[\frac{1}{\epsilon}\sum_{i=1}^{3}\frac{\partial N_i}{\partial y}h_{zi}^{(n)} - \frac{\partial Q_1}{\partial t}e_x^{(n-1)}\right] \tag{9a}$$

$$e_y^{(n+1)} = \frac{1}{\frac{\partial Q_2}{\partial t}}\left[-\frac{1}{\epsilon}\sum_{i=1}^{3}\frac{\partial N_i}{\partial x}h_{zi}^{(n)} - \frac{\partial Q_1}{\partial t}e_y^{(n-1)}\right] \tag{9b}$$

$$h_z^{(n+1)} = \frac{1}{\frac{\partial Q_2}{\partial t}}\left[-\frac{1}{\mu}\left(\sum_{i=1}^{3}\frac{\partial N_i}{\partial y}e_{yi}^{(n)} - \sum_{i=1}^{3}\frac{\partial N_i}{\partial y}e_{xi}^{(n)}\right) - \frac{\partial Q_1}{\partial t}h_z^{(n-1)}\right] \tag{9c}$$

where the field components with the $(n+1)$, (n) and $(n-1)$ superscripts are the future, current and the past values, respectively.

3. THE MESHES

The mesh is the most important part of all the FE-based methods. It allows discretization of an irregular shaped structure in a more accurate and efficient manner. The speed of an FE-based code largely depends on how efficiently the mesh discretizes the computational domain. Hence efficient meshing which reduces computational cells without sacrificing accuracy of the solution, is one of the key factors used to make a fast and efficient FE-based code.

3.1. The Space Mesh System

To use the linear shape functions, triangles with three nodes were considered to discretize the computational domain. This mesh can be termed the **"Main Mesh"**. For TE propagation, it maybe assumed that, all current e_z field components are stored at the nodes of the triangular mesh. Both h_x and h_y field components can be calculated from e_z using Eq. (8a) and Eq. (8b). Eq. (8a) calculates one future value of h_x using the current values of the e_z field components at the nodes of the triangular element. As there is only one value for the whole element, it cannot be stored in any specific node of the triangle, but instead it can be stored at the centroid of the triangle, which is unique. As a result, no value of h_x field component will be available at the corner nodes of the elements of main mesh. Similarly, for Eq. (8b) the calculated future value of h_y can be stored at the centroid. To obtain the next values of e_z from Eq. (8c), the current value of h_x and h_y are required which are placed at the centroids of the elements of main mesh. Hence, another triangular mesh is required which will be constructed using the centroids of the main mesh elements. This

new mesh can be termed the **"Auxiliary Mesh"**. As the h_x and h_y field components are stored at the auxiliary mesh, the elements of this mesh can be used to calculate the future e_z values which can be stored in the nodes of the main mesh, provided each element of the auxiliary mesh must surround one of the nodes of the main mesh. Similar arrangements can be made for TM propagation using Eq. (9).

To illustrate the meshing process, at first a simple square grid was considered. This grid can be converted into a triangular mesh by dividing each cell with one diagonal line. Fig. 1(a) shows a triangle mesh generated from a 4×4 square grid. As can be seen the resultant mesh is a **"Isosceles Right-angled Triangle (IRT) Mesh"**. The lower triangle of the square cell has been shaded in pink and the upper triangle is in white. The lower triangle is numbered with the cell index and the upper triangle is numbered with the cell index and additionally with a suffix "a". The black dot inside each triangle is the centroid of that triangle.

The auxiliary mesh has to be generated using the centroids of the main mesh. Each element of the auxiliary mesh has to surround one of the nodes of main mesh. The **Perfect Electric Conductor (PEC)** boundary condition will be applied at the boundaries of the main mesh to truncate the computational domain into a finite one. Therefore, calculation of field components of the boundary nodes of main mesh is not necessary. Centroids of the elements 2, 5 and 6 of the main mesh (Fig. 1(a)) are taken as the three nodes of the element number 1 of the auxiliary mesh (shaded in light blue) and the centroids of elements 1, 2 and 5 are taken for element 1b of the auxiliary mesh (white colored). Similarly, all the other elements were generated from the centroids of the main mesh. Fig. 1(b) shows the auxiliary mesh constructed from the centroids of the main mesh shown in Fig. 1(a). The centroids of

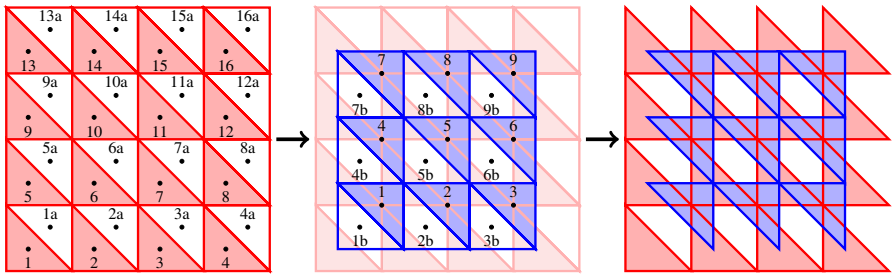


Figure 1. (a) The generating the linear mesh by dividing a square grid by diagonal line. (b) Generating the auxiliary mesh by connecting the centroids of the main mesh. (c) Discarding the unwanted elements from both mesh.

the auxiliary mesh are shown as black dots in Fig. 1(b).

It can be seen that the centroids of elements 1, 2, 3, ..., 9 of the auxiliary mesh (Fig. 1(b)) coincide with the node points of the main mesh, whereas the centroids of 1b, 2b, 3b, ..., 9b coincide with the centroids of 1a, 2a, 3a, ..., 11a of the main mesh. As a result, these elements cannot be used to calculate the field components at the nodes of the other mesh. Therefore, elements 1a, 2a, 3a, ... of the main mesh and 1b, 2b, 3b ... of the auxiliary mesh can be discarded from the respective meshes. By discarding unwanted elements, the number of elements in Fig. 1(c) will become half of Fig. 1(b). At this point, this unique coupled mesh system can be termed the **"Coupled Perforated Mesh System"**.

This perforated mesh system has a big advantage. It reduces the number of computational elements to less than half of the full mesh. As a result, the method proposed in this paper is twice faster than any other FEM method using full mesh with the same computational need per element. The memory requirement for the method will be less than an FEM approach using the full mesh discretization.

Although this technique removes half of the elements from both the meshes, it does not removed any of the nodes. As seen in Fig. 1(c), the main mesh elements surround their associated nodes on the auxiliary mesh and are used to update the value of the fields associated with the node and vice versa. Although the alternative elements (elements with suffix 'a' and 'b') have been removed from the mesh system, however, all of the three nodes associated with those triangle are still updated during the calculation of every time step. Therefore, the magnitude of the field components inside the perforated region can be calculated by using the space shape functions of that triangle (Eq. (5)).

As perforated meshing is a new concept, present day meshing libraries may not be optimized for perforated meshing.

3.2. The Time Mesh System

Along with the space domain, the time domain also needs to be discretized. This can be done using linear elements with two nodes. In Eq. (8a) and Eq. (8b), the future values of the h_x and the h_y field components may be calculated using the current e_z field components and the past h_x and h_y components respectively. Eq. (8c) calculates the future values of the e_z field components with the current h_x, h_y and the past e_z field components. Similar explanations can be given for Eq. (9). For all the equations, the current values of the **E** field components with the past values of the **H** field components are needed to calculate the

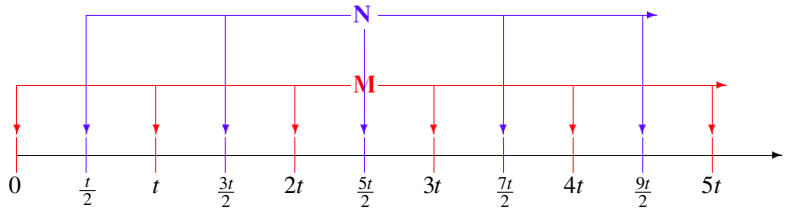


Figure 2. Arrangement of time mesh system for equal time spacing.

future values of **H** components and vice versa. Therefore, both the **E** and **H** field components cannot be calculated at the same time node.

For the TE propagation example given in Section 3.1, the simulation started with the calculation of the future h_x and h_y fields from the current e_z field components in the main mesh (with Eq. (8a) and Eq. (8b). Therefore, the first time node belongs to the e_z component associated with the main mesh. The future e_z field was calculated from the h_x and h_y fields in the auxiliary mesh. So, the second time node belongs to the h_x and h_y field components associated with the auxiliary mesh. This way the time domain can be divided into two time meshes **M** and **N** which are associated with the field components of the main mesh and the auxiliary mesh respectively. Fig. 2 shows the **M** and **N** meshes. In this example, the time step size for the calculation of the future e_z components from the previous e_z components is t. So, the e_z field was calculated at t, $2t$, $3t$, $4t$, ... and hence these time nodes belong to mesh **M** along with the initial e_z at time 0. The time step size for the calculation of the h_x and h_y field components have to be of the same duration, t. To calculate the e_z components, the current h_x and h_y components are required. Therefore, the h_x and h_y field components were calculated at $t/2$, $3t/2$, $5t/2$, $7t/2$, ... Thus, these time nodes belong to mesh **N**. Similar examples can be shown for TM propagation with Eq. (9).

4. RESULTS OF THE SIMULATIONS

A C++ code was developed to perform the numerical simulations. The implementation was made dimensionless or scale invariant [20] by taking the speed of light as $c = 1$. As a result permeability and permittivity of vacuum $\mu_0 = 1$ and $\epsilon_0 = 1$ respectively. This made the implementation dimensionless, scalable and for many problem reduces the affect of floating point errors. "Perfectly Matched Layers (PML)" [21, 22] were also implemented to perform long duration simulations in a minimal truncated computational domain. The outputs of the program were stored in the VTK file format to visualize

with Paraview software. All field plots in this paper were generated with Paraview.

4.1. Planar Waveguide

A Silicon core air clad planar waveguide was simulated at 1.55 μm wavelength to compare the field profile with the results from 1D FEM

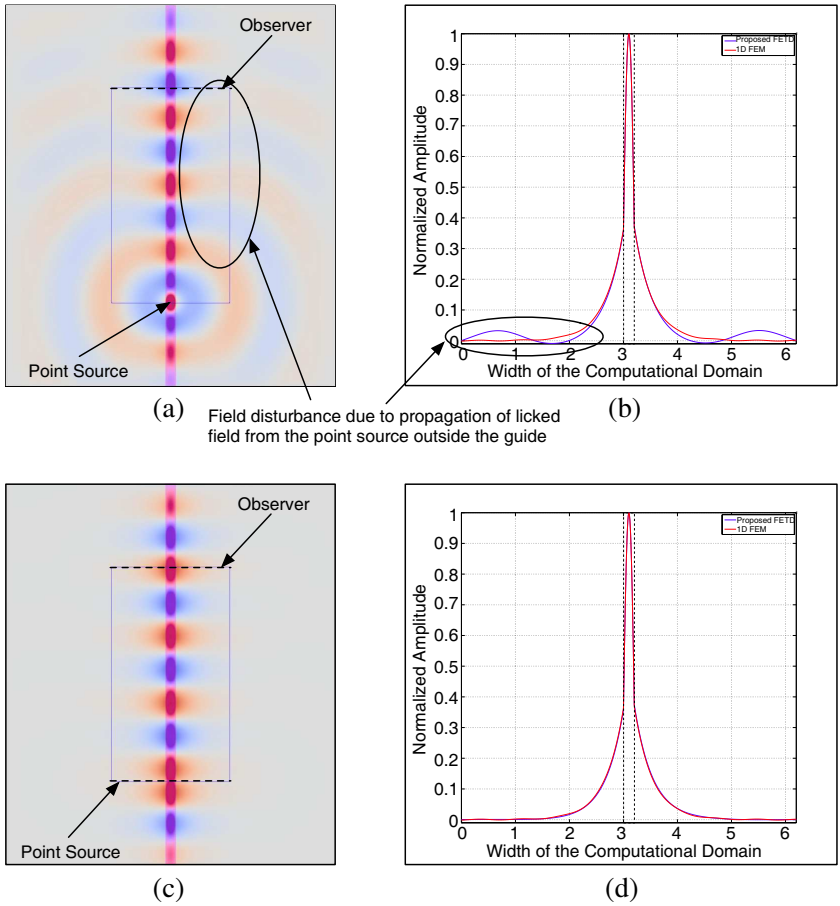


Figure 3. (a) E_z field profile for a dielectric planar waveguide with a E_z point source (red and blue parts are the positive and negative half cycles of the propagating wave). (b) Comparison of E_z field profile from the proposed FETD with the point source at the observer point with the mode profile. (c) E_z field profile for a dielectric planar waveguide excited with the E_z mode profile. (d) Comparison of E_z field profile from the proposed FETD and the mode profile.

mode solver [11, 12]. The width of the waveguide was taken to be 0.2 µm. At first, a TE simulation was performed with a point source at the middle of the waveguide. The computational domain was discretised with a mesh of resolution $\Delta = 50$ per unit length and time resolution, $\Delta t = \Delta/2$. Fig. 3(a) shows the E_z field profile after 5000 time steps. It can be observed, the E_z field is mostly confined inside the Silicon core. Some of the field outside the guide is radiating away from the guide. As in this case, a E_z point source was used to initiate the propagation. As shown in Fig. 3(a), an observing line was placed at the end of the guide before the PML boundary layer. The E_z field mode profile along the line was observed during the simulation. This was carried out to extract E_z field profile of the planar waveguide away from the point source. Fig. 3(b) shows comparison of the actual mode profile shown by red line and the observed field profile evolved from the point source. The field profile from the proposed method perfectly matches the mode profile in the core region, but shows some ripple in the air cladding. This ripple is due to the radiating field as shown earlier in Fig. 3(a).

Another simulation was performed by replacing the point source with a line source representing the mode profile. The results are presented in Fig. 3(c) and Fig. 3(d). As can be seen in Fig. 3(c), the structure is supporting the mode without any leakage and Fig. 3(d) shows the exact match of the mode profiles obtained from both the proposed FETD and the FEM mode solver. This also validates the accuracy of the newly developed approach using perforated meshes.

4.2. Metamaterial Flat Lens

The goal of this simulation was to show the flexibility and adaptability of the formulation. Here a dispersive Double Negative (DNG) metamaterial slab was simulated to show backward propagation of the wavefront inside and double focusing of the wave [23]. For this simulation, the Drude model [24] was added for both permittivity and permeability to the governing equations of the method. The Drude model equations are given as,

$$\epsilon(\omega) = \epsilon_0 \left(1 - \frac{\omega_{pe}^2}{\omega(\omega + j\gamma_e)}\right) \tag{10a}$$

$$\mu(\omega) = \mu_0 \left(1 - \frac{\omega_{pm}^2}{\omega(\omega + j\gamma_m)}\right) \tag{10b}$$

where, ω_{pe} and ω_{pm} are the plasma frequencies, and γ_e and γ_m are the collision frequencies.

For the simulation a rectangular computational domain surrounded with PML boundary layer was taken. A rectangular shaped metamaterial slab was placed at the centre of the domain. An E_z line source was placed above the slab parallel to its x-axis. Fig. 4 shows the result of the simulation. As shown, the wave generated from the line source approaches the metamaterial slab perpendicularly. Inside the slab the field becomes curved and focuses inside the slab. When the field comes out of the slab, again it focuses outside. The wavefront inside the guide moves in opposite direction to the direction of movement outside the guide. The wavelength used was 1.55 µm. The index of the metamaterial slab is approximately -1 at 1.55 µm.

5. NUMERICAL DISPERSION

For the numerical simulation, the computational domain has to be discretized. However, due to the discretisation, phase error can be introduced into the propagating plane wave. As a result, the speed of propagation may be slower than the actual speed of the wave. The phase lag with the actual wave increases with the length of its propagation in the computational domain. The issue with the phase error gets worse when the error varies with the direction of propagation. The result of this is different speeds of propagation in different directions, which is equivalent to an artificial anisotropy imposed on the wave by the discretisation even when the material is isotropic. This phenomenon is known as **"Numerical Dispersion"** [5] or **"Numeric Anisotropy"** [25]. This dispersion can be minimised by increasing the resolution of the discretization [26], but to do this would require more

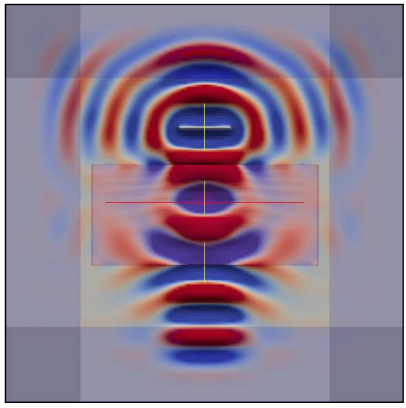


Figure 4. Forward wave outside the DNG slab and backward wave inside the slab. Double focusing of EM wave is visible.

memory and more computations.

The method proposed here can be very efficient with the discretization and can represent the structure accurately with fewer elements. However, even if the structure was accurately discretized with an efficient meshing algorithm, if the numerical dispersion of the output mesh remains high, an erroneous solution can be obtained for a longer propagation distance. To reduce the error, the resolution of the mesh needs to be increased. As a result, there will be little benefit in terms of memory usage and computational load. Therefore, to achieve maximum memory and computational efficiency, a mesh with a minimal numerical dispersion need to be considered.

5.1. The Numerical Dispersion Relation

For convenience of calculation, the matrix form of the equations of Eq. (8) and Eq. (9) will be used. For TE propagation, equations Eq. (8) can be written as,

$$\left\{h_{x\langle l\rangle}^{(n)}\right\}\left\{\frac{\partial Q^{(n)}}{\partial t}\right\}^T = -\frac{1}{\mu}\left\{e_{zk}^{[m]}\right\}\left\{\frac{\partial N_k}{\partial y}\right\}^T \tag{11a}$$

$$\left\{h_{y\langle l\rangle}^{(n)}\right\}\left\{\frac{\partial Q^{(n)}}{\partial t}\right\}^T = \frac{1}{\mu}\left\{e_{zk}^{[m]}\right\}\left\{\frac{\partial N_k}{\partial x}\right\}^T \tag{11b}$$

$$\left\{e_{z\langle k\rangle}^{(m)}\right\}\left\{\frac{\partial Q^{(m)}}{\partial t}\right\}^T = \frac{1}{\epsilon}\left(\left\{h_{yl}^{[n]}\right\}\left\{\frac{\partial N_l}{\partial x}\right\}^T - \left\{h_{xl}^{[n]}\right\}\left\{\frac{\partial N_l}{\partial y}\right\}^T\right) \tag{11c}$$

where, superscript T is the transpose operator to convert a row matrix into a column matrix and vice versa.

Section 3.1 introduced one main and one auxiliary mesh to hold the **E** and **H** field components. In Section 3.2, the time nodes were divided into sets **M** and **N** for the **E** and **H** fields, respectively. In Eq. (11), the field components for the space node of the main and the auxiliary meshes are denoted with k and l subscripts, respectively. For the time nodes, the field components for the members of **M** and **N** are denoted with (m) and (n) superscripts, respectively. The angle brackets $\langle\ \rangle$ are used to denote the centroid of the current element and the square brackets [ ] are used to denote the current time.

To study the numerical dispersion relation, a monochromatic source was assumed for the TE mode of propagation where E_z, H_x and H_y can be expressed as

$$h_{xl}^{(n)} = H_{x0}e^{j(\omega t^{(n)} - \tilde{\kappa_x}x_l - \tilde{\kappa_y}y_l)} \tag{12a}$$

$$h_{yl}^{(n)} = H_{y0}e^{j(\omega t^{(n)} - \tilde{\kappa_x}x_l - \tilde{\kappa_y}y_l)} \tag{12b}$$

$$e_{zk}^{(m)} = E_{z0} e^{j(\omega t^{(m)} - \tilde{\kappa_x} x_k - \tilde{\kappa_y} y_k)} \tag{12c}$$

where, $\bar{\kappa} = \hat{x}\tilde{\kappa}_x + \hat{y}\tilde{\kappa}_y$ is the numerical wave vector, ω the frequency of the source and H_{x0}, and H_{y0} and E_{z0} are the amplitudes of the H_x, H_y and E_z field components, respectively.

Applying Eq. (12) to Eq. (11a) and Eq. (11b), the expressions for H_{x0} and H_{y0} (in terms of E_{z0}) can be obtained.

$$H_{x0} = -\frac{E_{z0}}{\mu} \cdot \frac{\left\{e^{-j(\tilde{\kappa_x}\Delta x_k + \tilde{\kappa_y}\Delta y_k)}\right\}\left\{\frac{\partial N_k}{\partial y}\right\}^T}{\left\{e^{j\omega\Delta t^{(n)}}\right\}\left\{\frac{\partial Q^{(n)}}{\partial t}\right\}^T} \tag{13a}$$

$$H_{y0} = \frac{E_{z0}}{\mu} \cdot \frac{\left\{e^{-j(\tilde{\kappa_x}\Delta x_k + \tilde{\kappa_y}\Delta y_k)}\right\}\left\{\frac{\partial N_k}{\partial x}\right\}^T}{\left\{e^{j\omega\Delta t^{(n)}}\right\}\left\{\frac{\partial Q^{(n)}}{\partial t}\right\}^T} \tag{13b}$$

where, $\Delta x_{k(i)} = x_{k(i)} - x_{\langle l\rangle}$, $\Delta y_{k(i)} = y_{k(i)} - y_{\langle l\rangle}$, $\Delta t_\tau^{(n)} = t_\tau^{(n)} - t^{[m]}$, i is the local index of a node in space element and τ the local index of a node in time element.

Applying Eq. (12) and Eq. (13) on Eq. (11c) and dividing both sides of the equation by E_{z0}, the numerical dispersion relation can be obtained

$$\begin{aligned}
&\left\{e^{j\omega\Delta t^{(n)}}\right\}\left\{\frac{\partial Q^{(n)}}{\partial t}\right\}^T \cdot \left\{e^{j\omega\Delta t^{(m)}}\right\}\left\{\frac{\partial Q^{(m)}}{\partial t}\right\}^T \\
&= v_p^2 \cdot \left(\left\{e^{-j(\tilde{\kappa_x}\Delta x_k + \tilde{\kappa_y}\Delta y_k)}\right\}\left\{\frac{\partial N_k}{\partial x}\right\}^T \cdot \left\{e^{-j(\tilde{\kappa_x}\Delta x_l + \tilde{\kappa_y}\Delta y_l)}\right\}\left\{\frac{\partial N_l}{\partial x}\right\}^T\right. \\
&\quad \left. + \left\{e^{-j(\tilde{\kappa_x}\Delta x_k + \tilde{\kappa_y}\Delta y_k)}\right\}\left\{\frac{\partial N_k}{\partial y}\right\}^T \cdot \left\{e^{-j(\tilde{\kappa_x}\Delta x_l + \tilde{\kappa_y}\Delta y_l)}\right\}\left\{\frac{\partial N_l}{\partial y}\right\}^T\right)
\end{aligned} \tag{14}$$

where, $v_p = \frac{1}{\sqrt{\mu\epsilon}}$, $\Delta x_{l(i)} = x_{l(i)} - x_{\langle k\rangle}$, $\Delta y_{l(i)} = y_{l(i)} - y_{\langle k\rangle}$, $\Delta t_\tau^{(m)} = t_\tau^{(m)} - t^{[n]}$, i is the local index of a node in space element and τ the local index of a node in time element.

For omnidirectional propagation in an isotropic medium, Eq. (14) can be written as

$$\begin{aligned}
&\left\{e^{j\omega\Delta t^{(n)}}\right\}\left\{\frac{\partial Q^{(n)}}{\partial t}\right\}^T \cdot \left\{e^{j\omega\Delta t^{(m)}}\right\}\left\{\frac{\partial Q^{(m)}}{\partial t}\right\}^T \\
&= v_p^2 \cdot \left(\left\{e^{-j\tilde{\kappa}(\Delta x_k \cos\phi + \Delta y_k \sin\phi)}\right\}\left\{\frac{\partial N_k}{\partial x}\right\}^T\right.
\end{aligned}$$

$$\cdot \left\{ e^{-j\tilde{\kappa}(\Delta x_l \cos\phi + \Delta y_l \sin\phi)} \right\} \left\{ \frac{\partial N_l}{\partial x} \right\}^T + \left\{ e^{-j\tilde{\kappa}(\Delta x_k \cos\phi + \Delta y_k \sin\phi)} \right\} \left\{ \frac{\partial N_k}{\partial y} \right\}^T \cdot \left\{ e^{-j\tilde{\kappa}(\Delta x_l \cos\phi + \Delta y_l \sin\phi)} \right\} \left\{ \frac{\partial N_l}{\partial y} \right\}^T \Bigg) \quad (15)$$

where, $\tilde{\kappa_x} = \tilde{\kappa}\cos\phi$, $\tilde{\kappa_y} = \tilde{\kappa}\sin\phi$ and ϕ is the direction angle of propagation with respect to the x-axis. Eq. (15) does not assume any specific shape for the mesh. As a result, this relationship holds for all types of linear triangular meshes.

Equation (15) can be used with Newton's iterative method to obtain the numerical wave vector $\tilde{\kappa}$. In the work of Taflove and Hagness [26] a similar technique was used to calculate the wave vector in all directions. The normalized propagation velocity, $v_p/c = 2\pi/\tilde{\kappa}_{final}$ can be calculated using the final converged value for a specific angle of propagation.

5.2. Calculation of Numerical Dispersion

To calculate the numerical dispersion of the mesh presented in Section 3.1, Newton's iterative method was implemented using Eq. (15) in MATLAB. This code was used to calculate the phase velocity of the EM wave in different directions using two different meshes: first, the IRT mesh used in Section 3.1 (shown in Fig. 5(a)) and the second, an "Equilateral Triangle (ET) Mesh" shown in Fig. 5(b). The calculation of the numerical dispersion was performed for different resolutions. For simplicity, the resolution of a mesh is expressed in the form m/λ where, the resolution is m points per wavelength. This convention will be followed throughout this paper.

Figure 5(c) shows the phase velocity variation for resolutions from $4/\lambda$ to $10/\lambda$, with the propagation angle ϕ in degrees. As it can be seen, the phase velocities of the IRT mesh (dashed lines) show a high variation with the angle, ϕ. On the other hand, phase velocities of the ET mesh (solid lines) are almost constant. A slight ripple can be seen for more coarse resolutions of $4/\lambda$ and $5/\lambda$. Resolutions higher than $5/\lambda$ show no variation in phase velocity for the ET mesh. However, for the IRT mesh, a high phase velocity variation is visible for all resolutions, as shown in Fig. 5(c).

A more precise measurement of the numerical dispersion can be obtained from the standard deviation of the phase velocities for different propagation angles. Fig. 6 shows the relationship between the

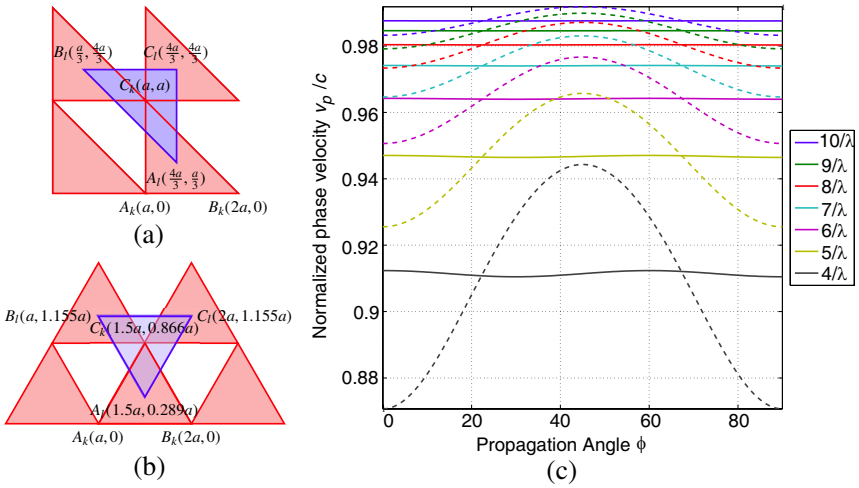


Figure 5. Calculation of phase velocity using (a) IRT mesh and (b) ET mesh. (c) Comparison of phase velocities in different directions for different resolutions in IRT and ET meshes. Dashed and solid lines are normalised phase velocity curves for the IRT and the ET meshes, respectively.

standard deviation of the normalized phase velocity with the resolution for both type of meshes. The horizontal green dashed line in Fig. 6(a) denotes the standard deviation of the phase velocity of $30/\lambda$ resolution with the IRT mesh (3.25×10^{-4}). As can be seen, the numerical dispersion of $5/\lambda$ resolution for the ET mesh is below the dashed line and the standard deviation of the phase velocity is lower (2.132×10^{-4}) than that of the $30/\lambda$ IRT mesh. So, the numerical dispersion is less in the ET mesh with $5/\lambda$ resolution than the IRT mesh with a much finer resolution of $30/\lambda$. This allows a reduction of resolution by a factor of 6 which can be used for the ET mesh to obtain a similar numerical dispersion. This factor can be called the "Resolution Reduction Factor (RRF)".

When the resolution of the ET mesh is increased, the RRF also increases. Fig. 6(b) shows a comparison of the standard deviation of the phase velocity of the $200/\lambda$ IRT mesh with that of the ET mesh. As can be seen, a slightly lower standard deviation can be obtained using a resolution of only $11/\lambda$ in the ET mesh with the RRF value of more than 18. To make a more precise measurement of the RRF, the following mathematical procedure is used.

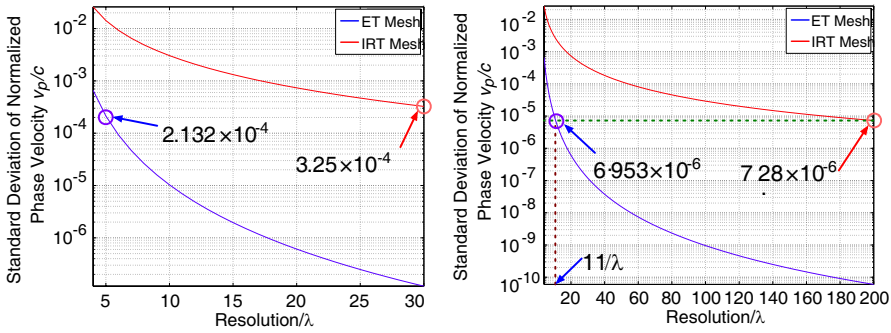


Figure 6. Comparison of the numerical dispersion performance of the ET mesh and the IRT mesh. (a) Standard deviation of normalised phase velocity from $4/\lambda$ to $30/\lambda$ resolution. (b) Standard deviation of normalised phase velocity from $4/\lambda$ to $200/\lambda$ resolution.

5.3. Calculating Resolution Reduction Factor

Since the calculation of normalized phase velocity for the ET and the IRT mesh has to be carried out separately using procedure described in Section 5.1, two different sets of resolutions $\mathbf{RES_{ET}}$ and $\mathbf{RES_{IRT}}$ are taken for the ET and the IRT meshes respectively. The resolutions of the ET and the IRT meshes can be expressed in set form using Eq. 16(a) and Eq. 16(b) respectively as

$$\mathbf{RES_{ET}} = \{i | i \in \mathbb{N}\} \tag{16a}$$

$$\mathbf{RES_{IRT}} = \{i | i \in \mathbb{N}\} \tag{16b}$$

where i is the number of points per wavelength on the mesh (i/λ). The standard deviation of the normalized phase velocity presented in Figs. 6(a) and 6(b) can be expressed as a function of the resolution as $\mathcal{S}_{V_{ET}}$ and $\mathcal{S}_{V_{IRT}}$ for the ET and the IRT meshes respectively. Two mapping sets for the ET and the IRT mesh can be taken as $\mathbf{STD_{V_{ET}}}$ and $\mathbf{STD_{V_{IRT}}}$ in Eqs. 17(a) and 17(b), respectively as

$$\mathbf{STD_{V_{ET}}} = \{s | s = \mathcal{S}_{V_{ET}}(i),\ i \in \mathbf{RES_{ET}},\ s \in \mathbb{R}\} \tag{17a}$$

$$\forall i,\ \mathcal{S}_{V_{ET}} : \mathbf{RES_{ET}} \mapsto \mathbf{STD_{V_{ET}}}$$

$$\mathbf{STD_{V_{IRT}}} = \{s | s = \mathcal{S}_{V_{IRT}}(i),\ i \in \mathbf{RES_{IRT}},\ s \in \mathbb{R}\} \tag{17b}$$

$$\forall i,\ \mathcal{S}_{V_{IRT}} : \mathbf{RES_{ET}} \mapsto \mathbf{STD_{V_{IRT}}}$$

To calculate the RRF resolution, i from the set $\mathbf{RES_{ET}}$ and j from the set $\mathbf{RES_{IRT}}$ can be taken where the standard deviations of i and j, $\mathcal{S}_{V_{ET}}(i)$ and $\mathcal{S}_{V_{IRT}}(j)$, respectively are almost equal. Hence a

set, **RRF** can be obtained as shown in Eq. (18).

$$\mathbf{RRF} = \left\{ r | r = \frac{j}{i},\ i \in \mathbf{RES_{ET}},\ j \in \mathbf{RES_{IRT}},\ \mathcal{S}_{V_{ET}}(i) \cong \mathcal{S}_{V_{ET}}(j) \right\} \quad (18)$$

As can be seen from Fig. 6, $\mathcal{S}_{V_{ET}}$ reduces faster than $\mathcal{S}_{V_{IRT}}$. Therefore, each resolution in set $\mathbf{RES_{ET}}$ maps to one unique member of set **RRF**. A mapping function, $\mathcal{R}_{ET}$, can be defined which relates the resolutions of the ET mesh to the RRF, as Eq. (19).

$$\forall i \in \mathbf{RES_{ET}},\ \mathcal{R}_{ET} : \mathbf{RES_{ET}} \mapsto \mathbf{RRF} \quad (19)$$

This procedure was implemented in MATLAB to determine nature of the relationship between the resolution for the ET mesh and the RRF. The relationship between the RRF and the resolution of the ET mesh, $\mathcal{R}_{ET}$ can be seen in Fig. 7 where $\mathcal{R}_{ET}$ is nearly a linear function. Hence, the RRF improves linearly when the resolution is progressively increased.

6. COMPARING NUMERICAL DISPERSION OF MESHES BY SIMULATION

The theoretical analysis of numerical dispersion shown in Section 5 can be verified by running simulations with both type of mesh using the C++ code. To compare the numerical dispersion of both the meshes, a very simple setup was made. A point source was placed at the center of a square computational domain of free-space ($\mu_r = 1$ and $\epsilon_r = 1$).

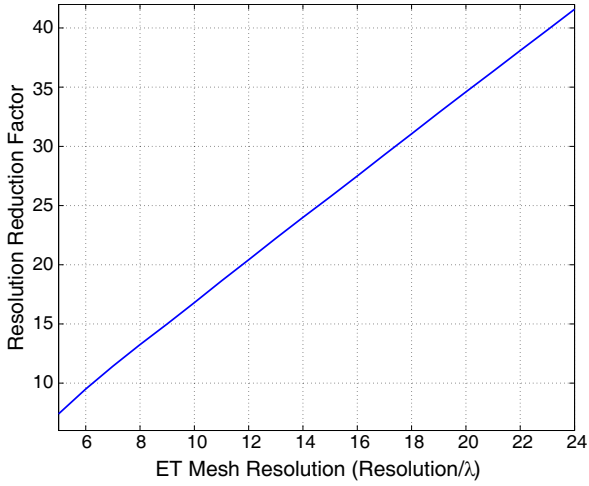
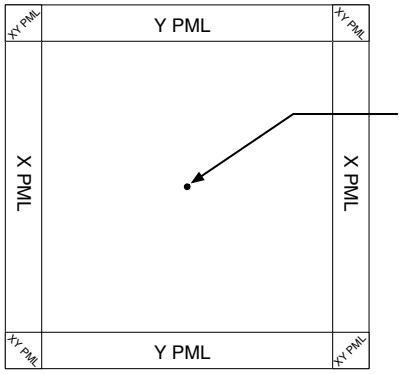


Figure 7. Resolution Reduction Factor vs resolution of the ET mesh, $\mathcal{R}_{ET}$.

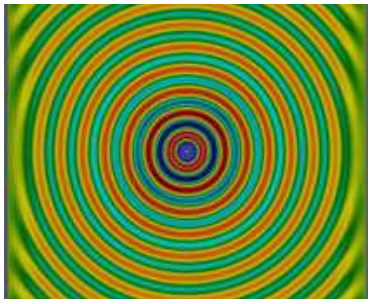
The computational domain was also surrounded by appropriate PML boundaries, as shown in Fig. 8(a). The point source placed at the center is an E_z field continuous sine wave source with frequency 1 Hz (normalized). For all the simulations, $c\Delta t/\Delta l = 0.1$ (where, Δt is the length of 1 time step and Δl is the length of both the x and y sides of an element in the IRT mesh and the length of any side of that of the ET mesh) was maintained.

Initially the simulations were performed at a resolution of $10/\lambda$. Figs. 8(b) and 8(c) show the E_z field profile after 2000 time steps. At this resolution, the standard deviation of the normalized phase velocity, v_p/c, is 3.041×10^{-3} for the IRT mesh and only 1.032×10^{-5} for the ET mesh (see Fig. 6(a)). Although the standard deviation of v_p/c for the ET mesh is much smaller than that of the IRT mesh, however in this case both the values are small enough to make the E_z field profiles almost identical (to the naked eye) for small computational domains, as shown in Figs. 8(b) and 8(c).

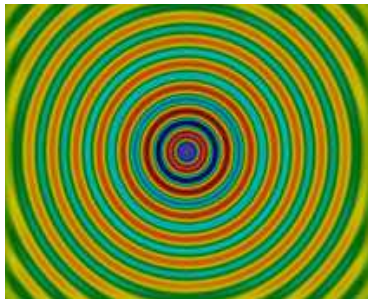
However, at lower resolution, the effect of the numerical dispersion can be easily visualized on the field profile, even in a small computational domain. Hence, simulations were performed at a



(a)



(b)



(c)

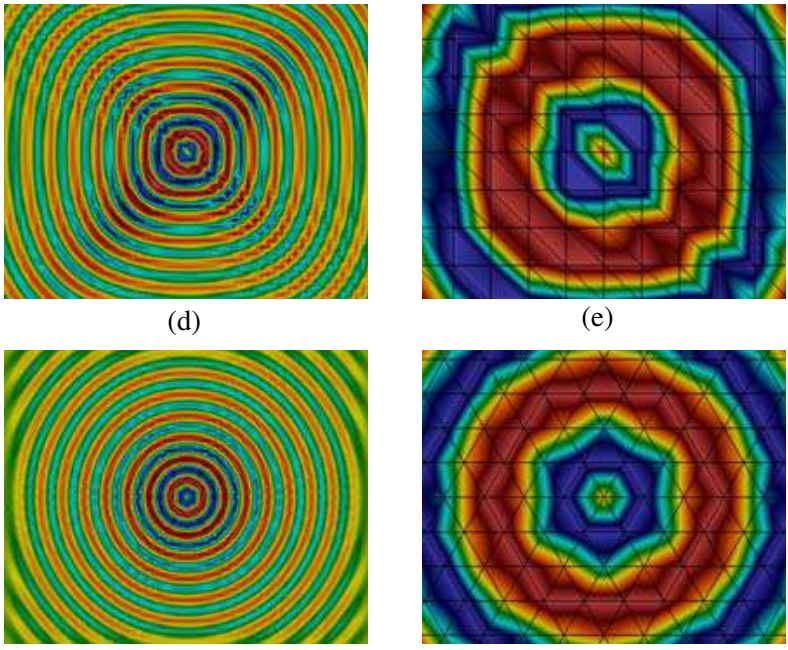


Figure 8. Simulation results of the proposed method using the IRT and the ET meshes. (a) Computational domain with a point source at the center and PML boundaries near the boundary of the domain. (b) E_z field profile after 2000 time steps with $10/\lambda$ IRT mesh. (c) E_z field profile after 2000 time steps with $10/\lambda$ ET mesh. (d) E_z field profile after 2000 time steps with $4/\lambda$ IRT mesh. (e) Magnified view of the central region with mesh overlay of the field shown in Fig. 8(d). (f) E_z field profile after 2000 time steps with $4/\lambda$ ET mesh. (g) Magnified view of the central region with mesh overlay of the field shown in Fig. 8(f).

resolution of $4/\lambda$. At this resolution, the standard deviation of v_p/c for the IRT mesh is 2.598×10^{-2} and that for the ET mesh is 6.611×10^{-4}. The standard deviation of v_p/c for the ET mesh at this resolution is lower than that of the $10/\lambda$ IRT mesh. So the E_z field profile of the $4/\lambda$ ET mesh and the $10/\lambda$ IRT mesh should be almost identical. As the standard deviation of v_p/c for the IRT mesh at this resolution is higher, the visible distortion must be present in the E_z field profile.

Figures 8(d) and 8(f) show the E_z field profiles obtained after 2000 time steps for the $4/\lambda$ IRT and the $4/\lambda$ ET meshes respectively. As has been discussed in the previous paragraph, the impact of the lower

resolution is clearly visible in the field profile presented in Fig. 8(d). At this resolution with the IRT mesh, the evolution of the E_z field is no longer circular; rather it became somewhat square in profile. This is due to the speed variation in different angle of propagation. As shown in Fig. 5(c), v_p/c at 0°, 45°, 90° are 0.8707, 0.9443 and 0.8707, respectively. These data indicate that the speeds of propagation at 0° and 90° will be 7.36% slower than that of the propagation at 45°. The variation in v_p/c is also a continuous function of the propagation angle. As a result, the E_z field profile presented in Fig. 8(d) became a rounded square shape instead of circle, as it should have been in the ideal case.

For the ET mesh at $4/\lambda$ resolution, the highest point of v_p/c in Fig. 5(c) is at the 60° angle and the lowest point is at the 30° angle with their values 0.9124 and 0.9105, respectively, Showing a difference of only 0.019%. Again, the variation of v_p/c is a continuous function of the propagation angle. As the difference of speed is much smaller compared to that of the IRT mesh of same resolution, the ET mesh at the $4/\lambda$ resolution retains the near circular shape of propagation of the E_z field profile in Fig. 8(f).

Results presented in Figs. 8(d) and 8(f) can be further explained by a closer examination of the evolution of field near the point source. Figs. 8(e) and 8(g) show the field close to the point source with an overlay of the mesh used during the simulation. It can be seen from Fig. 8(e) that, the points surrounding the point source in the IRT mesh are not equidistant. As a result, the calculated field at the direction of the farthest point moves faster than that of the closest points. As the points closest to the source are at 0° and 90° and the farthest point is at 45°, the maximum v_p/c is found at 45° and minima can be found along angles of 0° and 90°.

However, for the ET mesh, all six points surrounding the source are equidistant, as shown in Fig. 8(g). The closest point to the source is at 30° at the middle of an edge of the element. The six equidistant points are located at 0°, 60°, 120°, 180°, 240° and 300° and along these angles the maximum values of v_p/c can be found. The minimum values of v_p/c can be interpolated at a point on the outer edge of surrounding elements at 30°, 90°, 150°, 210°, 270° and 330°.

The above discussion highlights the accuracy of the method when used with the ET mesh. Even a low resolution ET mesh of $4/\lambda$ produces an acceptable solution, where the $4/\lambda$ IRT mesh is numerically unusable, even in the smallest possible computational domain.

In a more practical situation use of only ET meshes may not be able to represent the whole device to be analyzed. A small number of

other irregular types of elements may have to be introduced into the mesh system to represent arbitrary shape more conveniently. These non ET elements can introduce some additional numerical dispersion into the simulation. As long as most of the elements are close to equilateral, the overall numerical dispersion of the entire computational domain will remain considerably small.

7. COMPARISON WITH THE FDTD METHOD

To compare the numerical dispersion of the proposed method with that of the more widely used FDTD method, Eq. (15) can be further simplified for both the IRT mesh of Fig. 5(a) and the ET mesh of Fig. 5(b).

For the IRT mesh, the nodal data of Fig. 5(a) can be applied to Eq. (15) and the equation can be simplified as

$$\begin{aligned}
&\frac{1}{v_p^2 t^2}\left(e^{j\omega\frac{t}{2}} - e^{-j\omega\frac{t}{2}}\right)^2 \\
&= \frac{1}{a^2}\left[\left(e^{-j\tilde{\kappa}\frac{a}{2}\cos\phi} - e^{j\tilde{\kappa}\frac{a}{2}\cos\phi}\right)\cdot\left(e^{-j\tilde{\kappa}\frac{a}{2}\cos\phi} - e^{j\tilde{\kappa}\frac{a}{2}\cos\phi}\right)\right. \\
&\quad \left. + \left(e^{-j\tilde{\kappa}\frac{a}{2}\sin\phi} - e^{j\tilde{\kappa}\frac{a}{2}\sin\phi}\right)\cdot\left(e^{-j\tilde{\kappa}\frac{a}{2}\sin\phi} - e^{j\tilde{\kappa}\frac{a}{2}\sin\phi}\right)\right] \\
&\Rightarrow \left[\frac{a}{v_p t}\sin\left(\frac{\omega t}{2}\right)\right]^2 \\
&= \underline{\left[\sin\left(\frac{\tilde{\kappa}a\cos\phi}{2}\right)\right]^2} + \underline{\left[\sin\left(\frac{\tilde{\kappa}a\sin\phi}{2}\right)\right]^2} \qquad (20)
\end{aligned}$$

Similarly, the nodal data from Fig. 5(b) was applied on Eq. (15) and simplified as

$$\begin{aligned}
\left[\frac{a}{v_p t}\sin\left(\frac{\omega t}{2}\right)\right]^2 &= \underline{\left[\sin\left(\frac{\tilde{\kappa}a\cos\phi}{2}\right)\right]^2} + \underline{\left[0.577\cdot\sin\left(\tilde{\kappa}0.866a\sin\phi\right)\right]^2} \\
&\quad + \underline{\left[0.577\cdot\left(\cos\left(\frac{\tilde{\kappa}a\cos\phi}{2}\right) - \cos(\tilde{\kappa}0.866a\sin\phi)\right)\right]^2} \qquad (21)
\end{aligned}$$

Two parts of Eq. (20) are underlined in red and blue and similarly, three parts of Eq. (21) are underlined with red, blue and cyan colors respectively. As can be seen, the part underlined with red is common to both the equations. The constants of the blue underlined parts are different in the two equations. The cyan underlined part in Eq. (21) is absent in Eq. (20). Due to this extra element in Eq. (21), the proposed method with the ET mesh shows a more stable solution for all possible angles, compared to Eq. (20).

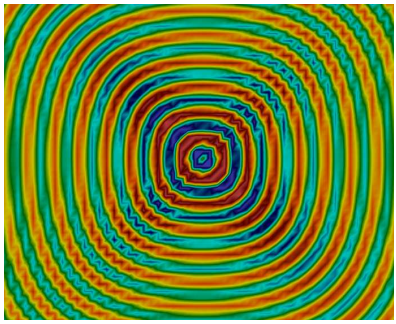


Figure 9. E_z field profile after 2000 time steps with the FDTD method.

In the work of Taflove and Hagness [26], the numerical dispersion relation for the FDTD method has been given, which is identical to that the proposed method when used with the IRT mesh, as shown in Eq. (20). Therefore, the numerical dispersion characteristics of the FDTD method will be similar to that of the proposed method when used with the IRT mesh.

To prove the similarity of the propagation with IRT mesh and the FDTD method, the FDTD method was implemented and a simulation was performed with a resolution of $4/\lambda$ (keeping everything same as shown in Section 6). Fig. 9 shows the E_z field profile after 2000 time steps. It can be seen that, both Figs. 8(d) and 9 show similar rounded square propagation for the FDTD and the proposed method with the IRT mesh respectively. With the ET mesh with the same resolution, the proposed method however, retains the near circular shape in Fig. 8(f), confirming the superiority of the method proposed here.

8. CONCLUSION

A novel finite element-based time domain method is presented here with a unique perforated double mesh technique. The perforated mesh technique reduces the number of elements required to less than half, which reduces the memory requirement and computational time. The advantage of using the ET mesh over the IRT mesh has also been clearly shown in the result presented. This paper also proves similarity of numerical dispersion characteristics of the popular FDTD method to the proposed method when used with the IRT mesh. Hence, the advantage in numerical dispersion of the proposed method with the ET mesh over the FDTD method is clearly demonstrated. It has

also suggested a way to reduce the resolution of the mesh keeping the numerical dispersion the same, which should allow a reduction of the memory usage and the computational time. The numerical dispersion of the proposed method with the ET mesh at low resolution is negligible, compared to the FDTD method. It lifts the constrain of a finer mesh being needed to reduce the numerical dispersion. As the method uses the FE mesh, any arbitrary shaped structures can be accurately approximated just by moving the nodes. Although this might introduce a small amount of additional numerical dispersion, but the overall computational domain will remain almost dispersion free. The FE mesh also allows dense and coarse regions to be defined in the same mesh representation. These characteristics of the method presented here might be advantageous for simulation and design of long complex and phase matching devices such as interferometers, acoustic waveguides, polarisation converters photonic crystal fibres, plasmonic guides metamaterial antennas etc. Additionally, the proposed method can perform accurate simulations with a relatively much reduced resolution.

The use of the FE-based mesh provides the flexibility to move the node points when necessary. As a result, structural deformations due to various physical influences (e.g., heating, pressure variations etc.) might be incorporated with the method to enable the better characterization of the interaction of physical effects with photonics devices.

ACKNOWLEDGMENT

An special thanks to U. S. Army Communications-Electronics Research, Development & Engineering Center (CERDEC) for partially funding the development of the method.

REFERENCES

1. Yee, K., "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, Vol. 14, No. 3, 302–307, 1966.
2. Taflove, A. and S. Hagness, *Computational Electrodynamics*, Artech House, Boston, 1995.
3. Taflove, A. and M. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 23, No. 8, 623–630, 1975.

4. Sun, G. and C. Trueman, "Some fundamental characteristics of the one-dimensional alternate-direction-implicit finite-difference time-domain method," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 52, No. 1, 46–52, 2004.
5. Lee, J., R. Lee, and A. Cangellaris, "Time-domain finite-element methods," *IEEE Transactions on Antennas and Propagation*, Vol. 45, No. 3, 430–442, 1997.
6. Guiffaut, C. and K. Mahdjoubi, "A parallel FDTD algorithm using the MPI library," *IEEE Antennas and Propagation Magazine*, Vol. 43, No. 2, 94–103, 2001.
7. Adams, S., J. Payne, and R. Boppana, "Finite difference time domain (FDTD) simulations using graphics processors," *IEEE DoD High Performance Computing Modernization Program Users Group Conference*, 334–338, 2007.
8. Sypek, P., A. Dziekonski, and M. Mrozowski, "How to render FDTD computations more effective using a graphics accelerator," *IEEE Transactions on Magnetics*, Vol. 45, No. 3, 1324–1327, 2009.
9. Smyk, A. and M. Tudruj, "Openmp/MPI programming in a multi-cluster system based on shared memory/message passing communication," *Advanced Environments, Tools, and Applications for Cluster Computing*, 157–160, 2002.
10. Farjadpour, A., D. Roundy, A. Rodriguez, M. Ibanescu, P. Bermel, J. Joannopoulos, S. Johnson, and G. Burr, "Improving accuracy by subpixel smoothing in the finite-difference time domain," *Optics Letters*, Vol. 31, No. 20, 2972–2974, 2006.
11. Rahman, B. M. A. and J. Davies, "Finite-element analysis of optical and microwave waveguide problems," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 32, No. 1, 20–28, 1984.
12. Rahman, B. M. A. and J. Davies, "Finite-element solution of integrated optical waveguides," *Journal of Lightwave Technology*, Vol. 2, No. 5, 682–688, 1984.
13. Hayata, K., M. Koshiba, M. Eguchi, and M. Suzuki, "Vectorial finite-element method without any spurious solutions for dielectric waveguiding problems using transverse magnetic-field component," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 34, No. 11, 1120–1124, 1986.
14. Cangellaris, A., C. Lin, and K. Mei, "Point-matched time domain finite element methods for electromagnetic radiation and scattering," *IEEE Transactions on Antennas and Propagation*, Vol. 35, No. 10, 1160–1173, 1987.
15. Feliziani, M. and E. Maradei, "Point matched finite element-time

domain method using vector elements," *IEEE Transactions on Magnetics*, Vol. 30, No. 5, 3184–3187, 1994.

16. Koshiba, M., Y. Tsuji, and M. Hikari, "Time-domain beam propagation method and its application to photonic crystal circuits," *Journal of Lightwave Technology*, Vol. 18, No. 1, 102, 2000.
17. Hesthaven, T. W. J. S., "High-order/spectral methods on unstructured grids I. Time-domain solution of Maxwell's equations," Tech. Rep. 2001-6, ICASE NASA Langley Research Center, Hampton, Virginia, March 2001.
18. Songoro, H., M. Vogel, and Z. Cendes, "Keeping time with Maxwell's equations," *IEEE Microwave Magazine*, Vol. 11, No. 2, 42–49, 2010.
19. Gedney, S. and U. Navsariwala, "An unconditionally stable finite element time-domain solution of the vector wave equation," *IEEE Microwave and Guided Wave Letters*, Vol. 5, No. 10, 332–334, 1995.
20. Joannopoulos, J. D., S. G. Johnson, J. N. Winn, and R. D. Meade, *Photonic Crystals: Molding the Flow of Light*, 2nd Edition, Princeton University Press, 2008.
21. Berenger, J., "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, Vol. 114, No. 2, 185–200, 1994.
22. Berenger, J., "Perfectly matched layer for the fdtd solution of wave-structure interaction problems," *IEEE Transactions on Antennas and Propagation*, Vol. 44, No. 1, 110–117, 1996.
23. Veselago, V., et al., "The electrodynamics of substances with simultaneously negative values of ε and μ," *Physics-Uspekhi*, Vol. 10, No. 4, 509–514, 1968.
24. Hao, Y. and R. Mittra, *FDTD Modeling of Metamaterials*, Artech House, 2009.
25. Juntunen, J. and T. Tsiboukis, "Reduction of numerical dispersion in FDTD method through artificial anisotropy," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 48, No. 4, 582–588, 2000.
26. Taflove, A. and S. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 2nd edition, Artech House, 2000.