ENPH479 - High Performance Computational Physics
The FDTD Method with Numba Optimization and MPI Implementation

Jane Cohen[1,*]

[1] *Department of Physics, Engineering Physics and Astronomy,*
Queen's University, Kingston, ON K7L 3N6, Canada

(Dated: April 4, 2024)

The Finite-Difference Time-Domain (FDTD) method has established itself as a cornerstone in computational electromagnetism, enabling the detailed modeling of electromagnetic wave interactions with various media. This paper explores the application and enhancement of the FDTD method, extending from one-dimensional to more complex two-dimensional simulations, incorporating a time-dependent electric-dipole source and analyzing field behavior in the presence of a dielectric material. A key focus is placed on the absorption and reflection of the electromagnetic fields, facilitated by the implementation of absorbing boundary conditions (ABCs), the total-field scattered-field (TFSF) approach, and the integration of a dielectric material into the simulation domain. The study evaluates the transmission and reflection coefficients derived from the simulations against theoretical values to validate the FDTD model's accuracy and reliability.

The paper further explores the significant advantages of utilizing Numba, an open-source compiler that translates Python functions to optimized machine code, and Message Passing Interface (MPI) for parallel computing. These optimizations are shown to drastically reduce computation times, enabling more extensive and complex simulations. The results confirm the FDTD method's ability to model electromagnetic phenomena and demonstrate the improvements in simulation efficiency and scalability achieved through computational optimizations.

I. INTRODUCTION

The Finite-Difference Time-Domain (FDTD) method is a valuable computational modeling technique, widely known for its ability to solve Maxwell's equations through spatial and temporal domain discretization. The discretized form of the spatial domain allows for unique geometries to be modelled to understand the evolution of electromagnetic fields throughout time. Originating from Yee's algorithm in the 1960s, FDTD has undergone extensive development, resulting in a tool that is used in various fields such as telecommunications, antenna design, and photonics [1].

While its use has become widespread, the FDTD method does not come without its challenges. The computational resources required to use the algorithms scale with the size of the spatial domain and number of time steps. The large number of values stored and used between time steps requires large amounts of memory and computational power. The careful implementation of various computing techniques can reduce this burden, allowing the FDTD method to be applied to complex simulations.

II. THEORY

A. Finite-difference time-domain method

Using central difference approximations and rearranging for an iterative solution, the FDTD algorithm for 1d free space propagation is given by Equation 1. In this set of equations, the electric field, $\tilde{E}$, is scaled using Equation 2 to be similar in size to H, the magnetic field. Here, k is the index in the spatial domain and n is the index in the time domain.

$$\begin{aligned}\tilde{E}_x^{n+1/2}(k) &= \tilde{E}_x^{n-1/2}(k) \qquad (1a)\\ &\quad - \frac{c\Delta t}{\Delta x}[H_y^n(k+\frac{1}{2}) - \hat{H}_y^n(k-\frac{1}{2})]\\ H_y^{n+1}(k+\frac{1}{2}) &= H_y^n(k+\frac{1}{2}) \qquad (1b)\\ &\quad - \frac{c\Delta t}{\Delta x}[\tilde{E}_x^{n+1/2}(k+1) - \tilde{E}_x^{n+1/2}(k)]\end{aligned}$$

$$\tilde{E} = \sqrt{\frac{\epsilon_0}{\mu_0}}E = \epsilon_0 c E \qquad (2)$$

Using the Courant condition ($\Delta t < \frac{\Delta x}{\sqrt{n}c}$) to ensure stability, the time step for the FDTD algorithm $\Delta t = \frac{\Delta x}{2c}$ is chosen. This further simplifies the FDTD algorithm to give Equation 3, where $\tilde{E}$ is expressed as E for simplicity

$$\begin{aligned}E_x^{n+1/2}(k) &= E_x^{n-1/2}(k) \qquad (3a)\\ &\quad - \frac{1}{2}[H_y^n(k+\frac{1}{2}) - \hat{H}_y^n(k-\frac{1}{2})]\\ H_y^{n+1}(k+\frac{1}{2}) &= H_y^n(k+\frac{1}{2}) \qquad (3b)\\ &\quad - \frac{1}{2}[E_x^{n+1/2}(k+1) - E_x^{n+1/2}(k)]\end{aligned}$$

Expanding the FDTD method to two dimensions gives a similar set of equations that again describe propagation of an electromagnetic field with no dispersion. For transverse magnetic (TM), these equations are a coupled set of equations for D_z, H_x and H_y that follow the same discretized

[*] 18jpcw@queensu.ca

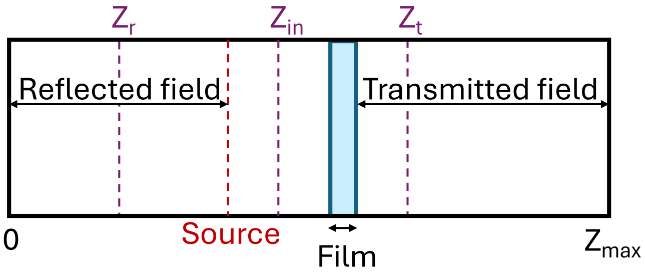

Figure 1. The spatial regions of a 1d domain for a polarization source injected into the domain towards the film. The reflected field can be measured at z_r, anywhere behind the source. The transmitted field is measured at z_t, in front of the film. The incident field is analogous with the source in this case.

form as the 1d algorithm. In this case, E_z is defined as $\frac{1}{\epsilon}D_z$. The full equations are given by Equation 10a-10c in the course notes [2].

B. Reflection and transmission across a dielectric

The 1d and 2d FDTD algorithms can be solved to explore the behaviour of an electric field in a spatial domain over time. The effects of adding a dielectric to the domain can be modelled by modifying the coefficient of the electric field to reflect the dielectric coefficient ϵ of the medium. When an incident field, E_{in}, interacts with the dielectric, its behaviour depends on the thickness and geometry of the film as well as it's dielectric constant. Adding the dielectric modifies the amplitude of the wave that passes through it (the transmitted wave E_t) and causes a wave to be reflected (E_r). When measuring these three fields in 1d, it is important that the transmitted field is measured after the film and the reflected field before the source. These regions are demonstrated in Figure 1, with z_{in}, z_t and z_r showing arbitrary points for measuring each field within their respective region.

Computing the Fourier transform of these three fields allows for the transmission and reflection coefficients to be found using Equation 4 and Equation 5.

$$T = \left| \frac{E_t(\omega)}{E_{in}(\omega)} \right|^2 \tag{4}$$

$$R = \left| \frac{E_r(\omega)}{E_{in}(\omega)} \right|^2 \tag{5}$$

These numerical values can be compared to the analytical values of T and R given by Equation 6 and Equation 7 where $n = \sqrt{\epsilon}$, $r_1 = \frac{1-n}{1+n}$, $r_2 = \frac{n-1}{n+1}$ and $k_0 = \frac{\omega}{c}$ and L is the width of the film.

$$T_{an}(\omega) = \left| \frac{(1+r_1)(1+r_2)e^{2ik_0Ln}}{1+r_1r_2e^{2ik_0Ln}} \right|^2 \tag{6}$$

$$R_{an}(\omega) = \left| \frac{r_1 + r_2e^{2ik_0Ln}}{1+r_1r_2e^{2ik_0Ln}} \right|^2 \tag{7}$$

III. METHOD

A. Simulating propagation in 1d using the FDTD method

A simulation solving for E_x and H_y in space over time can be run to test the 1d FDTD algorithm. This simulation uses a 2-fs pulse, with a center frequency of 200 THz. The field is launched in opposite directions from a specific z coordinate.

Using a spatial time step that ensures the points per wavelength (PPW) are greater than 20, the simple absorbing conditions (ABC) can be implemented. These boundary conditions are applied on both sides of the spatial domain using Equation 8 and Equation 9. The effectiveness of the ABCs can be determined by zooming in to observe small amplitudes during the simulation to observe the microscope behaviour of the field near the boundary.

$$E_x^n(0) = E_x^{n-2}(1) \tag{8}$$

$$E_x^n(kend-1) = E_x^{n-2}(kend-2) \tag{9}$$

Having the source propagate in both directions is not conducive to a clear simulation of the field's behaviour. Therefore, the part of the source behind the source's origin ($isource$) is subtracted from the H and E field resulting in the source being injected only on the 'forward' direction. In this case, the forward direction is in the positive spatial direction ($z > isource$). This approach is known as the "total-field scattered-field" (TFSF) approach and was implemented using Equation 10 and Equation 11 to update E and H in addition to the FDTD algorithm at each time step.

$$\begin{aligned} E_x(isource) = E_x(isource) \\ - 0.5H_y^{source}(isource) \end{aligned} \tag{10}$$

$$\begin{aligned} H_y(isource-1) = H_y(isource-1) \\ - 0.5E_x^{source}(isource) \end{aligned} \tag{11}$$

Verifying the correct implementation of the TFSF method can be done in a similar way to the ABCs. By observing the field behind the source for very small amplitudes, the amount of back-scattered field can be determined.

To explore the behaviour of the field through a medium, a thin film of L = 1 micron was added with a dielectric constant $\epsilon = 9$ in front of the source E_{in}. From there, the full behaviour of the three fields, E_{in}, E_r and E_t were studied. Using the reflection and transmission coefficients given by Equation 5 and Equation 4, the effects of the dielectric on electromagnetic waves in terms of transmission and reflection were explored. Comparing the analytical reflection and transmission coefficients to these values was used to

validate the numerical model and to provide insight into the accuracy and reliability of the simulations carried out by the FDTD method.
Once the foundational 1d FDTD method was established using the process described above, the solutions of the 2d FDTD algorithm were explored with a time-dependent electric-dipole source injected into a 2d spatial domain. A square box of dielectric material with $\epsilon = 9$ was placed at a fixed position within the spatial domain. For each time step, the D_z and E_z fields are updated according to their discretized equations. Next the dipole source is used to update D_z and E_z. Determining E_z in both these cases considers the dielectric box, which impacts the field at each time step. Next the H_x and H_y fields are updated according to Ez. To maintain stability, a time step of $\Delta t = \frac{\Delta x}{2c}$ is used.

B. Exploiting Numba and MPI

Using the results from the 2d FDTD simulation, various computing techniques were used to increase the speed of the algorithm. The first technique used was slicing and vectorization. Instead of updating each element of the spatial domain using loops, the fields were stored as one dimensional arrays and slicing was employed to update the field each time step. Next, Numba was used on both the original and vectorized code. Numba is an open-source compiler that translates Python code into fast machine code. It's designed to help speed up the execution of Python code, and is especially effective for functions that require heavy numerical computation, such as loops. By adding Numba decorators to functions, they are compiled as machine code just before execution and run at native machine code speed. Once the fastest method from above was determined, parallel computing was employed to increase the speed of the 2d FDTD algorithm. Because of the discretized form of the this method, it is highly suitable for parallel programming. To distribute the computation across multiple processes, the spatial domain was split into 2d "strips". Once the strips were scattered to multiple processes, only communication between the edges of these strips was required in addition to the original 2D FDTD algorithm. Non-blocking communication was used for the exchange of boundary rows to further optimize the program.

IV. RESULTS

A. Propagation in 1d through a dielectric

In practice, the ABCs and TSFS method were successfully implemented. The field being absorbed by both spatial domain boundaries can be seen in Figure 2. At the first time step, the field is moving symmetrically away from the source point. The second time snapshot clearly demonstrates the absorption that occurred at the left boundary. The same phenomena occurs at the right boundary if more time steps are observed.

While it appears as though the entire field is absorbed by the left boundary, there is a slight reflection. For this simulation, the reflection is approximately 0.0005 V/m: a relatively small value compared to the amplitude of the field.

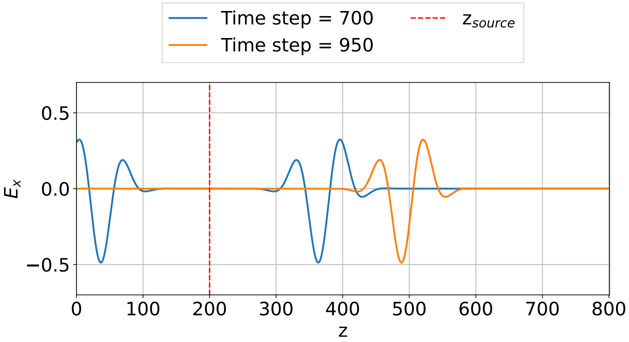

Figure 2. The application of absorbing boundary conditions on a 1d domain. During the first time step (700) the source is travelling in opposite directions. At the second time step (950) the source has been absorbed by the left boundary.

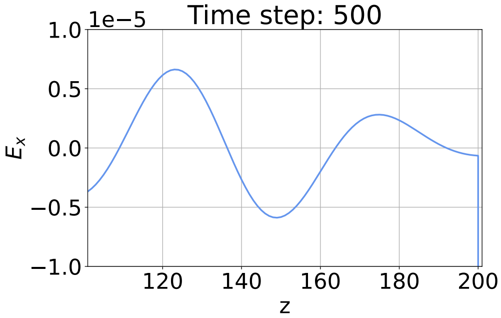

Figure 3. The back scatter from a pulse injected in the forward direction. The pulse is injected at z = 200. The back scatter is small relative to the amplitude of the source.

The TSFS method was generally effective at ensuring a one direction pulse, but does have some back scatter. Figure 3 shows the back scatter of the pulse. In this case, the back-scattered wave has an amplitude of less than 0.00001 V/m.

The main result obtained using the 1d FDTD method was the behaviour of an electromagnetic field as it propagates through free space and interacts with a dielectric film. For the $\epsilon = 9$ film of 1 micron, the propagation of the incident, transmitted and reflected wave can be seen in Figure 4. The incident field consists of the initial pulse. The transmitted and reflected fields diminish in amplitude, but show multiple peaks as the fields are reflected and transmitted through the dielectric.

The transmission and reflection coefficients were found for this dielectric numerically and analytically and are shown in Figure 5. The coefficients are computed in the frequency domain, meaning they show how much of the wave's energy is in the reflected and transmitted field at each frequency. Both coefficients display a significant frequency dependence, revealing how materials respond differently to electromagnetic waves of different frequencies. This result

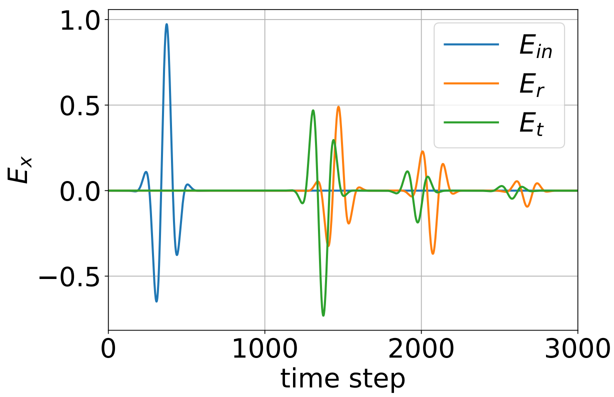

Figure 4. The incident, reflected and transmitted field of an electric-dipole source injected into a 1 dimensional spatial domain over time. The incident field interacts with a dielectric film, resulting in a transmitted and reflected field.

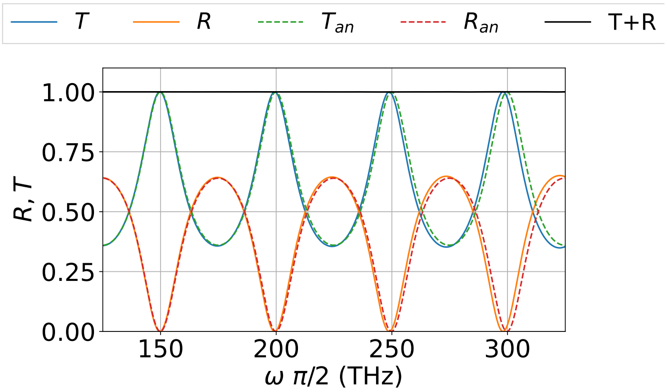

Figure 5. The numerical and analytical reflection and transmission coefficients for an electromagnetic field as it interacts with a dielectric film. The sum of both coefficients is consistently one, indicating the conservation of energy of the system. The discrepancy between the numerical and analytical values increases as the frequency increases.

is important because it allows any frequency bands where the material is particularly reflective or transmissive to be determined. As expected the sum of T and R is 1 across the entire frequency range, which supports the principle of energy conservation in linear, non-absorptive medium. As seen in Figure 5, the computational and analytical values are in agreement for lower frequencies. As the frequency increases, the values begin to diverge.

B. Propagation in 2d through a dielectric box

To determine the fastest algorithm for carrying out the FDTD method, the 2d algorithm was employed. Initially, a non-optimized code was used. For a spatial grid of 500 by 500 points, the resulting electric field found using this code is shown in Figure 6 for two arbitrary time steps. The field spreads out in a radially symmetric fashion, obstructed by the dielectric box shown in red. Within the bounds of the box, the field is reflected and transmitted in the same way as the 1d example, but now within a two dimensional area. The time-dependent source and electric field at the source are shown in a small sub-plot to monitor the injected field.

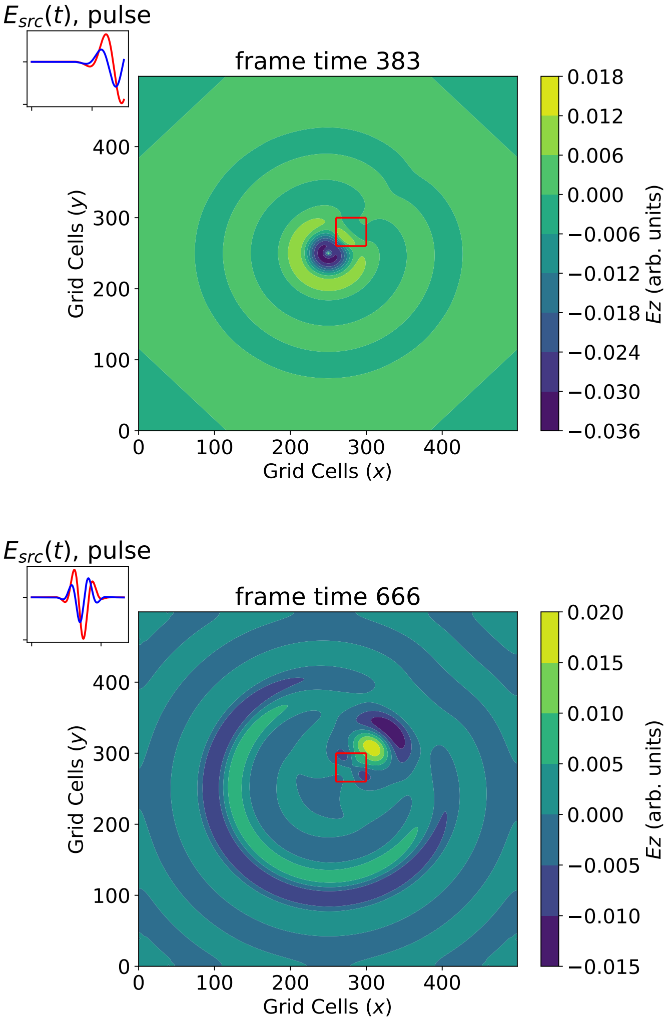

Figure 6. The electric field from a electric-dipole source injected into the centre of the square spatial domain at two time steps. Between the two time steps, the field travels outwards radially, interacting with a dielectric box shown in red. The dielectric causes reflection and attenuation of the field, resulting in the constructive and destructive patterns shown at both time steps. The small plot in the top left corner shows the evolution of the source and the electric field at the source point. The simulation is for a spatial domain of 500 by 500 points and was run for 1000 time steps.

C. Non-parallelized techniques

This simulation was timed for a spatial domain of 500 and 1000 points squared for four versions of the 2d FDTD solver. The first was the non-optimized code that uses loops to update the elements of the fields over the entire 2d domain. The second was a vectorized version of the code that uses splicing to update the fields. Lastly, Numba

Table I. The run times for four versions of serial code applying the 2d FDTD algorithms over 500 and 1000 spatial points squared. The non-optimized version of the code uses for-loops while the vectorized version applies slicing to update the field. Numba was applied to both the codes, with the non-vectorized code with Numba resulting in the fastest run time.

Method	**500x500**	**1000x1000**
Non-optimized (s)	103.76	524.94
Numba (s)	1.86	5.41
Vectorized (s)	3.56	20.53
Numba and vectorized (s)	7.42	33.59

was employed on the first two codes to determine its effects on run time. The results from these tests are displayed in Table I. For the slower methods, the simulation was run for 100 time steps and the run time was multiplied by 10 as the result is expected to scale linearly. Each version was run 3 times and the averaged value was recorded. Using Numba on the non-vectorized code was the fastest version of the 2d FDTD algorithm.

D. The power of Numba and parallel computing

Once it was determined that the non-vectorized code using Numba was the fastest way to implement the 2d FDTD method, it was converted to a parallelized program. This code was tested for a spatial domain of 3008 points squared for 1, 2, 4, 8 and 16 parallel processes with a single node. The algorithm was timed within the Python code around the main loop as well as using the SLURM sbatch file. The results of these timing trials are shown in Table II. Each value is the average of three time trails for that configuration.

The run time decreases as the number of processes is increased. This trend occurs as using more processes allows the computation of the fields at each time step to be distributed over multiple tasks, reducing the number of spatial points handled by each individual process. This result reinforces the fact that FDTD algorithms are highly suitable for parallelization. The run time for 16 processes and 3008 spatial points is comparable to that for the serial code for only 1000 points. This comparison highlights the vast improvement in algorithm speed that parallelization was able to achieve.

The results of the parallel program were also verified using plots saved at time steps throughout the program. An example of these snapshots is shown in Figure 7. The boundary communication is evidently handled correctly, with the graphics having a seamless 2d spatial domain that demonstrates the evolution of the electric field over time.

Table II. Time trials for various number of parallel processes (ntasks) for a FDTD simulation of transverse magnetic field propagation in a spatial domain of 3008 by 3008 points for 1000 time steps. The run time of the program decreases as the number of processes is increased.

ntasks	**Python timer** (s)	**Linux timer** (s)
1	68.99	70.43
2	35.10	36.86
4	12.04	21.67
8	10.15	13.76
16	4.76	5.57

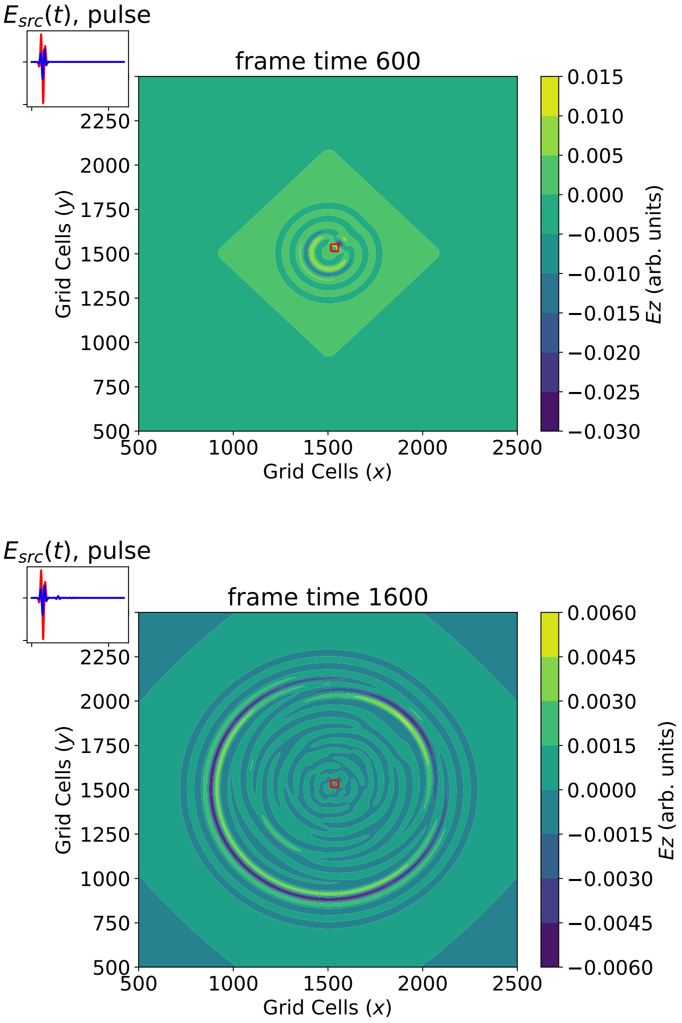

Figure 7. The electric field over a spatial domain of 3008 points at two time steps. The axes have been limited to 2500 points to focus on the details of the field. A small red square outlines a dielectric box placed at a fixed region. The field spreads out radially, interacting with the dielectric, causing the reflected and transmitted fields to interact. The source pulse and electric field at the source point shown in the top left corner does not appear to change as the source becomes constant at high time steps.

V. CONCLUSION

The findings in this paper demonstrate the effectiveness and adaptability of the Finite-Difference Time-Domain method and showcase the significant enhancements achievable through computational optimizations like Numba and parallel computing with MPI.

The application of the FDTD method, both in one-dimensional and two-dimensional domains, has confirmed its ability to model electromagnetic wave propagation through media. The detailed simulations involving dielectric films and boxes provided insightful observations on field behavior, including reflection, transmission, and the interactions within dielectric boundaries.

In addition, the integration of computational optimizations presented a significant improvement in the application of the FDTD method. The employment of Numba has demonstrated substantial reductions in computational time, thereby enhancing efficiency without compromising the accuracy of the simulations. Parallel computing, facilitated through MPI, has furthered the potential of the FDTD method. By distributing the computational workload across multiple processors, the simulations' scalability and speed have seen remarkable improvements.

In conclusion, this paper not only confirms the FDTD method's reliability in electromagnetic simulations but also highlights the beneficial impact of computational optimizations. The advancements in efficiency and scalability achieved by Numba and MPI demonstrate the remarkable improvements that can be applied to the FDTD algorithms.

[1] Stephen D. Gedney. *Introduction to the Finite-Difference Time-Domain Method for Electromagnetics*. Synthesis Lectures on Computational Electrodynamics. Springer Nature, May 2022.

[2] Stephen Hughes. PHYS479/879: FDTD Flux Density Formulation, 2d FDTD - TM Example, Electric Dipole Excitation, “Numba” and Slicing Speed Ups. Course notes, 2024. Department of Physics, Queen's University, Winter term.