

FRANKFURT UNIVERSITY OF APPLIED  
SCIENCES

MASTER THESIS

---

**Finite Difference Time Domain  
Method Implementation in C++**

---

*Author:*

Xhoni ROBO

*Supervisor:*

Prof. Dr. Peter THOMA

*Assistant Supervisor:*

Prof. Dr. Egbert

FALKENBERG

*A thesis submitted in fulfillment of the requirements  
for the degree of Master of Science*

*in*

High Integrity Systems

Faculty 2: Computer Science and Engineering

December 23, 2020

# Declaration of Authorship

I, Xhoni ROBO, declare that this thesis titled, “Finite Difference Time Domain Method Implementation in C++” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

---

Date:

---

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

# *Abstract*

Faculty 2: Computer Science and Engineering

Master of Science

**Finite Difference Time Domain Method Implementation in C++**

by Xhoni ROBO

This thesis is the final documentation for a project in the Master of Science degree of High Integrity Systems supervised by Prof. Dr. Peter Thoma and Prof. Dr. Egbert Falkenberg. In this thesis, the author will explain the basics of electromagnetic simulation and demonstrate a simple application that will produce both electric and magnetic field data, which can then be visualized through the help of third party applications such as Paraview<sup>[6]</sup>.

This thesis is heavily focused on theoretical aspect of such an application, and as such the code will be simplistic and not use any advanced external libraries not included by default in the basic C++ package. The application will not have a User Interface (UI), therefore the only way to customize the initial values of the code variables would be through an Integrated Development Environment (IDE) that can handle C++, such as Eclipse<sup>[3]</sup>. The benefit of not relying on any external open source libraries is the ability for this code to be used by any machine regardless of operating system and easy integration into applications that need such simulations.

Alongside this document, the project also included the code files found in the GitHub repository<sup>[8]</sup>. As the base L<sup>A</sup>T<sub>E</sub>X template of this thesis was found online<sup>[10]</sup>, these files are also included, with the license allowing viewing and modification so long as it is for a non-commercial use. After the official deadline of January 5th 2021, this project will be considered complete and no further changes will be made.

**Keywords:** *Finite, Time, Domain, Difference, C++, High, Integrity, Systems, 1D, 2D, 3D*

## *Acknowledgements*

First and foremost I would like to thank my academic supervisor, Prof. Dr. Peter Thoma. It is only due to his patience, perseverance, and willingness to spend his time aiding me, that I was able to complete this project. Even before that, I would like to thank him for his course of Simulation Methods in the High Integrity Systems M.Sc. degree, that convinced me that such a subject would make an interesting thesis for me.

I would also like to the Prof. Dr. Falkenberg, not only for his assistance throughout my higher academic studies, but also because I would not be able to officially start working on this thesis without his acceptance.

Thank you to my friends, who although far away have helped me keep my spirits high during a rather dark year not just for me, but for the world as a whole. It would have been difficult to push through to the end of this degree otherwise, if not impossible.

And lastly, my deepest thanks to my parents, who taught me discipline, and also acceptance. They did their best to set me up for a rich academic life, by straining themselves physically, mentally, and economically just so that I could have the best possibilities available to me. It will take an eternity to repay them back for their sacrifices, but hopefully this is, at the very least, a step in the right direction.

Sincerely,

Xhoni Robo

# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Electromagnetic Simulations . . . . .	1
1.1.1 Maxwell Equations . . . . .	2
1.1.2 Solving the Wave Equation . . . . .	5
1.2 Finite Difference Time Domain Method . . . . .	7
1.3 FDTD Implementation . . . . .	7
1.3.1 Project Plan, Requirements, and Tools Used . . . . .	8
1.3.2 Computational Limitations and Inaccuracies Explained	8
<b>2 FDTD - 1 Dimensional Scenario</b>	<b>10</b>
2.1 Main Section 1 . . . . .	10
2.1.1 Subsection 1 . . . . .	10
2.1.2 Subsection 2 . . . . .	10
2.2 C++ Implementation . . . . .	11
2.3 Data Visualization . . . . .	14
<b>3 FDTD - 2 Dimensional Scenario</b>	<b>16</b>
3.1 Main Section 1 . . . . .	16
3.1.1 Subsection 1 . . . . .	16

3.1.2 Subsection 2 . . . . .	16
3.2 C++ Implementation . . . . .	17
3.3 Data Visualization . . . . .	22
<b>4 FDTD - 3 Dimensional Scenario</b>	<b>23</b>
4.1 Main Section 1 . . . . .	23
4.1.1 Subsection 1 . . . . .	23
4.1.2 Subsection 2 . . . . .	23
4.2 C++ Implementation . . . . .	24
4.3 Data Visualization . . . . .	31
<b>5 Conclusion</b>	<b>32</b>
5.1 Main Section 1 . . . . .	32
5.1.1 Subsection 1 . . . . .	32
5.1.2 Subsection 2 . . . . .	32
5.2 Main Section 2 . . . . .	33
<b>A Frequently Asked Questions</b>	<b>34</b>
A.1 How do I change the colors of links? . . . . .	34
<b>Bibliography</b>	<b>35</b>

# List of Figures

1.1 Faraday's Experiment . . . . .	3
1.2 Magnetic Divergence . . . . .	4



# List of Tables

# List of Abbreviations

<b>FDTD</b>	Finite Difference Time Domain (Method)
<b>E</b>	Electric Field Intensity
<b>D</b>	Electric Displacement (Electric Divergence)
<b>H</b>	Magnetic Field Intensity
<b>B</b>	Magnetic Induction (Magnetic Divergence)
<b>J</b>	Current Density
$\rho$	Density of charge
<b>EMF</b>	Electromotive Force
<b>FEM</b>	Finite Element Method
<b>FIT</b>	Finite Integration Technique

# Physical Constants

Vacuum permeability  $\mu_0 = 1.256\,637\,062\,12(19) \times 10^{-6} \text{ H m}^{-1}$

Vacuum permittivity  $\epsilon_0 = 8.854\,187\,812\,8(13) \times 10^{-12} \text{ F m}^{-1}$

# List of Symbols

Symbol	Name	Unit of Measurement
$a$	distance	m
$P$	power	W (J s <sup>-1</sup> )
$\omega$	angular frequency	rad

# 1 Introduction

As humanity strives to better understand the world and universe around it, the physical limitations of our species become more and more apparent. While we have made considerable progress in our struggles to move forward, such as being able to record the movement of a light particle on camera despite it being the fastest moving object we know of so far<sup>[11]</sup>, or being able to capture an image of a black hole<sup>[4]</sup>, such achievements would not have been possible if our scientists did not have realistic expectations of how they should approach these challenges, or the expected results.

In order to achieve what they have, scientists needed to first understand the phenomena they were studying: the light having the particular properties of both particle and wave and the ability of black holes to distort space around them. All of this would not have been possible without simulations.

## 1.1 Electromagnetic Simulations

With the fast development of technology came new opportunities for gaining a better understanding of vast natural phenomena. We will be focusing on one of them, that being Electromagnetic Wave Propagation.

As one can imagine, analyzing electromagnetic fields through plain observation is near impossible, with only a few exceptions<sup>[1]</sup>. Even if we supposed that it was possible to easily achieve an acceptable amount of information from observing experiments, the cost and quality of the resulting data would

mostly be of scientific use, with little to no practical use whatsoever. Considering that electromagnetic waves are widely used in almost every industry, either as part of the building process or as a finished product, having data that cannot be used practically does not help.

That is why, thanks to the progress made in the computational capabilities of computers so far and the use of the theories and formulas gathered from past scientific endeavors, one can create data that is a close approximate of reality. Both shall be discussed in this chapter, but we cannot proceed without first going into what is believed by scientists to be the equations that govern large-scale electromagnetic phenomena<sup>[9]</sup>: Maxwell Equations.

### 1.1.1 Maxwell Equations

As mentioned above, Maxwell Equations are believed to dictate the behavior of all kinds of electromagnetic fields at a macroscopic level. These equations are the following:

$$\vec{\nabla} \times \vec{E}(\vec{r}, t) = -\frac{\partial \vec{B}(\vec{r}, t)}{\partial t} \quad (1.1)$$

$$\vec{\nabla} \times \vec{H}(\vec{r}, t) = \vec{J}(\vec{r}, t) + \frac{\partial \vec{D}(\vec{r}, t)}{\partial t} \quad (1.2)$$

$$\vec{\nabla} \cdot \vec{B}(\vec{r}, t) = 0 \quad (1.3)$$

$$\vec{\nabla} \cdot \vec{D}(\vec{r}, t) = \rho(\vec{r}) \quad (1.4)$$

To give a brief explanation over the meaning of each equation:

Equation 1.1 explains the effects of the electric field  $\vec{E}$  on the rate of change of the magnetic induction  $\vec{B}$ . This can also be referred to as the equation of electromagnetic induction, where the right hand side is the EMF or voltage and the left hand side is the magnetic flux. To those who study this particular

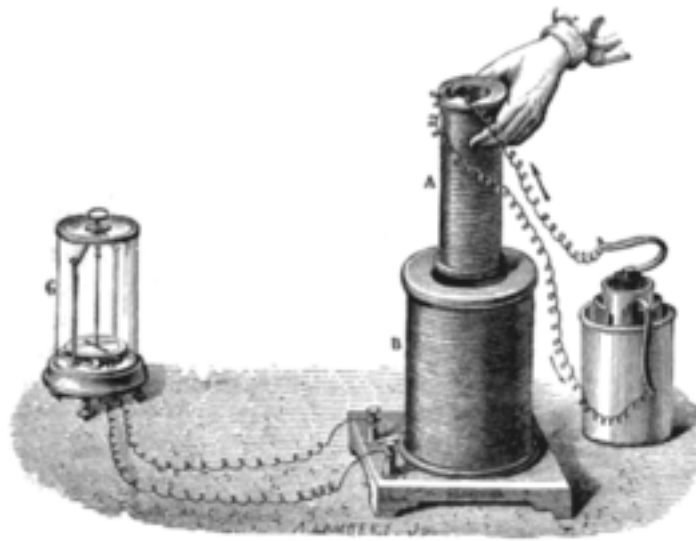


FIGURE 1.1: Faraday's Experiment,<sup>[7]</sup> which resulted in the law that was later used by Maxwell in making his equation.

area of physics, this equation will seem familiar, because it was derived from Faraday's law of induction.

Equation 1.2 is also known as Ampère–Maxwell law, because although it originated from Ampère, the current form was derived by Maxwell to include the magnetic current density  $\vec{j}$ . The equation explains the effects of the magnetic field on the electric current.

Equation 1.3, otherwise known as Gauss's law for magnetism, talks about the divergence of the magnetic field. According to this law the divergence is always 0. What this means is that in a magnetic field there is no such thing as a source (positive divergence) or a sink (negative divergence), rather a magnetic field can be more closely compared to a closed loop that flows in one direction. That is why every magnet that we know of has two poles. To translate this into something more easily understandable, it basically means that, if we were to pick any subset of the area of a magnetic field, no matter what area we pick, we would have vector fields going inside this area, and

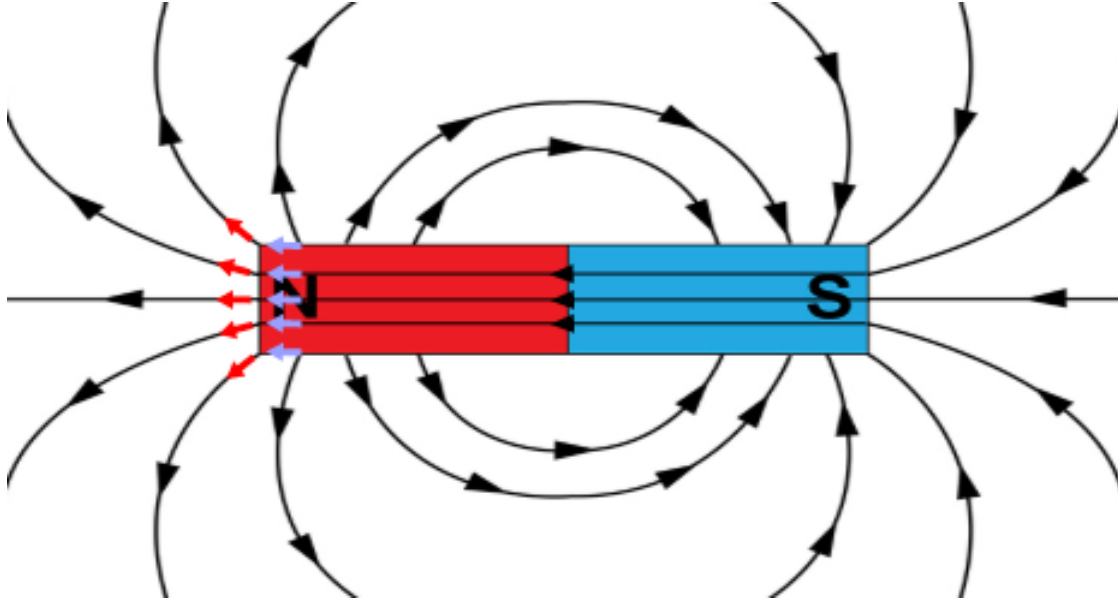


FIGURE 1.2: A crude representation of magnetic divergence through the use of a bar magnet.

vector fields going outside in equal number. A simple representation of this rule can be seen in 1.2, where it can be noted that the number of vectors heading towards the north pole are equal to the vectors going outside of it. Interestingly enough, if the bar magnet were to be cut in half, then the result would be two smaller bar magnets with 2 poles each and the exact same vector field.

Equation 1.4, is the main Gauss law for electric currents. It looks rather similar to his law of magnetism on the left hand side; the previous magnetic divergence is now replaced with the electric divergence. The bigger difference is the  $\rho$  on the right hand side, which is the density of charge of the electric field. In basic terms, it means that the divergence of the electric field is equal to the charge density for that point.

For the above equations, we also have the following material relations which will be useful later on:

$$\vec{D}(\vec{r}, t) = \epsilon(\vec{r}) \cdot \vec{E}(\vec{r}, t) \quad (1.5)$$



$$\vec{B}(\vec{r}, t) = \mu(\vec{r}) \cdot \vec{H}(\vec{r}, t) \quad (1.6)$$

The equations above are shown in their derivative form, but they can also be shown as their integral equivalent. There is no difference in implementation, regardless of which form is used.

### 1.1.2 Solving the Wave Equation

Using the formulas above, we can derive the electromagnetic wave equation, which is needed to proceed with the implementation further. Firstly for convenience, we will assume that the environment is the vacuum of space. Secondly, we will also assume that there is no charge in this space. This means that  $\epsilon(\vec{r}) = \epsilon_0$  and  $\mu(\vec{r}) = \mu_0$ . This allows us to simplify the above equations and give us the numerical solution for the wave equation

$$\Delta \vec{E}(\vec{r}, t) - \frac{1}{c^2} \frac{\partial^2 \vec{E}(\vec{r}, t)}{\partial t^2} = \mu_0 \frac{\partial \vec{J}(\vec{r}, t)}{\partial t}, \quad (1.7)$$

where:

$$c = \frac{1}{\sqrt{\mu_0 \epsilon_0}}$$

This equation is useful because it will be used to derive the formulas that are going to be needed for the simulations. Going forwards, this will be used as a basis to adapt our solution to every environment, regardless of dimensionality. From this point, there are many ways to proceed. Some of the most notable are the Finite Element Method (FEM), the Finite Integration Technique (FIT), and the Finite Difference Time Domain Method (FDTD). All of them are also known as Approximation Methods.

FEM is a well known numerical method used to obtain an approximation for a given boundary value problem. The basic principle is to divide a system into smaller subsets, called finite elements (hence the name), which are

much simpler to solve. This is done by discretizing the given space for each of its dimensions, and then constructing a mesh of the object. As a result, from the initial boundary value problem we get a system of equations that are further used to approximate each singular simplified function over the given domain. These equations are then compiled together into a system of equations that is then used to model the initial problem. The solution is then approximated by solving this system and minimizing the error function.<sup>[5]</sup>

The finite integration technique (FIT) is a bit more straightforward. It can help numerically solve electromagnetic field problems by discretizing in both the time and frequency domain. The first to introduce this technique was Thomas Weiland in 1977.<sup>[12]</sup> It has later seen continuous improvements and can now cover all electromagnetic problems and applications. This approach works by using the Maxwell equations above and applying their integral form to a set of staggered grids (e.g. Cartesian grid). This allows for a memory efficient implementation as well as giving the ability to handle different boundary conditions and variable material properties.

Finally we have a well known computational electromagnetic technique used for approximation, the Finite Difference Time Domain Method. It is arguably the easiest method out of the three to understand and implement, which is surprising when considering the capabilities it has in solving wave equations. This simplicity is also the reason it was chosen for this thesis, as it is the only technique that one person can realistically implement by themselves in a reasonable time frame.<sup>[2]</sup> As the name implies this is a time-domain method, meaning that a wide range of frequencies can be covered with a single simulation run. The only caveat is that the time step needs to be small enough to not cause any instabilities in the system.

## 1.2 Finite Difference Time Domain Method

FDTD belongs in the general class of grid-based differential time-domain numerical modeling methods. Maxwell's equations (in partial differential form) are modified to central-difference equations, discretized, and implemented in software. The equations are solved in a cyclic manner: the electric field is solved at a given instant in time, then the magnetic field is solved at the next instant in time, and the process is repeated over and over again.

The basic FDTD algorithm traces back to a seminal 1966 paper by Kane Yee in IEEE Transactions on Antennas and Propagation. Allen Taflove originated the descriptor "Finite-difference time-domain" and its corresponding "FDTD" acronym in a 1980 paper in IEEE Trans. Electromagn. Compat.. Since about 1990, FDTD techniques have emerged as the primary means to model many scientific and engineering problems addressing electromagnetic wave interactions with material structures. An effective technique based on a time-domain finite-volume discretization procedure was introduced by Mohammadian et al. in 1991.[12] Current FDTD modeling applications range from near-DC (ultralow-frequency geophysics involving the entire Earth-ionosphere waveguide) through microwaves (radar signature technology, antennas, wireless communications devices, digital interconnects, biomedical imaging/treatment) to visible light (photonic crystals, nanoplasmonics, solitons, and biophotonics). Approximately 30 commercial and university-developed software suites are available.

## 1.3 FDTD Implementation

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien,

in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

### 1.3.1 Project Plan, Requirements, and Tools Used

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

### 1.3.2 Computational Limitations and Inaccuracies Explained

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus

ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

## 2 FDTD - 1 Dimensional Scenario

### 2.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

#### 2.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

#### 2.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec

orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 2.2 C++ Implementation

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate.

```
#define _USE_MATH_DEFINES

#include <iostream>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <cmath>
#include <vector>
#include <string>

using namespace std;

const double permitivity = 8.854e-12;
const double permeability = 1.256e-6;
```

```
double L = 5;
int N = 200;
int iterNum = 800;
//double deltaX = L / N;
//double deltaY = L / N;
double deltaZ = L / N;
double deltaT = (deltaZ * sqrt(permitivity*permeability));

// variables needed for Gaussian Pulse excitation
double eps = 1e-3;
double Teps = 50 * deltaT;
double beta = -(pow((2/Teps), 2) * log(eps));

vector<double> E;
vector<double> H;
vector<double> tE;
vector<double> tH;

int main()
{
    E.assign(N, 0);
    H.assign(N, 0);

    for(int i = 0; i < iterNum; i++) {

        double t = i * deltaT;
```



```

double gamma = Teps / 2;

E[0] = exp(-(beta * pow((t - gamma), 2)));
↪ //TO-DO: Gaussian excitation, alpha = 1,
↪ Teps = 50*deltaT, eps = 1e-3, t = i *
↪ deltaT

// loop for values
for (int z = 0; z < N-1; z++) {
    H[z] = H[z] - (deltaT / permeability /
        ↪ deltaZ) * (E[z] - E[z+1]);
}

for (int z = 1; z < N-1; z++) {
    E[z] = E[z] + (deltaT / permitivity /
        ↪ deltaZ) * (H[z] - H[z-1]);
}

// time graph
tE.push_back(E[100]);
tH.push_back(H[100]);
}

cout << "\n\ntE\n";

// print E values

```

```
    for (int n = 0; n < iterNum; n++) {  
        cout << to_string(tE[n]) + ",";  
    }  
  
    cout << "\n\ntH\n";  
  
    // print H values  
    for (int n = 0; n < iterNum; n++) {  
        cout << to_string(tH[n]) + ",";  
    }  
}
```

Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

## 2.3 Data Visualization

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus

justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

## 3 FDTD - 2 Dimensional Scenario

### 3.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

#### 3.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

#### 3.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec

orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 3.2 C++ Implementation

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate.

```
double L = 5;
int N = 200;
int iterNum = 800;
double deltaX = L / N;
double deltaY = L / N;
double deltaZ = L / N;
double deltaT = (deltaZ * sqrt(permitivity*permeability)
↪ * (1/sqrt(2))); // 1/C * 1/sqrt2 * deltaZ

// variables needed for Gaussian Pulse excitation
double eps = 1e-3;
double Teps = 50 * deltaT;
double beta = -(pow((2/Teps), 2) * log(eps));
```

```

vector<vector<double>> Ex(N, vector<double> (N, 0));
vector<vector<double>> Ey(N, vector<double> (N, 0));
vector<vector<double>> Hz(N, vector<double> (N, 0));

const string filePath = "./Out/";

void writeEDataToCsvFile(string filename,
    ↪ vector<vector<double>> Ex, vector<vector<double>>
    ↪ Ey){

    //      "x", "y", Ex, Ey
    //      0, 0, Ex[x, y], Ey[x, y]

    ofstream csvFile(filename);
    csvFile << "x,y,z,Ex,Ey\n";

    for (int x = 0; x < Ex[0].size(); x++) {
        for (int y = 0; y < Ex[x].size(); y++) {
            csvFile << to_string(x) + "," +
            ↪ to_string(y) + ",0," +
            ↪ to_string(Ex[x][y]) + "," +
            ↪ to_string(Ey[x][y]) + "\n";
        }
    }

    csvFile.close();
}

```

```
void writeHDataToCsvFile(string filename,
    ↪ vector<vector<double>> Hz){

    //      "x", "y", Hz
    //      0, 0, Hz[x,y]

    ofstream csvFile(filename);
    csvFile << "x,y,z,HZ\n";

    for (int x = 0; x < Hz[0].size(); x++) {
        for (int y = 0; y < Hz[x].size(); y++) {
            csvFile << to_string(x) + "," +
            ↪ to_string(y) + ",0," +
            ↪ to_string(Hz[x][y]) + "\n";
        }
    }

    csvFile.close();
}

int main()
{
    for(int i = 0; i < iterNum; i++) {

        double t = i * deltaT;
        double gamma = Teps / 2;
```

```

// reducing the magnitude since in free
↪ space
Hz[99][99] = exp(-(beta * pow((t -
↪ gamma), 2))) * 10e-4; //TO-DO:
↪ Gaussian excitation, alpha = 1,
↪ Teps = 50*deltaT, eps = 1e-3, t = i
↪ * deltaT

for (int i = 0; i < N-1; i++) {
    for (int j = 1; j < N-2; j++) {
        Ex[i][j] = Ex[i][j] +
            ↪ (deltaT /
            ↪ permitivity /
            ↪ deltaZ) * (Hz[i][j]
            ↪ - Hz[i][j-1]);
    }
}

for (int i = 1; i < N-2; i++) {
    for (int j = 0; j < N-1; j++) {
        Ey[i][j] = Ey[i][j] -
            ↪ ((deltaT /
            ↪ permitivity /
            ↪ deltaZ) * (Hz[i][j]
            ↪ - Hz[i-1][j]));
    }
}

```



```

        writeEDataToCsvFile((filePath +
        ↪ "E/E.csv." + to_string(i)), Ex, Ey);

        // loop for values
        for (int i = 0; i < N-1; i++) {
            for (int j = 0; j < N-1; j++) {
                Hz[i][j] = Hz[i][j] -
                ↪ ((deltaT /
                ↪ permeability /
                ↪ deltaZ) * (Ex[i][j]
                ↪ - Ex[i][j+1] +
                ↪ Ey[i+1][j] -
                ↪ Ey[i][j]));
            }
        }

        writeHDataToCsvFile((filePath +
        ↪ "H/H.csv." + to_string(i)), Hz);

    }

}

```

Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

### 3.3 Data Visualization

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

## 4 FDTD - 3 Dimensional Scenario

### 4.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

#### 4.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

#### 4.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec

orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 4.2 C++ Implementation

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia.

```
const double permitivity = 8.854e-12;
const double permeability = 1.256e-6;

double L = 5;
int N = 50;
int iterNum = 200;
double deltaX = L / N;
double deltaY = L / N;
double deltaZ = L / N;
double deltaT = (deltaZ * sqrt(permitivity*permeability) *
    ↪ (1/sqrt(3))); // 1/C * 1/sqrt2 * deltaZ

// variables needed for Gaussian Pulse excitation
double eps = 1e-3;
double Teps = 50 * deltaT;
double beta = -(pow((2/Teps), 2) * log(eps));
```

```

vector<vector<vector<double>>> Ex(N, vector<vector<double>>(N,
    ↪ vector<double>(N, 0)));
vector<vector<vector<double>>> Ey(N, vector<vector<double>>(N,
    ↪ vector<double>(N, 0)));
vector<vector<vector<double>>> Ez(N, vector<vector<double>>(N,
    ↪ vector<double>(N, 0)));
vector<vector<vector<double>>> Hx(N, vector<vector<double>>(N,
    ↪ vector<double>(N, 0)));
vector<vector<vector<double>>> Hy(N, vector<vector<double>>(N,
    ↪ vector<double>(N, 0)));
vector<vector<vector<double>>> Hz(N, vector<vector<double>>(N,
    ↪ vector<double>(N, 0)));

const string filePath = "./Out/";

void writeDataToCsvFile(string filename,
    ↪ vector<vector<vector<double>>> Vx,
    ↪ vector<vector<vector<double>>> Vy,
    ↪ vector<vector<vector<double>>> Vz){
    ofstream csvFile(filename);
    csvFile << "x,y,z,Vx,Vy,Vz\n";

    for (unsigned x = 0; x < Vx[0][0].size(); x++) {
        for (unsigned y = 0; y < Vy[x][0].size(); y++)
            ↪ {

```

```

        for (unsigned z = 0; z <
            ↪ Vz[x][y].size(); z++) {
                csvFile << x << "," << y << ","
                    ↪ << z << "," << Vx[x][y][z]
                    ↪ << "," << Vy[x][y][z] << ","
                    ↪ << Vz[x][y][z] << "\n";
            }
        }
    }

    csvFile.close();
}

int main()
{
    for(int i = 0; i < iterNum; i++) {

        double t = i * deltaT;
        double gamma = Teps / 2;

        // reducing the magnitude since in free space
        Ex[24][24][24] = exp(-(beta * pow((t - gamma),
            ↪ 2))) * 10e-4; //T0-D0: Gaussian
            ↪ excitation, alpha = 1, Teps = 50*deltaT,
            ↪ eps = 1e-3, t = i * deltaT
        //Ey[9][9][9] = exp(-(beta * pow((t - gamma),
            ↪ 2))) * 10e-4;
    }
}

```

```

//Ez[9][9][9] = exp(-(beta * pow((t - gamma),
↪ 2))) * 10e-4;

// loop for values
for (int i = 0; i < N-1; i++) {
    for (int j = 0; j < N-2; j++) {
        for (int k = 0; k < N-2; k++) {
            Hx[i][j][k] =
                ↪ Hx[i][j][k] +
                ↪ (deltaT /
                ↪ permeability /
                ↪ deltaZ) *
                ↪ (Ey[i][j][k+1] -
                ↪ Ey[i][j][k] -
                ↪ Ez[i][j+1][k] +
                ↪ Ez[i][j][k]);
        }
    }
}

for (int i = 0; i < N-2; i++) {
    for (int j = 0; j < N-1; j++) {
        for (int k = 0; k < N-2; k++) {

```

```

Hy[i][j][k] =
    ↪ Hy[i][j][k] +
    ↪ (deltaT /
    ↪ permeability /
    ↪ deltaZ) *
    ↪ (Ez[i+1][j][k] -
    ↪ Ez[i][j][k] -
    ↪ Ex[i][j][k+1] +
    ↪ Ex[i][j][k]);

    }

}

}

for (int i = 0; i < N-2; i++) {
    for (int j = 0; j < N-2; j++) {
        for (int k = 0; k < N-1; k++) {
            Hz[i][j][k] =
                ↪ Hz[i][j][k] +
                ↪ (deltaT /
                ↪ permeability /
                ↪ deltaZ) *
                ↪ (Ex[i][j+1][k] -
                ↪ Ex[i][j][k] -
                ↪ Ey[i+1][j][k] +
                ↪ Ey[i][j][k]);

            }

        }

    }
}

```



```

writeDataToCsvFile((filePath + "H/H.csv." +
    ↪ to_string(i)), Hx, Hy, Hz);

for (int i = 0; i < N-2; i++) {
    for (int j = 1; j < N-2; j++) {
        for (int k = 1; k < N-2; k++) {
            Ex[i][j][k] =
                ↪ Ex[i][j][k] +
                ↪ (deltaT /
                ↪ permittivity /
                ↪ deltaZ) *
                ↪ (Hz[i][j][k] -
                ↪ Hz[i][j-1][k] -
                ↪ Hy[i][j][k] +
                ↪ Hy[i][j][k-1]);
        }
    }
}

for (int i = 1; i < N-2; i++) {
    for (int j = 0; j < N-2; j++) {
        for (int k = 1; k < N-2; k++) {

```

```

Ey[i][j][k] =
    ↪ Ey[i][j][k] +
    ↪ (deltaT /
    ↪ permittivity /
    ↪ deltaZ) *
    ↪ (Hx[i][j][k] -
    ↪ Hx[i][j][k-1] -
    ↪ Hz[i][j][k] +
    ↪ Hz[i-1][j][k]);

    }

}

for (int i = 1; i < N-2; i++) {
    for (int j = 1; j < N-2; j++) {
        for (int k = 0; k < N-2; k++) {
            Ez[i][j][k] =
                ↪ Ez[i][j][k] +
                ↪ (deltaT /
                ↪ permittivity /
                ↪ deltaZ) *
                ↪ (Hy[i][j][k] -
                ↪ Hy[i-1][j][k] -
                ↪ Hx[i][j][k] +
                ↪ Hx[i][j-1][k]);

        }

    }

}

```

```
        writeDataToCsvFile((filePath + "E/E.csv." +  
        ↪ to_string(i)), Ex, Ey, Ez);  
    }  
}
```

Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

### 4.3 Data Visualization

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

## 5 Conclusion

### 5.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

#### 5.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

#### 5.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec

orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 5.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

# A Frequently Asked Questions

## A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

```
\hypersetup{urlcolor=red}, or
```

```
\hypersetup{citecolor=green}, or
```

```
\hypersetup{allcolor=blue}.
```

If you want to completely hide the links, you can use:

```
\hypersetup{allcolors=.}, or even better:
```

```
\hypersetup{hidelinks}.
```

If you want to have obvious links in the PDF but not the printed text, use:

```
\hypersetup{colorlinks=false}.
```

# Bibliography

- [1] JB Cao et al. "First results of low frequency electromagnetic wave detector of TC-2/Double Star program". In: (2005).
- [2] David B Davidson. *Computational electromagnetics for RF and microwave engineering*. Cambridge University Press, 2010.
- [3] Eclipse. URL: <https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-cc-developers>.
- [4] Elizabeth Landau. "Black Hole Image Makes History; NASA Telescopes Coordinated Observations". In: *Nasa.gov* (2019). Ed. by Sarah Loff. URL: [https://www.nasa.gov/mission\\_pages/chandra/news/black-hole-image-makes-history](https://www.nasa.gov/mission_pages/chandra/news/black-hole-image-makes-history).
- [5] Daryl L Logan. "First course in finite element method, si". In: *Mason, OH: South-Western, Cengage Learning* (2011).
- [6] ParaView. URL: <https://www.paraview.org/>.
- [7] Arthur William Poyser. *Magnetism and electricity: A manual for students in advanced classes*. Longmans, Green and Company, 1918.
- [8] Xhoni Robo. *XhRobo/Finite-Difference-Time-Domain-Method-Implementation-in-C-*. URL: <https://github.com/XhRobo/Finite-Difference-Time-Domain-Method-Implementation-in-C->.
- [9] Julius Adams Stratton. *Electromagnetic theory*. Vol. 33. John Wiley & Sons, 2007.
- [10] Vel, Steve R. Gunn, and Sunil Patel. "Masters/Doctoral Thesis LaTeX Template". In: *LaTeX Templates* (). URL: <https://www.latextemplates.com/template/masters-doctoral-thesis>.

- 
- [11] Andreas Velten et al. "Femto-photography: capturing and visualizing the propagation of light". In: (2013).
  - [12] T Weiland. "A discretization method for the solution of Maxwell's equations for six-component fields.-Electronics and Communication,(AEÜ), Vol. 31". In: (1977).