Metamaterial- and Machine Learning-Assisted THz Sensing

Master Thesis

Group 5.325

Aalborg University
Department of Materials and Production

Department of Materials and Production
Aalborg University
http://www.mp.aau.dk/

AALBORG UNIVERSITY
DENMARK

Title:
Metamaterial- and Machine Learning-Assisted THz Sensing

Theme:
Applied Nanotechnology

Project Period:
Spring Semester 2022
10th Semester

Project Group:
Group 5.325

Participants:
Martin Borregaard Kristensen
Martin Maarslet Richter
Thomas Lindekilde Sørensen

Supervisors:
Esben Skovsen
Thomas Møller Søndergaard

ECTS Points: 30

Number of Pages: 95

Date of Completion:
2nd June, 2022

Abstract:

The aim of this project is to examine the capabilities of metamaterials for the use of field enhancement to complement THz time-domain spectroscopy of organic compounds. A finite-difference time-domain model was developed and used to model metamaterials, mainly on the form of periodic arrays of gold antennas on silicon wafers, but also split-ring resonators. The arrays of antennas were made for multiple lengths and array dimensions. With the dimensions obtained from the model, samples were created using UV direct write lithography and magnetron sputtering. Glucose and sucrose were deposited on metamaterial samples, and then measured in a THz time-domain spectrometer. Machine learning methods such as feature selection, feature transformation, and classification was adapted and used to classify measurements from different samples throughout the project. Different types of noise parameters were compared, and finally machine learning was used to test the field enhancement properties of the metamaterial samples.

Contents

Preface

This master thesis is written by group 5.325, a 10th semester Nanotechnology project group at Aalborg University, supervised by Associate Professors, Esben Skovsen and Thomas Møller Søndergaard. The project lasted from the 31st of January 2022 to the 2nd of June 2022.

The purpose of this project is to describe, model, and produce metamaterials, such as arrays of antennas. After production the metamaterial samples will be measured under different conditions, the measurements of which will be used for machine learning purposes. To calculate the ideal dimensions of the metamaterials, algorithms found from literature were implemented based on theory. The following chapters cover the used method, results, and discussion in relation to the theory and, lastly, a conclusion to the project. In this project the software MATLAB was used for the numerical calculations.

All sections, tables, and significant equations are numbered for reference. This report uses a numerical reference system, and a reference is represented as [*number*]. Referenced sources are in the bibliography. References before periods only refer to the associated sentence. References after periods are associated with the entire paragraph. The master thesis is written with oxford comma.

Aalborg University, 2nd June, 2022

Martin Borregaard Kristensen
makris16@student.aau.dk

Martin Maarslet Richter
mricht17@student.aau.dk

Thomas Lindekilde Sørensen
tlsa17@student.aau.dk

1 Introduction

Terahertz time-domain spectroscopy has shown good capabilities as a sensing tool, as it is non-ionising, non-label, and non-contact. An avenue of interest in supplementing the method is the use of metamaterials to create field enhancement, which increases the interaction of the samples with terahertz radiation, and thereby improving the sensitivity of the method. The metamaterials used for enhancement in this project are gold antennas on a silicon wafer. For this project, the compounds examined are glucose and sucrose with absorption peaks at around 1.4 THz, with glucose having the significantly largest peak. [1]

Metamaterials are artificial structures designed to obtain properties that are distinct from the properties of natural materials, for instance novel electromagnetic parameters. The difference allows the utilisation of compact super lenses, antennas, filters, field enhancement, and frequency-agile metamaterials with the functionality of permitting the real time adjustment of the electromagnetic response. [2, 3]

Machine learning, neural network, artificial intelligence, many names can be used to describe the concept of pushing a machine to complete tasks that otherwise only an intelligent living being could accomplish. In essence, a computer can be trained with large amounts of data and complicated statistical models to 'learn'. Common examples are image recognition machine learning, such as telling whether an image contains a dog or a cat, or to distinguish the road from the pavement. [4]

While image recognition is exciting and has numerous applications, it is not the only machine learning technique. Other types of data can also be used, such as text or spectra. Humans excel at image recognition and at understanding text, but spectra can be difficult. As such, machine learning research focused on spectra recognition is of interest, as the advanced algorithms can already exceed human capabilities. [5, 6]

The first use of the term machine learning was by Arthur Samuel in 1959 [7]. He found that a computer could be programmed to play the board game checkers better than the human who wrote the program could do [7, 8]. Since then, several signification breakthroughs have been made, and machine learning shifted from a knowledge based approach to a data based approach, as computers become more powerful and able to better handle large amounts of data. Today, machine learning can be applied with little knowledge about the data that is used. [9, 10, 11]

The focus of this project is to examine the viability of using machine learning and metamaterial field enhancement to improve on the sensitivity of terahertz time-domain spectroscopy, where a theoretical model will be established to determine the construction of the metamaterials.

2 | Numerical Model

This chapter will describe the fundamental theory and modelling process behind the finite-difference time-domain (FDTD) method in three spatial dimensions. First, a summary of Maxwell's equations is given. The standard FDTD method is then described, and the different modifications to the standard method used in this project is thereafter described.

2.1 Maxwell's Equations

This section will briefly outline and describe Maxwell's equations and the field continuity conditions, which are imperative for the understanding of the electromagnetic theory used in this project. Readers already familiar with these may inconsequentially skip it.

Maxwell's equations in their differential forms are given by

$$\boldsymbol{\nabla} \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{2.1}$$

$$\boldsymbol{\nabla} \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}, \tag{2.2}$$

$$\boldsymbol{\nabla} \cdot \mathbf{D} = \rho, \tag{2.3}$$

$$\boldsymbol{\nabla} \cdot \mathbf{B} = 0, \tag{2.4}$$

where $\mathbf{E}$, $\mathbf{D}$, $\mathbf{H}$ and $\mathbf{B}$ are the electric, displacement, magnetic, and magnetic induction fields, respectively. $\mathbf{J}$ is the current density. They are related by the constitutive relations, which describe the macroscopic properties of a material, given by

$$\mathbf{D} = \epsilon \mathbf{E}, \tag{2.5}$$

$$\mathbf{B} = \mu \mathbf{H}, \tag{2.6}$$

$$\mathbf{J} = \sigma \mathbf{E}, \tag{2.7}$$

where ϵ, μ, and σ are the permittivity, permeability, and conduction of the material, respectively. In the time-harmonic case, these are constants for isotropic, homogeneous materials, scalar functions of position for isotropic inhomogeneous materials, and tensors for anisotropic materials. In the case of time-dependent fields, these parameters are time dependent, as well.

At the interface between two different materials, the fields are described by the so-called field

continuity conditions given by

$$\hat{\mathbf{n}} \times (\mathbf{E}_1 - \mathbf{E}_2) = 0, \tag{2.8}$$
$$\hat{\mathbf{n}} \cdot (\epsilon_1 \mathbf{E}_1 - \epsilon_2 \mathbf{E}_2) = \rho_s, \tag{2.9}$$
$$\hat{\mathbf{n}} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_s, \tag{2.10}$$
$$\hat{\mathbf{n}} \cdot (\mu_1 \mathbf{H}_1 - \mu_2 \mathbf{H}_2) = 0, \tag{2.11}$$

where $\hat{\mathbf{n}}$ is normal unit vector of the interface pointing from medium 1 to medium 2, ρ_s is the surface charge density at the interface, and $\mathbf{J}_s$ is the surface current density at the interface. For a source free interface, both the surface charge and current densities are zero. These conditions essentially describes that the electric field components tangential to the interface must be continuous over an interface, whereas the components normal to an interface may be discontinuous over the interface. Likewise, the magnetic field components tangential to a source free interface may be continuous over the interface, and the components normal to an interface may be discontinuous over the interface. However, for an interface between dielectric materials, the components of the magnetic field normal to the interface must also be continuous over the interface.

2.2 Yee-Algorithm

The essence of the FDTD method is a discrete field updating scheme, i.e. each field component is discretised in both time and space, and are updated using discrete formulations of Faraday's law and Ampère's law. These laws are the two curl-Maxwell's equations, eqs. (2.1) and (2.2). Using the constitutive relations, eq. (2.5), Faraday's and Ampére's laws may be expressed as

$$\boldsymbol{\nabla} \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t}, \tag{2.12}$$

and

$$\boldsymbol{\nabla} \times \mathbf{H} = \mathbf{J} + \epsilon \frac{\partial \mathbf{E}}{\partial t}, \tag{2.13}$$

respectively. More explicitly, these are given as

$$\hat{\mathbf{x}} \left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) + \hat{\mathbf{y}} \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) + \hat{\mathbf{z}} \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial z} \right) = -\hat{\mathbf{x}} \mu \frac{\partial H_x}{\partial t} - \hat{\mathbf{y}} \mu \frac{\partial H_y}{\partial t} - \hat{\mathbf{z}} \mu \frac{\partial H_z}{\partial t}, \tag{2.14}$$

and

$$\hat{\mathbf{x}} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) + \hat{\mathbf{y}} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) + \hat{\mathbf{z}} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial z} \right) = \hat{\mathbf{x}} \epsilon \frac{\partial E_x}{\partial t} + \hat{\mathbf{y}} \epsilon \frac{\partial E_y}{\partial t} + \hat{\mathbf{z}} \epsilon \frac{\partial E_z}{\partial t}. \tag{2.15}$$

From these, the temporal evolution of each component of the electric and magnetic fields can easily be expressed through the spatial evolution of the other components of the magnetic and

electric fields, respectively. As such

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_y}{\partial z} - \frac{\partial H_z}{\partial y}\right), \tag{2.16}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_z}{\partial x} - \frac{\partial H_x}{\partial z}\right), \tag{2.17}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x}\right), \tag{2.18}$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z}\right), \tag{2.19}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x}\right), \tag{2.20}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\right). \tag{2.21}$$

The derivatives are numerically approximated by so-called forward, backward, or central difference schemes. In the forward difference scheme, the derivative at a sampling point is approximated as the slope of the secant that intersects the sampling point as well as the next sampling point, and in the backward difference scheme, it is approximated as the slope of the secant that intersects the sampling point as well as the previous sampling point. In the central difference scheme, the derivative at a sampling point is approximated as the slope of the secant that intersects the previous and next sampling points. It can easily be shown that the central difference scheme has a significantly higher accuracy with a comparable computational complexity relative to the forward and backward difference schemes. As such, the central difference scheme is generally considered superior, and it is therefore preferred in the FDTD method.

In the central difference scheme, the first derivative of some function, $f(x)$, is approximated as

$$\left.\frac{\partial f(x)}{\partial x}\right|_{x=x_i} \approx \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} = \frac{\delta f(x_i)}{\delta x_i}. \tag{2.22}$$

However, when calculating the discrete temporal evolution of each field component, the spatial derivatives needed are those of the other components of the other field. As such, all field components may be spatially shifted in their discretisation relative to each other, i.e. the electric field is sampled between sample points of the magnetic field. To this end, consider a computational domain that is discretised into a $(N_x \times N_y \times N_z)$-mesh. This mesh has a total of N_x, N_y, and N_z mesh points along the x-, y-, and z-directions, respectively. Instead of sampling the electric and magnetic fields at these mesh points, let the electric field components be sampled between the mesh points in the directions of the field components, and let the magnetic field be sampled between the mesh points in the tangential directions to the field components. That is, at the (i, j, k)'th mesh point, the fields are sampled at

$$E_x|^{n+1/2}_{i+1/2,j,k} = E_x(x_{i+1/2}, y_j, z_k, t_{n+1/2}), \tag{2.23}$$

$$E_y|^{n+1/2}_{i,j+1/2,k} = E_y(x_i, y_{j+1/2}, z_k, t_{n+1/2}), \tag{2.24}$$

$$E_z|^{n+1/2}_{i,j,k+1/2} = E_z(x_i, y_j, z_{k+1/2}, t_{n+1/2}), \tag{2.25}$$

$$H_x|^n_{i,j+1/2,k+1/2} = H_x(x_i, y_{j+1/2}, z_{k+1/2}, t_n), \tag{2.26}$$
$$H_y|^n_{i+1/2,j,k+1/2} = H_y(x_{i+1/2}, y_j, z_{k+1/2}, t_n), \tag{2.27}$$
$$H_z|^n_{i+1/2,j+1/2,k} = H_z(x_{i+1/2}, y_{j+1/2}, z_k, t_n). \tag{2.28}$$

These are illustrated in fig. 2.1. Such a staggered grid is called a Yee-mesh; named after the creator of the FDTD method. It is thus clear that field components associated by eqs. (2.16) to (2.21) lies directly adjacent to each other. As such, with the central difference scheme, eq. (2.22), the partial time derivative of e.g. the x-component of the magnetic field is approximated through the discrete derivatives of the y- and z-components of the electric fields as

$$\left.\frac{\partial E_x}{\partial t}\right|_{i+1/2,j,k} \approx \frac{1}{\epsilon|_{i+1/2,j,k}} \left(\frac{H_y|_{i+1/2,j,k+1/2} - H_y|_{i+1/2,j,k-1/2}}{\delta z_k} - \frac{H_z|_{i+1/2,j+1/2,k} - H_z|_{i+1/2,j-1/2,k}}{\delta y_j} \right). \tag{2.29}$$

Furthermore, the partial time derivative may itself be approximated by the central difference scheme, such that

$$\left.\frac{\partial E_x}{\partial t}\right|^n_{i,j+1/2,k+1/2} \approx \frac{E_x|^{n+1/2}_{i+1/2,j,k} - E_x|^{n-1/2}_{i+1/2,j,k}}{\delta t_n}. \tag{2.30}$$

Note that, so far, the temporal and spatial discretisations have been considered variable throughout the computational domain. While the temporal and spatial discretisation may indeed be time and position dependent, but not interdependent, the Yee-mesh has an inherent stability criteria that not only makes it highly impractical to implement variable discretisations, but also decreases the accuracy of the method. This stability criteria shall be explained in further detail later. Therefore, only constant temporal and spatial discretisations are generally used in the FDTD method. Applying the central difference schemes to the spatial derivatives and temporal derivatives, as in eqs. (2.29) and (2.30), respectively, on eqs. (2.16) to (2.21), results in the update equations as

$$E_x|^{n+1/2}_{i+1/2,j,k} = E_x|^{n-1/2}_{i+1/2,j,k} + \frac{\delta t}{\epsilon|_{i+1/2,j,k}} \left(\frac{H_z|^n_{i+1/2,j+1/2,k} - H_z|^n_{i+1/2,j-1/2,k}}{\delta y} - \frac{H_y|^n_{i+1/2,j,k+1/2} - H_y|^n_{i+1/2,j,k-1/2}}{\delta z} \right), \tag{2.31}$$

$$E_y|^{n+1/2}_{i,j+1/2,k} = E_y|^{n-1/2}_{i,j+1/2,k} + \frac{\delta t}{\epsilon|_{i,j+1/2,k}} \left(\frac{H_x|^n_{i,j+1/2,k+1/2} - H_x|^n_{i,j+1/2,k-1/2}}{\delta z} - \frac{H_z|^n_{i+1/2,j+1/2,k} - H_z|^n_{i-1/2,j+1/2,k}}{\delta x} \right), \tag{2.32}$$

$$E_z|^{n+1/2}_{i,j,k+1/2} = E_z|^{n-1/2}_{i,j,k+1/2} + \frac{\delta t}{\epsilon|_{i,j,k+1/2}} \left(\frac{H_y|^n_{i+1/2,j,k+1/2} - H_y|^n_{i-1/2,j,k+1/2}}{\delta x} - \frac{H_x|^n_{i,j+1/2,k+1/2} - H_x|^n_{i,j-1/2,k+1/2}}{\delta y} \right), \tag{2.33}$$

$$H_x|^{n+1}_{i,j+1/2,k+1/2} = H_x|^{n}_{i,j+1/2,k+1/2} + \frac{\delta t}{\mu|_{i,j+1/2,k+1/2}} \left(\frac{E_y|^{n+1/2}_{i,j+1/2,k+1} - E_y|^{n+1/2}_{i,j+1/2,k}}{\delta z} - \frac{E_z|^{n+1/2}_{i,j+1,k+1/2} - E_z|^{n+1/2}_{i,j,k+1/2}}{\delta y} \right), \tag{2.34}$$

$$H_y|^{n+1}_{i+1/2,j,k+1/2} = H_y|^{n}_{i+1/2,j,k+1/2} + \frac{\delta t}{\mu|_{i+1/2,j,k+1/2}} \left(\frac{E_z|^{n+1/2}_{i+1,j,k+1/2} - E_z|^{n+1/2}_{i,j,k+1/2}}{\delta x} - \frac{E_x|^{n+1/2}_{i+1/2,j,k+1} - E_x|^{n+1/2}_{i+1/2,j,k}}{\delta z} \right), \tag{2.35}$$

$$H_z|^{n+1}_{i+1/2,j+1/2,k} = H_z|^{n}_{i+1/2,j+1/2,k} + \frac{\delta t}{\mu|_{i+1/2,j+1/2,k}} \left(\frac{E_x|^{n+1/2}_{i+1/2,j+1,k} - E_x|^{n+1/2}_{i+1/2,j,k}}{\delta y} - \frac{E_y|^{n+1/2}_{i+1,j+1/2,k} - E_y|^{n+1/2}_{i,j+1/2,k}}{\delta x} \right). \tag{2.36}$$

Note that these are approximations, even though they are presented as equations. These update equations are the workhorses of the FDTD method. The principle of the Yee-algorithm is to take one step in time and subsequently update all field components using eqs. (2.31) to (2.36) iteratively throughout the entire Yee-mesh.

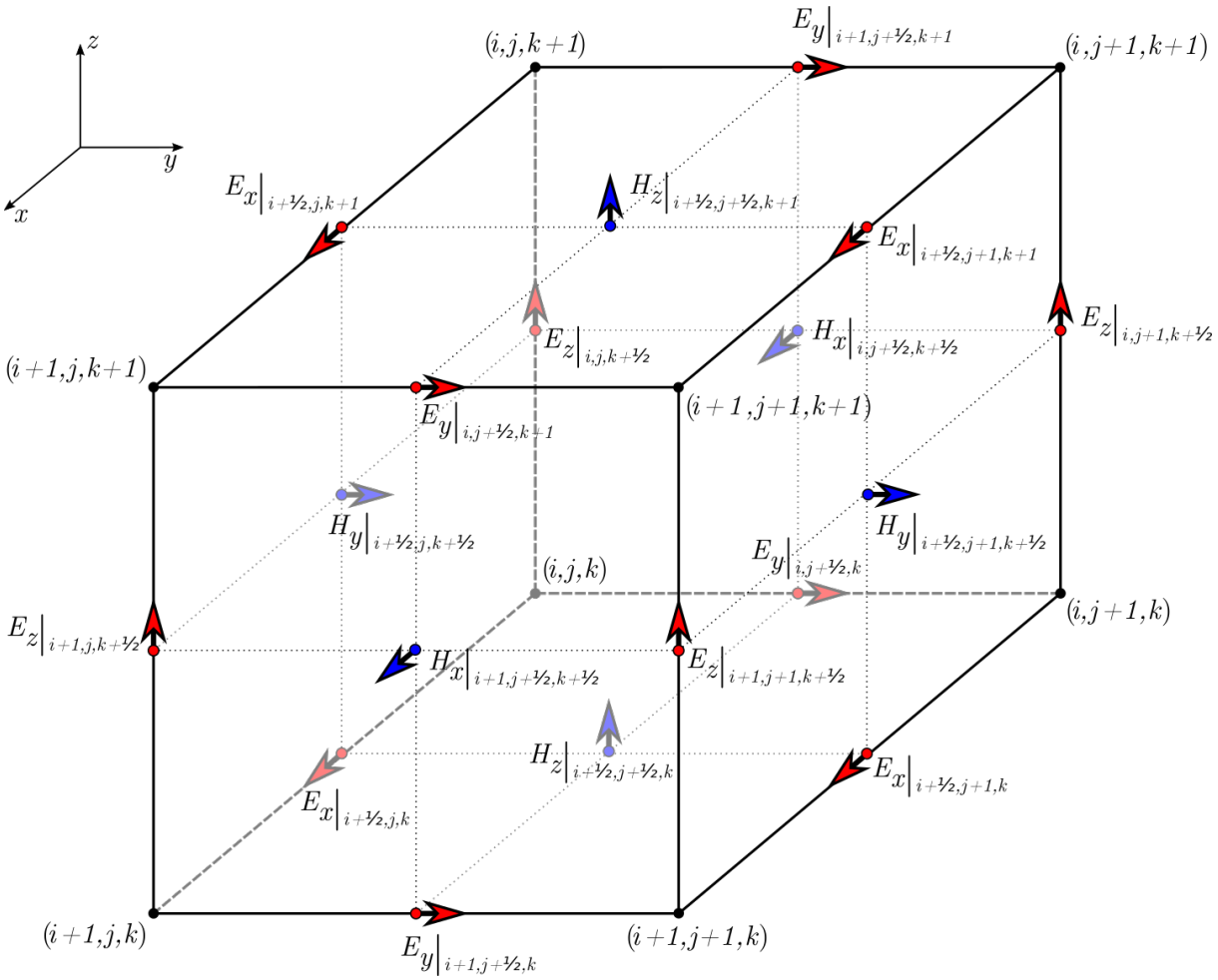


Figure 2.1: Illustration of a general three-dimensional Yee-mesh cell.

2.2.1 Numerical Dispersion and the Stability Criterion

As mentioned above, the grid-spacing and time-stepping are not inherently independent, and the dependence between these will be explored in this section. The section is based on chapter three of the book, *Numerical Methods in Photonics* by Andrei V. Lavrinenko et al. [12].

Firstly, consider the general solution to Maxwell's equations on the form of a monochromatic propagating wave as

$$\mathbf{E}(\mathbf{r},t) = \mathbf{E}_0 e^{i\mathbf{k}\cdot\mathbf{r}} e^{-i\omega t}. \tag{2.37}$$

Such a wave has a dispersion relation given as

$$\omega^2 = \mathbf{k}^2 c_0, \tag{2.38}$$

where c_0 is the free-space speed of light. In discrete space and time, a corresponding solution to Maxwell's equations will be on the form

$$\mathbf{E}(a,b,c,n) = \mathbf{E}_0 e^{ik_x a\delta x} e^{ik_y b\delta y} e^{ik_z c\delta z} e^{-i\omega n\delta t}, \tag{2.39}$$

where a, b, c, and n are integers. A similar solution exists for the magnetic field. By applying the central difference scheme, eq. (2.22), to eq. (2.39) and the corresponding equation for the magnetic field, a homogeneous system of six scalar, linear equations are obtained. Solving this will give

$$\frac{1}{c_0^2\delta t^2}\sin^2\left(\frac{\omega\delta t}{2}\right) = \frac{1}{\delta x^2}\sin^2\left(\frac{k_x\delta x}{2}\right) + \frac{1}{\delta y^2}\sin^2\left(\frac{k_y\delta y}{2}\right) + \frac{1}{\delta z^2}\sin^2\left(\frac{k_z\delta z}{2}\right), \tag{2.40}$$

which describes numerical dispersion. While eq. (2.38) predicts dispersion caused by material parameters, eq. (2.40) also predicts dispersion caused by the discretisation of space and time. Material and anisotropic dispersion, i.e. phase velocity dependence on material and direction, respectively, are physically real phenomena, and these are indeed described by the numerical dispersion, eq. (2.40). However, it is also noticed that a non-physical dispersion arises from this equation due to the lattice parameters. This kind of dispersion is a numerical artefact, and as these are non-physical phenomena, one would aim to minimise its effects in a simulation. Furthermore, the time-step is of great interest to optimise, as time-steps too small results in insurmountable computation times, and time-steps too large would break the causality between the electric and magnetic fields. As such, an analysis on the stability to estimate restraints for a reasonable time-step range is of interest.

First, a new variable, ξ, is introduced as

$$\xi = c_0\delta t\sqrt{\frac{1}{\delta x^2}\sin^2\left(\frac{k_x\delta x}{2}\right) + \frac{1}{\delta y^2}\sin^2\left(\frac{k_y\delta y}{2}\right) + \frac{1}{\delta z^2}\sin^2\left(\frac{k_z\delta z}{2}\right)}, \tag{2.41}$$

such that the numerical dispersion, eq. (2.40), can be solved for the frequency as

$$\omega = \frac{2}{\delta t}\sin^{-1}(\xi). \tag{2.42}$$

For a loss-less propagating wave solution, the frequency is required to be real, resulting in the condition that $|\xi| \leq 1$. By taking the supremum of the sines, it is found that

$$\delta t \leq \frac{1}{c_0\sqrt{\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2}}}, \tag{2.43}$$

which is called the Courant-Friedrichs-Lewy (CFL) condition. This condition connects the spatial with the temporal discretisation, and if it is violated, massive stability errors will occur. For illustratory purposes, consider a wave propagating in a one-dimensional space with a time-step that violates the CFL condition such that $|\xi| > 1$, which in turn means that the frequency becomes complex. Applying the identity for the inverse sine of some complex number, α, given by

$$\sin^{-1}\alpha = -i\ln\left(i\alpha + \sqrt{1-\alpha^2}\right), \tag{2.44}$$

to eq. (2.42) can be shown to give

$$\omega\delta t = \pi - 2i\ln\left(\xi + \sqrt{\xi^2 - 1}\right). \tag{2.45}$$

A propagating wave with time-dependence on the form $e^{-i\omega t}$ discretised in time as $t = n\delta t$ will thus have a time dependence that can be expressed as

$$e^{-i\omega n\delta t} = e^{-i\pi n}\left(\xi + \sqrt{\xi^2 - 1}\right)^{2n}, \tag{2.46}$$

which indicate an infinite growth of the propagating fields. Thus, the CFL condition must indeed be obeyed. In fact, one would usually choose a time-step that is 0.99 times that of the maximum in eq. (2.43).

2.2.2 Boundary Conditions

So far, the fields inside the computational domain are simulated by the update equations of the Yee-algorithm. However, at the boundaries of the computational domain, the fields cannot simply be updated based on adjacent field components. Instead, they must be separately handled using knowledge of said boundary. Three different boundary conditions are utilised for the modelling in this project: PEC, absorbing, and periodic boundary conditions.

A perfect electric conductor (PEC) boundary requires the electric field components tangential to the boundary to be zero on the boundary. This results in a perfect reflection of the wave incident on the boundary. Implementation of this is to simply have the electric field at the boundary set to zero initially and not to update the electric field at the boundary during the simulation.

A periodic boundary requires that the fields at the boundary are identical to the fields at the opposite boundary. As such, in principle, a wave that is incident on the boundary will continue from the opposite boundary. Implementation of a periodic boundary condition is to update the field at one boundary using the field components at the opposite boundary.

Numerous methods have been developed for absorbing boundary conditions. Most notable are the Engquist-Majda absorbing boundary condition [13] and the perfectly matched layer [14]. The Engquist-Majda absorbing boundary condition was originally developed for acoustic waves, but it was later adapted to the FDTD method by G. Mur [15]. Several modifications and improvements have since been developed [16, 17, 18, 19]. While, for certain problems, these boundary conditions have sufficiently low reflection errors, for most scattering problems, the boundary conditions must be extensively extended to avoid evanescent fields and grazing angle incidence to overcome their limitations. However, in 1994, J. Berenger [14] pioneered the perfectly matched layer (PML), which overcame several of the yet to be solved limitations of absorbing boundary conditions, e.g. it absorbs both propagating and evanescent waves, and it can be used in complex media such as anisotropic or lossy media.

2.3 Periodic Boundary Condition

This section will focus on the periodic boundary condition, and it is based on chapter two of *Scattering Analysis of Periodic Structures Using Finite-Difference Time-Domain Method* by K. ElMahgoub et al. [20].

Consider the situation illustrated in fig. 2.2. The periodic boundary condition shall be enforced at the boundaries of the staggered grid, $i = 1$ and $i = N_x$. The update equations for E_y and E_z at $i = 1$ requires H_z and H_y at $i = 1/2$. Even though this is outside the computational domain, the periodic boundary condition provides the necessary information. The computational domain, which is considered to encapsulate a single period of the simulated periodic structure, is artificially extended at $i = 1$, i.e. H_z and H_y at $i = 1/2$ are known, but not computed. From the periodicity, it is required that E_y and E_z at $i = 1$ is equal to E_y and E_z at $i = N_z$, respectively, and it immediately follows that H_z and H_y at $i = 1/2$ must be equal to H_z and H_y at $i = N_x - 1/2$, respectively.

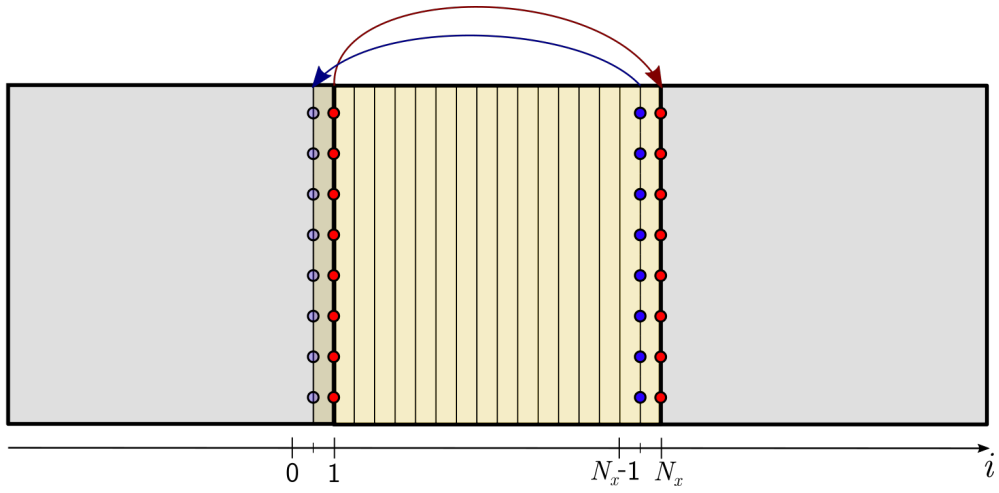


Figure 2.2: Illustration of the implementation of a periodic boundary condition along x. The computational domain is artificially extended by copying the magnetic field components at $i = N_x - 1/2$ and inserting them at $i = 1/2$. The electric field components at $i = 1$ are then updated and copied to replace the electric field components at $i = N_x$.

By also enforcing the periodic boundary condition at $j = 1$ and $j = N_y$, the update equations become

$$E_x\big|^{n+1/2}_{i+1/2,1,k} = E_x\big|^{n-1/2}_{i+1/2,1,k} + \frac{\delta t}{\epsilon\big|_{i+1/2,1,k}} \left(\frac{H_z\big|^{n}_{i+1/2,1+1/2,k} - H_z\big|^{n}_{i+1/2,N_y,k}}{\delta y} - \frac{H_y\big|^{n}_{i+1/2,1,k+1/2} - H_y\big|^{n}_{i+1/2,1,k-1/2}}{\delta z} \right), \tag{2.47}$$

$$E_y\big|^{n+1/2}_{1,j+1/2,k} = E_y\big|^{n-1/2}_{1,j+1/2,k} + \frac{\delta t}{\epsilon\big|_{1,j+1/2,k}} \left(\frac{H_x\big|^{n}_{1,j+1/2,k+1/2} - H_x\big|^{n}_{1,j+1/2,k-1/2}}{\delta z} - \frac{H_z\big|^{n}_{1+1/2,j+1/2,k} - H_z\big|^{n}_{N_x,j+1/2,k}}{\delta x} \right), \tag{2.48}$$

$$E_z\big|_{i,1,k+1/2}^{n+1/2} = E_z\big|_{i,1,k+1/2}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i,1,k+1/2}} \left(\frac{H_y\big|_{i+1/2,1,k+1/2}^{n} - H_y\big|_{i-1/2,1,k+1/2}^{n}}{\delta x} - \frac{H_x\big|_{i,1+1/2,k+1/2}^{n} - H_x\big|_{i,N_y,k+1/2}^{n}}{\delta y} \right), \tag{2.49}$$

$$E_z\big|_{1,j,k+1/2}^{n+1/2} = E_z\big|_{1,j,k+1/2}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{1,j,k+1/2}} \left(\frac{H_y\big|_{1+1/2,j,k+1/2}^{n} - H_y\big|_{N_x,j,k+1/2}^{n}}{\delta x} - \frac{H_x\big|_{1,j+1/2,k+1/2}^{n} - H_x\big|_{1,j-1/2,k+1/2}^{n}}{\delta y} \right), \tag{2.50}$$

$$E_z\big|_{1,1,k+1/2}^{n+1/2} = E_z\big|_{1,1,k+1/2}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{1,1,k+1/2}} \left(\frac{H_y\big|_{1+1/2,1,k+1/2}^{n} - H_y\big|_{N_x,1,k+1/2}^{n}}{\delta x} - \frac{H_x\big|_{1,1+1/2,k+1/2}^{n} - H_x\big|_{1,N_y,k+1/2}^{n}}{\delta y} \right), \tag{2.51}$$

$$E_x\big|_{i+1/2,N_y,k}^{n+1/2} = E_x\big|_{i+1/2,1,k}^{n+1/2}, \tag{2.52}$$

$$E_y\big|_{N_x,j+1/2,k}^{n+1/2} = E_y\big|_{1,j+1/2,k}^{n+1/2}, \tag{2.53}$$

$$E_z\big|_{N_y,j,k+1/2}^{n+1/2} = E_z\big|_{1,j,k+1/2}^{n+1/2}, \tag{2.54}$$

$$E_z\big|_{i,N_x,k+1/2}^{n+1/2} = E_z\big|_{i,1,k+1/2}^{n+1/2}, \tag{2.55}$$

$$E_z\big|_{1,N_y,k+1/2}^{n+1/2} = E_z\big|_{N_x,1,k+1/2}^{n+1/2} = E_z\big|_{N_x,N_y,k+1/2}^{n+1/2} = E_z\big|_{1,1,k+1/2}^{n+1/2}. \tag{2.56}$$

Since only fields at normal incidence on the structure is of interest in this project, these update equations at the periodic boundaries are adequate. However, if the incident angle on the structure is of interest, a good method for implementing the periodic boundary condition is the so-called constant horizontal wave number approach [21].

2.4 Perfectly Matched Layer

In 1994, J. Berenger [14] first developed the PML by considering a splitting of the field components into two subcomponents, e.g. $H_x = H_{xy} + H_{xz}$, and having the PML be a lossy

medium with carefully selected electric and magnetic conductivities to decay outgoing radiation. Shortly after, W. Chew and W. Weedon [22] developed a form of the PML based on a coordinate stretching approach where the coordinates of the differential operators are stretched, e.g. $\nabla_e = \hat{\mathbf{x}} e_x^{-1} \partial/\partial x + \hat{\mathbf{y}} e_y^{-1} \partial/\partial y + \hat{\mathbf{z}} e_z^{-1} \partial/\partial z$. Both of these approaches of constructing a PML requires significant modifications to Maxwell's equations. The following year, Z. Sacks et al. [23] developed an approach for deriving the PML based on anisotropic material properties such that no splitting of the field components is required. This approach does not increase the reflection error [24], and it simplifies the implementation of the PML significantly as it does not require modifications to Maxwell's equations; making it the preferable option compared to the split-field approach. Then, in 1996, M. Kuzuoglu and R. Mittra [25] examined the constitutive parameters of the PML, i.e. conductivities, permittivity, and permeability, and derived a strictly causal form of the PML that is now known as the complex-frequency shifted (CFS) PML. The CFS-PML has since been broadly regarded as the superior of existing PML-forms [26]. Several implementations of this form have since had success, e.g. the convolutional PML [27] and the auxiliary differential equation (ADE) PML [28]. The latter of which will is used for the numerical modelling in this project.

2.4.1 The Complex-Frequency Shifted Perfectly Matched Layer

The first step is to consider the coordinate stretching approach by W. Chew and W. Weedon [22], in which the source free Maxwell's equations are modified as

$$\boldsymbol{\nabla}_e \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t}, \tag{2.57}$$

$$\boldsymbol{\nabla}_h \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t}, \tag{2.58}$$

$$\boldsymbol{\nabla}_e \cdot \epsilon \mathbf{E} = 0, \tag{2.59}$$

$$\boldsymbol{\nabla}_h \cdot \mu \mathbf{H} = 0, \tag{2.60}$$

with

$$\boldsymbol{\nabla}_e = \hat{\mathbf{x}} \frac{1}{e_x} \frac{\partial}{\partial x} + \hat{\mathbf{y}} \frac{1}{e_y} \frac{\partial}{\partial y} + \hat{\mathbf{z}} \frac{1}{e_z} \frac{\partial}{\partial z}, \tag{2.61}$$

$$\boldsymbol{\nabla}_h = \hat{\mathbf{x}} \frac{1}{h_x} \frac{\partial}{\partial x} + \hat{\mathbf{y}} \frac{1}{h_y} \frac{\partial}{\partial y} + \hat{\mathbf{z}} \frac{1}{h_z} \frac{\partial}{\partial z}, \tag{2.62}$$

where e_ν and h_ν for $\nu \in \{x, y, z\}$ are the coordinate-stretching variables. It turns out that for a perfectly matched interface, i.e. for no reflection error, it is required that $e_\nu = h_\nu = s_\nu$ for $\nu \in \{x, y, z\}$ and that the coordinate-stretching variables associated with the directions tangential to the interface are equal, e.g. for an interface in the (x, y)-plane, a perfectly matched interface would require that $e_\nu = h_\nu = s_\nu$ for $\nu \in \{x, y, z\}$ and that $s_x = s_y$. [22]

With this considered, x-projection of the stretched coordinate modified Maxwell's equations, eqs. (2.57) and (2.58), in the frequency domain may be expressed as

$$i\omega\mu \tilde{H}_x = \frac{1}{s_y} \frac{\partial \tilde{E}_z}{\partial y} - \frac{1}{s_z} \frac{\partial \tilde{E}_y}{\partial z}, \tag{2.63}$$

$$-i\omega\epsilon \tilde{E}_x = \frac{1}{s_y} \frac{\partial \tilde{H}_z}{\partial y} - \frac{1}{s_z} \frac{\partial \tilde{H}_y}{\partial z}, \tag{2.64}$$

where tilde represents that it is in the frequency domain. The stretched-coordinate variables are chosen in accordance with the CFS-PML parameters as

$$s_\nu = \kappa_\nu + \frac{\sigma_\nu}{a_\nu + i\omega\epsilon_0}, \text{ for } \nu = x, y, z, \tag{2.65}$$

where a_ν, κ_ν, and σ_ν are positive real, and they may either be a one-dimensional function or a constant. Auxiliary variables, B_ν for $\nu \in \{x, y, z\}$, shall now be introduced by

$$\frac{1}{s_\nu} = \frac{1}{\kappa_\nu} - \frac{1}{B_\nu}, \text{ for } \nu = x, y, z, \tag{2.66}$$

such that

$$B_\nu = \frac{\kappa_\nu}{\sigma_\nu}\left[\kappa_\nu\left(a_\nu + i\omega\epsilon_0\right) + \sigma_\nu\right], \text{ for } \nu = x, y, z. \tag{2.67}$$

Applying eq. (2.66) to the the first derivative on the right hand side of eq. (2.63) gives

$$\frac{1}{s_y}\frac{\partial \tilde{E}_z}{\partial y} = \frac{1}{\kappa_y}\frac{\partial \tilde{E}_z}{\partial y} + \tilde{Q}^{(E)}_{y,z}, \tag{2.68}$$

where the introduced auxiliary parameter, $\tilde{Q}^{(E)}_{y,z}$, must satisfy

$$\tilde{Q}^{(E)}_{y,z} = -\frac{1}{B_y}\frac{\partial \tilde{E}_z}{\partial y}. \tag{2.69}$$

Applying eq. (2.67) to eq. (2.69) gives

$$i\omega\epsilon_0\kappa_y\tilde{Q}^{(E)}_{y,z} + \left(\kappa_y a_y + \sigma_y\right)\tilde{Q}^{(E)}_{y,z} = -\frac{\sigma_y}{\kappa_y}\frac{\partial \tilde{E}_z}{\partial y}. \tag{2.70}$$

Transforming this into the time domain finally gives the auxiliary differential equation as

$$-\epsilon_0\kappa_y\frac{\partial Q^{(E)}_{y,z}}{\partial t} + \left(\kappa_y a_y + \sigma_y\right)Q^{(E)}_{y,z} = -\frac{\sigma_y}{\kappa_y}\frac{\partial E_z}{\partial y}. \tag{2.71}$$

As such, eqs. (2.63) and (2.64) can, respectively, be expressed in the time domain as

$$-\frac{\partial}{\partial t}\mu H_x = \frac{1}{\kappa_y}\frac{\partial E_z}{\partial y} - \frac{1}{\kappa_z}\frac{\partial E_y}{\partial z} + Q^{(E)}_{y,z} + Q^{(E)}_{z,y}, \tag{2.72}$$

$$\frac{\partial}{\partial t}\epsilon E_x = \frac{1}{\kappa_y}\frac{\partial H_z}{\partial y} - \frac{1}{\kappa_z}\frac{\partial H_y}{\partial z} + Q^{(H)}_{y,z} + Q^{(H)}_{z,y}. \tag{2.73}$$

By following the same approach for the y- and z-projections of the stretched coordinate modified source free Maxwell's equations, the Maxwell's equations can be expressed as

$$-\frac{\partial}{\partial t}\mu\mathbf{H} = \boldsymbol{\nabla}_\kappa \times \mathbf{E} + \mathbf{Q}^{(E)}, \tag{2.74}$$

$$\frac{\partial}{\partial t}\epsilon\mathbf{E} = \boldsymbol{\nabla}_\kappa \times \mathbf{H} + \mathbf{Q}^{(H)}, \tag{2.75}$$

where

$$\boldsymbol{\nabla}_\kappa = \hat{\mathbf{x}}\frac{1}{\kappa_x}\frac{\partial}{\partial x} + \hat{\mathbf{y}}\frac{1}{\kappa_y}\frac{\partial}{\partial y} + \hat{\mathbf{z}}\frac{1}{\kappa_z}\frac{\partial}{\partial z}, \tag{2.76}$$

and $\mathbf{Q}^{(E)}$ must satisfy the auxiliary differential equation given by

$$\kappa_\nu\epsilon_0\frac{\partial \mathbf{Q}^{(E)}}{\partial t} + \left(\kappa_\nu a_\nu + \sigma_\nu\right)\mathbf{Q}^{(E)} = -\frac{\sigma_\nu}{\kappa_\nu}\frac{\partial}{\partial \boldsymbol{\nu}}\left(\hat{\boldsymbol{\nu}} \times \mathbf{E}\right), \text{ for } \nu = x, y, z, \tag{2.77}$$

and similarly for $\mathbf{Q}^{(H)}$.

CFS-PML in Finite-Difference Time-Domain

To implement the PML, the auxiliary fields, $\mathbf{Q}^{(E)}$ and $\mathbf{Q}^{(H)}$, are conveniently co-located with their respective electric and magnetic fields in both space and time in accordance with the Yee-algorithm. The auxiliary differential equation, eq. (2.77), may be solved at each time-step as a unit-step response solution. Consider the scalar equation for the auxiliary variable in eq. (2.71): It may be considered a first-order differential equation for $Q_{y,z}^{(E)}$ that is driven by the forcing function $\partial E_z/\partial y$, which may be sufficiently approximated as a piece-wise constant function. The solution is then on the form

$$Q_{y,z}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n+1/2} = b_y Q_{y,z}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n-1/2} - c_y \frac{\partial E_z}{\partial y}\Big|_{i,j,k+1/2}^{n+1/2}, \tag{2.78}$$

where

$$b_y = e^{-\frac{\delta t}{\tau}}, \tag{2.79}$$

$$c_y = \frac{\sigma_y}{\kappa_y\left(\kappa_y a_y + \sigma_y\right)}\left(1 - b_y\right), \tag{2.80}$$

and

$$\tau = \frac{\kappa_y \epsilon_0}{\kappa_y a_y + \sigma_y}. \tag{2.81}$$

By applying the finite-difference approximation to the partial derivative of E_z, the auxiliary parameter may finally be expressed as

$$Q_{y,z}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n+1/2} = b_y Q_{y,z}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n-1/2} - \frac{c_y}{\delta y}\left(E_z\Big|_{i,j+1,k+1/2}^{n+1/2} - E_z\Big|_{i,j,k+1/2}^{n+1/2}\right). \tag{2.82}$$

As such the update equations in the Yee-algorithm inside the PML becomes

$$\begin{aligned} E_x\big|_{i+1/2,j,k}^{n+1/2} = E_x\big|_{i+1/2,j,k}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i+1/2,j,k}} \Bigg(& \frac{H_z\big|_{i+1/2,j+1/2,k}^{n} - H_z\big|_{i+1/2,j-1/2,k}^{n}}{\kappa_y \delta y} \\ & - \frac{H_y\big|_{i+1/2,j,k+1/2}^{n} - H_y\big|_{i+1/2,j,k-1/2}^{n}}{\kappa_z \delta z} \Bigg) \\ & + Q_{y,z}^{(H)}\Big|_{i+1/2,j,k}^{n} + Q_{z,y}^{(H)}\Big|_{i+1/2,j,k}^{n}, \end{aligned} \tag{2.83}$$

$$\begin{aligned} E_y\big|_{i,j+1/2,k}^{n+1/2} = E_y\big|_{i,j+1/2,k}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i,j+1/2,k}} \Bigg(& \frac{H_x\big|_{i,j+1/2,k+1/2}^{n} - H_x\big|_{i,j+1/2,k-1/2}^{n}}{\kappa_z \delta z} \\ & - \frac{H_z\big|_{i+1/2,j+1/2,k}^{n} - H_z\big|_{i-1/2,j+1/2,k}^{n}}{\kappa_x \delta x} \Bigg) \\ & + Q_{z,x}^{(H)}\Big|_{i,j+1/2,k}^{n} + Q_{x,z}^{(H)}\Big|_{i+1/2,j,k}^{n}, \end{aligned} \tag{2.84}$$

$$E_z\big|_{i,j,k+1/2}^{n+1/2} = E_z\big|_{i,j,k+1/2}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i,j,k+1/2}} \left(\frac{H_y\big|_{i+1/2,j,k+1/2}^{n} - H_y\big|_{i-1/2,j,k+1/2}^{n}}{\kappa_x \delta x} - \frac{H_x\big|_{i,j+1/2,k+1/2}^{n} - H_x\big|_{i,j-1/2,k+1/2}^{n}}{\kappa_y \delta y} \right) + Q_{x,y}^{(H)}\Big|_{i,j,k+1/2}^{n} + Q_{y,x}^{(H)}\Big|_{i,j,k+1/2}^{n}, \tag{2.85}$$

$$H_x\big|_{i,j+1/2,k+1/2}^{n+1} = H_x\big|_{i,j+1/2,k+1/2}^{n} + \frac{\delta t}{\mu\big|_{i,j+1/2,k+1/2}} \left(\frac{E_y\big|_{i,j+1/2,k+1}^{n+1/2} - E_y\big|_{i,j+1/2,k}^{n+1/2}}{\kappa_z \delta z} - \frac{E_z\big|_{i,j+1,k+1/2}^{n+1/2} - E_z\big|_{i,j,k+1/2}^{n+1/2}}{\kappa_y \delta y} \right) - Q_{z,y}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n+1/2} - Q_{y,z}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n+1/2}, \tag{2.86}$$

$$H_y\big|_{i+1/2,j,k+1/2}^{n+1} = H_y\big|_{i+1/2,j,k+1/2}^{n} + \frac{\delta t}{\mu\big|_{i+1/2,j,k+1/2}} \left(\frac{E_z\big|_{i+1,j,k+1/2}^{n+1/2} - E_z\big|_{i,j,k+1/2}^{n+1/2}}{\kappa_x \delta x} - \frac{E_x\big|_{i+1/2,j,k+1}^{n+1/2} - E_x\big|_{i+1/2,j,k}^{n+1/2}}{\kappa_z \delta z} \right) - Q_{x,z}^{(E)}\Big|_{i+1/2,j,k+1/2}^{n+1/2} - Q_{z,x}^{(E)}\Big|_{i+1/2,j,k+1/2}^{n+1/2}, \tag{2.87}$$

$$H_z\big|_{i+1/2,j+1/2,k}^{n+1} = H_z\big|_{i+1/2,j+1/2,k}^{n} + \frac{\delta t}{\mu\big|_{i+1/2,j+1/2,k}} \left(\frac{E_x\big|_{i+1/2,j+1,k}^{n+1/2} - E_x\big|_{i+1/2,j,k}^{n+1/2}}{\kappa_y \delta y} - \frac{E_y\big|_{i+1,j+1/2,k}^{n+1/2} - E_y\big|_{i,j+1/2,k}^{n+1/2}}{\kappa_x \delta x} \right) - Q_{y,x}^{(E)}\Big|_{i+1/2,j+1/2,k}^{n+1/2} - Q_{x,y}^{(E)}\Big|_{i+1/2,j+1/2,k}^{n+1/2}, \tag{2.88}$$

with

$$Q_{y,z}^{(H)}\Big|_{i+1/2,j,k}^{n} = b_y Q_{y,z}^{(H)}\Big|_{i+1/2,j,k}^{n-1} - \frac{c_y}{\delta y}\left(H_z\Big|_{i+1/2,j+1/2,k}^{n} - H_z\Big|_{i+1/2,j-1/2,k}^{n}\right), \tag{2.89}$$

$$Q_{z,y}^{(H)}\Big|_{i+1/2,j,k}^{n} = b_z Q_{z,y}^{(H)}\Big|_{i+1/2,j,k}^{n-1} - \frac{c_z}{\delta z}\left(H_y\Big|_{i+1/2,j,k+1/2}^{n} - H_y\Big|_{i+1/2,j,k-1/2}^{n}\right), \tag{2.90}$$

$$Q_{z,x}^{(H)}\Big|_{i,j+1/2,k}^{n} = b_z Q_{z,x}^{(H)}\Big|_{i,j+1/2,k}^{n-1} - \frac{c_z}{\delta z}\left(H_x\Big|_{i,j+1/2,k+1/2}^{n} - H_x\Big|_{i,j+1/2,k-1/2}^{n}\right), \tag{2.91}$$

$$Q_{x,z}^{(H)}\Big|_{i,j+1/2,k}^{n} = b_x Q_{x,z}^{(H)}\Big|_{i,j+1/2,k}^{n-1} - \frac{c_x}{\delta x}\left(H_z\Big|_{i+1/2,j+1/2,k}^{n} - H_z\Big|_{i-1/2,j+1/2,k}^{n}\right), \tag{2.92}$$

$$Q_{x,y}^{(H)}\Big|_{i,j,k+1/2}^{n} = b_x Q_{x,y}^{(H)}\Big|_{i,j,k+1/2}^{n-1} - \frac{c_x}{\delta x}\left(H_y\Big|_{i+1/2,j,k+1/2}^{n} - H_y\Big|_{i-1/2,j,k+1/2}^{n}\right), \tag{2.93}$$

$$Q_{y,x}^{(H)}\Big|_{i,j,k+1/2}^{n} = b_y Q_{y,x}^{(H)}\Big|_{i,j,k+1/2}^{n-1} - \frac{c_y}{\delta y}\left(H_z\Big|_{i,j+1/2,k+1/2}^{n} - H_z\Big|_{i,j-1/2,k+1/2}^{n}\right), \tag{2.94}$$

$$Q_{z,y}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n+1/2} = b_z Q_{z,y}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n-1/2} - \frac{c_z}{\delta z}\left(E_y\Big|_{i,j+1/2,k+1}^{n+1/2} - E_y\Big|_{i,j+1/2,k}^{n+1/2}\right), \tag{2.95}$$

$$Q_{y,z}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n+1/2} = b_y Q_{y,z}^{(E)}\Big|_{i,j+1/2,k+1/2}^{n-1/2} - \frac{c_y}{\delta y}\left(E_z\Big|_{i,j+1,k+1/2}^{n+1/2} - E_z\Big|_{i,j,k+1/2}^{n+1/2}\right), \tag{2.96}$$

$$Q_{x,z}^{(E)}\Big|_{i+1/2,j,k+1/2}^{n+1/2} = b_x Q_{x,z}^{(E)}\Big|_{i+1/2,j,k+1/2}^{n-1/2} - \frac{c_x}{\delta x}\left(E_z\Big|_{i+1,j,k+1/2}^{n+1/2} - E_z\Big|_{i,j,k+1/2}^{n+1/2}\right), \tag{2.97}$$

$$Q_{z,x}^{(E)}\Big|_{i+1/2,j,k+1/2}^{n+1/2} = b_z Q_{z,x}^{(E)}\Big|_{i+1/2,j,k+1/2}^{n-1/2} - \frac{c_z}{\delta z}\left(E_x\Big|_{i+1/2,j,k+1}^{n+1/2} - E_x\Big|_{i+1/2,j,k}^{n+1/2}\right), \tag{2.98}$$

$$Q_{y,x}^{(E)}\Big|_{i+1/2,j+1/2,k}^{n+1/2} = b_y Q_{y,x}^{(E)}\Big|_{i+1/2,j+1/2,k}^{n-1/2} - \frac{c_y}{\delta y}\left(E_x\Big|_{i+1/2,j+1,k}^{n+1/2} - E_x\Big|_{i+1/2,j,k}^{n+1/2}\right), \tag{2.99}$$

$$Q_{x,y}^{(E)}\Big|_{i+1/2,j+1/2,k}^{n+1/2} = b_x Q_{x,y}^{(E)}\Big|_{i+1/2,j+1/2,k}^{n-1/2} - \frac{c_x}{\delta x}\left(E_y\Big|_{i+1,j+1/2,k}^{n+1/2} - E_y\Big|_{i,j+1/2,k}^{n+1/2}\right), \tag{2.100}$$

where

$$b_\nu = e^{-\frac{\delta t}{\tau_\nu}}, \text{ for } \nu = x, y, z, \tag{2.101}$$

$$c_\nu = \frac{\sigma_\nu}{\kappa_\nu\left(\kappa_\nu a_\nu + \sigma_y\right)}\left(1 - b_\nu\right), \text{ for } \nu = x, y, z, \tag{2.102}$$

and

$$\tau_\nu = \frac{\kappa_\nu \epsilon_0}{\kappa_\nu a_\nu + \sigma_\nu}, \text{ for } \nu = x, y, z. \tag{2.103}$$

Finally, it is noted that, for the CFL-condition to hold, it is required that

$$0 \leq b_\nu \leq 1, \text{ for } \nu = x, y, z, \tag{2.104}$$

and

$$0 \leq c_\nu \leq \frac{1}{\kappa_\nu}, \text{ for } \nu = x, y, z. \tag{2.105}$$

With the update equations now explicitly known for the field components inside the PML, all that is left to determine are the PML-parameters: κ_ν, a_ν, and σ_ν.

2.4.2 PML-Parameters and Reflection Errors

To illustrate the relevance of scaling the PML-parameters, κ_ν, a_ν, and σ_ν, consider an arbitrarily polarised plane wave entering a PML in the (x, y)-plane with the interface at $z = z_0$. The electric and magnetic fields must satisfy the modified Maxwell's equations given in the frequency domain as

$$i\omega\mu\tilde{\mathbf{H}} = \boldsymbol{\nabla}_s \times \tilde{\mathbf{E}}, \tag{2.106}$$

$$-i\omega\epsilon\tilde{\mathbf{E}} = \boldsymbol{\nabla}_s^* \times \tilde{\mathbf{H}}, \tag{2.107}$$

where

$$\boldsymbol{\nabla}_s = \hat{\mathbf{x}}\frac{1}{s_x}\frac{\partial}{\partial x} + \hat{\mathbf{y}}\frac{1}{s_y}\frac{\partial}{\partial y} + \hat{\mathbf{z}}\frac{1}{s_z}\frac{\partial}{\partial z}, \tag{2.108}$$

$$\boldsymbol{\nabla}_s^* = \hat{\mathbf{x}}\frac{1}{s_x^*}\frac{\partial}{\partial x} + \hat{\mathbf{y}}\frac{1}{s_y^*}\frac{\partial}{\partial y} + \hat{\mathbf{z}}\frac{1}{s_z^*}\frac{\partial}{\partial z}. \tag{2.109}$$

These versions of Maxwell's equations are equivalent to eqs. (2.74) and (2.75). General plane-wave solutions are on the forms

$$\tilde{\mathbf{E}} = \tilde{\mathbf{E}}_0 e^{i\mathbf{k}\cdot\mathbf{r}} e^{-i\omega t}, \tag{2.110}$$

$$\tilde{\mathbf{H}} = \tilde{\mathbf{H}}_0 e^{i\mathbf{k}\cdot\mathbf{r}} e^{-i\omega t}. \tag{2.111}$$

Substituting eqs. (2.110) and (2.111) into eqs. (2.106) and (2.107), it is found that

$$\omega\mu\tilde{\mathbf{H}} = \mathbf{k}_s \times \tilde{\mathbf{E}}, \tag{2.112}$$

$$-\omega\epsilon\tilde{\mathbf{E}} = \mathbf{k}_s^* \times \tilde{\mathbf{H}}, \tag{2.113}$$

where

$$\mathbf{k}_s = \hat{\mathbf{x}}\frac{k_x}{s_x} + \hat{\mathbf{y}}\frac{k_y}{s_y} + \hat{\mathbf{z}}\frac{k_z}{s_z}, \tag{2.114}$$

$$\mathbf{k}_s^* = \hat{\mathbf{x}}\frac{k_x}{s_x^*} + \hat{\mathbf{y}}\frac{k_y}{s_y^*} + \hat{\mathbf{z}}\frac{k_z}{s_z^*}. \tag{2.115}$$

Combining eqs. (2.112) and (2.113) gives a wave vector equation as

$$-\omega^2\epsilon\mu\tilde{\mathbf{E}} = \mathbf{k}_s^* \times \left(\mathbf{k}_s \times \tilde{\mathbf{E}}\right). \tag{2.116}$$

Applying the triple vector product identity,

$$\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = \mathbf{B}(\mathbf{C} \cdot \mathbf{A}) - \mathbf{C}(\mathbf{A} \cdot \mathbf{B}), \tag{2.117}$$

gives

$$-\omega^2\epsilon\mu\tilde{\mathbf{E}} = \mathbf{k}_s\left(\tilde{\mathbf{E}} \cdot \mathbf{k}_s^*\right) - \tilde{\mathbf{E}}\left(\mathbf{k}_s^* \cdot \mathbf{k}_s\right). \tag{2.118}$$

From eq. (2.113), it is clear that $\tilde{\mathbf{E}}$ and $\mathbf{k}_s^*$ are orthogonal, and thus $\tilde{\mathbf{E}} \cdot \mathbf{k}_s^* = 0$, such that the dispersion relation is found to be

$$\omega^2\epsilon\mu = \mathbf{k}_s \cdot \mathbf{k}_s^*, \tag{2.119}$$

or

$$k_0\sqrt{\epsilon_r\mu_r} = \sqrt{\frac{k_x^2}{s_x s_x^*} + \frac{k_y^2}{s_y s_y^*} + \frac{k_z^2}{s_z s_z^*}}, \tag{2.120}$$

where $k_0 = \omega\sqrt{\epsilon_0\mu_0}$.

Consider now the case depicted in fig. 2.3. The electric field of the wave incident on the PML-interface given by

$$\tilde{\mathbf{E}}_i = \tilde{\mathbf{E}}_{0,i} e^{i\mathbf{k}_i\cdot\mathbf{r}} e^{-i\omega t}, \tag{2.121}$$

and the subsequent reflected and transmitted waves are given by

$$\tilde{\mathbf{E}}_r = \tilde{\mathbf{E}}_{0,t} e^{i\mathbf{k}_r\cdot\mathbf{r}} e^{-i\omega t}, \tag{2.122}$$

$$\tilde{\mathbf{E}}_t = \tilde{\mathbf{E}}_{0,r} e^{i\mathbf{k}_t\cdot\mathbf{r}} e^{-i\omega t}. \tag{2.123}$$

From eq. (2.112), the corresponding magnetic fields are found by

$$\tilde{\mathbf{H}}_i = \frac{\mathbf{k}_{s,i}\times\tilde{\mathbf{E}}_i}{\omega\mu_0\mu_1}, \tag{2.124}$$

$$\tilde{\mathbf{H}}_r = \frac{\mathbf{k}_{s,r}\times\tilde{\mathbf{E}}_r}{\omega\mu_0\mu_1}, \tag{2.125}$$

$$\tilde{\mathbf{H}}_t = \frac{\mathbf{k}_{s,t}\times\tilde{\mathbf{E}}_t}{\omega\mu_0\mu_2}, \tag{2.126}$$

where

$$\mathbf{k}_{s,i} = \hat{\mathbf{x}}\frac{k_{x,i}}{s_{x,1}} + \hat{\mathbf{y}}\frac{k_{y,i}}{s_{y,1}} + \hat{\mathbf{z}}\frac{k_{z,i}}{s_{z,1}}, \tag{2.127}$$

$$\mathbf{k}_{s,r} = \hat{\mathbf{x}}\frac{k_{x,r}}{s_{x,1}} + \hat{\mathbf{y}}\frac{k_{y,r}}{s_{y,1}} + \hat{\mathbf{z}}\frac{k_{z,r}}{s_{z,1}}, \tag{2.128}$$

$$\mathbf{k}_{s,t} = \hat{\mathbf{x}}\frac{k_{x,t}}{s_{x,2}} + \hat{\mathbf{y}}\frac{k_{y,t}}{s_{y,2}} + \hat{\mathbf{z}}\frac{k_{z,t}}{s_{z,2}}. \tag{2.129}$$

Notice that the phase matching condition at the interface requires that

$$k_{x,i} = k_{x,r} = k_{x,t}, \tag{2.130}$$

$$k_{y,i} = k_{y,r} = k_{y,t}. \tag{2.131}$$

It also follows that

$$k_{z,1} = k_{z,i} = -k_{z,r}, \tag{2.132}$$

$$k_{z,2} = k_{z,t}. \tag{2.133}$$

From eqs. (2.124) to (2.126) the magnetic fields in region 1 and 2 can, respectively, be found as

$$\tilde{\mathbf{H}}_1 = \frac{\mathbf{k}_{s,i}\times\tilde{\mathbf{E}}_i + \mathbf{k}_{s,i}\times\tilde{\mathbf{E}}_r}{\omega\mu_0\mu_1}, \tag{2.134}$$

$$\tilde{\mathbf{H}}_2 = \frac{\mathbf{k}_2\times\tilde{\mathbf{E}}_t}{\omega\mu_0\mu_2}. \tag{2.135}$$

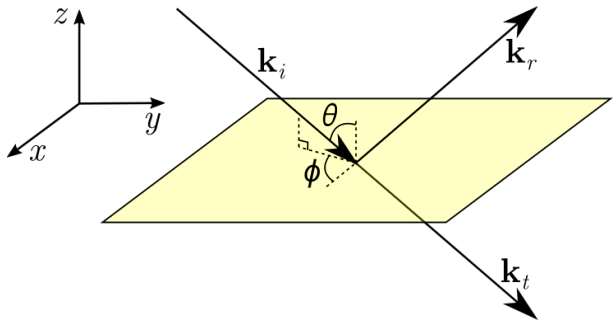


Figure 2.3: Illustration of an incident wave on a PML.

The plane wave shall now be decomposed into transverse electric (TE) and transverse magnetic (TM) waves. TE- and TM-waves will have no electric and magnetic field components in the direction normal to the interface, respectively. Consider now a TE-wave. From the continuity condition that requires the tangential electric field components to be continuous over an interface, it is found that

$$\tilde{\mathbf{E}}_{0,i}^{(\mathrm{TE})} + \tilde{\mathbf{E}}_{0,r}^{(\mathrm{TE})} = \tilde{\mathbf{E}}_{0,t}^{(\mathrm{TE})}, \text{ for } z = z_0, \tag{2.136}$$

for a TE-wave. Furthermore, from the phase matching condition, it can be inferred that the incident, reflected, and transmitted fields must be oscillating in the same plane. Consequently, they can be normalised with respect to the incident field as

$$\tilde{\mathbf{E}}_{0,i}^{(\mathrm{TE})} = \tilde{\mathbf{E}}_{0}^{(\mathrm{TE})}, \tag{2.137}$$

$$\tilde{\mathbf{E}}_{0,r}^{(\mathrm{TE})} = R^{(\mathrm{TE})}\tilde{\mathbf{E}}_{0}^{(\mathrm{TE})}, \tag{2.138}$$

$$\tilde{\mathbf{E}}_{0,t}^{(\mathrm{TE})} = T^{(\mathrm{TE})}\tilde{\mathbf{E}}_{0}^{(\mathrm{TE})}, \tag{2.139}$$

such that eq. (2.136) becomes

$$1 + R^{(\mathrm{TE})} = T^{(\mathrm{TE})}, \tag{2.140}$$

where $R^{(\mathrm{TE})}$ and $T^{(\mathrm{TE})}$ represent the reflection and transmission coefficients, respectively. By also requiring no surface current on the interface, the tangential magnetic field components must also be continuous over the interface, such that

$$\hat{\mathbf{n}} \times \tilde{\mathbf{H}}_{1}^{(\mathrm{TE})} = \hat{\mathbf{n}} \times \tilde{\mathbf{H}}_{2}^{(\mathrm{TE})}, \text{ for } z = z_0, \tag{2.141}$$

where $\hat{\mathbf{n}}$ is the unit normal vector chosen to be pointing from region 2 into region 1. In this particular case, $\hat{\mathbf{n}} = \hat{\mathbf{z}}$. Substituting eqs. (2.134) and (2.135) into eq. (2.141) and applying the triple vector product identity, eq. (2.117), gives

$$\begin{aligned}\frac{1}{\mu_1}\mathbf{k}_1\left(\hat{\mathbf{n}}\cdot\tilde{\mathbf{E}}_{i}^{(\mathrm{TE})} - \hat{\mathbf{n}}\cdot\tilde{\mathbf{E}}_{r}^{(\mathrm{TE})}\right) - \frac{1}{\mu_1}\left(\tilde{\mathbf{E}}_{i}^{(\mathrm{TE})} - \tilde{\mathbf{E}}_{r}^{(\mathrm{TE})}\right)\hat{\mathbf{n}}\cdot\mathbf{k}_1 \\ = \frac{1}{\mu_2}\mathbf{k}_2\left(\hat{\mathbf{n}}\cdot\tilde{\mathbf{E}}_{t}^{(\mathrm{TE})}\right) - \frac{1}{\mu_2}\tilde{\mathbf{E}}_{t}^{(\mathrm{TE})}\left(\hat{\mathbf{n}}\cdot\mathbf{k}_2\right), \text{ for } z = z_0.\end{aligned} \tag{2.142}$$

Substituting eqs. (2.121) to (2.123) with eqs. (2.137) to (2.139) into eq. (2.142) gives

$$\frac{k_{z,1}}{\mu_1 s_{z,1}}\left(e^{ik_{z,1}z_0} - R^{(\mathrm{TE})}e^{-ik_{z,1}z_0}\right) = \frac{k_{z,2}}{\mu_2 s_{z,2}}T^{(\mathrm{TE})}e^{ik_{z,2}z_0}, \tag{2.143}$$

where it has been used that $\hat{\mathbf{n}} \cdot \tilde{\mathbf{E}}_0 = 0$. Since the choice of z_0 should be inconsequential to the reflection and transmission at the interface, it must hold that

$$\frac{k_{z,1}}{\mu_1 s_{z,1}} \left(1 - R^{(\mathrm{TE})}\right) = \frac{k_{z,2}}{\mu_2 s_{z,2}} T^{(\mathrm{TE})}. \tag{2.144}$$

Now, eqs. (2.140) and (2.144) constitute a system of two equations with two unknowns, $R^{(\mathrm{TE})}$ and $T^{(\mathrm{TE})}$, which may be solved to find that

$$R^{(\mathrm{TE})} = \frac{\mu_2 s_{z,2} k_{z,1} - \mu_1 s_{z,1} k_{z,2}}{\mu_2 s_{z,2} k_{z,1} + \mu_1 s_{z,1} k_{z,2}}, \tag{2.145}$$

$$T^{(\mathrm{TE})} = \frac{2\mu_2 s_{z,2} k_{z,1}}{\mu_1 s_{z,1} k_{z,2} + \mu_2 s_{z,2} k_{z,1}}. \tag{2.146}$$

Following a similar procedure for the TM-component results in

$$R^{(\mathrm{TM})} = \frac{\epsilon_2 s^*_{z,2} k_{z,1} - \epsilon_1 s^*_{z,1} k_{z,2}}{\epsilon_2 s^*_{z,2} k_{z,1} + \epsilon_1 s^*_{z,1} k_{z,2}}, \tag{2.147}$$

$$T^{(\mathrm{TM})} = \frac{2\epsilon_2 s^*_{z,2} k_{z,1}}{\epsilon_1 s^*_{z,1} k_{z,2} + \epsilon_2 s^*_{z,2} k_{z,1}}. \tag{2.148}$$

The wave vector components in each region may be expressed from the dispersion relation, eq. (2.119), in terms of the incident and transmitted angles as

$$k_{x,1} = \sqrt{s_{x,1} s^*_{x,1}}\, k_0 \sqrt{\epsilon_1 \mu_1}\, \sin\theta_1 \cos\phi_1, \tag{2.149}$$

$$k_{y,1} = \sqrt{s_{y,1} s^*_{y,1}}\, k_0 \sqrt{\epsilon_1 \mu_1}\, \sin\theta_1 \sin\phi_1, \tag{2.150}$$

$$k_{z,1} = \sqrt{s_{z,1} s^*_{z,1}}\, k_0 \sqrt{\epsilon_1 \mu_1}\, \cos\theta_1, \tag{2.151}$$

$$k_{x,2} = \sqrt{s_{x,2} s^*_{x,2}}\, k_0 \sqrt{\epsilon_2 \mu_2}\, \sin\theta_2 \cos\phi_2, \tag{2.152}$$

$$k_{y,2} = \sqrt{s_{y,2} s^*_{y,2}}\, k_0 \sqrt{\epsilon_2 \mu_2}\, \sin\theta_2 \sin\phi_2, \tag{2.153}$$

$$k_{z,2} = \sqrt{s_{z,2} s^*_{z,2}}\, k_0 \sqrt{\epsilon_2 \mu_2}\, \cos\theta_2. \tag{2.154}$$

The phase matching condition, i.e. $k_{x,1} = k_{x,2}$ and $k_{y,1} = k_{y,2}$, requires that

$$\sqrt{s_{x,1} s^*_{x,1}}\, k_0 \sqrt{\epsilon_1 \mu_1}\, \sin\theta_1 \cos\phi_1 = \sqrt{s_{x,2} s^*_{x,2}}\, k_0 \sqrt{\epsilon_2 \mu_2}\, \sin\theta_2 \cos\phi_2, \tag{2.155}$$

$$\sqrt{s_{y,1} s^*_{y,1}}\, k_0 \sqrt{\epsilon_1 \mu_1}\, \sin\theta_1 \sin\phi_1 = \sqrt{s_{y,2} s^*_{y,2}}\, k_0 \sqrt{\epsilon_2 \mu_2}\, \sin\theta_2 \sin\phi_2. \tag{2.156}$$

By requiring the media of regions 1 and 2 to be perfectly matched, i.e. $\epsilon_1 = \epsilon_2$ and $\mu_1 = \mu_2$, as well as requiring that $s_{x,1} = s_{x,2}$ and $s_{y,1} = s_{y,2}$, then the incident and transmitted angles must be equal, i.e. $\theta_1 = \theta_2$ and $\phi_1 = \phi_2$. The reflection coefficients for TE- and TM-waves becomes

$$R^{(\mathrm{TE})} = \frac{1 - \sqrt{\frac{s_{z,1} s^*_{z,2}}{s_{z,2} s^*_{z,1}}}}{1 + \sqrt{\frac{s_{z,1} s^*_{z,2}}{s_{z,2} s^*_{z,1}}}}, \tag{2.157}$$

$$R^{(\mathrm{TM})} = \frac{\sqrt{\frac{s_{z,1} s^*_{z,2}}{s_{z,2} s^*_{z,1}}} - 1}{\sqrt{\frac{s_{z,1} s^*_{z,2}}{s_{z,2} s^*_{z,1}}} - 1}. \tag{2.158}$$

Thus, the reflection coefficients are independent of the incident angle. Furthermore, the coefficients are equal in size but opposite in sign. Therefore, it is convenient to define a reflection error from the interface between simulation domain and PML as

$$\Gamma_0 \equiv \left|R^{(\mathrm{TE})}\right| = \left|R^{(\mathrm{TM})}\right|. \tag{2.159}$$

Choosing that $s_z = s_z^*$ in both regions will give $\Gamma_0 = 0$, such that the perfectly matched layer interface is indeed reflectionless.

Thus, it has been shown that a propagating wave will not be reflected on the interface of a perfectly matched layer. However, since the PML is of finite thickness, the field will enter the PML, propagate to, and be reflected on, the PEC boundary of the computational domain, and finally propagate and reenter the simulation domain. Since the media are perfectly matched, the transmission out of and into the PML will be complete, and for a PEC, the reflection will be complete. As such, it is easily shown that the reflection coefficient for the entire PML is given by

$$R(\theta) = \pm e^{-2is_{z,2}\omega\sqrt{\epsilon\mu}d\cos\theta}, \tag{2.160}$$

where d is the thickness of the PML. The positive $(+)$ solution is for TE-waves, and the negative $(-)$ solution is for TM-waves. The reflection error over the entire PML, $\Gamma(\theta)$ may thus be given by the magnitude of the coefficients as

$$\Gamma(\theta) = |R(\theta)|. \tag{2.161}$$

It is clear that, for a propagating wave to be attenuated in the PML, the stretched coordinate variable, $s_{z,2}$ must be either complex or imaginary. Furthermore, increasing the PML thickness will decrease the reflection, and normal incidence will have the lowest reflection, while grazing incidence will result in total reflection.

Consider now a wave with a complex wave number entering a PML. The complex wave number, β, is on the form

$$\beta_z = i\alpha_z + k_z, \tag{2.162}$$

such that the propagation term of the wave is on the form

$$e^{i\beta_z z} = e^{ik_z z}e^{-\alpha_z z}, \text{ for } z < z_0. \tag{2.163}$$

By assuming that $s_{z,1} = 1$, and choosing that $s_z = s_{z,2}$, the propagation term inside the PML will then be on the form

$$e^{i\beta_z z} = e^{is_z k_z z}e^{-s_z\alpha_z z}, \text{ for } z > z_0, \tag{2.164}$$

where it has been used that the wave number inside the PML is given as the the wave number of the wave entering the PML times the ratio of the respective stretched coordinate variable, i.e. $k_{z,2} = s_z k_{z,1}$. Substituting the stretched coordinate variable with that of the CFS-PML, eq. (2.65), gives the propagation term as

$$e^{i\beta_z z} = e^{i\kappa_z k_z z}e^{-\kappa_z\alpha_z z}e^{i\frac{\sigma_z}{a_z+i\omega\epsilon_0}k_z z}e^{-\frac{\sigma_z}{a_z+i\omega\epsilon_0}\alpha_z z}, \text{ for } z > z_0. \tag{2.165}$$

From the above propagation term, it can be concluded that, if $\omega \gg a_z/\epsilon_0$, then the propagating wave will not attenuate, and if $\omega \ll a_z/\epsilon_0$, then the attenuation of the evanescent wave might become too high for the wave to be properly sampled and cause a reflection. As such, a_z should be chosen with the simulated frequencies in mind.

While a perfectly matched layer is indeed theoretically reflectionless, numerical reflection arises from the staggered grid of the FDTD method. This is a phenomenon caused by the spatial spreading of the Electric and magnetic field components. At the interface between the simulation domain and the PML, only one of the fields are being calculated, and therefore only one of the stretched coordinate variables of the PML is being incorporated in the calculation of the fields. [29]

Assuming the interface lies at the electric field sampling points, the reflection error from the interface, eq. (2.159), becomes

$$\Gamma_0 = \left| \frac{1 - \sqrt{s_{z,2}}}{1 - \sqrt{s_{z,2}}} \right|, \text{ for } z = z_0, \tag{2.166}$$

where it has been used that $s_{z,1} = s_{z,1}^* = s_{z,2}^* = 1$ for $z = z_0$. Inserting the CFS-PML parameter, the reflection error becomes

$$\Gamma_0 = \left| \frac{1 - \sqrt{\kappa_z + \frac{\sigma_z}{a_z + i\omega\epsilon_0}}}{1 + \sqrt{\kappa_z + \frac{\sigma_z}{a_z + i\omega\epsilon_0}}} \right|, \text{ for } z = z_0. \tag{2.167}$$

From eq. (2.165), it can be inferred that κ_z, σ_z, and a_z should all be nonzero and positive to ensure a forward-propagating wave as well as to ensure that the evanescent terms do not become exponentially increasing. As such, to achieve a low reflection error, the immediate idea would be to choose $\kappa_z = 1$ and $\omega \gg a_z/\epsilon_0$, however, as mentioned previously, the latter choice would result in the propagating wave to not be attenuated in the PML. To combat this issue, the constitutive parameters may be scaled polynomially as one-dimensional functions as [30]

$$\kappa_z(z) = \begin{cases} 1 + \left(\kappa_z^{(\max)} - 1\right)\left(\frac{|z - z_0|}{d}\right)^m, & z_0 \le z \le z_0 + d, \\ 1, & \text{else}, \end{cases} \tag{2.168}$$

$$\sigma_z(z) = \begin{cases} \sigma_z^{(\max)}\left(\frac{|z - z_0|}{d}\right)^m, & z_0 \le z \le z_0 + d, \\ 0, & \text{else}, \end{cases} \tag{2.169}$$

$$a_z(z) = \begin{cases} a_z^{(\max)}\left(\frac{|d + z_0 - z|}{d}\right)^{m_a}, & z_0 \le z \le z_0 + d, \\ 0, & \text{else}. \end{cases} \tag{2.170}$$

Thus, κ_z and σ_z are m'th polynomials going from unity and zero to $\kappa_z^{(\max)}$ and $\sigma_z^{(\max)}$, respectively. In contrast, a_z is an m_a'th polynomial going from $a_z^{(\max)}$ to zero. This is important, as a_z must be small to make sure low-frequency propagating waves are attenuated in the PML, however, for a low reflection error at the interface, a_z must be large.

This scaling of the constitutive parameters, of course, changes how a wave propagates in the PML and therefore also the reflection error. To examine this, assume that the PML is split into N layers, each with constant constitutive parameters, such that for the k'th layer of the PML,

the constitutive parameters are given by

$$\kappa_z\big|_k = \kappa_z(z_k), \tag{2.171}$$

$$\sigma_z\big|_k = \sigma_z(z_k), \tag{2.172}$$

$$a_z\big|_k = a_z(z_k), \tag{2.173}$$

where z_k is in the middle of the k'th layer. By assuming the interface reflection errors caused by the staggered grid are negligible, the reflection coefficients over the entire PML may be found similarly to eq. (2.160) as

$$R(\theta) = \pm \exp\left[-2i\omega\sqrt{\epsilon\mu}\,\delta z \cos\theta \sum_{k=1}^{N} \kappa_z\big|_k + \frac{\sigma_z\big|_k}{a_z\big|_k + i\omega\epsilon_0}\right]. \tag{2.174}$$

Separating complex and real exponentials gives

$$\begin{aligned} R(\theta) = \pm \exp&\left[-2i\omega\sqrt{\epsilon\mu}\cos\theta\delta z \sum_{k=1}^{N}\left(\kappa_z\big|_k + \frac{\sigma_z\big|_k a_z\big|_k}{\left(a_z\big|_k\right)^2 + \omega^2\epsilon_0^2}\right)\right] \\ &\times \exp\left[-2\omega^3\epsilon_0\sqrt{\epsilon\mu}\,\delta z\cos\theta\sum_{k=1}^{N}\frac{\sigma_z\big|_k}{\left(a_z\big|_k\right)^2 + \omega^2\epsilon_0^2}\right], \end{aligned} \tag{2.175}$$

the magnitude of which is given by the latter exponential as

$$\Gamma(\theta) = \exp\left[-2\omega^2\epsilon_0\sqrt{\epsilon\mu}\,\delta z\cos\theta\sum_{k=1}^{N}\frac{\sigma_z\big|_k}{\left(a_z\big|_k\right)^2 + \omega^2\epsilon_0^2}\right], \tag{2.176}$$

which is the reflection error over the entire PML. While numerical reflection will indeed occur at each interface between the different PML-sections, these may be considered insignificant [30].

The choice of parameters should be made with the condition, $\omega \ll a_z/\epsilon_0$, in mind, while also making sure that the fields are properly sampled. However, it has been empirically found that the optimal $\sigma_z^{(\max)}$ is given by [30]

$$\sigma_z^{(\text{opt})} = \frac{0.8(m+1)}{\eta_0\delta z\sqrt{\epsilon_{\text{eff}}\mu_{\text{eff}}}}, \tag{2.177}$$

where η_0 is the vacuum impedance given by

$$\eta_0 = \sqrt{\frac{\mu_0}{\epsilon_0}}, \tag{2.178}$$

and ϵ_{eff} and μ_{eff} are the effective relative permittivity and permeability given by

$$\epsilon_{\text{eff}} = \frac{1}{V_{\text{PML}}}\int_{V_{\text{PML}}}\epsilon_r(x,y,z)d^3r, \tag{2.179}$$

$$\mu_{\text{eff}} = \frac{1}{V_{\text{PML}}}\int_{V_{\text{PML}}}\mu_r(x,y,z)d^3r, \tag{2.180}$$

where V_{PML} is the volume of the PML. The effective relative permittivity and permeability are in other words given by the average permittivity and permeability, respectively, of the PML. Finally, it is noted that the polynomial scaling, m, is often found to be optimal in the range $3 \leq m \leq 4$. [30] As a final note: While this section has focused on a single PML in the (x,y)-plane, all principles and equations are directly transferable to PMLs in the (x,z)- and (y,z)-planes.

2.5 Conforming to Curved Metallic Surfaces

The simplest way to model a curved conducting surface in the FDTD is to use the staircase method, in which the conducting structure is conformed to the grid, such that the surface lies directly on the staggered grid points. However, this of course means that the structure is not accurately portrayed in the simulation and errors as well as spurious solutions may occur [31]. The alternative is to conform the grid to the structure instead. One such approach is the so-called contour path method [32, 33]. The contour path method conforms the contours of Faraday's and Ampére's laws, i.e. the path integrals of the integral forms of the Maxwell's equations, to the surface, and subsequently changes the update equations accordingly. A simpler method is the so-called conformal method [34, 35]. This method has been shown to have better stability when compared to the contour path method [36]. As such, the conformal method will be used in this project, and the rest of this section will be based on [37].

Consider a staggered grid in which a curved PEC surface is placed, as illustrated in fig. 2.4.

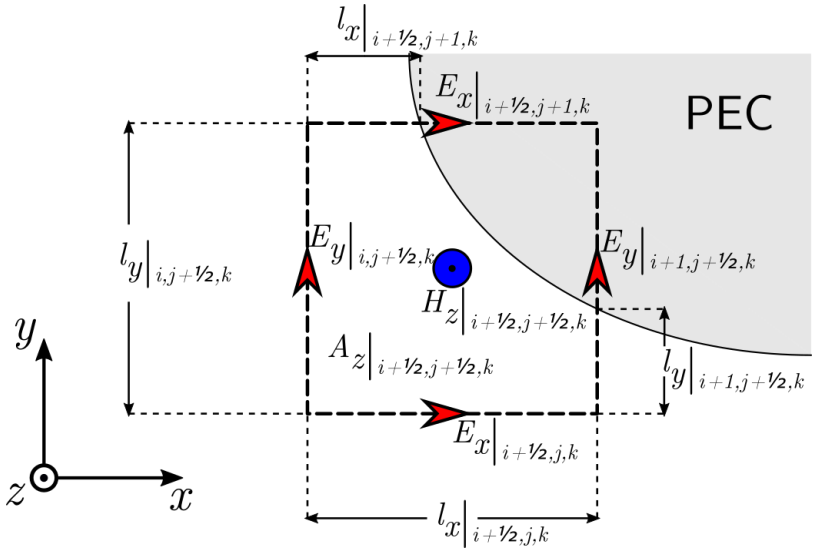


Figure 2.4: Illustration of a grid cell the the curved surface of a PEC.

Faraday's law states that

$$\oint_{\partial S} \mathbf{E} d\mathbf{l} = -\int_S \frac{\partial \mu \mathbf{H}}{\partial t} d\mathbf{A}, \tag{2.181}$$

where ∂S is the path enclosing the surface S, $\mathbf{l}$ is parallel to the surface, and $\mathbf{A}$ is normal to the surface. In the standard Yee-algorithm, the integral on the left-hand side follows a path along the rectangular grid, and the integral on the right-hand side is over the entire area enclosed by said path. In the conformal method, however, the path is shortened by following along the surface rather than going into the PEC, as illustrated in fig. 2.5.

It is assumed that the electric field is constant along each line in the path, i.e. E_x is constant along the path from (i, j, k) to $(i + 1, j, k)$ and also constant along the path from $(i, j + 1, k)$ to $(i + 1, j + 1, k)$, and E_y is constant along the path from (i, j, k) to $(i + 1, j, k)$. Furthermore, it

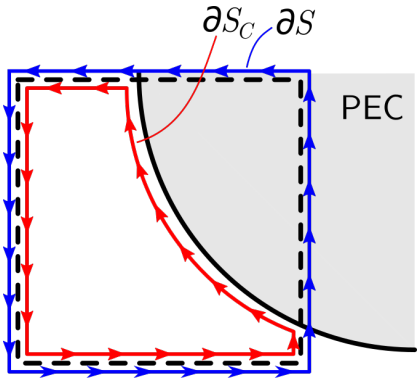


Figure 2.5: Illustration of how the grid is conformed to a curved PEC surface. ∂S is the path of the traditional Yee-grid, and ∂S_C is the path of the conformed grid.

is assumed that the magnetic field is constant in the enclosed area outside the PEC. Faraday's law then becomes

$$-\frac{\partial \mu H_z}{\partial t}\bigg|_{i+1/2,j+1/2,k} A_z|_{i+1/2,j+1/2,k} = E_x|_{i+1/2,j,k} l_x|_{i+1/2,j,k} + E_y|_{i,j+1/2,k} l_y|_{i,j+1/2,k} - E_x|_{i+1/2,j+1,k} l_x|_{i+1/2,j+1,k} - E_y|_{i+1,j+1/2,k} l_y|_{i+1,j+1/2,k}. \tag{2.182}$$

Note that the integral along the PEC surface is zero as an electric field tangential to a PEC-surface is zero at the interface. By applying the central difference approximation to the time-derivative, the update equation for H_z becomes

$$H_z|^{n+1}_{i+1/2,j+1/2,k} = H_z|^{n}_{i+1/2,j+1/2,k} + \frac{\delta t}{\mu|_{i+1/2,j+1/2,k} A_z|_{i+1/2,j+1/2,k}} \times \left(E_x|^{n+1/2}_{i+1/2,j+1,k} l_x|_{i+1/2,j+1,k} - E_x|^{n+1/2}_{i+1/2,j,k} l_x|_{i+1/2,j,k} - E_y|^{n+1/2}_{i+1,j+1/2,k} l_y|_{i+1,j+1/2,k} - E_y|^{n+1/2}_{i,j+1/2,k} l_y|_{i,j+1/2,k} \right). \tag{2.183}$$

It is convenient to normalise the lengths and areas as

$$\ell_x = \frac{l_x}{\delta x}, \tag{2.184}$$

$$\ell_y = \frac{l_y}{\delta y}, \tag{2.185}$$

$$\ell_z = \frac{l_z}{\delta z}, \tag{2.186}$$

$$S_x = \frac{A_x}{\delta y \delta z}, \tag{2.187}$$

$$S_y = \frac{A_y}{\delta x \delta z}, \tag{2.188}$$

$$S_z = \frac{A_z}{\delta x \delta y}, \tag{2.189}$$

such that eq. (2.183) becomes

$$H_z\big|_{i+1/2,j+1/2,k}^{n+1} = H_z\big|_{i+1/2,j+1/2,k}^{n} + \frac{\delta t}{\mu\big|_{i+1/2,j+1/2,k} S_z\big|_{i+1/2,j+1/2,k}} \times \left(\frac{E_x\big|_{i+1/2,j+1,k}^{n+1/2} \ell_x\big|_{i+1/2,j+1,k} - E_x\big|_{i+1/2,j,k}^{n+1/2} \ell_x\big|_{i+1/2,j,k}}{\delta y} - \frac{E_y\big|_{i+1,j+1/2,k}^{n+1/2} \ell_y\big|_{i+1,j+1/2,k} - E_y\big|_{i,j+1/2,k}^{n+1/2} \ell_y\big|_{i,j+1/2,k}}{\delta x} \right). \tag{2.190}$$

By scaling the electric field as

$$\breve{E}_x = E_x \ell_x, \tag{2.191}$$
$$\breve{E}_y = E_y \ell_y, \tag{2.192}$$
$$\breve{E}_z = E_z \ell_z, \tag{2.193}$$

eq. (2.190) becomes

$$H_z\big|_{i+1/2,j+1/2,k}^{n+1} = H_z\big|_{i+1/2,j+1/2,k}^{n} + \frac{\delta t}{\mu\big|_{i+1/2,j+1/2,k} S_z\big|_{i+1/2,j+1/2,k}} \left(\frac{\breve{E}_x\big|_{i+1/2,j+1,k}^{n+1/2} - \breve{E}_x\big|_{i+1/2,j,k}^{n+1/2}}{\delta y} - \frac{\breve{E}_y\big|_{i+1,j+1/2,k}^{n+1/2} - \breve{E}_y\big|_{i,j+1/2,k}^{n+1/2}}{\delta x} \right). \tag{2.194}$$

Similarly, update equations can be found for H_x and H_y. The scaling of the electric fields also causes a change in the electric field update equation, e.g. E_x becomes

$$\breve{E}_x\big|_{i+1/2,j,k}^{n+1/2} = \breve{E}_x\big|_{i+1/2,j,k}^{n-1/2} + \frac{\delta t \ell_x}{\epsilon\big|_{i+1/2,j,k}} \left(\frac{H_z\big|_{i+1/2,j+1/2,k}^{n} - H_z\big|_{i+1/2,j-1/2,k}^{n}}{\delta y} - \frac{H_y\big|_{i+1/2,j,k+1/2}^{n} - H_y\big|_{i+1/2,j,k-1/2}^{n}}{\delta z} \right). \tag{2.195}$$

Finally, the permittivities and permeabilities are also scaled as

$$\breve{\epsilon}_x = \frac{\epsilon}{\ell_x}, \tag{2.196}$$
$$\breve{\epsilon}_y = \frac{\epsilon}{\ell_y}, \tag{2.197}$$
$$\breve{\epsilon}_z = \frac{\epsilon}{\ell_z}, \tag{2.198}$$
$$\breve{\mu}_x = \mu S_x, \tag{2.199}$$
$$\breve{\mu}_y = \mu S_y, \tag{2.200}$$
$$\breve{\mu}_z = \mu S_z, \tag{2.201}$$

such that the field update equations become

$$\breve{E}_x\big|_{i+1/2,j,k}^{n+1/2} = \breve{E}_x\big|_{i+1/2,j,k}^{n-1/2} + \frac{\delta t}{\breve{\epsilon}\big|_{i+1/2,j,k}} \left(\frac{H_z\big|_{i+1/2,j+1/2,k}^{n} - H_z\big|_{i+1/2,j-1/2,k}^{n}}{\delta y} - \frac{H_y\big|_{i+1/2,j,k+1/2}^{n} - H_y\big|_{i+1/2,j,k-1/2}^{n}}{\delta z} \right), \tag{2.202}$$

$$\breve{E}_y\big|_{i,j+1/2,k}^{n+1/2} = \breve{E}_y\big|_{i,j+1/2,k}^{n-1/2} + \frac{\delta t}{\breve{\epsilon}\big|_{i,j+1/2,k}} \left(\frac{H_x\big|_{i,j+1/2,k+1/2}^{n} - H_x\big|_{i,j+1/2,k-1/2}^{n}}{\delta z} - \frac{H_z\big|_{i+1/2,j+1/2,k}^{n} - H_z\big|_{i-1/2,j+1/2,k}^{n}}{\delta x} \right), \tag{2.203}$$

$$\breve{E}_z\big|_{i,j,k+1/2}^{n+1/2} = \breve{E}_z\big|_{i,j,k+1/2}^{n-1/2} + \frac{\delta t}{\breve{\epsilon}\big|_{i,j,k+1/2}} \left(\frac{H_y\big|_{i+1/2,j,k+1/2}^{n} - H_y\big|_{i-1/2,j,k+1/2}^{n}}{\delta x} - \frac{H_x\big|_{i,j+1/2,k+1/2}^{n} - H_x\big|_{i,j-1/2,k+1/2}^{n}}{\delta y} \right), \tag{2.204}$$

$$H_x\big|_{i,j+1/2,k+1/2}^{n+1} = H_x\big|_{i,j+1/2,k+1/2}^{n} + \frac{\delta t}{\breve{\mu}\big|_{i,j+1/2,k+1/2}} \left(\frac{\breve{E}_y\big|_{i,j+1/2,k+1}^{n+1/2} - \breve{E}_y\big|_{i,j+1/2,k}^{n+1/2}}{\delta z} - \frac{\breve{E}_z\big|_{i,j+1,k+1/2}^{n+1/2} - \breve{E}_z\big|_{i,j,k+1/2}^{n+1/2}}{\delta y} \right), \tag{2.205}$$

$$H_y\big|_{i+1/2,j,k+1/2}^{n+1} = H_y\big|_{i+1/2,j,k+1/2}^{n} + \frac{\delta t}{\breve{\mu}\big|_{i+1/2,j,k+1/2}} \left(\frac{\breve{E}_z\big|_{i+1,j,k+1/2}^{n+1/2} - \breve{E}_z\big|_{i,j,k+1/2}^{n+1/2}}{\delta x} - \frac{\breve{E}_x\big|_{i+1/2,j,k+1}^{n+1/2} - \breve{E}_x\big|_{i+1/2,j,k}^{n+1/2}}{\delta z} \right), \tag{2.206}$$

$$H_z\big|_{i+1/2,j+1/2,k}^{n+1} = H_z\big|_{i+1/2,j+1/2,k}^{n} + \frac{\delta t}{\breve{\mu}\big|_{i+1/2,j+1/2,k}} \left(\frac{\breve{E}_x\big|_{i+1/2,j+1,k}^{n+1/2} - \breve{E}_x\big|_{i+1/2,j,k}^{n+1/2}}{\delta y} - \frac{\breve{E}_y\big|_{i+1,j+1/2,k}^{n+1/2} - \breve{E}_y\big|_{i,j+1/2,k}^{n+1/2}}{\delta x} \right). \tag{2.207}$$

Thus, to implement the conformal FDTD method, only the material parameters need to be scaled to take the PEC surface into account during time-stepping. However, this scaling of

parameters has implications on stability so that the CFL condition becomes

$$\delta t \leq \frac{1}{\breve{c}_{\max}\sqrt{\frac{1}{\delta x^2}+\frac{1}{\delta y^2}+\frac{1}{\delta z^2}}}, \tag{2.208}$$

where

$$\breve{c}_{\max} = \max\left(\frac{1}{\sqrt{\breve{\epsilon}\breve{\mu}}}\right). \tag{2.209}$$

2.6 Source Excitation and Field Detection

Several excitation schemes exist for the FDTD method. It is common to excite the electric field by injecting a current source or a voltage source. For scattering problems where a plane-wave excitation is necessary, one would typically choose to utilise the so-called total-field/scattered-field formulation. [37]

2.6.1 Plane-Wave Injector

In this project, a plane-wave is injected into the simulation using the same injection scheme used in the total-field/scattered-field formulation. In this formulation, each field is decomposed into a scattered field and an incident field as [37]

$$\mathbf{E}^{\text{tot}} = \mathbf{E}^{\text{scat}} + \mathbf{E}^{\text{inc}}, \tag{2.210}$$

$$\mathbf{H}^{\text{tot}} = \mathbf{H}^{\text{scat}} + \mathbf{H}^{\text{inc}}. \tag{2.211}$$

The entire computation domain is separated into two virtual regions: The total-field region and the scattered-field region denoted by Ω^{tot} and Ω^{scat}, respectively. The total fields are considered in the total-field region, and the scattered fields are considered in the scattered-field region. In the general total-field/scattered-field formulation, the total-field region is encapsulated by the scattered-field region, however, in this project, the scattered-field region is of little to no interest, and the scattered-field region is consequently chosen to overlap the PML-region on the side from which the plane-wave is injected. This is illustrated in fig. 2.6.

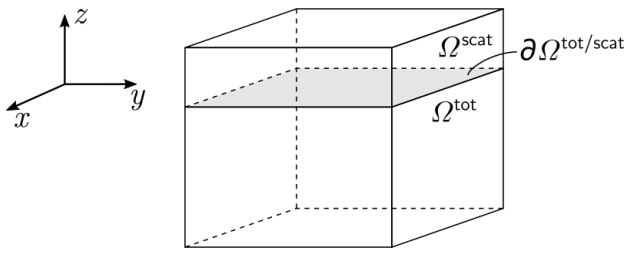


Figure 2.6: Illustration of the total-field/scattered-field setup. Ω^{tot} and Ω^{scat} are the total-field and scattered-field regions, respectively. $\partial\Omega^{\text{tot/scat}}$ is the interface between the regions.

The interface between the total field and the scattered field, $\partial\Omega^{\text{tot/scat}}$, is assumed to lie directly on $k = k^{\text{src}}$, and field components that lie on this interface are assumed to belong to the total-field region. Consider now E_x at $k = k^{\text{src}}$ and H_y at $k = k^{\text{src}} + 1/2$ as illustrated in fig. 2.7.

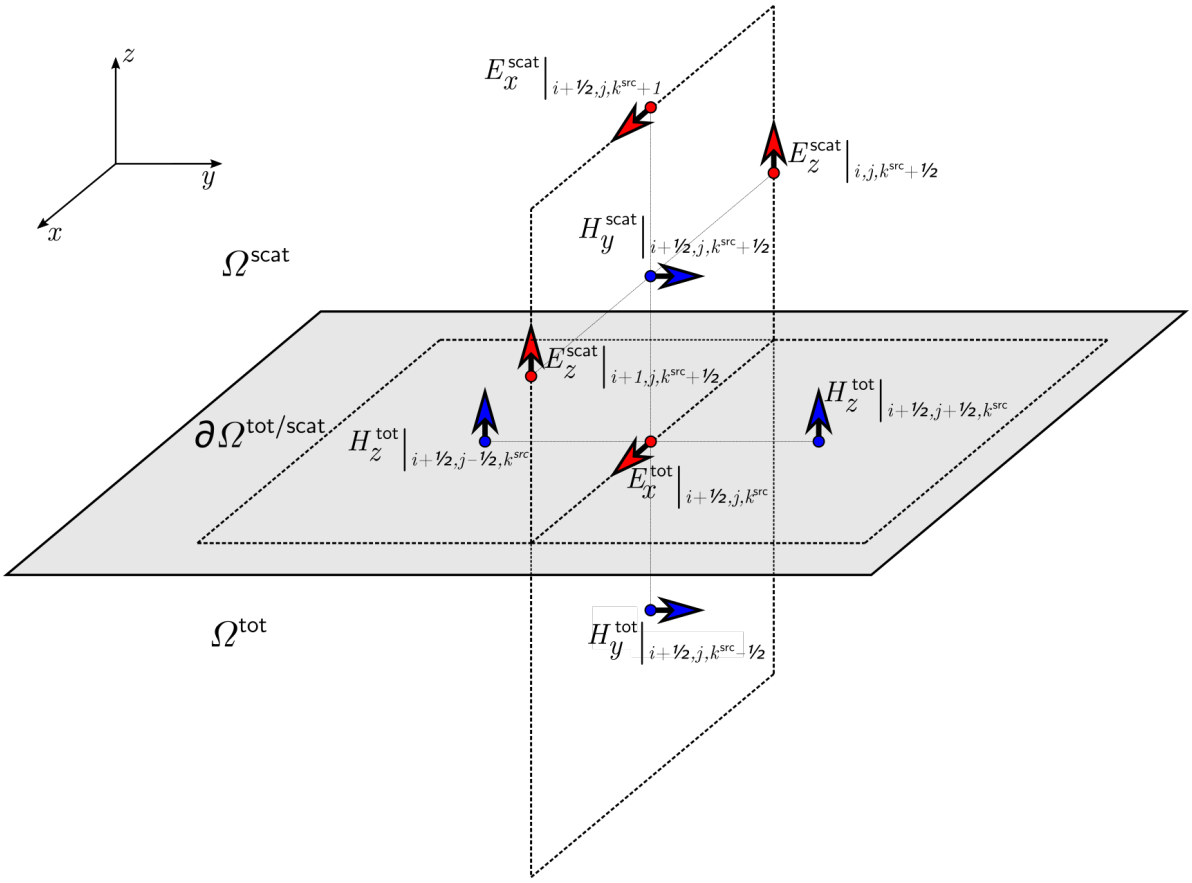


Figure 2.7: Illustration of field component placements near the total-field/scattered-field interface, $\partial\Omega^{\text{tot/scat}}$.

Following the procedure for deriving the field update equations in the Yee-algorithm gives

$$E_x^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2} = E_x^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i+1/2,j,k^{\text{src}}}} \left(\frac{H_z^{\text{tot}}\big|_{i+1/2,j+1/2,k^{\text{src}}}^{n} - H_z^{\text{tot}}\big|_{i+1/2,j-1/2,k^{\text{src}}}^{n}}{\delta y} - \frac{H_y^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} - H_y^{\text{tot}}\big|_{i,j+1/2,k^{\text{src}}-1/2}^{n}}{\delta z} \right), \tag{2.212}$$

and

$$H_y^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n+1} = H_y^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} + \frac{\delta t}{\mu\big|_{i+1/2,j,k^{\text{src}}+1/2}} \left(\frac{E_z^{\text{scat}}\big|_{i+1,j,k^{\text{src}}+1/2}^{n+1/2} - E_z^{\text{scat}}\big|_{i,j,k^{\text{src}}+1/2}^{n+1/2}}{\delta x} - \frac{E_x^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1}^{n+1/2} - E_x^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2}}{\delta z} \right). \tag{2.213}$$

Recall that the total fields are considered when $k \leq k^{\text{src}}$, and the scattered fields are considered when $k > k^{\text{src}}$, such that

$$H_y^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} = H_y^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} + H_y^{\text{inc}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n}, \tag{2.214}$$

and

$$E_x^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2} = E_x^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2} - E_x^{\text{inc}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2}, \tag{2.215}$$

which follows from eqs. (2.210) and (2.211). Substituting eqs. (2.214) and (2.215) into eqs. (2.212) and (2.213), respectively, gives

$$\begin{aligned} E_x^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2} = E_x^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i+1/2,j,k^{\text{src}}}} \Bigg(& \frac{H_z^{\text{tot}}\big|_{i+1/2,j+1/2,k^{\text{src}}}^{n} - H_z^{\text{tot}}\big|_{i+1/2,j-1/2,k^{\text{src}}}^{n}}{\delta y} \\ & - \frac{H_y^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} + H_y^{\text{inc}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} - H_y^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}-1/2}^{n}}{\delta z} \Bigg), \end{aligned} \tag{2.216}$$

and

$$\begin{aligned} H_y^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n+1} = H_y^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} + \frac{\delta t}{\mu\big|_{i+1/2,j,k^{\text{src}}+1/2}} \Bigg(& \frac{E_z^{\text{scat}}\big|_{i+1,j,k^{\text{src}}+1/2}^{n+1/2} - E_z^{\text{scat}}\big|_{i,j,k^{\text{src}}+1/2}^{n+1/2}}{\delta x} \\ & - \frac{E_x^{\text{scat}}\big|_{i+1/2,j,k^{\text{src}}+1}^{n+1/2} - E_x^{\text{tot}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2} + E_x^{\text{inc}}\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2}}{\delta z} \Bigg). \end{aligned} \tag{2.217}$$

By following the same procedure for E_y at $k = k^{\text{src}}$ and H_x at $k = k^{\text{src}} + 1/2$, the update equations become

$$\begin{aligned} E_x\big|_{i+1/2,j,k^{\text{src}}}^{n+1/2} = E_x\big|_{i+1/2,j,k^{\text{src}}}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i+1/2,j,k^{\text{src}}}} \Bigg(& \frac{H_z\big|_{i+1/2,j+1/2,k^{\text{src}}}^{n} - H_z\big|_{i+1/2,j-1/2,k^{\text{src}}}^{n}}{\delta y} \\ & - \frac{H_y\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n} - H_y\big|_{i+1/2,j,k^{\text{src}}-1/2}^{n}}{\delta z} \Bigg) + \frac{\delta t}{\delta z \epsilon\big|_{i+1/2,j,k^{\text{src}}}} H_y^{\text{inc}}\big|_{i+1/2,j,k^{\text{src}}+1/2}^{n}, \end{aligned} \tag{2.218}$$

$$\begin{aligned} E_y\big|_{i,j+1/2,k^{\text{src}}}^{n+1/2} = E_y\big|_{i,j+1/2,k^{\text{src}}}^{n-1/2} + \frac{\delta t}{\epsilon\big|_{i,j+1/2,k^{\text{src}}}} \Bigg(& \frac{H_x\big|_{i,j+1/2,k^{\text{src}}+1/2}^{n} - H_x\big|_{i,j+1/2,k^{\text{src}}-1/2}^{n}}{\delta z} \\ & - \frac{H_z\big|_{i+1/2,j+1/2,k^{\text{src}}}^{n} - H_z\big|_{i-1/2,j+1/2,k^{\text{src}}}^{n}}{\delta x} \Bigg) - \frac{\delta t}{\delta z \epsilon\big|_{i,j+1/2,k^{\text{src}}}} H_x^{\text{inc}}\big|_{i,j+1/2,k^{\text{src}}+1/2}^{n}, \end{aligned} \tag{2.219}$$

$$\begin{aligned} H_x\big|_{i,j+1/2,k^{\text{src}}+1/2}^{n+1} = H_x\big|_{i,j+1/2,k^{\text{src}}+1/2}^{n} + \frac{\delta t}{\mu\big|_{i,j+1/2,k^{\text{src}}+1/2}} \Bigg(& \frac{E_y\big|_{i,j+1/2,k^{\text{src}}+1}^{n+1/2} - E_y\big|_{i,j+1/2,k^{\text{src}}}^{n+1/2}}{\delta z} \\ & - \frac{E_z\big|_{i,j+1,k^{\text{src}}+1/2}^{n+1/2} - E_z\big|_{i,j,k^{\text{src}}+1/2}^{n+1/2}}{\delta y} \Bigg) + \frac{\delta t}{\delta z \mu\big|_{i,j+1/2,k^{\text{src}}+1/2}} E_y^{\text{inc}}\big|_{i,j+1/2,k^{\text{src}}+1}^{n+1/2}, \end{aligned} \tag{2.220}$$

$$H_y\big|^{n+1}_{i+1/2,j,k^{\mathrm{src}}+1/2} = H_y\big|^{n}_{i+1/2,j,k^{\mathrm{src}}+1/2} + \frac{\delta t}{\mu\big|_{i+1/2,j,k^{\mathrm{src}}+1/2}} \left(\frac{E_z\big|^{n+1/2}_{i+1,j,k^{\mathrm{src}}+1/2} - E_z\big|^{n+1/2}_{i,j,k^{\mathrm{src}}+1/2}}{\delta x} - \frac{E_x\big|^{n+1/2}_{i+1/2,j,k^{\mathrm{src}}+1} - E_x\big|^{n+1/2}_{i+1/2,j,k^{\mathrm{src}}}}{\delta z} \right) - \frac{\delta t}{\delta z \mu\big|_{i+1/2,j,k^{\mathrm{src}}+1/2}} E_x^{\mathrm{inc}}\big|^{n+1/2}_{i+1/2,j,k^{\mathrm{src}}}, \tag{2.221}$$

where it has been used that

$$\mathbf{E} = \begin{cases} \mathbf{E}^{\mathrm{tot}}, & k \leq k^{\mathrm{src}}, \\ \mathbf{E}^{\mathrm{scat}}, & k > k^{\mathrm{src}}, \end{cases} \tag{2.222}$$

$$\mathbf{H} = \begin{cases} \mathbf{H}^{\mathrm{tot}}, & k \leq k^{\mathrm{src}}, \\ \mathbf{H}^{\mathrm{scat}}, & k > k^{\mathrm{src}}. \end{cases} \tag{2.223}$$

2.6.2 Source Signature

The plane-wave injector of the previous section describes the spatial distribution of the incident field as a plane-wave that is constant along x and y, and that propagates along z. However, since the FDTD method is also a temporally resolved simulation, the temporal distribution, or time signature, of the incident field must also be determined. [37]

The most common time signature is the Gaussian pulse. However, the Blackman-Harris window is also broadly used due to its high-frequency behaviour. Furthermore, depending on the application, these may be modified either by using their respective derivatives or by modulating the amplitude with a sinusoidal function. This project mainly utilises a modulated Gaussian pulse. The Gaussian pule is given by

$$g(t) = e^{-\frac{(t-t_0)^2}{\tau^2}}, \tag{2.224}$$

where t_0 is the centre of the pulse, also called the time delay, and τ is the half-width of the pulse. The amplitude is then modulated to give

$$m_g(t) = \sin(2\pi f_m t) g(t), \tag{2.225}$$

where f_m is the modulating frequency, which determines the centre-frequency.

If a single frequency is desired to be simulated a sinusoidal steady-state source is used. However, a sinusoidal source cannot simply be directly applied, since instantaneously turning on the sinusoidal source would cause a slope discontinuity. This, in turn, causes the frequency spectrum to become unbounded which cannot be accurately simulated using a discrete simulated scheme. The problem can, however, easily be solved by slowly turning on the sinusoidal source using a Gaussian function

$$s_g(t) = \begin{cases} 0, & t < 0, \\ g(t)\sin(2\pi f_s t), & 0 \leq t \leq t_0, \\ \sin(2\pi f_s t), & t_0 < t, \end{cases} \tag{2.226}$$

where f_s is the steady-state frequency.

As mentioned above, the choice of source signature in this project is mainly the modulated Gaussian pulse. The general goal of these simulations is to provide transmittance spectra of the examined structures in order to determine resonance frequencies, and the examined spectrum is limited to the terahertz regime. As such, a modulated pulse is preferred to limit the number of low-frequency components. Another argument for this choice is made by referring back to the PML: The interaction time of a propagating wave and the PML is inversely proportional to the wavelength, or proportional to the frequency, i.e. a propagating wave with a lower frequency will be less attenuated throughout the PML relative to a propagating wave with a higher frequency. Consequently, the reflection error will be lowered by limiting the number of low-frequency components.

2.6.3 Field Detection

This section will describe how data is extracted from the FDTD simulations performed in this project. The first thing to consider is what information is desired. The needed information for creating transmission spectra is the incident power on the structures and the transmitted power through the structure. This section will go back and forth between the time-domain and the frequency-domain, so to avoid confusion, functions are explicitly expressed in terms of their dependence, e.g. $\mathbf{E} = \mathbf{E}(\mathbf{r}, t)$.

The time-averaged power flow, P_m, through a surface, S, is given by

$$P_m(\omega) = \int_S \mathbf{S}_m(\mathbf{r}, \omega) d\mathbf{A}, \tag{2.227}$$

where $\mathbf{A}$ is normal on the surface, and $\mathbf{S}$ is the Poynting vector given by [12]

$$\mathbf{S}_m(\mathbf{r}, \omega) = \frac{1}{2}\mathrm{Re}\left\{\tilde{\mathbf{E}}(\mathbf{r}, \omega) \times \tilde{\mathbf{H}}^*(\mathbf{r}, \omega)\right\}, \tag{2.228}$$

where * denotes the complex conjugate. The surface, through which the power flow is measured, lies in the (x, y) plane. Therefore, the outgoing time-averaged power flow is given by

$$P_m^{\mathrm{out}}(\omega) = \int_0^{\Lambda_y} \int_0^{\Lambda_x} \mathbf{S}_m(\mathbf{r}, \omega) \cdot \hat{\mathbf{z}} dx dy. \tag{2.229}$$

The time-harmonic electric and magnetic fields are given by the inverse Fourier transforms

$$\tilde{\mathbf{E}}(\mathbf{r}, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathbf{E}(\mathbf{r}, t) e^{i\omega t} d\omega, \tag{2.230}$$

$$\tilde{\mathbf{H}}(\mathbf{r}, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathbf{H}(\mathbf{r}, t) e^{i\omega t} d\omega. \tag{2.231}$$

The transmittance is found by the ratio of the outgoing power and the incident power as

$$T(\omega) = \frac{P_m^{\mathrm{out}}(\omega)}{P_m^{\mathrm{inc}}(\omega)}, \tag{2.232}$$

where P_m^{inc} is the time-averaged power flow that is incident on the structure, and P_m^{out} is the time-averaged power flow on the other side of the structure. While P_m^{inc} is easily found, as the incident fields are predetermined, P_m^{out} would, so far, require that $\mathbf{E}$ and $\mathbf{H}$ are stored for all time steps of the simulation so that they can be Fourier transformed post simulation. If the transmittance through a small area was required, this would be a satisfactory method, however, since the the transmittance through an entire period is needed, the amount of information required to be stored becomes unmanageable. Instead, the spectral information of the fields will be stored using a concept borrowed from the Fourier modal method.

Because the structure is periodic ad infinitum, the fields can be expanded into Bloch modes as [12]

$$\tilde{\mathbf{E}}(\mathbf{r},\omega) = \sum_{p,q} \tilde{\mathbf{E}}_{p,q}(\omega) e^{ipG_x x} e^{iqG_y y} e^{-ik_{z,p,q}(\omega) z}, \tag{2.233}$$

where $\tilde{\mathbf{E}}_{p,q}$ are periodic functions with the same periodicity as the structure, G_x and G_y are given by

$$G_x = \frac{2\pi}{\Lambda_x},\ G_y = \frac{2\pi}{\Lambda_y}, \tag{2.234}$$

where Λ_x and Λ_y are the periods along x and y, respectively, and $k_{z,p,q}$ are the propagation constants given by

$$k_{z,p,q}(\omega) = \sqrt{\omega^2\mu\epsilon - (pG_x)^2 - (qG_y)^2}. \tag{2.235}$$

Consider now eq. (2.233) expressed as

$$\tilde{\mathbf{E}}(\mathbf{r},\omega) = \sum_{p,q} \tilde{\mathbf{f}}_{p,q}(\omega) e^{ipG_x x} e^{iqG_y y}, \tag{2.236}$$

where

$$\tilde{\mathbf{f}}_{p,q}(\omega) = \tilde{\mathbf{E}}_{p,q}(\omega) e^{-ik_{z,p,q}(\omega) z}. \tag{2.237}$$

Fourier transforming both of these gives

$$\mathbf{E}(\mathbf{r},t) = \sum_{p,q} \mathbf{f}_{p,q}(t) e^{ipG_x x} e^{iqG_y y}, \tag{2.238}$$

where

$$\mathbf{f}_{p,q}(t) = \int_{-\infty}^{\infty} \tilde{\mathbf{f}}_{p,q}(\omega) e^{-i\omega t} dt, \tag{2.239}$$

and

$$\tilde{\mathbf{f}}_{p,q}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathbf{f}_{p,q}(t) e^{i\omega t} d\omega. \tag{2.240}$$

As such, the electric field, $\mathbf{E}(\mathbf{r},t)$, is spatially described as a Fourier series with Fourier coefficients, $\mathbf{f}_{p,q}(t)$. It is therefore inferable from Fourier analysis that

$$\mathbf{f}_{p,q}(t) = \frac{1}{\Lambda_x \Lambda_y} \int_0^{\Lambda_y} \int_0^{\Lambda_x} \mathbf{E}(\mathbf{r},t) e^{-ipG_x x} e^{-iqG_y y} dxdy. \tag{2.241}$$

From the source free Maxwell's equations, the magnetic field is given by

$$\tilde{\mathbf{H}}(\mathbf{r},\omega) = \frac{1}{i\omega\mu} \boldsymbol{\nabla} \times \tilde{\mathbf{E}}(\mathbf{r},\omega). \tag{2.242}$$

Substituting eq. (2.236) into eq. (2.242) gives

$$\tilde{\mathbf{H}}(\mathbf{r},\omega) = \frac{1}{\omega\mu} \sum_{p,q} \left(\hat{\mathbf{x}} p G_x + \hat{\mathbf{y}} q G_y - \hat{\mathbf{z}} k_{z,p,q}(\omega)\right) \times \tilde{\mathbf{f}}_{p,q}(\omega) e^{ipG_x x} e^{iqG_y y}. \tag{2.243}$$

The time-averaged Poynting vector is then found by substituting eqs. (2.236) and (2.243) into eq. (2.228) as

$$\mathbf{S}_m(\mathbf{r},\omega) = \mathrm{Re}\left\{ \frac{1}{2\omega\mu} \sum_{p,q} \tilde{\mathbf{f}}_{p,q}(\omega) \times \left[\left(\hat{\mathbf{x}} p G_x + \hat{\mathbf{y}} q G_y - \hat{\mathbf{z}} k^*_{z,p,q}(\omega) \right) \times \tilde{\mathbf{f}}^*_{p,q}(\omega) \right] \right\}. \tag{2.244}$$

Substituting eq. (2.244) into eq. (2.229) then becomes

$$P_m^{\mathrm{out}}(\omega) = \int_0^{\Lambda_y} \int_0^{\Lambda_x} \frac{-1}{2\omega\mu} \sum_{p,q} \left| \tilde{\mathbf{f}}_{p,q}(\omega) \right|^2 \mathrm{Re}\left\{ k^*_{z,p,q}(\omega) \right\} dxdy. \tag{2.245}$$

Notice how the integrand is independent of position, such that

$$P_m^{\mathrm{out}}(\omega) = \frac{-\Lambda_x \Lambda_y}{2\omega\mu} \sum_{p,q} \left| \tilde{\mathbf{f}}_{p,q}(\omega) \right|^2 \mathrm{Re}\left\{ k^*_{z,p,q}(\omega) \right\}. \tag{2.246}$$

Thus, all field components at all time steps do not need to be stored. Instead, what is needed to be stored is $\tilde{\mathbf{f}}_{p,q}(t)$ at all time steps, and only for the relevant diffraction orders, p and q. Since evanescent waves are excluded by taking the real part of the propagation constant, the number of relevant diffraction orders may be limited by some maximum considered frequency, $f_{\max}$, of the incident spectrum, such that

$$f_{\max} < \frac{\sqrt{(pG_x)^2 + (qG_y)^2}}{2\pi\sqrt{\epsilon\mu}}. \tag{2.247}$$

Thus, only combinations of p and q that satisfies this condition needs to be considered. Good choices of $f_{\max}$ for an incident modulated Gaussian pulse include $f_{\max} = f_m + \mathrm{FWTM}$ and $f_{\max} = f_m + \tau^{-1}/2$, where FWTM is the full width at a tenth of maximum. The latter will contain ~98.7% of the signal, which in some cases might be excessive.

In practice, at each time step, $\mathbf{f}_{p,q}(t)$ is calculated according to eq. (2.241). Then, when the simulation has run all time steps, $\tilde{\mathbf{f}}_{p,q}(\omega)$, is calculated by performing a discrete Fourier transformation on $\mathbf{f}_{p,q}(t)$. With this, $P_m^{\mathrm{out}}(\omega)$ is finally calculated according to eq. (2.246).

3 | Manufacturing

3.1 Ultraviolet Direct Write Lithography

The antenna and split-ring resonator arrays were made with UV lithography. Silicon wafers of size 1×1 cm^2 were cleaned with acetone in an ultrasound bath, spin coated with MICROPOSIT™ S1813™ photoresist, and then baked for one minute at 120 °C. After baking, the samples were placed in the UV direct write setup sketched in fig. 3.1. The setup uses a 633 nm HeNe laser to align the tilt, as well as x-, y-, and z-coordinates of the sample stage, as to focus the beam on the sample surface. This is done by comparing the beam waist of the incoming beam from the laser and the reflected beam from the sample. The goal is to make them equal in size on the beam blocker, as the objective lens is infinity corrected, meaning for the sample to be in focus the returning beam must be collimated. Once the sample is in focus the mirrors and the glass slide are removed from the beam path of the 400 nm laser. A computer controls the x- and y-axis of the sample stage, while turning on and off the 400 nm laser, in order to manage which areas of the sample are being exposed. The samples were developed in a MICROPOSIT™ developer for two minutes and then cleaned in Milli-Q water. A 120 nm layer of gold, with a 2 nm layer of titanium below for better adhesion, was deposited by sputtering in a Scientific Vacuum Systems V2400. After which, lift-off was performed by letting the samples soak in acetone for two minutes, and lastly giving them an ultrasound bath for a short time at low intensity.

Carefully measured amounts of sucrose and glucose was deposited on the samples by first creating aqueous solutions of the sugars. The wafers were placed on a hot plate at 50 °C and an amount of the aqueous solution containing the desired mass of sucrose and/or glucose was pipetted onto the wafer. The samples were removed from the hotplate once evaporation of the water was complete.

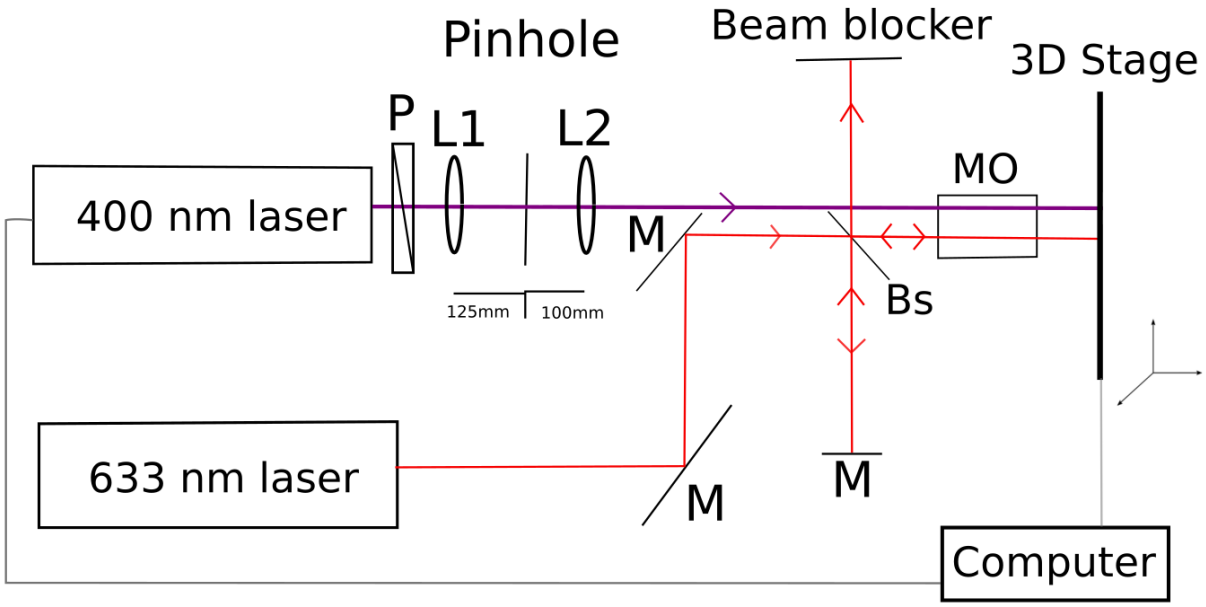


Figure 3.1: Sketch of the UV direct write setup. P: Thin film linear polariser. L1: DCX +125 mm lens. L2: PCX +100 mm lens. Pinhole: 20 µm. M: Mirror. MO: M plan Apo 100x / NA 0.70 ∞/0 f=200 microscope lens. Bs: Glass slide.

3.2 Terahertz Spectroscopy

A MenloSystems TeraSmart Terahertz Time-Domain Spectrometer, modified with an enclosure to add water vapour purging capabilities, was used to measure the terahertz spectra of the samples. The TeraSmart has a spectral range of >5 THz and a dynamic range of >90dB. A sketch of the measurement setup can be seen in fig. 3.2.

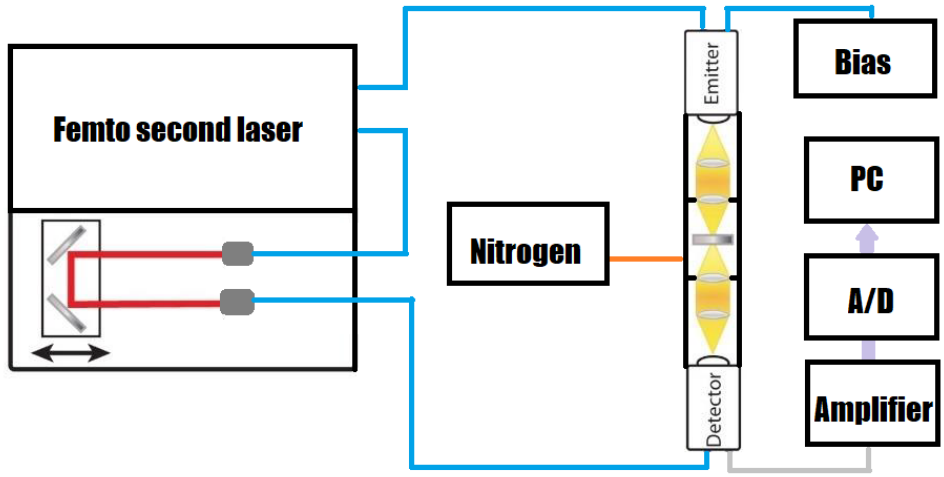


Figure 3.2: Sketch of TeraSmart Terahertz Time-Domain Spectrometer setup, modified with an enclosure around the sample to allow purging of water vapour with nitrogen.

The principle of operation for the TeraSmart is to first create a femtosecond laser pulse and splitting it in two, with one part going to an antenna called the emitter and the other going to an antenna called the detector. The emitter is a semiconductor structure with a low carrier lifetime and a bias over it. When excited, the structure becomes conductive so a pulse generating carriers will have those carriers be accelerated by the bias. This acceleration causes radiation and because of the low carrier lifetime of the structure, the radiation emitted is in the terahertz regime. The terahertz pulse then goes to the detector through a series of lenses and the sample of interest. The lenses are to collect and focus the pulse first onto the sample then the detector. The detector is similar to the emitter in that it has a dipole shape with a gap. When excited by the femtosecond laser pulse the detector will act like a dipole and a current is induced when the terahertz pulse from the emitter is incident. As the detector is blind when not influenced by the laser pulse it can act as a stroboscopic sampling scheme. By controlling the relative travel length of the laser pulse and the terahertz pulse, the detector can be used to sample the terahertz pulse in the time-domain.

For ease of interpretation, the time-domain measurements are transformed into the frequency-domain utilising a fast Fourier algorithm. The frequency-domain spectrum will have an oscillation that is caused by an echo of the terahertz pulse from the raw time-domain data, that in

turn is caused by the internal reflection in the silicon wafer. To avoid this oscillation a Tukey window is used to remove the echo, while keeping the data continuous to avoid an unbounded frequency spectrum.

The spectra achieved with the TeraSmart are constructed by averaging a number of measurements in order to reduce the noise in the final spectrum. Since signal noise gets more pronounced at higher frequencies, a higher number of measurements to average is required to depict them accurately. A different parameter that affects the measurements is the delay interval that is measured over. A large interval gives the best discretisation of the frequency domain. The measurement time scales with both the number of measurements to average over, and with the interval size. In this project, all terahertz spectra are measured with a pulse interval of 200 ps, and the number of measurements to average depends on whether the spectra are to be used for direct comparison or to be used for machine learning. For direct comparison, the numbers of measurements to average are from 100 to 300, and for machine learning the number varies between 1 and 100.

4 | Machine Learning

A successful implementation of machine learning can remove the need for human supervision in many cases of data processing. In order to teach a machine, data is necessary. The more data available, the more accurate the resulting model, at the cost of initial computational power. There are several machine learning methods, which can be used depending on the purpose.

The type of machine learning to use depends on the type of input data and output data. The data can be either numerical or categorical (such as car brands or antenna shapes). This project works with numerical input data and categorical output data, and thus classification machine learning algorithms will be used.

The data is stored in a matrix of size $M \times F$, where M is the number of measurements and F is the number of features, with features being a specific variable. To accompany this, a vector of length M with a label to classify each measurement is created, which are called classification labels, or simply labels.

4.1 Dimensionality Reduction

The dimensionality of data refers to the number of features contained in the data, where a feature can be anything from something generic, such as hair colour, to something specific, such as the amplitude of a THz pulse at 42.33 ps delay.

Often, when working with numerical input data, the dimensionality can be large and therefore computationally intensive. To overcome this complication, dimensionality reduction (DR) can be performed. There are multiple ways to perform this. One way is to perform feature selection on the data, where the most distinct features are kept, and the remaining features are ignored.

4.1.1 Feature Selection

Two methods of feature selection are used in this project: One using chi-square tests and one using a minimum redundancy maximum relevance (MRMR) algorithm. The feature selection method using chi-square tests ranks each feature based on its p-value. The chi-squared test is a statistical hypothesis test - a null hypothesis, which assumes that two variables are independent of each other, with the resulting p-value being the probability of the two variables being independent.

For ease of use, the p-value is converted into a score as $-\ln(p)$, so the smaller the p-value, the higher the score of said feature. Due to a MATLAB limitation, $-\ln(p) = \infty$ for p-values smaller than $2^{-1074} \approx 4.941 \cdot 10^{-324}$, which is the smallest positive nonzero floating-point number for double-precision floating-point values. An example plot showing the distribution of predictor scores for a given machine learning setting can be seen in fig. 4.1. Predictor is another word for feature, which can be used when considering the predictive properties of a feature in regard to the classification. Once the features have been ranked, the most important features can be selected. The number of features to select can vary depending on the circumstances. This feature selection method is performed in MATLAB by using the `fscchi2` function.

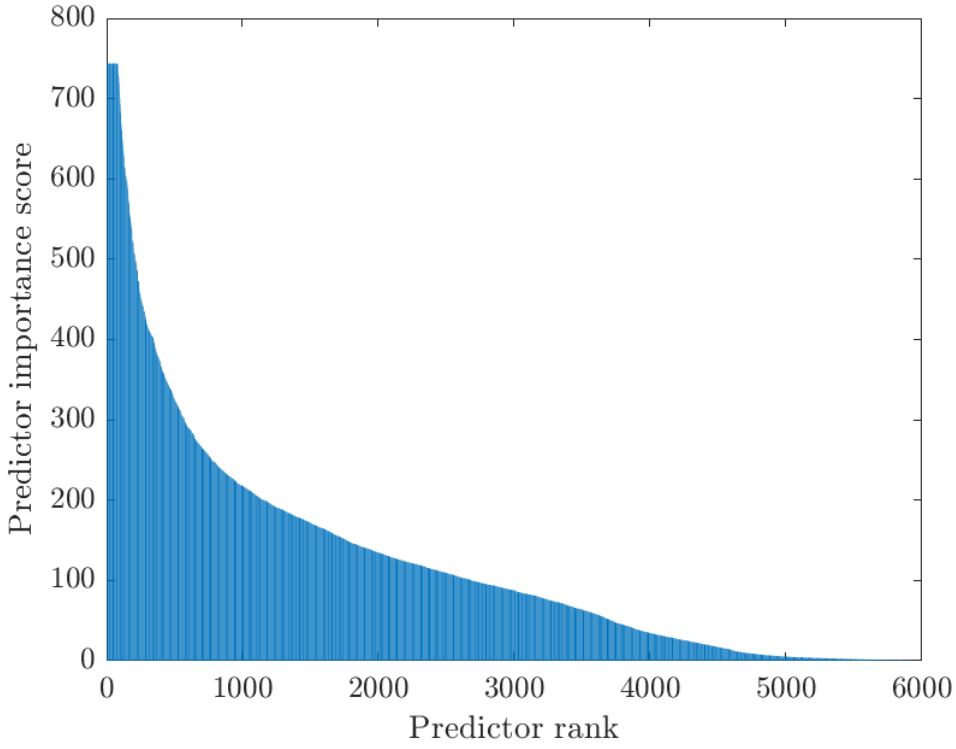


Figure 4.1: Example of ranking of predictors from the chi-square feature selection method. Scores equal to ∞ have been changed to the largest finite score.

The second feature selection method uses the MRMR algorithm to determine the importance of a feature. In order to best represent the classification labels, the algorithm finds a set of features that are maximally and mutually distinguishable. To do this, the algorithm uses the pairwise mutual information of features and the mutual information of a feature and the classification label. [38]

Mutual information is a measure between two variables. It quantifies the amount of uncertainty that can be removed by knowing the other variable. The mutual information, I, is defined as [39]

$$I(X, Z) = \sum_{i,j} P(X = x_i, Z = z_j) \ln \left(\frac{P(X = x_i, Z = z_j)}{P(X = x_i)P(Z = z_j)} \right), \tag{4.1}$$

where X and Z are two random variables. I is equal to 0 if X and Z are independent, and I is equal to the information entropy of X if X and Z are the same.

To find an optimal set of features, S, the relevance of S with respect to label y and the redundancy of S must be found. These are defined through I as [38]

$$V_S = \frac{1}{|S|} \sum_{x \in S} I(x, y), \tag{4.2}$$

$$W_S = \frac{1}{|S|^2} \sum_{x,z \in S} I(x, z), \tag{4.3}$$

where V_S is the relevance of S with respect to label y, W_S is the redundancy of S, and $|S|$ is the number of features in S. To find S all $2^{|\Omega|}$ combinations must be considered, with Ω being the entire feature set. Alternatively, the MRMR algorithm uses the forward addition scheme to rank features [40], requiring just $|\Omega||S|$ computations. This is done by using the mutual information quotient MIQ, written as [38]

$$\text{MIQ}_x = \frac{V_x}{W_x}, \tag{4.4}$$

where V_x is the relevance of a feature and W_x is the redundancy of a feature. V_x and W_x are written as [38]

$$V_x = I(x, y), \tag{4.5}$$

$$W_x = \frac{1}{|S|} \sum_{z \in S} I(x, z). \tag{4.6}$$

To use the algorithm, the MATLAB function `fscmrmr` is used, which ranks all features in Ω and quantifies their importance using a heuristic algorithm, which returns a score. An example plot showing the predictor rankings for a given setting can be seen in fig. 4.2. The rank 1 feature is the feature with the largest relevance, next are the features with nonzero relevance and zero redundancy, then features with nonzero relevance and nonzero redundancy, and lastly the features with zero relevance in random order. [38]

4.1.2 Feature Transformation

Another type of dimensionality reduction is feature transformation, where existing features are transformed into new arbitrary features, while lowering the overall number of features. This is a very effective way of reducing the dimensionality. There are two sub-categories to feature transformation: Supervised and unsupervised. In this project one of each type will be used.

Principal Component Analysis

Principal component analysis (PCA) is an unsupervised feature transformation method. Unsupervised refers to PCA not using the classification labels of the data, even when available. Commonly, when working with experimental data of high dimensionality, most of the dimensions are dependent on each other. In order to greatly reduce the redundancy of the data, PCA can be performed. PCA analyses the data and returns the so-called principal components; returning as many as there were dimensions originally. However, only a handful of the principal components are necessary to explain the vast majority of the variance in the data. PCA can often reduce

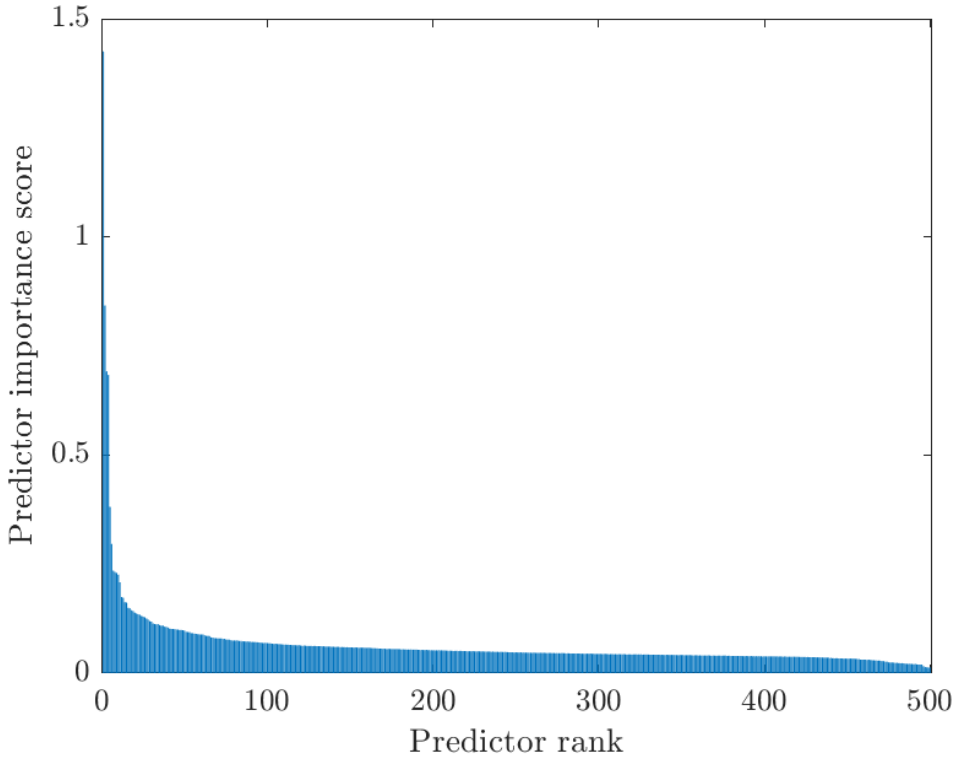
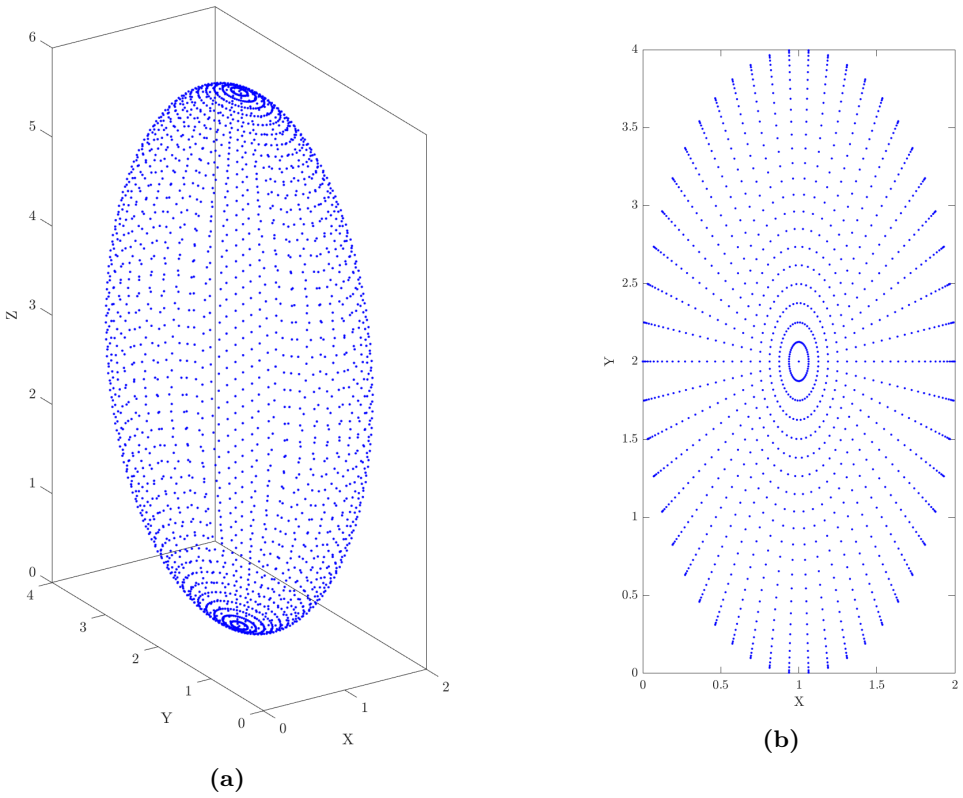


Figure 4.2: Example of ranking of predictors from the MRMR feature selection method.

the dimensionality from several thousands of features to merely a single digit number of features while still retaining over 95% of the data variance.

The principal components of a numerical data set are a set of orthogonal vectors. The principal components are used to change the basis of the data set, where the first axis is the principal component that explains the largest variance. The second principal component explains the most variance when the explained part of the first principal component is ignored. This goes on until as many principal components as original dimensions have been created. Figure 4.3 helps give a simplified understanding of this. The figure shows the plot of an example data set, with just three features; meaning it can be plotted in its entirety in three dimensions. The first principal component seeks out the direction with the most variance, which is along the z-axis. For the second principal component, the z-axis is then removed, ensuring that any chosen principal component will be orthogonal to the first one, and as seen in fig. 4.3b, the largest remaining variance is in y-direction, so the second principal component will be along the y-axis.

Mathematically, PCA finds the principal components by performing eigendecomposition on the covariance matrix of the data [41]. This is done in MATLAB with the function `eig`. The returned eigenvectors are sorted by their eigenvalues, as their eigenvalues are proportional to the amount of variance explained by each principal component. An example of the cumulative fraction of variance explained based on the number of principal components selected to be kept can be seen in fig. 4.4. Starting with 3000 features, PCA returns principal components where keeping three of them are enough to explain over 95% of the original data.



(a) (b)

Figure 4.3: Plot of a data set, viewed in both three and two dimensions.

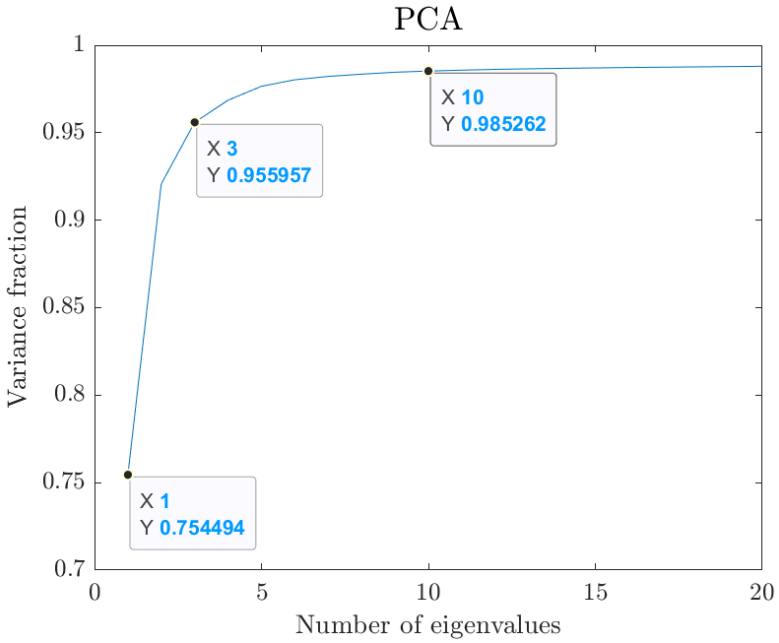


Figure 4.4: Example of cumulative variance fraction based on number of principal components. The dimensionality of the data used is 3000 features.

Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a supervised feature transformation method. Contrary to PCA, LDA uses the classification labels in its calculation by being 'supervised' in what the correct answer to each measurement is. Similar to PCA, LDA finds linear combinations of variables that best explain the data. However, LDA works based on two parameters like the MRMR algorithm does. LDA transforms the feature space in a way that maximises the distance between classes and minimises the spread in each group of classes, called a cluster. [41]

To perform LDA the between-class and within-class scatter matrices, S_b and S_w, respectively, must be calculated. They are defined as [41]

$$S_b = \frac{1}{n}\sum_{j=1}^{m} n_j \left(c^{(j)} - c\right)\left(c^{(j)} - c\right)^T, \tag{4.7}$$

$$S_w = \frac{1}{n}\sum_{j=1}^{m}\sum_{x=X_j} \left(x - c^{(j)}\right)\left(x - c^{(j)}\right)^T, \tag{4.8}$$

where T denotes transposition, c is a global centroid (the centre of all data points), $c^{(j)}$ is a centroid of class j, x is a vector associated with a specific observation, n is the number of observations, and m is the number of classes [41]. Instead of S_w, a total scatter matrix S_t can be used. It is defined as [41]

$$S_t = \frac{1}{n}\sum_{i=1}^{n} (x_i - c)(x_i - c)^T = S_b + S_w. \tag{4.9}$$

The LDA task can be simplified to the eigendecomposition of $S_t^{-1}S_b$ or $S_w^{-1}S_b$, depending on the preferred choice of scatter matrices. Since S_b is the sum of m matrices of rank 1, the rank of S_b is limited to $m-1$. As such, the LDA result contains a maximum of $m-1$ independent eigenvectors that are associated with nonzero eigenvalues [41]. This is often a restriction on the dimensionality, as the number of classes typically are smaller than the number of features. Though, the dimensionality can be reduced below $m-1$ by simply choosing to select fewer features from the feature transformation method.

A perk of the significant reduction of dimensionality possible with feature transformation methods is the option to visualise the data in two or three dimensions. Figure 4.5 is one such visualisation where an example of high-dimensionality data transformed with PCA and LDA is plotted. The filled markers are the measurements used to train the algorithms and the hollow markers are the test measurements. The axes are arbitrary linear combinations of original features, and the feature transformation methods themselves are not models that predict the class of an unknown measurement, however it can be used to better understand patterns in the data and similarities between classes.

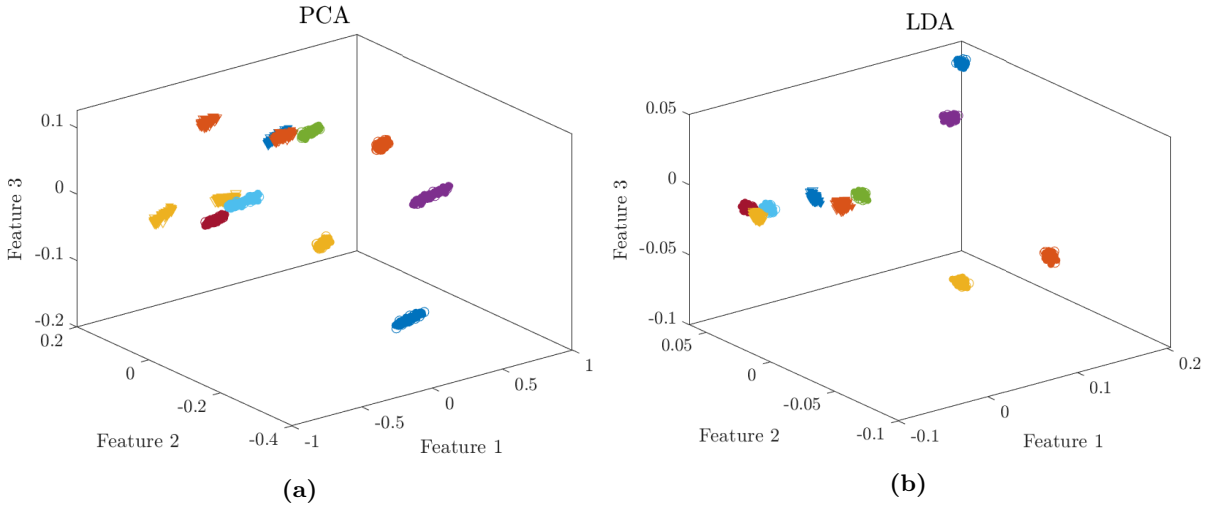


Figure 4.5: Example of the feature space after using principal component analysis and linear discrimination analysis as a means of dimensionality reduction. The filled markers is training data and the empty markers is testing data.

4.2 Machine Learning Classification Algorithms

After the data has been appropriately prepared, next step is to use a machine learning algorithm, called a classifier, to see if a model with a decent accuracy can be created. Several classifiers exist, and which one to use depends on a variety of things. In this project, the three different classifiers used are naive Bayes, discriminant analysis, and k-nearest neighbour.

Naive Bayes

The naive Bayes (NB) classifier is a probabilistic model based on Bayes rule, which can be written as

$$P(\mathbf{C}|\mathbf{x}) = \frac{P(\mathbf{x}|\mathbf{C})P(\mathbf{C})}{P(\mathbf{x})}, \tag{4.10}$$

where $\mathbf{x}$ are the feature vectors and $\mathbf{C}$ are the class labels. The naive Bayes classifier is trained with a labelled training set, from which it can calculate the right-hand side of eq. (4.10). Considering just a single feature vector $\mathbf{x_i}$ and a single class C_j; $P(\mathbf{x_i}|C_j)$ is the probabilities of class C_j having the values in $\mathbf{x_i}$, $P(C_j)$ is the probability of a random measurement being of class C_j, and $P(\mathbf{x_i})$ is the probability of a random feature vector being equal to feature vector $\mathbf{x_i}$. The left-hand side of eq. (4.10), $P(C_j|\mathbf{x_i})$, is the probability that $\mathbf{x_i}$ belongs to class C_j. $P(\mathbf{x})$ can be expanded to be a sum,

$$P(\mathbf{x}) = \sum_{\mathbf{C}} P(\mathbf{x}|\mathbf{C})P(\mathbf{C}), \tag{4.11}$$

it is often insignificant and is instead used as a normalisation factor. [41, 42]

The naive Bayes classifier is 'naive' because it assumes all features to be independent of each other. The classifier also uses an estimation for the distribution of the data. The MATLAB function used for the naive Bayes classifier is `fitcnb`, and it has an option to specify the data

distribution estimation method. The simplest data distribution is to assume a normal distribution, but a more accurate approach is to use a kernel density estimation, however, this comes at a significant increase in computational time when working with large data sets and high dimensionality. As such, a normal density distribution will be used unless otherwise stated. [43]

Discriminant Analysis

Besides being used for feature transformation, discriminant analysis (DA) can also be used as a machine learning classifier. In MATLAB, this is done using the `fitcdiscr` function. The discriminant analysis can be specified to be either linear, as in the feature transformation method, or to be quadratic. The main difference between the two is that in LDA the covariance matrix is based on all classes and in quadratic discriminant analysis (QDA) each class has its own covariance matrix. QDA will be the standard discriminant analysis classifier used, unless otherwise stated. [44]

K-Nearest Neighbour

The k-nearest neighbour (kNN) classifier is a simple, yet fundamental classifier. The major advantage is that little to no knowledge about the data distribution is necessary. While being a machine learning algorithm, kNN has no 'training-phase'; it simply stores the training data for when kNN is being used to predict the class of some data. For each measurement, kNN finds the distance to the nearest k neighbouring measurements and classifies the input data based on which class is the most abundant in the k nearest neighbours. The k-value is specified by the user, where k is an integer between 1 and the number of measurements used in the training data. For a small k, the classifier is prone to misclassification due to outliers, but is flexible. For a large k, kNN is robust, but with an uneven number of measurements per class in the training set, the classes with fewer measurements will be harder to correctly predict. At k = number of measurements, kNN will predict the class with the highest number of measurements, no matter what the input data is. [41, 45]

The point of kNN is to be able to find the distance between measurements regardless of how the data is distributed. However, the distance between two measurements can be defined in a multitude of ways, examples include city block distance, Chebychev distance, and Euclidean distance. The function for kNN classification in MATLAB is `fitcknn`, which takes an optional input for which distance metric to use. In this project, the standard distance metric is Euclidean, and the standard k-value is 47.

5 | Results and Discussion

5.1 Numerical Model Results

This section will examine the validity of the model developed in chapter 2, and the model is used to examine periodic arrays of antennas to find their resonance frequencies. Since the extinction efficiency is proportional to the transmittance [46], the frequency at minimum transmittance will directly correspond to the frequency at maximum extinction efficiency, which is the resonance frequency. Therefore, only transmittance spectra are required for determining the resonance frequencies, and as such, only these will be examined for that purpose. Furthermore, note that, in the transmittance spectrum, the resonance frequency will be a dip, or a valley, however, since these correspond to a peak in the extinction efficiency spectrum, all resonance dips will be referred to as resonance peaks.

Unless explicitly stated otherwise, FDTD-simulations for determining transmittance spectra, were made in a simulation space as shown in fig. 5.1. The structure of interest is placed on a 50 µm silicon-layer with a 50 µm air-layer above it. The PML at the air-layer was chosen to be 10 cells thick, and the PML at the silicon-layer was chosen to be 40 cells thick, as these were found to be adequate. Furthermore, the source signature was on the form of the modulated Gaussian pulse shown in fig. 5.2. All antennas will be constructed in such a way that they lie lengthwise along the x-direction. Therefore, the excitation direction is also along x, and all incident fields are consequently polarised along x. Finally, it is noted that the time-step is always chosen as 99% of the limit of the CFL condition, eq. (2.208).

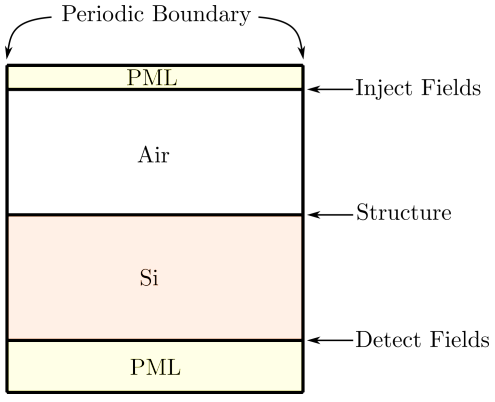


Figure 5.1: Sketch of the simulation space.

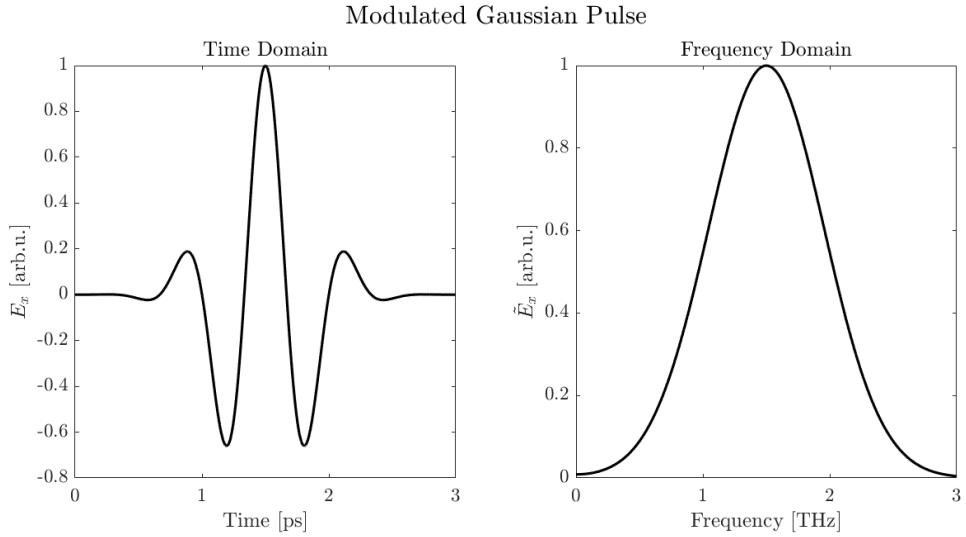


Figure 5.2: The time and frequency domains of the modulated Gaussian source signature used for obtaining transmittance spectra of antenna structures. The pulse has a time-delay of $t_0 = 1.5$ ps, a width of $\tau = 0.35$ ps, and a modulation frequency of $f_m = 1.5$ THz.

5.1.1 Infinitely Thin Antennas

The structures examined in this project are typically ∼0.12 µmthick, and the shortest wavelength in the THz-regime is typically considered to be ∼100 µm at a frequency of 3 THz. Though, in many practical situations, the high-end frequency lies ∼5 THz with a wavelength of ∼60 µm, which is still considerably larger than the structure thickness. Therefore, the structures could be considered infinitely thin from the perspective of the incident field. This assumption is tested by simulating periodic arrays of antennas with finite thickness and infinitely thin antennas, referred to as 3D and 2D antennas, respectively.

The 2D structures are simulated by projecting the actual (3D) structure onto the air-Si-interface. Due to the staggered grid, all field components cannot be sampled exactly at this interface. From the continuity conditions, eqs. (2.8) to (2.11), the immediate choice of placement of the interface is along the grid points where E_x and E_y are sampled, as this prevents the need for special treatment to handle the discontinuity of E_z. However, the strongest argument for this choice of interface placement is that the 2D structure is a PEC, and, as such, E_x and E_y must be zero within it. Thus, by placing the interface, and therefore the 2D structure, along the same grid points where E_x and E_y are sampled, these field components may simply be set to zero inside the structure at every time step, and E_z does not need be considered.

The simulated structure is a periodic array of 45 µm long and 5 µm wide antennas separated by 5 µm along the length and 10 µm along the width of the antenna, such that the periods are $\Lambda_x = 50$ µm and $\Lambda_x = 15$ µm.

The 3D antennas have a thickness of 1.00 µm. While this is thickness is several factors higher compared to the rest of the antennas examined in this project, it is still significantly lower than the wavelengths in the THz-regime. A 1.00 µm thickness was chosen to speed up the simulation, as a small discretisation not only requires more calculations per time step, but also requires more

time steps as the time step will change with the spatial discretisation. The simulation was made in a domain of $100 \times 30 \times 350$ cells with discretisations of $\delta x = 0.50$ µm, $\delta y = 0.50$ µm, and $\delta z = 0.33$ µm. The simulation was run over 12000 time steps with a discretisation of $\delta t = 0.81$ fs.

Four different simulations were made of 2D antennas in comparison with the simulation of 3D antennas to not only test the validity of the 2D approximation, but also to see, if the discretisation along z is of significant importance. The simulations of 2D antennas were made in domains of $100 \times 30 \times 450$ (/350/250/150) cells with discretisations of $\delta x = 0.50$ µm, $\delta y = 0.50$ µm, and $\delta z = 0.25$ (/0.33/0.50/1.00) µm, respectively. The simulations were run over 15000 (/12000/10000/9000) time steps with a discretisations of $\delta t = 0.68$ (/0.81/0.97/1.12) fs, respectively.

The transmittance spectra from these simulations are plotted together in fig. 5.3. The overall signatures of the transmittance spectra seem to agree, and most importantly: All simulations show a resonance frequency of 1.65 THz. This is a good indication that the 2D approximation does indeed hold for the structures examined in this project, and, as such, it shall be used for simulations henceforth.

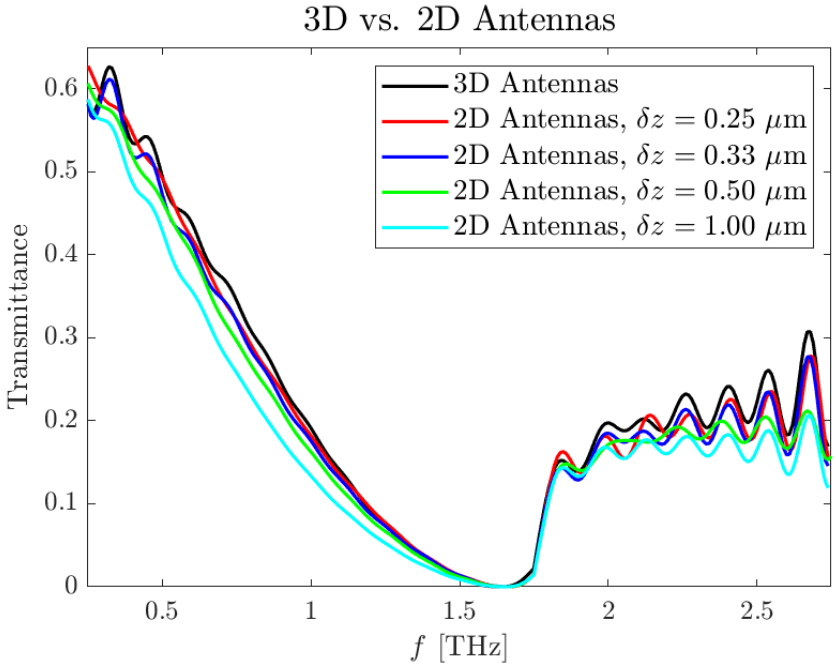


Figure 5.3: FDTD simulation results of 1 µm thick (3D) antennas and infinitely thin (2D) antennas. All antennas have the parameters: $L = 45$ µm, $W = 5$ µm, $G_x = 5$ µm, and $G_y = 10$ µm.

5.1.2 Slit Array

The Fourier modal method is a mode expansion technique that separates the structure into layers, solves the eigenmodes in each layer, and matches the modes. A good description of the method can be found in *Numerical Methods in Photonics* by Lavrinenko et al. [12].

The structure considered is a periodic array of 40 µm long and 1.5 µm wide slits in a 75 nm

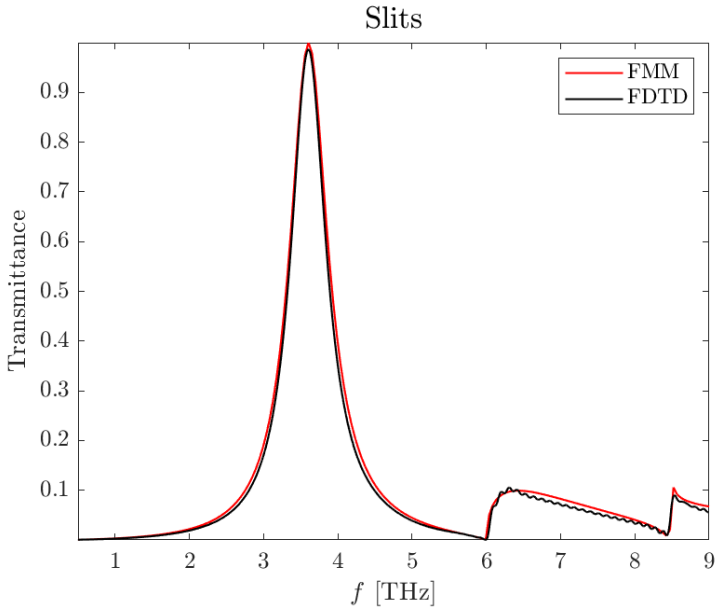


Figure 5.4: FDTD simulation results of an array of slits compared to the results of the Fourier modal method (FMM).

thick PEC layer. The period is 50 µm in both directions. The layers above and below the PEC/slit-layer is air, and the filling in the slits is air as well. Slits are chosen because analytic solutions exist for all three layers, so that the numerical simulations are held up against analytical simulations. It is noted that, oppositely to antennas, the excitation direction of slits is along the width of the slit, and the incident fields are polarised along the excitation direction.

The FDTD simulation had a domain of $100 \times 100 \times 170$ cells with discretisations of $\delta x = 0.50$ µm, $\delta y = 0.50$ µm, and $\delta z = 0.33$ µm. The simulation was run over 22000 time steps with a discretisation of $\delta t = 0.44$ fs. The pulse has a time-delay of $t_0 = 0.5$ ps, a width of $\tau = 0.12$ ps, and a modulation frequency of $f_m = 5$ THz. The results are shown in fig. 5.4.

Notice that at ∼6 THz and ∼8.4 THz, the transmittance reaches zero. These frequencies are characterised as the cut-off frequencies of the first and second diffraction orders. At these frequencies, the propagation constant is zero, and all scattered waves are consequently propagating along the structure. Thus, besides predicting the main resonance frequency at ∼3.6 THz, the FDTD simulation also accurately predicts the cut-off frequencies of at least the first and second diffraction orders.

5.1.3 Antenna Arrays

Arrays of antennas are the main focus in this project, and it is therefore of interest to examine the transmittance spectra when different parameters are varied. The parameters include the length, L, the width, W, and the gaps between antennas, G_x and G_y. These parameters are illustrated in fig. 5.5.

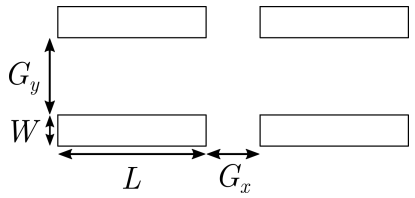


Figure 5.5: Illustration of the antennas simulated. L and W are the length and width of the antennas. G_x and G_y are the gaps between antennas along the length and width of the antennas, respectively.

Figure 5.6 shows the transmittance spectra from FDTD-simulations of antennas with parameters: $L = 30$ (/35/40/50/60) µm, $W = 1$ µm, $G_x = 10$ µm, and $G_y = 20$ µm. The resonance frequencies appear at 1.90 (/1.67/1.49/1.24/1.06) THz. It is observed that the transmittance spectrum is redshifted, but keeping its form, when the length of the antenna is increased.

Figure 5.7 shows the transmittance spectra from FDTD-simulations of antennas with parameters: $L = 50$ µm, $W = 0.2$ (/0.5/1.0/2.0/5.0) µm, $G_x = 10$ µm, and $G_y = 20$ µm. The resonance frequency appears at 1.83 (/1.85/1.90/1.92/1.95) THz. It is observed that the resonance peak is blueshifted and broadened, when the width of the antenna is increased.

Figure 5.8 shows the transmittance spectra from FDTD-simulations of antennas with parameters: $L = 30$ µm, $W = 1$ µm, $G_x = 5$ (/10/15/20/25) µm, and $G_y = 20$ µm. The resonance frequency appears at 1.24 (/1.24/1.22/1.19/1.15) THz. It is observed that the resonance peak is redshifted, and broadened, when the gap along the length of the antenna is increased.

Figure 5.9 shows the transmittance spectra from FDTD-simulations of antennas with parameters: $L = 30$ µm, $W = 1$ µm, $G_x = 10$ µm, and $G_y = 10$ (/20/30/40/50) µm. The resonance frequency appears at 1.41 (/1.24/1.16/1.11/1.07) THz. The resonance frequency appears at 1.24 (/1.24/1.22/1.19/1.15) THz. It is observed that the resonance peak is redshifted, and broadened, when the gap along the width of the antenna is increased.

Razzari et al. [46] produced and examined arrays of gold antennas of various lengths deposited on silicon. The antennas had a thickness of 60 nm, a width of 200 nm, and the gap between antennas was 20 µm in both directions. The antennas had lengths $L = 30$ (/35/40/50/60) µm with resonance frequencies at around 1.75 (/1.60/1.41/1.16/0.98) THz, respectively [46].

Similar structures were simulated with the FDTD method. However, because a width of 200 nm in 20.2 µm domain requires a relatively high discretisation, a width of 1.00 µm in a 21 µm domain was chosen instead to increase computation speed. The transmittance spectra are plotted in fig. 5.10. The resonance frequencies lie at NaN (/1.57/1.42/1.19/1.03) THz. The resonance frequency for $L = 30$ lies beyond the cut-off frequency of the first diffraction order, and it is therefore not possible to extract it from the results. These numerical results deviate from the experimental results of Razzari et al. [46] by $< 5\%$, which is a good validation of the numerical model.

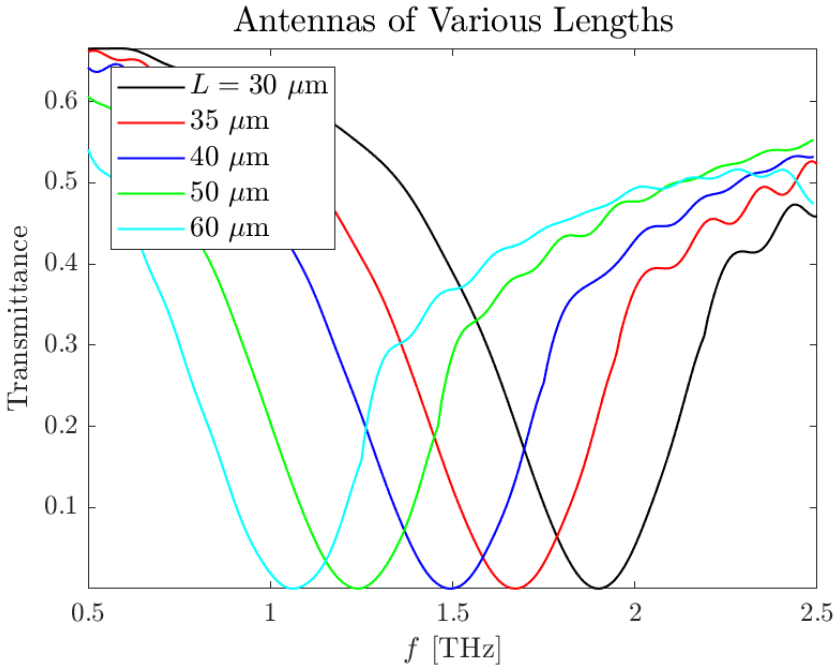


Figure 5.6: FDTD simulation results of antennas with parameters: $L = 30$ (/35/40/50/60) µm, $W = 1$ µm, $G_x = 10$ µm, and $G_y = 20$ µm. The simulations were made in domains of 80(/90/100/120/140) × 42 × 200 cells.

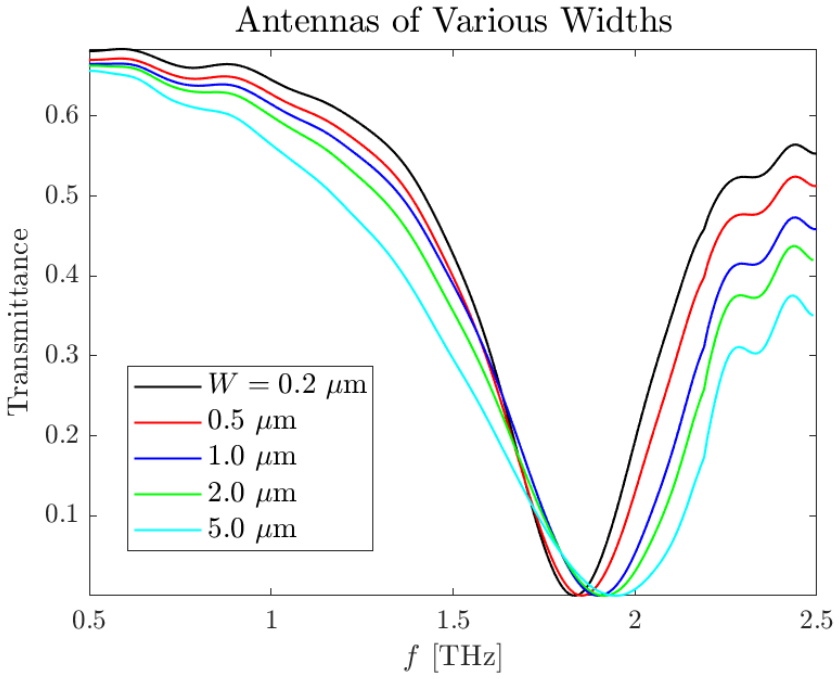


Figure 5.7: FDTD simulation results of antennas with parameters: $L = 50$ µm, $W = 0.2$ (/0.5/1.0/2.0/5.0) µm, $G_x = 10$ µm, and $G_y = 20$ µm. The simulations were made in domains of 80 × 200(/53/42/44/50) × 200 cells. The simulations were run over 31000 (/11000/10000/14000/10000) time steps with time discretisations of $\delta t = 0.32$ (/0.87/0.97/0.74/0.97) fs.

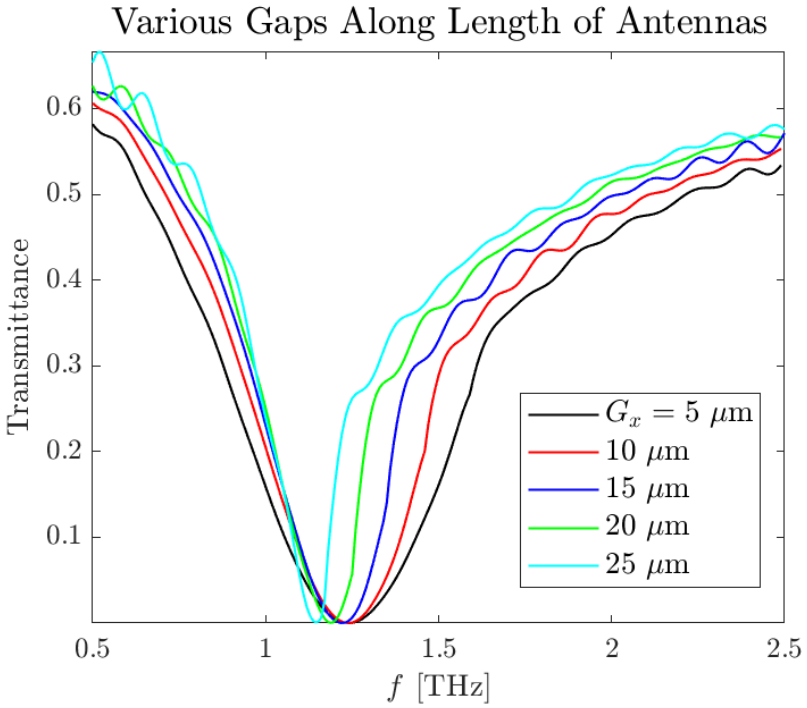


Figure 5.8: FDTD simulation results of antennas with parameters: $L = 30$ µm, $W = 1$ µm, $G_x = 5$ (/10/15/20/25) µm, and $G_y = 20$ µm. The simulations were made in domains of 70(/80/90/100/110) × 42 × 200 cells. All simulations were run over 10000 time steps with a time discretisation of $\delta t = 0.97$ fs.

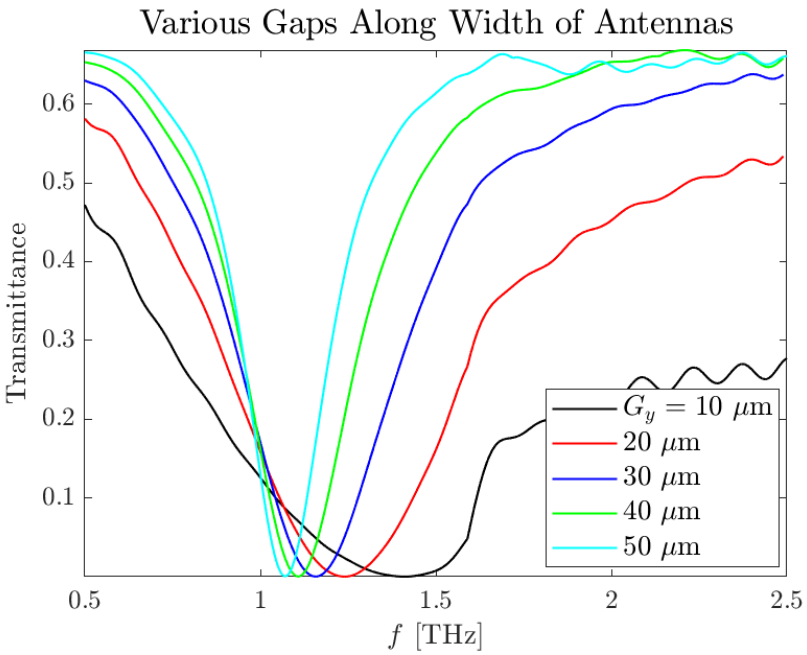


Figure 5.9: FDTD simulation results of antennas with parameters: $L = 30$ µm, $W = 1$ µm, $G_x = 10$ µm, and $G_y = 10$ (/20/30/40/50) µm. The simulations were made in domains of 70×24(/42/62/82/102)$\times 200$ cells. All simulations were run over 10000 time steps with a time discretisation of $\delta t = 0.96$ fs.

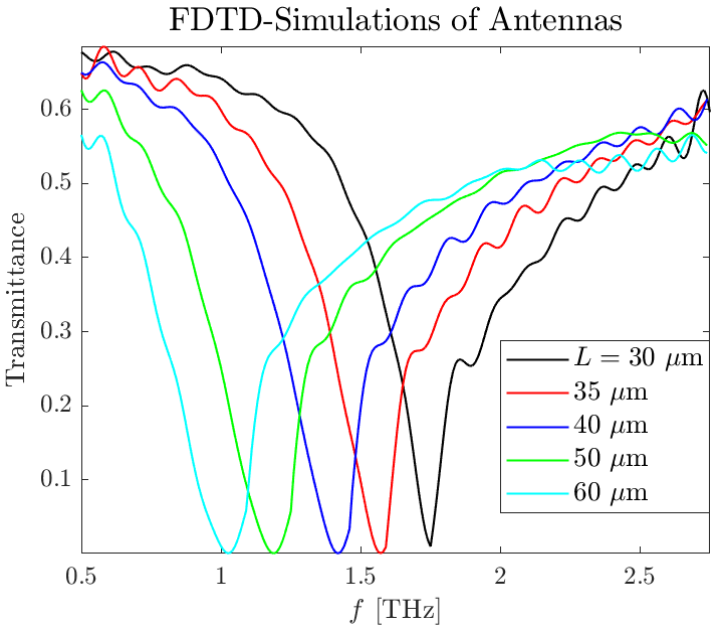
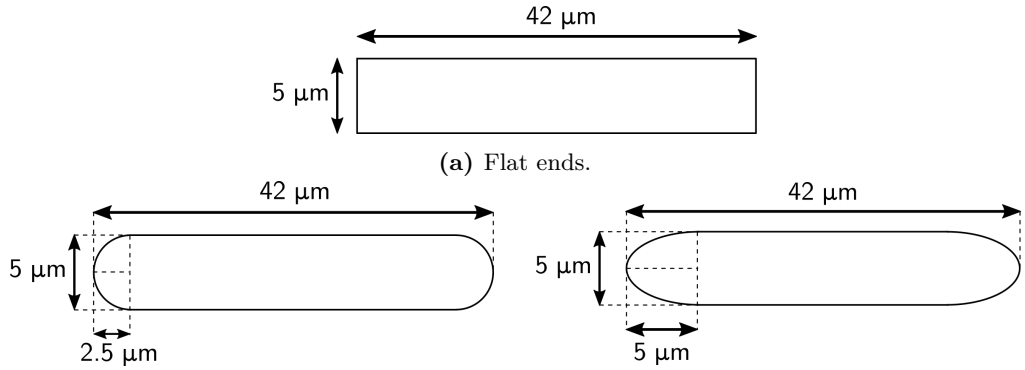


Figure 5.10: Transmittance spectra results from FDTD-simulations of periodic arrays of antennas with different lengths, L, a width of 1.00 µm, and a gap between antennas of 20 µm in both directions.

To summarise, it has been found that adjusting the length of the antennas will greatly affect the resonance frequency, but not the width of the peak. On the other hand, changing the width and the gaps will only affect resonance frequency slightly, but are important factors in defining the width of the peak. Thus, to obtain a narrow peak at a desired frequency, the width of the antenna should be minimised, and the gaps should be maximised. Then, the length of the antennas should be determined to obtain the desired resonance frequency. However, a well-defined resonance peak is not the only desired characteristic of the structures. Since the main intend of the structures is to create field enhancement effects for the purpose of sensing substances, another important characteristic is the fraction of the total area, where the field is enhanced. As will be explained in the next section, the field is mainly enhanced closely to the structures. Therefore, one would wish to fit as many structures into as small an area as possible, to increase the total area of field enhancement. As such, compromises must be made: The antennas must be sufficiently separated to have an acceptable peak width, while also making sure that a satisfactory fraction of the total area will have field enhancement. A similar compromise must also be made for the width of the antennas.

5.1.4 Antenna Shapes and Field Enhancement

So far, only perfectly rectangular antennas have been considered. However, from a production side of things, the shape of the ends of the antennas will typically deviate from being perfectly flat. Therefore, two alternative shapes for the ends of the antennas are examined in this section, see fig. 5.11. The gaps between antennas are $G_x = 8$ µm and $G_y = 25$ µm. Furthermore, the purpose of the antennas produced in this project is for sensing applications. As such, the field enhancement of the antennas at the resonance frequency is an important characteristic to examine.



(a) Flat ends.

(b) Circular ends.

(c) Elliptical ends.

Figure 5.11: Schematic of the different antenna designs used for examining the effects of different end shapes.

Figure 5.12 shows the transmittance spectra from FDTD-simulations of arrays of antennas with flat, circular, and elliptical ends with resonance frequencies at 1.46, 1.49, and 1.52 THz, respectively. Besides a slight shift in the resonance frequency, the overall transmittance spectrum does not appear to change when the ends are rounded.

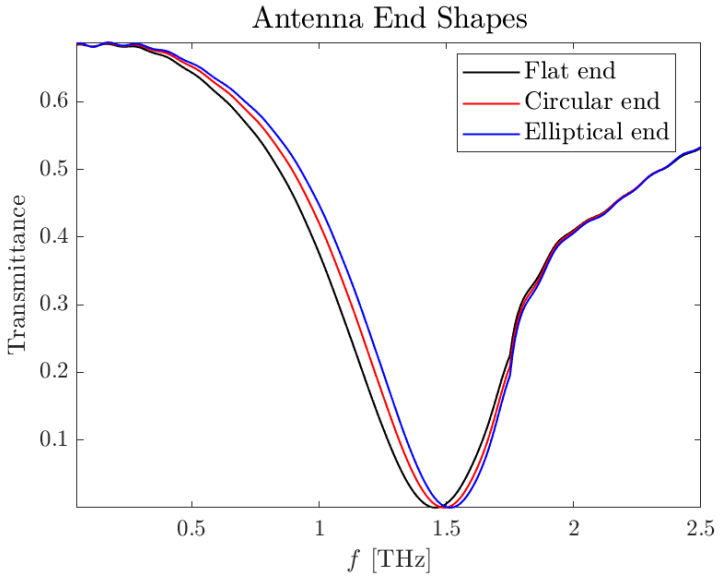
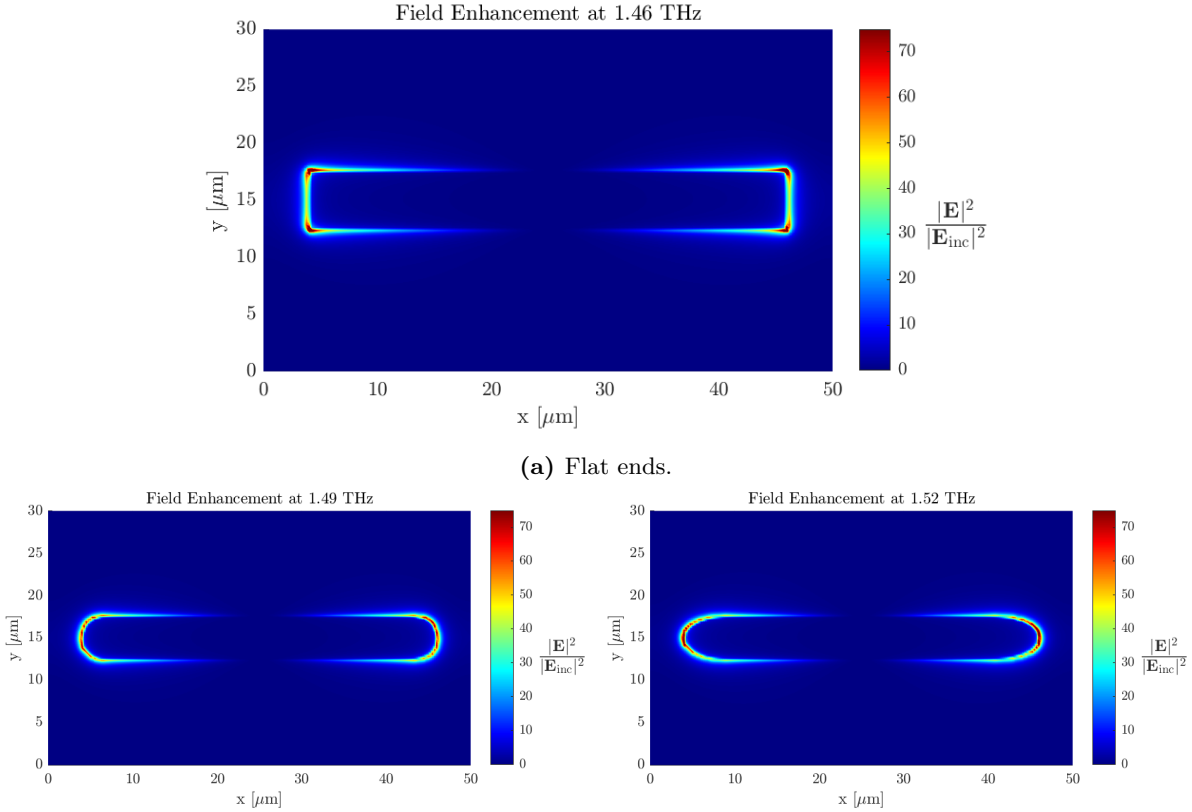


Figure 5.12: Transmittance spectra results from FDTD-simulations of periodic arrays of antennas with different end shapes.

Figure 5.13 shows the spatial distributions of the field enhancement for the transient state at the resonance frequencies. These were found by using a sinusoidal source with a frequency equal to the resonance frequency of the given structure. The expression for this source signature is given in eq. (2.226). The simulation was run until the steady-state was reached, and then the time-averaged field in the same plane as the structure was saved and divided by the time-averaged incident field. The simulation was made with a simulation domain of $200 \times 120 \times 250$ cells.

The flat-ended antennas appear to localise the field around the sharp corners, while the rounded antennas seem to distribute the fields more evenly at the ends. However, all antennas localise the field strongly along the ends of the antennas. Figure 5.14 shows similar antennas, but with a gap of $G_x = 1$ µm. A stronger field enhancement is observed, and the fields are better contained between the antennas. fig. 5.15 shows antennas with a width of $W = 1$ µm, and a similar behaviour is seen when the gap is shortened, however, it is also seen that the field enhancement is stronger. Note that all field enhancement plots are at the resonance frequency of their respective structure, however, since the transmittance spectra are all similar, they are not plotted.

In order to get the strongest field enhancement, the antennas should thus be as thin and close as possible. However, as discussed in the previous section, this will affect the width of the resonance peak, and the area in which the field is contained. Even though a round-ended antenna and a thin antenna gives stronger field enhancements, the effect is so well-contained, that the flat-ended antenna gives a better field enhancement-to-area ratio and is therefore the preferred option of the examined structures.

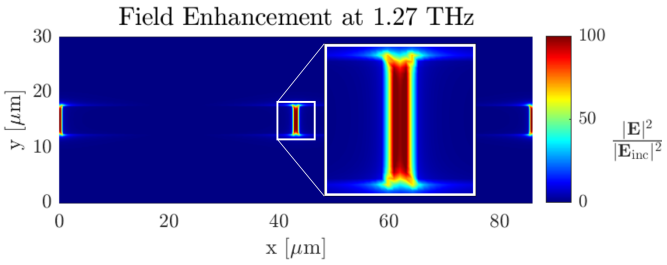


(a) Flat ends.

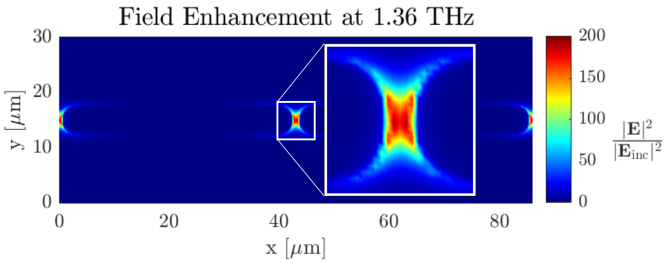
(b) Circular ends.

(c) Elliptical ends.

Figure 5.13: Plots of the field enhancements of a single period for arrays of antennas with flat, circular, and elliptical ends, and with parameters: $L = 42$ µm, $W = 5$ µm, $G_x = 8$ µm, and $G_y = 25$ µm.

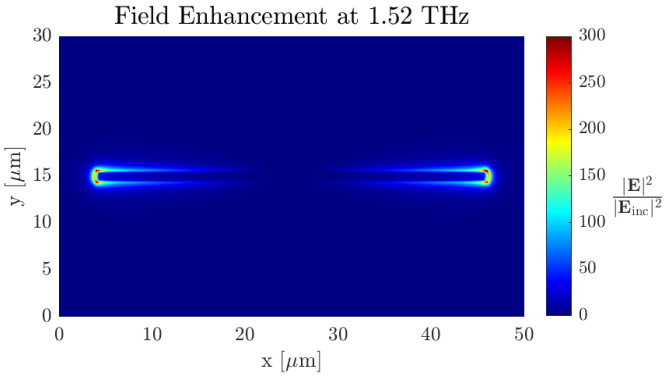


(a) Flat ends.

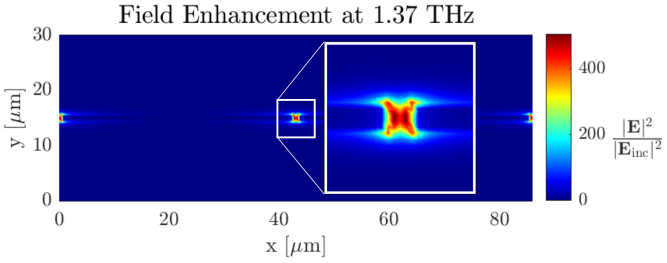


(b) Circular ends.

Figure 5.14: Plots of the field enhancements of arrays of antennas with flat and circular end, and with parameters: $L = 42$ µm, $W = 5$ µm, $G_x = 1$ µm, and $G_y = 25$ µm.



(a) $G_x = 8$ µm.



(b) $G_x = 1$ µm.

Figure 5.15: Plots of the field enhancements of arrays of antennas with parameters: $L = 42$ µm, $W = 1$ µm, $G_x = 8(/1)$ µm, and $G_y = 29$ µm.

5.2 Experimental Results

This section will characterise and analyse the antenna and split-ring resonator arrays produced. Scanning electron microscopy (SEM) with a Zeiss Evo 60 was used to ascertain the characteristics and dimensions of the antenna and split-ring resonator arrays. Mean values with standard deviation of antenna and resonator dimensions, as measured in the SEM, can be seen in table 5.2.

Antenna sample	Length [µm]	Width [µm]	X gap [µm]	Y gap [µm]
L24	24.06±1.81	5.10±1.10	16.98±1.48	26.02±1.09
L34	34.77±0.70	5.33±0.54	16.43±0.92	26.08±0.70
L39C	39.30±2.03	5.73±0.55	2.93±1.17	20.21±0.82
L39D	39.08±1.52	5.78±0.52	13.44±1.77	20.26±1.14
L40	40.19±1.49	5.47±0.32	12.37±1.34	20.67±0.76
L42	41.73±1.22	5.80±0.79	30.69±1.49	25.93±0.94
L45	44.93±1.57	6.40±0.80	28.14±1.96	24.90±0.62
L66	65.91±1.26	6.49±0.66	17.18±1.24	24.81±0.75
L86A	85.82±1.09	6.12±0.87	30.00±1.55	25.14±0.64
L86B	86.30±1.13	6.40±0.41	12.53±1.19	24.96±0.68
L87	87.45±0.88	6.00±0.68	27.26±1.34	25.93±0.85
L108	108.1±1.61	6.15±0.34	26.49±0.58	25.42±0.68

Table 5.1: Mean and standard deviation of antennas measured by SEM.

Split-ring sample	Side length [µm]	Width [µm]	X gap [µm]	Y gap [µm]	gap [µm]
S36	35.60±0.86	4.80±0.54	27.37±0.93	27.56±1.00	6.87±3.45
S68	68.03±1.34	5.85±0.93	25.42±1.46	27.03±1.04	2.65±0.83

Table 5.2: Mean and standard deviation of split-ring resonators measured by SEM.

The goal was to produce well-defined resonance peaks, and it was found that a small width will contribute to this. Using the Rayleigh resolution limit and multiplying it by two gives an idea of the theoretical minimum spot size of the laser in the UV direct write setup, found as

$$d_0 \approx 2 \cdot \frac{0.61\lambda}{\mathrm{NA}}, \tag{5.1}$$

where d_0 is the spot size, λ is the wavelength of the light, and NA is the numerical aperture. The spot size is found to be approximately 700 nm. Even when accounting for the inaccuracies involved in replicating this in a laboratory setting, the widths attained in this project have been unexpectedly large. The structures have also exhibited some other interesting features. One such feature, which is seen on all antenna array samples, is a noticeable change in width of the antennas as one moves from one side to another on the sample, as can be seen in fig. 5.17. This occurrence indicates that spot size changes when moving over the sample during the direct write process. The feature is making the antenna array less uniform, which is unwanted as it widens the resonance peak. This problem likely arose from difficulties when trying to keep the laser focused on the sample as it moves.

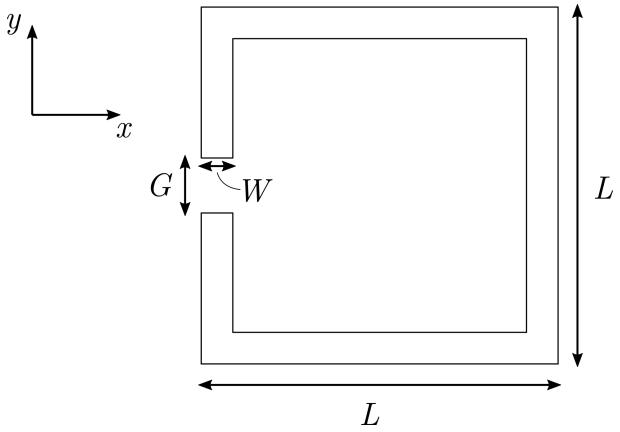
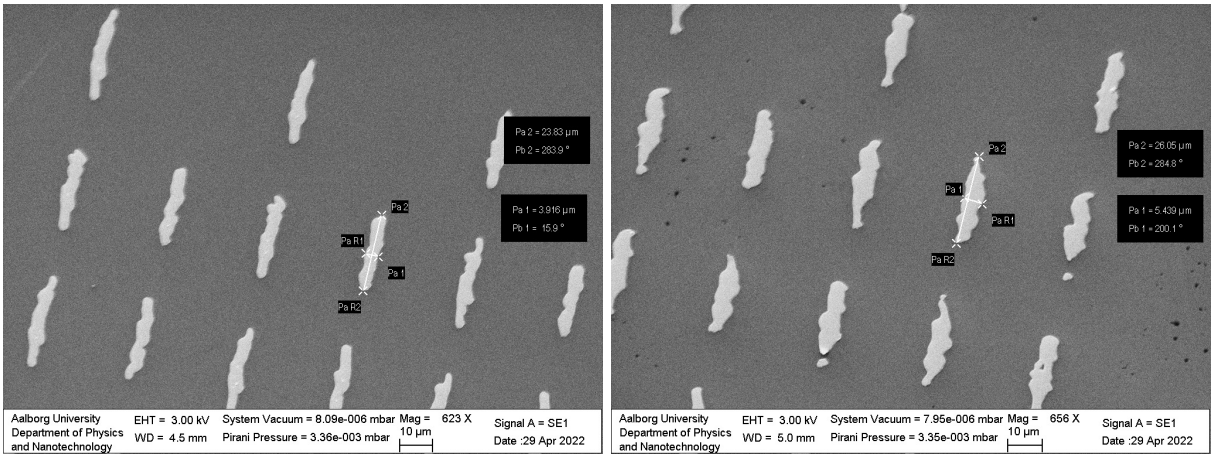
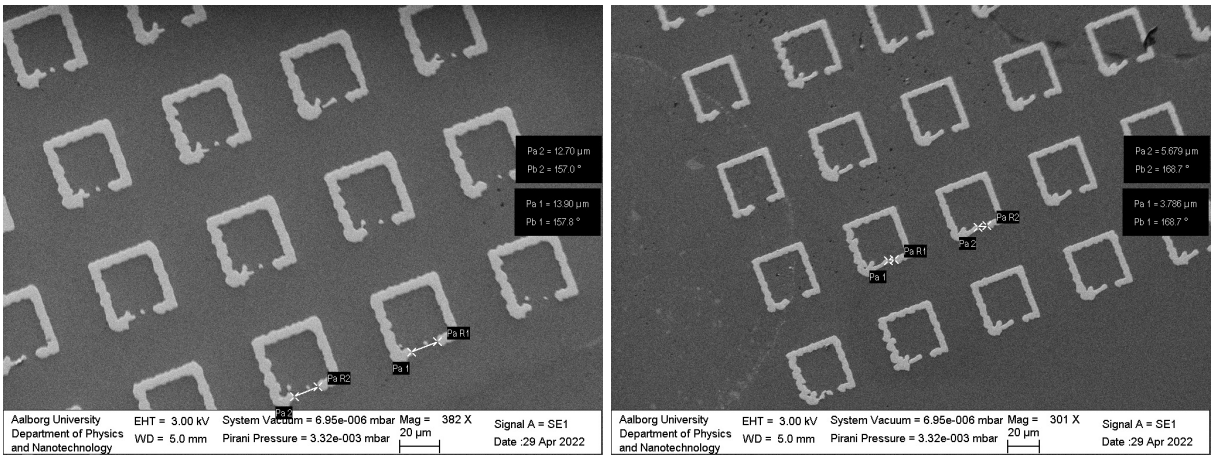


Figure 5.16: Sketch of the a split-ring resonator. L is the side length, W is the width, and G is the gap.



(a) **(b)**

Figure 5.17: SEM pictures showing sample L24 at two different ends of the antenna array.



(a) **(b)**

Figure 5.18: SEM pictures showing sample S36.

Two other features that are seen on some samples are the shortening of an antenna by omission or breakdown of the end and the lengthening of the antennas by the addition of a 'tail'. An example is given in fig. 5.18. The unpredictability and inconsistency of these phenomena causes difficulties in the production of samples with small gaps, as fusing structures causes new resonances and lessens the desired resonance. A possible way of stabilising the lengths of the antennas could be to deposit thinner gold layers to reduce stress during lift-off. However, this could be problematic, as the thickness of the gold layer currently in use is nearing the skin-depth of the terahertz regime.

Most of the structures produced exhibit a wavelike pattern on the edges, as seen in fig. 5.19. For some samples, these are so significant that a percentage of the antennas are split in two.

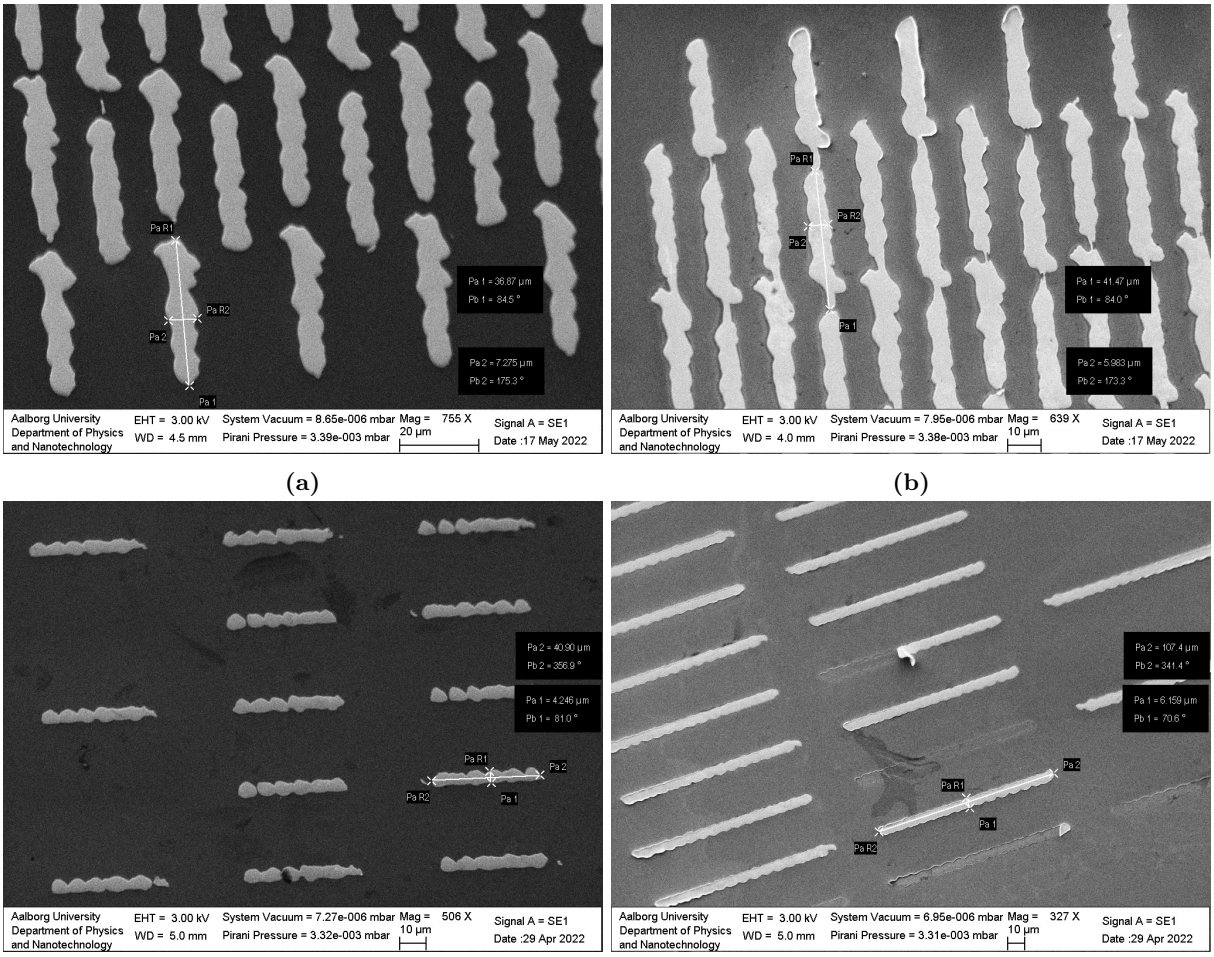


Figure 5.19: SEM pictures showing examples of the wave pattern on produced antennas.

The relative transmission of the produced antennas, with a reference produced by rotating the sample by 90° , can be seen in fig. 5.20. The experimental data is seen to mirror the model to an adequate degree. Though, there are some outliers like `L39C` and `L39D`. `L39C` can be explained by antennas fusing together, while `L39D` can be explained be the antennas breaking into smaller antennas.

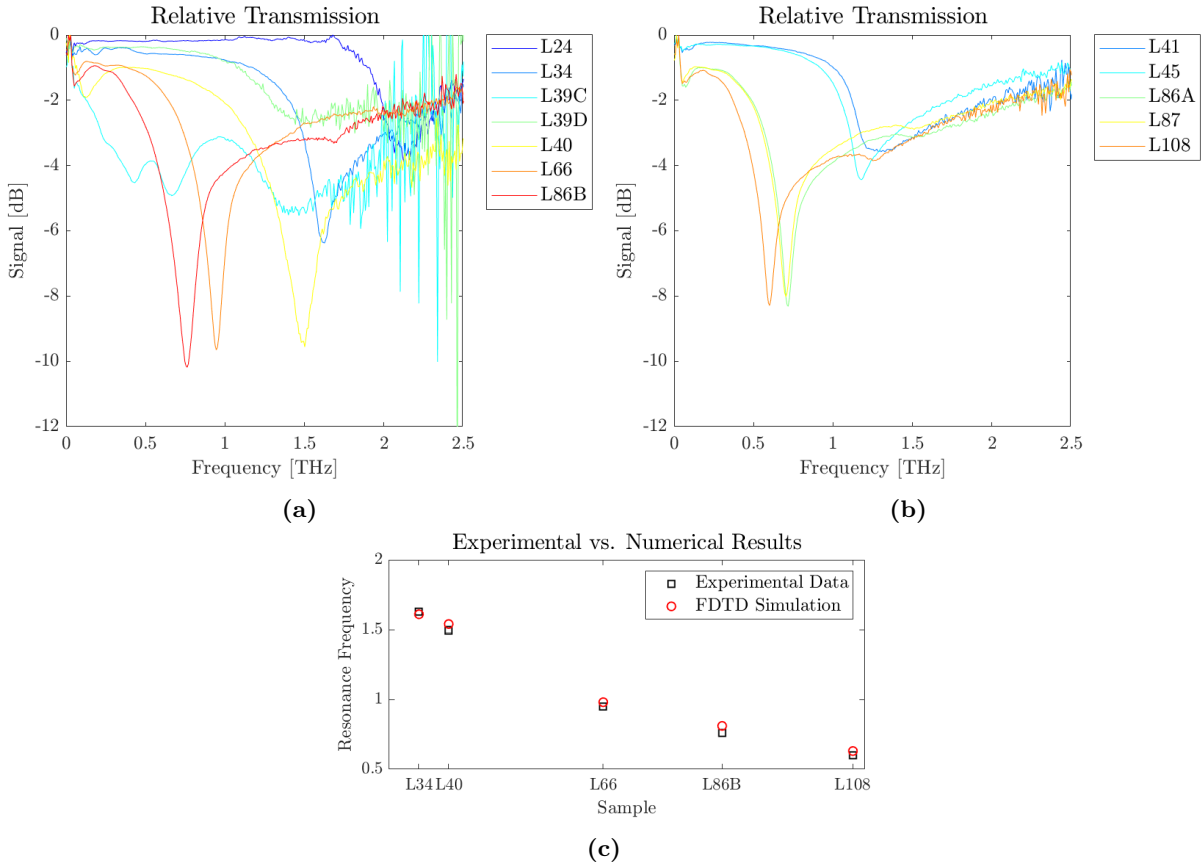
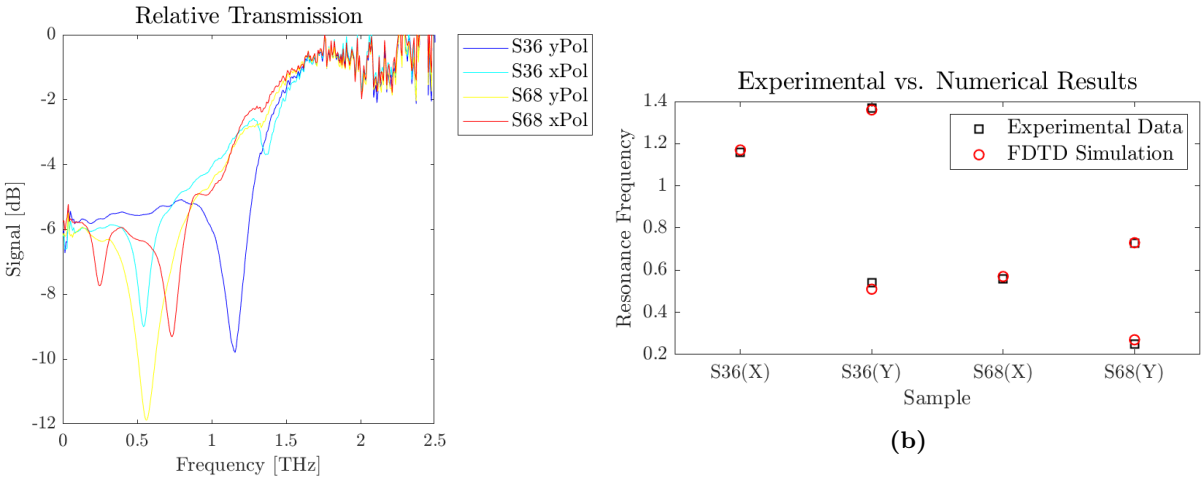


Figure 5.20: **(a)** and **(b)** are the relative transmission of the produced antennas in relation to the reference found by rotating the wafer by 90°. Measurements were taken with a delay interval of 0 to 200 ps and 100 measurements for averaging. **(c)** is a comparison of the measured resonance peaks of the sample with their FDTD modelled counterpart.

The relative transmission of split-ring resonator with the reference of an empty Si wafer can be seen in fig. 5.21. The spectra with the two peaks are the ones of interest as the field enhancement is found in the split-ring gap. This data fits the model to the same degree the antennas did.

In fig. 5.22, relative transmission of sample `L42` with glucose deposited, with a clean `L42` as reference. Looking at the spectra it is very difficult to perceive a pattern between the graphs and the amount of glucose on the sample.

The numerical model results were made for ideal antennas. However, it is evident that the antennas produced in this project are not ideal. As such, an antenna design that somewhat resembles the produced antennas have been modelled, as shown in fig. 5.23. The FDTD simulated transmittance spectra of these antennas are shown in fig. 5.24, and the field enhancement of the non-ideal antenna array is plotted in fig. 5.25. These simulations are made with identical parameters to those used when examining antenna end shapes in section 5.1.4. The resonance frequencies of the ideal and non-ideal antennas are 1.49 and 1.52 THz, respectively. The non-ideal antenna is very similar to the circular-ended antenna in both transmittance spectrum and field enhancement, however, the field enhancement is slightly weaker and less confined to the ends. The antennas are thus considered acceptable.



(a) **(b)**

Figure 5.21: **(a)** is the relative transmission of split-ring resonators with the reference of an empty Si wafer. Measurements were taken with a delay interval of 0 to 200 ps and 100 measurements for averaging. The plots of same name are a 90°rotation of the same sample. **(b)** is a comparison of the measured resonance peaks of the sample with their FDTD modelled counterpart. (X) and (Y) represent the polarisation of the incident light with respect to the orientation shown in fig. 5.16.

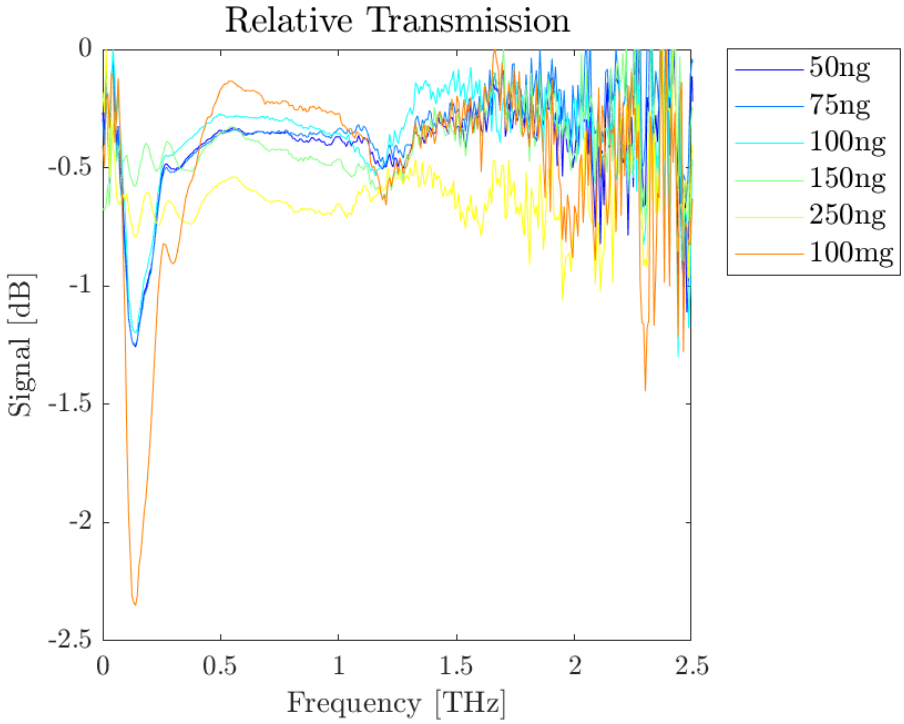
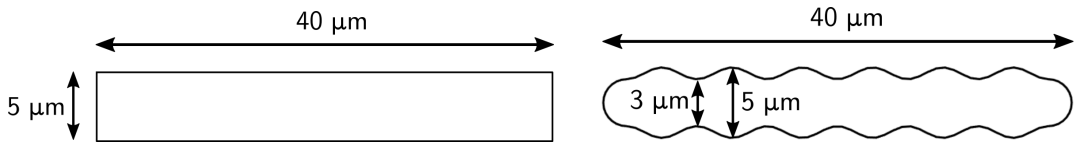


Figure 5.22: Relative transmission of `L42` with glucose deposited, compared to a clean `L42`. Measurements were taken with a delay interval of 0 to 200 ps and 100 measurements for averaging.



(a) Ideal antenna. **(b)** Non-ideal antenna.

Figure 5.23: Schematic of the ideal and non-ideal antenna designs.

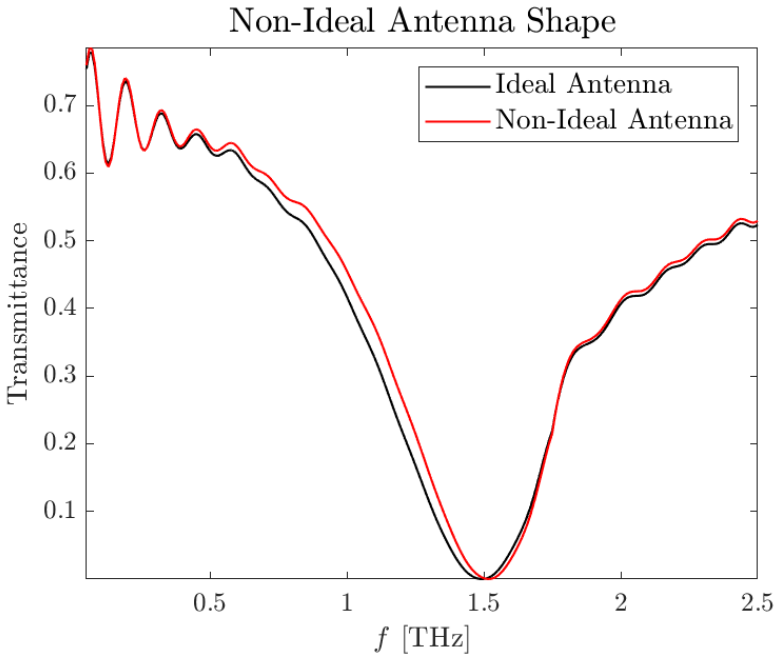


Figure 5.24: Transmittance spectra of the ideal and non-ideal antenna arrays.

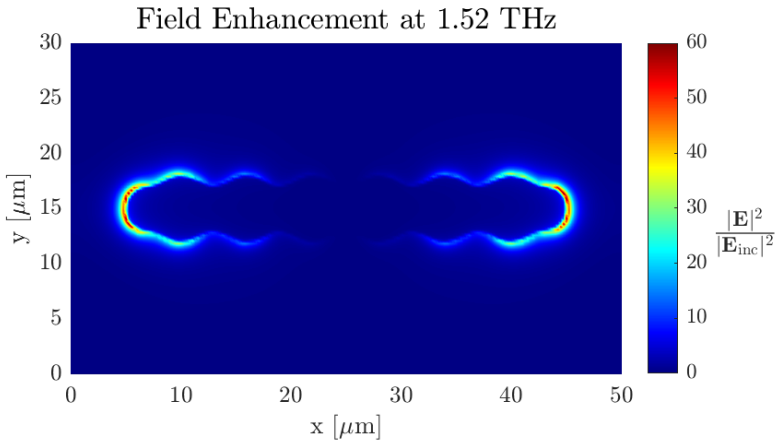


Figure 5.25: Field enhancement plot of the non-ideal antenna array.

5.3 Machine Learning Results

The MATLAB code used to perform machine learning in this project has been inspired by the code written by Paweł Piotr for the article "*Analysis and Classification of Frequency-Domain Terahertz Reflection Spectra Using Supervised and Unsupervised Dimensionality Reduction Methods*" [41].

For machine learning, a lot of data has been gathered: In total 17972 measurements were taken and have been used. All measurements share the same 6000 features, where the features are a linear discretisation of time delays from 0 to 199.9667 ps. Performing machine learning on all of this raw data at once will cause any normal computer to run out of memory and halt the process. Before even starting the machine learning process, the measurements of interest should be selected. There are 162 different types of measurements, and, unless stated otherwise, each type of measurement has 100 repeated measurements contained in it. The full extent of the data used in this project can be seen in Appendix A.

5.3.1 Feature Selection Results

For this section, 2800 measurements are used. The next step is dimensionality reduction: The simple form of which is feature selection. For the comparison of feature selection methods, the accuracy of the three machine learning algorithms in section 4.2 will be used for three different cases. The three cases are: Only feature selection, feature selection and PCA, and feature selection and LDA, all followed by classifiers. The accuracy of a classifier in this project is given by the fraction,

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total predictions}}. \tag{5.2}$$

Table 5.3 shows the accuracies of when feature selection is skipped, so the raw data is used for the classifiers, though, with feature transformation in column 3 and 4. This table serves as a reference for tables 5.4 to 5.6.

In tables 5.5 and 5.6, chi-squared and MRMR are used to select the 100 features with the highest score rated by the respective methods. To compare with this, 100 random features were selected 10 times over, and the average accuracies are shown in table 5.4. Table 5.4 shows a clear loss in accuracy compared to table 5.3, with the exception of the DA classifier, which performs significantly better. However, this is no surprise as discriminant analysis works best when the number of measurements exceeds the number of features. Tables 5.5 and 5.6 both show better accuracies than when performing no feature selection and when selecting 100 features at random, but with the caveat that the LDA accuracy in table 5.3 is the highest of all four tables.

However, when looking at fig. 5.26 a less than ideal result can be seen. Unlike the compact clusters in fig. 4.5, fig. 5.26 shows a large spread in the testing measurements - even overlapping with neighbouring clusters. It also appears as if there is only a single filled dot for each cluster, which is not the case, as each training measurement is simply plotted so tightly on top of each other that they appear as one. This is called overfitting, which is prone to happen with discriminant analysis when the number of features exceeds the number of measurements - both when used as a feature transformation method and when used as a classifier. In this case, the training set had 6000 features and 2240 measurements (80% of the 2800 measurements are used

for training and 20% for testing).

Algorithm	Raw Data	PCA	LDA
NB Train	0.5380	0.6982	1.0000
NB Test	0.5304	0.6661	0.8304
DA Train	0.2848	0.6763	1.0000
DA Test	0.3571	0.6554	0.8286
kNN Train	0.8835	0.8223	1.0000
kNN Test	0.8429	0.8018	0.8429

Table 5.3: Reference table where no feature selection is performed, at the cost of increased computational time, the full 6000 features are used.

Algorithm	Selection	PCA	LDA
NB Train	0.5066	0.5477	0.7702
NB Test	0.5013	0.5204	0.7129
DA Train	0.9829	0.5458	0.7713
DA Test	0.6834	0.5157	0.7123
kNN Train	0.7931	0.7109	0.7766
kNN Test	0.7638	0.6736	0.7107

Table 5.4: Feature selection where 100 features out of 6000 are chosen at random. This was repeated 10 times and the average accuracy is in the table.

Algorithm	Selection	PCA	LDA
NB Train	0.6513	0.7170	0.8067
NB Test	0.6357	0.6839	0.7589
DA Train	0.9906	0.6826	0.8125
DA Test	0.8214	0.6661	0.7661
kNN Train	0.9219	0.8406	0.8272
kNN Test	0.9036	0.8018	0.7750

Table 5.5: Feature selection where 100 features out of 6000 are chosen with the chi-squared feature selection method.

Algorithm	Selection	PCA	LDA
NB Train	0.6969	0.7348	0.8580
NB Test	0.6625	0.7054	0.8196
DA Train	0.9924	0.7321	0.8661
DA Test	0.8464	0.7018	0.8196
kNN Train	0.9263	0.8424	0.8710
kNN Test	0.9000	0.8107	0.8268

Table 5.6: Feature selection where 100 features out of 6000 are chosen with the MRMR feature selection method.

Table 5.5 and table 5.6 are not identical, even though both methods used the top 100 ranked features by the respective methods, which means the two feature selection methods do not value the same features in the same order. This could in turn indicate that the chi-squared method should be ignored, since the MRMR gives a better result. However, a disadvantage to MRMR is the time required. For the case of 2800 measurements and 6000 features, performing MRMR requires in the neighbourhood of half an hour. On the other hand, the chi-squared feature selection for the same data is completed in a fraction of a second. It would be ideal to keep the precision of MRMR while having the low computational cost of the chi-squared method. The computational cost of the MRMR method is proportional to the amount of features squared, so if chi-squared can be used first to lower the amount of features, while keeping most if not all of the features MRMR would select, then the two feature selection methods can be used in series to reach the same result at a much lower computational cost. As an example, if chi-square is specified to select the 1000 features out of 6000 that it itself ranks the highest, and MRMR is then specified to select the top 100 features out of those remaining 1000 features, the same result as in table 5.6 was obtained. This time, the amount of features the MRMR method was given had been reduced sixfold, so the calculation time went from half an hour to less than a minute.

In fig. 5.27, the differences in score based on the rank of the feature can be seen. The grey areas indicate the cut-off for which features are selected, and fig. 5.27a uses two shades of grey to differentiate the 100 and 1000 feature marks. The scores for the rank 1 and rank 100 feature in fig. 5.27a are of comparable value, which would indicate that relevant data is being forgotten by choosing the cut-off at 100 features. On the other hand, fig. 5.27b shows a significant difference

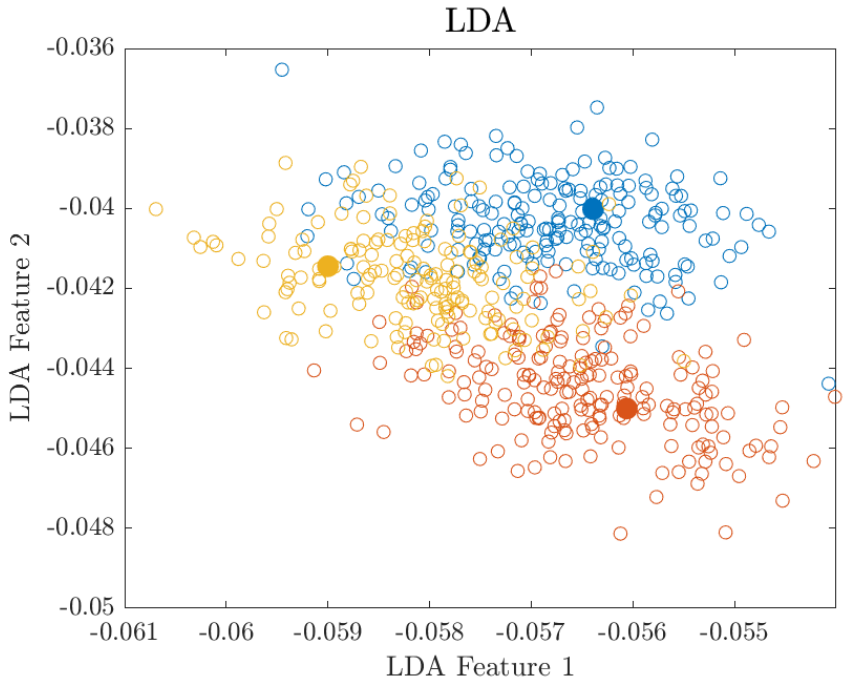


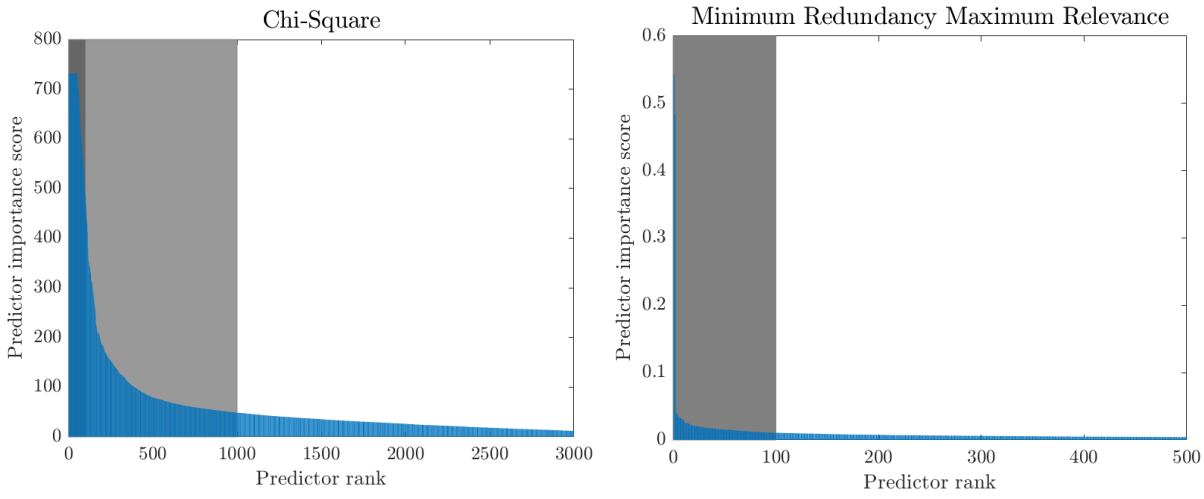
Figure 5.26: Visual representation of the training and testing measurements plotted after LDA has been performed with the settings used in table 5.3. The filled dots have been enlarged in order to be identified more easily.

in the scores of rank 1 and rank 100 features; indicating that, by selecting 100 features with the MRMR method, the important data is well covered. If instead chi-squared is being used to select the top 1000 features, a major difference between the new maximum and minimum scores is observed.

The overall highest feature selection-based accuracy is found when chi-squared and MRMR are performed in series. First chi-squared is used to select the best 1000 features out of the original 6000 features, then MRMR is used to select the best 400 features out of the selected 1000 features. The accuracies from these selections can be seen in table 5.7.

Algorithm	Selection	PCA	LDA
NB Train	0.6661	0.7156	0.9241
NB Test	0.6393	0.6839	0.8232
DA Train	1.0000	0.6839	0.9268
DA Test	0.6482	0.6643	0.8464
kNN Train	0.9179	0.8433	0.9281
kNN Test	0.8893	0.8161	0.8500

Table 5.7: Feature selection where first 1000 features are selected with chi-squared, then 400 features are selected with MRMR.



(a) (b)

Figure 5.27: Plots showing the scores for the features kept (grey background) and showing the score difference for the features removed in tables 5.5 and 5.6. The graphs have a limited x-axis for better visibility.

5.3.2 Feature Transformation Results

A 2D visual representation of the feature transformation results by PCA and LDA based on the settings in table 5.7 can be seen in fig. 5.28. For this, 28 different samples were measured, each with the standard 100 repetitions. The 28 samples have been classified as `SiGlu`X`ng` where X is the amount of glucose that has been deposited on a clean silicon wafer in nanograms. In this case, the 28 samples have been grouped in three classes. The 20 ng and 100 ng classes each contain 10 samples while the 60 ng class contain 8 samples. The difference between the samples of the same class are the conditions for which the measurements were taken. Conditions include whether or not the measurement chamber was purged of water vapour, if the sample was rotated, or if it was moved slightly between samplings.

For fig. 5.28a, the testing measurements hardly have any variation from the training data, while in fig. 5.28b, outliers for both training and testing measurements can be seen. However, with PCA being unsupervised several more clusters are created, as moving the sample, or adding water vapour can be seen in the measured data. On the other hand, since LDA is supervised, the different samples that lie within the same class are 'forced' together into a single cluster, as seen in fig. 5.28b. Since nanograms of glucose distributed on a silicon wafer is a low concentration, noise and other factors can impact the ability for LDA to make compact clusters, hence the larger spread seen in fig. 5.28b compared to the LDA example in fig. 4.5b.

PCA and LDA can also give very similar results, as seen in fig. 5.29. Here, four samples of different structures and sizes are compared. The two antennas are very similar in size, yet both feature transformation methods are able to distinguish between all four with 100% accuracy from the classifiers.

While, as a feature transformation method, PCA might be less accurate than LDA when it comes to classification, PCA can still prove useful to the user. In fig. 5.30, PCA and LDA are

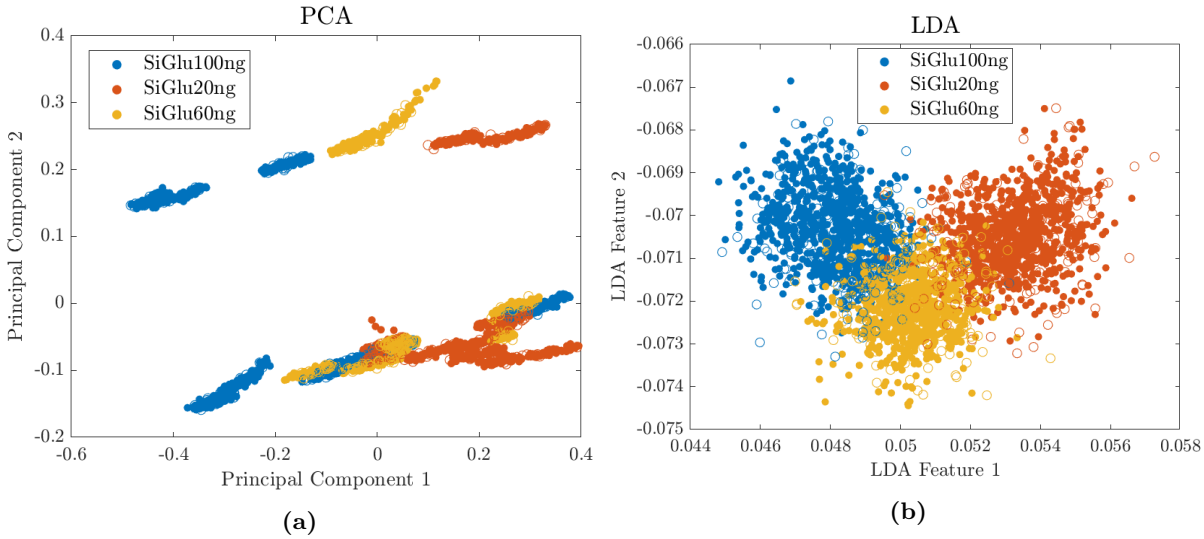


Figure 5.28: Feature transformation by PCA and LDA for silicon wafer samples with glucose deposited on them. The filled markers represent training data, and the empty markers represent testing data.

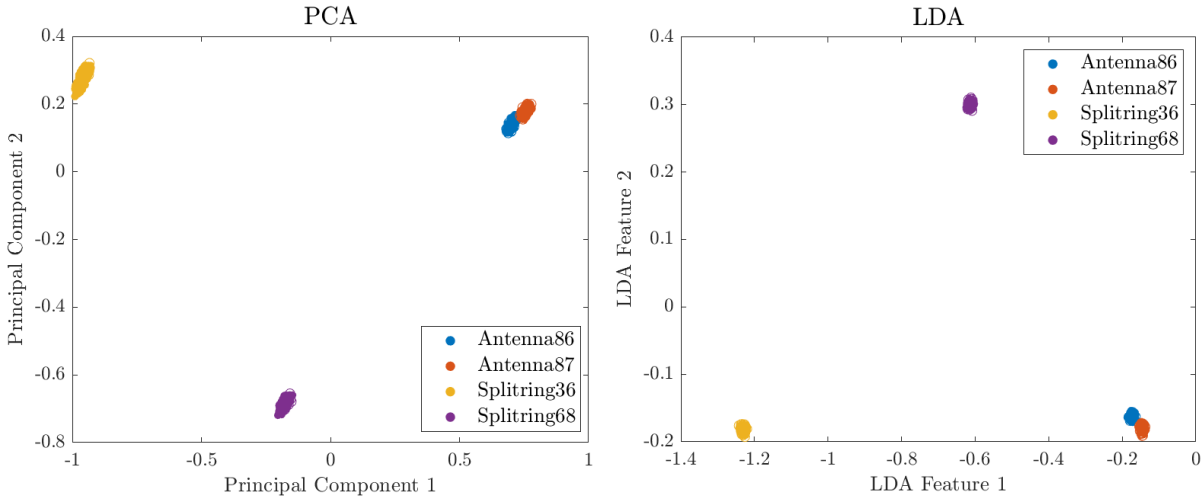
used to perform the same task, and the results are noticeably different. Figure 5.30b shows four clusters; some located closer to each other than others. PCA in fig. 5.30a shows a larger spread in the data, which indicates that it has a hard time finding significant patterns, however, `SiGlu20ng` and `Sifresh` are clearly split in two. The reasons could be several; noise could be of significant magnitude compared to the measurement differences, a sample could have been labelled incorrectly, or it could be an indication of the measurement setup being slightly different, such as the THz beam not hitting the centre of the sample. Thus, an advantage of PCA being unsupervised is its ability to show the user differences that LDA hides away in its clusters.

Even though the accuracy of PCA is lower than that of LDA when it comes to a means of dimensionality reduction, PCA does have a use in improving the overall accuracy for the classifiers. Table 5.8 shows five different approaches to DR and the accuracies associated with each approach. The numbers in the parentheses are the number of features after the DR method has been used. The columns are sorted in order of increasing accuracy; showing how using PCA and LDA in series as a means of DR is effective, and that using chi-squared before PCA has some, but smaller, effect. Though, performing chi-squared feature selection prior to PCA does lower the computational time needed.

5.3.3 Classifier Results

The naive Bayes classifier can be used with various types of density distributions. To compare two types of distribution, table 5.9 was created. This table contains the accuracies for the NB classifier under the same settings seen previously, except with the kernel density distribution used. It can be seen that the accuracies do increase with the kernel setting, though, not in a significant manner.

Table 5.10 shows the difference in the type of discriminant analysis classifier used. For the most part, QDA and LDA are similar in accuracy, and the most significant difference is in the second



(a) (b)

Figure 5.29: Feature transformation by PCA and LDA for silicon wafers with gold nanostructures of different shapes and sizes.

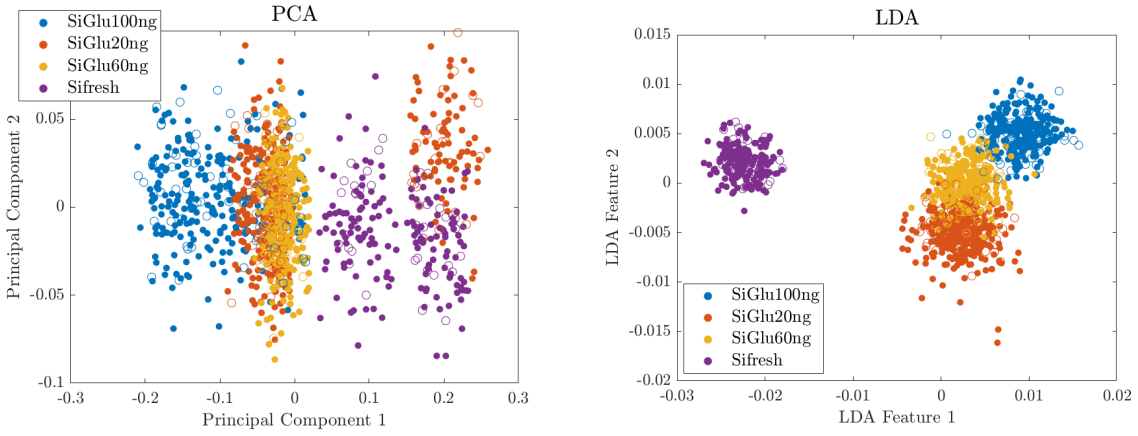
Algorithm	Chi2 (400)	Chi2 (1000) MRMR (400)	Chi2 (1000) PCA (200)	PCA (200)	Chi2 (3000) PCA (200)
NB Train	0.9268	0.9241	0.9232	0.9420	0.9438
NB Test	0.8321	0.8232	0.8929	0.9143	0.9232
DA Train	0.9308	0.9268	0.9223	0.9451	0.9446
DA Test	0.8268	0.8464	0.9000	0.9143	0.9196
kNN Train	0.9317	0.9281	0.9277	0.9491	0.9482
kNN Test	0.8321	0.8500	0.9036	0.9143	0.9214

Table 5.8: Accuracies of the different classifiers when DR is performed in series. The five last columns are five different DR approaches followed up by LDA. The numbers in the parentheses are the number of features after the DR method has been used.

column, where it appears that QDA has overfitted the data and LDA did not. The last two columns show a slight advantage to QDA and are the two columns with highest overall accuracy.

With the k-nearest neighbour classifier, the k refers to the number of nearest neighbours that are in consideration. To compare different k-values, table 5.11 was created. In this table, columns two and three show a noticeable drop in accuracy as the k-value increases, while the last two columns are much less affected. The data used is presented in fig. 5.28, where data is overlapping. It appears being flexible, as with a low k-value, gives an advantage.

In order to visualise the decisions made by a classifier, decision maps can be created, and figs. 5.31 to 5.33 contain such decision maps. They are created by using the classifiers to predict the class of every coordinate within the relevant feature space. In fig. 5.31, the decision maps for the standard settings used in this project are shown. Here, similarities and differences between the classifiers can be seen. Figure 5.31b shows the three classifiers to be rather similar, where the biggest difference is that kNN does not have smooth lines between the decision areas. In fig. 5.31a, the decision maps are less similar. The NB decision map consists of just three regions,



(a) (b)

Figure 5.30: Feature transformation by PCA and LDA for silicon wafers with glucose in nanograms, or no glucose at all.

Algorithm	Chi2 (1000) MRMR (400)	Chi2 (1000) MRMR (400) PCA	Chi2 (1000) MRMR (400) LDA	Chi2 (1000) PCA (200) LDA
NB-Normal Train	0.6661	0.7156	0.9241	0.9232
NB-Normal Test	0.6393	0.6839	0.8232	0.8929
NB-Kernel Train	0.7643	0.7705	0.9304	0.9259
NB-Kernel Test	0.7179	0.7607	0.8357	0.9000

Table 5.9: Naive Bayes classifier used with kernel density distribution, compared to the naive Bayes classifier with normal density distribution.

which are fitted to best match the principal components from PCA. The DA decision map has four regions; adapting slightly to how the data visualised in the upper left corner is split. Lastly, the kNN decision map appears to be much more adaptive to the split data of the PCA method, creating several small decision areas.

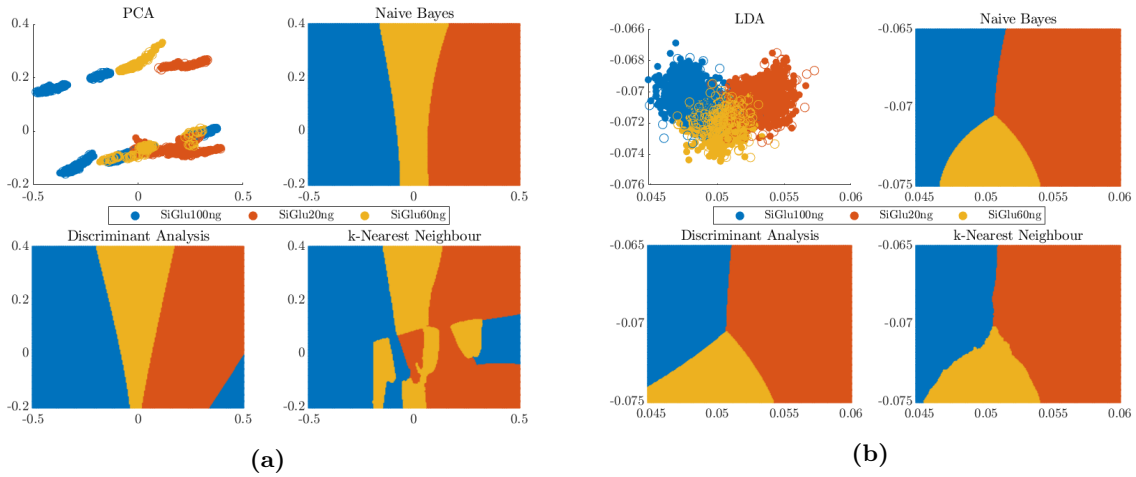
Tables 5.9 to 5.11 show the differences in accuracies when using alternative settings. The differences in decision maps for these alternative settings can be seen in figs. 5.32 and 5.33. The NB decision maps in fig. 5.32 are much more flexible compared to the ones in fig. 5.31. For LDA, the lines are less smooth, but the overall shape is mostly the same, and for PCA, it is

Algorithm	Chi2 (1000) MRMR (400)	Chi2 (1000) MRMR (400) PCA	Chi2 (1000) MRMR (400) LDA	Chi2 (1000) PCA (200) LDA
QDA Train	1.0000	0.6839	0.9268	0.9223
QDA Test	0.6482	0.6643	0.8464	0.9000
LDA Train	0.9259	0.6920	0.9259	0.9223
LDA Test	0.8125	0.6768	0.8125	0.8875

Table 5.10: Discriminant analysis classifier with linear and quadratic discriminant types.

Algorithm	Chi2 (1000) MRMR (400)	Chi2 (1000) MRMR (400) PCA	Chi2 (1000) MRMR (400) LDA	Chi2 (1000) PCA (200) LDA
3-NN Train	0.9638	0.9259	0.9549	0.9438
3-NN Test	0.9268	0.8804	0.8893	0.8964
17-NN Train	0.9451	0.8924	0.9321	0.9295
17-NN Test	0.9250	0.8607	0.8500	0.9054
47-NN Train	0.9179	0.8433	0.9281	0.9277
47-NN Test	0.8893	0.8161	0.8500	0.9000
149-NN Train	0.8487	0.7795	0.9263	0.9277
149-NN Test	0.8411	0.7607	0.8554	0.8982
301-NN Train	0.7451	0.7255	0.9250	0.9277
301-NN Test	0.7179	0.6964	0.8554	0.8982

Table 5.11: The k-nearest neighbour classifier for five different values of k.



(a) **(b)**

Figure 5.31: Decision maps under standard settings with Chi2(1000) and MRMR(400).

more significant. The NB decision map in fig. 5.32a is sporadic in order to best match the split PCA data, and table 5.9 also shows a larger increase in accuracy for PCA than for LDA, which makes sense looking at how the decision map has adapted with the kernel setting.

The decision maps made with a linear discriminant in the DA classifier in fig. 5.32 show straight lines, whereas the quadratic discriminant in fig. 5.31 show curved lines. Intuitively, straight lines are worse at representing clusters, which also matches the finds in table 5.10.

The kNN decision maps for PCA in figs. 5.31 to 5.33 show how an increase in k-value makes the kNN classifier lose its flexibility, as fewer regions in the plot are present and accuracy is lost. Table 5.11 agrees with this, showing a large increase in accuracy for small k-values when using PCA. For the kNN decision maps for LDA in figs. 5.31 to 5.33, the differences are less significant. In all five cases, only three regions are present, with only a small number of outlying dots at small k-values. The borders between the decision areas are the most noticeable difference, where a large k-value straightens out the borders, and a low k-value curves and bends the borders.

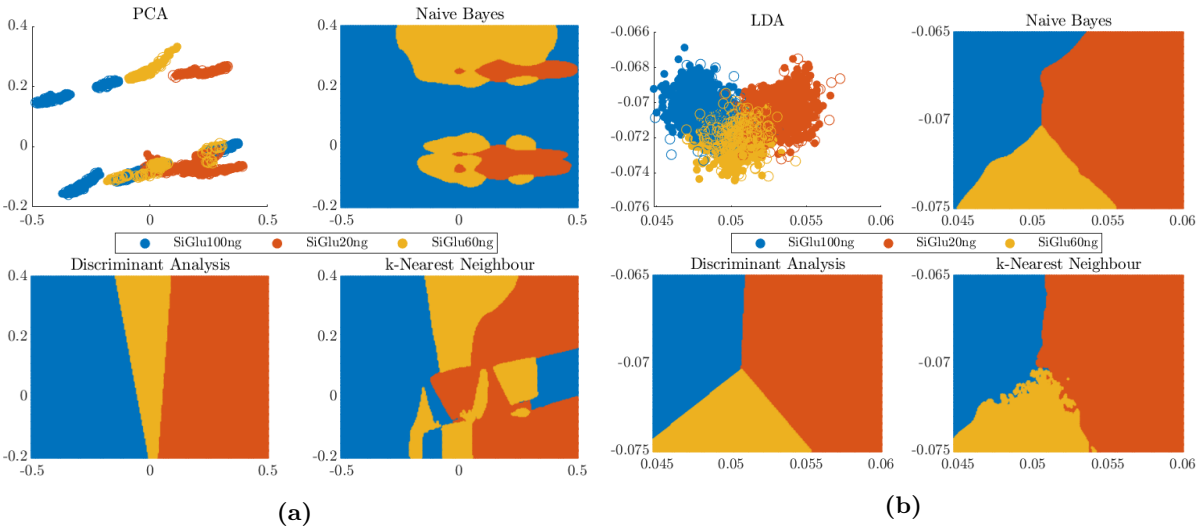


Figure 5.32: Decision maps for PCA and LDA where NB uses kernel density distribution, DA uses linear discriminant type, and kNN uses a k-value of 3 with Chi2(1000) and MRMR(400).

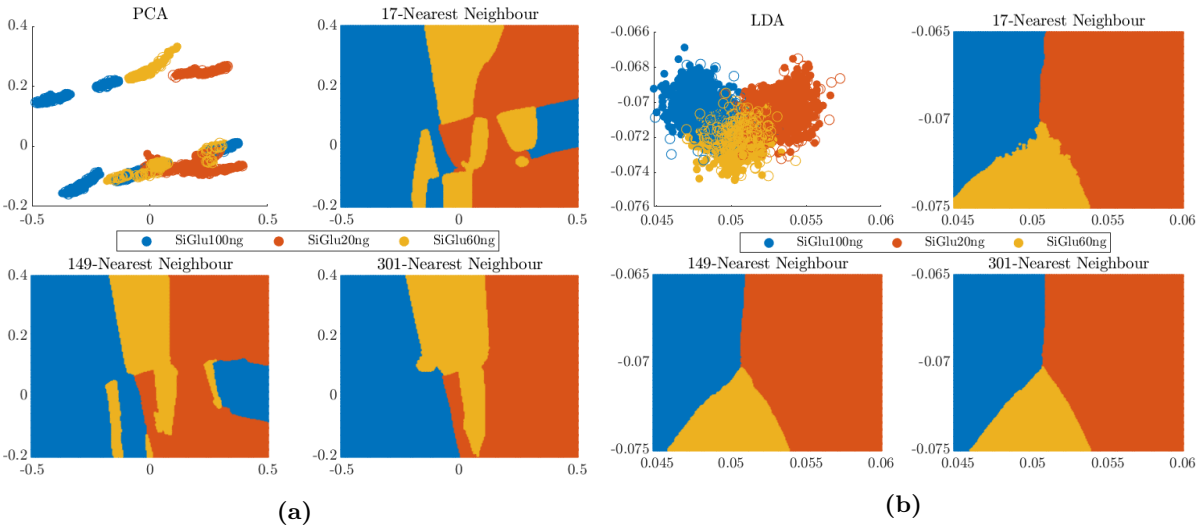


Figure 5.33: Decision maps for PCA and LDA with triple kNN, with k-values of 17, 149, and 301 with Chi2(1000) and MRMR(400).

5.4 Experimentally Based Machine Learning

Based on the results in section 5.3, the standard machine learning program will use three categories, selection, PCA, and LDA. The algorithm order in the three categories are: Chi2(1000), Chi2(1000) → PCA, and Chi2(1000) → PCA(200) → LDA, respectively. Several samples of different shapes and dimensions were created. The first thing to test with machine learning is whether the program is able to distinguish between the different samples when they are being measured under the same conditions in the THz spectrometer setup. Figure 5.34 shows the visual part of a machine learning calculation, where both PCA and LDA are able to distinguish between all samples to a high degree. The classifier accuracies can be seen in table 5.12; showing perfect predictions for LDA. To indicate that the results shown by PCA and LDA are not random, consider how antennas of similar size are plotted within close distance, especially `Antenna86` and `Antenna87`. Both `Splitring36` and `Splitring68` are plotted far from the other samples, indicating that the shape of the structure of the sample is significant.

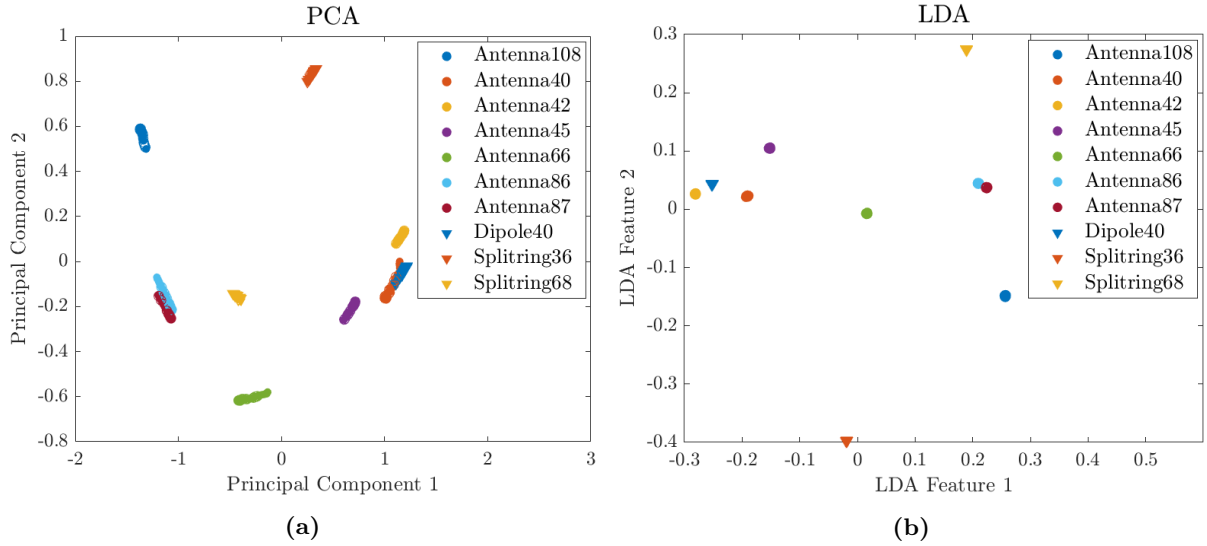
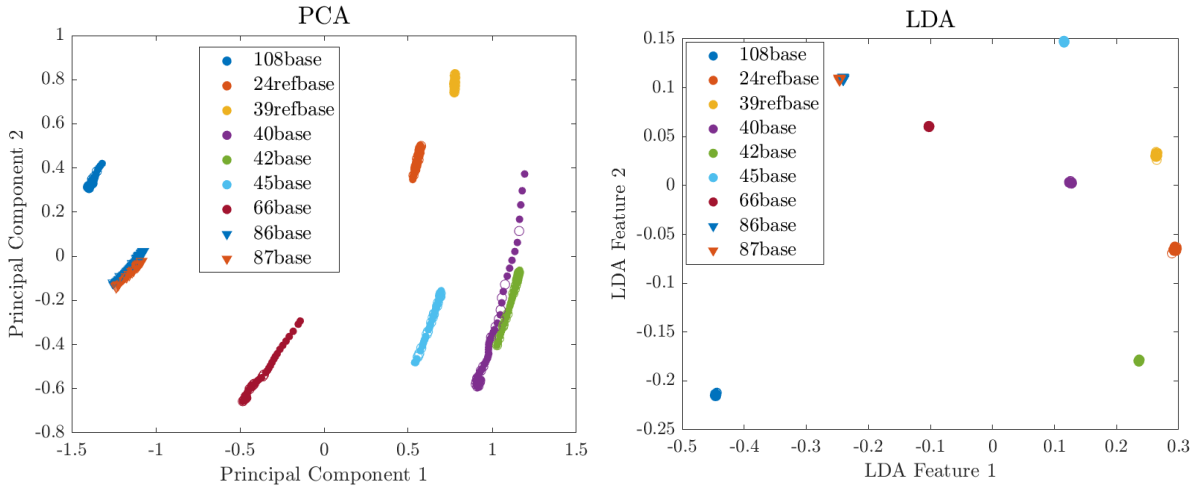


Figure 5.34: PCA and LDA plots of feature transformation for nanostructures with different shape and size.

The samples plotted in fig. 5.35 are made to differentiate between samples with antennas resonating with the THz waves and antennas rotated 90° in order to not have a resonance with the THz waves. In the naming scheme, the number refers to the length of the antennas on the sample, `refbase` refers to samples rotated 90°, and `base` refers to samples measured under standard conditions. In both fig. 5.35a and fig. 5.35b, the `86base` and `87base` samples are near each other; suggesting that similar data will be plotted closely. In fig. 5.35a, `20refbase` and `38refbase` are in the same region of the graph, but not as closely plotted. The same applies to fig. 5.35b, where `20refbase` and `38refbase` are near each other; indicating a similarity. Additionally, they are generally further away from the rest of the samples than the `base` samples are from each other. The accuracies in table 5.13 again show a perfect accuracy for LDA, and, when using the kNN classifier, a near perfect accuracy for only feature selection and PCA.



(a) (b)

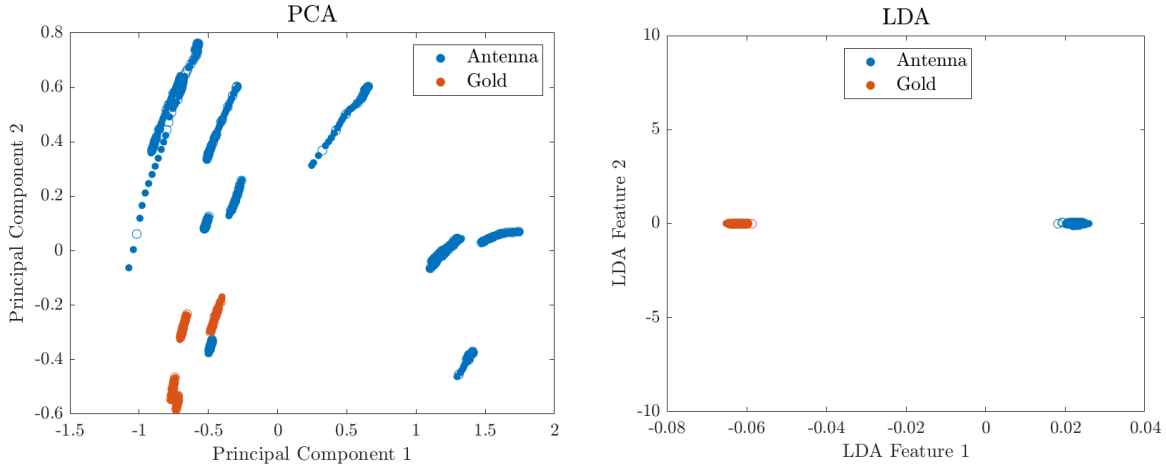
Figure 5.35: PCA and LDA plots for antennas with various lengths, where some samples are rotated by 90° as a reference.

Algorithm	Selection	PCA	LDA
NB Train	1.0000	0.9100	1
NB Test	1.0000	0.9150	1
DA Train	0.8000	1.0000	1
DA Test	0.8000	1.0000	1
kNN Train	0.9750	0.9800	1
kNN Test	0.9850	0.9900	1

Table 5.12: Table of accuracies for nanostructures with different shapes and sizes.

Algorithm	Selection	PCA	LDA
NB Train	1.0000	0.9167	1
NB Test	1.0000	0.9222	1
DA Train	0.8889	1.0000	1
DA Test	1.0000	1.0000	1
kNN Train	0.9861	0.9236	1
kNN Test	0.9967	0.9222	1

Table 5.13: Table of accuracies for antennas with various length, with some of them being rotated by 90°.



(a) (b)

Figure 5.36: PCA and LDA plots for antennas of different lengths, denoted as `Antenna`, with some being rotated by 90°, labelled as `Gold`.

The classes in fig. 5.35 can be simplified to `Antenna` and `Gold`, where `Gold` are antennas that have been rotated 90°. This have been done in fig. 5.36, where fig. 5.36a is the same graph as in fig. 5.35a, just with different colouring. Though, the LDA plot in fig. 5.36b is much different. It should be noted that, since there are only two classes, the LDA plot should be limited to a single dimension, for which the y-axis limits have been adjusted to make it appear as in 1D only. The figure shows that LDA creates a large between-class distance, and the perfect accuracies in table 5.14 reaffirm this.

In fig. 5.37, another class has been added, such that the `Antenna` and `Gold` classes are now being compared to an empty silicon wafer labelled `Wafer`. PCA shows the empty wafer measurements as being in the same area of the plot as the `Gold` measurements, indicating a larger difference between `Wafer` and `Antenna` than between `Wafer` and `Gold`, as expected. In the LDA plot, it seems that the three classes are roughly equally spaced from each other, however, LDA features (and principal components) are sorted in decreasing order of importance, so LDA feature 1 is more significant than LDA feature 2. `Gold` and `Wafer` have similar LDA feature 1 position, and have a large spacing between them and `Antenna`, suggesting that the resonance of the antennas is clearly noticeable in the data. The associated accuracies in table 5.15 show a perfect accuracy for LDA, and again a near perfect accuracy for the kNN classifier for all three cases.

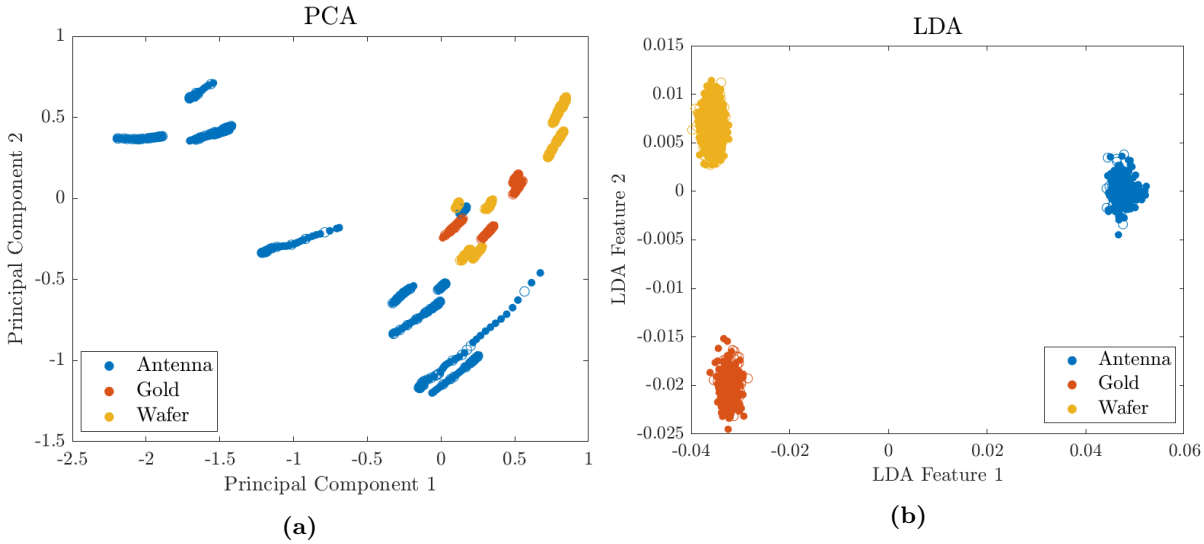


Figure 5.37: PCA and LDA plots for antennas, compared to references rotated by 90°, and to empty silicon wafers.

Algorithm	Selection	PCA	LDA
NB Train	1.0000	0.9175	1
NB Test	0.9967	0.9300	1
DA Train	1.0000	0.9333	1
DA Test	0.7600	0.9333	1
kNN Train	0.9983	0.9992	1
kNN Test	1.0000	0.9967	1

Table 5.14: Table of accuracies for machine learning classification of `Antenna` and `Gold`, as per fig. 5.36.

Algorithm	Selection	PCA	LDA
NB Train	0.9644	0.7289	1
NB Test	0.9673	0.7423	1
DA Train	1.0000	0.7851	1
DA Test	0.5808	0.7865	1
kNN Train	0.9990	0.9986	1
kNN Test	1.0000	0.9962	1

Table 5.15: Table of accuracies for machine learning classification of `Antenna`, `Gold`, and `Wafer`, as per fig. 5.37.

Figure 5.38 shows an additional class being added, namely `Air`, which is a set of measurements taken with no sample - just air. Adding `Air` appears to have a larger effect than adding `Wafer`; now `Air` is the class that is different from the others along the x-axis. In fact, they are so far away on the x-axis that the volume of the clusters has been compacted into just a line. It has thus gotten harder to differentiate between `Gold` and `Wafer`, even in the LDA plot, and when looking at the accuracies in table 5.16 LDA no longer have perfect predictions, albeit they are close to perfect.

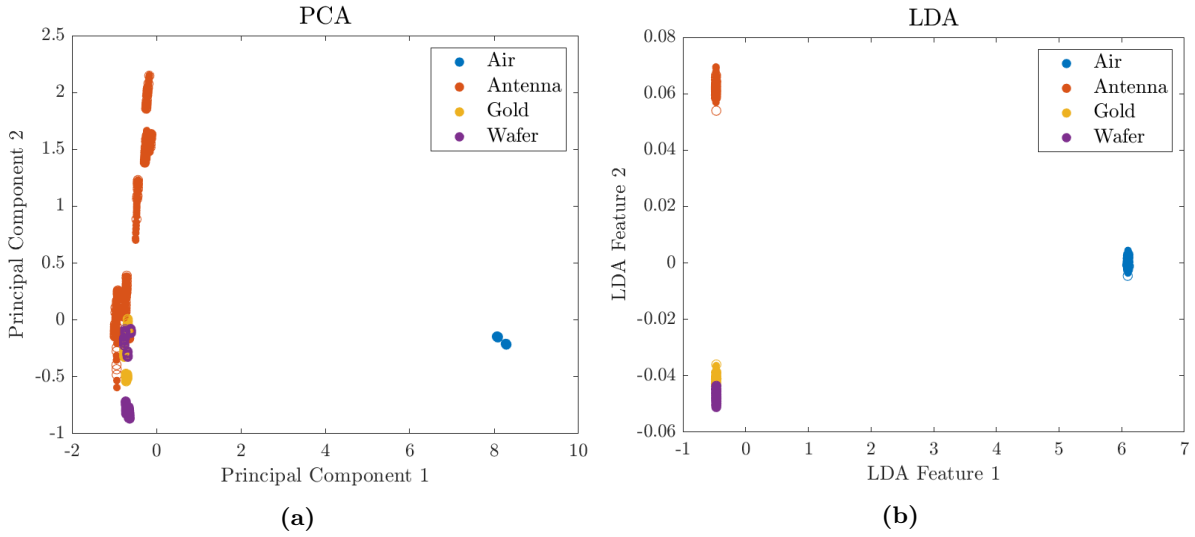


Figure 5.38: PCA and LDA plots for antennas, rotated antennas, silicon wafers, and air.

To improve upon the plots in fig. 5.38, the dimensionality can be increased to 3D. When doing this, the same information as in fig. 5.37 can be seen when considering the y- and z-axis of the plots in fig. 5.38. The accuracies as per table 5.17 improve, putting LDA back at a perfect accuracy.

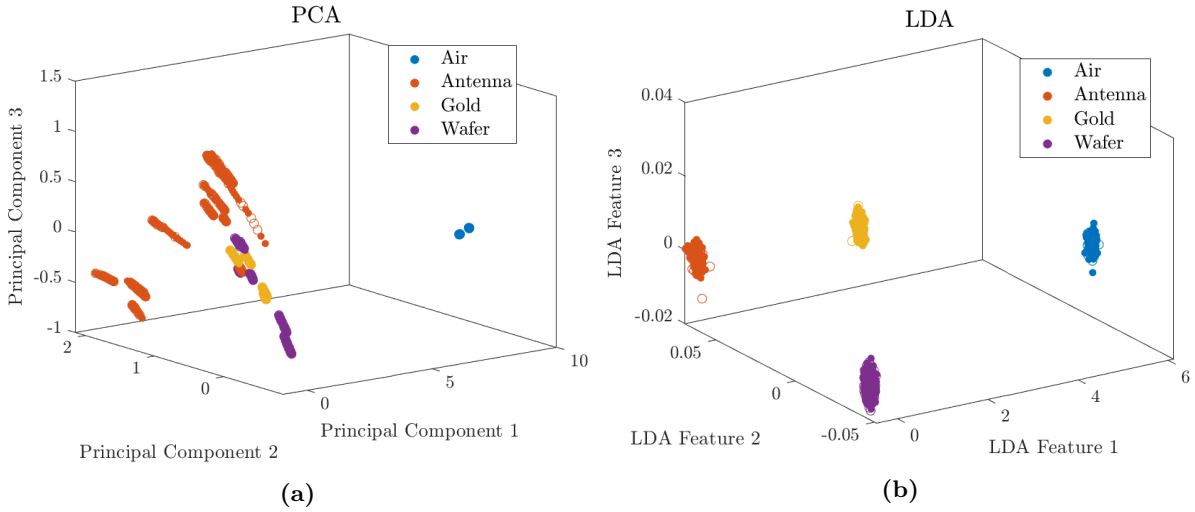


Figure 5.39: PCA and LDA plots for antennas, rotated antennas, silicon wafers, and air, plotted in 3D to increase the accuracy.

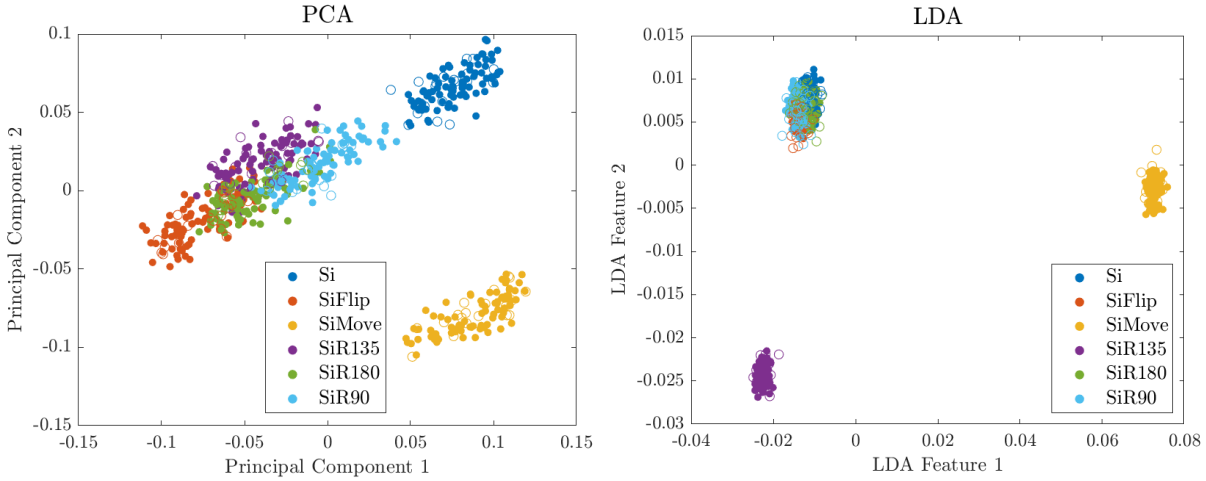
Algorithm	Selection	PCA	LDA
NB Train	0.9353	0.7679	0.9951
NB Test	0.9464	0.7464	0.9946
DA Train	0.6433	0.8174	0.9946
DA Test	0.6661	0.8089	0.9946
kNN Train	0.9991	0.9978	0.9951
kNN Test	0.9982	0.9929	0.9946

Table 5.16: Table of accuracies for classification of `Antenna`, `Gold`, `Wafer`, and `Air`, related to fig. 5.38.

Algorithm	Selection	PCA	LDA
NB Train	0.9353	0.8201	1
NB Test	0.9464	0.8161	1
DA Train	0.6433	0.9545	1
DA Test	0.6661	0.9589	1
kNN Train	0.9991	0.9991	1
kNN Test	0.9982	0.9964	1

Table 5.17: Table of accuracies for classification of `Antenna`, `Gold`, `Wafer`, and `Air` in 3D, related to fig. 5.39.

When a sample is being mounted in the measurement setup, removed, and then mounted again, there will be minor differences in the position of the sample. In fig. 5.40, this is being tested with a clean silicon wafer. Here, the wafer was flipped so its backside faced the beam, it was moved so the beam hit its edge, and it was rotated by various degrees. PCA shows that measurements made when the wafer is moved laterally with respect to the beam gives the most significant difference. LDA agrees that moving the wafer results in the most significant change, where rotating the wafer by 135° is the next most noticeable.



(a) **(b)**

Figure 5.40: PCA and LDA plots of a silicon wafer for various positional differences.

To compare how the signal artefacts from the placement of the sample compares to other types of noise, see fig. 5.41. Here the same placement conditions are tested and repeated, but this time they are tested in the presence of water vapour. The standard measurement procedure includes purging the sample area with nitrogen gas in order to remove water vapour, as water absorbs THz waves quite well. There are a lot of classes in fig. 5.41, but a few patterns are easily noticed. For both PCA and LDA, the measurements taken with water vapour present (denoted with `W` in the label) are on the left-hand side of the plots. The water vapour versus purged difference is mainly on the x-axis; suggesting it is more meaningful than moving the sample.

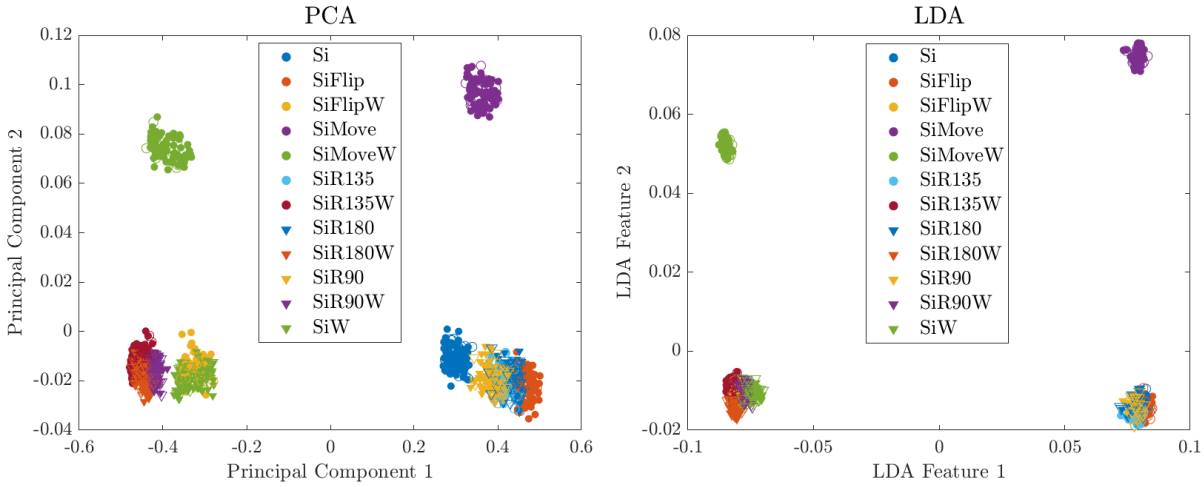


Figure 5.41: PCA and LDA plots for a silicon wafer for various positional differences, with and without water vapour present.

To test the field enhancement properties of the nanostructures on the samples, two sugars, glucose and sucrose, have been deposited on some of the samples. In fig. 5.42, various amounts of glucose have been deposited on the `L40` sample, which is a sample with antennas of length 40 µm. The figure also shows the measurements taken for `L40` without any sugars deposited on it. While the clusters are spread, they do not seem to follow a pattern, and some clusters are hard to distinguish between, even though they are expected not to be.

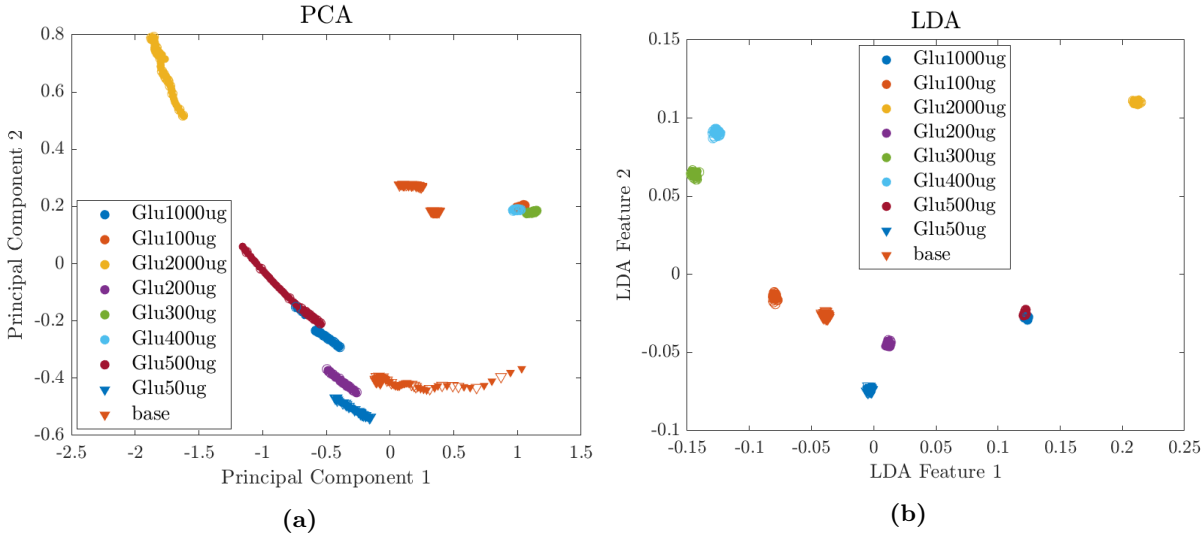
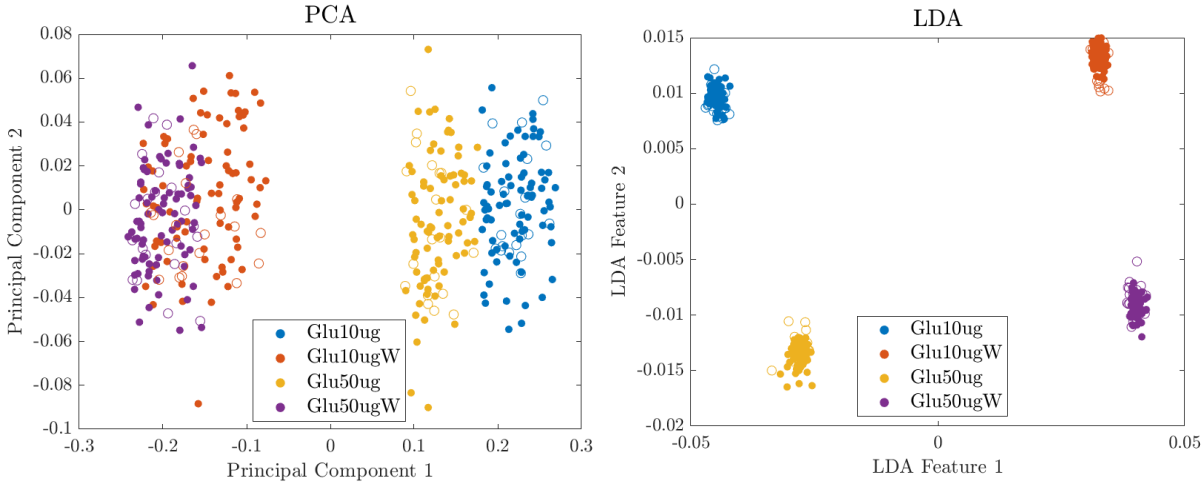


Figure 5.42: PCA and LDA plots of antennas with length 40 µm at various amounts of glucose deposited.

Glucose was also deposited on the `L39C` sample, the results of which is plotted in fig. 5.43. This time, the samples are measured with and without water vapour present, and at two different amounts of glucose deposited. In both plots, the measurements with and without water vapour

are split, however, in the PCA plot, the two water vapour measurements overlap somewhat, while LDA has no issue spreading out all four clusters.



(a) **(b)**

Figure 5.43: PCA and LDA plots for antennas of length 39 µm at two different amounts of glucose deposited, with and without water vapour present.

To challenge the machine learning program and test its capabilities, a larger portion of the database was used for fig. 5.44, namely all antenna samples with only glucose deposited on the sample. Included herein are samples with and without water vapour, samples with different antenna lengths, and all the antennas without any glucose as well - totalling 37 different samples corresponding to 3700 measurements. The glucose deposits on the samples range from 10 ng to 1000 µg, and the same weight have been deposited on samples with antennas of different length. In 2D, the LDA accuracies for the test data are 0.8716, 0.8811, and 0.8743 for NB, DA, and kNN, respectively. In 3D, the LDA accuracies for the same classifiers are 0.9095, 0.9068, and 0.8987, so fitting all this data into the 10 different classes does prove difficult. However, since 10 classes are at play, the maximum number of dimensions LDA can utilise has increased to 9D where the accuracy for all three classifiers is 0.9987 for the test data. The calculation time is practically the same, so the most noticeable disadvantage is that when going beyond 3D, the visual aspect is lost.

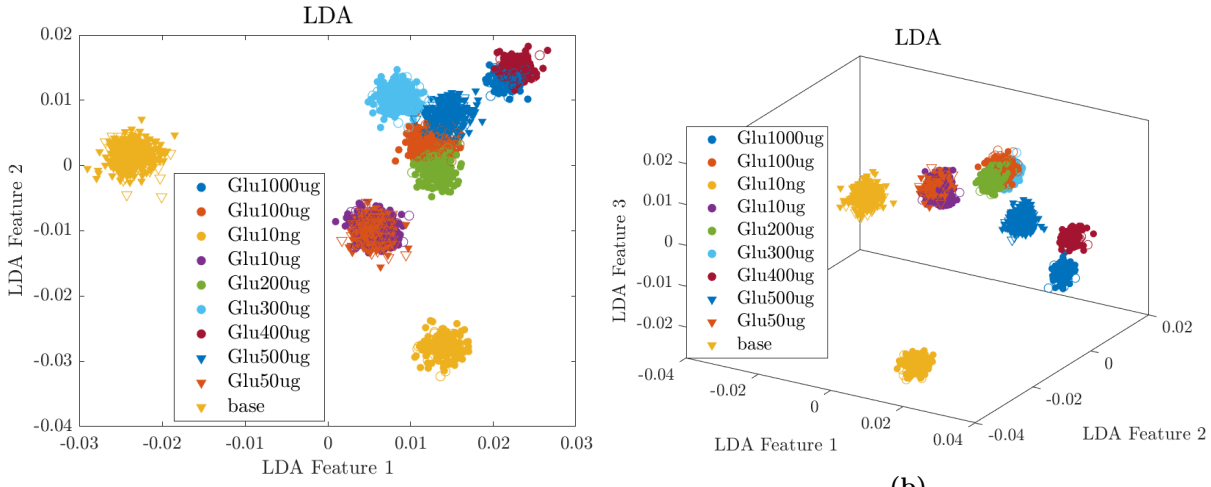


Figure 5.44: LDA plots in 2D and 3D for antennas with different length, at various amounts of glucose deposited, labelled by the amount of glucose deposited.

The second type of sugar used was sucrose, less samples were created with sucrose alone, but two different amounts of sucrose, with and without water vapour present, as well as a comparison with the sample without sucrose, can be seen in fig. 5.45. Here, it appears that the presence of water vapour has a smaller effect compared to the effect of differing sucrose amounts. This was done on the L40 sample.

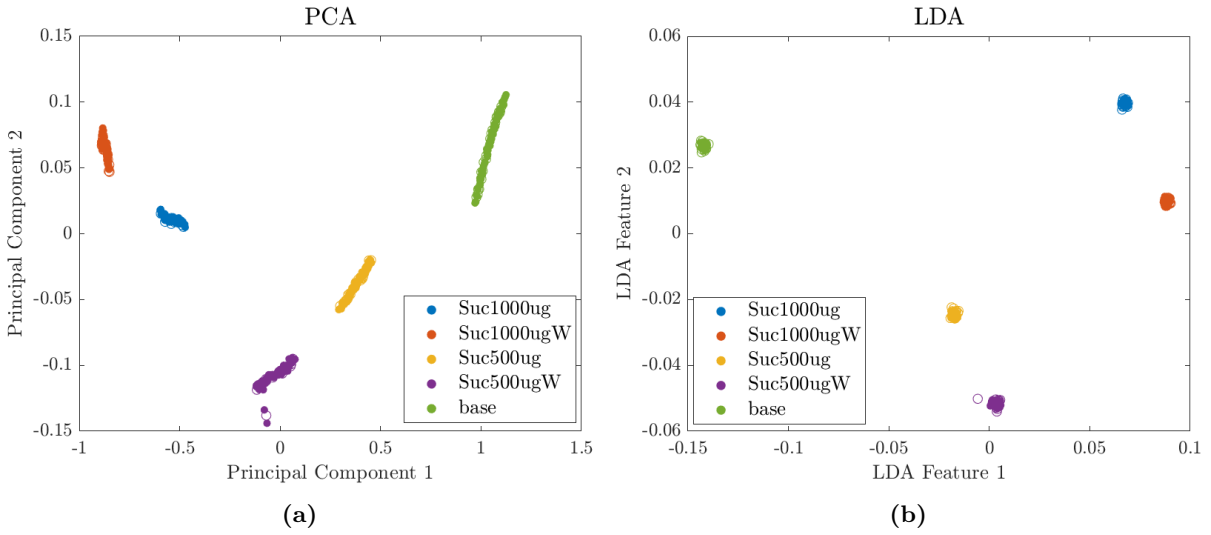


Figure 5.45: PCA and LDA plots of sucrose on antennas of length 40 µm at two different amounts of sucrose deposited, with and without water vapour present.

To again challenge the machine learning program, though this time in a different manner, measurements on the L40 sample with glucose, sucrose, a mix of the two, and no sugar are compared. Samples with water vapour present are included, but they are labelled in the same class as their purged counterpart. The result of this large comparison can be seen in fig. 5.46, where fig. 5.46a

shows LDA in 2D. The clusters are compact and, for the most part, nicely spread around the plot. There is a slight overlap between `Glu500ug` and `Glu1000ug`, and `Glu375Suc125` and `Glu250Suc250` are very close, but do not touch. The accuracy for the test set for NB, DA, and kNN are 0.9870, 0.9891, and 0.9870, respectively; showing a high accuracy even in just 2D. Figure 5.46b contains the decision maps for the three classifiers, the NB and DA decision maps show very similar results. The kNN decision map is different in the `Glu1000ug` and `Glu125Suc375` regions, but otherwise quite similar as well.

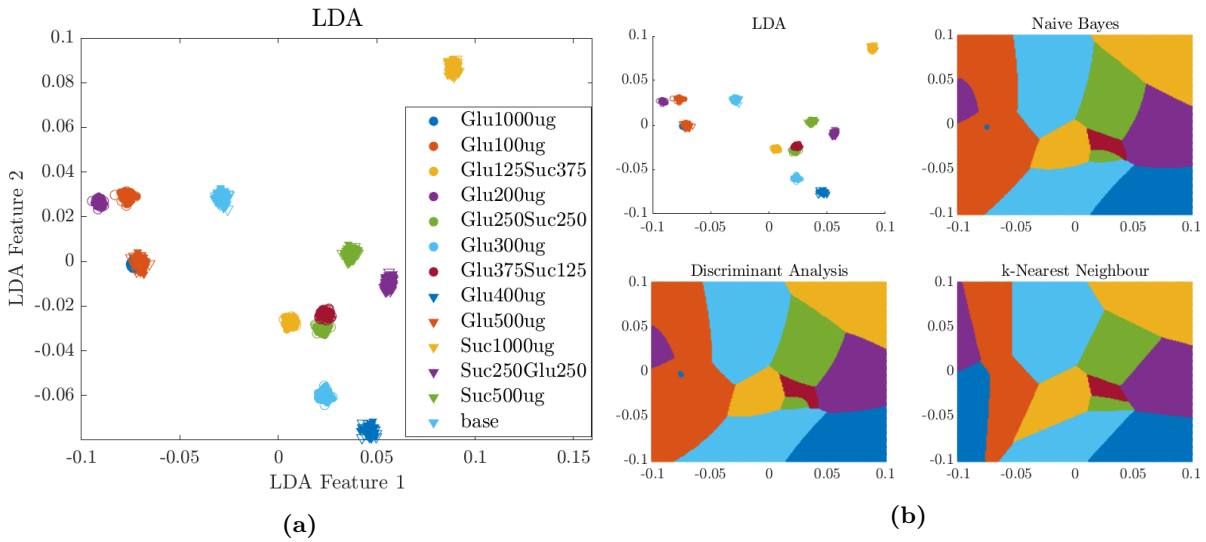
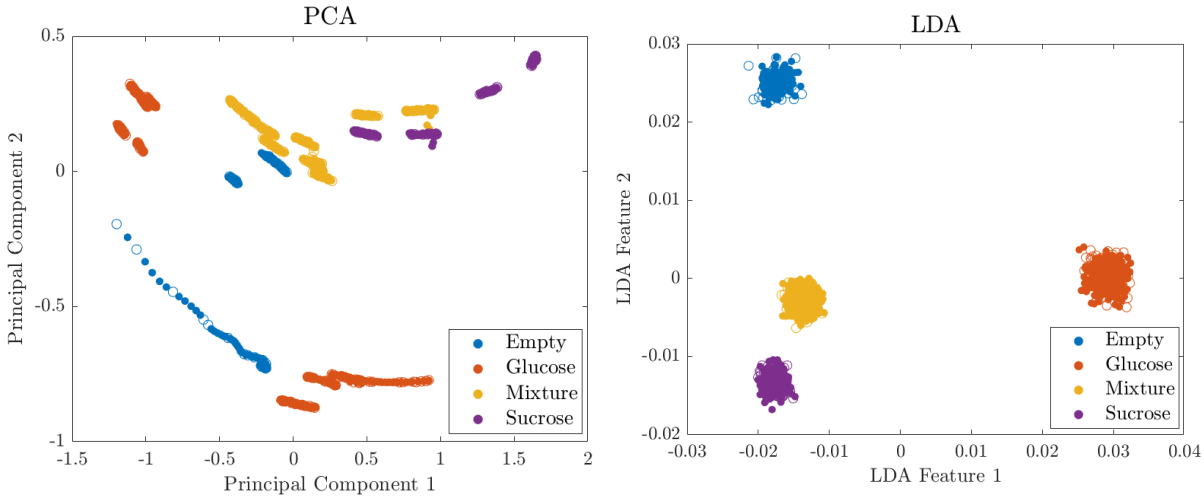


Figure 5.46: LDA plot and decision maps for the three classifiers for antennas of length 40 µm at various amounts of glucose and sucrose deposited on the sample. The legend in **(a)** applies to **(b)** as well.

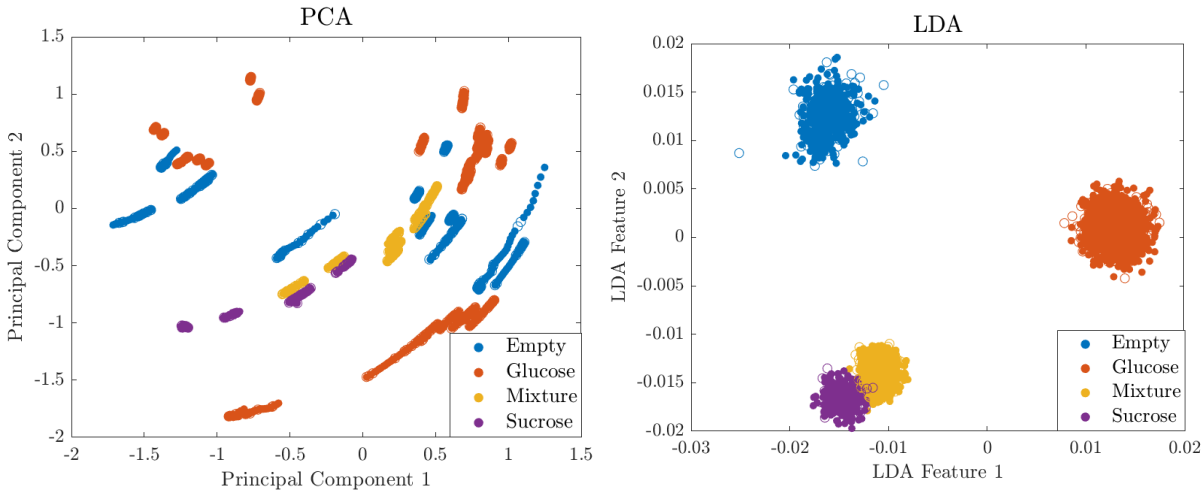
The classes in fig. 5.46 can be grouped into four general classes: Glucose, Sucrose, Mixture, and Empty. This have been done in fig. 5.47, where LDA now has no trouble distinguishing between the four clusters. In both the PCA and the LDA plot, `Sucrose` and `Mixture` are closer to each other compared to `Glucose` and `Mixture`, this could be due to the way the samples were handled, or, unexpectedly, it suggests that the signal from sucrose dominates the signal of glucose when mixed on the same sample.



(a) (b)

Figure 5.47: PCA and LDA plots of antennas with length 40 µm at different amounts of glucose and sucrose deposited, grouped into four classes.

The four-class category can be expanded to include all antenna lengths measured, with and without sugar, which is the case in fig. 5.48. Here, 50 different samplings for a total of 5000 measurements are compared. PCA shows that `Sucrose` and `Mixture` are similar in data, while `Glucose` and `Empty` vary a lot. LDA shows slight overlap between `Sucrose` and `Mixture`, but otherwise reaches an accuracy for 0.9980 for all three classifiers.



(a) (b)

Figure 5.48: PCA and LDA plots of antennas with various lengths at various amounts of glucose and sucrose deposited on the samples, grouped into four classes.

To visualise what the machine learning program is working with, see fig. 5.49. The two figures are the THz pulses in the time-domain, coloured by each class. In fig. 5.49a each class is offset in the y-axis, and in fig. 5.49b they are not. Humans are great at image recognition, which

is a stereotypical form of machine learning. However, in the case of spectra related machine learning, humans will need assistance in order to classify measurements at the same level as machine learning algorithms do.

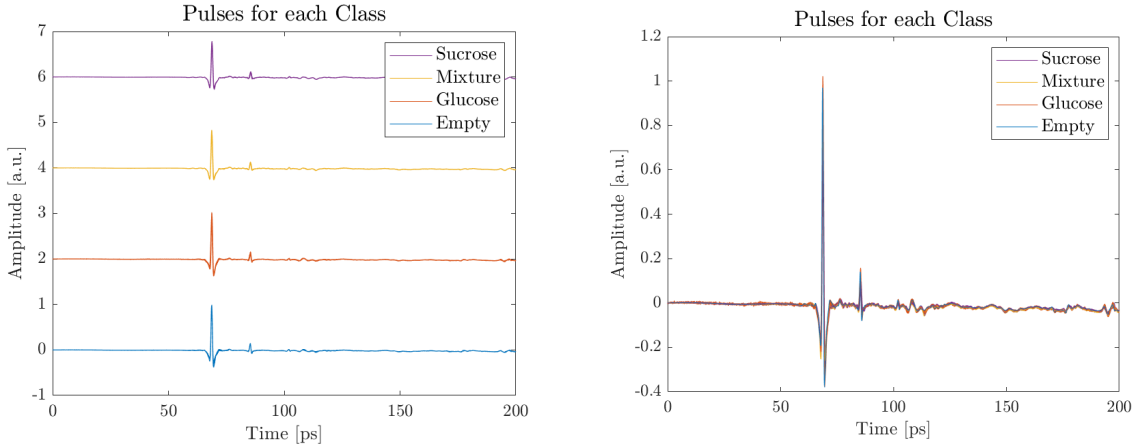


Figure 5.49: Plots of pulses for each class, **(a)** is offset along the y-axis for visibility while **(b)** is not.

Finally, to examine the field enhancement effect, see fig. 5.50. In fig. 5.50a tiny amounts of glucose have been deposited on an empty silicon wafer. The deposits are deposited via an aqueous solution, thus the reference, denoted with `Clean`, pure Milli-Q water was put on the wafer and dried off. The clean cluster lies alone, but the three clusters with nanograms of glucose deposited overlap somewhat, the accuracies of the test data for NB, DA, and kNN are 0.8516, 0.8484, and 0.8516, respectively. On the other hand in fig. 5.50b, tiny amounts of glucose were deposited on `L39` samples with the same approach. This time all three classifiers have a perfect accuracy, the four clusters are nicely spread, and they follow an intuitive pattern of smallest glucose deposit being the closest to the `Clean` class, suggesting the antennas do indeed enhance the field.

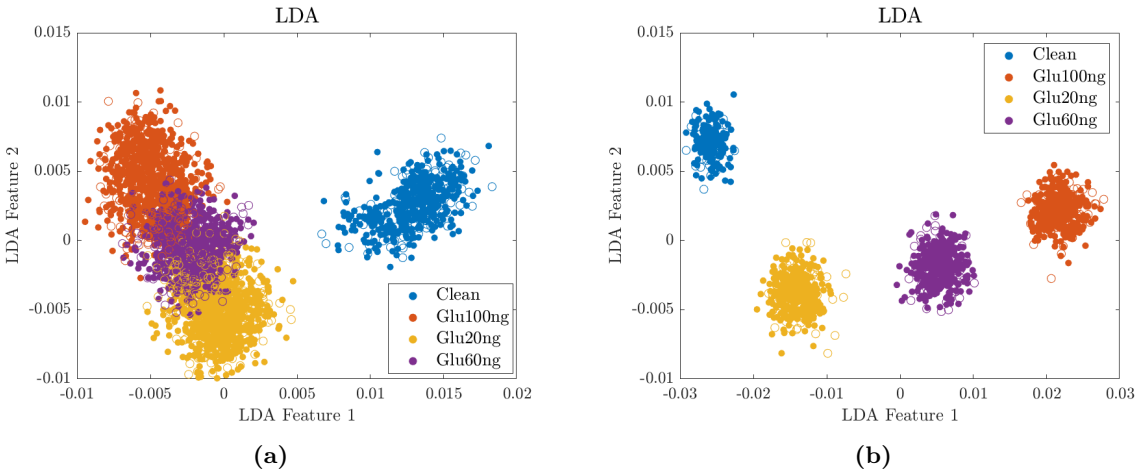


Figure 5.50: LDA plots of nanogram amounts of glucose deposited on samples, with **(a)** being on a empty silicon wafer and **(b)** being on antennas of length 39 µm.

6 | Conclusion

A finite-difference time-domain (FDTD) model was developed for modelling metamaterials. Periodic arrays of slits were modelled and compared to results from the Fourier modal method. The FDTD model was in good agreement with the Fourier model method. Periodic arrays of antennas were modelled and examined to optimise the antenna dimensions. The numerical results were found to agree with experimental results found in literature. Different shapes of antennas were examined, and it was found that, while the resonance frequency only changes slightly, the field enhancement changes significantly when modelling structures deviating from an ideal antenna, in this case a rectangular antenna. The field enhancement was found to be very localised to the surface at the ends of the antennas, and rapidly decreasing when moving away from the antenna. From the numerical results, it was found that decreasing the gap between antennas would not only increase the field enhancement, but also create a more evenly distributed field between the antennas.

The metamaterial samples used in this project were created using a lift-off technique with UV direct write and deposition using magnetron sputtering, creating arrays of gold antennas. The antennas were designed to match desired terahertz resonance frequencies, matching the absorption peaks of glucose and sucrose in the terahertz spectrum. To characterise the dimensions of the antennas a scanning electron microscope was utilised, which showed notable differences between the produced and ideal antennas. To account for the imperfect antennas, the model was successfully adapted to handle rounded and wavy antennas. The transmittance of the samples was measured with terahertz time-domain spectroscopy. The procedure was replicated to create and characterise split-ring resonators, thereafter the split-ring spectra were compared with the model. The produced resonators complied with the model further advocating for its efficacy.

To work with, analyse, and understand the terahertz spectroscopy data, a machine learning program was developed. The program makes use of several existing algorithms to use for dimensionality reduction, classification, and prediction of data. The parameters of the machine learning algorithms have been optimised to find which settings to use in order to get high-accuracy predictions.

The highest accuracy was found when using a four-step process; first the chi-squared feature selection method was used to select the most relevant data. Next, principal component analysis was used to transform the remaining features into a small number of high-information features, then linear discriminant analysis was used as a supervised method to transform the high-information features into even more compact features in 2D and 3D, giving a highly valuable visual insight into the process, and lastly machine learning algorithms were trained with training data and used to predict the classifications of test data.

Machine learning was successfully utilised to differentiate between data stemming from different nanostructures for various conditions. Two sugars, glucose and sucrose, were deposited on metamaterial samples to test the field enhancement effect. When compared to a reference, machine learning correctly predicted which sugars were used when. Nanogram amounts of glucose was deposited on a non-metamaterial sample to test the difficulty in accurately predicting the sample, and it was compared to the same nanogram amounts of glucose deposited on a sample with arrays of antennas, where the machine learning program could predict the classification of the measurements to a perfect degree.

Bibliography

[1] Dong-Kyu Lee, Ji-Hun Kang, Jun-Seok Lee, Hyo-Seok Kim, Chulki Kim, Jae Hun Kim, Taikjin Lee, Joo-Hiuk Son, Q-Han Park, and Minah Seo. Highly sensitive and selective sugar detection by terahertz nano-antennas. *Scientific reports*, 5(1):1–7, 2015.

[2] Shohreh Nouri Novin, Ferdows B Zarrabi, Maryam Bazgir, Samaneh Heydari, and Sepideh Ebrahimi. Field enhancement in metamaterial split ring resonator aperture nano-antenna with spherical nano-particle arrangement. *SILICON*, 11(1):293–300, 2018.

[3] Tom Driscoll, Hyun-Tak Kim, Byung-Gyu Chae, Bong-Jun Kim, Yong-Wook Lee, N Marie Jokerst, Sabarni Palit, David R Smith, Massimiliano Di Ventra, and Dimitri N Basov. Memory metamaterials. *Science*, 325(5947):1518–1521, 2009.

[4] Hironobu Fujiyoshi, Tsubasa Hirakawa, and Takayoshi Yamashita. Deep learning-based image recognition for autonomous driving. *IATSS research*, 43(4):244–252, 2019.

[5] Georgios Mastorakis. Human-like machine learning: limitations and suggestions. *arXiv preprint arXiv:1811.06052*, 2018.

[6] Thomas L Griffiths. Understanding human intelligence through human limitations. *Trends in Cognitive Sciences*, 24(11):873–883, 2020.

[7] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM JOURNAL OF RESEARCH AND DEVELOPMENT*, pages 71–105, 1959.

[8] Arthur L Samuel. Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, 11(6):601–617, 1967.

[9] H.T. Siegelmann and E.D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 1995.

[10] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. Citeseer, 2007.

[11] Muhammad Usama, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha. Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE access*, 7:65579–65615, 2019.

[12] Andrei V. Lavrinenko, Jesper Lægsgaard, Niels Gregersen, Frank Schmidt, and Thomas Søndergaard. *Numerical Methods in Photonics*. CRC Press, 2014.

[13] Bjorn Engquist and Andrew Majda. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of computation*, 31(139):629–651, 1977.

[14] Jean-Pierre Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, 1994.

[15] Gerrit Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Transactions on Electromagnetic Compatibility*, EMC-23(4):377–382, 1981.

[16] Robert L Higdon. Absorbing boundary conditions for difference approximations to the multi-dimensional wave equation. *Mathematics of computation*, 47(176):437–459, 1986.

[17] V. Betz and R. Mittra. Comparison and evaluation of boundary conditions for the absorption of guided waves in an fdtd simulation. *IEEE Microwave and Guided Wave Letters*, 2(12):499–501, 1992.

[18] O.M. Ramahi. The complementary operators method in fdtd simulations. *IEEE Antennas and Propagation Magazine*, 39(6):33–45, 1997.

[19] O.M. Ramahi. The concurrent complementary operators method for fdtd mesh truncation. *IEEE Transactions on Antennas and Propagation*, 46(10):1475–1482, 1998.

[20] Khaled ElMahgoub, Fan Yang, and Atef Elsherbeni. *Scattering Analysis of Periodic Structures Using Finite-Difference Time-Domain Method*. Morgan & Claypool, 2012.

[21] Fan Yang, Ji Chen, Rui Qiang, and Atef Elsherbeni. A simple and efficient fdtd/pbc algorithm for scattering analysis of periodic structures. *Radio Science*, 42(04):1–9, 2007.

[22] Weng Cho Chew and William H. Weedon. A 3d perfectly matched medium from modified maxwell's equations with stretched coordinates. *Microwave and Optical Technology Letters*, 7(13):599–604, 1994.

[23] Z.S. Sacks, D.M. Kingsland, R. Lee, and Jin-Fa Lee. A perfectly matched anisotropic absorber for use as an absorbing boundary condition. *IEEE Transactions on Antennas and Propagation*, 43(12):1460–1463, 1995.

[24] J.-P. Bérenger. Numerical reflection from fdtd-pmls: a comparison of the split pml with the unsplit and cfs pmls. *IEEE Transactions on Antennas and Propagation*, 50(3):258–265, 2002.

[25] M. Kuzuoglu and R. Mittra. Frequency dependence of the constitutive parameters of causal perfectly matched anisotropic absorbers. *IEEE Microwave and Guided Wave Letters*, 6(12):447–449, 1996.

[26] S.D. Gedney. Scaled cfs-pml: it is more robust, more accurate, more efficient, and simple to implement. why aren't you using it? In *2005 IEEE Antennas and Propagation Society International Symposium*, volume 4B, pages 364–367 vol. 4B, 2005.

[27] J. Alan Roden and Stephen D. Gedney. Convolution pml (cpml): An efficient fdtd implementation of the cfs–pml for arbitrary media. *Microwave and Optical Technology Letters*, 27(5):334–339, 2000.

[28] Stephen D. Gedney and Bo Zhao. An auxiliary differential equation formulation for the complex-frequency shifted pml. *IEEE Transactions on Antennas and Propagation*, 58(3):838–847, 2010.

[29] J.-P. Berenger. Perfectly matched layer for the fdtd solution of wave-structure interaction problems. *IEEE Transactions on Antennas and Propagation*, 44(1):110–117, 1996.

[30] Stephen D Gedney and A Taflove. The perfectly matched layer absorbing medium. *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method*, pages 263–344, 1998.

[31] A.C. Cangellaris and D.B. Wright. Analysis of the numerical error caused by the stair-stepped approximation of a conducting boundary in fdtd simulations of electromagnetic phenomena. *IEEE Transactions on Antennas and Propagation*, 39(10):1518–1525, 1991.

[32] THomas G. Jurgens, Korada Umashankar, and Thomas G. Moore. Finite-difference time-domain modeling of curved surfaces. *IEEE Transactions on Antennas and Propagation*, 40(4):357–365, 1992.

[33] T.G. Jurgens and A. Taflove. Three-dimensional contour fdtd modeling of scattering from single and multiple bodies. *IEEE Transactions on Antennas and Propagation*, 41(12):1703–1708, 1993.

[34] S. Dey and R. Mittra. A locally conformal finite-difference time-domain (fdtd) algorithm for modeling three-dimensional perfectly conducting objects. *IEEE Microwave and Guided Wave Letters*, 7(9):273–275, 1997.

[35] Supriyo Dey and Raj Mittra. A modified locally conformal finite-difference time-domain algorithm for modeling three-dimensional perfectly conducting objects. *Microwave and Optical Technology Letters*, 17(6):349–352, 1998.

[36] C.J. Railton and J.B. Schneider. An analytical and numerical analysis of several locally conformal fdtd schemes. *IEEE Transactions on Microwave Theory and Techniques*, 47(1):56–66, 1999.

[37] Stephen D. Gedney. *Introduction to the Finite-Difference Time-Domain (FDTD) Method for Electromagnetics*. Morgan & Claypool, 2011.

[38] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[39] Georges A Darbellay and Igor Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, 1999.

[40] Martina Mueller, Carol L Wagner, David J Annibale, Thomas C Hulsey, Rebecca G Knapp, and Jonas S Almeida. Predicting extubation outcome in preterm newborns: a comparison of neural networks with clinical expertise and statistical modeling. *Pediatric research*, 56(1):11–18, 2004.

[41] Paweł Piotr Cielecki, Mathias Hedegaard Kristensen, and Esben Skovsen. Analysis and classification of frequency-domain terahertz reflection spectra using supervised and unsupervised dimensionality reduction methods. *Journal of Infrared, Millimeter, and Terahertz Waves*, 42(9):1005–1026, 2021.

[42] Kevin P Murphy et al. Naive bayes classifiers. *University of British Columbia*, 18(60):1–8, 2006.

[43] S Madeh Piryonesi and Tamer E El-Diraby. Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. *Journal of Transportation Engineering, Part B: Pavements*, 146(2):04020022, 2020.

[44] Alaa Tharwat. Linear vs. quadratic discriminant analysis classifier: a tutorial. *International Journal of Applied Pattern Recognition*, 3(2):145–180, 2016.

[45] Mohammed J Islam, QM Jonathan Wu, Majid Ahmadi, and Maher A Sid-Ahmed. Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers. In *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, pages 1541–1546. IEEE, 2007.

[46] Luca Razzari, Andrea Toma, Matteo Clerici, Mostafa Shalaby, Gobind Das, Carlo Liberale, Manohar Chirumamilla, Remo Proietti Zaccaria, Francesco De Angelis, Marco Peccianti, Roberto Morandotti, and Enzo Di Fabrizio. Terahertz dipole nanoantenna arrays: resonance characteristics. *Plasmonics*, 8(1):133–138, 2013.

A | Database Information

Information about all the measurements taken in this project can be found in table A.1, the table contains information about the sample that was measured, and how many repetitive measurements of said sample were taken. The first letter in each name describes the 'shape' of the sample, with A being air, D being double dipoles, L being antennas (straight lines), S being split rings, and N being none, which is just a silica wafer. The next two characters describe the size of the nanostructure, where the size for air and silica wafers are Na for not applicable, the rest of the name contains information about special conditions, such as whether it is a reference, standard conditions, sugar deposited, a second measurement taken another day, and whether water vapour is present or not.

Name	**Measurements**	**Name**	**Measurements**
ANaAir	100	L99base	100
ANaAirFlipHolder	100	NNaSi	100
ANaAirFlipHolderW	100	NNaSiFlip	100
ANaAirW	100	NNaSiFlipHolder	100
ANaNoholder	100	NNaSiFlipHolderW	100
ANaNoholderW	100	NNaSiFlipW	100
D40base	101	NNaSiGlu100ng	100
L24refbase	100	NNaSiGlu100ng1-1	100
L24refbaseR	100	NNaSiGlu100ng1-2	100
L35Glu10ngW	100	NNaSiGlu100ng1-3	100
L39Glu10ug	100	NNaSiGlu100ng2	100
L39Glu10ug2	100	NNaSiGlu100ng2-1	100
L39Glu10ug2W	100	NNaSiGlu100ng2-2	100
L39Glu10ugW	100	NNaSiGlu100ng2-3	100
L39Glu300ug2	100	NNaSiGlu100ng3	100
L39Glu50ug	100	NNaSiGlu100ngW	100
L39Glu50ug2	100	NNaSiGlu100ug	100
L39Glu50ugW	100	NNaSiGlu100ug2	100
L39Glu50ugWref	100	NNaSiGlu100ug3	100
L39Glu50ugref	100	NNaSiGlu100ugW	100
L39acGlu100ng-1	100	NNaSiGlu20ng	100
L39acGlu100ng-2	100	NNaSiGlu20ng1-1	100
L39acGlu100ng-3	100	NNaSiGlu20ng1-2	100
L39acGlu100ng-4	100	NNaSiGlu20ng1-3	100
L39acGlu20ng-1	100	NNaSiGlu20ng1-4	100

Table A.1 continued from previous page

L39acGlu20ng-2	100	NNaSiGlu20ng2	100
L39acGlu20ng-3	100	NNaSiGlu20ng2-1	100
L39acGlu20ng-4	100	NNaSiGlu20ng2-2	100
L39acGlu60ng-1	100	NNaSiGlu20ng3	100
L39acGlu60ng-2	100	NNaSiGlu20ngW	100
L39acGlu60ng-3	100	NNaSiGlu20ug	100
L39acGlu60ng-4	100	NNaSiGlu20ug2	100
L39acbase-1	100	NNaSiGlu20ug3	100
L39acbase-2	100	NNaSiGlu20ugW	100
L39refbase	100	NNaSiGlu300ug	100
L39refbaseR	100	NNaSiGlu300ug2	100
L40Glu1000ug	100	NNaSiGlu300ug3	100
L40Glu100ug	100	NNaSiGlu300ugW	100
L40Glu125Suc375	100	NNaSiGlu500ug	100
L40Glu125Suc375W	100	NNaSiGlu500ug2	100
L40Glu2000ug	100	NNaSiGlu500ug3	100
L40Glu200ug	100	NNaSiGlu500ugW	100
L40Glu250Suc250	100	NNaSiGlu60ng	100
L40Glu250Suc250W	100	NNaSiGlu60ng1-1	100
L40Glu300ug	100	NNaSiGlu60ng1-2	100
L40Glu375Suc125	100	NNaSiGlu60ng2	100
L40Glu375Suc125W	100	NNaSiGlu60ng2-1	100
L40Glu400ug	100	NNaSiGlu60ng2-2	100
L40Glu500ug	100	NNaSiGlu60ng3	100
L40Glu500ug2	100	NNaSiGlu60ngW	100
L40Glu500ug2W	100	NNaSiGlu60ug	100
L40Glu500ugMove	100	NNaSiGlu60ug2	100
L40Glu500ugNew	100	NNaSiGlu60ug3	100
L40Glu500ugRot	100	NNaSiGlu60ugW	100
L40Glu50ug	100	NNaSiMove	100
L40Suc1000ug	100	NNaSiMoveW	100
L40Suc1000ugW	100	NNaSiR135	100
L40Suc250Glu250	100	NNaSiR135W	100
L40Suc250Glu250W	100	NNaSiR180	100
L40Suc500ug	100	NNaSiR180W	100
L40Suc500ugW	100	NNaSiR90	100
L40base	100	NNaSiR90W	100
L40base2	100	NNaSiSuc1000ug	100
L40base3	100	NNaSiW	100
L42base	101	NNaSidriedmiliq	100
L42base2	100	NNaSidriedmiliq1-1	100
L42base2W	100	NNaSidriedmiliq1-2	100
L45Glu10ng	100	NNaSidriedmiliq2	100
L45base	101	NNaSidriedmiliq2-1	100
L66base	101	NNaSifresh	100
L86Glu1000ug	100	NNaSifresh1-1	100

Table A.1 continued from previous page

L86Glu100ug	100	NNaSifresh1-2	100
L86Glu200ug	100	NNaSifresh2	100
L86Glu500ug	100	NNaSifresh2-1	100
L86base	101	NNaSifresh2-2	100
L86base2	100	NNaSifreshW	100
L86fast	500	NNaSifreshW2	100
L87base	100	S36base	100
L87fast	500	S36fast	667
L99Glu100ug	100	S68base	100
L99Glu300ug	100	S68fast	500

Table A.1: A table of all the data