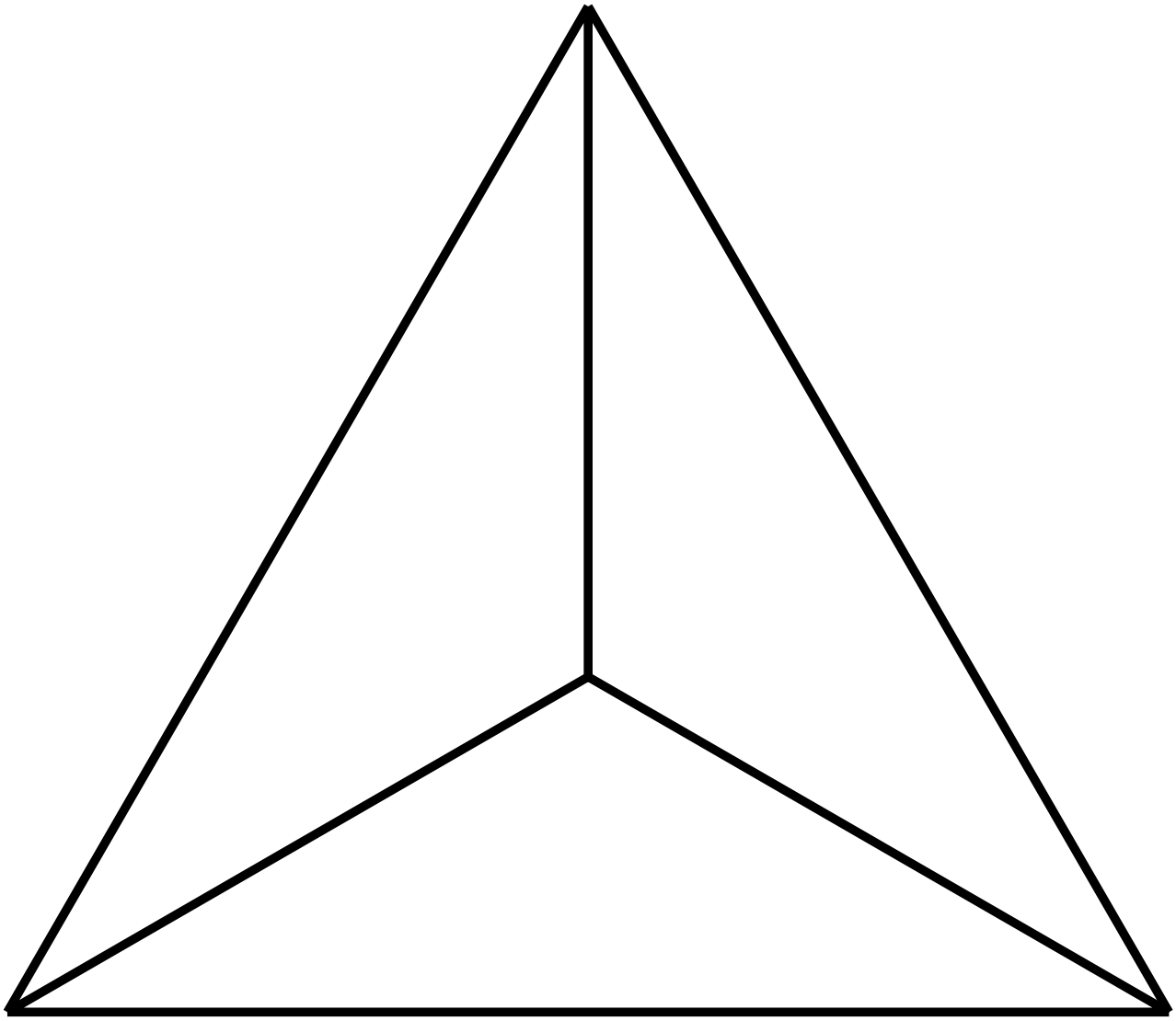
Finite Element Electromagnetic Simulator



Master Thesis

Group 1051
Electronic Engineering & IT
Communication Technology
Aalborg University
The 3rd of June 2021

AALBORG UNIVERSITET
STUDENTERRAPPORT

Aalborg Universistet/Det Tekniske Fakultet for IT og Design
Communication Technology
Frederik Bajers Vej 7
9220 Aalborg Øst
http://www.es.aau.dk

Title:
Finite Element Electromagnetic Simulator

Project:
Master Thesis

Project period:
February 2021 - July 2021

Author:
Benjamin Becsei Christiansen
Jeppe Thiellesen

Supervisors:
Ondrej Franek
Stanislav Stefanov Zhekov

Abstract:

An electromagnetic Finite Element Method (FEM) simulator is implemented for multi-port homogeneous waveguide networks.
A mathematical and theoretical description of FEM is discussed. Principles of reference triangles and numerical integration are utilized to implement the algorithm. The application Gmsh is used to generate a single order tetrahedron mesh used by the FEM algorithm.
Scattering parameters and electric fields are constructed for the FEM implementation created in this thesis. The S-parameters and E-field from the developed implementation are compared with the analytical solution of a TE_{10} perfect conductor waveguide, the Finite Integration Technique (FIT), and the FEM algorithm from CST Studio Suite by *Dassault Systems*.
In the passband region for S_{21}, it is found that the magnitude attained from our implementation has an error of -71.56 dB and -70.26 dB for CST-FEM and CST-FIT, respectively.

Number of pages: 74

Contents

1 | Introduction

An increase in demand for wireless communication services has increased the utilization of the shared frequency spectrum. The demand for wireless communication is expected to rise to a large extent in the near future. The International Telecommunication Union (ITU) predicts that the data usage in 2030 will reach 5016 Exabyte (10^{18} byte) per month. This is a considerable rise from 62 Exabyte in 2020 [1].

For wireless devices to coexist and transfer a vast amount of data using the wireless medium, some form of sharing of this medium is needed [2]. Sharing of the wireless resources is enabled through, but not limited to, the following properties: time, frequency, space, power, polarization, and coding [2]. Regulatory rules and mass production capabilities limit the use of the frequency spectrum. Technologies need to be mature enough to enable mass production in terms of unit cost, production time, and size. Similar parameters limit the transmitter power levels as the frequency spectrum. These parameters include regulatory requirements and mass production capabilities. Additional requirements for transmitter power levels arise for mobile devices. Mobile devices have limited power available, as most mobile devices are battery-powered. The frequency spectrum is typically used for the separation of different systems/communication standards and allows for more efficient use of the spectrum [2].

In order to utilize these parameters efficiently, the ability to precisely control the electromagnetic properties in a communication system is essential. These properties include but are not limited to return loss, coupling, beam pattern, impedance, and power levels. Systems where these are applicable, are filters, couplers, multiplexers, antennas, etc. No universally applicable analytical method exists to calculate the electromagnetic fields and currents of an arbitrary structure. With an increase in the uses of wireless systems and their complexity, an efficient method of simulating the electromagnetics of these circuits with the use of numerical methods is needed to efficiently and precisely develop such systems.

The thesis aims to develop a computational electromagnetic (CEM) tool optimized to simulate highly resonant homogeneous waveguide structures, with three main functional objectives. These objectives being the three-dimensional modeling of electric fields, multiport simulation on homogeneous waveguides, and the emulation of scattering parameter measurements.

1.1 Computational Electromagnetics

The increase of computational capabilities over the last decades has resulted in the advent of computational electromagnetics (CEM) [3]. A set of partial differential equations describes the behavior of electromagnetic fields. Partial differential equations are complicated to solve for simple structures and not possible (or very complicated) for an arbitrary structure. Some analytical solutions do exist for simple structures. CEM solvers based on numerical methods have enabled researchers and designers to model the EM characteristics of RF circuits before production. Before CEM methods existed, researchers and designers had to produce and test multiple prototypes to validate their designs. This process was both costly and time-consuming.

With an increase in the uses of wireless systems and their complexity, an efficient and accurate method of simulating these systems electromagnetics is needed to develop such systems efficiently. The better the CEM model, the fewer prototypes need to be produced, making it cheaper and faster to develop complex systems.

1.2 Motivation:

With the advent of electromagnetic (EM) simulations, radio frequency (RF) designs have become accessible to the general electronic engineer.

High precision simulation is needed for developing complex mmWave EM structures. One solution to this is the implementation of the Finite Element Method (FEM) simulator, which enables accurate modeling of oblique and curved structures [3]. FEM is chosen due to its ability to accurately model complex structures, whereas other methods, such as the Finite Difference Time Domain (FDTD) method, utilizes the staircase approach, which is not as accurate [4]. The FEM method complements the FDTD method, as each simulation method is good for different usages and structures. RF and mmWave engineers utilize electromagnetic simulation tools without detailed understandings of their inner workings. Developing and implementing the theory behind the FEM algorithm to a detailed extend allows the reader to understand the workings of the tools used daily by an RF and mmWave engineer.

1.3 Organization:

The report structure is as follows: the current chapter presents an introduction to the subject of computational electromagnetics and the motivation thereof. Chapter 2 introduces the essential Maxwell equations, wave equation, and the fundamentals of electromagnetism. Chapter three introduces the fundamental properties and uses of waveguides and cavity resonators and the modes associated with them. Chapter 4 introduces the most used simulation algorithms and the most common commercial software tools available. Special attention is given to the FEM since this is the method of interest, arguments for the selection of FEM are presented. Chapter 5 goes into detail with the theory of FEM. FEM for the three-dimensional and the two-dimensional case are analyzed for the structure volume and ports. Both a cavity resonator and a simple waveguide are analyzed. The features to be implemented in the software are introduced in chapter 6.

Chapter 7 presents the implementation of the FEM algorithm in Matlab, presenting main implementation aspects differing from the theoretical analysis in chapter 5. Chapter 8 compares the developed EM simulator with existing simulators, specifically the frequency and time domain methods in the commercial electromagnetic simulator software CST Studio. Lastly, the results are analyzed, and a conclusion is formed. The conclusion of the thesis is presented in chapter 9; future research and improvements are discussed in chapter 10.

2 | Electromagnetic Fundamentals

This chapter is divided up into three sections. First, an overview of the theory for Maxwell's equations, including the terms and mathematical operators used, is presented. Secondly, a section on partial differential equations and how they relate to electromagnetism and Maxwell's equations are given, and lastly, common boundary conditions used for electromagnetism are discussed.

2.1 Maxwells Equations

Maxwell's equations are a collection of four equations. These equations can be written in either differential or integral form. Equations (2.1) to (2.4) are expressed in chronological order: Gauss's law, Gauss's law for magnetism, Maxwell–Faraday equation and Ampère's circuital law [5].

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon} \tag{2.1}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2.2}$$

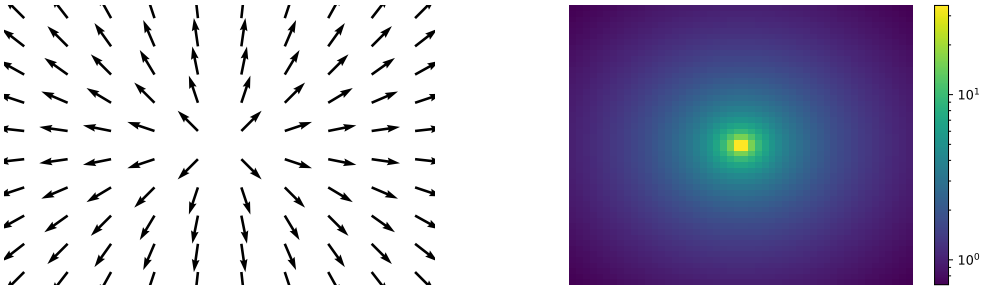
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2.3}$$

$$\nabla \times \mathbf{B} = \mu \left(\mathbf{J} + \epsilon \frac{\partial \mathbf{E}}{\partial t} \right) \tag{2.4}$$

Variable	Description
∇	Differential operator, defined as $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$
$\mathbf{B}$	Magnetic field vector
$\mathbf{E}$	Electric field vector
$\mathbf{J}$	Current
ρ	Charge density
ϵ	Permittivity
μ	Permeability
t	Time

The nabla (∇) operator is used to calculate the properties of the vector fields. The gradient of a function is calculated by element-wise use of the nabla operator on the vector, $\nabla \mathbf{E}$. It can be used to calculate other properties of the vector function. Because of the vector-like

definition of the operator, it can be used with the dot ($\cdot$) and cross product ($\times$) operators. When the nabla ∇ operator is used with the dot product, it is expressed as $\nabla \cdot \mathbf{E}$. The dot product results in the divergence of the function; this is illustrated in Fig. 2.1. The divergence of a vector function results in a scalar function. The divergence is illustrated in Fig. 2.1b. Figure 2.1a shows a sample vector field. The divergence operator outputs a scalar for each input, where the input is a vector. An example use case can be seen in Gauss's law as shown in Eq. (2.1). The electric field flux is dependent on the charge density and the distance to it. It defines the amount of flow resulting from the charges, which is characterized with a scalar quantity with no direction. Gauss's law for magnetism Eq. (2.2), defines the divergence of a magnetic field. The divergence of a magnetic field and the magnetic flux in any area is always zero. This is due to no magnetic charges existing. The magnetic flux is balanced; the amount of magnetic flux going into a volume is the same as the amount of magnetic flux going out of the volume.



(a) Vector field (b) The divergence of the vector field

Figure 2.1: Illustration of the vector field (a), the divergence of a vector field (b) is a scalar value and not a new vector field.

The equation used for Fig. 2.1 is the electric field of a point charge. The divergence is found by executing the operation $\nabla \cdot \mathbf{E}$, and can be expanded to form Eq. (2.5).

$$\nabla \cdot \mathbf{E} = \left(\frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} + \frac{\partial E_z}{\partial z} \right) \tag{2.5}$$

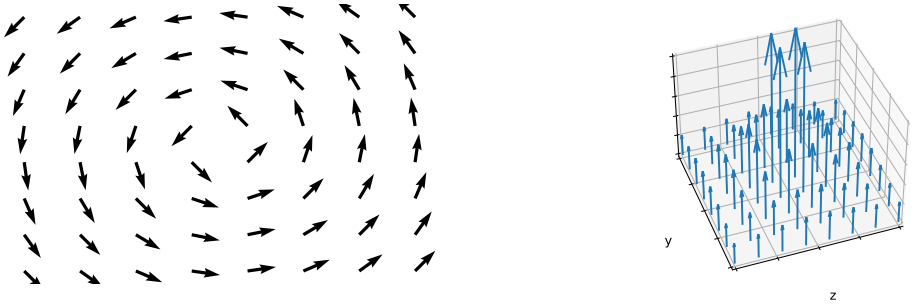
When using the nabla operator (∇) with the cross product operator as presented,$\nabla \times \mathbf{E}$, it results in the curl of a function. The curl of a vector field is a new vector field. Equation (2.5) illustrates how the magnetic field can create an E-field and vice versa. Two examples of a curl of a vector field is demonstrated in Figs. 2.2 and 2.3, where the vector fields are shown in Fig. 2.2a and Fig. 2.3a and the curls are shown in Fig. 2.2b and Fig. 2.3b [5] .

The curl of two vector fields is demonstrated in Figs. 2.2 and 2.3.

The full calculation is presented in Eq. (2.6).

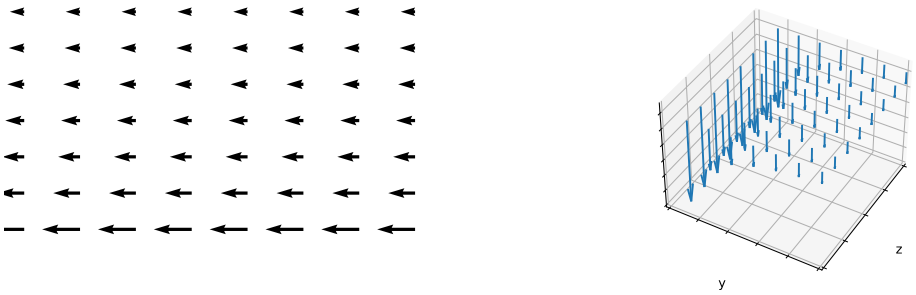
$$\nabla \times \mathbf{E} = \left(\left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right), \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right), \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \right) \tag{2.6}$$

The Maxwell-Faraday equation expressed as Eq. (2.3) states that where there is a changing E-field, there is a magnetic field. If a magnetic field exists, a current or displacement current



(a) Vector field. (b) The curl of the vector field.

Figure 2.2: Illustration of the curl of a vector field. The left image displays a vector field. The right images displays the curl of the vector filed on the left. The curl of the vector field is a new vector field. Fig. 2.2a



(a) Vector field. (b) The curl of the vector field.

Figure 2.3: Illustration of curl on a vector field.

exists where a displacement current does not have a charge. An example of a displacement current is the current flow through a capacitor [5].

2.2 Differential Equations

The two primary differential equations utilized in the analysis of EM fields are the wave equation and the Laplace equation. The Laplace equation is defined in Eq. (2.7).

$$\nabla^2 f = 0 \tag{2.7}$$

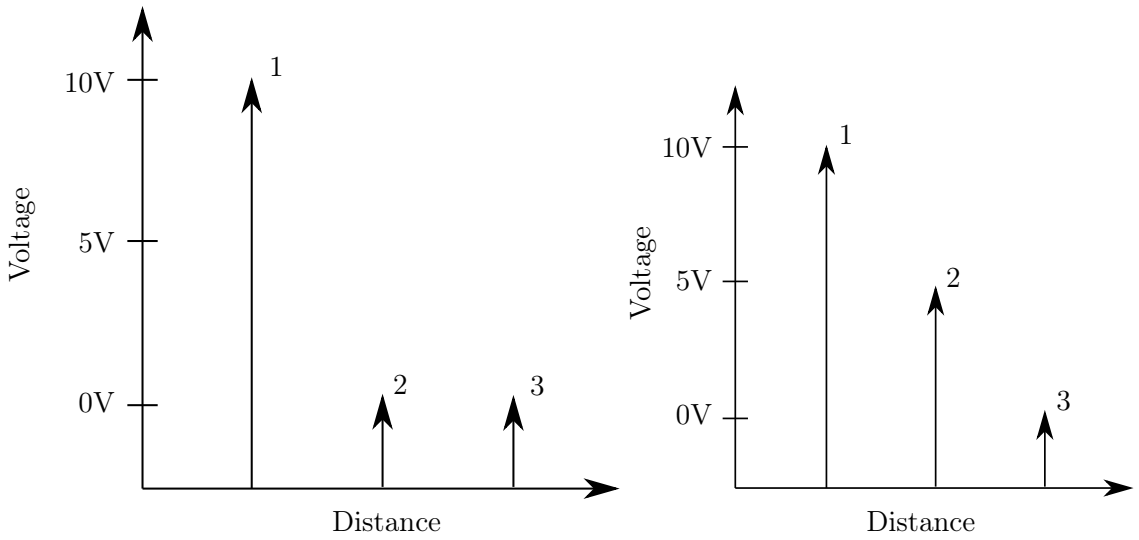
The Laplace equations describe the steady-state of a system or process. Laplace equation can be generalize into the Poisson equation. Poisson equation for electromagnetism, is the solution to electrostatics and is given by the equation is shown in Eq. (2.8).

$$\nabla^2 \mathbf{E} = \frac{\rho}{\epsilon} \tag{2.8}$$

When no charges are included in the domain used, Eq. (2.8) simplifies to Eq. (2.9).

$$\nabla^2 \mathbf{E} = 0 \tag{2.9}$$

To illustrate Eq. (2.7), Eq. (2.8) and Eq. (2.9), an example use case is constructed, consisting of two parallel plates with a potential of 10 and 0 V. It is desired to find the field distribution between the two plates. Three discrete voltage points are selected as shown in Figure 2.4. The boundary values are set to a fixed voltage range, 10 V and 0 V. The second-order derivative defines that the difference of the differences of the surrounding points is 0 if in steady-state; however, the derivative can have other values than 0 in a non-steady case. The difference between the 10 V and the center point is the same as the difference between 0 V and the center point. In Figure 2.4a, it can be seen that the center point has a value of 0 V, resulting in the difference in the differences greater than 0 ($|10-0| = 10$ and $|0-0| = 0$). It can therefore be noted that the system in Figure 2.4a is not in a steady state. When Poisson's equation is valid, a steady-state is achieved. A system in steady-state is illustrated in Fig. 2.4b. Here, it can be noted that the difference in the differences is 0 ($|10-5| = 5$ and $|5-0| = 5$, $|5-5| = 0$).



(a) The system is not in a steady-state; node two's potential will change.

(b) The difference of the differences are the same, and this system is in a steady-state.

Figure 2.4: Illustration of Poisson/Laplace equation.

The second notable differential equation is the wave equation. For the one-dimensional case, it is generally given by Eq. (2.10).

$$\frac{\partial^2 f(x,t)}{\partial t^2} = c\frac{\partial^2 f(x,t)}{\partial x^2} \tag{2.10}$$

Where "t" is time, "x" is position in space, "c" is a constant and $f(x,t)$ is the function to be found. The wave equation describes how a wave behaves when moving through time and space. For E-fields in the three-dimensional domain, the wave equation is given by Eq. (2.11).

$$\nabla \times (\nabla \times \mathbf{E}) = -\mu_0 \varepsilon_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} \tag{2.11}$$

The Helmholtz equation is a time-independent version of the wave equations. The one-dimensional version is given in Eq. (2.12).

$$\frac{\partial^2}{\partial x^2} f(x) = -k^2 f(x) \tag{2.12}$$

Where $f(x)$ is the function to be found, "k" is a scalar. The Helmholtz equation does not change with time. It is used to find the resonating frequencies of structures, where constant standing waves which only change their phase exist. Solving the equation is analogous to solving eigenvalue problem. A solution to the Helmholtz equation consists of an eigenvalue and an eigenfunction. The eigenvalue is the frequency associated with the eigenfunction. The eigenfunction is the function that describes the shape of the solution. For a simple one-dimensional case with the boundary values of $f(0) = 0$ and $f(a) = 0$, the solution to the problem is $A\sin(kx)$ with $k_m = \frac{m\pi}{a}$. $A\sin(kx)$ is the eigenfunction, and in this case there is only one eigenfunction. k_m is the associated eigenvalues [3]. This solution is illustrated for three eigenvalues in Fig. 2.5.

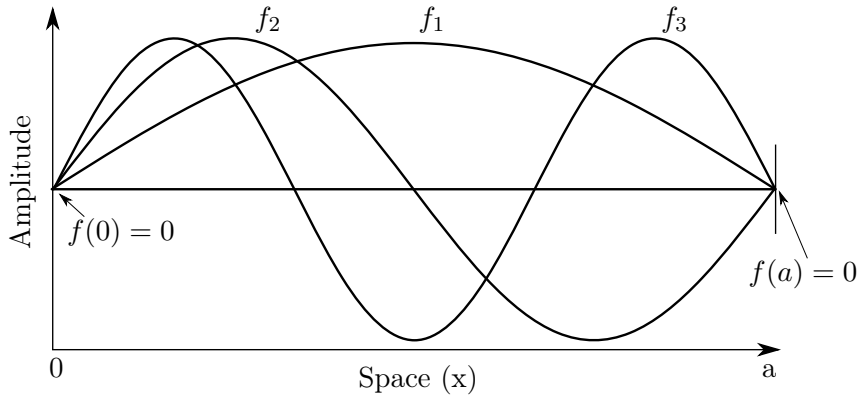


Figure 2.5: Illustration of the one-dimensional Helmholtz solution with boundary values of 0.

A one-dimensional cavity resonator is shown, and the three curves f_1, f_2 and f_3 illustrate three different solutions. The eigenvalues are the frequency that the functions resonates with, these are given by $k_1 = \frac{\lambda}{2}$, $k_2 = \frac{\lambda}{4}$ and $k_3 = \frac{3\lambda}{2}$.

2.3 Boundary Conditions

In solving electromagnetic problems, boundary conditions are essential as they are used to limit the solution domain. The essential boundary conditions describe a perfect electric conductor (PEC) and an open boundary.

The PEC boundary condition definition states that the E-field is tangential to the PEC surface and is equal to zero, as any non-zero tangential field at the conductor's surface would make electrons move and cancel out the field. In other words, the electric potential across a PEC has to be 0 [6].

The open boundary condition is used to simulated open boundaries such as antennas and radiating structures. For the open boundary condition, no reflections occur due to the surroundings of the structure simulated, and the E-field attenuates as if the E-field radiates away to infinity [6].

3 Waveguides and Cavity Resonators

This report will utilize cavity resonators and waveguides to illustrate the implementation of our FEM algorithm. This section gives a deeper understanding of these two structures and their similarities. Physically realizable waveguides use conductors to guide a wave to the desired direction. Waveguides have a lower power loss and can handle higher power levels. This is due to waveguides utilizing a vacuum as a dielectric. It has no loss and the waveguide will be large preventing arcing. Fig. 3.1.

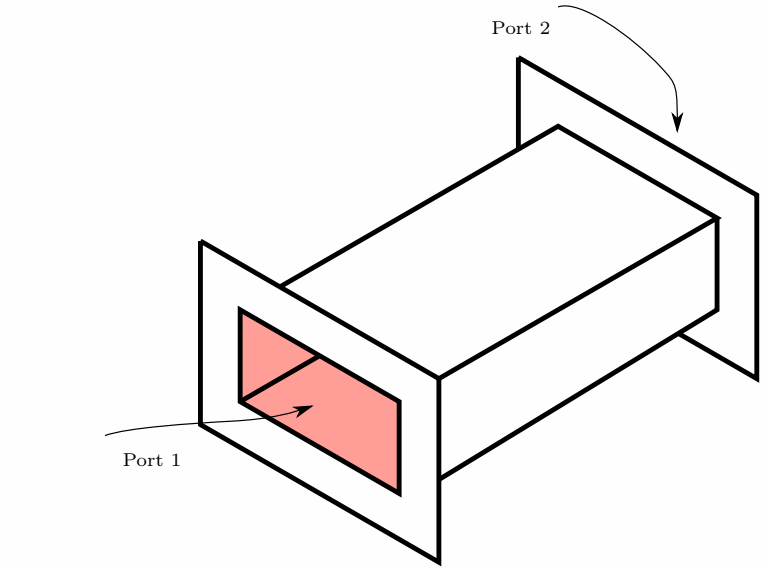


Figure 3.1: Illustration of a two-port waveguide. A rectangular tube with Flanges for mounting.

A cavity resonator uses a volumetric enclosure constructed using good conductors. The primary volumetric enclosure has no inputs and outputs. A closed cavity resonator has oscillating frequencies, which are also known as eigenfrequencies. Each frequency component corresponds to a mode; multiple modes can, however, have the same frequency. For both the cavity resonator and the waveguide to be ideal, perfect electric conductive (PEC) boundaries must be used. No loss on the boundary/walls of the cavity or waveguide if it is constructed using PEC. In practical applications, boundaries are constructed using non PEC materials. Utilizing these materials results in energy loss due to the current induced on the boundary/waveguide wall. Mathematically waveguides can be described as

cavity resonators with an infinite length in one of the three-dimensions. Waveguides are practical as they can be used in an extensive frequency range. The bandwidth is typically defined between the cut-off frequency of the first mode (dominant mode) and the cut-off frequency of the second mode. A waveguide has a cut-off frequency and thereby functions as a high pass filter. Typical rectangular waveguides have a width dimension of $\frac{\lambda}{2}$ of the center frequency, and the waveguide height is half the height or less in order to suppress higher modes.

Waveguides are widely used within the commercial and research areas as a way of guiding RF signals. Standard waveguide flange connections with specified slot dimensions optimized for a specific frequency are developed. Various dimensions for commonly used square waveguides are presented in Table 3.1. The presented waveguides have the designation WR. This designation indicates "Waveguide Rectangular".

Other types of rectangular waveguide dimensions as well as different shapes exist. This report focuses on rectangular waveguide ports only and will therefore not go into further detail with other types. Rectangular waveguide ports have been selected due to their structural simplicity and analytical solution to verify the simulations.

In order to simulate waveguides, input and output ports need to be defined. These ports are waveguide specific, and this is due to the boundary conditions limiting the possible fields at the ports. The port E-field is therefore dependent on the waveguide geometry as well as dimensions. Different types of waveguide ports exist. One such port is a homogeneous waveguide port. A homogeneous port only consists of the structure boundary, with the port surface consisting of only one material. Such port has a two-dimensional analytical solution. Inhomogeneous ports can be applied to any waveguide type, such as substrate integrated waveguides (PCBs), partial substrate integrated waveguides, and cables. Inhomogeneous ports simulated using FEM are complex as they do not have an analytical solution to the port boundary, the theoretical and mathematical formulation is presented in Appendix A. A two-dimensional simulation of the ports before the system simulation needs to be conducted. Homogeneous waveguide ports can be calculated using modes, the Transverse Electric (TE) and Transverse Magnetic (TM) mode. Each of these categories has multiple modes. A rectangular waveguide can only operate in TE or TM mode, whereas parallel-plate waveguides, also have a mode consisting of electric and magnetic fields called Transverse Electric Magnetic (TEM) mode. Coaxial cables only have

Waveguide [EIA]	Dimensions [mm]	f_c [GHz]	Specified f [GHz]	Band
WR650	165.1000×82.5500	0.91	1.12-1.70	L
WR284	72.1360×34.03600	2.08	2.60-3.95	S
WR187	47.5488×22.1488	3.15	3.95-5.85	C
WR90	22.8600×10.1600	6.56	8.20-12.40	X
WR62	15.7988×7.8994	9.49	12.40-18.00	K
WR42	10.6680×4.3180	14.05	18.00-26.50	K
WR28	7.1120×3.5560	21.08	26.50-40.00	K a

Table 3.1: Industry-defined standard waveguide sizes. [7].

TEM modes.

3.1 Transverse Magnetic Modes (TM)

The TM mode or Transverse Magnetic mode occurs when $H_z = 0$ and $E_z \neq 0$. TM modes have multiple sub-modes $TM_{m,n}$, where $m \geq 0$ and $n \geq 0$. "m" denotes the number of $\frac{\lambda}{2}$ wavelengths in the Y-direction, where as "n" denotes the number of $\frac{\lambda}{2}$ wavelengths in the X-direction. $TM_{m,n}$ modes have a cut-off frequency where no wave passes through below it. The lowest TM mode is the TM_{11} mode, the cut-off frequency is defined as Eq. (3.1),

$$f_{c_{11}} = \frac{1}{2\pi\sqrt{\mu\epsilon}}\sqrt{\left(\frac{\pi}{a}\right)^2 + \left(\frac{\pi}{b}\right)^2} \tag{3.1}$$

where "b" is the width and "a" is the height of the waveguide. At frequencies below the cut-off frequency ($f < f_c$) an imaginary propagation constant exists [5]. The wave impedance of the TM modes is defined as Eq. (3.2).

$$Z_{TM} = \frac{\beta\eta}{\omega\sqrt{\mu\epsilon}} \tag{3.2}$$

Where β is dependent on the mode and can be calculated using Eq. (3.3). η is the intrinsic impedance of the material in the volume of the waveguide, defined as $\eta = \sqrt{\frac{\mu}{\epsilon}}$ [5].

$$\beta = \sqrt{k^2 - k_c^2} = \sqrt{k^2 - \left(\frac{m\pi}{a}\right)^2 - \left(\frac{n\pi}{b}\right)^2} \tag{3.3}$$

The wave impedance is real when $f > f_c$ and imaginary when $f < f_c$. The TM mode is not further used within the report and will therefore not be explored further.

3.2 Transverse Electric Modes (TE)

The TE mode or Transverse Electric mode occurs when $E_z = 0$ and $H_z \neq 0$. TE modes have multiple modes "n", where $n > 1$. TE modes have a cut-off frequency, for $TE_{m,n}$ modes, the cut-off frequency is defined as Eq. (3.4),

$$f_{c_{mn}} = \frac{k_c}{2\pi\sqrt{\mu\epsilon}} = \frac{1}{2\pi\sqrt{\mu\epsilon}}\sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}. \tag{3.4}$$

The lowest mode for TE modes is the TE_{10} mode. The cut-off frequency for TE_{10} can mathematically be expressed as Eq. (3.5).

$$f_{c_{10}} = \frac{1}{2a\sqrt{\mu\epsilon}} \tag{3.5}$$

The wave impedance of the TE modes is defined as Eq. (3.6).

$$Z_{TE} = \frac{k\sqrt{\frac{\mu}{\epsilon}}}{\beta} \tag{3.6}$$

Where β is mode number dependent and can be calculate using Eq. (3.4). β is the same for both general modes, TE and TM.

$$\beta = \sqrt{k^2 - k_c^2} = \sqrt{k^2 - \left(\frac{m\pi}{a}\right)^2 - \left(\frac{n\pi}{b}\right)^2} \tag{3.7}$$

β is the propagation constant and defines the speed of phase changes within a material. Similarly to the TM mode, the wave impedance is real when $f > f_c$ and imaginary when $f < f_c$.

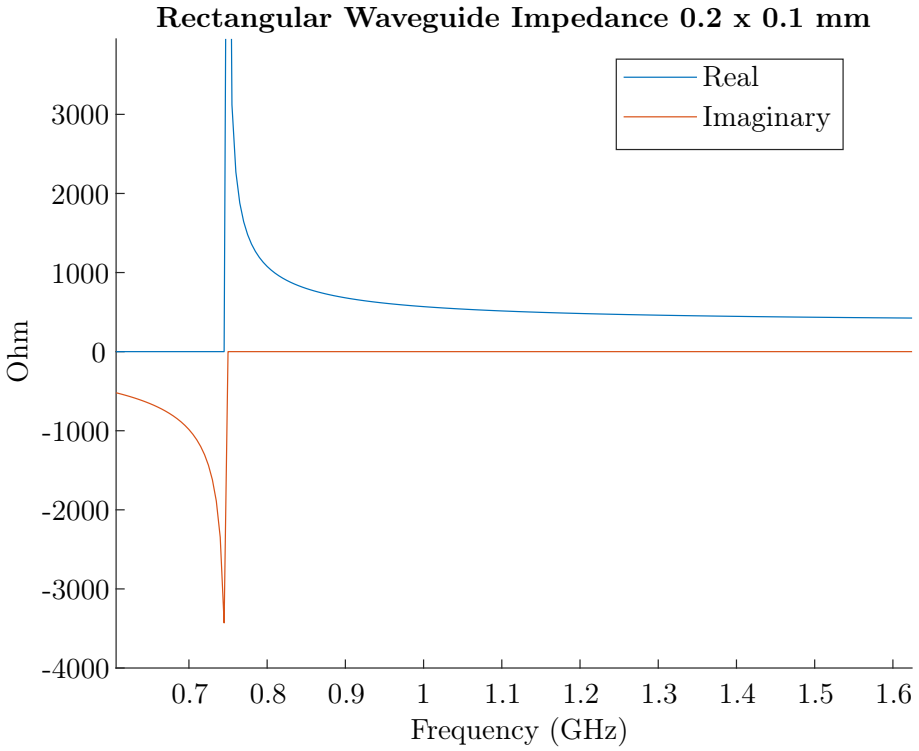


Figure 3.2: 0.2 by 0.1 mm waveguide impedance.

In Fig. 3.2 it can be seen that the actual impedance has an asymptote at 500 ohms. 500 Ohms is therefore a standard impedance in the waveguide passband. The homogeneous waveguide port has been selected due to its analytical port formulation. Based on this, the dominant mode TE_{10} has been selected to be implemented.

In order to verify the later implemented simulations, an analytical solution to a rectangular waveguide is desired. A rectangular waveguide is one of the simple structures where an analytical solution exists. The analytical solution to both a TE_{mn} and TM_{mn} mode are explored.

The E-field for the X, Y, and Z directions, formulation for the X direction is presented in Eq. (3.8).

$$\mathbf{E}_x = \frac{j\omega\mu n\pi}{k_c^2 b} A \cos\frac{m\pi x}{a} \sin\frac{n\pi y}{b} e^{-j\beta z} \tag{3.8}$$

The E-field formulation for the Y direction is presented in Eq. (3.9).

$$\mathbf{E}_y = \frac{-j\omega\mu m\pi}{k_c^2 a} A \sin\frac{m\pi x}{a} \cos\frac{n\pi y}{b} e^{-j\beta z} \tag{3.9}$$

The E-field is defined as $\mathbf{E}_z = 0$ in the Z direction based on the TE_{mn} mode definition. The "A" in Eqs. (3.8) and (3.9) is an amplitude constant for the E-field. For TE_{10} mode, the power flow in the waveguide is given by

$$P_{10} = \frac{\omega\mu a^3 \left|A_{10}\right|^2 b}{4\pi^2} \mathrm{Re}(\beta). \tag{3.10}$$

This can be used to find a desired value of A. The H-field for the X, Y, and Z directions are used. The E-field formulation for the X direction is presented in Eq. (3.11).

$$\mathbf{H}_x = \frac{j\omega\epsilon n\pi}{k_c^2 b} B \sin\frac{m\pi x}{a} \cos\frac{n\pi y}{b} e^{-j\beta z} \tag{3.11}$$

The H-field formulation for the Y direction is presented in Eq. (3.12).

$$\mathbf{H}_x = \frac{-j\omega\epsilon m\pi}{k_c^2 a} B \cos\frac{m\pi x}{a} \sin\frac{n\pi y}{b} e^{-j\beta z} \tag{3.12}$$

The H-field is defined as $\mathbf{H}_z = 0$ in the Z direction based on the TM_{mn} mode definition. The "B" variable is a scaling factor of the H-field.

A CST studio frequency-domain simulation of a 0.2 x 0.1 mm waveguide with a PEC wall has been conducted. Fig. 3.3 shows S_{11} and S_{12} curves of the waveguide.

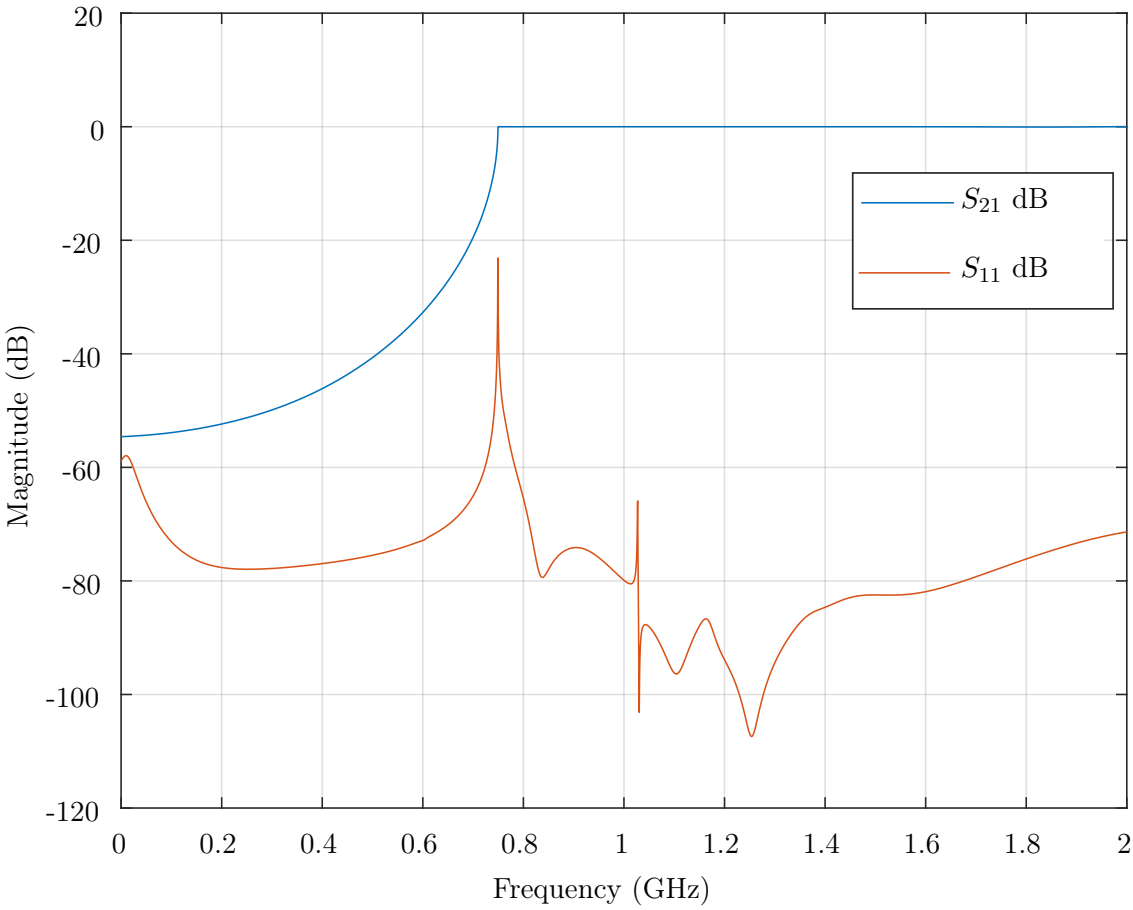


Figure 3.3: Two-port S-parameter CST simulation of a waveguide with the dimensions 0.20 m x 0.10 m x 0.40 m.

The data from Fig. 3.3 and the analytical solution will be used to evaluate the performance of our implemented system.

4 | Computational Methods

There are three main simulation methods used in the industry. The three methods are the Finite Difference Time Domain (FDTD), Methods of Moments (MOM), and the Finite Element Method (FEM). Other methods are used; however, most of them are modified versions of the previously mentioned methods. This section will be giving a brief introduction to these three numerical methods.

4.1 Finite Difference Time Domain (FDTD)

Finite Difference Time Domain (FDTD) is one of the most commonly used methods to date [3]. It was developed by K.S. Yee and is therefore also known as Yee's method [3]. FDTD is a time-domain method and utilizes the time-dependent Maxwell's equations [3]. The time-domain simulations can be converted to the frequency domain after the simulation using a fast Fourier transform (FFT). Due to this, a single simulation in the time domain covers the whole frequency domain. This is one reason that the FDTD method is fast and efficient. The FDTD method does not naturally handle oblique or curved shapes of material boundaries; variants of the method for curved surfaces do exist [3].

FDTD utilizes a volumetric mesh, where the entire volume of the simulation domain has to be meshed [3]. In order to terminate the simulation volume, perfectly matched layer (PML) or PEC boundary conditions are used; other methods do exist [3]. At each time step, the FDTD method calculates the electric field at a given discrete point, using the nearby magnetic field. The electric and magnetic field values are stored at these discrete points. Electric and magnetic field grids are often separated in time (E, H, E, H). This "leapfrog" method improves accuracy over having both E and H-field at the same time instance [3]. This technique is illustrated in Fig. 4.1 when half-time steps between field types are used. All-time marching algorithms, including the "leapfrog" method, allow for efficient memory use as the calculations can be done in memory places, where the new E field replaces the old E field. To calculate the next field from the current, it utilizes the difference between the nearby points using a Taylor series expansion of Maxwell's equations. The E- and H-field are shifted in space relative to each other, the surrounding

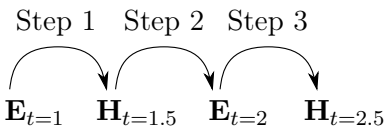


Figure 4.1: Leapfrogging alternates between the E- and H-field for each step. The H-field is calculated for half a time step.

points are used to calculate the new field; this is shown in Fig. 4.2 [4].

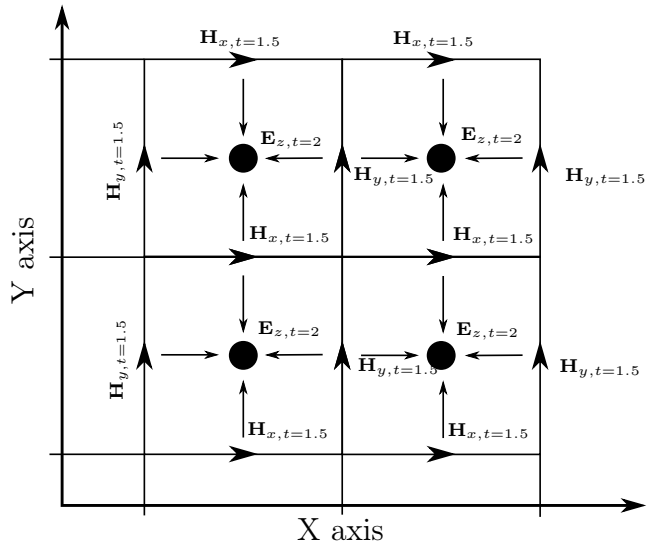


Figure 4.2: An update step in a two dimensional FDTD. $\mathbf{H}_x$- and $\mathbf{H}_y$-field at time 1.5, used to find $\mathbf{E}_z$-field at time 2. The H-field are at the indicated arrows, the E-field are at the circles.

The formulation is explicit as future states depend on current states [3]. FDTD is a popular method for wideband systems as simulations are of the same complexity without dependence on frequency bandwidth, if the mesh is kept the same size in terms of wavelength. In exceptional cases where large objects with a large amount of free space need to be simulated, FDTD is not optimal as it needs to discretize the whole volume, including the free space in the system. FDTD has some limitations in terms of numerical dispersion and anisotropy [3]. Numerical dispersion means that the speed of propagation on the grid varies with frequency. Numerical anisotropy means that the speed of propagation on the grid changes depending on the direction of propagation. For a square grid, wave propagation of 45 degrees travels faster than wave propagation at a 90-degree angle to the grid. The error due to both dispersion and anisotropy increases with increasing frequency when the mesh size is kept the same.

4.2 Method of Moments (MOM)

The Method of Moments (MOM), also known as Boundary Element Method (BEM), is a standard method used in commercial simulators [3]. MOM is a frequency domain algorithm. It utilizes a surface mesh and, thereby, integral equations [3]. For each material, the surface is meshed. This surface mesh can be both structured and unstructured. When a structure is constructed of multiple layers, each layer is meshed on its surface/transition to the next material [3]. All surface boundaries are discretized. The MOM method can be implemented in two ways, using discrete points or using a basis functions [3].

The MOM method is an implicit method and solves integral equations resulting in matrix inversion. MOM solves the integral form of Maxwell's equations. Currents within the structures are discretized and expressed in terms of basis functions or discrete points.

MOM is then used to find the constants (amplitudes) for the basis function or discrete point.

The MOM algorithm is efficient at calculating systems where a low amount of hard boundaries exist. Hard boundaries are boundaries where a discrete change in material occurs. MOM can not be used for soft boundaries, i.e., material properties change slowly over distance. Due to the frequency domain characteristics, the MOM method is primarily suited for narrowband simulations as each frequency point needs a separate calculation compared to other methods such as FDTD. MOM is often utilized when the system is large and contains a vast amount of free space. This is due to MoM being based on surface meshes; therefore, MoM only needs to calculate the surfaces it does not need to model and mesh all of free space. It uses the Greens function to calculate the interaction[3].

4.3 Finite Element Method (FEM)

The Finite Element Method (FEM) is a commonly used method in the industry. It is a universal numerical algorithm used to solve partial differential equations. FEM is used in many disciplines, some of which are: fluid dynamics, mechanical structure analyses, and electromagnetics.

FEM solves a linear system of equations. Structures are split up into smaller segments using unstructured or structured mesh, and these elements are called "Finite Elements" [3]. FEM utilizes basis functions to discretize the simulated structure in space [3]. This results in a system of equations. FEM then solves for the equation constants. Basis functions with their constants are then combined to form the solution to the whole problem.

In electromagnetics, FEM is used to find a numerical solution to the wave equation. It has advantages compared to other algorithms, such as FDTD, to model complex structures, such as curves. It uses complex grids/meshes to represent these structures. The mesh can be both structured or unstructured. In an unstructured mesh, each subdivision of the structure to be simulated can have different sizes. Complex curves require a fine mesh in order to be accurately simulated. Linear structures can be simulated accurately with a coarser mesh.

In wideband problems, FEM is CPU and memory-intensive compared to other methods such as FDTD [3]. The CPU intensity increases due to the method having to compute each frequency point. This is due to each point having a relationship to all other points. There are as many equations as there are points, each with as many unknowns as points.

4.4 Other Computational Electromagnetic Methods

The previously mentioned methods are not the only computational electromagnetic (CEM) methods available; these only represent the most common methods. Other methods based on the aforementioned one exist as well as other independent methods.

One method based on one of the above is the Multilevel fast multipole (MLFMM) method which is a version of MOM optimized for large structures [8]. It is less computationally expensive for very large structures as it has a high computational starting cost but with a

slow cost growth. Very large structures include structures where elements are separated by a large amount of free space. The high stating cost makes it unsuited for small structures [3].

4.5 Subdivision of Structures: Mesh Generation

In order for any simulation algorithm to be able to simulate a structure, the structure has to be divided into smaller pieces. The simulation algorithm can then solve these pieces. This section will look at subdivisions using rectangular structures (cubes in three-dimensions) and triangles (tetrahedrons in three-dimensions).

The subdivision of structures known as meshing has a significant effect on simulation performance [9]. A higher degree of subdivision (meshing) will be able to replicate structures to a greater extent. Meshing can be structured or unstructured. Structured meshing occurs when all cells are of the same size, whereas unstructured cells can be stretched. Unstructured and structured cells need all to be of the same basic geometry.

The various simulator algorithms utilize different meshing approaches. Two main approaches are taken into account, surface meshing and volumetric meshing. The Finite Element Method (FEM) and Finite Difference Time Domain (FDTD) method utilize a volumetric mesh, whereas Methods of Moments (MoM) utilizes a purely surface mesh. Unstructured meshing needs to be consistent. Consistency means that edge endpoints or nodes need to be shared between different cells. The goal is to create a meshing that is equivalent to the structure to be simulated. This is achieved by utilizing cells that have individual edges as close to being equivalent as possible within the same cell [9].

The structure shape of individual cells varies depending on the method of simulation. Simulation methods often enable the use of different meshing types. Structure shapes that FEM can utilize in three-dimensions are; cubes, prisms, pyramids, and tetrahedrons. Two of the most popular structures are the previously mentioned cubes (rectangular in two-dimensions) and tetrahedrons (triangles).

Errors resulting from the meshing/subdivision have various factors, two of which are density and structure type. The main error component that occurs from meshing is a result of selecting a cell structure that is not suited for a specific structure to be simulated. Structures such as circles require versatile meshing to simulate the curved surface. Curved complex surfaces are primary sources for error, where cells are often not very accurate at replicating the structure. Some cell structures are better than others for such purposes. Triangles are superior to rectangular cell structures in this regard; this can be seen in Fig. 4.3.

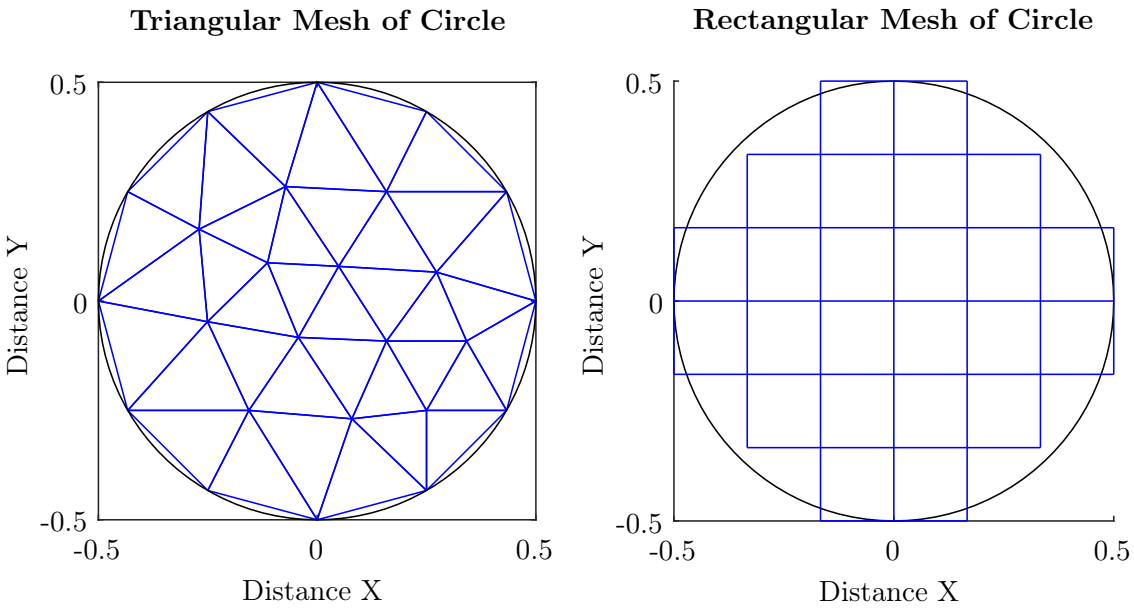


Figure 4.3: Illustration of rectangular (right) and triangular (left) mesh.

Phase error occurring due to mesh formation is a common issue. The total phase error is the sum of individual cell phase errors [3]. The phase error is lower for triangular cells or tetrahedron cells in three dimensions than other cell structures. This is due to the cancellation of phase error due to random orientations of the individual cells [3]. Triangles and tetrahedrons can have random orientation and need not be parallel to any axis [9]. Rectangular cells have to be parallel to at least one of the defined axis.

4.6 Implementations

In this section, an overview of existing implementations is presented.In order to explore the uses for the different methods utilized in the commercial and research sector. This section will discuss each solution, what methods individual software suites have and what problem type is recommended for each method. Some implementations are optimized for both high frequency and low-frequency problems, such as electrostatics, motors, wireless power, and similar. Other implementations are optimized for high frequencies methods, such as ray tracing. Individual methods have advantageous properties for different structure types and desired simulations; they are therefore combined in software packages. The type of solver can then be selected depending on the problem, its structure characteristics, desired accuracy, and computation time. The software packages have a domain that they advertise as their expertise. The software suite will therefore be best suited for that specific domain.

The commercial and open source software discussed in this section is shown in Table 4.1 with their relevant solvers.

CST studio from *Dassault Systems* and FEKO from *Altair* focuses on relatively large structures like antennas, PCBs, humans, vehicles, ships, and planes. Since this is an extensive range of dimensions, they have multiple methods implemented.

CST utilizes an optimized version of FDTD called the Finite Integration Method (FIT),

Software	Frequency Domain	Time Domain
CST	FEM, MoM MLFMM	FIT
Feko	NA	FDTD, FEM MoM, MLFMM
OpenEMS	NA	FDTD
HFSS	FEM, MoM	FEM
Pathwave	FEM, MoM[10]	FDTD
AWR	MoM, FEM	NA

Table 4.1: List of CEM software and their solvers.

where Perfect Boundary Approximation (PBA) is utilized. FIT eliminated the staircase approximation limitation in FDTD. CST recommends the use of the time domain solver for medium to large structures. Frequency domain solvers are recommended for very resonant structures. When using the time domain solver for very resonant structures, the field takes a long time to dissipate and thereby resulting in phase errors due to the iterative approach, where the next time step is depending on the previous and thereby summing all errors from each time step [8]. The frequency-domain solver in CST is based on FEM. The Integral Equation Solver in CST uses MOM, MLFMM, and ACA (adaptive cross approximation) [8]. MLFMM and ACA are MOM like. MLFMM is recommended for very large structures with long distances. It is well suited for radar cross-sections and similar problems with far Field interaction. ACA is good when using a detailed mesh that is small relative to the wavelength. Therefore. They all utilizes a surface mesh.

Altair FEKO, similar to CST, focuses on antennas, cars, and larger structures. It does not have the FIT method; it instead utilizes an implementation of the FDTD method.

HFSS from *Ansys*, Pathwave from *keysight*, and AWS from *Cadence Design Systems* focus on the simulation of PCBs and ICs. They have FEM and MoM in Frequency Domain. HFSS has the FEM algorithm implemented in the time domain. HFSS advertises its core abilities for the following structures: "antennas, antenna arrays, RF or microwave components, high-speed interconnects, filters, connectors, IC packages and printed circuit boards" [11]. Specialization in IC design and small structures differentiates it from CST. This explains the use of different solver types. ICs are very small structures in comparison to the wavelength.

OpenEMS is an open-source FDTD solver. It has two different versions of the FDTD method. One using a Cartesian grid and one using a cylindrical grid. The cylindrical grid is optimized for cylindrical structures, such as magnetic resonance imaging systems (MRI). The software is advertised as a tool for MRI systems, antennas, and PCBs [12]. The OpenEms software is a command-line tool and does not include a user interface, meshing, or data displays.

Implementations Summary

In order to solve a specific problem, the structure characteristics associated with this problem dictate the type of method used. For structures with fine details relative to the

wavelength like ICs, PCBs, software suites such as HFSS, Pathwave, and AWS use FEM and MoM as their solvers are well suited for these types of problems. For larger structures, CST, FEKO, OpenEMS using the FDTD or FIT method is appropriate algorithms. For very large structures, MLFMM should be used.

4.7 Algorithm Selection

With FDTD, MOM, and FEM introduced in Sections 4.1 to 4.3, a suitable algorithm needs to be selected for implementation. Depending on the use case, each algorithm is suitable for different purposes. Our goal is to simulate highly resonant narrowband waveguide networks, such as filters constructed using complex three-dimensional structures. In this section, the algorithms will be evaluated based on the examples presented in [13], an estimate of the computational cost presented in [3] and the previous information given in this chapter.

In the paper [13], all three methods are used to simulate a Vivaldi antenna from 0.6 to 3 GHz. It concludes that all the methods are equally good at solving the far-field and S-parameters of the antenna used. The authors in [13] conclude that the primary difference between the methods can be found in the computation time and memory usage. From the computation time presented in [13], it can be seen that the FEM algorithm is faster than FDTD and both faster and more memory efficient than MoM. It can be noted that the FEM algorithm is the fastest of the three, however, with a substantially larger memory usage than FDTD but less than MoM.

The number of computations can be analyzed theoretically. For FEM, FDTD and MOM, the number of computations scales with the frequency. For FEM and FDTD the scaling factor is f^4 [3]. If the frequency is doubled, the amount of computations to be done is increased by a factor of 16. This applies for one frequency in FEM and all frequency components up to f in FDTD. For MoM, the scaling factor for the full three-dimensional problems is f^6. However, it can be improved by using only partly three- or two-dimensional structures [3]. The initial cost of the different methods differs greatly depending on the specific problem, resulting in a constant multiplication factor of the computational cost.

In [13] all algorithms simulate the same bandwidth; both FDTD and FEM have the same scaling function of f^4, but attain a different computational time. The extra computation time for FDTD therefore must be due to the constant scaling factor. The constant factor for FDTD is potentially due to the Vivaldi antenna being a resonant structure, further indicating that the FDTD is not optimal for very resonant circuits due to the long settling time and slow dissipation of energy [8].

The FEM algorithm is the most flexible of the methods in terms of meshing of complex three-dimensional structures using unstructured three-dimensional elements for the mesh as described in Section 4.5.

In summary, because of the long computation time when using the FDTD method for resonant structure and the flexibility of FEMs modeling abilities of any three-dimensional

geometry with a high degree of accuracy and its fast computation time, FEM is chosen as the algorithm to implement.

5 | Finite Element Method

The Finite Element Method is a numerical approach used by many disciplines to solve partial differential equations. One such special case is the wave equation used in the modeling of electromagnetic circuits. FEM can model objects with complex geometries, such as curves. FEM has the ability to accomplish this with the use of an unstructured mesh and grid. This chapter will detail the theoretical working of the FEM method, the general understanding of FEM, the three-dimensional case, and the introduction of two-dimensional ports. It is primarily based on two books, *Computational Electromagnetics* by Anders Bondeson, Thomas Rylander and Pär Ingelström [3] for the implementation technique and *The Finite Element Method in Electromagnetics* by Jian-Ming Jin [9] for the FEM problem formulation.
The mesh utilized by FEM depends on the dimension of modeling. Fig. 5.1 shows the most utilized structures for mesh generation divided up into the different dimensions, ranging from the one-dimensional case to the three-dimensional case.

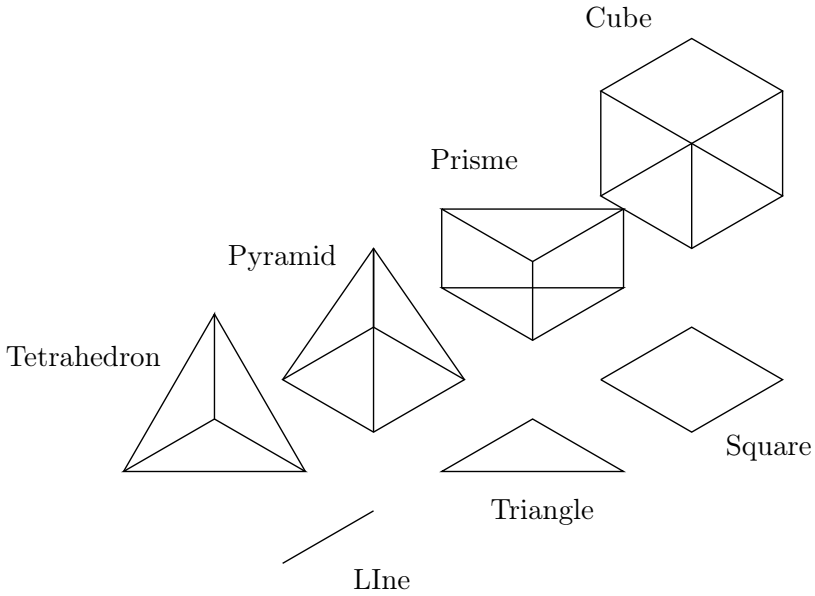


Figure 5.1: The most common shapes used to construct a mesh in FEM.

The use of unstructured mesh enables the modeling of complex objects. Unstructured mesh enables a higher degree of meshing where the most significant change in electric and magnetic fields occurs.
The FEM method solves a set of linear equations; this is costly, requiring matrix inversion. The size of these matrices is large and therefore also requires a substantial amount of

memory.

5.1 The Ideas of FEM

In this section, an intuitive overview of the FEM algorithm is presented. Section 5.2 will dive into detailed mathematical formulations utilized in this report. This section is primarily based on the books "The Finite Element Method in Electromagnetics" [9], "Theory and computation of the Electromagnetic Fields" and "The Finite Element Method: Theory, Implementation, and Applications" [4].

The goal of the Finite Element Method is to discretize partial differential equations, enabling them to be solved by a computer. In order to discretize an equation, a representation in terms of discrete numbers instead of continuous functions needs to be derived. To explain the discretization of a partial differential equation, the discretization of a function is introduced as an example. An example function is shown in Fig. 5.2.

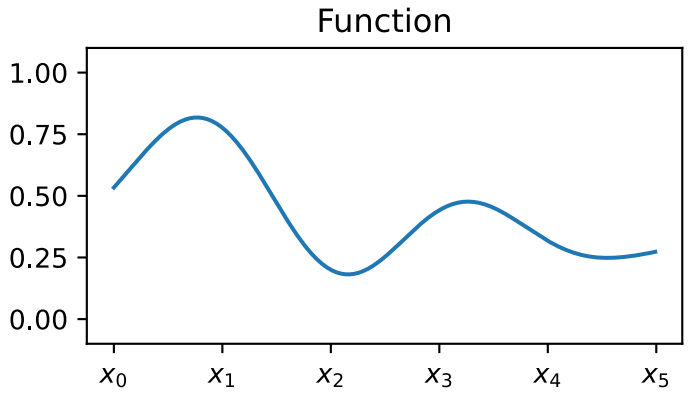


Figure 5.2: Sample function f to be interpolated.

In Fig. 5.3, an approximation of the function presented in Fig. 5.2 is demonstrated. This approximation is derived with the use of interpolation. Linear pieces are used to represent the function; these pieces have start and endpoints at x_i; x_i are discrete values on the X-axis. The combined linear pieces create a continuous linear piecewise function that approximates the original function and can be described uniquely by the discrete values at x_i.

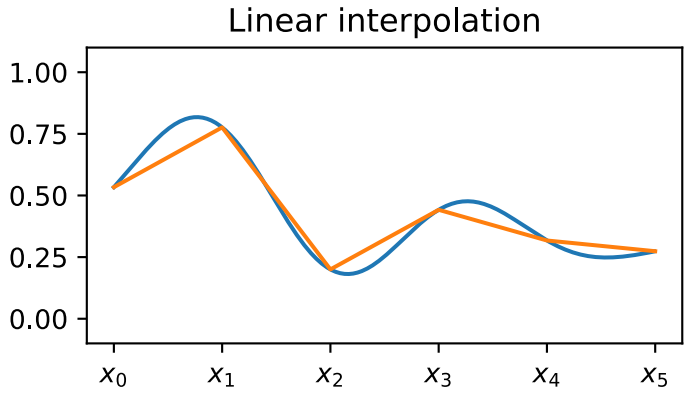


Figure 5.3: Function f, interpolated using continuous linear piecewise function.

Instead of describing the interpolation function as a continuous piecewise linear function, it can be described as a sum of weighted basis functions, a linear combination of basis functions. This is done using the *hat* basis functions shown in Fig. 5.4 and weights.

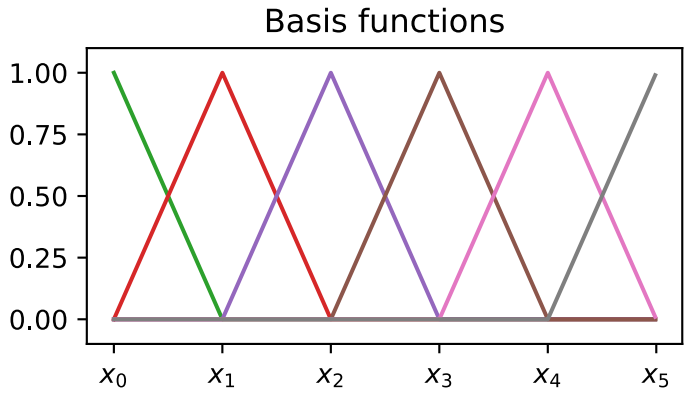


Figure 5.4: *Hat* basis functions.

In order to do so, the weights need to be found. The weights are the sampled values of the original function at the input of x_i; these values are shown in orange on Fig. 5.5.

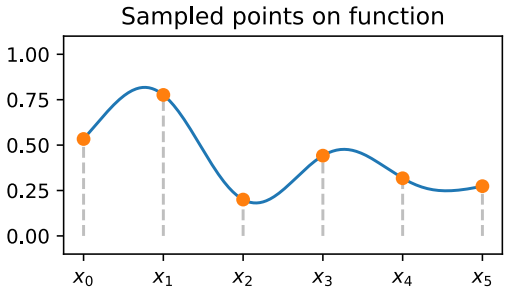


Figure 5.5: The sampled points at x_i of the original function used for the interpolation method.

The *hat* basis functions presented in Fig. 5.4 is scaled by the sampled values at the input of x_i of the original function. The sum of the basis functions results in the continuous piecewise linear function that interpolates the original function, this is demonstrated in Fig. 5.6, and the result is shown with the orange function.

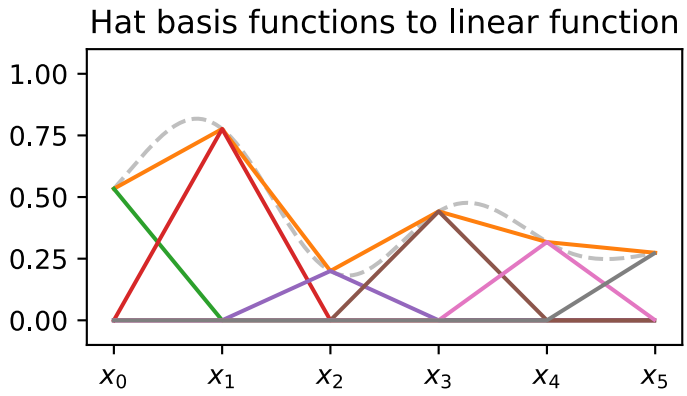


Figure 5.6: Demonstration of the *hat* functions creating a continuous linear function is shown in orange. The values of hat functions are added together, forming the result in orange.

An alternate approach to discretization is to make the linear functions follow the function of interest so that it minimizes the L2 norm between the function and the piecewise linear function. The L2 norm is minimized when the residual of the piecewise linear function and the estimated function are orthogonal to the basis functions. This means that there is nothing left in the function f that the basis functions can describe. This is analogous to conduct a projection from a three-dimensional space to a plane; the part of the removed vector should be orthogonal to all of the projected space [6]. This is illustrated in Fig. 5.7.

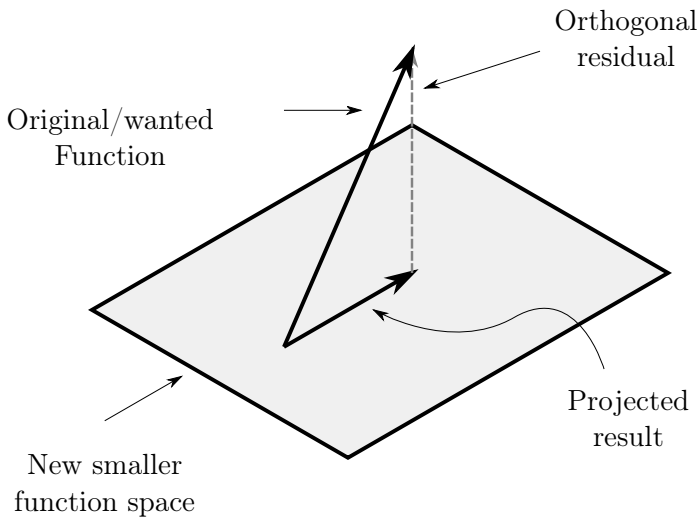


Figure 5.7: Illustration of the analogy in vector space described using the problem of function spaces.

Therefore it is called the L2 projection. It represents a projection from one function space to another in the shortest possible distance. Finding the L2 projection for the function f is given in Eq. (5.1),

$$\int_I \left(f - \left(\sum_{j=0}^{n} \xi_j \varphi_j \right) \right) \varphi_i \, dx = 0, \quad i = 0, 1, \ldots, n \tag{5.1}$$

where f is the function to estimate, $\sum_{j=0}^{n} \xi_j \varphi_j$ is the piecewise linear function made from weighted basis functions, ξ_j is the weights associated with the basis functions φ_j. φ_i represents the basis functions from 0 to i. I is the line for $x_0 < x < x_5$. There are n equations since Eq. (5.1) have to be true for all basis functions. Equation (5.1) can be expressed as:

$$\begin{aligned} \int_I f \varphi_i dx &= \int_I \left(\sum_{j=0}^{n} \xi_j \varphi_j \right) \varphi_i dx \\ &= \sum_{j=0}^{n} \xi_j \int_I \varphi_j \varphi_i dx, \quad i = 0, 1, \ldots, n \end{aligned} \tag{5.2}$$

Equation (5.2) results in "n" equations with "n" unknowns and can be rewritten in the matrix form,

$$M\xi = b \tag{5.3}$$

where the indexes of the matrix are calculated as:

$$M_{ij} = \int_I \varphi_j \varphi_i dx \tag{5.4}$$

$$b_i = \int_I f \varphi_i dx. \tag{5.5}$$

The values for the variables ξ_j are found and the piecewise linear function is given by $\sum_{j=0}^{n} \xi_j \varphi_j$. The result is shown in Fig. 5.8. As it can be seen, the endpoints of the linear functions can over and undershoot f. The piecewise linear function estimates the function f in the least squares sense.

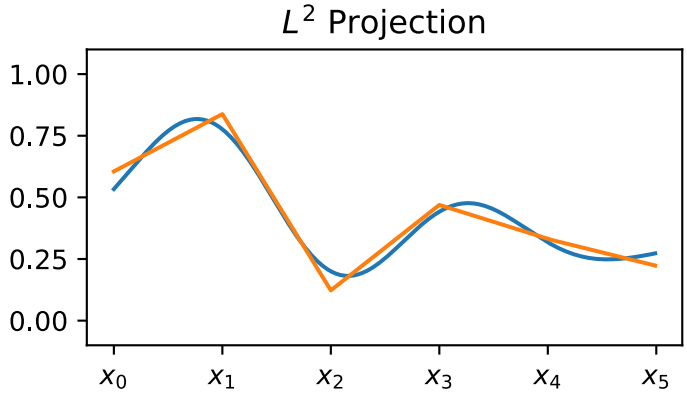


Figure 5.8: Piecewise linear approximation of function f.

In Eqs. (5.6) and (5.7) the "M" matrix, the "b" and "ξ" vector is shown for $x_0, x_1, \ldots, x_5$ equal to $0, 1, \ldots, 5$. As it can be seen in Eq. (5.6) the "M" matrix is sparse as the value at a point is only influenced by the surrounding two points. In an implementation it is therefor advantages to store only the indices of the matrix.

$$M = \begin{bmatrix} 0.33835 & 0.16665 & 0 & 0 & 0 & 0 \\ 0.16665 & 0.6667 & 0.16665 & 0 & 0 & 0 \\ 0 & 0.16665 & 0.6667 & 0.16665 & 0. & 0 \\ 0 & 0 & 0.16665 & 0.6667 & 0.16665 & 0 \\ 0 & 0 & 0 & 0.16665 & 0.6667 & 0.16665 \\ 0 & 0 & 0 & 0 & 0.16665 & 0.32835 \end{bmatrix} \tag{5.6}$$

$$b = \begin{bmatrix} 0.30428294 \\ 0.59994273 \\ 0.12453382 \\ 0.23561381 \\ 0.16996559 \\ 0.03644494 \end{bmatrix} \quad \xi = \begin{bmatrix} 0.50655841 \\ 0.79741315 \\ -0.09667314 \\ 0.33661507 \\ 0.16383511 \\ 0.02784168 \end{bmatrix} \tag{5.7}$$

The results attained in Eq. (5.7) result in a mean error of $2.4 \cdot 10^{-14}$ and a mean square error of 0.84. In comparison, the interpolation method results in a mean error of -3.59 and a mean square error of 1.422. To solve a partial differential equation with a similar method, the problem-specific equations for M and b need to be derived. This can be done using two methods; the Galerkin's method and the Ritz method.

5.1.1 Galerkin's Method

Galerkin's method is a method for discretely solving partial differential equations. Galerkin's method is a weighted residual method, where the equation to be solved have there residual set to 0. An example where Galerkin's method can be utilized is with the one-dimensional Poisson equation, as shown in Eq. (5.8).

$$-\frac{\partial^2 u}{\partial x^2} = f \tag{5.8}$$

Equation (5.8) is solved for I, which is the line for $0 < x < L$ with the boundary values of $u(0) = 0$ and $u(L) = 0$. Eq. (5.8) can also be rewritten as Eq. (5.9).

$$f + \frac{\partial^2 u}{\partial x^2} = 0 \tag{5.9}$$

Replacing the exact solution of "u" by the estimate, $\sum_{j=0}^{n} \xi_j \varphi_j$, as shown in Eq. (5.10), the residual is no longer equal to zero, due to the inaccuracies in the estimate.

$$\begin{aligned} r =& f + \frac{\partial^2}{\partial x^2}\left(\sum_{j=0}^{n} \xi_j \varphi_j\right) \neq 0 \\ r =& f + \left(\sum_{j=0}^{n} \xi_j \frac{\partial^2 \varphi_j}{\partial x^2}\right) \neq 0 \end{aligned} \tag{5.10}$$

In Eq. (5.10) the variable φ_j is called the weighting function as it is the basis function that is weighted in order to find the solution. The best solution to the problem is, therefore, one that satisfies Eq. (5.11), so that the residual is orthogonal to the basis functions.

$$\int_I \varphi_i r \, dx = 0, \quad i = 0, 1, \ldots, n \tag{5.11}$$

φ_i is the test function, as it is used to tests for orthogonality. Inserting Eq. (5.10) into Eq. (5.11), Eq. (5.12) is attained.

$$\int_I \varphi_i \left(f + \left(\sum_{j=0}^{n} \xi_j \frac{\partial^2 \varphi_j}{\partial x^2} \right) \right) dx = 0, \quad i = 0, 1, \ldots, n \tag{5.12}$$

Rewriting Eq. (5.12) using integration by parts and enforcing the boundary conditions, Eq. (5.13) is attained.

$$\begin{aligned}
\int_I \varphi_i f \, dx &= \int_I \varphi_i \left(\sum_{j=0}^{n} \xi_j \frac{\partial^2 \varphi_j}{\partial x^2} \right) dx \\
&= \int_I \frac{\partial \varphi_i}{\partial x} \left(\sum_{j=0}^{n} \xi_j \frac{\partial \varphi_j}{\partial x} \right) dx \\
&\quad - \frac{\partial \varphi_i(L)}{\partial x} \left(\sum_{j=0}^{n} \xi_j \varphi_j(L) \right) + \varphi_i(0) \left(\sum_{j=0}^{n} \xi_j \frac{\partial \varphi_j(0)}{\partial x} \right) \\
&= \int_I \frac{\partial \varphi_i}{\partial x} \left(\sum_{j=0}^{n} \xi_j \frac{\partial \varphi_j}{\partial x} \right) dx \\
&= \sum_{j=0}^{n} \xi_j \int_I \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} \, dx, \quad i = 0, 1, \ldots, n
\end{aligned} \tag{5.13}$$

The system of equations to be solved can be written up in matrix form as presented in Eq. (5.14),

$$M\xi = b \tag{5.14}$$

where the matrices are given by Eqs. (5.15) and (5.16)

$$M_{ij} = \int_I \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} \, dx \tag{5.15}$$

$$b_{ij} = \int_I \varphi_i f \, dx \tag{5.16}$$

Galerkin's method is often utilized due to its simplicity.

5.1.2 Ritz Variational Method

Like Galerkin's method, the Ritz variational method is a method for discretely solving partial differential equations. The method is based on calculus of variation, which is similar

to ordinary calculus, but instead of working with functions, it works with functionals. A functional function maps functions to a scalar, whereas a standard function maps a scalar to a different scalar. These can, in a similar way as functions, be used to find stationary points. Stationary points are the points that describe minimums, maximums, and saddle points. Instead of finding a number, it finds a function which minimizes, maximizes, or finds a saddle point of a problem. It is desired to find the function that minimizes the error (the orthogonality) of the approximation to the real solution. A stationary point is found by setting the derivative equal to zero; this is shown in Eq. (5.17) for a functional, F of u.

$$\frac{\delta}{\delta u}F(u) = 0 \tag{5.17}$$

The method is demonstrated by finding the formulation for the one-dimensional Poisson equation Eq. (5.18).

$$-\frac{\partial^2 u}{\partial x^2} = f \tag{5.18}$$

The problem is rewritten in terms of a functional. The result of this is given by Eq. (5.19) [9],

$$F(\tilde{u}) = \frac{1}{2}\int_I \frac{\partial^2 \tilde{u}}{\partial x^2}\tilde{u}\,dx - \frac{1}{2}\int_I \tilde{u}f\,dx - \frac{1}{2}\int_I f\tilde{u}\,dx \tag{5.19}$$

where $\tilde{u}$ is the estimate of u, given by Eq. (5.20).

$$\tilde{u} = \sum_{j=0}^{n}\xi_j\varphi_j \tag{5.20}$$

Inserting this into Eq. (5.19) results in Eq. (5.21).

$$\begin{aligned} F(\tilde{u}) =& \frac{1}{2}\int_I \sum_{j=0}^{n}\xi_j\frac{\partial\varphi_j}{\partial x^2}\sum_{j=0}^{n}\xi_j\varphi_j\,dx - \frac{1}{2}\int_I\sum_{j=0}^{n}\xi_j\varphi_j f\,dx - \frac{1}{2}\int_I f\sum_{j=0}^{n}\varphi_j\xi_j\,dx \\ =& \frac{1}{2}\sum_{j=0}^{n}\xi_j\int_I\frac{\partial^2\varphi_j}{\partial x^2}\varphi_i\,dx\;\xi_i - \sum_{j=0}^{n}\xi_j\int_I\varphi_j f\,dx, \quad i = 0,1,\ldots,n \end{aligned} \tag{5.21}$$

The derivative of the functional with respect to ξ, as this is the desired value, is set to zero.

$$\frac{\delta}{\delta\xi}F(\tilde{u}) = \frac{1}{2}\sum_{j=0}^{n}\xi_j\int_I\frac{\partial^2\varphi_j}{\partial x^2}\varphi_i + \frac{\partial^2\varphi_i}{\partial x^2}\varphi_j\,dx - \int_I\varphi_j f\,dx, \quad i = 0,1,\ldots,n \tag{5.22}$$

The same basis function for the test and weighting variables are utilized, $(\partial^2\varphi_j/\partial x^2)\varphi_i = (\partial^2\varphi_i/\partial x^2)\varphi_j$ resulting in the final form Eq. (5.23).

$$\frac{\delta}{\delta\xi}F(\tilde{u}) = \sum_{j=0}^{n}\xi_j\int_I\varphi_j\frac{\partial^2\varphi_i}{\partial x^2}\,dx - \int_I\varphi_j f\,dx, \quad i = 0,1,\ldots,n \tag{5.23}$$

The final form can be formulated as the matrix system shown in Eq. (5.24),

$$M\xi = b \tag{5.24}$$

where the indices of the matrices are given by Eqs. (5.25) and (5.26).

$$M_{ij} = \int_I \varphi_i \frac{\partial^2 \varphi_j}{\partial x^2}\, dx \tag{5.25}$$
$$= \int_I \frac{\partial \varphi_i}{\partial x}\frac{\partial \varphi_j}{\partial x}\, dx - \int_I \varphi_i(0)\frac{\partial \varphi_j(0)}{\partial x}\, dx - \int_I \frac{\partial \varphi_i(0)}{\partial x}\varphi_j(0)\, dx$$
$$= \int_I \frac{\partial \varphi_i}{\partial x}\frac{\partial \varphi_j}{\partial x}\, dx$$
$$b_{ij} = \int_I \varphi_i f\, dx \tag{5.26}$$

For this specific problem, the Ritz method results in the same formulation as Galerkin's method. Galerkin's method and the Ritz variational method can be used to find a system of equations for the vector wave equation and the wanted boundaries.

5.2 3D FEM Formulation

In this section, the three-dimensional FEM formulation is considered. A meshing structure consisting of first-order tetrahedrons is utilized. These can be structured or unstructured. Higher-order tetrahedrons can be utilized but will not be discussed in this report.

The tetrahedron structure can be described in terms of elements, nodes, and surfaces. These are shown in Fig. 5.9.

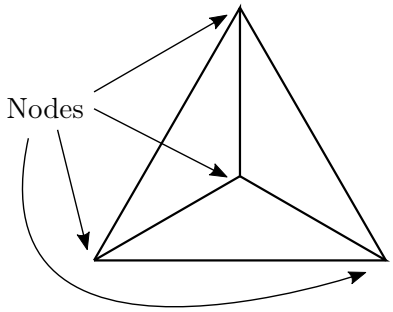


Figure 5.9: Single Tetrahedron element. It has 4 nodes, 6 edges, and 4 faces.

Nodes are the coordinates of the structure's endpoints. An element is one complete tetrahedron and will be indexed with the variable "e" throughout the report. The faces of a tetrahedron consist of four triangles, each constructed between three nodes. These characteristics will later be used to calculate the electromagnetic fields numerically.

In the majority of FEM implementations, a reference-element is utilized [3]. The structure to be simulated is represented in an (x, y, z) domain, whereas the reference-element is represented in an (u, v, w) domain [3]. These reference-elements can then be used to perform all operations. The result is then mapped from the reference-element (u, v, w domain) to the physical-element (x, y, z domain) [3]. The mapping occurs with the use of a transformation. A basis function, as well as a numerical integration, is applied before the transformation to the reference element [3]. Figure 5.10 shows the reference tetrahedron element utilized in this report.

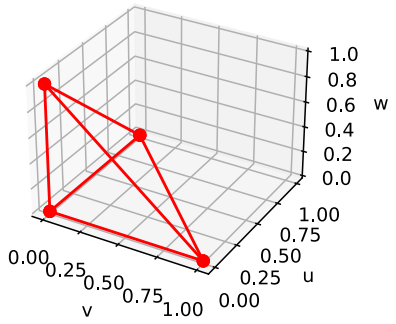


Figure 5.10: Illustration of a reference tetrahedron in its corresponding coordinate system.

When integration is needed on the physical domain, the integration is first conducted on the reference element and then transformed to the physical domain. This mapping is only applicable when the reference element is of the same structure as the physical element (Both structures have to be a tetrahedron). For the FEM system, the basis function is utilized to discretize the solution space.

A tetrahedron has four points or nodes, four surfaces or triangles, and six edges or lines. The reference tetrahedron has nodes in the coordinates shown in Eq. (5.27).

$$\begin{aligned} \boldsymbol{r}_1 &= [0,0,0], \\ \boldsymbol{r}_2 &= [+1,0,0], \\ \boldsymbol{r}_3 &= [0,+1,0], \\ \boldsymbol{r}_4 &= [0,0,+1]. \end{aligned} \tag{5.27}$$

The corresponding basis functions are as shown in Eq. (5.28).

$$\begin{aligned} \varphi_1 &= 1-u-v-w, \\ \varphi_2 &= u, \\ \varphi_3 &= v, \\ \varphi_4 &= w \end{aligned} \tag{5.28}$$

It can be noted that each node has its own basis function and therefore results in 4 basis functions. The basis function for each face of the tetrahedron can be constructed and is expressed in Eq. (5.29).

$$\begin{aligned} \boldsymbol{M}_1 &= 2\left(\varphi_3\tilde{\nabla}\varphi_2\times\tilde{\nabla}\varphi_1+\varphi_2\tilde{\nabla}\varphi_1\times\tilde{\nabla}\varphi_3+\varphi_1\tilde{\nabla}\varphi_3\times\tilde{\nabla}\varphi_2\right) \\ &= 2[u\hat{u}+v\hat{v}+(w-1)\hat{w}] \\ \boldsymbol{M}_2 &= 2\left(\varphi_1\tilde{\nabla}\varphi_2\times\tilde{\nabla}\varphi_4+\varphi_2\tilde{\nabla}\varphi_4\times\tilde{\nabla}\varphi_1+\varphi_4\tilde{\nabla}\varphi_1\times\tilde{\nabla}\varphi_2\right) \\ &= 2[u\hat{u}+(v-1)\hat{v}+w\hat{w}] \\ \boldsymbol{M}_3 &= 2\left(\varphi_2\tilde{\nabla}\varphi_3\times\tilde{\nabla}\varphi_4+\varphi_3\tilde{\nabla}\varphi_4\times\tilde{\nabla}\varphi_2+\varphi_4\tilde{\nabla}\varphi_2\times\tilde{\nabla}\varphi_3\right) \\ &= 2[u\hat{u}+v\hat{v}+w\hat{w}] \\ \boldsymbol{M}_4 &= 2\left(\varphi_3\tilde{\nabla}\varphi_1\times\tilde{\nabla}\varphi_4+\varphi_1\tilde{\nabla}\varphi_4\times\tilde{\nabla}\varphi_3+\varphi_4\tilde{\nabla}\varphi_3\times\tilde{\nabla}\varphi_1\right) \\ &= 2[(u-1)\hat{u}+v\hat{v}+w\hat{w}] \end{aligned} \tag{5.29}$$

Similarly, the 6 edges each have a basis function as shown in Eq. (5.30).

$$\begin{aligned}
\boldsymbol{N}_1 &= \varphi_1\tilde{\nabla}\varphi_2 - \varphi_2\tilde{\nabla}\varphi_1 = (1-w-v)\hat{u} + u\hat{v} + u\hat{w} \\
\boldsymbol{N}_2 &= \varphi_2\tilde{\nabla}\varphi_3 - \varphi_3\tilde{\nabla}\varphi_2 = -v\hat{u} + u\hat{v} \\
\boldsymbol{N}_3 &= \varphi_3\tilde{\nabla}\varphi_1 - \varphi_1\tilde{\nabla}\varphi_3 = -v\hat{u} + (u+w-1)\hat{v} - v\hat{w} \\
\boldsymbol{N}_4 &= \varphi_1\tilde{\nabla}\varphi_4 - \varphi_4\tilde{\nabla}\varphi_1 = w\hat{u} + w\hat{v} + (1-v-u)\hat{w} \\
\boldsymbol{N}_5 &= \varphi_2\tilde{\nabla}\varphi_4 - \varphi_4\tilde{\nabla}\varphi_2 = -w\hat{u} + u\hat{w} \\
\boldsymbol{N}_6 &= \varphi_3\tilde{\nabla}\varphi_4 - \varphi_4\tilde{\nabla}\varphi_3 = -w\hat{v} + v\hat{w}
\end{aligned} \tag{5.30}$$

Where, elements with " ˆ " represent unit vectors. In the Fig. 5.11 an illustration of the N_1 basis function is shown. A three-dimensional cut of the basis function is shown in Fig. 5.11.

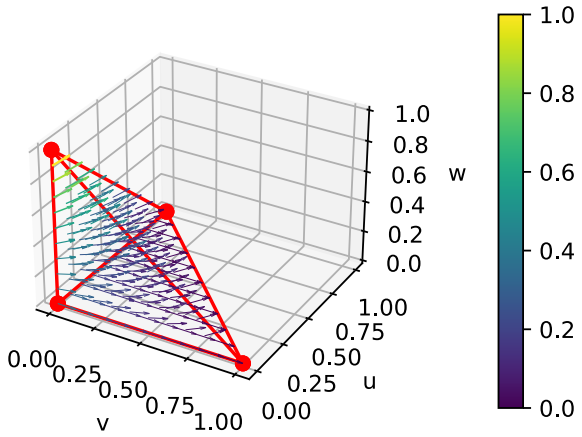


Figure 5.11: Illustration of the basis function $\boldsymbol{N}_1$.

5.2.1 Reference to Physical Element

It is assumed that a meshing structure consisting of three-dimensional tetrahedrons is utilized in the FEM algorithm. Each tetrahedron used to construct the model structure is assigned an index "e". A relation between the reference element and the physical element needs to be constructed, this is done with the mapping $\boldsymbol{r}^e = \boldsymbol{r}^e(u,v,w)$ and is extended in Eq. (5.31).

$$\begin{aligned}
\boldsymbol{r}^e(u,v,w) &= \hat{\boldsymbol{x}}x^e(u,v,w) + \hat{\boldsymbol{y}}y^e(u,v,w) + \hat{\boldsymbol{z}}z^e(u,v,w) \\
&= \sum_{i=1}^{N} \boldsymbol{r}_i^e \varphi_i(u,v,w)
\end{aligned} \tag{5.31}$$

For tetrahedron, $N = 4$ as it consists of 4 nodes (points). $\boldsymbol{r}^e(u,v,w)$ can be expressed as Eq. (5.32).

$$\boldsymbol{r}^e(u,v,w) = (1-u-v-w)r_1^e + u\cdot r_2^e + v\cdot r_3^e + w\cdot r_4^e \tag{5.32}$$

The volume of the reference element $\tilde{V}$, is bound by the conditions in Eq. (5.33).

$$\begin{aligned} 0 &\geqslant u \geqslant 1 - v - w \\ 0 &\geqslant v \geqslant 1 - w \\ 0 &\geqslant w \geqslant 1 \end{aligned} \tag{5.33}$$

The translation of nodes results in the entire reference-element being transformed to the physical domain. This includes edge-elements and face-elements as they can be expressed as a function of nodes. The transformation utilizes linear basis functions, and the transformation is, therefore, a linear transformation.

5.2.2 Reference Element Integration

All operations are conducted on the reference element and then transformed to the physical element. Therefore, the integration is conducted on the reference element. The integration of the reference element with the transformation can be expressed as a change of bases and can be seen in Eq. (5.34).

$$\int_{V^e} f(x,y,z)\, dx\, dy\, dz = \int_{\tilde{V}} f(u,v,w)\, det(\mathbf{J}^e)\, du\, dv\, dw \tag{5.34}$$

Where $det(\mathbf{J}^e)$ is the determinant of the Jacobian. The Jacobian is presented in Eq. (5.35). The volume V^e represents the volume of the physical element at index "e". The volume $\tilde{V}$ represents the volume of the reference element.

$$\mathbf{J}^e = \begin{pmatrix} \partial x^e/\partial u & \partial y^e/\partial u & \partial z^e/\partial u \\ \partial x^e/\partial v & \partial y^e/\partial v & \partial z^e/\partial v \\ \partial x^e/\partial w & \partial y^e/\partial w & \partial z^e/\partial w \end{pmatrix} \tag{5.35}$$

The Jacobian $\mathbf{J}^e$ is calculated under the assumption of the mapping $r^e = r^e(u,v,w)$ [3].

5.3 Cavity Resonator: E- and H-Field Method

A cavity resonator is explored as it has the same mathematical formulation as a waveguide without ports. A cavity resonator is the equivalent of a Quasi-Static weak form formulation. The Quasi-static weak form eigenvalue problem is often used to solve the eigenfrequencies ω of a specified resonator [3]. The Quasi-static form utilizes parts of the Maxwell formulation, specifically Amperes and Faraday's law. These are shown in rewritten form to include the conductivity Eq. (5.36),

$$\begin{aligned} \nabla \times E &= -j\omega \mathbf{B} \\ \nabla \times \frac{\mathbf{B}}{\mu_0} &= (\sigma + j\omega\varepsilon)\mathbf{E} \end{aligned} \tag{5.36}$$

where epsilon (ε) is the permittivity, sigma (σ) is the conductivity, and micro (μ) is the permeability, $\mathbf{B}$ is the magnetic flux given by $\mathbf{B} = \mathbf{H}/\mu$. These two equations need to be bounded in order to be able to be solved numerically. For a perfect electric conductive (PEC) surface, the boundary condition is given by Eq. (5.37),

$$\hat{\boldsymbol{n}} \times \boldsymbol{E} = 0. \tag{5.37}$$

This results in complex eigenfrequencies, where the real part represents the frequency component, and the complex part represents the dampening at a specific frequency [3].

In order to modify Eq. (5.36) to the weak form, the entire volume V of the structure to be simulated, needs to be taken into account. Eq. (5.36) is therefore modified to the volumetric integrals shown in Eq. (5.38) and Eq. (5.39).

$$\int_V \boldsymbol{W}_i^{\mathrm{FL}} \cdot (\nabla \times \boldsymbol{E}) dV = -j\omega \int_V \boldsymbol{W}_i^{\mathrm{FL}} \cdot \boldsymbol{B} dV \tag{5.38}$$

$$\int_V \boldsymbol{W}_i^{\mathrm{AL}} \cdot (\nabla \times \boldsymbol{B}) dV = \mu_0 \int_V \sigma \boldsymbol{W}_i^{\mathrm{AL}} \cdot \boldsymbol{E} dV + j\omega\mu_0 \int_V \epsilon \boldsymbol{W}_i^{\mathrm{AL}} \cdot \boldsymbol{E} dV \tag{5.39}$$

The superscript FL represents Faraday's law, and AL, represents Ampere's law. Equation (5.38) and Eq. (5.39) differ from Eq. (5.36) by weights $\boldsymbol{W}_i^{\mathrm{AL}}$ and $\boldsymbol{W}_i^{\mathrm{FL}}$. The volumetric integral is used in order to utilize the bulk of the structure. Part of Eq. (5.39) ($\int_V \boldsymbol{W}_i^{\mathrm{AL}} \cdot (\nabla \times \boldsymbol{B}) dV$) can be further simplified, and is simplified with the use of integration by parts, with the use of the identity presented in Eq. (5.40).

$$\nabla \cdot \left(\boldsymbol{W}_i^{\mathrm{AL}} \times \boldsymbol{B}\right) = \left(\nabla \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} - \boldsymbol{W}_i^{\mathrm{AL}} \cdot (\nabla \times \boldsymbol{B}) \tag{5.40}$$

The identity presented in Eq. (5.40) can be rearranged to form Eq. (5.41).

$$\boldsymbol{W}_i^{\mathrm{AL}} \cdot (\nabla \times \boldsymbol{B}) dV = \left(\nabla \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} - \nabla \cdot \left(\boldsymbol{B} \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \tag{5.41}$$

The integral is used to simplify $\int_V \boldsymbol{W}_i^{\mathrm{AL}} \cdot (\nabla \times \boldsymbol{B}) dV$ as shown in Eq. (5.42).

$$\begin{aligned}
\int_V \boldsymbol{W}_i^{\mathrm{AL}} \cdot (\nabla \times \boldsymbol{B}) dV &= \int_V \left[\left(\nabla \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} - \nabla \cdot \left(\boldsymbol{W}_i^{\mathrm{AL}} \times \boldsymbol{B}\right)\right] dV \\
&= \int_V \left(\nabla \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} dV - \int_S \left(\boldsymbol{W}_i^{\mathrm{AL}} \times \boldsymbol{B}\right) \cdot \hat{\boldsymbol{n}} dS \\
&= \int_V \left(\nabla \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} dV - \int_S \left(\hat{\boldsymbol{n}} \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} dS \\
&= \int_V \left(\nabla \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} dV
\end{aligned} \tag{5.42}$$

On the boundary, the boundary condition states that $\hat{\boldsymbol{n}} \times \boldsymbol{W}_i^{\mathrm{AL}} = \boldsymbol{0}$ on the surface of the structure i.e the surface is PEC [3]. Based on this, the final problem formulation is shown in Eq. (5.43).

$$\begin{aligned}
&\int_V \boldsymbol{W}_i^{\mathrm{FL}} \cdot (\nabla \times \boldsymbol{E}) dV = -j\omega \int \boldsymbol{W}_i^{\mathrm{FL}} \cdot \boldsymbol{B} dV \\
&\int_V \left(\nabla \times \boldsymbol{W}_i^{\mathrm{AL}}\right) \cdot \boldsymbol{B} dV = \mu_0 \int_V \sigma \boldsymbol{W}_i^{\mathrm{AL}} \cdot \boldsymbol{E} dV + j\omega\mu_0 \int_V \epsilon \boldsymbol{W}_i^{\mathrm{AL}} \cdot \boldsymbol{E} dV
\end{aligned} \tag{5.43}$$

5.4 Matrix Assembly

For ease of implementation, Eq. (5.43) can be represented in vector form as shown in Eq. (5.44).

$$\mathbf{A}\mathbf{z} = \lambda \mathbf{L}\mathbf{z} \tag{5.44}$$

Equation (5.44) can be expressed as Eq. (5.45),

$$\begin{pmatrix} \mathbf{0} & -\mathbf{C} \\ \mathbf{C}^T & -Z_0\mathbf{M}^{(\sigma)} \end{pmatrix} \begin{bmatrix} c_0 B \\ E \end{bmatrix} = \frac{j\omega}{c_0} \begin{pmatrix} \mathbf{M}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{(\epsilon_r)} \end{pmatrix} \begin{bmatrix} c_0 B \\ E \end{bmatrix} \tag{5.45}$$

where each of the matrix and vector-elements are expressed as Eq. (5.46).

$$\begin{aligned} C_{ij} &= \int_{V^e} \boldsymbol{M}_i^e \cdot \left(\nabla \times \boldsymbol{N}_j^e\right)\ dx\ dy\ dz \\ M_{ij}^{(1)} &= \int_{V^e} \boldsymbol{M}_i^e \cdot \boldsymbol{M}_j^e\ dx\ dy\ dz \\ M_{ij}^{(\epsilon_r)} &= \int_{V^e} \epsilon_r \boldsymbol{N}_i^e \cdot \boldsymbol{N}_j^e\ dx\ dy\ dz \\ M_{ij}^{(\sigma)} &= \int_{V^e} \sigma \boldsymbol{N}_i^e \cdot \boldsymbol{N}_j^e\ dx\ dy\ dz \end{aligned} \tag{5.46}$$

and E represents the electric field. The variable B represents the electric flux. Both the magnetic and E-field variables are unknowns. The wave impedance Z_0 is defined as Eq. (5.47).

$$Z_0 = \sqrt{\frac{\mu_0}{\epsilon_0}} \tag{5.47}$$

The eigenvalue λ is defined as $\lambda = \frac{j\omega}{\epsilon_0}$. Elements presented in Eq. (5.46), can be expressed in terms of reference element transformations. The physical basis functions for the edges N_i^e are constructed from the reference element N_i as shown in Eq. (5.48).

$$N_i^e = [J^e]^{-1} N_i \tag{5.48}$$

The reference element is transformed with the use of the Jacobian. The Jacobian is presented in Eq. (5.35). Eq. (5.48) can be expanded to apply to the curl of the physical element as shown in Eq. (5.49).

$$\nabla \times \boldsymbol{N}_i^e = \frac{[\mathbf{J}^e]^T}{\det(\mathbf{J}^e)} \tilde{\nabla} \times \boldsymbol{N}_i \tag{5.49}$$

The faces M_i^e need to be mapped from the reference element to the physical element. This is done with the use of the Jacobian as presented in Eq. (5.50).

$$\mathbf{M_i^e} = \frac{[\mathbf{J}^e]^T}{\det(\mathbf{J}^e)} \boldsymbol{M}_i \tag{5.50}$$

Inserting the above transformations into Eq. (5.46) results in Eq. (5.51).

$$\begin{aligned} C_{ij} &= \int_{\tilde{V}} \left(\frac{[\mathbf{J}^e]^T}{\det(\mathbf{J}^e)} \boldsymbol{M}_i\right) \cdot \left(\frac{[\mathbf{J}^e]^T}{\det(\mathbf{J}^e)} (\nabla \times \boldsymbol{N}_j)\right) \det(\mathbf{J}^e)\ du\ dv\ dw \\ M_{ij}^{(1)} &= \int_{\tilde{V}} \left(\frac{[\mathbf{J}^e]^T}{\det(\mathbf{J}^e)} \boldsymbol{M}_i\right) \cdot \left(\frac{[\mathbf{J}^e]^T}{\det(\mathbf{J}^e)} \boldsymbol{M}_j\right) \det(\mathbf{J}^e)\ du\ dv\ dw \\ M_{ij}^{(\epsilon_r)} &= \int_{\tilde{V}} \epsilon_r \left([\mathbf{J}^e]^{-1} \boldsymbol{N}_i\right) \cdot \left([\mathbf{J}^e]^{-1} \boldsymbol{N}_j\right) \det(\mathbf{J}^e)\ du\ dv\ dw \\ M_{ij}^{(\sigma)} &= \int_{\tilde{V}} \sigma \left([\mathbf{J}^e]^{-1} \boldsymbol{N}_i\right) \cdot \left([\mathbf{J}^e]^{-1} \boldsymbol{N}_j\right) \det(\mathbf{J}^e)\ du\ dv\ dw \end{aligned} \tag{5.51}$$

Eq. (5.51) is only valid if the transformation follows the mapping $r^e = r^e(u, v, w)$. The mapping also applies to conductivity and permittivity. The weak form presented above can be solved with the use of the previously described FEM method.

5.5 Cavity Resonator: E-Field Method

The solution to the eigenvalue problem can also be computed with the use of only the E-field. Instead of using the first-order equations for both the E-field and the H-field, their equations are combined to form a single second-order equation as shown in Eq. (5.52),

$$\nabla \times \left(\frac{1}{\mu_\mathrm{r}} \nabla \times \mathbf{E} \right) - k_0^2 \epsilon_\mathrm{r} \mathbf{E} = 0 \tag{5.52}$$

where, $k_0 = \sqrt{\lambda}$, k_0 ensures the frequency dependence of the formulation. Theoretically, solving Eq. (5.52) results in the same solution as solving the two first-order equations individually. From the resulting E-field, the H-field can be directly calculated. When utilizing this method, only the edge-elements are used.

First, the variations formulation of the problem is set up, including the boundary values. In Eq. (5.53) this is done with the boundary of PEC.

$$\begin{aligned} &\delta F(\mathbf{E}) = 0 \\ &\hat{n} \times \mathbf{E} = 0 \quad \text{on cavity wall} \end{aligned} \quad . \tag{5.53}$$

The variational function formulation is presented in Eq. (5.54).

$$F(\mathbf{E}) = \frac{1}{2} \int_V \left[\frac{1}{\mu_\mathrm{r}} (\nabla \times \mathbf{E}) \cdot (\nabla \times \mathbf{E}) - k_0^2 \epsilon_\mathbf{r} \mathbf{E} \cdot \mathbf{E} \right] \mathrm{d}V \tag{5.54}$$

The E-field of a single element is given by Eq. (5.55).

$$\mathbf{E}^e = \sum_{i=1}^{n} \mathbf{N}_i^e E_i^e = \{E^e\}^\mathrm{T} \{\mathbf{N}^e\} = \{\mathbf{N}^e\}^\mathrm{T} \{E^e\} \tag{5.55}$$

In order to attain the solution of the E-field for all elements, Eq. (5.55) is substituted into Eq. (5.54). An solution for the single element E-field is then attained in Eq. (5.56),

$$F(E^e) = \frac{1}{2} \sum_{e=1}^{M} \left(\{E^e\}^\mathrm{T} [A^e] \{E^e\} - k_0^2 \{E^e\}^\mathrm{T} [B^e] \{E^e\} \right) \tag{5.56}$$

where A and B are defined in Eq. (5.57) and Eq. (5.58).

$$[A^e] = \int_{V^e} \frac{1}{\mu_\mathrm{r}^e} \{\nabla \times \mathbf{N}^e\} \cdot \{\nabla \times \mathbf{N}^e\}^\mathrm{T} \mathrm{d}V \tag{5.57}$$

$$[B^e] = \int_{V^e} \epsilon_\mathrm{r}^e \{\mathbf{N}^e\} \cdot \{\mathbf{N}^e\}^\mathrm{T} \mathrm{d}V \tag{5.58}$$

The entries in the element matrix are then given by Eq. (5.59) and Eq. (5.60).

$$\left[A_{ij}^{e}\right] = \int_{V^e} \frac{1}{\mu_{\mathrm{r}}^{e}} \left\{\nabla \times \mathbf{N}_i^e\right\} \cdot \left\{\nabla \times \mathbf{N}_j^e\right\} \mathrm{d}V \tag{5.59}$$

$$\left[B_{ij}^{e}\right] = \int_{V^e} \epsilon_{\mathrm{r}}^{e} \left\{\mathbf{N}_i^e\right\} \cdot \left\{\mathbf{N}_j^e\right\} \mathrm{d}V \tag{5.60}$$

In order to calculate the matrices A and B, they are expressed in terms of the reference-elements and then transformed into the physical element with the use of the Jacobian as illustrated in Eq. (5.61) and Eq. (5.62).

$$\left[A_{ij}^{e}\right] = \int_{V^e} \frac{1}{\mu_{\mathrm{r}}^{e}} \left(\frac{[\mathbf{J}^e]^T}{\det\left(\mathbf{J}^e\right)} \left(\nabla \times \boldsymbol{N}_i\right)\right) \cdot \left(\frac{[\mathbf{J}^e]^T}{\det\left(\mathbf{J}^e\right)} \left(\nabla \times \boldsymbol{N}_j\right)\right) \det\left(\mathbf{J}^e\right)\ du\ dv\ dw \tag{5.61}$$

$$\left[B_{ij}^{e}\right] = \int_{V^e} \epsilon_{\mathrm{r}} \left([\mathbf{J}^e]^{-1} \boldsymbol{N}_i\right) \cdot \left([\mathbf{J}^e]^{-1} \boldsymbol{N}_i\right) \det\left(\mathbf{J}^e\right)\ du\ dv\ dw \tag{5.62}$$

The final form of the weak form is presented in Eq. (5.63),

$$F = \frac{1}{2}\left(\{E\}^{\mathrm{T}}[A]\{E\} - k_0^2\{E\}^{\mathrm{T}}[B]\{E\}\right) \tag{5.63}$$

where $[A]\{E\} = k_0^2[B]\{E\}$.

5.6 E-Field and H-Field against Only E-Field

In order to get an understanding of the two different methods, a comparison of the resonance frequency is made. A simulation is constructed consisting of a $200\,\mathrm{mm} \times 200\,\mathrm{mm} \times 400\,\mathrm{mm}$ rectangular cavity resonator. The results for the implemented E-field only method and E and H-field method are compared. In addition, the CST version of the E-field only and E and H-field is also included in the comparison shown in Table 5.1.

CST (GHz)	E and H	Only E (GHz)	CST - E and H (GHz)	CST - E (GHz)
0.83795	0.83626	0.83626	0.00169	0.00169
0.83795	0.83749	0.83749	0.00046	0.00046
1.05993	1.05076	1.05076	0.00917	0.00917
1.05993	1.05492	1.05492	0.00501	0.00501
1.05993	1.05579	1.05579	0.00414	0.00414
1.12422	1.10884	1.10884	0.01538	0.01538
1.12423	1.11370	1.11370	0.01052	0.01052
1.29815	1.28013	1.28013	0.01801	0.01801
1.29815	1.28201	1.28201	0.01614	0.01614

Table 5.1: Resonance frequency comparison of the E- and H-field method with E-field only, comparing CST with our implementation.

Our Implementations of FEM are shown in Table 5.1 as "E and H" and "Only E". From Table 5.1 it can be noted that the two methods have the same accuracy in their calculations

of the resonant frequency. Both of the methods have the same error. This indicates that the error originates from the meshing structure and not from the two different FEM methods. The difference between the two methods lie in the the computation time. The method using only the E-field is faster than the method using both the E- and H-field. This is due to the smaller number of variables needed to be solved by the eigenvalue solver. Using Matlabs eigenvalue solver, the computation time for the two methods is presented in Table 5.2.

E and H field	7.922	seconds
Only E field	0.789	seconds

Table 5.2: Computation time of the Different methods.

The noted computational time only considers the time taken for the eigenvalue solver to find the eigenvalues. The computational time presented does not include the mesh generation or assembling of the matrices to be solved. For demonstrational purposes, only 60 eigenvalues are solved. Only the actual solving of the system is taking into account as the rest is implementation-specific. The E-field only method has a smaller matrix to be assembled and is faster than the E- and H-field method.

A disadvantage of the E-field only method is that only the E-field is calculated. The H-field can be calculated in post processing with the E-field. Due to the computational advantages of the E-field only method, this method is utilized for the remainder of the thesis.

5.7 Two port E-Field Homogeneous Waveguide

In order to analyze microwave components such as waveguides and filters, a method of emulating a device with multiple inputs and multiple outputs needs to be found. In this section, a discontinuous homogeneous waveguide will be analyzed. A homogeneous waveguide is a waveguide constructed using PEC or other metalized materials, where the ports need to be homogeneous. Scattering (S) parameters will later be used to analyze the resulting E-fields at the port surfaces. A discontinuous waveguide is illustrated in Fig. 5.12 with two ports S_1 and S_2.

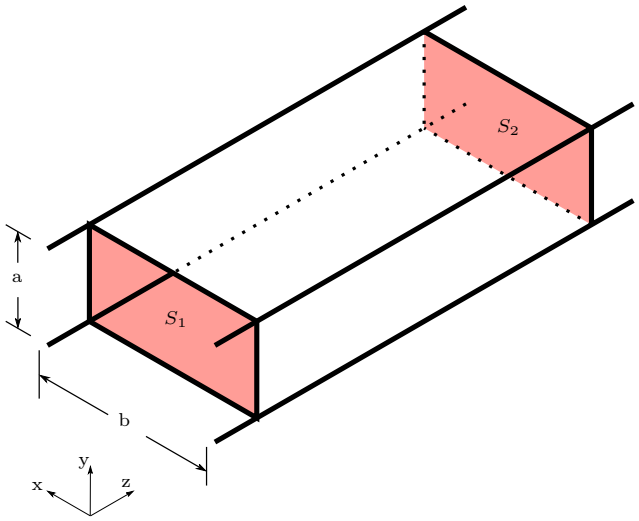


Figure 5.12: Illustration of a two port waveguide with the two ports S_1 and S_2. The height of the waveguide is "a" and the width is "b". The ports should behave has if the waveguide continued, this is illustrated by the extra length beyond the ports.

A waveguide is constructed with the use of PEC for boundaries/walls. The waveguide is filled with a dielectric substrate, in this case, vacuum. The waveguide is bounded at both ends with the surfaces S_1 and S_2. Some assumptions need to be made for the boundary conditions. These assumptions consist of the system only operating in its dominant mode (TE_{10}), and no attenuation is present [9].

The E-field at both ports can be described as a dominant wave (TE_{10}) and a reflected wave due to impedance mismatch. This is expressed mathematically in Eq. (5.64).

$$\begin{aligned}\mathbf{E}(x,y,z) &= \mathbf{E}^{\text{inc}}(x,y,z) + \mathbf{E}^{\text{ref}}(x,y,z) \\ &= E_0\mathbf{e}_{10}(x,y)\mathrm{e}^{-\mathrm{j}k_{z_{10}}z} + RE_0\mathbf{e}_{10}(x,y)\mathrm{e}^{\mathrm{j}k_{z_{10}}z}\end{aligned} \tag{5.64}$$

where, $\mathbf{E}(x,y,z)$ represents the E-field at a given point and can be calculated at the two ports. $\mathbf{E}^{\text{inc}}$ represents the known incident wave applied to the port and $\mathbf{E}^{\text{ref}}$ represents the unknown reflected wave from the port. The power of the applied E-field is expressed as $\mathbf{E}_0$ and R is the reflection constant. For the dominant mode (TE_{10}), e and k_z are expressed as Eq. (5.65) and Eq. (5.66) [9].

$$\mathbf{e}_{10}(x,y) = \hat{y}\sin\frac{\pi x}{a} \tag{5.65}$$

$$k_{z_{10}} = \sqrt{k_0^2 - \left(\frac{\pi}{a}\right)^2} \tag{5.66}$$

Eq. (5.64) can be expressed with the use of the curl of the E-field as shown in Eq. (5.67).

$$\begin{aligned}\hat{n}\times(\nabla\times\mathbf{E}) &= -\hat{z}\times(\nabla\times\mathbf{E}) \\ &= -\mathrm{j}k_{z_{10}}\mathbf{E}^{\text{inc}} + \mathrm{j}k_{z_{10}}\mathbf{E}^{\text{ref}} \\ &= \mathrm{j}k_{z_{10}}\mathbf{E} - 2\mathrm{j}k_{z_{10}}\mathbf{E}^{\text{inc}}\end{aligned} \tag{5.67}$$

Eq. (5.67) can then be further simplified to form Eq. (5.68),

$$\mathbf{U}^{\text{inc}} = \hat{n} \times (\nabla \times \mathbf{E}) + \gamma \hat{n} \times (\hat{n} \times \mathbf{E}) \tag{5.68}$$

where,

$$\begin{aligned} \gamma &= \mathrm{j}k_{z_{10}} \\ \mathbf{U}^{\text{inc}} &= -2\mathrm{j}k_{z_{10}}\mathbf{E}^{\text{inc}} \,. \end{aligned} \tag{5.69}$$

In the case analyzed here, it is assumed that an E-field is applied to S_1 resulting in a reflected field and an incident field. The reflected wave is a result of a mismatch in port impedance. In this example, the second surface S_2 is assumed only to absorb the incident wave and is simplified by removing the reflected component.

The $\mathbf{E}(x, y, z)$ expressed in the previous derivation is for surface S_1. The second surface S_2, is constructed in a similar approach as to S_1, however without the reflected component as shown in Eq. (5.70) and Eq. (5.71).

$$\begin{aligned} \mathbf{E}(x, y, z) &= \mathbf{E}^{\text{trans}}(x, y, z) \\ &= TE_0\mathbf{e}_{10}(x, y)e^{-jk_{z_{10}}z} \end{aligned} \tag{5.70}$$

$$\hat{n} \times (\nabla \times \mathbf{E}) + \gamma \hat{n} \times (\hat{n} \times \mathbf{E}) = 0 \tag{5.71}$$

The boundary conditions for the surfaces are defined in Eq. (5.68) and Eq. (5.71).

It is desired to know which parts of the incident wave are reflected and which are transmitted. In order to find the transmitted wave, the integral of port 2 is calculated. The transmitted wave is divided by the integral of the incident wave to determine the ratio of incident to transmitted wave.

In order to isolate the amount of a specific mode present in the received field (port 2), the analytical formulation of the desired mode is included in the integral of the received field. This is shown in Eq. (5.72) where $e10$ is the formulation of the TE_{10} modes field.

At port 1, the incident and reflected wave are present. The field at port 1 is found similarly to port 2. The reflected wave is then determined by subtracting the incident wave from the field at port 1. The coefficients R and T are used for the reflection coefficient and the transmission coefficients. These are shown in Eq. (5.72).

$$T = \frac{2\mathrm{e}^{\mathrm{j}k_{z_{10}}z_2}}{abE_0} \int_{S_2} \mathbf{E}\,(x, y, z_2) \cdot \mathbf{e}_{10}(x, y)\mathrm{d}S \tag{5.72}$$

$$R = \frac{2\mathrm{e}^{-\mathrm{j}k_{z_{10}}z_1}}{abE_0} \int_{S_1} \mathbf{E}\,(x, y, z_1) \cdot \mathbf{e}_{10}(x, y)\mathrm{d}S - \mathrm{e}^{-2\mathrm{j}k_{z_{10}}z_1} \tag{5.73}$$

For a lossless system such as the system analyzed, $R^2 + T^2 = 1$ needs to be upheld, as all power can only be reflected or received. The final boundary condition for a system consisting of a two-port waveguide with a PEC wall, the boundary can be expressed as Eq. (5.74),

$$\begin{aligned} &\delta F(\mathbf{E}) = 0 \\ &\hat{n} \times \mathbf{E} = 0 \quad \text{on waveguide wall,} \end{aligned} \tag{5.74}$$

where, the function F is expressed as Eq. (5.75) for the two port waveguide.

$$\begin{aligned} F(\mathbf{E}) =& \frac{1}{2}\int_V \left[\frac{1}{\mu_\mathrm{r}}(\nabla\times\mathbf{E})\cdot(\nabla\times\mathbf{E}) - k_0^2\epsilon_\mathrm{r}\mathbf{E}\cdot\mathbf{E}\right]\mathrm{d}V \\ &+ \int_{S_1}\left[\frac{\gamma}{2}(\hat{n}\times\mathbf{E})\cdot(\hat{n}\times\mathbf{E}) + \mathbf{E}\cdot\mathbf{U}^{\mathrm{inc}}\right]\mathrm{d}S \\ &+ \int_{S_2}\left[\frac{\gamma}{2}(\hat{n}\times\mathbf{E})\cdot(\hat{n}\times\mathbf{E})\right]\mathrm{d}S. \end{aligned} \tag{5.75}$$

The surface E-field (E_s) is calculated by separating the surface into "faces". This process is conducted during the meshing of the structure. Only the triangular faces from the tetrahedron need to be taken into account for analyzing the fields on the surface. This process is illustrated in Eq. (5.76).

$$\begin{aligned} \hat{n}\times\mathbf{E}^s &= \sum_{i=1}^{n_\mathrm{s}} \mathbf{S}_i^s E_i^s \\ &= \{E^s\}^\mathrm{T}\{\mathbf{S}^s\} \\ &= \{\mathbf{S}^s\}^\mathrm{T}\{E^s\} \end{aligned} \tag{5.76}$$

In Eq. (5.76) the exponent "s" represents the face indices of all triangular surfaces associated with a specific port surface. Where "S" is the cross product of the normal vector and the basis function $\mathbf{S}_i^s = \hat{n}\times\mathbf{N}_i^s$. Where $\mathbf{N}_i^s$ are the the basis functions of the edges associated with the specified port surface. The function $F(\mathbf{E})$ can then be expressed as Eq. (5.77).

$$F = \frac{1}{2}\sum_{e=1}^{M}\{E^e\}^\mathrm{T}[K^e]\{E^e\} + \frac{1}{2}\sum_{s=1}^{M_\mathrm{s}}\{E^s\}^\mathrm{T}[B^s]\{E^s\} - \sum_{s=1}^{M_{\mathrm{s}1}}\{E^s\}^\mathrm{T}\{b^s\} \tag{5.77}$$

where, M is the amount of elements within the volume, M_s is the number of elements for all surfaces, and M_{S1} is the number of elements for surface S_1. The function in Eq. (5.77) can be expressed in matrix form $[K]\{E\} = \{b\}$. Where the elements are defined in Eq. (5.78).

$$\begin{aligned} [K^e] &= \int_{V^e}\left[\frac{1}{\mu_\mathrm{r}^e}\{\nabla\times\mathbf{N}^e\}\cdot\{\nabla\times\mathbf{N}^e\}^\mathrm{T} - k_0^2\epsilon_\mathrm{r}^e\{\mathbf{N}^e\}\cdot\{\mathbf{N}^e\}^\mathrm{T}\right]\mathrm{d}V \\ [B^s] &= \int_{S^s}\gamma\{\mathbf{S}^s\}\cdot\{\mathbf{S}^s\}^\mathrm{T}\mathrm{d}S \\ \{b^s\} &= \int_{S^s}\{\hat{n}\times\mathbf{S}^s\}\cdot\mathbf{U}^{\mathrm{inc}}\mathrm{d}S \end{aligned} \tag{5.78}$$

The ports can not be placed close to any internal structure in the waveguide if accurate results are to be attained. The ports are formulated for the TE_{10} mode, any other mode generated from internal structures inside the waveguide has to die out before the field at the port is measured. This results in a higher computational cost as a larger structure needs to be meshed and analyzed [9].

5.8 Two-dimensional Port Formulation

The ports assigned to the analyzed waveguide are surfaces; due to this, a three-dimensional FEM formulation can not be used on these surfaces directly. For the port surfaces, a two-dimensional FEM formulation needs to be constructed.

Similar to the three-dimensional FEM formulation presented in Section 5.2, the two-dimensional case utilizes a reference-element [3]. The basics for utilizing a reference element will not be described here, as they have already been described in Section 5.2. A reference triangular element is presented in Fig. 5.13.

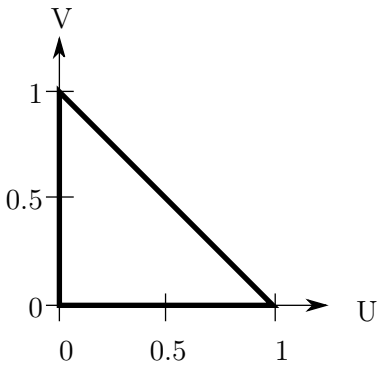


Figure 5.13: Reference triangle with the corner nodes in (0,0), (0,1) and (1,0).

A triangle has 3 points or nodes, 1 surface and 3 edges or lines. The reference triangle has nodes in the coordinates shown in Eq. (5.79).

$$\begin{aligned} \boldsymbol{r}_1 &= [0,0], \\ \boldsymbol{r}_2 &= [+1,0], \\ \boldsymbol{r}_3 &= [0,+1] \end{aligned} \tag{5.79}$$

The corresponding basis functions are as shown in Eq. (5.80).

$$\begin{aligned} \varphi_1 &= 1-u-v, \\ \varphi_2 &= u, \\ \varphi_3 &= v \end{aligned} \tag{5.80}$$

It can be noted that each node has its own basis function and therefore results in 3 basis functions.

The three edges each have a basis function as shown in Eq. (5.81).

$$\begin{aligned} \boldsymbol{N}_1(u,v) &= \varphi_1\tilde{\nabla}\varphi_2 - \varphi_2\tilde{\nabla}\varphi_1 = \hat{\boldsymbol{u}}(1-v) + \hat{\boldsymbol{v}}u \\ \boldsymbol{N}_2(u,v) &= \varphi_2\tilde{\nabla}\varphi_3 - \varphi_3\tilde{\nabla}\varphi_2 = -\hat{\boldsymbol{u}}v + \hat{\boldsymbol{v}}u \\ \boldsymbol{N}_3(u,v) &= \varphi_3\tilde{\nabla}\varphi_1 - \varphi_1\tilde{\nabla}\varphi_3 = -\hat{\boldsymbol{u}}v + \hat{\boldsymbol{v}}(u-1) \end{aligned} \tag{5.81}$$

Where, elements with '^' represent unit vectors. Similarly to the three dimensional case, a mapping function is defined, $\boldsymbol{r}^e = \boldsymbol{r}^e(u,v)$ [3]. This mapping function maps the reference element in the u,v domain to the physical element x,y domain [3]. This is done according to Eq. (5.82).

$$\begin{aligned} \boldsymbol{r}^e(u,v) &= \hat{\boldsymbol{x}}x^e(u,v) + \hat{\boldsymbol{y}}y^e(u,v) \\ &= \sum_{i=1}^{N} \boldsymbol{r}_i^e \varphi_i(u,v), \end{aligned} \tag{5.82}$$

A triangle has 3 nodes and thereby $N = 3$. By inserting the basis function in Eq. (5.82), Eq. (5.83) is attained.

$$\boldsymbol{r}^e(u,v) = (1-u-v)\boldsymbol{r}_1^e + u\boldsymbol{r}_2^e + v\boldsymbol{r}_3^e \tag{5.83}$$

The conditions apply for the reference domain. The conditions are presented in Eq. (5.84).

$$\begin{aligned} 0 &\geqslant u \geqslant 1-v \\ 0 &\geqslant v \geqslant 1 \end{aligned} \tag{5.84}$$

In order to conduct the mapping, the Jacobian is utilized to map from reference element to physical element [3]. The two-dimensional Jacobian is presented in Eq. (5.85), it must be noted that the Jacobian is constructed in three dimensions, with the third dimension being equal to zero. This enables the simple integration of two-dimensional elements in a three-dimensional system.

$$\mathbf{J}^e = \begin{pmatrix} \partial x^e/\partial u & \partial y^e/\partial u & 0 \\ \partial x^e/\partial v & \partial y^e/\partial v & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.85}$$

Based on the previously mentioned Jacobian, the integration mapping can be derived as shown in Eq. (5.86).

$$\int_{S^e} f(x,y)dxdy = \int_{\tilde{S}} f(u,v)\det\left(\mathbf{J}^e\right)dudv \tag{5.86}$$

6 | Software Features

This chapter describes the abilities of our FEM implementation. FEM algorithm. The implemented features are listed below:

- Three dimensional modeling of electric and magnetic fields using FEM.
- Multi-port simulation of homogeneous waveguides.
- Emulation of scattering parameter measurements.

The developed code is written in Matlab and requires a mesh input. Mesh generation is conducted with the use of the meshing software Gmsh. The Python Application Programming Interface (API) of Gmsh is utilized. A structure can be constructed with the use of the Gmsh user interface.

7 | Implementation

In this section, the implementation of the Finite Element Method (FEM) code is analyzed and explained in relation to the FEM mathematical formulation. Only the core of the developed software will be analyzed. The core aspects of the implementation include GMSH meshing, numerical integration, matrix assembly for both volumetric and port structures, and scattering parameters. These functions and features are implemented in the general case, enabling the simulation of an arbitrary structure. The software has been implemented in a combination of Matlab and Python. The FEM algorithm has in its entirety been implemented in Matlab. Python has been used for mesh generation and export of mesh information to a Matlab readable format.

7.1 GMSH Mesh Generation

A mesh from a three-dimensional structure needs to be generated, which can be utilized by our FEM algorithm. Meshing algorithms are outside of the scope of this thesis, and an external meshing solution has been selected. The selected meshing tool is the open-source GMSH application. GMSH is a three-dimensional meshing tool developed for general purpose FEM simulations and not specifically for electromagnetic implementations [14]. It enables meshing using tetrahedrons and can be used for both structured and unstructured meshes, with the mesh density being user-definable [14]. Structures can be created using the integrated user interface with parametric modeling and can be exported to GMSH specific files (*.geo*). Similarly, surfaces can be selected and defined for port specifications. The GMSH software user interface is shown in Fig. 7.1.

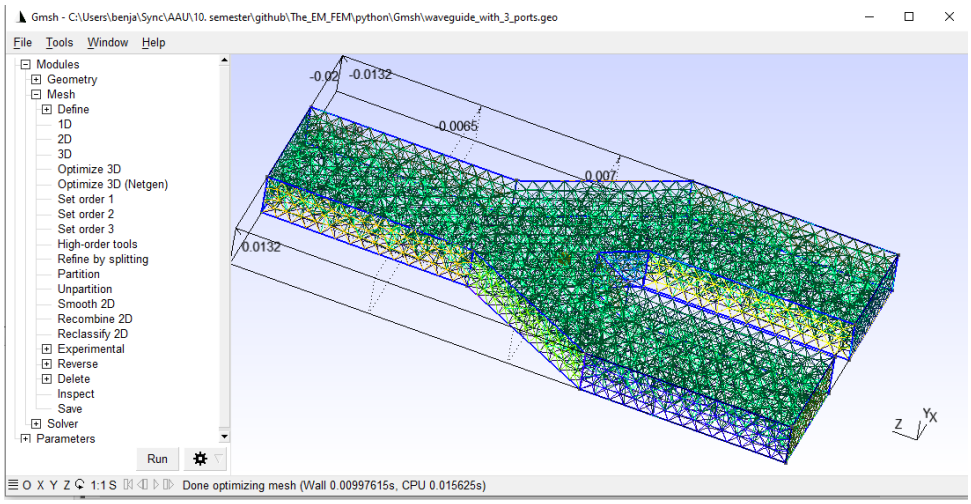


Figure 7.1: Three port power divider meshed in the GMSH software.

A three-dimensional tetrahedron meshed three-port power divider can be seen in Fig. 7.1. GMSH features various programming application programming interfaces (API), of which the Python API is utilized. The Python API is used to mesh the specific *.geo* files and export the information needed for our Matlab implemented FEM algorithm. The parameters needed from the mesh for our FEM implementation are presented in Section 7.1.

Varible	Description
$ed2no\_all$	Edge id to node ids for all
$ed2no\_pec$	Edge id to node ids for PEC
$el2ma$	Element id to material id
$el2no$	Element id to node id all
$fa2no\_all$	Face id to node id all
$no2xyz$	node id to x, y, z coordinates
$port\_fac2no\_list$	port triangles ids to nodes ids

Table 7.1: FEM algorithm-specific meshing variables exported from Python to Matlab readable file.

Similarly, parameters needed for the purpose of plotting only are presented in Section 7.1.

Variable	Description
$ed2no\_port$	Edge id to node id for port surfaces

Table 7.2: Plot specific meshing variables exported from python to Matlab readable file.

Other meshing applications can be utilized as long as they can supply the input parameters listed in Section 7.1.

7.2 Numerical Approximation of Integrals

Integration for both the volumetric as well as the port formulation is needed in our FEM algorithm implementation. Numerical integration is conducted on the reference element; in order to do this, the quadrature rule is utilized. The quadrature rule is defined in Eq. (7.1) for the three-dimensional case.

$$\int_V f(u,v,w)\,du\,dv\,dw \approx \sum_{i=1}^{N} \omega_i f(u_i,v_i,w_i) \tag{7.1}$$

The variable ω_i is the weight associated with the quadrature rule and u_i, v_i, w_i are the quadrature points [3]. The weights and constants (u_i, v_i, w_i) can be seen in Table 7.3.

i	u -coordinate	v -coordinate	w -coordinate	Weight
1	0.5854101966249685	0.1381966011250105	0.1381966011250105	$0.041\bar{6}$
2	0.1381966011250105	0.5854101966249685	0.1381966011250105	$0.041\bar{6}$
3	0.1381966011250105	0.1381966011250105	0.5854101966249685	$0.041\bar{6}$
4	0.1381966011250105	0.1381966011250105	0.1381966011250105	$0.041\bar{6}$

Table 7.3: Quadrature weights associated with the quadrature points [3].

An illustration of these points is shown in Fig. 7.2.

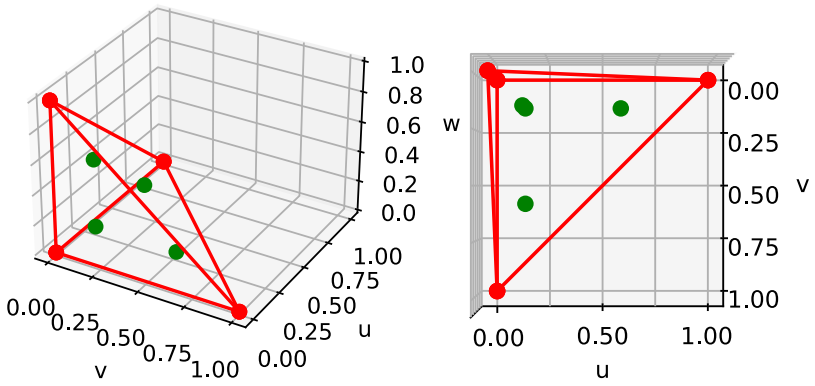


Figure 7.2: Two viewers of the quadrature points, the points where the function that is going to be integrated, should be sampled in the tetrahedron.

The quadrature rule is only a valid approximation if the function can be represented by a $2n-1$ degree polynomial, where "n" is the number of samples taken of the function. As a result, the quadrature rule is valid for our implementation.

7.3 Matrix Assembly

The matrix for the complete system (port surfaces, volume, and PEC boundary) is implemented based on the theory presented in Section 5.7. The matrix assembly procedure is separated into two; the three-dimensional volume and the two-dimensional ports. The

system of equations to be solved are combined in matrix form, as shown in Eq. (7.2).

$$\left(\begin{bmatrix} K_{11} & \dots & K_{1M} \\ \vdots & \ddots & \vdots \\ K_{M1} & \dots & K_{MM} \end{bmatrix} + \begin{bmatrix} B_{11} & \dots & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & \dots & B_{MM} \end{bmatrix} \right) \begin{bmatrix} E_1 \\ \vdots \\ E_M \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix} \tag{7.2}$$

The index "M" represents the number of edges in the meshed structure. In order to solve the FEM formulation for the complete system, the individual equations within the system need to be constructed. These are the "K" matrix, "B" matrix, and "b" vector. For the "K" and the "B" matrix, individual entries represent the relationship between two edges, the column, and the row index represent all edges. The **E** vector describes the E-field at each edge. Similar to **E**, there is a "b" entry for each edge. **E** represents the scaling factor by which the edge basis function needs to be scaled in order to attain the correct solution. In order to know the location of each edge, it is assigned a unique edge ID in the range of 1 to M. This edge ID is used for the indexing of the matrices and vectors.

7.3.1 Volumetric Matrix Assembly

A matrix needs to be derived to relate the propagation properties to the system's volume, which is done by the "K" matrix. Since each edge only has a relationship with edges that are part of the same tetrahedron, the global matrix can be calculated in smaller element matrices, representing individual tetrahedrons. These smaller matrices are constructed using local edge ID numbers. The local ID numbers are integers from 1 to 6, as there are 6 edges in a tetrahedron. The global edge number must be used to assemble the global matrix, as the individual calculated element matrix values are inserted into the global matrix. The majority of edges are part of multiple elements, and if indices from multiple element matrices are to be added to the same entry in the global matrix, they have to be summed. The process is illustrated in Fig. 7.3.

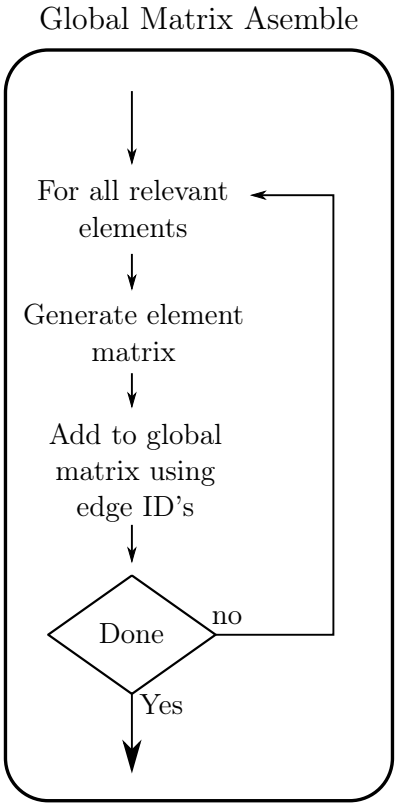


Figure 7.3: Flow diagram of assembling the global matrices from element matrices.

The values in the element matrix are calculated based on the equations in Eq. (5.78). In order to do so, the equations are expressed in terms of individual elements within the matrices. This is shown for the "K" matrix in Eq. (7.3).

$$\left[K_{ij}^{e}\right] = \int_{V^e} \left[\frac{1}{\mu_{\mathrm{r}}^{e}} \left\{\nabla \times \mathbf{N}_i^e\right\} \cdot \left\{\nabla \times \mathbf{N}_j^e\right\} - k_0^2 \epsilon_{\mathrm{r}}^{e} \left\{\mathbf{N}_i^e\right\} \cdot \left\{\mathbf{N}_j^e\right\}\right] \mathrm{d}V \tag{7.3}$$

The element basis function, $\mathbf{N}^e$, is unknown and needs to be transformed from the reference-element into the element basis function. The transformations are conducted as described in Section 5.2.1, with the use of the Jacobian and the determinant of the Jacobian. Due to this, the basis functions are initially defined in terms of the reference function. Only the points used in the numerical integration are used since these are the ones needed for the integration procedure. The nodal functions are shown in Listing 7.1.

```
up{1} = 1 - q2u(1,:) - q2u(2,:) - q2u(3,:);
up{2} = q2u(1,:);
up{3} = q2u(2,:);
up{4} = q2u(3,:);
```

Listing 7.1: Nodel basis fucntions φ.

The variable "$q2u$" is the u, v, and w coordinate for the individual points used for the integration in the reference tetrahedron. The first index in the matrix $q2u$ represents the u, v, and w (1 - 3) coordinate of the integration point, whereas the second index represents which integration point, out of four, in the tetrahedron is selected. The gradient of the node basis functions are constant and are found as shown in Listing 7.2.

```
ug{1} = [-1 -1 -1]';
ug{2} = [+1 0 0]';
ug{3} = [0 +1 0]';
ug{4} = [0 0 +1]';
```

Listing 7.2: gradient of node basis functions $\nabla\varphi$.

The edge basis functions are defined by the reference tetrahedron and are shown in Listing 7.3.

```
uin{1} = ug{2}*up{1} - ug{1}*up{2};
uin{2} = ug{3}*up{2} - ug{2}*up{3};
uin{3} = ug{1}*up{3} - ug{3}*up{1};
uin{4} = ug{4}*up{1} - ug{1}*up{4};
uin{5} = ug{4}*up{2} - ug{2}*up{4};
uin{6} = ug{4}*up{3} - ug{3}*up{4};
```

Listing 7.3: Creation of the element matrix.

An illustration of "N_1" ($uin\{1\}$) is shown in Fig. 7.4. A single cutting plane is illustrated where the "w" coordinate is zero. The points utilized in the integration code are highlighted in the figure by the black arrows. These are used with the quadrature rule to integrate as discussed in Section 7.2.

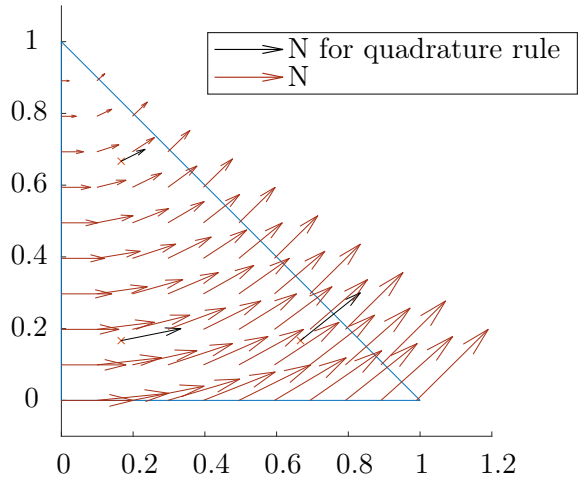


Figure 7.4: The edge basis functions with the integration point highlighted for a cutting plane of the tetrahedron, where w = 0.

The curl of the "N" basis functions are implemented in Listing 7.4.

```
ouTmp = ones(size(q2w));
ucn{1} = 2*cross(ug{1},ug{2})*ouTmp;
ucn{2} = 2*cross(ug{2},ug{3})*ouTmp;
ucn{3} = 2*cross(ug{3},ug{1})*ouTmp;
ucn{4} = 2*cross(ug{1},ug{4})*ouTmp;
ucn{5} = 2*cross(ug{2},ug{4})*ouTmp;
ucn{6} = 2*cross(ug{3},ug{4})*ouTmp;
```

Listing 7.4: curl of edge basis functions $\nabla \times N$.

In order to attain the basis function for the desired physical element, the basis functions for the reference-element needs to be transformed; this is done using the Jacobin. It is derived based on the points and the gradient of the nodal basis function, as shown in Listing 7.5.

```
jac = zeros(3);
for iIdx = 1:4
    jac = jac ...
        + [xyz(1,iIdx)*ug{iIdx}, ...
           xyz(2,iIdx)*ug{iIdx}, ...
           xyz(3,iIdx)*ug{iIdx}];
end
```

Listing 7.5: curl of edge basis functions $\nabla \times N$.

The Jacobian is utilized to find the transformations of the "**N**" and the $\nabla \times \mathbf{N}$ basis functions.

```
map_ccs = inv(jac);    % mapping for curl-conforming space
map_dcs = jac'/det_jac; % mapping for div-conforming space
for iIdx = 1:6
    gin{iIdx} = map_ccs*uin{iIdx};
    gcn{iIdx} = map_dcs*ucn{iIdx};
end
```

Listing 7.6: Transformation from reference space (u,v,w) to physical (x,y,z).

After the transformation from the reference-space to physical-space, the basis function shown in Fig. 7.5 is attained. The basis function is shown for $w = 0$. Since this is a larger triangle than the reference, the vectors values are small relative to the triangle and still need to be adjusted. This is due to the transformation only being a transformation of the input space and not its function. This means that in order for the integral to be correct, the function needs to be scaled by the change of volume. The determinant of the Jacobian gives this change of volume.

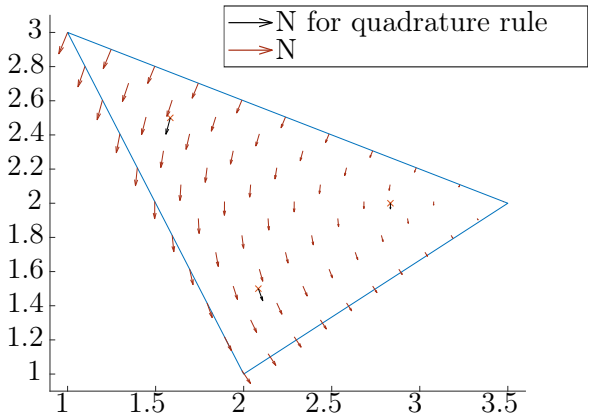


Figure 7.5: The basis function after it has been transformed for the triangle with the nodes (1,3), (2,1), and (3,2). The change in the area has not scaled the function vector values.

The code for calculating a single element matrix is given in Listing 7.7.

```
    for iIdx = 1:6
        for jIdx = 1:6
            er = ma2er(q2x(1,:),q2x(2,:),q2x(3,:));
            ipTmp = (1./mu).*sum(gcn{iIdx}.*gcn{jIdx} - (k0^2)*er.*gin{iIdx}.* gin{jIdx});
            KElMtx_EE(iIdx,jIdx) = ipTmp * q2w' * det_jac;
        end
```

```
end
```

Listing 7.7: Creation of the element matrix for the volume.

In order to calculate the integral of Eq. (7.3) in the Listing 7.7, the dot product is used. The dot product is used due to it being a vector function and it describes how orthogonal two functions are compared to each other. The dot product is calculated using the sum function after the vectors have been element-wise multiplied. The final integration across all triangles is done by the multiplication of "$q2w$" and "$ipTemp$", as shown in Listing 7.7, with "$q2w$" being the weights generated from the quadrature rule. Furthermore, the result is scaled with the Jacobian determinant to compensate for the volumetric change.

7.3.2 Two-dimensional Port Boundary Matrix Assembly

Matrices need to be derived to relate the boundary conditions to the surface of the ports within the system. This is done by the "B" matrix and "b" vector. The "B" matrix represents the absorbing aspect of the ports and is therefore applied to all ports within the system. The "b" matrix represents the incident wave aspect of the system and is therefore only applied to the port which is to be excited. The "B" matrix and the vector "b" hold information regarding all edges. However, only the edges of which are part of the triangles used in the port surfaces need to be considered.

Both the "B" matrix and the "b" vector have the dimensions discussed in the introduction of this section. Elements not included in the surface triangles, however, are zero. These exist due to the ports being two-dimensional and therefore being triangles and not tetrahedrons. The element matrix of which the global matrix is constructed is assembled from single triangles. Similarly, the vector of which the global matrix is constructed is created from single triangles. For the "B" matrix, the equation for each index is given by Eq. (7.4).

$$\left[B_{ij}^{s}\right] = \int_{S^s} \gamma \left\{\mathbf{S}_i^s\right\} \cdot \left\{\mathbf{S}_j^s\right\} \mathrm{d}S \tag{7.4}$$

In Eq. (7.4), the basis function '$\mathbf{S}$" is utilized and is defined as $\mathbf{S} = n \times \mathbf{N}$. The variable "$n$" is normal to the port pointing away from the structure, and "$\mathbf{N}$" is the basis function associated with the specific edge. "$\mathbf{N}$" is found in a similar fashion as previously for the tetrahedron but only for a triangle. Similarly, "$\mathbf{N}$" is mapped from the references-element to the physical-element. The transformation is done by using the two-dimensional Jacobian. There is no need to transform "$\mathbf{S}$" basis function as it is constructed from the already transformed "$\mathbf{N}$" basis function, as shown in Fig. 7.6.

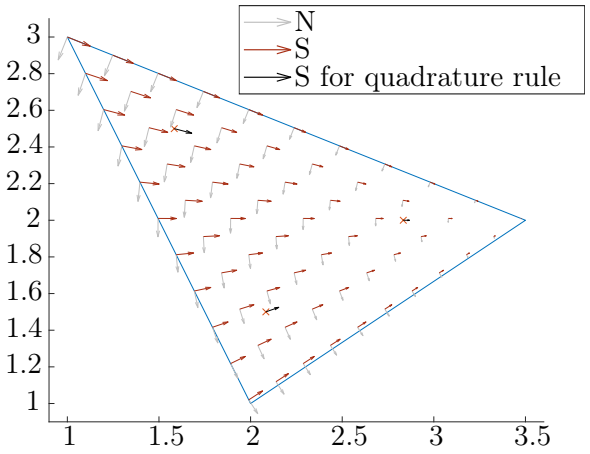


Figure 7.6: **S** basis function.

An illustration of the "**S**" and "**N**" basis functions is shown in Fig. 7.6, it can be noted that the new basis function "**S**", is a 90° rotation of the "**N**" basis function. In Listing 7.8 the implementation of the "**S**" basis function is shown.

```
usn{1} = cross(n,gsn{1});
usn{2} = cross(n,gsn{2});
usn{3} = cross(n,gsn{3});
```

Listing 7.8: calculation of the S basis function.

The implementation of the element matrix for the global matrix "B" is similar to the one used to find the "K" matrix. The dot product of the vectors is utilized, the triangle is integrated, and the area change scales the triangle.

```
for iIdx = 1:3
    for jIdx = 1:3
        ipTmp = sum(usn{iIdx}.* usn{jIdx});
        BElMtx_EE(iIdx,jIdx) = gamma* ipTmp * q2w' * det_jac;
    end
end
```

Listing 7.9: calculation of the element matrix for the global matrix B.

The equations for the entries in the "b" vector are presented in Eq. (7.5). This formulation includes the cross product "**S**" with the normal "n".

$$\{b_i^s\} = \int_{S^s} \{\hat{n} \times \mathbf{S}_i^s\} \cdot \mathbf{U}^{\text{inc}} \mathrm{d}S \tag{7.5}$$

This can be evaluated directly on the "**S**" basis function, as shown in Listing 7.10.

```
usnn{1} = cross(n,usn{1});
usnn{2} = cross(n,usn{2});
usnn{3} = cross(n,usn{3});
```

Listing 7.10: calculation of the $n \times S$ basis function.

An illustration of $n \times \mathbf{S}$ is shown in Fig. 7.7. The expected result of a 90° turn relative to "**S**" can be observed.

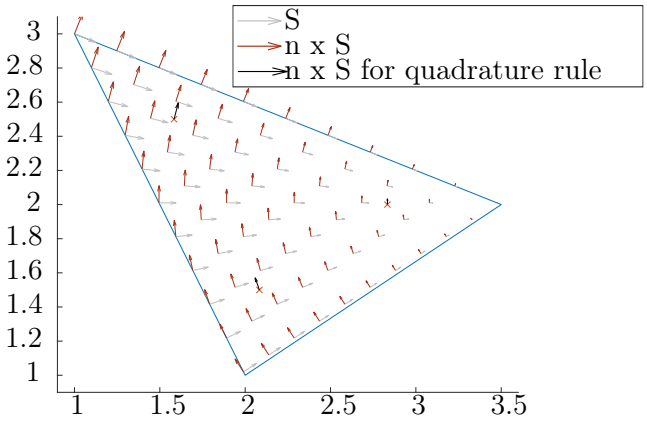


Figure 7.7: $n \times \mathbf{S}$ basis function.

The implementation of $\mathbf{U}^{\text{inc}}$ is a direct implementation of the two-dimensional analytical E-field. The implementation calculates the values for the E-fields at the integration points as illustrated in Listing 7.11. It is, however, not scaled to the triangle area; this is done by the multiplication of the Jacobian in the integration.

```
Uinc = -2j*k_z10*E0*[0,0,0;...
                     sin((pi*(q2x(1,:)+offset)) / a );...
                     0,0,0]...
                     .*exp(-1j*k_z10*Z); %needs z in exponent
```

Listing 7.11: calculation of the $n \times S$ basis function.

The calculation of the element vector is given in Listing 7.12 and uses the dot product, as the goal is to find how orthogonal $\mathbf{U}^{\text{inc}}$ is to the basis functions.

```
for iIdx = 1:3
    ipTmp = sum(usnn{iIdx} .* Uinc);
    bElMtx_EE(iIdx) = (ipTmp * q2w')*det_jac;
end
```

Listing 7.12: Element vector for global vector "b".

7.3.3 PEC Boundary

The PEC boundary has to be included in the system matrix. The E-field in a PEC material is always equal to zero. This can be represented within the system matrix by equating the equation for the edges, which are PEC, to zero; this is illustrated in Eq. (7.6).

$$\begin{bmatrix} K_{11} & & \dots & & K_{1N} \\ \vdots & & & & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & & & & \vdots \\ K_{N1} & & \dots & & K_{NN} \end{bmatrix} \begin{bmatrix} E_1 \\ \vdots \\ E_{PEC} \\ \vdots \\ E_N \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ 0 \\ \vdots \\ b_N \end{bmatrix} \tag{7.6}$$

Since the solution for the equations of the PEC structure is known (E_{PEC} can only be zero), it can be removed from the system of equations without affecting the solution. This will decrease the computation time of the complete system as fewer variables and equations need to be solved.

7.4 Scattering Parameters

Scattering parameters, also known as S-parameters, characterize a linear RF network using reference impedance matched loads at all ports. S-parameters can be used to characterize any linear RF network with one or more ports. No information is needed of the circuit itself other than its ports and its port design impedance. It is, therefore, a black box testing approach. An S-parameter measurement or simulation is unitless and specifies the amount of energy transmitted or received and the reflected energy [5]. A sample two-port linear RF network is presented in Fig. 7.8 and mathematically presented in Eq. (7.8). S-parameters are part of all commercial CEM software suites discussed in this thesis. Our solution is optimized for homogeneous waveguide structures, specifically filters, and it has been decided to include S-parameters in our FEM implementation.

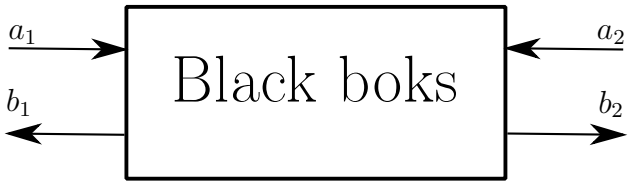


Figure 7.8: Arbitrary two port linear RF network, where a_1 and a_2 represents the incident waves. b_1 and b_2 represents the outgoing waves.

For a sample two-port device, "a_1" represents the incident wave to port 1, "b_1" represents the reflected wave, and "b_2" represents the transmitted wave. Similarly, for port 2, if "a_2" represents an incident wave to port 2, "b_2" represents the reflected wave, and "b_1" represents the transmitted wave [5]. These can be combined in a S-parameter matrix as shown in Eq. (7.7).

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{7.7}$$

$$\begin{aligned} S_{11} &= \left.\frac{b_1}{a_1}\right|_{a_2=0} \quad S_{12} = \left.\frac{b_1}{a_2}\right|_{a_1=0} \\ S_{21} &= \left.\frac{b_2}{a_1}\right|_{a_2=0} \quad S_{22} = \left.\frac{b_2}{a_2}\right|_{a_1=0} \end{aligned} \tag{7.8}$$

Our implementation of FEM, can only simulate S_{11} and S_{21}. In order to be able to simulate S_{22} and S_{12}, one would need to apply the incident wave to port 2 instead of port 1, meaning that two or more simulation are needed if the structure is asymmetrical (e.g. three-port power divider). For our FEM implementation, the linear S-parameters can be calculated using the incident wave and the transmitted wave. To find how much of the wave is transmitted and reflected the transmission and reflection coefficients from Eqs. (7.9) and (7.10) are used.

This is expressed in Eqs. (7.9) and (7.10).

$$S_{21} = T = \frac{2\mathrm{e}^{\mathrm{j}k_{z_{10}}z_2}}{abE_0} \int_{S_2} \mathbf{E}\left(x, y, z_2\right) \cdot \mathbf{e}_{10}(x, y)\mathrm{d}S \tag{7.9}$$

$$S_{11} = R = \frac{2\mathrm{e}^{-j\kappa_{z_{10}}z_1}}{abE_0} \int_{S_1} \mathbf{E}\left(x, y, z_1\right) \cdot \mathbf{e}_{10}(x, y)\mathrm{d}S - \mathrm{e}^{-2jk_{z_{10}}z_1} \tag{7.10}$$

The liner S-parameters are often expressed in dB, therefore both cases are expressed in this thesis. The dB versions of the S-parameters are expressed in Eqs. (7.11) and (7.12).

$$S_{11}^{dB} = 20\log(|S_{11}|) \tag{7.11}$$

$$S_{21}^{dB} = 20\log(|S_{21}|) \tag{7.12}$$

Both the linear and dB versions of the S-parameters can directly be compared to S-parameters generated by commercial software suites.

8 | Performance Analysis

Our implementation of the FEM algorithm is tested and compared with both the commercially available CST Studio, using CST-FEM, CST-FIT, and the analytical solution. The test structure to be used is a section of waveguide with the dimensions 0.20 m by 0.10 m, and a length of 0.40 m, constructed with PEC and the interior is a vacuum, as shown in Fig. 8.1. The incident wave is applied to port 1, and the transmitted wave is received at port 2.

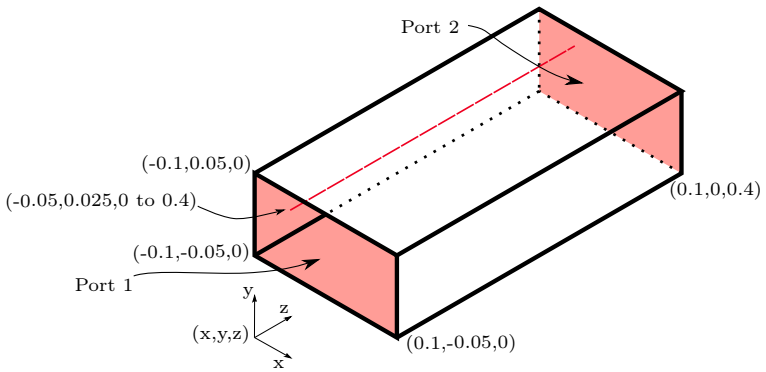


Figure 8.1: Two-port waveguide 0.2 m x 0.1 m x 0.4 m. Along the red line, the E-field is sampled and studied.

The Y component (Ey) of the E-field is sampled along the red dotted line, as shown in Fig. 8.1. Some higher-order modes are 0 at the center point of the port, such as TE_{20}. In order to ensure that the higher-order modes, if such exist, are taken into account, the line over which the sampling is conducted is off-centered (x=0.1 m, y= 0.05 m, and z=0:0.4 m). Apart from the E-field results, the S-parameters are calculated with the use of each method (CST-FIT, CST-FEM, Our Implementation) and presented in this chapter.

8.1 Mesh Density Selection

In this section, the waveguides Y component of the E-field calculated by using our implementation along the dotted line in Fig. 8.1 is used for comparison purposes in this section. In order to conduct a test of our FEM implementation, the mesh resolution should be of the right density to ensure low numerical error. In order to find a mesh with sufficient density, multiple simulations with different mesh are run. Each simulation instance is run for five frequency points, 0.25, 0.55, 0.85, 1.15, and 1.45 GHz. An initial simulation consisting of a mesh of 736 unstructured tetrahedrons is conducted. The mesh

density is increased in steps, the mean square difference to the previous E-field magnitude is calculated with the current mesh and the previous mesh. Individual samples are shown with "*", and are calculated according to Eq. (8.1),

$$C(mesh, f) = \frac{1}{N}\sum_{i=1}^{N}(|\mathbf{E}_{y,mesh,i,f}| - |\mathbf{E}_{y,mesh-1,i,f}|)^2 \tag{8.1}$$

where, $C(, f)$ is the function for the points defining the curves in Fig. 8.2. The E-field $\mathbf{E}_{y,mesh-1,i,f}$ represents the Y component of the E-field, for the previous mesh for frequency f at sample i on the red dotted line in Fig. 8.1. The subscript $mesh$ represents a specific set of tetrahedrons, whereas $mesh - 1$ represents the previous set of tetrahedrons. Each curve for the individual frequency is plotted on a log-log graph in Fig. 8.2. Ideally, the optimal mesh density can be found when the field stops changing or has approached an asymptote. This can, however, not occur; there will always be a change in the E-field when changing the mesh. Therefore, a small change in the E-field is an indication that the E-field has converged to its actual value, and an increase in the mesh density will only result in small changes.

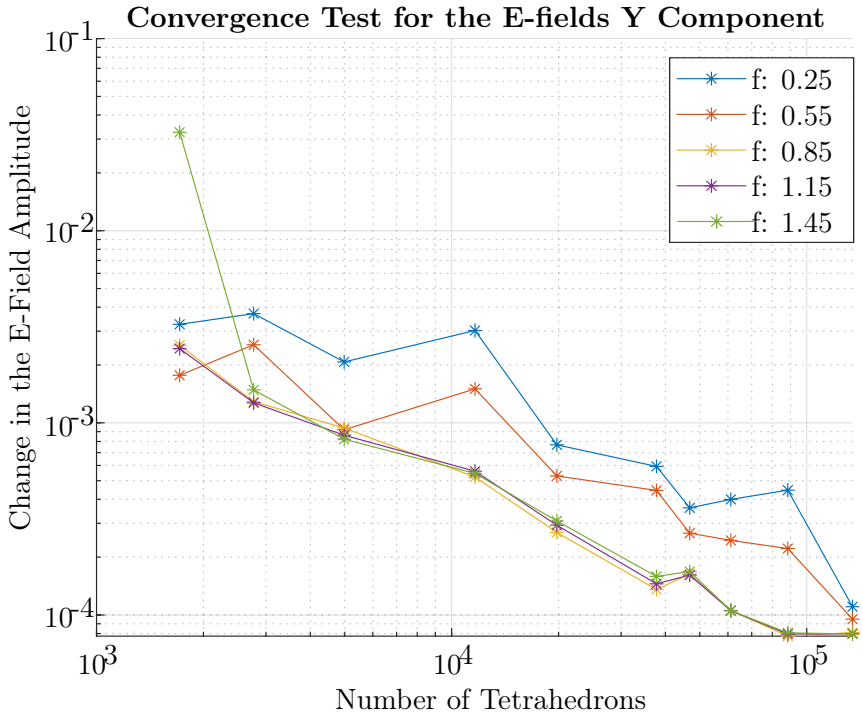


Figure 8.2: The amplitude convergence of the calculated E-fields. When the mesh increases, the E-field changes. The larger the mesh, the smaller the change in the E-field from mesh to mesh.

It can be observed that the curves for all frequencies have a linear regression, and therefore it can be concluded that the change in E-field decreases.

The E-field approaches its actual value when the variation of the field becomes negligible by increasing the mesh and adding more tetrahedrons results in a smaller change per tetrahedron. A large increase in tetrahedrons, therefore, only results in a minor increase in

inaccuracy. The blue and the orange curves represent the frequencies 0.25 and 0.55 GHz. It can be observed that the blue and the orange curves have a greater change in E-field compared to the passband frequencies. This could potentially be due to them being below the cut-off frequency. In the range 37000 to 88000 tetrahedrons, a change of $3.5 \cdot 10^{-4}$ V^2/m^2 occurs in the passband. We have chosen to utilize a mesh with 46831 tetrahedrons. This mesh size was selected because the commercial FEM implementation in CST utilizes 46948 unstructured first order tetrahedrons; the number 46831 tetrahedrons are selected for comparison purposes. Selecting a mesh greater than 46831 will only result in a minor change in the E-field.

8.2 E-field Comparison

In this section, the fields for the different methods Analytical, CST-FIT, CST-FEM, and Our Implementation will be analyzed and compared with each other using Ey along the dotted line on the waveguide in Fig. 8.1 at the frequencies; 0.25, 0.55, 0.85, 1.15, and 1.45 GHz for all implementations. The results for the different frequency points obtained at the four CEM implementations are illustrated in Fig. 8.3.

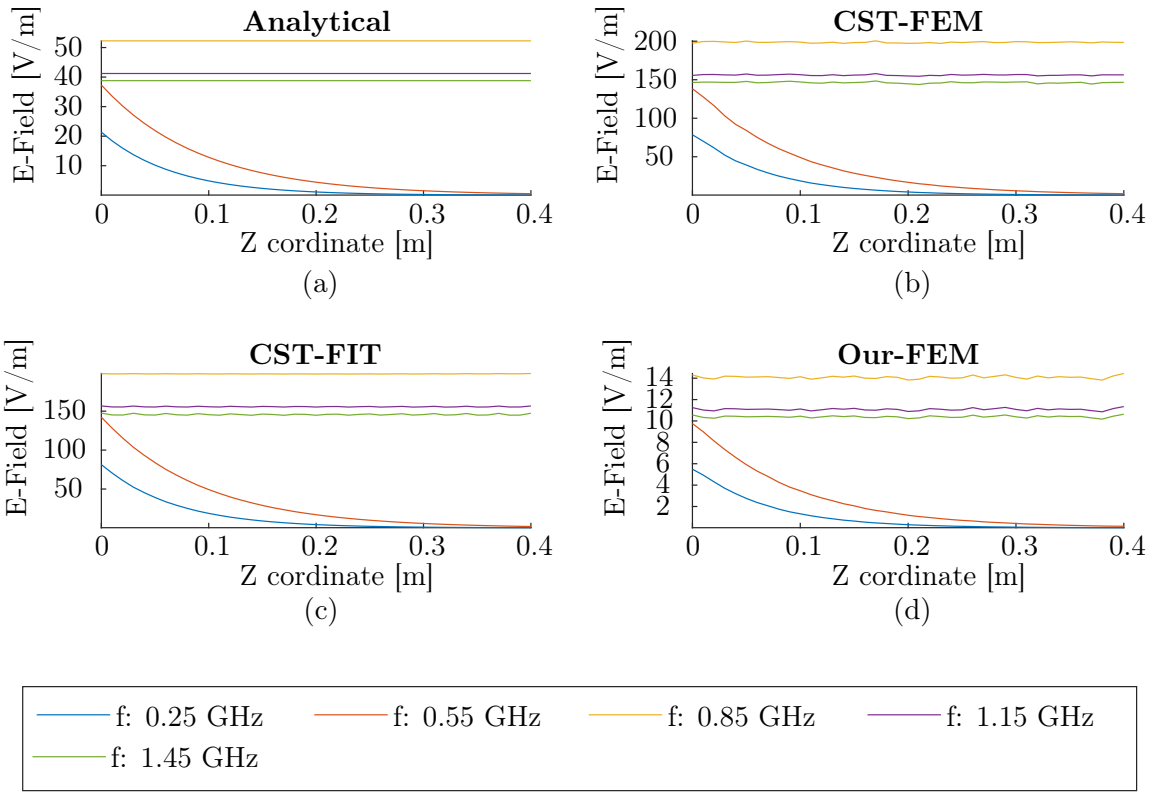


Figure 8.3: The E-field magnitude of the Y component for the different frequencies along the red dotted line, with each graph, represents a different implementation.

It can be observed from "a", "b", "c" and "d" in Fig. 8.3 that an amplitude difference exists. The source for the amplitude difference between all implementations is most likely due to the fact that each implementation uses a different input power. Comparing individual frequency curves for each implementation suggests a fixed multiplier factor.

The implementations are tested for a constant multiplication factor. This is done in order to verify if the implementation can be normalized by a constant multiplier. A potential multiplication factor function M is found using Eq. (8.2),

$$M(\mathbf{E}_{y,f,Imp}) = \frac{1}{N}\left(\sum_{i=1}^{N}\frac{|\mathbf{E}_{y,Ana,i}|}{|\mathbf{E}_{y,Imp,i}|}\right) \tag{8.2}$$

The results for $M(\mathbf{E}_{y,f,Imp})$ in Ey component of the E-field at a specific frequency (f) and implementation (Imp), are shown in Fig. 8.4. $\mathbf{E}_{y,Ana,i}$ represents the analytical solution to the Ey field on the dotted line, where Ana represents the analytical solution. $\mathbf{E}_{y,Imp,i}$ represents the solution to the Ey field for the i'th point on the dotted line for a specific implementation.

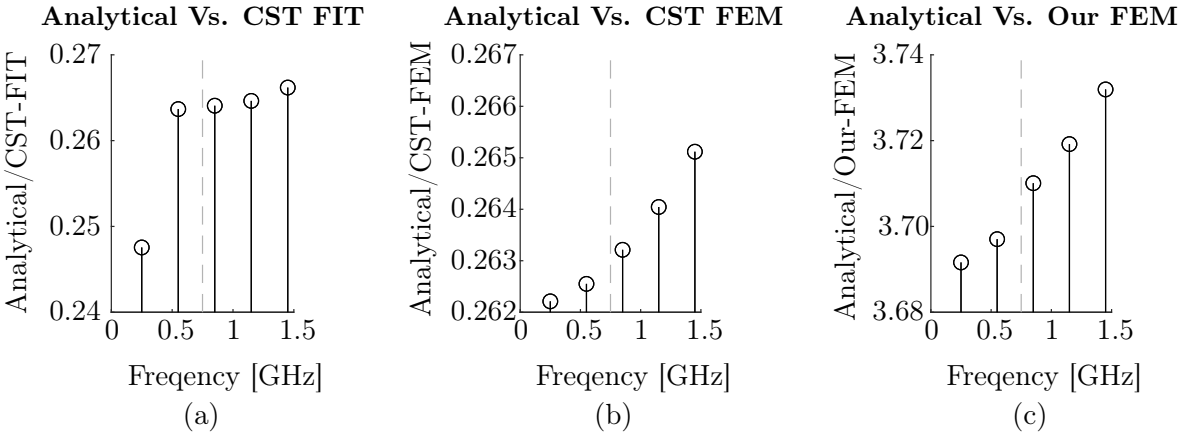


Figure 8.4: The magnitude ratio between the implementation and the analytical solution is derived for each implementation and the frequencies of interest.

The multiplication factor does vary negligibly with the frequency; it does, however, have a noise component. A maximum noise component for the individual implementations is 0.0187 for CST-FIT, 0.0029 for CST-FEM, and 0.040 for Our-FEM. They are obtained by calculating the difference between the largest and the smallest factor attained in our samples. Since the noise is so low, the multiplication factor is considered a constant.

In order to compare the individual implementations, each implementation is normalized by using the corresponding multiplication factor. In this way, it was possible to compare different implementations regardless of the input power; this is shown in Fig. 8.5.

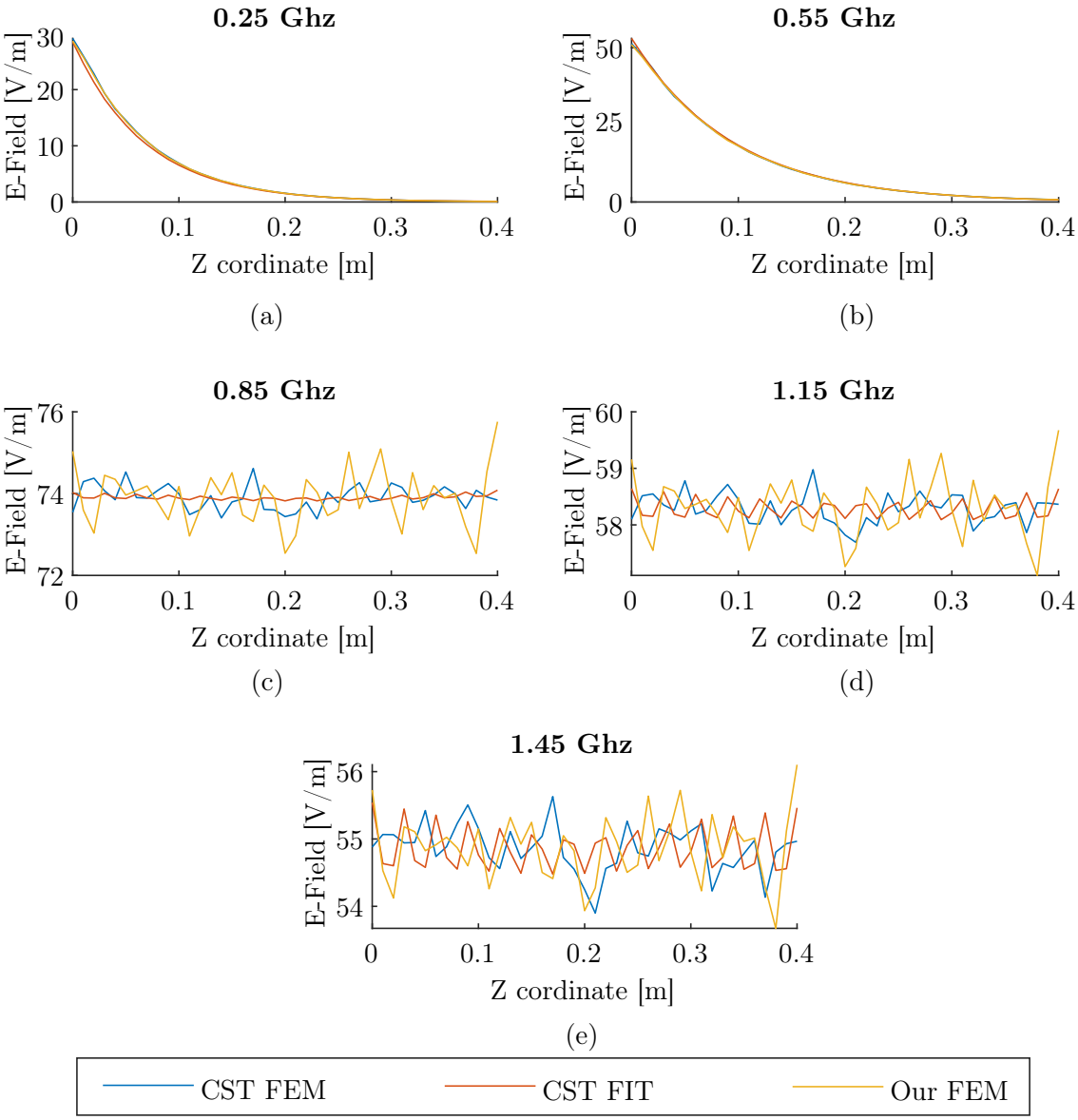


Figure 8.5: Each implementation is normalized using the ratio to the analytical solution. The individual graphs show the normalized magnitude of Ey along the dotted line for specific frequencies.

The E-fields in Fig. 8.5 "a" and "b" are both exponentially decreasing and represent the frequency band below the cut-off frequency. Graph "c", "d" and "e" represent the passband region of the waveguide and illustrates a constant value with a noise component—all implementations when normalized show high correspondence. A variance further investigates the noise component.

The variance of the noise component is calculated by Eq. (8.3),

$$var(\mathbf{E}_{y,imp,f}) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{E}_{y,imp,f,i} - \bar{\mathbf{E}}_{y,imp,f})^2 \tag{8.3}$$

$$\bar{\mathbf{E}}_{y,imp,f} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{E}_{y,imp,f,i} \tag{8.4}$$

The noise variance at the Y component of the E-field for the individual implementations are expressed as $var(\mathbf{E}_{y,imp,f})$ and the mean of the field at Ey as $\bar{\mathbf{E}}_{y,imp,f}$ for a specific frequency at a given implementation with imp representing the individual implementations. The noise variances of the three algorithms in the passband are presented in Table 8.1.

	0.85 GHz	1.15 GHz	1.45 GHz
CST FEM (V^2/m^2)	0.088906	0.072689	0.126604
CST FIT (V^2/m^2)	0.003600	0.029259	0.106693
Our FEM (V^2/m^2)	0.474482	0.292365	0.251274

Table 8.1: Noise variance for the individual implementations at the passband frequencies.

The noise component decreases with frequency for our implementation, whereas the noise increases with frequency for the CST implementations. The noise is low in all implementations; our implementation, however, has noise an order of magnitude higher than the CST implementations. Theoretically, the noise should increase with frequencies as the mesh appears larger for higher frequencies due to the wavelength decreasing with frequency.

The higher noise in the E-field for our FEM implementation could be caused by the method of extracting data points. To attain the data points for the comparison, the E-field at the tetrahedron corners are calculated and interpolated to the specific points along the dotted line in Fig. 8.1. Equation (8.5) for finding the field at a specific point in the tetrahedron should be used instead.

$$\mathbf{E}^e = \sum_{i=1}^{n}\mathbf{N}_i^e E_i^e \tag{8.5}$$

The desired point is evaluated for the basis functions and scaled with the corresponding value in E^e and summed.

A noise component for all numerical implementations does exist, but with a negligible impact on the results. Our solution is designed for linear RF networks, and S-parameters are therefore of interest. For S-parameters, the ratio between the incident and transmitted wave is essential, while the magnitude by itself is not. The noise due to inconsistent sampling is not present in the S-parameters, as the S-parameters utilize the quadrature rule for integration, where the method in Eq. (8.5) is used to find the quadrature points.

8.3 S-Parameters

S-parameters are one of the primary methods used for characterizing linear RF networks. The FEM and FIT simulations attained using CST are compared with the results from our implementation. S_{21} for our implementation is calculated based on Eqs. (7.9) and (7.10),

the magnitudes for the CST implementations and our implementation are illustrated in Fig. 8.6.

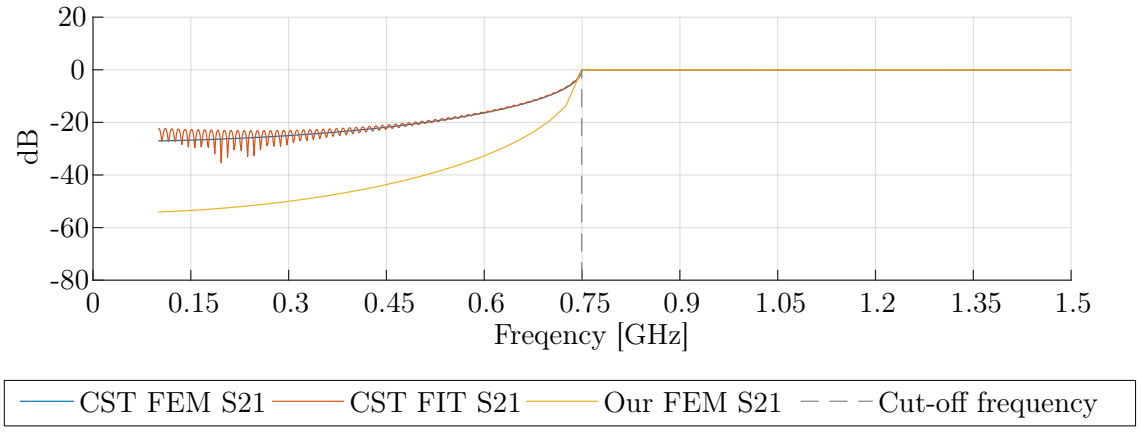


Figure 8.6: S_{21} parameter attained from CST-FEM, CST-FIT and our FEM.

The sampled band is shown, with significant differences below the cut-off frequency. The band of interest, however, is only the passband, separated from the cut-off band (for better visualization) in Fig. 8.7.

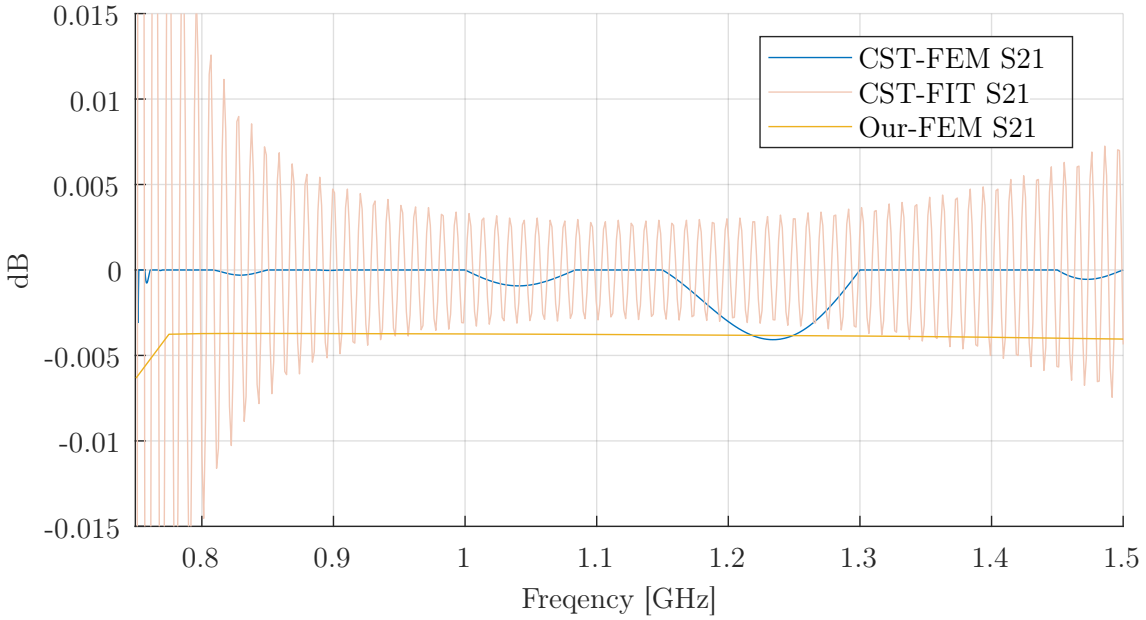


Figure 8.7: S_{21} parameter for the passband, for CST-FEM, CST-FIT, and our FEM. The magnitude of S_{21} is given in (a) while the phase in (b).

The CST-FIT passband shows ripple; this might be due to the FFT of the time domain solution. The ripple could potentially increase for very resonant circuits if the simulation time is not adjusted for the time it takes for all power to be dissipated. Analytically, the passband should be 0 dB; both our implementation of FEM and CST-FEM attain this very well. Our solution, however, has an offset of 0.005 dB; this offset is minimal and has a minor effect.

An important parameter for linear RF networks and, therefore S-parameters, is the phase. Fig. 8.8 shows the phase for the entire sampled frequency band.

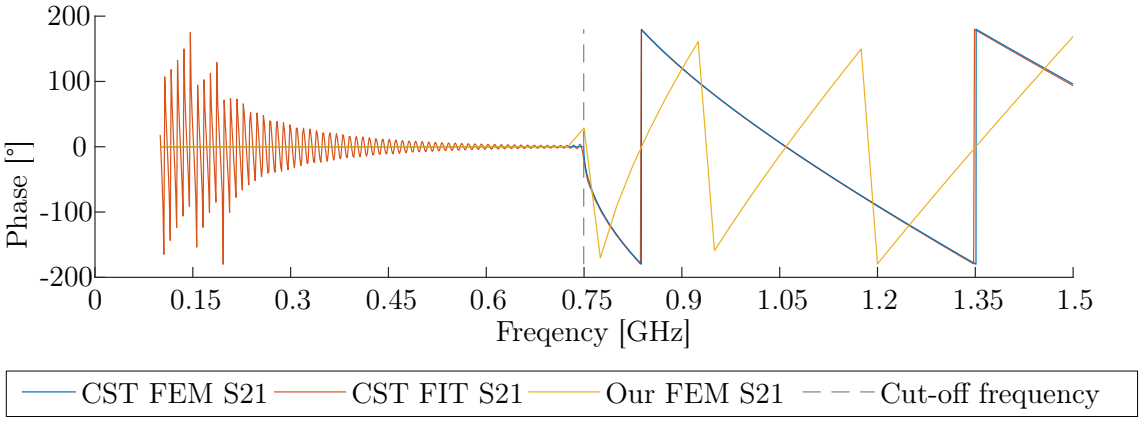


Figure 8.8: The phase of the S_{21} parameter from CST-FEM, CST-FIT, and our FEM.

Below the cut-off frequency, the phase should be 0, and this is achieved. The FIT shows a high ripple in the band of interest. The passband shows a continuous decreasing phase for both CST-FEM and CST-FIT. For our implementation, the phase has a positive slope with a substantially higher phase change. The phase slope is further explored in Fig. 8.9.

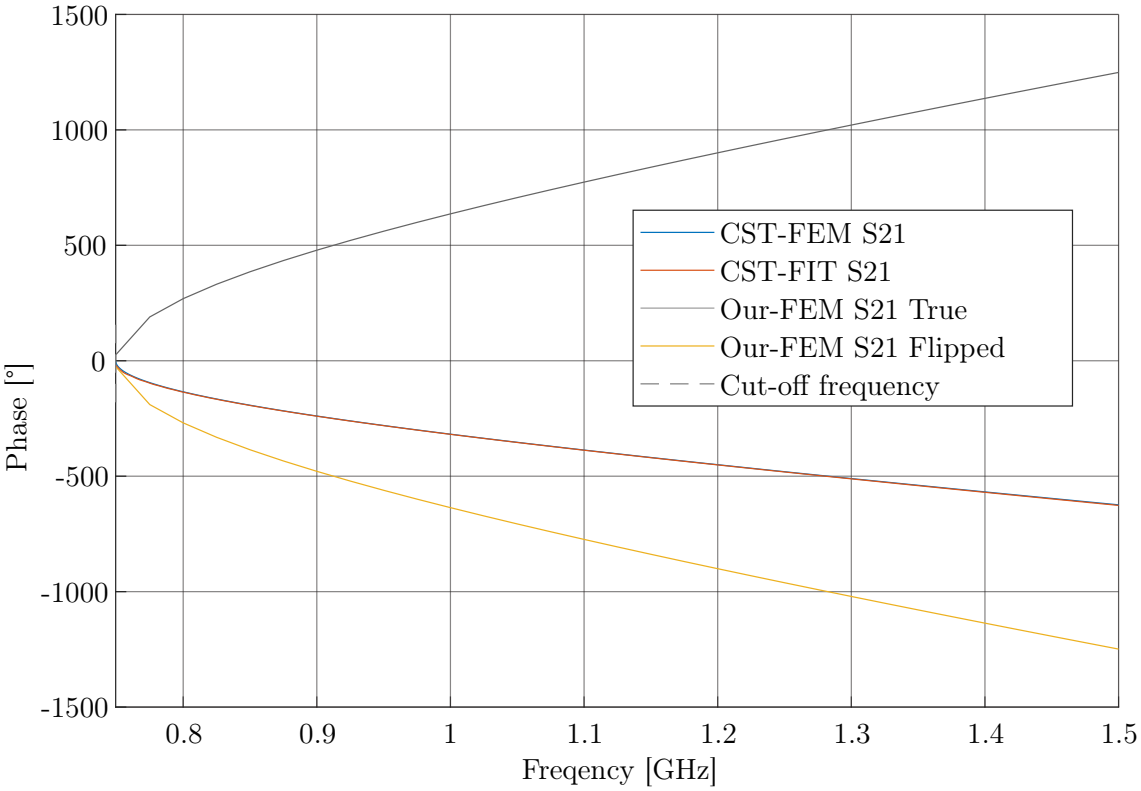


Figure 8.9: S_{21} unwrapped phase from CST-FEM CST-FIT and our FEM.

An unwrapped version of the passband phase is shown in Fig. 8.9. It shows that our implementation has a phase in the opposite direction of the CST implementations, the

phase of our implementation is shown in gray, and a flipped version of the phase is shown in yellow. When comparing the flipped version of our FEM implementation with the CST implementations, it can be noted that our implementation has a slope twice the slope of the CST implementation as well as twice the phase. The phase change and phase velocity are both defined by the propagation constant. The wrong propagation constant may be utilized, or an error in implementing the mathematically derived formulation could have occurred. A wrong propagation constant would explain the lower level of the S_{21} before the cut-of frequency in Fig. 8.6. The exponential decay of the field through the waveguide below the cut-off frequency is determined by the propagation constant.

The S_{11} curves for the two CST implementations and our implementation are illustrated in Fig. 8.10.

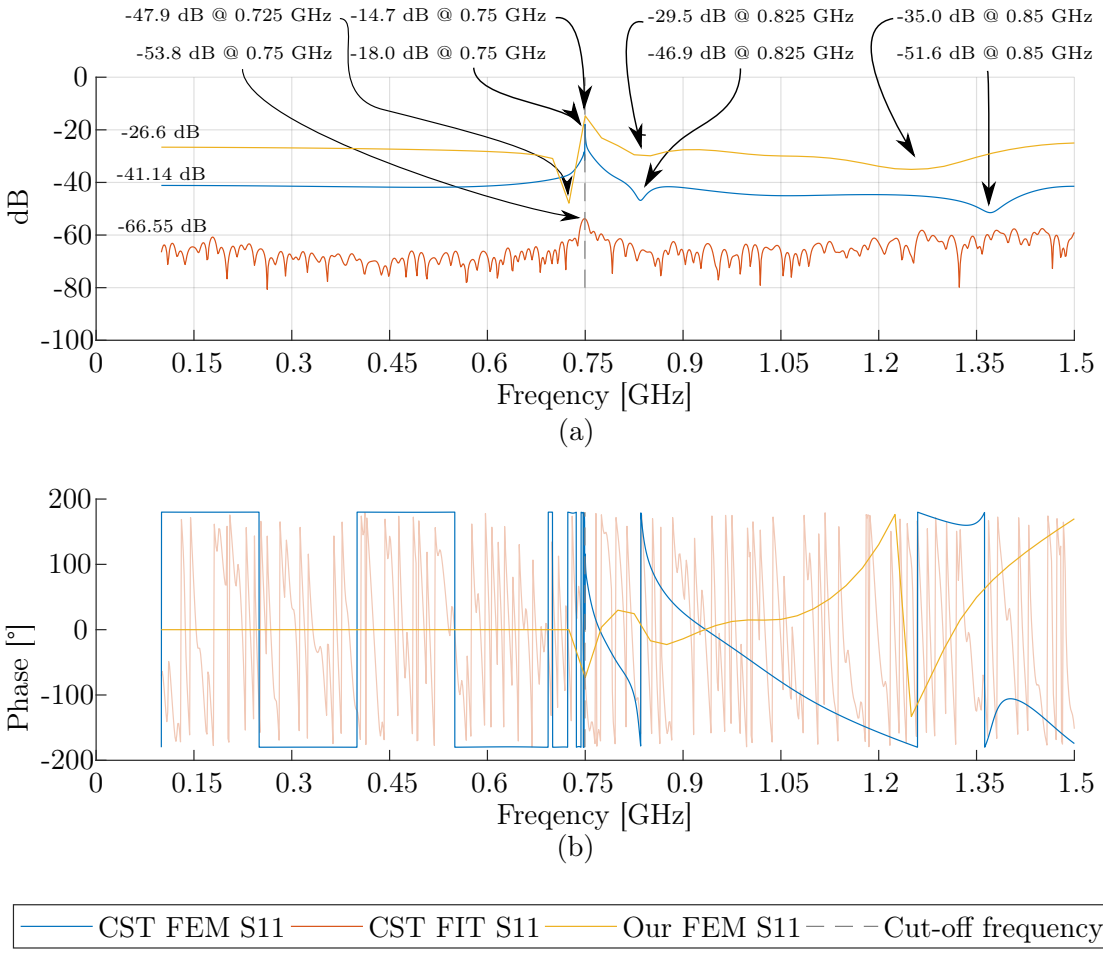


Figure 8.10: S_{11} parameter from CST-FEM CST-FIT and our FEM.

It can be observed that the curves for both implementations have similar characteristics, where they primarily differ is in the reflection levels. Our implementation has lower reflections, with a starting S_{11} of; -26.60 dB for our implementation, -41.14 dB for the CST-FEM implementation and -66.55 dB for the CST-FIT implementation, these values are however all very low. It must be noted that values below -20 dB can be considered noise. The two curves follow the same shape with an offset until the cut-off frequency; past the cut-off frequency, the two curves differ. These differences, or more specifically the

valleys and peaks, are labeled in Fig. 8.10 (a). Below the cut-off frequency, an imaginary impedance is present, and therefore an imaginary voltage is applied. The ports are defined as perfectly matched, therefore S_{11} should be $-\infty$ dB over the whole frequency range.

When observing the phase, it can be noted that the phase difference between the CST-FEM implementation and our implementation is 180°. Past the cut-off frequency, both phases show individual random behavior. This random behavior is due to noise in the E-field. Looking at the phase attained from CST-FIT, it can be noted that it follows a random pattern from below the cut-off to the end of the sampled passband. Therefore it can be concluded that it has a noise behavior.

9 | Conclusion

The purpose of this thesis was to develop and implement a CEM simulator for waveguide filter networks and compare the developed implementation with the commercial implementation in CST Studio by *Dassault Systems*. This thesis analyzed different computational electromagnetic simulation methods for waveguides; one method, namely, FEM, was selected to be implemented. The FEM method was implemented with homogeneous ports in Matlab, utilizing a tetrahedron meshing generated using the open-source Gmsh application programming interface. We chose to develop the system for waveguide-specific structures, in particular filters. A test waveguide with an analytical solution was designed and simulated using our implementation of the FEM algorithm. The result generated was compared to the analytical solution, CST-FEM, and CST-FIT. The purpose of these comparisons was to determine the performance of the developed system compared to commercial solutions and the analytical solution.

The thesis had three main functional objectives; these are presented in a prioritized order below.

1. Three dimensional modeling of electric fields using FEM.
2. Multi-port simulation on homogeneous waveguides.
3. Emulation of scattering parameter measurements.

Based on our observations, we found that at similar meshing densities (46000 tetrahedron), the performance of our implementation is matching in the passband region for S_{21}. It is found that the magnitude has an error of -71.56 dB and -70.26 dB for CST-FEM and CST-FIT, respectively.

For S_{11}, our return loss curve has a starting point at -26.60 dB compared to the CST's FEM -41.14 dB. For S_{11} below the cut-off frequency, differences occur with our implementation and the methods implemented in CST, where our implementation has a starting point at -54.90 dB compared to the CST's FEM -27.24 dB. When looking at the phase between our implementation and the two in CST, our phase has a higher phase propagation speed at two times CST's FEM algorithm.

The phase observed for S_{21} is found to have twice the rate of change and the opposite sign compared to CST-FEM and CST-FIT.

Based on our observations and simulations, it can be concluded that our implementation of the FEM algorithm can be utilized to develop homogeneous waveguide multi-port RF networks with the similar S_{21} and S_{11} curves in the passband region.

10 | Future Work

We consider the thesis to have met the three primary functional objectives; three-dimensional modeling of EM fields using FEM, Multi-port simulation on homogeneous waveguides, and Emulation of scattering parameter measurements. There are, however, areas for improvements if the implemented system should be able to numerically have the same accuracy against commercial implementations such as the FEM simulations in CST studio. These features are analyzed in this chapter.

10.1 Absorbing Boundary Condition

In a realizable structure, the boundaries are not constructed using a PEC material and therefore need some form of open boundary condition. Similarly, when applying the FEM algorithm to radiating structures such as antennas, an open boundary condition is needed. Future work could be done in implementing Absorbing Boundary Conditions to our solution. It needs to be able to truncate an infinite open boundary to a realizable mesh which the computational domain can realize [4]. These conditions should not interfere with the structure to be simulated and appear like an anechoic chamber lowering or ideally removing all reflections appearing from the boundary [4].

10.2 Multi Order Tetrahedron Mesh

Multi-order tetrahedrons should be explored for potentially faster and more accurate simulations. Higher-order elements would reduce the required elements and model complex structures more accurately. Higher-order tetrahedrons would require more complex calculations but would potentially reduce memory for a specific accuracy compared to the single-order equivalent.

10.3 Inhomogeneous Waveguide Ports

The FEM implementation in this thesis is limited to homogeneous waveguide ports and therefore has limited uses. Future development could be conducted in expanding the current implementation to inhomogeneous ports. This would enable the implementation to simulate multiple materials within the port surface, such as PCBs and connectors. The theory and implementation specifics which differ from the homogeneous solution are presented in Appendix A.

10.4 Distributed Computational Implementation

The current implementation is implemented in Matlab and is not optimized for distributed computing. The parallel computing aspect can be implemented in Matlab or Python. Future work could be conducted in porting the current version of the software to a lower-level programming language such as C or Cpp, where individual simulations could be separated into individual computational tasks enabling multithreaded computations of the FEM algorithm, increasing the computational efficiency and speed.

10.5 H-field and Surface Currents

In the current form, the implementation only utilizes the E-field for the S-parameter calculation as it is faster than using both the E-field and H-field. Potential use cases exist for the utilization of the H-field. Future work could be conducted in implementing a post-processing feature to our implementation, which calculates the H-field from the E-field. Engineers often use H-fields plots to study RF structures such as antennas.

A Inhomogeneous Waveguide Ports

Some limitations apply to the 2 port model utilized in Section 5.7. These include the limitation of not being able to place ports directly at waveguide ends due to their approximation to each other [9] and the use of multiple modes. The approach presented in this section will take into account more accurate boundary conditions enabling ports to be placed closer together and a single-mode for the whole waveguide [9]. This method of implementation also improves computational efficiency due to the port placements.

Inhomogeneous waveguides do not support TE and TM modes. A structure-specific mode exists in their place [9]. This mode is called a modal function and can not be calculated analytically [9]. The modal functions need to be estimated numerically using the FEM method in a two-dimensional domain, using basis functions. In this section, the pre-calculated modal functions for the ports are defined as e_m and h_m. e_m is the modal function for the electric field, and h_m is the modal function for the magnetic field. Both of these modal functions must satisfy Eq. (A.1).

$$\iint_S (\mathbf{h}_m \times \mathbf{e}_n) \cdot (-\hat{z}) \mathrm{d}S = \begin{cases} \kappa_m & m = n \\ 0 & m \neq n \end{cases} \tag{A.1}$$

Where the propagation direction in is the negative z direction ($-\hat{z}$) and the constant k_m is a scaling factor. The incident wave in the z direction can be defined similarly to Appendix A as shown in Eq. (A.2).

$$\begin{aligned} \mathbf{E}^{tot} &= \mathbf{E}^{\mathrm{inc}} + \mathbf{E}^{\mathrm{ref}} \\ &= \mathbf{E}^{\mathrm{inc}} + \sum_{m=1}^{\infty} a_m \mathbf{e}_m \mathrm{e}^{\gamma_m z} \\ a_m &= -\frac{\mathrm{e}^{-\gamma_m z}}{\kappa_m} \int_S \left[\mathbf{h}_m \times \left(\mathbf{E}_{tot} - \mathbf{E}^{\mathrm{inc}}\right)\right] \cdot \hat{z} \, \mathrm{d}S \end{aligned} \tag{A.2}$$

Variable	Description
$\mathbf{E}^{tot}$	Total electric field at port surface
$\mathbf{E}^{inc}$	Inclining E-field, applied charge to port surface
$\mathbf{E}^{ref}$	Reflected E-field
e_m	E-field modal function
h_m	H-field modal function
γ_m	Phase constant
k_m	Amplitude constant

The variable "a_m" in Eq. (A.2) can be further simplified with the previously calculated function in Eq. (A.3).

$$\begin{aligned}\hat{z} \times \mathbf{h}_m \mathrm{e}^{\gamma_m z} &= -\frac{1}{\mathrm{j}\omega\mu}\hat{z} \times [\nabla \times (\mathbf{e}_m \mathrm{e}^{\gamma_m z})] \\ &= \frac{1}{\mathrm{j}\omega\mu}\left(\gamma_m \mathbf{e}_{\mathrm{t}m} - \nabla_{\mathrm{t}} e_{zm}\right)\mathrm{e}^{-\gamma_m z}\end{aligned} \tag{A.3}$$

By inserting Eq. (A.3) into a_m from Eq. (A.2), a simplified version of a_m can be found as shown in Eq. (A.4).

$$a_m = -\frac{\mathrm{e}^{-\gamma_m z}}{\mathrm{j}\omega\mu\kappa_m}\iint_S \left(\gamma_m \mathbf{e}_{\mathrm{t}m} - \nabla_{\mathrm{t}} e_{zm}\right) \cdot \left(\mathbf{E} - \mathbf{E}^{\mathrm{inc}}\right) \mathrm{d}S \tag{A.4}$$

The curl of Eq. (A.2) is presented in Eq. (A.5).

$$(\nabla \times \mathbf{E}_{tot}) = (\nabla \times \mathbf{E}^{\mathrm{inc}}) + \sum_{m=1}^{\infty} a_m [\nabla \times \mathbf{e}_m (\mathrm{e}^{\gamma_m z})] \tag{A.5}$$

The cross product of Eq. (A.5) and the normal $\hat{n}$ results in Eq. (A.6).

$$\hat{n} \times (\nabla \times \mathbf{E}_{tot}) = \hat{n} \times (\nabla \times \mathbf{E}^{\mathrm{inc}}) + \sum_{m=1}^{\infty} a_m \hat{n} \times [\nabla \times \mathbf{e}_m (\mathrm{e}^{\gamma_m z})] \tag{A.6}$$

From previous calculation Eq. (A.7) is known.

$$\hat{z} \times [\nabla \times (\mathbf{e}_m \mathrm{e}^{\gamma_m z})] = \left(-\gamma_m \mathbf{e}_{tm} + \nabla_{\mathrm{t}} e_{zm}\right)\mathrm{e}^{\gamma_m z} \tag{A.7}$$

By inserting $\hat{n} = -\hat{z}$ for the last $\hat{n}$ in Eq. (A.6), Eq. (A.8) is formed.

$$\hat{n} \times (\nabla \times \mathbf{E}_{tot}) = \hat{n} \times \left(\nabla \times \mathbf{E}^{\mathrm{inc}}\right) - \sum_{m=1}^{\infty} a_m \hat{z} \times [\nabla \times (\mathbf{e}_m \mathrm{e}^{\gamma_m z})] \tag{A.8}$$

Eq. (A.8) can be further simplified by inserting Eq. (A.7) as shown in Eq. (A.9).

$$\hat{n} \times (\nabla \times \mathbf{E}) = \hat{n} \times \left(\nabla \times \mathbf{E}^{\mathrm{inc}}\right) + \sum_{m=1}^{\infty} a_m \left(\gamma_m \mathbf{e}_{tm} - \nabla_{\mathrm{t}} e_{zm}\right)\mathrm{e}^{\gamma_m z} \tag{A.9}$$

where,

Variable	Description
e_{zm}	$\hat{z}$ part of the E-field modal function
$\mathbf{e}_{tm}$	Transverse part of the E-field modal function
∇_t	The gradient of the transverse part

The final port boundary condition for the waveguide can be derived by inserting the simplified version of a_m, as derived in Eq. (A.4), into Eq. (A.9) as shown in Eq. (A.10).

$$\mathbf{U}^{\mathrm{inc}} = \hat{n} \times (\nabla \times \mathbf{E}) + P(\mathbf{E}) \tag{A.10}$$

Where the remaining variables are defined as Eq. (A.11) and Eq. (A.12).

$$P(\mathbf{E}) = \sum_{m=1}^{\infty} \frac{1}{\mathrm{j}\omega\mu\kappa_m} (\gamma_m \mathbf{e}_{tm} - \nabla_\mathrm{t} e_{zm}) \iint_S (\gamma_m \mathbf{e}_{tm} - \nabla_\mathrm{t} e_{zm}) \cdot \mathbf{E} \mathrm{d}s \tag{A.11}$$

$$\begin{aligned} &\mathbf{U}^{\mathrm{inc}} = \hat{n} \times \left(\nabla \times \mathbf{E}^{\mathrm{inc}}\right) \\ &+ \textstyle\sum_{m=1}^{\infty} \frac{1}{\mathrm{j}\omega\mu\kappa_m} (\gamma_m \mathbf{e}_{\mathrm{tm}} - \nabla_\mathrm{t} e_{zm}) \iint_S (\gamma_m \mathbf{e}_{tm} - \nabla_\mathrm{t} e_{zm}) \cdot \mathbf{E}^{\mathrm{inc}} \mathrm{d}S \end{aligned} \tag{A.12}$$

Where,

$$\kappa_m = \frac{\mathrm{J}}{\omega\mu} \iint_S (\gamma_m \mathbf{e}_{\mathrm{tm}} \cdot \mathbf{e}_{\mathrm{tm}} - \mathbf{e}_{tm} \cdot \nabla_\mathrm{t} e_{zm}) \, \mathrm{d}S \tag{A.13}$$

Similar to the previously described method, this method only utilizes the E-field calculations. No magnetic field components need to be calculated. The modal functions e_m are dependent on frequency and need to be calculated accordingly.

Bibliography

[1] *IMT traffic estimates for the years 2020 to 2030*. Report. Geneva, CH: International Telecommunication Union, July 2015.

[2] Petar Popovski. *Wireless Connectivity: An Intuitive and Fundamental Guide*. Wiley, 2020.

[3] Anders Bondeson, Thomas Rylander, and Pär Ingelström. *Computational Electromagnetics*. Chalmers University of Technology, Hörsalsvägen 11 Göteborg, Sweden: Springer, 2005.

[4] Jian-Ming Jin. *THEORY AND COMPUTATION OF ELECTROMAGNETIC FIELDS*. Second Edition. John Wiley Sons, Inc, 2015. ISBN: 978-1-119-10804-7.

[5] David M. Pozar. *Microwave engineering*. fourth Edition. Wiley, 2012. ISBN: 978-0-470-63155-3.

[6] Fredrik Bengzon Mats G. Larson. *The Finite Element Method: Theory, Implementation, and Applications*. Springer, 2013. ISBN: 978-3-642-33286-9.

[7] everythingrf.com. *Waveguide Sizes*. URL: `https://www.everythingrf.com/tech-resources/waveguides-sizes`. (accessed: 20.05.2021).

[8] Dassault Systems. *ELECTROMAGNETIC SIMULATION SOLVERS*. URL: `https://www.3ds.com/products-services/simulia/products/cst-studio-suite/solvers/`. (accessed: 01.03.2020).

[9] Jian-Ming Jin. *The Finite Element Method in Electromagnetics, 3rd Edition*. John Wiley Sons, Inc, May 2014.

[10] Keysight. *State of the Art in EM Software for Microwave Engineers*. URL: `https://www.keysight.com/dk/en/assets/7018-01998/white-papers/5990-3225.pdf`. (accessed: 01.03.2020).

[11] Ansys. *3D Electromagnetic Field Simulator for RF and Wireless Design*. URL: `https://www.ansys.com/products/electronics/ansys-hfss`. (accessed: 01.03.2020).

[12] openEMS. *OpenEMS a free and open electromagnetic field solver*. URL: `https://openems.de`. (accessed: 01.03.2020).

[13] Gopinath Gampala and C. J. Reddy. "Comparison of various CEM methods for antenna array applications". In: *2015 31st International Review of Progress in Applied Computational Electromagnetics (ACES)*. 2015, pp. 1–2.

[14] *A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. URL: `https://gmsh.info/`. (accessed: 20.05.2021).