University of Utah
UNDERGRADUATE RESEARCH JOURNAL

NUMERICAL SIMULATION OF ELECTROMAGENTIC IONOSPHERE DISTURBANCES DETECTING EARTHQUAKE ACTIVITY

Audrey Madeline Rubart (Dr. Jamesina Simpson)
Department of Electrical and Computer Engineering

ABSTRACT

On March 4, 2020, DARPA released a call for research proposals that would fund projects "in the area of modeling, simulating, and experimentally observing transient disturbances (both mechanical and electromagnetic) in the Earth's atmosphere due to meteorological and geophysical sources" [1]. This research would directly apply to their AtmoSense (Atmosphere as a Sensor) project that wants to find novel methods for geolocation by taking advantage of measured disturbances in the Earth's atmosphere. A novel approach to geolocating earthquakes is by studying the interactions between the ionosphere and the disturbed electromagnetic fields from power lines during the event. This paper presents a finite-difference time-domain (FDTD) model that simulates a 2D cross section of a power line and surrounding space to showcase how the electric fields change during movement. A Discrete Fourier Transform is performed on the data to which shows the signals seen above the power line. The interactions between shaken power lines and the ionosphere could allow researchers to backtrack earthquake locations, giving them another way to passively measure earthquake phenomenon and understand how earthquakes affect different aspects of the environment.

TABLE OF CONTENTS

1. INTRODUCTION

On March 4, 2020, DARPA released a call for research proposals that would fund projects (starting in December of 2020 at the earliest) "in the area of modeling, simulating, and experimentally observing transient disturbances (both mechanical and electromagnetic) in the Earth's atmosphere (from the troposphere through the ionosphere) due to meteorological and geophysical sources" [1]. More specifically, DARPA is looking for research proposals that can look at disturbances in the atmosphere that can pinpoint locations on Earth where natural phenomena are occurring, such as tornadoes, hurricanes, or earthquakes.

This research would directly apply to DARPA's AtmoSense (Atmosphere as a Sensor) project that wants to find novel methods for geolocation by taking advantage of measured disturbances in the Earth's atmosphere. Specifically, AtmoSense wants to understand the fundamentals of the energy propagation by answering the following questions:

1) What is the nature of transmitted signals?
2) What mode structure (mechanical and electromagnetic) can the mesosphere and lower ionosphere support?
3) What dynamic variables are best measured and at what altitude to capture source disturbed information?

AtmoSense also identifies three technical areas that will try to answer these questions. This specific research project applies to technical area 1, the modeling and simulation division [1].

This project looks to answer this call for research by simulating the result of an earthquake disturbing power stations and power lines, which then create electromagnetic ionospheric disturbances. With passive detection in the ionosphere, researchers could see disturbances in the atmosphere and pinpoint where an earthquake occurred on the surface of the Earth. The proposed project will use finite-difference time-domain (FDTD) modeling techniques to simulate the changing electromagnetic waves caused by disturbed power lines. This project will also use discrete Fourier transforms to study the frequencies of those waves.

1.1 *THE IONOSPHERE AND IONOSPHERIC DATA COLLECTION*

The ionosphere is the section of the atmosphere that describes the electron content throughout it. It's measured in the Total Electron Content (TEC) which is measured in Total Electron Content Units (TECUs) [2]. Fig. 1 shows how the ionosphere permeates almost all layers of the atmosphere.

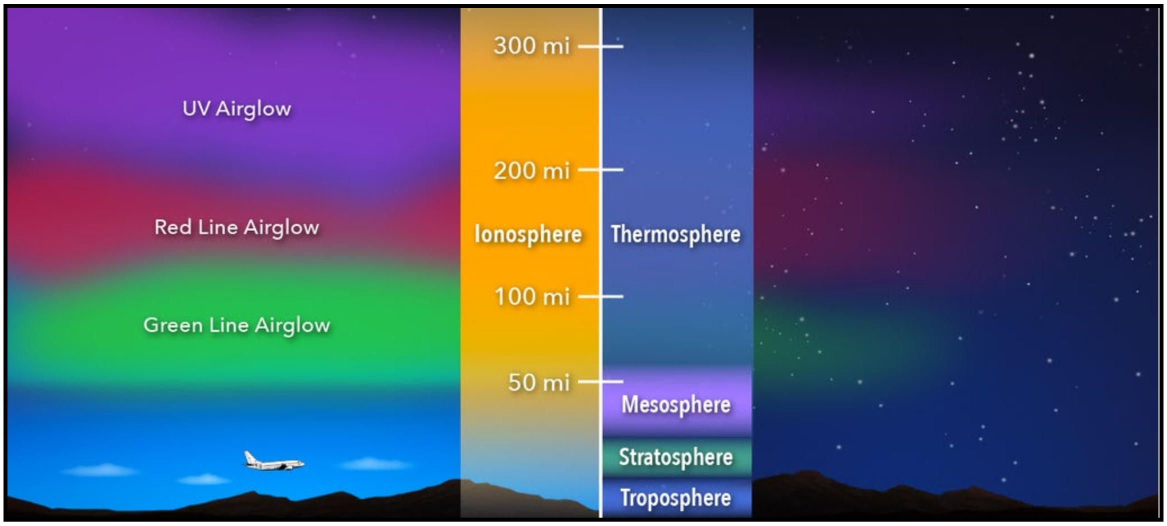


Fig. 1 Illustration showcasing the different layers of the atmosphere, including how the ionosphere permeates all layers within the atmosphere [2].

Ionosphere disturbances caused by natural disasters or other phenomenon is not new. One of the deadliest earthquakes on record, the 2011 Tohoku earthquake, had significant TECU perturbations during and after the earthquake. One way to demonstrate ionospheric perturbations is to correlate the events with the energetic magnitude in terms TNT explosive equivalents [1]. Some significant events with measured disturbances are found on Fig. 2.

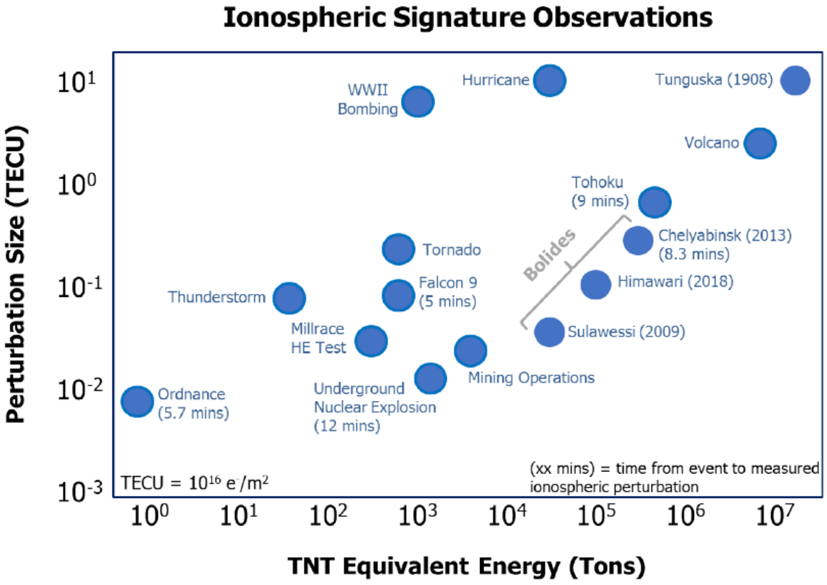


Fig. 2 Ionospheric Signature Observations for different events occurring on Earth [1].

Measurements for ionospheric perturbations can happen in a variety of ways. In a 2012 study, researchers measured ionospheric signatures caused by the 2011 Tohoku earthquake and tsunami using GPS receivers used in Japan's GEONET network [3]. Japan's GEONET network is widely known for being one of the densest GPS networks in the world and has the ability to measure TECUs. These receivers were able to make measurements of the disturbances before, during, and after the Tohoku earthquake [3].

A different study in 2012 also proved the usefulness of Japans GPS receivers for this kind of study. GPS receivers can not only measure the TECU disturbances but also the vertical and horizontal movement on the ground during quakes [4]. This is helpful for us because it can tell us accurate movement data to simulate for this project.

Measuring ionospheric changes due to power lines has also been studied. In 2015, a research group studied the impact on the ionosphere caused by the increase in the number of power lines, power plants, and industrial plants on Earth [5]. The ionosphere changes were measured using both ground observations and satellites that used highly sensitive magnetic noise detectors. The researchers found that monitoring the fields emitting from power lines and power stations could tell them about energy consumption around the globe [5]. We believe that we could use similar data and data collection techniques that could measure power line movement caused by earthquakes and their subsequent interactions in the ionosphere.

1.2 *EARTHQUAKE MOVEMENT – LOVE WAVES*

Based on data from the 2011 Tohoku earthquake, the majority of the seismic movement was caused by Love waves [4]. Love waves are one of two types of seismic surface waves that moves through the earth and shake the ground in a lateral motion [6]. In our simulation, the power line moves sideways only, with zero vertical movement, to match these Love waves.

Fig. 3 shows a sequence of a Love wave traveling through media [6]:

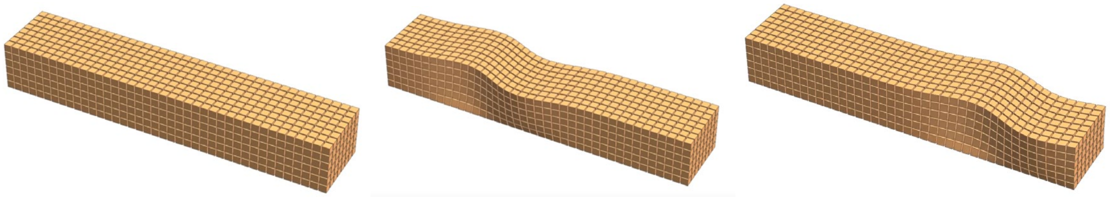
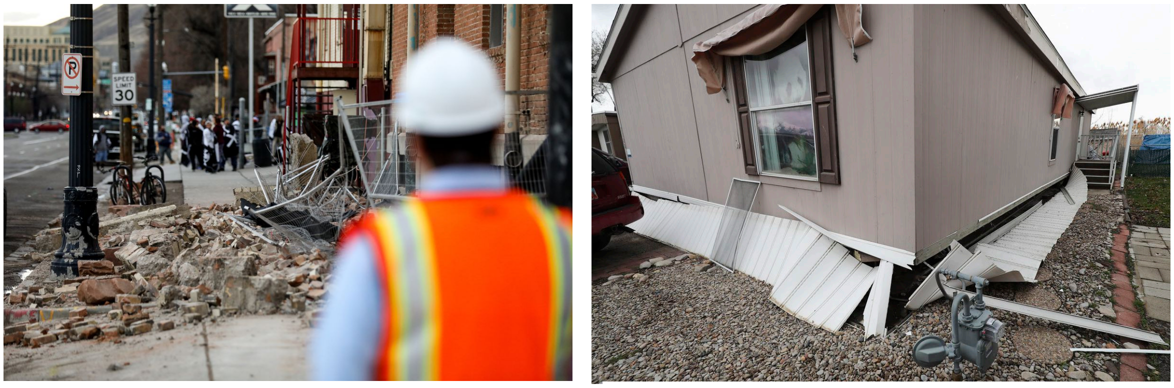


Fig. 3 Demonstration of Love wave traveling through a medium (modified from Chang) [6].

In the Methodology section of this paper, I explain how the shaking of the line is simulated according to this Love wave movement.

2. PURPOSE

On March 18, 2020, a 5.7 magnitude earthquake rocked the Salt Lake Valley. The epicenter was about a 45 minute drive away from downtown and caused widespread damage to many homes and businesses. Fig. 4 shows some photos taken immediately following the earthquake of the damage in downtown Salt Lake City [7].



(a) (b)

Fig. 4 Images of building damage from the 5.7 Magna Earthquake. (a) Damage to a building in Downtown Salt Lake City. (b) Mobile home in Magna moved off its foundation [7].

There were over a dozen aftershocks within the hour after the earthquake [7]. Some of the damage occurred at the University of Utah including a campus building getting exterior cracks and developing ceiling issues [8]. Kennecott Copper Mine, an open pit min on the edge of the Salt Lake Valley, had an acid spill which released a plume of acid into the air. Luckily, that spill was non-threatening to the public [8]. The Utah Department of Transportation closed a ramp entering Interstate 215 for “three to seven days until a full assessment [could] be completed” on the ramp, making it much more difficult for residents to commute across the valley [8].

However, a 5.7 is not the strongest earthquake that can happen on the Earth’s surface, giving rise to the possibility of even more damage to places like the Wasatch Front and beyond. According to the Working Group on Utah Earthquake Probabilities, “there was a 43% chance that the Wasatch Front will experience a large earthquake – magnitude 6.75 or greater – in the next 50 years.” [8] Residents in the Salt Lake Valley were extremely fortunate since no deaths or serious injuries were reported, but that may not be the case in the next quake. Any new type of data that we can collect regarding natural disasters like earthquakes and their widespread effects could be enlightening and should be studied.

3. METHODOLOGY

In this section I will describe the formulation behind the Finite-Difference Time-Domain method used in this simulation. This includes the derivation of the method starting with Maxwell's Equations, how we discretize Maxwell's Equations to be solved in a computer, different limitations within the Finite-Difference Time-Domain method, and how we apply this method to this specific simulation problem.

3.1 *MAXWELL'S EQUATIONS*

Understanding the Finite-Difference Time-Domain method first starts with reviewing the derivation of Maxwell's Equations. Electromagnetics can be summarized by understanding that any electric charge will induce an electric field and any electric current will induce a magnetic field [9]. Those two fields are coupled and will move through space as electromagnetic waves. Any change in one field, in space or time, will change the overall wave [9]. Maxwell's Equations describe the time varying electromagnetic space for any electronics problem and are necessary to describe any radiation from any source. Maxwell's Equations are described as four fundamental equations, which are [10]:

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E} - \boldsymbol{M} \tag{1}$$

$$\frac{\partial}{\partial t}\iint_{A} \boldsymbol{B}\cdot d\boldsymbol{A} = -\oint_{L} \boldsymbol{E}\cdot d\boldsymbol{L} - \iint_{A} \boldsymbol{M}\cdot d\boldsymbol{A} \tag{2}$$

$$\frac{\partial \boldsymbol{D}}{\partial t} = \nabla \times \boldsymbol{H} - \boldsymbol{J} \tag{3}$$

$$\frac{\partial}{\partial t}\iint_{A} \boldsymbol{D}\cdot d\boldsymbol{A} = -\oint_{L} \boldsymbol{H}\cdot d\boldsymbol{L} - \iint_{A} \boldsymbol{J}\cdot d\boldsymbol{A} \tag{4}$$

Equations 1 and 2 are Faraday's law and Equations 3 and 4 are Ampere's Law. ***E*** represents the electric field, ***H*** represents the magnetic field, ***D*** represents the electric flux density, ***B*** represents the magnetic flux density, and ***J*** represents the electric current density. ***M*** represents the magnetic current density, which is mathematically analogous to the electric current density ***J***. Note that the magnetic current density does not really exist in the physical world but will ease in calculations for Finite-Difference Time-Domain methods shown later.

Gauss' Laws are also important to note as they provide special circumstances in electromagnetics which will help simplify the above four equations. Gauss' Laws for the electric field are [10]:

$$\nabla \cdot \boldsymbol{D} = 0 \tag{5}$$

$$\oiint_A \boldsymbol{D} \cdot d\boldsymbol{A} = 0 \tag{6}$$

Gauss' Laws for the magnetic field are [10]:

$$\nabla \cdot \boldsymbol{B} = 0 \tag{7}$$

$$\oiint_A \boldsymbol{B} \cdot d\boldsymbol{A} = 0 \tag{8}$$

In this project the materials used are considered isotropic and nondispersive. In that case, ***D*** and ***B*** can be related to ***E*** and ***H***, respectively, in the following way [10]:

$$\boldsymbol{D} = \varepsilon \boldsymbol{E} = \varepsilon_r \varepsilon_0 \boldsymbol{E} \tag{9}$$

$$\boldsymbol{B} = \mu \boldsymbol{H} = \mu_r \mu_0 \boldsymbol{H} \tag{10}$$

where ε represents the electrical permittivity and μ represents the magnetic permeability [10]. The current density and magnetic current density are also considered independent from ***E*** and ***H*** respectively since they are defined as independent sources. Equations for the electric and magnetic current densities can be re-written as follows [10]:

$$\boldsymbol{J} = \boldsymbol{J_{source}} + \sigma \boldsymbol{E} \tag{11}$$

$$\boldsymbol{M} = \boldsymbol{M_{source}} + \sigma^* \boldsymbol{H} \tag{12}$$

We can use these equations to simplify Maxwell's Equations so they can be eventually used for the Finite-Difference Time-Domain method. Using the conditions described by Gauss' Laws in Equations 5 – 8 and substituting Equations 9 – 12 into Equations 1 and 3, we eventually end up with the curl equation of Maxwell's Equations for linear, isotropic, nondispersive medium [10]:

$$\frac{\partial \boldsymbol{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \boldsymbol{E} - \frac{\boldsymbol{1}}{\boldsymbol{\mu}} (\boldsymbol{M_{source}} + \sigma^* \boldsymbol{H}) \tag{13}$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\varepsilon} \nabla \times \boldsymbol{H} - \frac{1}{\varepsilon} (\boldsymbol{J_{source}} + \sigma \boldsymbol{E}) \tag{14}$$

These equations are the most useful when solving large electromagnetic problems and form the basis for Finite-Difference Time-Domain method. Next, we will simplify Maxwell's Equations into 1 and 2 dimensions.

3.2 *MAXWELL'S EQUATIONS IN 1-DIMENSION AND 2-DIMENSIONS*

3.2.1 *Derivation of Maxwell's Equations for 1-Dimension*

Maxwell's Equations can be simplified by looking at the equations in only one dimension. This can be helpful if we want to solve Maxwell's Equations in 1D situations, such as determining the voltage and current that runs through a transmission line.

First, let's expand Maxwell's Equations from the curl definition into their attributed partial differentials [10]:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - (M_{source_x} + \sigma^* H_x)\right] \tag{15}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - (M_{source_y} + \sigma^* H_y)\right] \tag{16}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - (M_{source_z} + \sigma^* H_z)\right] \tag{17}$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - (J_{source_x} + \sigma E_x)\right] \tag{18}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - (J_{source_y} + \sigma E_y)\right] \tag{19}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - (J_{source_z} + \sigma E_z)\right] \tag{20}$$

We consider waves that move in one dimension to be Transverse Electromagnetic Waves, or TEM Waves [11]. TEM waves are considered for many flat waveguide applications such as microstrip transmission lines. Standard convention states that the direction of propagation is in the z direction with uniformity in the y direction. For the TEM standard convention, we can therefore assume that partial y and z derivatives equal zero [10]. TEM is also defined where $E_z = 0$ and $H_z = 0$, so any source in those directions will equal zero as well. For the one dimension case, any partial derivative that is dependent on z or y is equal to zero, drastically simplifying the six equations above to three [10].

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_z}{\partial x} - \left(M_{source_y} + \sigma^* H_y\right)\right] \tag{21}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_y}{\partial x} - \left(J_{source_z} + \sigma E_z\right)\right] \tag{22}$$

We will use these two equations when generating equations for the Finite-Difference Time-Domain method.

3.2.2 *Derivation of Maxwell's Equations in 2-Dimensions*

As we did for Maxwell's Equations in 1D, we will derive them again for Maxwell's Equations in 2D. First, recall Equations 15 through 20. Just as the 1-dimensional case is described by the Transverse Electromagnetic Mode (TEM), our 2-dimensional case will be described by the Transverse Magnetic Mode (TM) [11]. The Transverse Magnetic Mode is characterized by $E_z \neq 0$ and $H_z = 0$, so the electric field propagates in the z-direction. The TM mode is also considered uniform in the z-direction, so any partial differential with respect to z or any magnetic source in the z direction within Maxwell's Equations will become zero. We will end up with the following three equations [10]:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left[-\frac{\partial E_z}{\partial y} - \left(M_{source_x} + \sigma^* H_x\right)\right] \tag{23}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_z}{\partial x} - \left(M_{source_y} + \sigma^* H_y\right)\right] \tag{24}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \left(J_{source_z} + \sigma E_z\right)\right] \tag{25}$$

A two-dimensional model of Maxwell's Equation is a little more complicated than the 1-directional case. A good example of a 2-dimensional problem builds off our transmission line example used previously. Instead of focusing on the electromagnetic fields within the line, we'll focus on the fields outside the line. We can assume again that structure extends to infinity along the z-axis with uniform waves. However, there are variations in both the x and y axes as the electromagnetic field propagates away perpendicular to the line. We use the TM mode definition for this problem because it does a good job describing problems describing radiation [10].

We will use the above three equations to derive the Finite-Difference Time-Domain method for Maxwell's Equations in 2-dimensions.

3.3 *INTRODUCTION TO FINITE-DIFFERENCE TIME-DOMAIN METHOD*

Maxwell's Equations are difficult to calculate over a large, continuous range of points in space. For larger scale applications, this requires the use of computers to quickly calculate the equations for that large space. However, Maxwell's Equations are partial differential equations that expand over continuous space rather than discretized points. In order for the computer to quickly solve Maxwell's Equations we must provide a discretized version of the equations.

Discretization of mathematical formulae is not new or novel by any means. A commonly recognized discretized mathematical formula is the use of Riemann Sums when computing integrals. Riemann Sums easily break up the area under a curve into discrete rectangles or trapezoids to approximate the area. The more rectangles or trapezoids that are used, the more exact the answer is. Fig. 5 shows an example of Reimann Sums using trapezoids to find the area under a curve.

The Finite-Difference Time-Domain method can be considered analogous to the method of Riemann Equations: as Riemann Equations approximate the integral of a curve, the Finite-Difference Time-Domain method approximate Maxwell's Equations over a large area. Finite-Difference Time-Domain, commonly known as FDTD, is used in electromagnetic computations in some engineering CAD software.

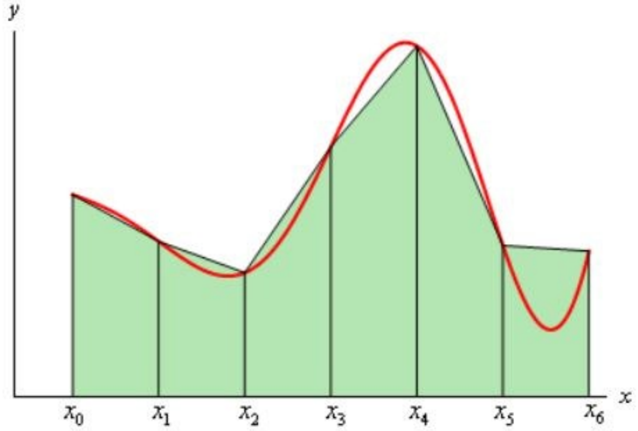


Fig. 5 The area under a curve computed using a Riemann Sums technique that adds discrete trapezoids. The more trapezoids that are used under the curve, the more accurately the area under is calculated [12].

3.3.1 *The Central Difference Theorem*

FDTD can be used in one, two, and three dimensions. For this research project, only the one and two dimension cases are studied. For ease of understanding, let's begin with discussion of the 1D case. Let's return to the transmission line example. We can use FDTD to solve the values of the electric and magnetic fields along that line at any discretized point along it. If we use many discretized points, the better our analysis is going to be.

Recall the equations for electric and magnetic fields from Maxwell's Equations for the 1 dimensional case, which are listed as Equations 21 and 22. The partial differential equations given are coupled together and must be solved simultaneously – however, computers cannot solve partial differential equations like this.

What computers can solve are basic arithmetic equations, similar to the Riemann Sums. A common way to arithmetically solve partial derivatives is the Central Difference Theorem, which states [13]:

$$\frac{\partial u}{\partial t} \approx \frac{u\left(x, t + \frac{\Delta t}{2}\right) - u\left(x, t - \frac{\Delta t}{2}\right)}{\Delta t} + O[(\Delta t)^2] \qquad (26)$$

where the O term arises from the Taylor series expansion of $u(x, t + \Delta t)$ [13]. FDTD uses the Central Difference Theorem to approximate the partial derivatives in Maxwell's Equations, which can be applied to any dimension.

The Central Difference Theorem, however, does not entirely solve our coupled equation issue. To solve this, we can choose specific geometry to solve the Central Difference Theorem that more or less allows us to solve the equations simultaneously. The geometry we use is based upon the Yee Grid [10].

3.4 *THE YEE GRID*

In 1966, Kane Yee developed a geometric algorithm that "robustly represents both the differential and integral forms of Maxwell's Equations." [10] Yee's geometry in 1D samples the electric field at every discrete point and the magnetic field at every half discrete point. In a discretized sense, the electric field can be solved by using the magnetic field points that surround it and vice versa. A visual of the three dimensional Yee grid is given in Fig. 6 [10].

Using the combination of the Central Difference Theorem and the Yee Grid allows us to solve Maxwell's Equations numerically. The neighboring values on the Yee Grid provide which values to use when solving the coupled Maxwell's Equations in the Central Difference Equation. This method is called "leapfrogging" and allows us to solve coupled equations.

Fig. 7 shows the Yee Grid implemented in 1-dimension, which will aid in the discussion about leapfrogging. In the Yee Grid, the electric and magnetic fields are staggered against each other by one half step in time and space. The electric fields are solved on integer values on the number line, and the magnetic fields are solved on integer + 0.5 values on the number line. Then, each value of either the electric field or magnetic field is solved after every half time step, each field taking turns and being updated over time. This leapfrogging effect allows us to solve Maxwell's Equations simultaneously in a discretized fashion [10].

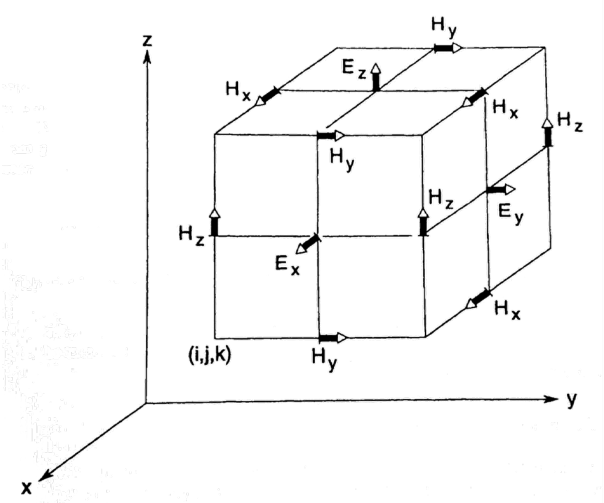


Fig. 6 3-dimensional Yee Grid, demonstrating electric and magnetic field components at discretized points in space [10].

3.5 *FINITE-DIFFERENCE TIME-DOMAIN METHOD EQUATIONS*

3.5.1 *Finite-Difference Time-Domain Notation*

Since Maxwell's Equations can be dependent upon four variables at once, it's necessary to create easier notation to reference which value of time and which point in space we are referencing at any given time on the Yee Grid. As a general case, we will discuss the notation in three dimensions. Derivation for the first and second dimension will be trivial once the derivation for the third dimension is explained.

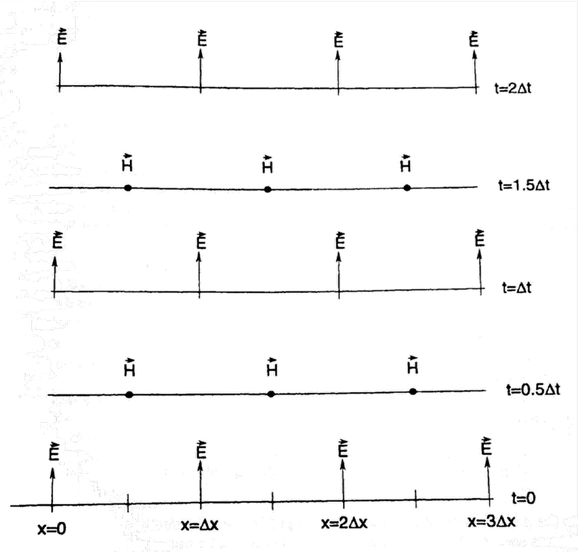


Fig. 7 Demonstration of leapfrogging effect on the electric and magnetic fields. Electric fields are solved for integer time and space steps, and magnetic fields are solved for integer + 0.5 time and space steps [10].

In 3-dimensions our plane exists in the (i, j, k) vector space. Since we are discretizing our space in terms of Δx, Δy, and Δz, respectively, we will consider our vector space in terms of $(i\Delta x, j\Delta y, k\Delta z)$. Since we are also discretizing time as Δt, any point in time will be considered as $(n\Delta t)$. Overall, any 3-dimensional variable that is also dependent in time will be described with the following notation [10]:

$$u(i\Delta x, j\Delta y, k\Delta z, n\Delta t) \; = \; u_{i,j,k}^{n} \tag{27}$$

Trivially, the first-dimension and second-dimension cases, respectively, are as follows:

$$u(i\Delta x, n\Delta t) \; = \; u_i^n \tag{28}$$

$$u(i\Delta x, j\Delta y, n\Delta t) \; = \; u_{i,j}^n \tag{29}$$

3.5.2 *Finite-Difference Time-Domain in 1-Dimension*

When we apply the Central Difference Theorem described in Equation 26, the geometry of the Yee Grid, and use the notation given in Equation 28, we can discretize the 1-dimensional Maxwell's Curl Equations [14].

$$\frac{Ez_i^{n+1} \; - \; Ez_i^n}{\Delta t} = \frac{Hy_i^n \; - \; Hy_{i-1}^n}{\varepsilon_i \Delta x} - \frac{\sigma_i}{2\varepsilon_i}(Ez_i^n \; + \; Ez_i^{n-1}) \; - \; J_i^k \tag{30}$$

$$\frac{Hy_i^n \; - \; Hy_i^{n-1}}{\Delta t} = \frac{Ez_{i+1}^n \; - \; Ez_i^n}{\mu_i \Delta x} - \frac{\sigma_i^*}{2\mu_i}(Hy_i^{n+1} \; + \; Hy_i^n) \; - \; M_i^k \tag{31}$$

We can group the constants in Equation 30 and Equation 31 to make simplifying and solving the equations easier. We define our constants to be [14]:

$$C_a \; = \frac{2\varepsilon_i \; - \sigma_i \Delta t}{2\varepsilon_i + \sigma_i \Delta t} \tag{32}$$

$$C_b = \frac{2\Delta t}{\Delta x(2\varepsilon_i + \sigma_i \Delta t)} \tag{33}$$

$$D_a \; = \frac{2\mu_i \; - \sigma_i^* \Delta t}{2\mu_i + \sigma_i^* \Delta t} \tag{34}$$

$$D_b = \frac{2\Delta t}{\Delta x(2\mu_i + \sigma_i^* \Delta t)} \tag{35}$$

If we use the constants defined above and solve for the staggered values of E and H, we will arrive at our final equations for the one dimensional Finite-Difference Time-Domain case [14]:

$$Ez_i^{n+1} = C_a Ez_i^n + C_b(Hy_i^n - Hy_{i-1}^n) - J_i^k \quad (36)$$

$$Hy_i^n = D_a Hy_i^{n-1} + D_b(Ez_{i+1}^n - Ez_i^n) - M_i^k \quad (37)$$

3.5.3 *Finite-Difference Time-Domain in 2-Dimensions*

We will apply the Yee Grid and the Central Difference Theorem to the 2-dimensional cases of Maxwell's Equations the same way we applied them to the 1-dimensional case. This will discretize the 2-dimension case of the Finite-Difference Time-Domain method and will give us the following three equations [10]:

$$\frac{Ez_i^{n+1} - Ez_i^n}{\Delta t} = \frac{Hy_i^n - Hy_{i-1}^n}{\varepsilon_i \Delta x} - \frac{\sigma_i}{2\varepsilon_i}(Ez_i^n + Ez_i^{n-1}) - J_i^k \quad (38)$$

$$\frac{Hx_i^n - Hx_i^{n-1}}{\Delta t} = \frac{Ez_{i+1}^n - Ez_i^n}{\mu_i \Delta y} - \frac{\sigma_i^*}{2\mu_i}(Hx_i^{n+1} + Hx_i^n) - M_i^k \quad (39)$$

$$\frac{Hy_i^n - Hy_i^{n-1}}{\Delta t} = \frac{Ez_{i+1}^n - Ez_i^n}{\mu_i \Delta x} - \frac{\sigma_i^*}{2\mu_i}(Hy_i^{n+1} + Hy_i^n) - M_i^k \quad (40)$$

Another quick trick we want to complete before we continue is breaking up the *E_z* field into two fields – this is called the split E_z field [10]. This mathematically won't make a difference in our derivation but is necessary when we apply the Berenger Perfectly Matched Layer (PML) boundary conditions later. Boundary conditions are described in section 4.6 of this paper.

The easiest way to differentiate between the two split E_z fields is to split them along the x and y component of the traveling wave. Our split wave formulation will look like this [10]:

$$E_z = E_{zx} + E_{zy} \quad (41)$$

When we calculate our field within the Finite-Difference Time-Domain loop, we will add our split fields as described in Equation 41 before updating our magnetic fields. That way we can still use the total E_z field to solve Equations 39 and 40. It's also important to note that the source value, *J*, is only added to the total value of the electric field when implemented in the FDTD code. *J* is left in the equations listed below for simplicity and mathematical symmetry.

Our four discretized Maxwell's Equations, including the split field formulation, will look like this [10]:

$$\frac{Ezx_i^{n+1} - Ezx_i^n}{\Delta t} = \frac{Hy_i^n - Hy_{i-1}^n}{\varepsilon_i \Delta x} - \frac{\sigma_i}{2\varepsilon_i}(Ezx_i^n + Ezx_i^{n-1}) - J_i^k \quad (42)$$

$$\frac{Ezy_i^{n+1} - Ezy_i^n}{\Delta t} = \frac{Hy_i^n - Hy_{i-1}^n}{\varepsilon_i \Delta x} - \frac{\sigma_i}{2\varepsilon_i}(Ezy_i^n + Ezy_i^{n-1}) - J_i^k \tag{43}$$

$$\frac{Hx_i^n - Hx_i^{n-1}}{\Delta t} = \frac{Ez_{i+1}^n - Ez_i^n}{\mu_i \Delta y} - \frac{\sigma_i^*}{2\mu_i}(Hx_i^{n+1} + Hx_i^n) - M_i^k \tag{44}$$

$$\frac{Hy_i^n - Hy_i^{n-1}}{\Delta t} = \frac{Ez_{i+1}^n - Ez_i^n}{\mu_i \Delta x} - \frac{\sigma_i^*}{2\mu_i}(Hy_i^{n+1} + Hy_i^n) - M_i^k \tag{45}$$

As we did in the 1-dimension case, we can simplify Equations 42 through 45 by grouping constant values as follows [10]:

$$C_{ax} = C_{ay} = \frac{2\varepsilon_i - \sigma_i \Delta t}{2\varepsilon_i + \sigma_i \Delta t} \tag{46}$$

$$C_{bx} = C_{by} = \frac{2\Delta t}{\Delta x(2\varepsilon_i + \sigma_i \Delta t)} \tag{47}$$

$$D_{ax} = D_{ay} = \frac{2\mu_i - \sigma_i^* \Delta t}{2\mu_i + \sigma_i^* \Delta t} \tag{48}$$

$$D_{bx} = D_{by} = \frac{2\Delta t}{\Delta x(2\mu_i + \sigma_i^* \Delta t)} \tag{49}$$

Note that we know have constants for all the fields and their associated x and y components. Even though they are equivalent now, this distinction will be important later when discussing boundary conditions.

Using the constants defined above and using the split field formulation, we can solve for our final equations that we will use for the 2-dimensional case for the Finite-Difference Time-Domain Method [10]:

$$Ezx_i^{n+1} = C_{ax} Ezx_i^n + C_{bx}(Hy_i^n - Hy_{i-1}^n) - J_i^k \tag{50}$$

$$Ezy_i^{n+1} = C_{ay} Ezy_i^n + C_{by}(Hy_i^n - Hy_{i-1}^n) - J_i^k \tag{51}$$

$$Hy_i^n = D_{ay} Hy_i^{n-1} + D_{by}(Ez_{i+1}^n - Ez_i^n) - M_i^k \tag{52}$$

$$Hx_i^n = D_{ay} Hx_i^{n-1} + D_{by}(Ez_{i+1}^n - Ez_i^n) - M_i^k \tag{53}$$

3.5.4 *The Courant Limit*

The Finite-Difference Time-Domain method works well for almost any electromagnetics application. However, we must be mindful of limitations within the math, so the simulation works properly.

The most common way for FDTD to be implemented is to use a loop that solves the different values of the electric and magnetic field iteratively. Since we discretized the value of ∂t we need to give it a defined value.

FDTD relies upon the idea of leap-frogging so each value of the electric and magnetic field is solved and updated to help solve the next values of the electric and magnetic fields. If we solve the equations and update too quickly, our simulation becomes unstable. The limit that describes how quickly our loops can be updated is called the Courant Limit.

The Courant Limit is defined as [10]:

$$\Delta t < \frac{\Delta x}{c\sqrt{2}} \tag{54}$$

where c is defined by the speed of wave propagation. If this is through air or a vacuum, c is equal to the speed of light.

If the Courant Limit is exceeded and the simulation updates too quickly, the simulation “explodes” and becomes incredibly unstable. Fig. 8 shows a 1-dimensionsal FDTD example and demonstrates how the simulation becomes unstable when the Courant Limit is exceeded.

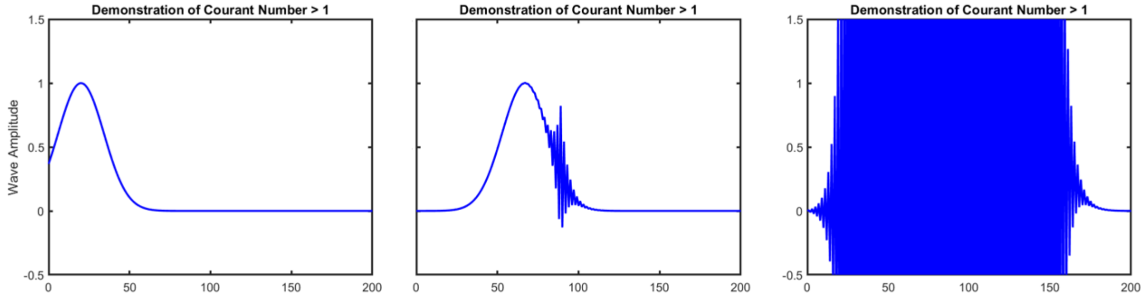


Fig. 8 1-dimensional FDTD simulation where the Courant Limit is exceeded. The simulation explodes and is unstable.

3.6 *BOUNDARY CONDITIONS*

In FDTD we are limited in scope on what we can simulate because our computing abilities are not infinite and we have to set boundaries. In this simulation, we want our boundaries to act like the boundaries of a windowpane – if we were to watch something occur outside, the windowpane limits how much we can see but has no effect on what happens. Similarly, our boundaries need to not affect what happens in the simulation.

There are two main kinds of boundary conditions that are used in this code. The first boundary condition is the Perfect Electric Conductor (PEC) boundary condition, and the second boundary condition we use is the Berenger Perfectly Matched Layer (PML) boundary condition.

3.6.1 *The Perfect Electric Conductor Boundary Condition*

A Perfect Electric Conductor (PEC) is described by any material that has an infinite conductivity associated with it. PECs are often used to approximate metals since they are extremely easy to work with. A PEC material means that no electric field can exist within it and any **E** field is reduced to zero.

In an FDTD code, the PEC can be easily added to any boundary condition by only allowing the **E** field along the edges to equal zero. This emulates an electromagnetic wave entering a PEC material and immediately attenuating to zero. This also means we will see reflections from the wave hitting the boundary the same way we would expect an electromagnetic wave reflecting off of any metal surface in real life [14].

A drawback to the PEC boundary condition is that the simulation window now performs like a cavity. If we want to simulate anything that extends beyond our simulation window or that does not perform like a PEC, we will need different boundary conditions.

3.6.2 *The Berenger Perfectly Matched Layer Boundary Condition*

The Berenger Perfectly Matched Layer (PML) Boundary Condition is a premium way to simulate our window without getting any unwanted reflections. The Berenger PML works to absorb the electric and magnetic fields that hit the edges of the simulation grid, so the simulation space is unaffected. This PML uses the split field formulation described in Section 4.5 to absorb any incident wave at any angle or magnitude. Reflections seen when using the Berenger PML are 1/3,000th of the reflection seen using other absorbing boundary conditions [10].

PML boundaries use thick layers of the simulation grid which slowly increase the conductivity values seen in the simulation space until the conductivity is large enough for the wave to be absorbed like it would in a PEC. The slow increase of the sigma values in the layer also avoids the reflections that we saw from the PEC boundary condition. The structure of a PML matches the structure of the split field FDTD equations so the waves are absorbed correctly.

Simulations with lower frequencies require thicker PMLs. High frequency applications, say in the 1 GHz range, can get away with PML layers that are around 10 grid cell lengths thick. However, since our simulation uses an extremely low frequency, around 60 Hz, we will need to use a PML that is around 100 grid cells thick.

Applying the PML on different edges of the simulation window will not be the same on all sides. Only certain sigma values will need to be increased along certain edges. This requires that our split field formulation have different x and y values for the electrical and magnetic conductivity. This differentiation is why we defined separate coefficient values for the x and y fields, even though at first glance they look the same.

Fig. 9 shows a graphical representation of the PML layers in a simulation window, and Fig. 10 demonstrates how the conductivity slowly increases in value to the final edge with the PEC boundary.

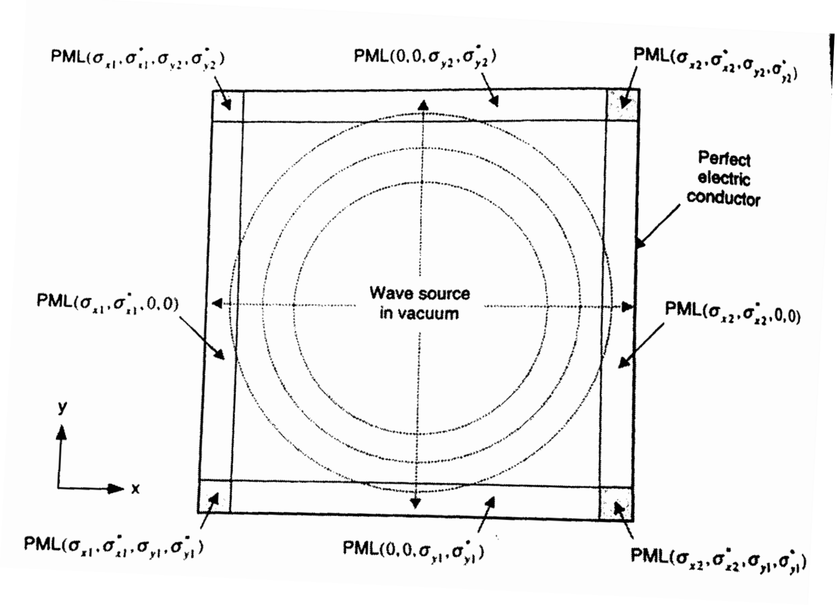

PML($\sigma_{x1}, \sigma^*_{x1}, \sigma_{y2}, \sigma^*_{y2}$)
PML($0, 0, \sigma_{y2}, \sigma^*_{y2}$)
PML($\sigma_{x2}, \sigma^*_{x2}, \sigma_{y2}, \sigma^*_{y2}$)
Perfect electric conductor
PML($\sigma_{x1}, \sigma^*_{x1}, 0, 0$)
Wave source in vacuum
PML($\sigma_{x2}, \sigma^*_{x2}, 0, 0$)
y
x
PML($\sigma_{x1}, \sigma^*_{x1}, \sigma_{y1}, \sigma^*_{y1}$)
PML($0, 0, \sigma_{y1}, \sigma^*_{y1}$)
PML($\sigma_{x2}, \sigma^*_{x2}, \sigma_{y1}, \sigma^*_{y1}$)

Fig. 9 Graphical representation of the Berenger Perfectly Matched Layer [10]

3.7 *PARALLEL CODE AND MESSAGE PASSING INTERFACE STANDARDS*

While the Finite-Difference Time-Domain method is a great tool for simulating electromagnetics problems, it is computationally expensive. The Courant Limit described earlier can create an issue if the limit is low. For example, if the Courant Limit is extremely small, on the order of nanoseconds, the simulation will need to update the fields over 200 million times to represent something happening less than 1 second of real time.

While a common tool for simulating FDTD includes MATLAB, MATLAB is too slow to efficiently compute a simulation space that requires millions of updates. For this project we investigated if using the supercomputers at the University of Utah Center for High Performance Computing would be a reliable option to simulate our problem more quickly. At first, the results from the supercomputer were compelling – a simulation that took over 12 hours in MATLAB only took around 45 minutes in the supercomputer. However, this approach was let go due to segmentation errors within the code that we could not resolve. That being said, a quick note on Parallel Code and Message Passing Interface Standards is still a worthy discussion since it is a promising avenue for projects like this one.

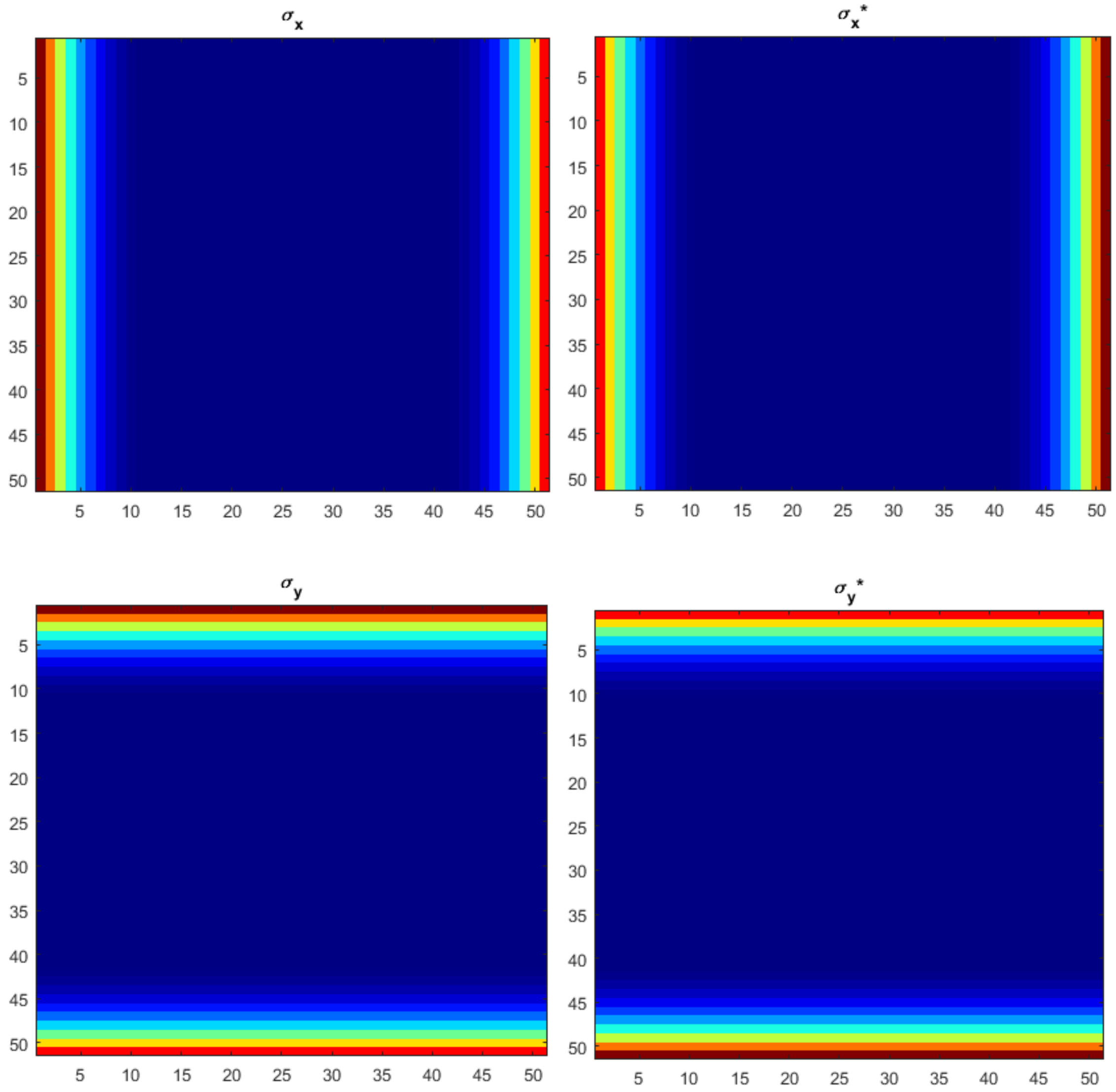


Fig. 10 Demonstration of the increased sigma values for the Berenger Perfectly Matched Layer. The PML is exaggerated so it is easy to see the values of sigma increase.

3.7.1 *Description of Parallel Computing and Message Passing Interface Standards*

In a nutshell, Parallel Computing is when multiple processors are working on a different portion of the same problem at the same time. This allows for the processor to solve a much smaller problem so the time for computation is slashed. However, as discussed in Section 4, the Yee grid geometry requires that neighboring values of the electric and magnetic fields be known so updates can be performed.

Message Passing Interface (MPI) standards fixes this problem. MPI standards often come as pre-programmed functions in lower level codes like Fortran. These functions describe how data and information are shared among processors. Fig. 11 showcases what parallel code and MPI standards may look like for a problem what was expanded from one processor to four processors.

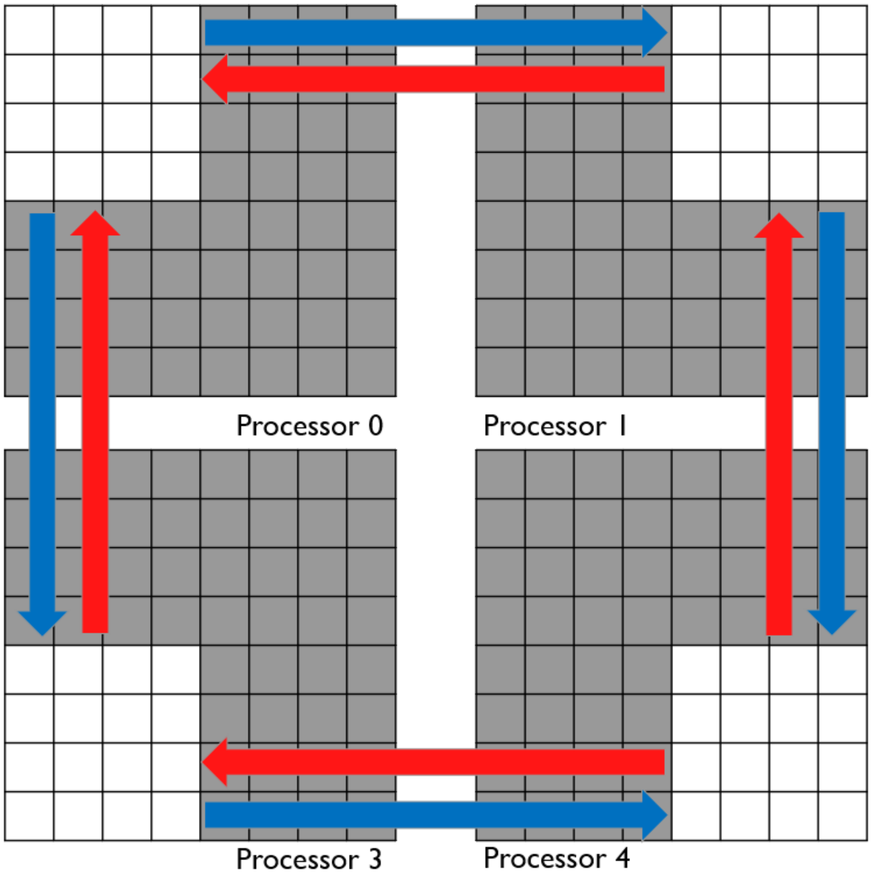


Fig. 11 Graphical description of parallel code and MPI standards. The original simulation grid is the space of one grid. Parallel code allocates the same grid space on each processor to solve the problem, but only performs the computations on its much smaller corner. The arrows on the grids represent the MPI standard functions sending information about the data along the grid edges to the other processors.

3.8 *APPLICATION OF FDTD FOR EARTHQUAKE SIMULATION*

To simulate a power line moving due to the earthquake, we will need to do three things. First, we will need to determine a way to simulate the movement of the line in a discretized way. Second, we will need to set boundary conditions that mirror what we would see for this scenario. Third, we will need to determine where we will observe the electromagnetic lines that are propagating away from the power line.

3.8.1 *Simulating Movement of Power Line*

We decided to model the movement of the power line as described in *Surface Waves of the 2011 Tohoku earthquake: Observations of Taiwan's dense high-rate GPS network* [4]. In the paper we are told the power line moves due to a Love Wave, is accelerated laterally at $a = 2.99g\ \frac{m}{s^2}$, and travels $d = 2.4\ m$. To make the movement computation easier, we decided to make our FDTD grid values equal half of that movement, or 1.2 meters.

If we assume the power line is starting from rest before being shaken by the earthquake, we also know that $v_0 = 0\ \frac{m}{s}$. Using these three knowns, we can use the Kinematic Equations from basic physics to solve for both the maximum velocity of the line and the total time that it will take to move the line [15].

Since the line starts at rest and the acceleration is constant, we know that the maximum velocity will occur at the mid-way point of travel, or at 1.2 meters. We can solve for the maximum velocity as follows:

$$v_{max} = \sqrt{v_0^2 + 2ad_{1/2}} = 8.386\frac{\mathrm{m}}{\mathrm{s}} \qquad (55)$$

Now knowing the maximum velocity, we can solve for the amount of time it takes to start from rest and accelerate to the final velocity. Once that time is solved, we can multiply it by two to get the final total time of movement.

$$t_{1/2} = \frac{2d_{1/2}}{(v_0 + v_{max})} = 0.286\ s \qquad (56)$$

$$t = 2t_{1/2} = 0.572\ s \qquad (57)$$

The amount of time it takes the line to move 2.4 meters at an acceleration $a = 2.99g\ \frac{m}{s^2}$ is a little over half of a second.

We can divide *t* by the Courant Limit to determine how many loops are needed in our FDTD code to model this entire movement. Using Equation 54, we can solve for the Courant Limit:

$$\Delta t = 2.828\ ns \qquad (57)$$

and then solve for the number of loops, *n*:

$$n = \frac{t}{\Delta t} = 2.024 \times 10^8 \qquad (57)$$

Since we defined our grid space to be 1.2 meters, we can perform a cool trick to simulate the movement of the line in a discrete way. Since the total distance, 2.4 meters, is represented by 2 grid cells and the source already exists within one grid cell, we can simulate the movement by slowly turning off the center source grid cell and slowly turn on the source to the grid cell to the right. This turning on and off movement is just as fast as the movement we just determined above. Fig. 12 shows a demonstration of this turning on and off movement on a grid, and Fig. 13 shows how quickly the movement occurs over the total number of loops *n*.

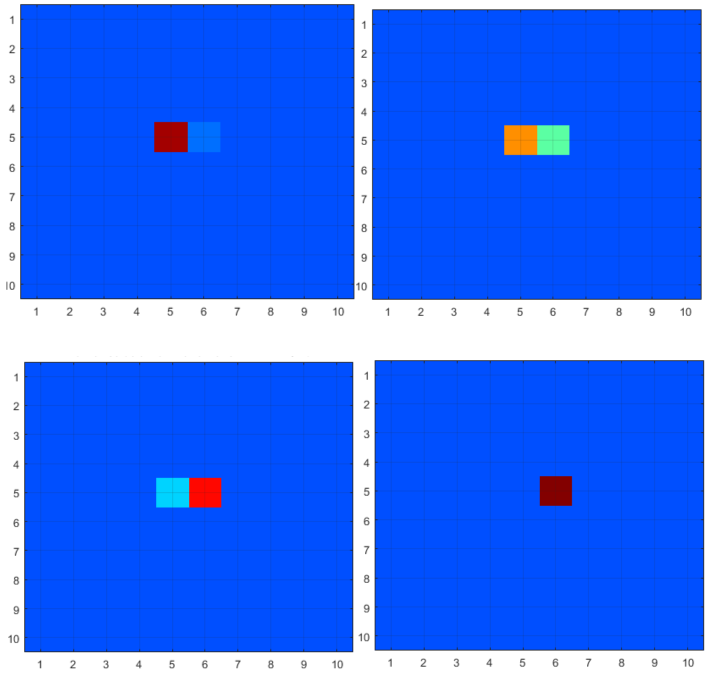


Fig. 12 Demonstration on how the center cell's source is slowly turned off and the cell adjacent to the center is slowly turned on.

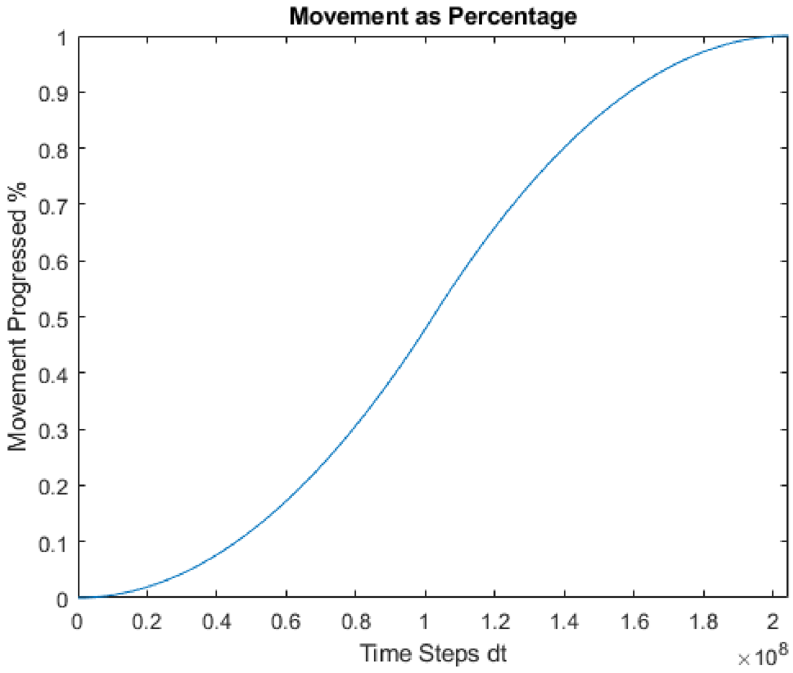


Fig. 13 Graph displaying the percentage of total movement that has occurred per number of time steps.

3.8.2 *Boundary Conditions for Simulation*

Every edge in our simulation, except the bottom, will want to look like free space. On the left, right, and top edges of our simulation we will want to make our simulation absorb any electromagnetic radiation, so it looks like that the simulation space is endless. To achieve this, we will use a Berenger Perfectly Matched Layer boundary condition as described in Section 4.6.

The bottom boundary, however, will want to look like the ground, just as if a power line was strung above it. We can assume the ground at this low of a frequency will act like a PEC.

We know that the ground will behave like a PEC based on its material parameters. According to the authors in *Fundamentals of Applied Electromagnetics* any material that has properties $\frac{\varepsilon''}{\varepsilon'} > 100$ is considered a good conductor [9]. A good conductor can be easily approximated with a PEC.

For dry ground, the permittivity, or ε', is around $3\varepsilon_r = 26.56 \frac{pF}{m}$ [9].

ε'' is defined by the conductivity, σ, divided by the angular frequency, ω. The conductivity of dry ground is around $10^{-4} \frac{S}{m}$ [9]. Since we are looking at a power line that has a frequency of 60 Hz, we know that $\varepsilon'' = \frac{10^{-4}}{2\pi\times 60} = 265.258e - 9$.

We can find the ration of $\frac{\varepsilon''}{\varepsilon'}$:

$$\frac{\varepsilon''}{\varepsilon'} = \frac{265.258e - 9}{26.56e - 12} = 9{,}986.4 > 100 \tag{58}$$

The ratio is much greater than 100, so we can make the bottom boundary perform as a PEC.

3.8.3 *Observing the Electromagnetic Fields*

In order to evaluate the effectiveness of our simulation we need to analyze our electromagnetic fields. To do this, we will collect values of the electromagnetic field at an observation point and perform a Discrete Fourier Transform on the data. This will tell us what frequency we see in space around the line and verify that our simulation is working properly.

We want to know what the electric fields will look like above the line as they propagate towards the ionosphere. Good practice also says that our observation point be at least 20 grid cells away from any boundary to decrease the risk of seeing unwanted reflections or other affects from the simulation. For this simulation, we chose an observation point that was halfway between the center of the grid and the bottom of the top PML.

4. RESULTS

To study the electromagnetic waves that were present in the simulation space we collected the electric field amplitude over time. Since our parallel code was not working, we went back to MATLAB, which is known to be slow and inefficient for this type of simulation. Since the computation time is increased when using MATLAB, we only simulated the first 18 million time steps, which also represents three periods of the frequency in the line. Performing the FDTD simulation in MATLAB with the thick PML layers for three time periods caused the simulation to last around 12 hours or more per simulation.

The first set of data collected was the stationary line with a 60 Hz signal hanging over a PEC ground, seen in Fig. 14. That data showed that we see a decent sinusoidal wave above the observation point. The discrepancies in the beginning of the waveform are most likely caused by both reflections from the ground and the quick turn on of the waveform in the space.

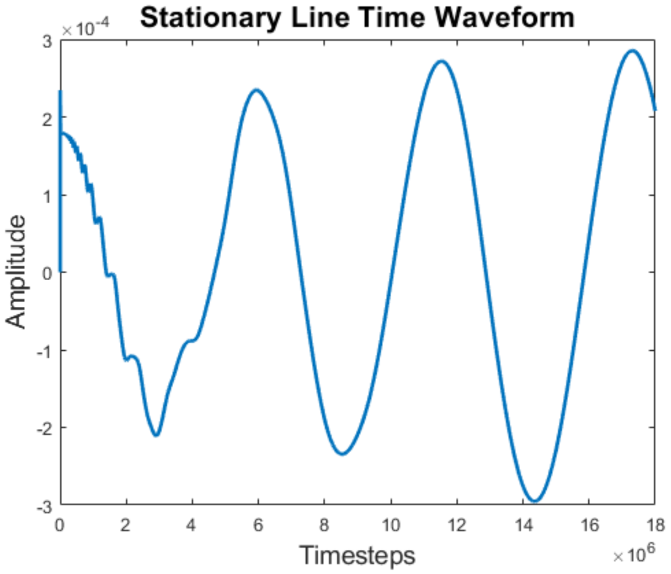


Fig. 14 Time waveform of the line without simulated movement.

The next simulation collected was the line moving at the same rate caused by the Love Wave, seen in Fig. 15. Since we only simulated three time periods of movement, this corresponded to approximately 8% of the total movement simulated.

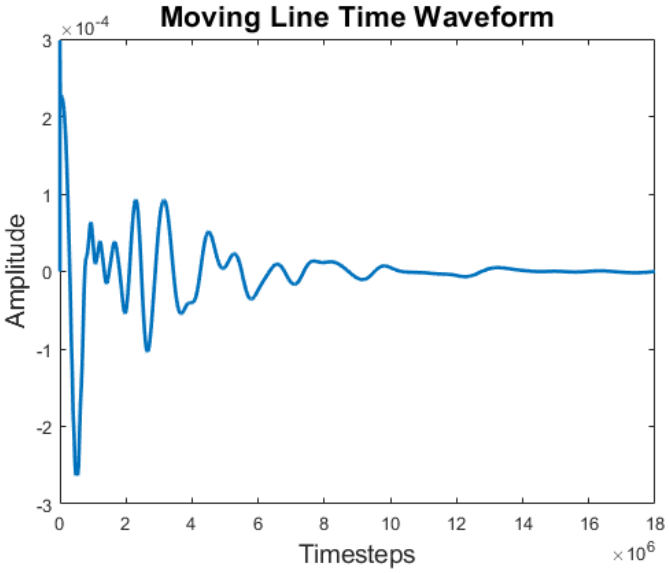


Fig. 15 Time waveform of the line with simulated movement. This movement corresponded with the Love wave data and corresponded to about 8% of the total movement.

The moving waveform also show strange peaks at the beginning, which is most likely caused by the sharp turning on of the line. The rest of the interference is most likely caused by PEC reflections and interactions between the two sources in the simulation.

Discrete Fourier Transforms (DFT) cannot be accurately computed on continuous time signals. If they are, the DFT shows periodicities that are inaccurate. In order to compute the DFT of our observation to determine the frequencies present, we subtracted the stationary waveform and the moving line wave form from each other. Fig. 16 shows this subtracted wave form.

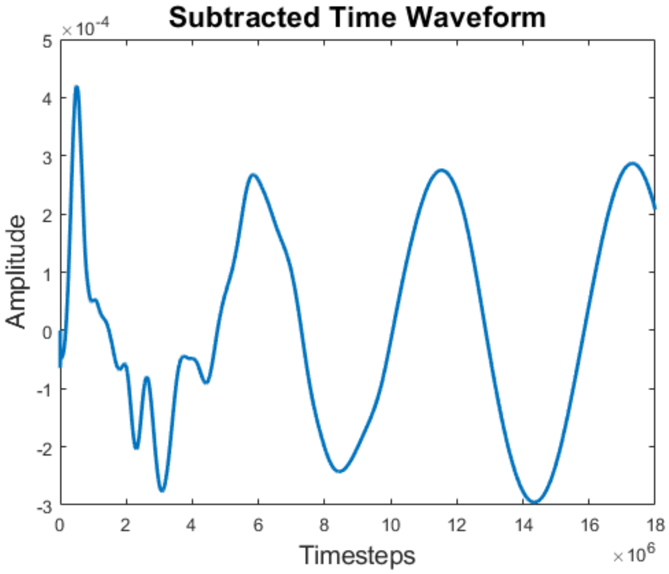


Fig. 16 Time waveform of the stationary wave subtracted from the moving wave.

Fig. 17 shows the resulting DFT which was computed on the subtracted waveform. While there are other peaks in the data, the largest and most prominent peak occurs at 63.06 Hz.

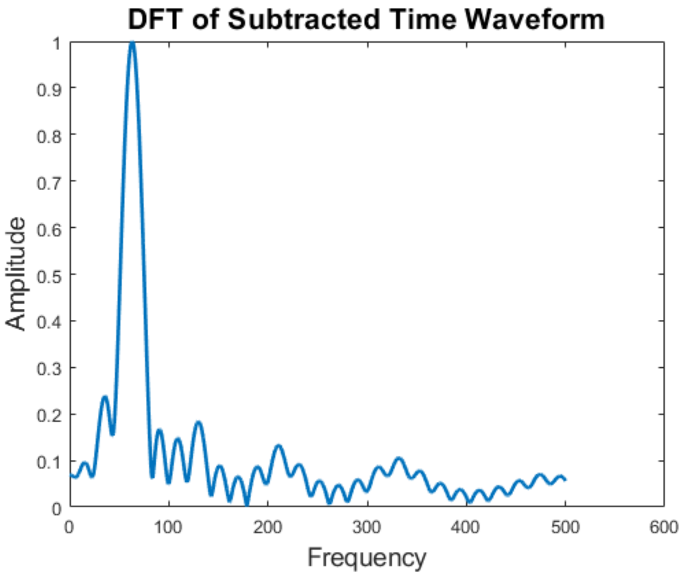


Fig. 17 Discrete Fourier Transform computed on the subtracted waveform, seen in Fig. 16.

The periodicity in the DFT is most likely caused by the subtracted waveform not acting completely like a pulse but rather still operating like a continuous function. One way to remove some of these extra harmonics is to cut off the data after the data crosses zero. Fig. 18 shows the subtracted wave form that has been cut off, and Fig. 19 shows the resulting DFT.

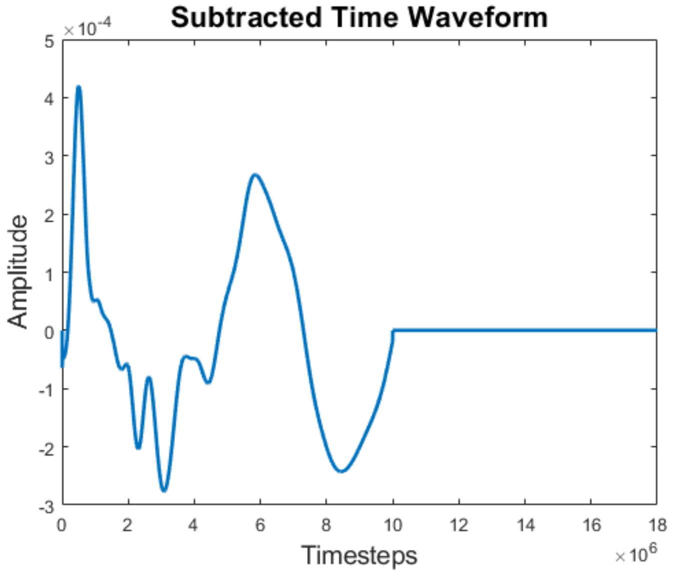


Fig. 18 Time waveform of the stationary wave subtracted from the moving wave and cut off after the source wave becomes periodic.

The resulting DFT shown in Fig. 19 looks similar to Fig. 17 but has less harmonics. Fig. 19 has a peak at 62.56 Hz.

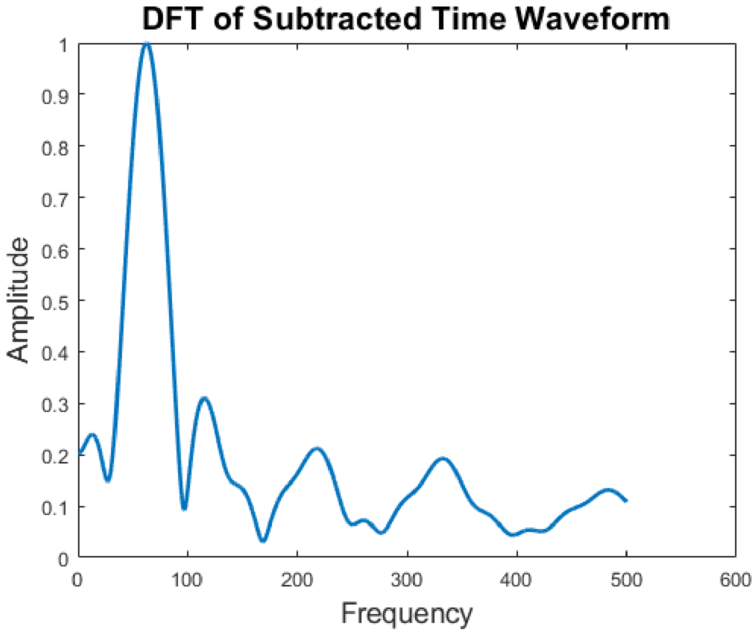


Fig. 19 Discrete Fourier Transform computed on the cut off subtracted waveform, seen in Fig. 18.

These peaks are extremely close to 60 Hz and is what we would expect from a simulation that contains a source cycling at 60 Hz. This tells us that our simulation is fairly accurate.

Since we were unable to compute the full movement in a reasonable amount of time, we decided to also create another simulation that moved the line completely within the three time periods simulating previously. Realistically, the line would never move this quickly, so this simulation was computed as a proof of concept of our moving source idea.

Fig. 20 shows the time waveform for the proof of concept simulation, and Fig. 21 shows the DFT computed on it.

In the proof of concept simulation, we did see a weird peak at the end of the waveform that we did not expect to see. We believe it was there due to the movement abruptly ending so that outlier was cut out from Fig. 20. This allows the waveform could be seen properly on the plot.

In this simulation, we see a peak at 62.06 Hz. The two plots shown in Fig. 20 and Fig. 21 demonstrate that our movement code does work properly and does not remove the frequencies we expected to see in the DFT.

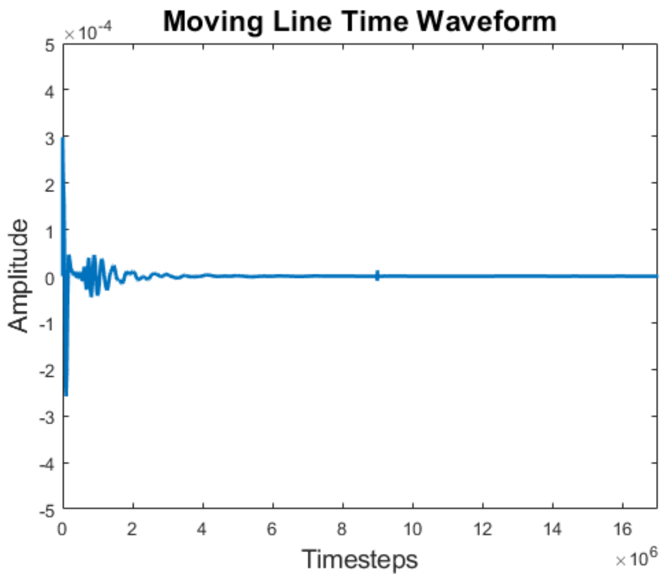


Fig. 20 Moving line time waveform for proof of concept idea.

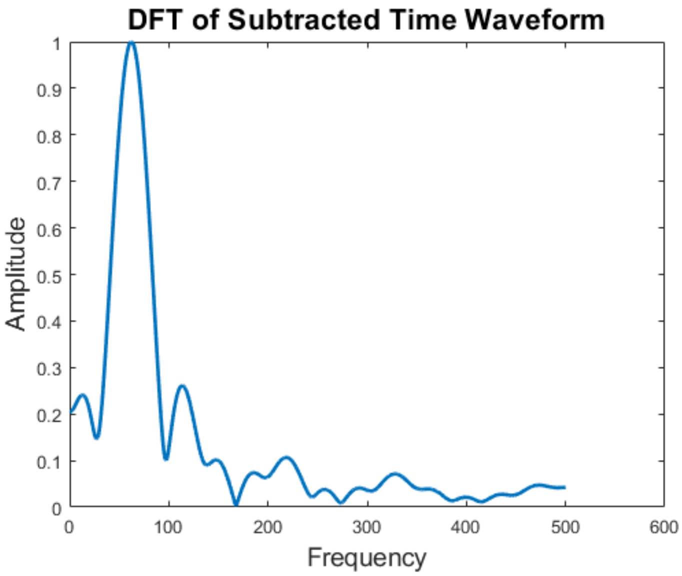


Fig. 21 Discrete Fourier Transform computed on proof of concept idea.

5. CONCLUSION

In this paper we were able to show that a simulation demonstrating power line movement with low frequency sources can be simulated using grid dimensions that match earthquake data. However, more research and data are needed. First, the MATLAB code should continue to be transferred into parallel code to be run on the supercomputers at the University of Utah Center for High Performance Computing. The MATLAB code is too slow and inefficient for it to continue to be a viable option. Once the code is successfully moved into parallel code, researchers can then begin looking at the simulation for the full movement of the line. More research can be done to simulate ionosphere layers above the power line to have an even better understanding of the interactions in the atmosphere.

Learning more about the different effect's earthquakes can have is a good pursuit. It not only answers DARPA's call for research but it will allow us another way to study earthquakes that are not reliant upon direct observation. This is part of a first step to using passive measurements in the atmosphere to tell us what is happening on the surface no matter where it happens on the globe.

6. REFERENCES

[1] Defense Advanced Research Projects Agency Defense Sciences Office, "DARPA Broad Agency Announcement: Atmosphere as a Sensor (AtmoSense)," Mar. 2020.

[2] Mary Pat Hrybyk-Keith, NASA Goddard Space Flight Center, *Ionosphere Graphics*. Greenbelt, 2018.

[3] D. A. Galvin *et al.*, "Ionospheric signatures of the Tohoku-Oki tsunami of March 11, 2011: Model comparisons near the epicenter," *Radio Science*, vol. 47, pp. 1–10, Jul. 2012.

[4] H.-K. Hung and R.-J. Rau, "Surface waves of the 2011 Tohoku earthquake: Observations of Taiwan's dense high-rate GPS network," *Journal of Geophysical Research: Solid Earth*, vol. 118, pp. 332–345, Jan. 2013.

[5] V. E. Korepanov, F. L. Dudkin, and V. A. Pronenko, "Observations of radiation from power lines in near-earth space," *Geomagnetism and Aeronomy*, vol. 55, no. 0016–7932, pp. 688–692, Sep. 2015.

[6] Y.-S. Chang, "Propagation of Seismic Waves: Love waves," 17-Mar-2011. [Online]. Available: https://blog.wolfram.com/2011/03/18/built-to-last-understanding-earthquake-engineering/.

[7] H. Scribner, "Utah earthquake damage: See photos of Salt Lake City, the Angel Moroni statue and more," *Deseret News*, 18-Mar-2020.

[8] The Salt Lake Tribune, "Utah's big earthquake: Buildings damaged, but no major injuries, as state braces for days of aftershocks," *The Salt Lake Tribune*, 18-Mar-2020.

[9] F. Ulaby, E. Michielssen, and U. Ravaioli, *Fundamentals of Applied Electromagnetics*. Upper Saddle River, NJ: Pearson Education, 2010.

[10] A. Taflov and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Norwood, MA: Artech House, 2012.

[11] D. M. Pozar, *Microwave Engineering*. Hoboken, NJ: John Wiley & Sons, 2012.

[12] P. Dawkins, "Section 1-10: Approximating Definite Integrals," *Paul's Online Notes*, 01-Nov-2020. [Online]. Available: https://tutorial.math.lamar.edu/classes/calcii/approximatingdefintegrals.aspx. [Accessed: 06-Apr-2021].

[13] P. Wilmott, *The Mathematics of Financial Derivatives*. New York City, NY: Cambridge University Press, 1995.

[14] J. Nagel, "Maxwell FDTD in 1D." Feb-2020.

[15] the Physics Classroom, "The Kinematic Equations," *the Physics Classroom*, 2021. [Online]. Available: https://www.physicsclassroom.com/class/1DKin/Lesson-6/Kinematic-Equations. [Accessed: 14-Apr-2021].