NTNU

Norwegian University of
Science and Technology

Polarization beam-splitting metasurfaces analyzed using the finite-difference time-domain software EMTL

Einar Nyberg Karlsen

Master of Science in Physics and Mathematics
Submission date: August 2018
Supervisor: Morten Kildemo, IFY

Norwegian University of Science and Technology
Department of Physics

Abstract

Using the Finite-Difference Time-Domain electromagnetic solver package EMTL, this thesis numerically reproduces the results of the polarization splitting metasurface developed by Pors et al. [1] and introduces new variations, both in analysis and design. EMTL proved to be sufficiently accurate to recreate the characteristics of the system.

An alternative analysis method for designing the metasurface was formulated and verified numerically. This analysis showed that contiguousness and sub-wavelength separation were not required for the metasurface pattern to function.

A flaw in the original design was the interaction between neighboring components in the metasurface. These interactions happened with the same periodicity as the pattern in the original design, causing them to create interference with the primary reflections. By interleaving two metasurfaces in a checkered pattern, the unwanted interactions were spread out to the common period of two patterns. Due to the unwanted interactions being spread out, asymmetry in the reflection intensity was reduced between 20% and 80%. This alternative design has the potential of miniaturizing the metasurface design further, and the symmetry improvement can reduce the required post-processing needed to analyse the output quickly.

Sammendrag

Ved å bruke programvaren EMTL, som bruker diskret differanse tidsdomenemetoden, gjenskaper denne avhandlingen resultatene for en polarisasjonssplittende metaoverflate som var utviklet av Pors et al. [1] og introduserer nye variasjoner, både i analyse og design. EMTL var i stand til å gjenskape karakteristikkene til systemet til en tilstrekkelig grad.

En alternativ metode for å analysere metaoverflaten ble lagd, og det ble bekreftet at det fungerte numerisk. Denne analysen viste at et kontinuerlig mønster og sub-bølgelengde separasjon ikke var nødvendig for at metaoverflaten skulle fungere.

En svakhet i det originale designet var forekomsten av interaksjoner mellom nabokomponenter. Disse interaksjonene forekom med samme periodisitet som mønsteret i originaldesignet, som førte til at de skapte interferens med hovedrefleksjonene. Ved å sammenveve to metaoverflate-mønstre i et kriss-kross mønster ble de uønskede interaksjonene spredd utover til den felles perioden til de to mønstrene. Takket være denne spredningen ble asymmetrien i refleksjonsintensiteten redusert med mellom 20% og 80%. Dette alternative designet har potensialet til å forminske metaoverflaten ytterligere, samt at forbedringen i symmetri kan redusere mengden post-prossesering som kreves for å analysere refleksjonene.

Preface

This document was submitted as a master thesis in technical physics, which is the physics specialization of the integrated master's programme in Applied Physics and Mathematics. The thesis amounts to 30 ECTS-credits.

I would like to thank my supervisor Morten Kildemo for guidance and patience. His ability to be able to point me to sources regarding any topic I asked about was invaluable. I wish to thank Per Magnus Walmsnes, a PhD student under Kildemo, for helping with some of the programming when I started using the code. A large thanks is also owed to Ilya Valuev and Sergey Belousov at Kintech labs, for providing EMTL for free, and allowing us to access the source code in order to make adjustments for our own use, as well as being very helpful in providing assistance whenever I had questions regarding EMTL. Finally, I wish to thank Jabir Ali Ouassou, for providing a lot of help with programming, as well as discussions and typesetting in LaTeX.

This thesis was typeset in LaTeX, with a configuration made by Jabir Ali Ouassou which he let me copy.

Contents

1 Introduction 1

2 Theory 3

2.1 Fundamentals of electromagnetism 3

2.2 Dielectric function models 7

2.3 Plasmons 13

2.4 Polarization optics 16

2.5 Coherence width 22

3 Numerics of finite-difference time-domain 23

3.1 Fast Fourier transform 23

3.2 Finite difference method 26

3.3 Yee cells 27

3.4 Dispersive media 32

3.5 Boundary conditions 36

3.6 Signal waves 40

3.7 Detectors 44

3.8 Numerical accuracy 46

3.9 Runtime and memory scaling 50

3.10 EMTL 52

4 Blazed gold nanogratings 55

4.1 Mechanism 55

4.2 The Pors metagrating 63

4.3 Alternative method of analyzing mechanics 66

4.4 Secondary plasmonic behaviour 72

5 Simulation results 81

5.1 Single gold box 82

5.2 Single metasurface rows 90

5.3 Full Pors metasurface 100

5.4 Sparse metasurface 103

5.5 Interleaved lines 105

6 Final remarks and outlook 117

Bibliography 119

Sample code 127

1 Introduction

At the turn of the millennium a new field of optical material science came to the forefront, namely metamaterials. With Ebbesen et al. demonstrating optical transmission through a metal film with sub-wavelength holes [2], and Pendry hypothesising perfect lensing being achievable by the use of materials with a negative refractive index [3], the search for both production and application of such materials started in full. The concept of optical metamaterials was born, structures where microscopic structural effects produce macroscopic behaviour that is outside of the normal behaviour for regular materials [4–6]. Both negative permeability [7–9] and negative refractive index [10–12] are achievable for certain wavelengths by having electrically and/or magnetically responsive structures at a sub-wavelength scale.

There are many emerging areas of metamaterial research. Hyperlenses that are capable of magnifying beyond the limits of conventional lenses [13–15], color generation using plasmonic nanostructures [16], cloaking technology[17], and improving fiber-optics [18].

The biggest challenge for true metamaterials is the limitations in producing structures at scales of micrometers or below in three dimensions[19]. Therefore, as it is the optical properties at the macroscopic scale that are important, the problem has been circumvented by focusing on the surfaces of the materials. By having advanced structures at the interfaces of a material, one can achieve an effective metamaterial property to a satisfactory degree[20]. These surface structures, normally using a mix of plasmonic and non-plasmonic materials, are called metasurfaces. The advantage of a metasurface is that it is far more feasible to produce, with electron beam lithography becoming ever more accessible and accurate. Therefore, the field shifted its focus to metasurfaces as the more viable pathway to producing metamaterial behaviour.

Though it is possible to analytically understand the simpler plasmonic phenomena, e.g. surface plasmons matching the incoming wave perfectly, which is the mechanism behind mirrors, to predict the behaviour of a plasmonically active element of finite size is not something that is done analytically. Therefore, the field of metasurfaces is, as most fields of physics are, dependent on numerical simulations to progress. For the field of metasurfaces, the relevant simulation methods are the finite difference time domain (FDTD), and finite element method (FEM). Both are good methods with their strengths and weaknesses, and as such, should be used according to their strengths. With personal computers having several gigabytes of RAM at their disposal, and supercomputers having up to terabytes, the potential problem size of a numerical analysis is enormous. However, limitations still exist. FEM solvers are especially vulnerable to the problem size becoming untenable in RAM, while FDTD struggles with algorithmic limitations.

In this thesis, the focus will be the FDTD method and its application to simulating a specific metasurface specified in Ref. [1]. This system is interesting as it has directional refraction dependent on the polarization of the incoming light, which may open the path to multiplexing in optical information cables among other things. It is also a system that cannot be simulated in its entirety in FEM, and is therefore a good candidate for FDTD simulation. Furthermore, the methodology and design processes underlying the metasurface will be investigated in the hope of improving the understanding and design of the system in question.

Theory

2

In this chapter, the fundamental theory of electromagnetism (EM), optics and material interactions with EM fields will be reviewed. This thesis is focused on the specific use of the Finite Difference Time Domain (FDTD) method to simulate electromagnetism, so most of the theoretical consideration is geared towards being applicable in an FDTD scheme.

2.1 Fundamentals of electromagnetism

Maxwell's equations

Maxwell's equations are the fundamental equations in understanding the nature of electromagnetic waves. Since this thesis will be focused on a time stepping algorithm, the differential version of the equations are most relevant, which can be written as follows [21].

$$\nabla \cdot \boldsymbol{D} = \rho_b \tag{2.1}$$

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

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

$$\nabla \times \boldsymbol{H} = \boldsymbol{J}_b + \frac{\partial \boldsymbol{D}}{\partial t} \tag{2.4}$$

Here $\boldsymbol{D}$ is the dielectric field, ρ_b is the bound electric charge density, $\boldsymbol{B}$ is the magnetic field, $\boldsymbol{E}$ is the electric field, $\boldsymbol{H}$ is the magnetizing field, and $\boldsymbol{J}_b$ is the the bound current.

These fields have the following constitutive relationships in frequency space [22]:

$$\boldsymbol{D}(\omega) = \varepsilon_0 \varepsilon(\omega)\boldsymbol{E}(\omega) \qquad \boldsymbol{P}(\omega) = \varepsilon_0 \chi_e(\omega)\boldsymbol{E}(\omega) \tag{2.5}$$

$$\boldsymbol{B}(\omega) = \mu_0 \mu(\omega)\boldsymbol{H}(\omega) \qquad \boldsymbol{M}(\omega) = \mu_0 \chi_m(\omega)\boldsymbol{H}(\omega) \tag{2.6}$$

$$\boldsymbol{J}(\omega) = \sigma(\omega)\boldsymbol{E}(\omega) \tag{2.7}$$

Here ω is the frequency, $\boldsymbol{P}$ is polarization, $\boldsymbol{M}$ is magnetization, χ_e and χ_m are respectively the electric and magnetic susceptibilities, and σ is the conductivity.

These equations are most easily expressed in the frequency domain because of the nature of the susceptibility functions. If one wishes to view them in the time domain, an inverse Fourier transform needs to be used:

$$\boldsymbol{D}(t) = \int_{\infty}^{\infty} \varepsilon(\omega)E(\omega)e^{2\pi it\omega}d\omega = \int \varepsilon(t - t')E(t')dt' \tag{2.8}$$

If the susceptibility is also anisotropic, the same has to be done for the k-space to real space transformation, resulting in

$$\boldsymbol{D}(\boldsymbol{r}, t) = \int\int \varepsilon(\boldsymbol{r} - \boldsymbol{r}', t - t')E(\boldsymbol{r}', t')dr'dt' = \varepsilon * E(\boldsymbol{r}, t)$$

Thus, if one sees any of the constitutional relationships shown earlier in the time domain, the susceptibility functions either have to be constants, or they should be written as convolutions. This is a common error that can even be found in some textbooks. Because this thesis will primarily focus on numerics, it will be shown later in section 2.2.3 that if the susceptibilities are polynomial functions with regards to ω, one can instead use the relation

$$\mathcal{F}\left(\frac{\partial}{\partial t}g(t)\right) = i\omega G(\omega) \tag{2.9}$$

to avoid the convolution and instead end up with a differential equation, which are more manageable in a numerical context.

In this thesis, no magnetically susceptible materials will be used, i.e. $\chi_m = 0$, so for the rest of the thesis $\boldsymbol{M} = 0$, and χ_e will only be called χ. Furthermore, assuming the media to be linear and isotropic, one has that $\varepsilon = (1 + \chi_e)$ and $\mu = (1 + \chi_m) = 1$, resulting in the following constitutive relationships:

$$\boldsymbol{D}(\omega) = \varepsilon_0\boldsymbol{E}(\omega) + \varepsilon_0\chi(\omega)\boldsymbol{E}(\omega) = \varepsilon_0\boldsymbol{E}(\omega) + \boldsymbol{P}(\omega) \tag{2.10}$$

$$\boldsymbol{B} = \mu_0\boldsymbol{H} \tag{2.11}$$

One can with this see that there is a straightforward relationship between equations (2.10) and (2.11) and equations (2.3) and (2.4), which allows one to find the time derivatives of the $\boldsymbol{D}$ and $\boldsymbol{B}$ fields, and related them back to $\boldsymbol{E}$ and $\boldsymbol{H}$ respectively, forming a closed computational loop suitable of time stepping.

The final constitutive relationship that is important is that the current, $\boldsymbol{J}$, can be expressed as either

$$\boldsymbol{J}(\boldsymbol{r}, t) = \frac{\partial \boldsymbol{P}(\boldsymbol{r}, t)}{\partial t}, \qquad \text{or} \qquad \boldsymbol{J}(\boldsymbol{k}, \omega) = i\omega \boldsymbol{P}(\boldsymbol{k}, \omega), \tag{2.12}$$

depending on whether one is in real or Fourier space. Further, using equations (2.7), (2.10) and (2.12), one gets the following relationship:

$$\varepsilon(\boldsymbol{k}, \omega) = 1 + \frac{i\sigma(\boldsymbol{k}, \omega)}{\varepsilon_0 \omega} \tag{2.13}$$

2.1.1 Wave equation

Having established the base equations from Maxwell, the next step is to find equations that describe electromagnetic wave propagation.

Recalling the constitutive relationships of $\boldsymbol{D}$ and $\boldsymbol{B}$ from equations (2.5) and (2.6), we have the following:

$$\boldsymbol{D} = \varepsilon_0 \boldsymbol{E} + \boldsymbol{P} \tag{2.14}$$

$$\boldsymbol{B} = \mu_0 \boldsymbol{H} + \boldsymbol{M} \tag{2.15}$$

From this, one can rewrite the Maxwell curl equations:

$$\nabla \times \boldsymbol{E} = -\frac{\partial}{\partial t}(\mu_0 \boldsymbol{H} + \boldsymbol{M}) \tag{2.16}$$

$$\nabla \times \boldsymbol{H} = \boldsymbol{J}_b + \frac{\partial}{\partial t}(\varepsilon_0 \boldsymbol{E} + \boldsymbol{P}) \tag{2.17}$$

Then, take the curl of both equations, and use that partial derivate operators commute to substitute $\nabla \times \boldsymbol{E}$ and $\nabla \times \boldsymbol{H}$.

$$\nabla \times \nabla \times \boldsymbol{E} = -\frac{\partial}{\partial t}\left(\mu_0 \left[\boldsymbol{J}_b + \frac{\partial}{\partial t}(\varepsilon_0 \boldsymbol{E} + \boldsymbol{P})\right] + \nabla \times \boldsymbol{M}\right) \tag{2.18}$$

$$\nabla \times \nabla \times \boldsymbol{H} = \nabla \times \boldsymbol{J}_b + \frac{\partial}{\partial t}\left(\varepsilon_0 \left[-\frac{\partial}{\partial t}(\mu_0 \boldsymbol{H} + \boldsymbol{M})\right] + \nabla \times \boldsymbol{P}\right) \tag{2.19}$$

Rearranging the terms a bit, and using $c = 1/\sqrt{\varepsilon_0 \mu_0}$ one gets what is commonly referred to as the inhomogeneous wave equations:

$$\nabla \times \nabla \times \boldsymbol{E} + \frac{1}{c^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = -\mu_0 \frac{\partial}{\partial t}\left(\boldsymbol{J}_b + \frac{\partial \boldsymbol{P}}{\partial t} - \nabla \times \boldsymbol{M}\right) \tag{2.20}$$

$$\nabla \times \nabla \times \boldsymbol{H} + \frac{1}{c^2}\frac{\partial^2 \boldsymbol{H}}{\partial t^2} = \nabla \times \boldsymbol{J}_b + \nabla \times \frac{\partial \boldsymbol{P}}{\partial t} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{M}}{\partial t^2} \tag{2.21}$$

The double curl of a vector can be rewritten as follows:

$$\nabla \times \nabla \times \boldsymbol{V} = \nabla(\nabla \cdot \boldsymbol{V}) - \nabla^2 \boldsymbol{V}$$

In vacuum the electromagnetic waves have no matter to interact with, which reduces the constitutive relationships and Maxwell's divergence equations as follows:

$$\begin{array}{lclcl}
\boldsymbol{P} = 0 & \Longrightarrow & \boldsymbol{D} = \varepsilon_0 \boldsymbol{E} & \Longrightarrow & \nabla \cdot \boldsymbol{E} = 0 \\
\boldsymbol{M} = 0 & \Longrightarrow & \boldsymbol{B} = \mu_0 \boldsymbol{H} & \Longrightarrow & \nabla \cdot \boldsymbol{H} = 0 \\
\boldsymbol{J} = 0 & & & &
\end{array}$$

This results in the homogeneous electromagnetic wave equations

$$\nabla^2 \boldsymbol{E} = \frac{1}{c^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} \qquad \nabla^2 \boldsymbol{H} = \frac{1}{c^2}\frac{\partial^2 \boldsymbol{H}}{\partial t^2} \tag{2.22}$$

This also holds true for light propagating through isotropic non-dispersive media, where the adjusted dielectric constant only lowers the light speed. For short distances, air can be approximated to be such a medium, or more commonly, just approximated to be a vacuum.

For an EM wave travelling in the $\hat{z}$-direction, the general solution can be written as follows:

$$E_x = A_x e^{i(kz - \omega t + \varphi_x)} \qquad H_y = A_x e^{i(kz - \omega t + \varphi_x + \pi/2)} \tag{2.23}$$

$$E_y = A_y e^{i(kz - \omega t + \varphi_y)} \qquad H_x = A_y e^{i(kz - \omega t + \varphi_y + \pi/2)} \tag{2.24}$$

with $\omega = c^2k$. The $\boldsymbol{E}$ and $\boldsymbol{H}$ fields have to be orthogonal as the Maxwell curl equations would stop all non-orthogonal fields from propagating. Therefore, it is common to only discuss the $\boldsymbol{E}$-fields when discussing electromagnetic waves, as the $\boldsymbol{H}$-field follows automatically.

2.2 Dielectric function models

Now that the general behaviour of EM fields has been explained, next is to look at their interactions with matter.

2.2.1 Drude model

In metals, and particularly the noble ones, the electron structure is such that it can be approximated by a classic "Free Electron Model", also known as the Drude model, wherein one assumes free movement of electrons with damping through random collisions. Within its region of validity, the Drude model accurately describes the reaction of free electrons to electromagnetic stimuli. The criteria for validity are as follows [23]:

- The electrons have to be sufficiently free, i.e. have a linear response to interacting fields.
- The energy of the interacting field is not able to cause interband transitions.

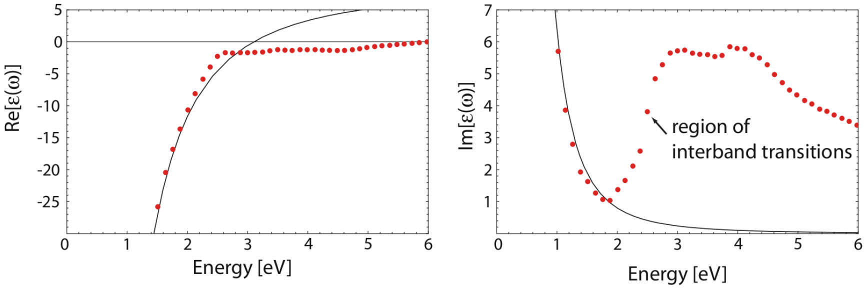


Figure 2.1: Drude model fit to gold (line) based on the experimental data (dots) based on Ref. [24]. As can be seen, at roughly 1.9 eV the Drude model stops working due to interband transitions. Figure was taken from Ref. [23].

If these are satisfied, one can approximate the free electrons to form an electron gas, in which one assumes an electron to statistically behave according to the basic momentum equation with friction. In 1D the equation reads as follows:

$$m\ddot{x} + m\gamma\dot{x} = -eE_x(t) \tag{2.25}$$

The friction, γ, is in this case the statistical average of random collisions that dampen the momentum. Assuming the $\boldsymbol{E}$-field to be a wave, one expects a solution of the form of an harmonic oscillator. However, due to a harmonic oscillator being phase-dependent, one has to first introduce $x(t)$ as the inverse Fourier transform of a frequency-dependent function $x(\omega)$, and the same for $E(t)$.

$$x(t) = \int e^{-i\omega t} x(\omega) d\omega \tag{2.26}$$

$$E_x(t) = \int e^{-i\omega t} E_x(\omega) d\omega \tag{2.27}$$

Thus, equation (2.25) becomes

$$\int e^{-i\omega t} \left[m\omega^2 x(\omega) + im\gamma\omega x(\omega) \right] d\omega = \int e^{-i\omega t} \left[E_x(\omega) \right] d\omega \tag{2.28}$$

Then, due to the same integral being present on both sides, and its validity has to work for any arbitrary set of frequencies, including any singular frequency, the terms inside the brackets have to be equal, eliminating the integrals and resulting in the following:

$$x(\omega) = \frac{e}{m(\omega^2 + i\gamma\omega)} E_x(\omega) \tag{2.29}$$

As the medium is isotropic, x can be replaced with a general direction $\boldsymbol{r}$ without any problems.

$$\boldsymbol{r}(\omega) = \frac{e}{m(\omega^2 + i\gamma\omega)} \boldsymbol{E}(\omega) \tag{2.30}$$

As this is a harmonic oscillation solution, the values for x account for displaced electrons, and can therefore be related to macroscopic polarization with $\boldsymbol{P} = -n e \boldsymbol{r}$. Inserting this into equation (2.5), one gets:

$$\boldsymbol{D}(\omega) = \varepsilon_0 \left(1 - \frac{ne^2}{m\varepsilon_0(\omega^2 + i\gamma\omega)}\right) \boldsymbol{E}(\omega) = \varepsilon_0 \left(1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega}\right) \boldsymbol{E}(\omega) \quad (2.31)$$

Here $\omega_p^2 = ne^2/m\varepsilon_0$, with ω_p being the plasma frequency of the free electron gas.

Some forms of the equation include a term $\Delta\varepsilon$ in the susceptibility, but this is only necessary if one uses the bare electron mass, $m = m_e$, instead of effective mass, $m = m_{eff}$. For simplicity, I chose effective mass.

The reason this is called the Drude model is because one can find the same relation by rewriting equation (2.25) to use the momentum instead, and relate that to the Drude conductivity formulas. One then uses equation (2.13) to get the same results as equation (2.31).

2.2.2 Generalized dielectric model

The Drude model properly explains the movement of the free electrons present in an ideal conductor. However, if the energy of the incoming $\boldsymbol{E}$-field is sufficiently high, bound electrons can get excited. When this happens, the Drude model becomes insufficient, and a new term needs to be added to the dielectric function.

As such, several other dielectric functions have been made. A good overview of the different models is found in Ref. [25], as it lists the models commonly used to model the dielectric properties of metals.

$$\chi(\omega) = \chi_0 + \Sigma_{i=1}^{i_{max}} \chi_{X,i}(\omega) \quad (2.32)$$

Here X is any of the models listed according to Ref. [25].

$$\text{Drude,} \quad \chi_D = \frac{-\omega_D^2}{\omega^2 - i\omega\Gamma_D} \tag{2.33}$$

$$\text{Lorentz,} \quad \chi_{L,i} = f_{L,i}\frac{\omega_{L,i}^2}{\omega_{L,i}^2 - i\omega\Gamma_{L,i} - \omega^2} \tag{2.34}$$

$$\text{Sellmeier,} \quad \chi_{S,i} = f_{S,i}\frac{\omega_{S,i}^2}{\omega_{S,i}^2 - \omega^2} \tag{2.35}$$

$$\text{Critical Points,} \quad \chi_{C,i} = 2f_{C,i}\omega_{C,i}\frac{\omega_{C,i}cos\varphi_{C,i} + (i\omega - \Gamma_{C,i})sin\varphi_{C,i}}{\Gamma_{C,i}^2 + \omega_{C,i}^2 - 2i\omega\Gamma_{C,i} - \omega^2} \tag{2.36}$$

The Lorentz equation, equation (2.34), is found in the same manner as the Drude equation, just by using a damped harmonic oscillator instead of free particles. Similarly, the Sellmeier equation equation (2.35) can be found using an undamped harmonic oscillator. The critical point model in equation (2.36) was found as a recurring result when fitting empirical data to a general analytical function, which according to Ref. [26] has the form:

$$\chi(\omega) = C + Ae^{i\varphi}(\omega - E + i\Gamma)^n \tag{2.37}$$

As can be seen, all of the functions listed by Prokopeva have similar terms. Therefore, numerical programs use a unified dispersion model.[1]

$$\chi_i(\omega) = \frac{a_{0,i} - i\omega a_{1,i}}{b_{0,i} - i\omega b_{1,i} - \omega^2} \tag{2.38}$$

By organizing all the different properties in one generalized form, it is highly suitable for numerical functions, as it can treat all different models equally.

[1] Some papers have included a fitting variable in front of ω^2 as well, allowing for ω^2 to be set to zero in the model. This is not done in EMTL, as it is numerically inefficient to do so, as will be shown in section 3.4.1. In other words, the modified Lorentz model will only be used to model susceptibilities where the ω^2 term is relevant.

Because equation (2.38) is superficially similar to equation (2.34), it is commonly referred to as the Modified Lorentz Model. This is, in this author's opinion, a misnomer, as a far more suitable name would be the Generalized Dielectric Model. However, due to the prevalence of the name "Modified Lorentz Model", I will use that name in this thesis. Worth noting in this context is the Critical Point model, as it is originally a fitting model to begin with, and it stands to reason that future model fits might try to use the Modified Lorentz model directly, as done in Ref. [27], or use critical points as the first guess in a Modified Lorentz numerical fit.

Because I've already listed all the common conductive models, I'll also list the pure dipole models. These functions are primarily useful for microwaves or longer, so they are not relevant for this thesis, but can be useful if this thesis is given to students as an intro to EM-simulations.

$$\text{\textit{Debye},} \qquad \chi_i = \frac{\Delta\chi}{1 + i\omega\tau} \tag{2.39}$$

$$\text{\textit{Cole-Cole},} \qquad \chi_i = \frac{\Delta\chi}{1 + (i\omega\tau)^{\alpha}} \tag{2.40}$$

$$\text{\textit{Havriliak-Negami},} \qquad \chi_i = \frac{\Delta\chi}{(1 + (i\omega\tau)^{\alpha})^{\beta}} \tag{2.41}$$

Ref. [28] showed that Cole-Cole is computationally inefficient, and it stands to reason that Havriliak–Negami should be as well. Moreover, it states that Debye is a sufficiently accurate numerical model when considering the increased calculation speed compared to the others. Essentially, because Debye is simple, one can use far greater numerical accuracy when calculating, resulting in comparably better results.

For reference, the FDTD software used in this thesis, the ElectroMagnetic Template Library (EMTL), has implemented their dielectricity function container class as follows:

$$\chi(\omega) = \chi_0 + \sum_{i=1}^{i_{max}} \chi_{Drude,i} + \sum_{j=1}^{j_{max}} \chi_{Debye,j} + \sum_{k=1}^{k_{max}} \chi_{MLM,k} \tag{2.42}$$

Here, MLM means "modified Lorentz model".

2.2.3 Polarization currents

Having established a set of dielectric susceptibility functions, it is now necessary to be able to express them in the time domain in a form suitable for a time stepping algorithm. The primary goal is to be able to use the Maxwell curl equations, equations (2.3) and (2.4), as they relate a time differential to the present state of the fields. As such, one needs to look at equation (2.4) as it requires the dielectric function. First off, one can combine equations (2.4), (2.10) and (2.12) to get a new form of equation (2.4).

$$\nabla \times \boldsymbol{H} = \boldsymbol{J}_b + \frac{\partial \boldsymbol{E}}{\partial t} + \boldsymbol{J}_p(t) \tag{2.43}$$

Here $\boldsymbol{J}_p$ is the term for the polarization currents induced by EM waves. The next necessary step is to split the current into parts dependent on each term of the total dielectric susceptibility. In other words, treat the dielectric susceptibilities as independent of each other, and look at the currents each generate on their own. These currents are referred to as polarization currents, as they relate to the time differential of the polarization state of the material. Remembering from equation (2.8), we have that the susceptibility functions have to be treated in the frequency space initially.

$$\boldsymbol{J}_p(\omega) = \sum_X \boldsymbol{J}_X(\omega) = \sum_X i\omega \boldsymbol{P}_X(\omega) = \sum_X i\omega\chi_X(\omega)\boldsymbol{E}(\omega) \tag{2.44}$$

Isolating each part, and inserting the general dielectric model yields:

$$\boldsymbol{J}_X(\omega) = \frac{a_{0,X} - i\omega a_{1,X}}{b_{0,X} - i\omega b_{1,X} - \omega^2} i\omega \boldsymbol{E}(\omega) \tag{2.45}$$

$$(b_{0,X} - i\omega b_{1,X} - \omega^2)\boldsymbol{J}_X(\omega) = (i\omega a_{0,X} + \omega^2 a_{1,X})\boldsymbol{E}(\omega) \tag{2.46}$$

Doing an inverse Fourier transform on equation (2.46) yields the following differential equation:

$$b_{0,X}\boldsymbol{J}_X(t) + b_{1,X}\frac{d\boldsymbol{J}_X(t)}{dt} + \frac{d^2\boldsymbol{J}_X(t)}{dt^2} = a_{0,X}\frac{d\boldsymbol{E}(t)}{dt} + a_{1,X}\frac{d^2\boldsymbol{E}(t)}{dt^2} \tag{2.47}$$

and thus a differential equation relating the polarization currents to the electric field has been given as the solution to the problem. This does assume that there are no effects from one dielectricity current on the other, which is true so long as the availability of free electrons is sufficient. If that is no longer the case, one would have to consult quantum mechanics to sort out the effects, which would be far beyond the scope of any FDTD algorithm discussed in this thesis.

2.3 Plasmons

A plasmon is a localized plasma oscillation within a material. Any material with the property of dislocated or free electrons, such as in conductive materials, has the potential for plasmons. Plasmons arise when an electric field starts affecting the free electrons present in the material and disturb the equilibrium. Of course, an electric current will not qualify as this is a steady movement of electrons, and not an oscillation.

Plasmons are generally divided into two categories: surface plasmons, where the plasmonic properties are confined to a surface, and volume plasmons, where the plasmonic phenomena spans a volume [23]. Furthermore, it is also common to discuss localized plasmons, i.e. plasmons where the size of the plasmonic media restricts the behaviour.

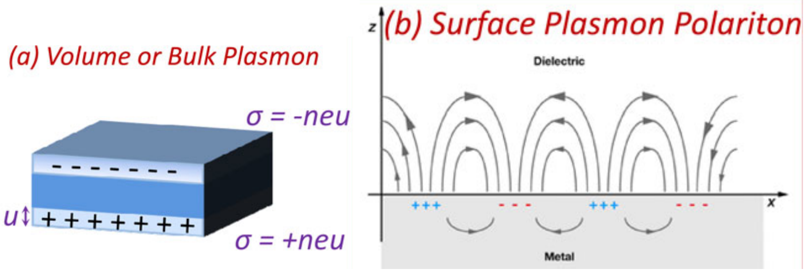


Figure 2.2: Illustration of Volume and Surface plasmons. These images was taken from Ref. [29].

It is worth noting that plasmons will often only have visible charge accumulation at surfaces, such as in figure 2.2, as the free electrons in the material will move to nullify any field, resulting in no perceptible charge within the material. This is of course dependent on the free electrons being free to travel in any direction in the material, i.e. not restricted as a consequence of lattice structure. This said, such restrictions would invalidate the basic assumptions made when handling the polarization currents as well, which would prevent any standard FDTD EM algorithm from being able to simulate the system. Therefore, all materials will be assumed to have no intrinsic directionality in their lattice structure.

A plasmon will have zero net electrical charge as the electrons are only displaced, but will create localized dipoles. These localized dipoles will be able to interact with either electromagnetic waves along the surface, or other dipoles in close proximity. This coupling is called a polariton, and it is common to distinguish between the polaritons purely interacting with EM waves, which are called surface plasmon polaritons,[2] and polaritons between different plasmon excitations, called localized surface plasmon resonance (LSPR). LSPR happens when there are non-conductive gaps between plasmonic media, causing any localized surface plasmon to interact through EM fields to propagate beyond the particle which carries it.

Then there is the gap surface plasmon (GSP), which is a special case of the LSPR,[3] where two surfaces with a small gap between them can generate a coupled plasmon interaction. Analogous to how a magnetic field will induce a current in a closed conducting loop, a gap surface plasmon can react to EM waves and interfere in a similar pattern, as illustrated by figure 2.3. This allows a GSP to effectively absorb, retain and re-emit the EM waves incoming if the EM waves resonate with the GSP structure. A vital difference between a regular surface plasmon and a GSP is the retention time of the GSP can be far greater.

[2] This effect only appears at interfaces between media of different susceptibilities which gives rise to one media having stronger plasmonic behaviour than the other, causing the other to have EM waves which the plasmons couple with on the interface.

[3] I am somewhat unsure about this categorization, as surface plasmons normally will travel along a surface. A GSP however, can stay in one place due to the interaction between the two surfaces. Therefore, if the propagation of the plasmon is integral to it being categorized as a surface plasmon, then a GSP will not be an LSPR, but instead be its own category entirely.

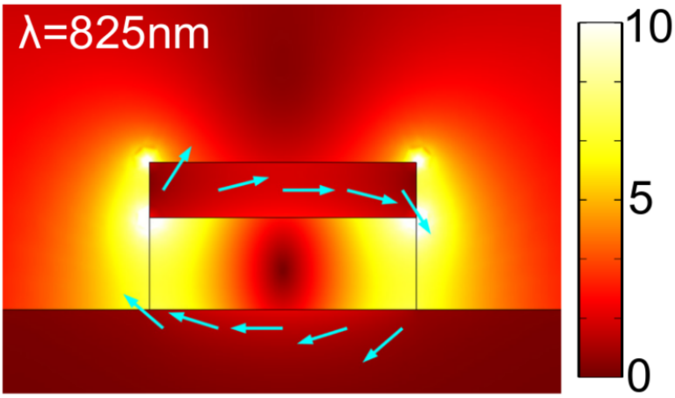


Figure 2.3: Image of simulation data displaying a gap surface plasmon produced from TM-polarized light. The scale is electric field intensity, and the arrows display the direction of the polarization current at a suitable point in time. This image was taken from Ref. [30].

In order to simulate plasmons, one therefore needs accurate models of EM interactions with materials.

2.3.1 Properties of gold

In the system considered in this thesis, gold will be the only medium capable of generating plasmons. As such, a cursory introduction to its properties should be made.

- Gold has a very weak magnetic response, to the degree that it can be considered non-magnetic [31].
- Gold is highly conductive, i.e. it has electrons in energy bands that allow for much freer movement than most other normally conducting materials [21].
- Gold starts having interband transitions between electrons at around 1.9 eV [23], which corresponds to a 650 nm wavelength photon's energy.

These properties make gold a suitable material for plasmonic systems of wavelengths at the high end of the visible light spectrum and above.

2.4 Polarization optics

Knowing that for EM waves the field vectors $\boldsymbol{E}$ and $\boldsymbol{B}$ are always perpendicular to the wave vector $\boldsymbol{k}$, it is useful to simplify the mathematical description of the field polarization. The two most common formalisms are Stokes and Jones formalism, and each have their niche.

Recalling equations (2.23) and (2.24)

$$E_x = A_x e^{i(kz-\omega t+\varphi_x)} \qquad E_y = A_y e^{i(kz-\omega t+\varphi_y)}$$

It is common to consider two distinct cases of the EM wave solution. The first when $A_x = A_y$ and $\varphi_y = \varphi_x \pm \pi/2$, which produces a circular polarization state. The second is when $\varphi_y = \varphi_x + \pi n$, with n being 0 or one of the amplitudes being zero, which produces a linear polarization state.

Any polarization state can be expressed as a linear combination of linear and circular polarization.

2.4.1 Jones vectors and matrices

Jones vectors are split into the two orthogonal axes to the wave's propagation direction. The common choice is that the wave travels along the z axis, while the x and y axis are the polarization axes [32].

$$\boldsymbol{j} = \begin{bmatrix} E_{0x}e^{i\varphi_x} \\ E_{0y}e^{i\varphi_y} \end{bmatrix} \tag{2.48}$$

The Jones vector correlates to the EM field functions as follows:

$$\begin{bmatrix} E_x(t) \\ E_y(t) \end{bmatrix} = \begin{bmatrix} E_{0x}e^{i\varphi_x} \\ E_{0y}e^{i\varphi_y} \end{bmatrix} e^{i(kz-\omega t)} \tag{2.49}$$

with z being the propagation direction.

The core polarization states that are normally discussed in optics are 0°, 90°, 45°, and 135° degree linearly polarized light (with regards to the x-axis), and left hand and right hand circularly polarized light. These are called $|x\rangle$, $|y\rangle$, $|a\rangle$, $|b\rangle$, $|l\rangle$, and $|r\rangle$ respectively, and have the following normalized Jones vectors:

$$\boldsymbol{j}_{|x\rangle} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{j}_{|a\rangle} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \boldsymbol{j}_{|l\rangle} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$$

$$\boldsymbol{j}_{|y\rangle} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \boldsymbol{j}_{|b\rangle} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \boldsymbol{j}_{|r\rangle} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$$

In order to more easily see what polarization state a Jones vector is representing, it is common to set it so that one of the terms is always purely real [33]. In this thesis I have chosen the x-component to be real:

$$\boldsymbol{j} = A_0 \begin{bmatrix} \cos\psi \\ \sin\psi \cdot e^{i\Delta} \end{bmatrix} \tag{2.50}$$

This form is capable of completely describing any polarization state with $0 \leq \psi \leq \pi/2$ and $0 \leq \Delta \leq 2\pi$, however it only describes the phase relevant to polarization, and not the absolute. The biggest disadvantage of Jones formalism is that it is incapable of describing an unpolarized wave.

The interactions between Jones vectors and a system are described with Jones Matrices.

$$\boldsymbol{J} = \begin{bmatrix} T_{xx} & T_{yx} \\ T_{xy} & T_{yy} \end{bmatrix}, \tag{2.51}$$

which relate input and output as follows:

$$\boldsymbol{j}_{out} = \begin{bmatrix} T_{xx} & T_{yx} \\ T_{xy} & T_{yy} \end{bmatrix} \boldsymbol{j}_{in} \tag{2.52}$$

As can be seen, by probing a system with $\boldsymbol{j}_{|x\rangle}$ and $\boldsymbol{j}_{|y\rangle}$, it is possible to extract the Jones matrix from the system.

$$\begin{bmatrix} T_{xx} & T_{yx} \\ T_{xy} & T_{yy} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} T_{xx} \\ T_{xy} \end{bmatrix} \qquad \begin{bmatrix} T_{xx} & T_{yx} \\ T_{xy} & T_{yy} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T_{yx} \\ T_{yy} \end{bmatrix} \tag{2.53}$$

This of course assumes a linear combination-friendly response from the system, which is not necessarily correct when dealing with plasmonic behaviour. For instance, a volume plasmon will behave differently depending on the charge accumulation points being along a flat surface, or if it's in a corner.

Of course, in a laboratory setting, it is useful to have a form of the Jones formalism that is suitable for the measurable parameters of the EM wave. Expanding on equation (2.50), we have the equation for the absolute state of the EM wave.

$$\begin{bmatrix} E_x(t) \\ E_y(t) \end{bmatrix} = E_0 \begin{bmatrix} \cos \psi \\ \sin \psi e^{i\Delta} \end{bmatrix} e^{i\varphi} \cdot e^{i(kz-\omega t)} \tag{2.54}$$

Here φ is the relative phase difference between the original wave and the current. One can quite confidently measure ψ with just polarization filters. Further, using anisotropic wave retarders, one can find Δ, and finally, using interferometry, one finds φ.

2.4.2 Stokes formalism

The Stokes formalism is a commonly used formalism as it is capable of accurately expressing the macroscopic properties of a system.

Stokes vector

A Stokes vector has 4 elements. The first one is the amplitude, then there are three polarization state values. The innovation of the Stokes vector is that a pure polarization state will give the value zero in the other polarization states [32].

$$S = \begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix} = \begin{pmatrix} I_x + I_y \\ I_x - I_y \\ I_a - I_b \\ I_l - I_r \end{pmatrix} \tag{2.55}$$

Where I stands for intensity, which is commonly normalized to 1 for practical purposes. I_x is the intensity of the wave along the $|x\rangle$ polarization, and the other intensities follow the same logic. A useful property of the Stokes vector is that $I^2 \geq Q^2 + U^2 + V^2$. It is therefore normal to discuss the polarization parameter of a Stokes vector, which is found as follows:

$$P = (\sqrt{Q^2 + U^2 + V^2})/I \tag{2.56}$$

This allows the Stokes formalism to describe macroscopic systems where depolarization happens.

Relating this to the Jones vector form mentioned earlier

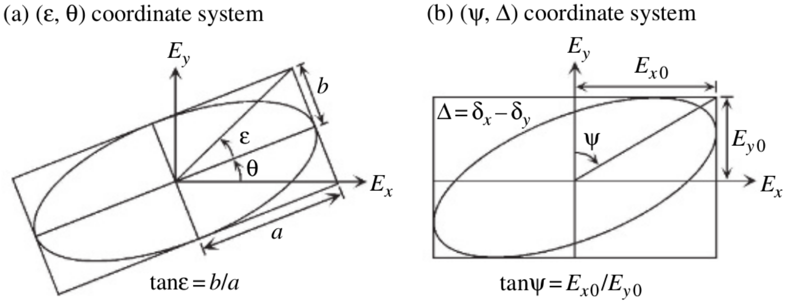
$$\begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi e^{i\Delta} \end{bmatrix} e^{i\varphi} \cdot e^{i(kz-\omega t)} \tag{2.57}$$

One gets

$$S = \begin{pmatrix} I_x + I_y \\ I_x - I_y \\ I_a - I_b \\ I_l - I_r \end{pmatrix} = \mathrm{Re} \begin{pmatrix} E_x E_x^* + E_y E_y^* \\ E_x E_x^* - E_y E_y^* \\ E_x E_y^* + E_x^* E_y \\ iE_x E_y^* - iE_x^* E_y \end{pmatrix} = \mathrm{Re} \begin{pmatrix} (\cos\psi)^2 + (\sin\psi)^2 \\ (\cos\psi)^2 - (\sin\psi)^2 \\ 2\sin\psi\cos\psi e^{i\Delta} \\ -2i\sin\psi\cos\psi e^{i\Delta} \end{pmatrix}$$

$$S = \begin{pmatrix} 1 \\ \cos 2\psi \\ \sin 2\psi\cos\Delta \\ \sin 2\psi\sin\Delta \end{pmatrix} \tag{2.58}$$

which captures the fact that the absolute phase expressed in the Jones vector disappears in the Stokes vector. This form looks similar to the polarization ellipsis derivation of the Stokes vector, which uses the geometric angles, but is different as it is based on the Jones vector elements, and should not be confused with the geometric form. The difference can be seen in figure 2.4.



(a) (ε, θ) coordinate system
E_y
b
ε
θ
E_x
a
tanε = b/a
(b) (ψ, Δ) coordinate system
E_y
E_{x0}
Δ = $\delta_x - \delta_y$
ψ
E_{y0}
E_x
tanψ = E_{x0}/E_{y0}

Figure 2.4: Illustration of the geometric form (a), and the lab frame (b). The lab frame in this image is different from the one used in this thesis, as the Δ used in the thesis is $\Delta = \delta_y - \delta_x$ and belongs to the y-axis element of the Jones vector, while the one in the image has the opposite sign and belongs to the x-axis element. This image was taken from Ref. [33].

Poincaré sphere

A projection method that is common when dealing with polarized light is the Poincaré sphere. The sphere is a three-dimensional representation of the Stokes vector. The sphere's radius is the intensity parameter, I, and $x = Q$, $y = U$, and $z = V$. If the Stokes vector is normalized to 1, it becomes a unit sphere.

A Jones vector will always appear at the surface of the Poincaré sphere, while a Stokes vector can appear anywhere within the sphere as it can represent an unpolarized state as well. The length of the Stokes vector in relation to the Poincaré sphere becomes

$$P = \left(\sqrt{Q^2 + U^2 + V^2}\right)/I \qquad (2.59)$$

with P being a measure of the degree of polarization of the light.

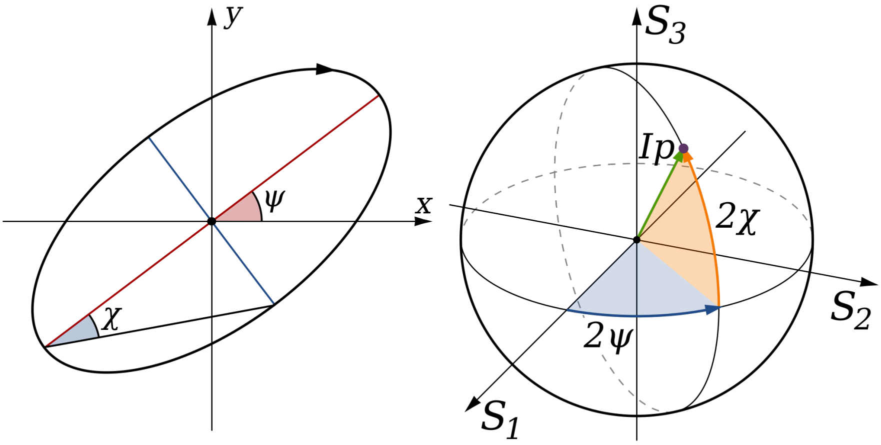


Figure 2.5: Left image: Image of the polarization ellipsis of an EM wave. Right image: The Poincaré sphere as function of the polarization ellipsis. It is important to note that the angles here are different from the ones used in section 2.4.1, and are incompatible with the angle in equation (2.58). Figures are taken from https://en.wikipedia.org/wiki/File:Polarisation_ellipse2.svg and https://en.wikipedia.org/wiki/File:Poincar%C3%A9_sphere.svg

Mueller matrix

A Mueller matrix is the interaction matrix corresponding to the Stokes vector. A perfect X direction polarization filter will have the following Mueller matrix.

$$\boldsymbol{M} = \begin{pmatrix} M_{I,I} & M_{Q,I} & M_{U,I} & M_{V,I} \\ M_{I,Q} & M_{Q,Q} & M_{U,Q} & M_{V,Q} \\ M_{I,U} & M_{Q,U} & M_{U,U} & M_{V,U} \\ M_{I,V} & M_{Q,V} & M_{U,V} & M_{V,V} \end{pmatrix} \qquad S_{out} = \boldsymbol{M}\boldsymbol{S}_{in} \qquad (2.60)$$

Just as with the Jones calculus, the output is determined by multiplying the Mueller matrix with a Stokes vector.

One approach to constructing a Mueller matrix is to use the equations described in either Ref. [34] or [35], which show the operations needed to transform a Jones matrix into a Mueller matrix. Or, one can probe the system with each Stokes vector state and construct it from that.

Due to Mueller matrices not being preferable in this thesis, as FDTD simulations are incapable of simulating unpolarized light, the conversion methods will not be discussed further.

2.5 Coherence width

When using optical systems outside of a purely theoretical framework, coherence is an issue. One can divide it further into temporal and spatial coherence. Since temporal coherence is a laboratory demand, it will not be discussed.

Spatial coherence however, is a concern. In any physical system, the light source will have a finite size. The light source will have a center point of incidence, at which the light is completely coherent, and a radius from that point at which the light gets incoherent, called the coherence width. As shown in Ref. [32] , the coherence width is defined as follows:

$$l_s = \frac{r\lambda}{s} = R\lambda \tag{2.61}$$

Here r is the distance between the source and the system, s is the size of the source (usually the size of the last lens), λ is the wavelength, and l_s is the distance from the center at which maximal incoherence happens.[4] As one can see, as long as the source is small, and the distance large, the spatial coherence width can become quite large.

Relating this back to a physical system, the lab at NTNU that would be the one to carry out an experiment on this system normally operates with a 1 cm wide lens, and a distance of 50 cm, resulting in a total R factor of 50, and a coherence width of about 40 μm for 800 nm light.

[4] For a circular source, the real value is $1.22R\lambda$ due to the summed wave having a Bessel function form instead of a simple sine. But, this level of accuracy is beyond the considerations done in this thesis.

3 Numerics of finite-difference time-domain

Finite-Difference Time-Domain (FDTD) is a categorization of simulations that iterate over time. It is the most intuitive calculation scheme as it is incredibly straight-forward conceptually. One sets up a grid of physical locations, and propagate time in steps by calculating the next state based on the current (and sometimes previous) state.

In this thesis, FDTD will refer to it being applied in EM simulation, specifically with the software ElectroMagnetic Template Library (EMTL).[1] Unless EMTL is mentioned, any method discussed will be a general method used within FDTD EM simulations. Similarly, any competing methods, such as Finite Element Method(FEM), will also be referred to in the context of EM simulations.

In section 2.1 we saw that the fields are related to one another by susceptibility constants due to how they are defined in the SI system, which would require an EM simulation to multiply and divide by these constants every time step. Therefore, it is an ubiquitous method of simplification in the FDTD scheme to set all the base constants to 1.

$$\varepsilon_0 = \mu_0 = c = 1 \tag{3.1}$$

So long as one assumes linearity in all effects, this simplification holds, and makes the $\boldsymbol{E}$ and $\boldsymbol{H}$ fields directly relatable.

Finally, this thesis will focus on the three dimensional FDTD algorithms as there are far fewer interesting problems left to investigate in 2D space than in 3D, as 3D simulations used to be too expensive in runtime before due to computational power. All concepts for the 3D FDTD algorithms are applicable to 2D if one reduces the dimensionality.

3.1 Fast Fourier transform

The algorithms of FDTD are in the time domain, and therefore any data produced by the algorithms will be in the time domain as well. To get phase infor-

[1] http://fdtd.kintechlab.com/

mation out of it, one needs to use a Fourier transform (FT) to convert it into frequency space. Numerically, this is done with a Discrete Fourier Transform (DFT), more specifically the common set of algorithms which most numerical libraries call the Fast Fourier Transform (FFT).

$$\textit{Analytical (FT)},\qquad f(\omega) = \int_{\infty}^{\infty} f(t)e^{-2\pi i t\omega}dt \tag{3.2}$$

$$\textit{Discrete (DFT)},\qquad F_{\omega} = \sum_{n=0}^{N-1} f_n \cdot e^{-2\pi i\omega n/N} \tag{3.3}$$

The most important property of the discrete FFT is what is commonly known as the Nyquist condition: a discrete frequency transform can only output frequencies half of the sample rate or below.[2] Conversely, it can also only output frequencies as low as to have a full period within the complete sample.

However, a third property that is often neglected is that a sharpness relation exists in sampling. Gabor explained in Ref. [36] that the time and frequency standard deviation had a similar relation as the momentum and position has in Heisenberg's principle.

$$\sigma_t\sigma_f \geq \pi/4 \tag{3.4}$$

Here σ_t is the standard deviation in time, and σ_f is the standard deviation in frequency. Because of its similarity to the Heisenberg uncertainty principle, it is also referred to as the "Heisenberg–Gabor limit". To explain this, it is necessary to understand that the DFT creates a Fourier series that is meant to reproduce the input data as best as possible with frequency responses. In other words, trailing zeroed data has an effect in a DFT that is not present in the analytical Fourier transform, as the zeroes will expand the data set and thus expand the frequency range.

[2] Though the condition states half, it is commonly known that the accuracy suffers tremendously when it comes close to half, as it is possible to get a similar effect as wave modulation when the sampling frequency is too low. This is the reason that the standard music sampling is now at 48 kHz, and the peak of outputted frequencies from this is at 22 kHz. Essentially, the top 10% of the frequency spectrum available from the Nyquist condition is deemed inaccurate. This has also been shown in resolution tests of EMTL simulations done prior to the specific work of this thesis.

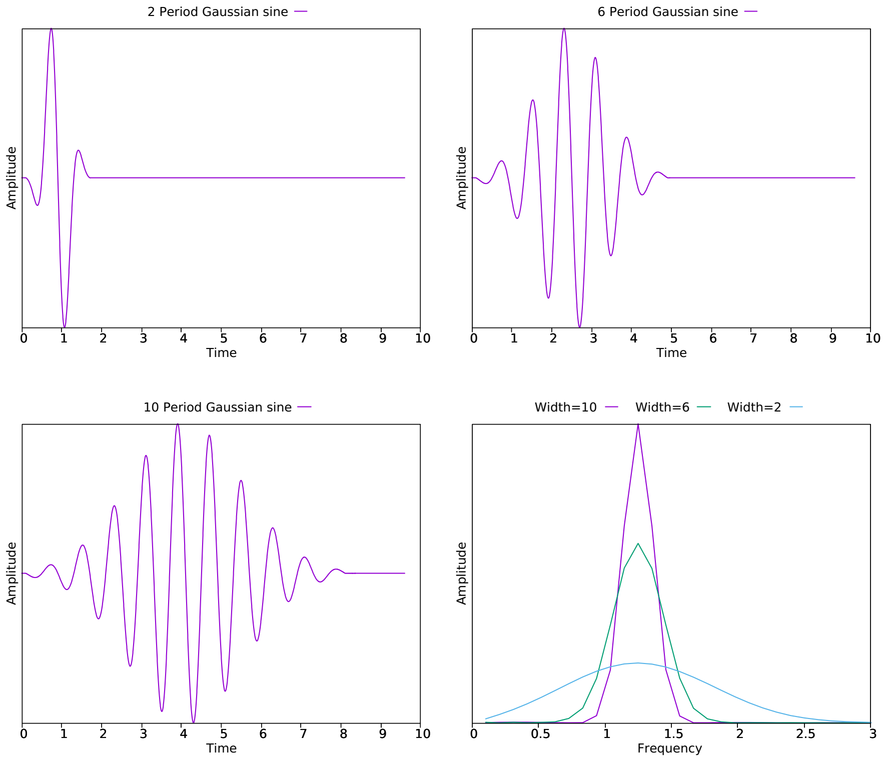


Figure 3.1: Top two images and bottom left: Gaussian modulated sine waves with same sine frequency but different width, axes are amplitude and time. Bottom right: the corresponding amplitude spectra in frequency space, the frequency of the sine wave is 1.25.

As can be seen in figure 3.1, the shorter the wavelet is in time, the wider the frequency spectrum. This has advantages and disadvantages. The obvious advantage is that it makes a frequency sweep incredibly fast, but the disadvantage is that the phase accuracy suffers. Conversely, one can get very high accuracy by using a pure sine wave, but this requires far more time steps and will restrict the results to a narrow frequency space (if not a pure singular peak).

Because the most common DFT method is what's called the ”Fast Fourier Transform” (FFT), the rest of the thesis will refer to FFT instead of DFT.

3.2 Finite difference method

The core principle behind a finite difference method is that it relates an analytical derivation operation to a discrete system. There are well-known finite difference methods outside of numerics as well, such as the trapezoid method and Simpson's method which are discrete integration methods. The idea for discrete derivation is simple; one goes back to the

$$\lim_{dx \longrightarrow 0} d/dx$$

definition of derivation and set $dx = \Delta x$, where Δx is the distance between points. This results in the following equation, called the central difference approximation [37]:

$$dF/dx \simeq (F^{n+1} - F^{n-1})/2\Delta x \tag{3.5}$$

Here F is the function one wants to evaluate, n is the point of evaluation, and Δx is the step length between two neighboring points. Depending on the structure of the system, some methods only use the previous value and the current value, and some use only the current value and the next. For a second derivative, one can apply the approximation for the first derivative one more time and get

$$d^2F/dx^2 = \left(\frac{dF^{n+1}}{dx} - \frac{dF^{n-1}}{dx}\right)\frac{1}{2\Delta x} = (F^{n+2} - 2F^n + F^{n-2})/(2\Delta x)^2 \tag{3.6}$$

we then change the 2 step distance to 1 and adjust the denominator accordingly, resulting in the following central finite difference approximations:

$$dF/dx = (F^{n+1} - F^{n-1})/2\Delta x \tag{3.7}$$

$$d^2F/dx^2 = (F^{n+1} - 2F^n + F^{n-1})/(\Delta x)^2 \tag{3.8}$$

In some cases, such as with time stepping algorithms, or at the edges of a grid, one might not have the points available to do a central difference approximation. In this case, it is common to use what is called a forward or backward finite difference approximation.

$$\text{Forward: } dF/dx = (F^{n+1} - F^n)/\Delta x \tag{3.9}$$

$$\text{Backward: } dF/dx = (F^n - F^{n-1})/\Delta x \tag{3.10}$$

3.2.1 Time domain for electromagnetic solvers

FDTD stands for Finite Difference Time Domain, and is a time stepping algorithm. The first step in constructing the algorithm is to look at the equations governing the physics. The two most relevant equations for an electromagnetism solver are of course the Maxwell equations (2.3) and (2.4).

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$
$$\nabla \times \boldsymbol{H} = J_b + \frac{\partial \boldsymbol{D}}{\partial t}$$

Then, using equations (2.5) and (2.6), and assuming that ε and μ are constants,[3] while having set $\varepsilon_0 = \mu_0 = 1$, one gets:

$$\boldsymbol{D} = \varepsilon \boldsymbol{E} \qquad \boldsymbol{B} = \mu \boldsymbol{H}$$

One can see that as long as there are no external currents the Maxwell equations can be formulated as

$$\nabla \times \boldsymbol{E} = -\mu(\boldsymbol{H}^{t+1} - \boldsymbol{H}^{t})/\Delta t$$
$$\nabla \times \boldsymbol{H} = \varepsilon(\boldsymbol{E}^{t+1} - \boldsymbol{E}^{t})/\Delta t$$

This form looks promising as it relates two fields to each other with a curl differential on one side, and time on the other. As the curl differential is a spatial differential, it will not require information outside of the current time step. The problem is that in this form, both fields require the next time step of the other field to progress. To solve this, FDTD employs a specific grid structure: The Yee cell.

3.3 Yee cells

The first breakthrough in efficiently simulating EM the time domain was the Yee cell [37]. Simply put, because of the nature of EM waves where the $\boldsymbol{E}$-field induces the $\boldsymbol{H}$-field and vice versa, it is optimal to make a grid

[3] This is done so that we can derive the time stepping algorithm for the non-dispersive case before tackling introducing dispersive media to the simulation.

structure that is designed around this relationship. This is where the Yee cell comes in.

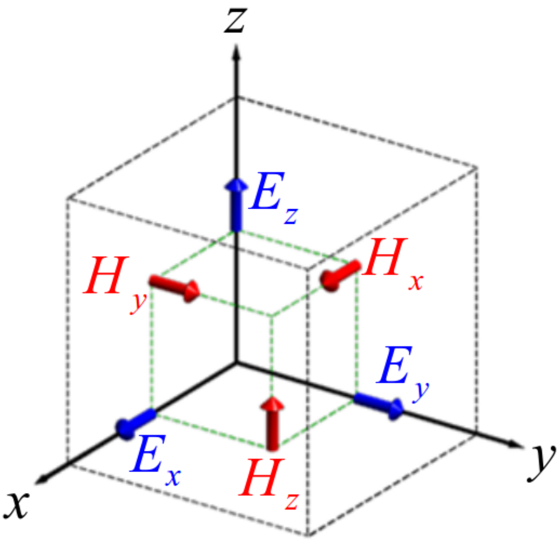


Figure 3.2: Illustration of a 3D Yee cell. The arrows are centered on the location of the field within the grid. This image was taken from http://emlab.utep.edu/ee5390cem/Lecture%2011%20--%20Maxwell%27s%20Equations%20on%20a%20Yee%20Grid.pdf

The defining feature of the Yee cell is that all the fields are displaced. If one says that the physical location of the Yee cell with indices i, j, k, is at x_i, y_j, z_k, the location of $E_z^{i,j,k}$ is at x_i, y_j, $z_k + \Delta z/2$ and $H_z^{i,j,k}$ is at $x_i + \Delta x/2$, $y_j + \Delta y/2$, z_k. This arrangement is for the Yee cell to be efficient in calculating the curl of each field which is crucial for electromagnetic simulations due to the Maxwell curl equations. The downside to the Yee cell is that in order to get a physical field in a point, one has to average varying fields across several Yee cells in order. Using the aforementioned coordinate translation,

$$E_z(x_i, y_j, z_k) = (E_z^{i,j,k-1} + E_z^{i,j,k})/2 \tag{3.11}$$

$$H_z(x_i, y_j, z_k) = (H_z^{i,j,k} + H_z^{i,j-1,k} + H_z^{i-1,j,k} + H_z^{i-1,j-1,k})/4 \tag{3.12}$$

While this displacement seems unintuitive, it confers specific advantages over having the fields localized physically in one point. The numerical advantage will be shown later in this section.

As can be seen, every field of the Yee cell is located separately, however it is also necessary to displace the *H*-field components in time. By using a leap-frogging method which alternates between calculating the E and H field, i.e. to calculate *E*-fields at $t = n\Delta t$ and *H*-fields at $t = (n + 0.5)\Delta t$, the Yee cell grid allows for direct use of the Maxwell curl equations. Adapting the Maxwell equations to the time displaced H field yields

$$\begin{aligned} \nabla \times \boldsymbol{E}^n &= -\mu(\boldsymbol{H}^{t+1/2} - \boldsymbol{H}^{t-1/2})/\Delta t \\ \nabla \times \boldsymbol{H}^{n+1/2} &= \varepsilon(\boldsymbol{E}^{t+1} - \boldsymbol{E}^t)/\Delta t \end{aligned}$$

But this form is not directly useful as the goal of the algorithm is to acquire the value of the next time step. Therefore, we switch it around to express the next time step as a function of the previous step and of current curl values of the opposite field:

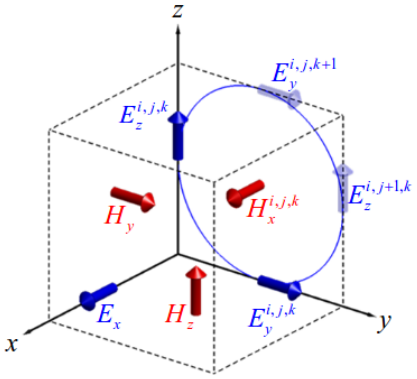
$$\begin{aligned} \boldsymbol{H}^{t+1/2} &= \boldsymbol{H}^{t-1/2} - (\Delta t/\mu)\, \nabla \times \boldsymbol{E}^n \\ \boldsymbol{E}^{t+1} &= \boldsymbol{E}^t + (\Delta t/\varepsilon)\, \nabla \times \boldsymbol{H}^{n+1/2} \end{aligned}$$

Now that the time stepping is sorted out, we address the curl term. Let us look at $\nabla \times \boldsymbol{E}$, which expanded looks like the following:

$$\begin{aligned} \nabla \times \boldsymbol{E} &= E'_x\hat{x} + E'_y\hat{y} + E'_z\hat{z} \\ E'_x &= \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \\ E'_y &= \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \\ E'_z &= \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \end{aligned}$$

We then use Stokes theorem which translates the curl to a sum of field lines along a circumference around the point.

$$\frac{\partial E_z}{\partial y'} - \frac{\partial E_y}{\partial z'} = \mu_{xx}\tilde{H}_x$$



$$\frac{E_z^{i,j+1,k} - E_z^{i,j,k}}{\Delta y'} - \frac{E_y^{i,j,k+1} - E_y^{i,j,k}}{\Delta z'} = \mu_{xx}^{i,j,k}\tilde{H}_x^{i,j,k}$$

Figure 3.3: Illustration of the calculation method for the value of the *H*-field's *x*-component in the Yee cell and the related equations. Taken from http://emlab.utep.edu/ee5390cem/Lecture%2011%20--%20Maxwell%27s%20Equations%20on%20a%20Yee%20Grid.pdf

As can be seen, the Yee cell allows for one field component to be updated by a single curl calculation, and each since the two fields are updated at separate time steps, each field value can be updated immediately. Going back to the naive approach mentioned at the start of the section, i.e. if all fields were localized in at the same point, the curl would have to use the four neighboring points in the relevant plane. This means that the effective radius with regards to curl calculations of a non-Yee cell grid is doubled. The doubling of the radius means that the resolution is effectively halved in all dimensions. For a 3D FDTD algorithm, this has the consequence of making everything at least 8 times more expensive computationally, which is a significant slowdown.

Though most implementations only include a rectangular grid, it is possible to make a Yee cell that is not rectangular, but unless there is some structural property that allows for a simple conversion to be used, it will normally be more suitable to use a FEM based solver instead, as it is more specialized towards irregular geometries.

3.3.1 The Courant–Friedrich–Lewy condition

The most important restriction of the Yee cell lies in the Courant-Friedrich-Lewy (CFL) condition [38].[4] The Courant factor, C, describes the effective propagation speed of the wave with regards to time and spatial resolution.

$$C = \sum_{i=1}^{D} \frac{u_i \Delta t}{\Delta x_i} \leq \sqrt{D} \tag{3.13}$$

Here, Δt refers to the time step used, while i is the direction. u_i is the propagation speed in said direction, and Δx_i is the width of the Yee cell in said dimension. The effective speed of a wave cannot exceed the square root of the dimensionality of the problem, and it is also a condition that any of the terms in the sum should not exceed 1 as well.

Another way of understanding this restriction is to go back to looking at how the field in a physical point is calculated

$$\begin{aligned} E_z(x_i, y_j, z_k) &= (E_z^{i,j,k-1} + E_z^{i,j,k})/2 \\ H_z(x_i, y_j, z_k) &= (H_z^{i,j,k} + H_z^{i,j-1,k} + H_z^{i-1,j,k} + H_z^{i-1,j-1,k})/4 \end{aligned}$$

With this, one can see that a key assumption in the Yee cell is that the field is only affected by neighboring field points. Therefore the time step length has to be such that this assumption is valid.

For a general 3D FDTD application, with a uniform grid, speed of light set to 1, the equation simplifies to the following:

$$C = \frac{3\Delta t}{\Delta x} \leq \sqrt{D} \tag{3.14}$$

[4] This paper is originally from 1928, and has been expanded upon by others, but the foundation of the CFL condition is from here, and is the name everyone uses.

The maximal value, $\sqrt{D}$ - often referred to as C_{max}, is the limit to how large a time step can get before numerical accuracy is lost. The default value is usually either set to be just below C_{max} , $\sqrt{3}$ in the case of $3D$, or below. Thus, common time resolutions for $3D$ problems are therefore $\Delta t/\Delta x = 0.5$ or 0.25, as they will satisfy having a Courant number of $C = 1.5 < \sqrt{3}$ and $C = 0.75$ respectively.

Relating this back to the FFT, one sees that the CFL condition also determines the peak frequency one can get out, due to the Nyquist condition. In fact, the CFL condition can be considered a reformulation of the Nyquist condition that applies to time stepping simulations. Thus, it is useful to know that the peak frequency outputted by an FDTD simulation is directly tied to the grid resolution. Specifically for 3D, with the CFL parameter set to two timesteps per grid length, the shortest wavelength obtainable in an FDTD simulation will be the grid resolution.

Throughout the thesis, I have used a time resolution of $\Delta t/\Delta x = 0.25$, which has a Courant value of $C = 0.75$.

3.4 Dispersive media

Now that the behaviour of non-dispersive media is accounted for, it is time to go back to the assumption of ε and μ being constants. Because magnetism has unfortunate properties with regards to simulation, such as inhomogeneity at the nanometer level in certain materials, often linked to quantum mechanical effects, it is often left out of nanoscale FDTD simulations. Furthermore, it is not relevant for the systems being investigated in this thesis as gold can be considered non-magnetic. Therefore, we keep the focus on the dielectricity function, and resume where we left off in section 2.2.3.

3.4.1 Auxiliary differential equation

When constructing the primary time stepping loop, the assumption was made that $\varepsilon = \text{const}$, but this will not allow us to simulate materials with any significant interactions, rendering the software worthless. Therefore, we have to set $\varepsilon = \varepsilon_\infty + \sum_j \chi_j$. Because of magnetism either being weak and close to constant, such as paramagnetism and diamagnetism, or strong and non-linear, such as ferromagnetism, we assume that μ is a constant, and therefore will not alter the core behaviour of the basic time step algorithm.

Recalling the relevant curl equation and the constitutive relationships from section 2.1, we have:

$$\begin{aligned}
\nabla \times \boldsymbol{H} &= \frac{\partial \boldsymbol{D}}{\partial t} \\
\boldsymbol{D} &= \varepsilon_0 \boldsymbol{E} + \boldsymbol{P} \\
\boldsymbol{P}(\omega) &= \varepsilon_0 \chi_e(\omega) \boldsymbol{E}(\omega) \\
\boldsymbol{J}(\boldsymbol{k}, \omega) &= i\omega \boldsymbol{P}(\boldsymbol{k}, \omega)
\end{aligned}$$

We also need to recall the modified Lorentz model:

$$\chi_i(\omega) = \frac{a_{0,i} - i\omega a_{1,i}}{b_{0,i} - i\omega b_{1,i} - \omega^2} \tag{3.15}$$

And from here, we can resume the thread from section 2.2.3 where the analytical expression for the induction of polarization currents by EM waves had gotten the following form:

$$b_{0,j}\boldsymbol{J}_j(t) + b_{1,j}\frac{d\boldsymbol{J}_j(t)}{dt} + \frac{d^2\boldsymbol{J}_j(t)}{dt^2} = a_{0,j}\frac{d\boldsymbol{E}(t)}{dt} + a_{1,j}\frac{d^2\boldsymbol{E}(t)}{dt^2}$$

The subscript j is a numerator for polarization currents, and each polarization current is assumed to be independent of each other. Converting it to a finite difference form yields:

$$\begin{aligned}
&b_{0,j}\boldsymbol{J}_j^t + b_{1,j}(\boldsymbol{J}_j^{t+1} - \boldsymbol{J}_j^{t-1})/2\Delta t + (\boldsymbol{J}_j^{t-1} - 2\boldsymbol{J}_j^t + \boldsymbol{J}_j^{t+1})/(\Delta t)^2 \\
&= -a_{0,j}(\boldsymbol{E}^{t+1} - \boldsymbol{E}^{t-1})/2\Delta t - a_{1,j}(\boldsymbol{E}^{t-1} - 2\boldsymbol{E}^t + \boldsymbol{E}^{t+1})/(\Delta t)^2
\end{aligned}$$

Sorting these out one gets:

$$\begin{aligned}
&\boldsymbol{J}_j^{t+1}\,(\Delta t\, b_{1,j} + 2) && + \boldsymbol{J}_j^t\,(2\Delta t^2 b_{0,j} - 4) && + \boldsymbol{J}_j^{t-1}\,(-\Delta t\, b_{1,j} + 2) \\
&= \boldsymbol{E}^{t+1}\,(-\Delta t\, a_{0,j} - 2a_{1,j}) && + \boldsymbol{E}^t\,(4a_{1,j}) && + \boldsymbol{E}^{t-1}\,(\Delta t\, a_{0,j} - 2a_{1,j})
\end{aligned}$$

It is worth noting that the only variable that by itself is able to fundamentally change the number of terms is a_1. Any permittivity that has an a_1 value will have optimal behaviour in the Auxiliary Differential Equation (ADE) designed for the general case of the Modified Lorentz model, even if another parameter is zero. However, if two parameters or more are missing, there is a lot of room for optimization.

Remembering the form of the Drude model, which has no a_1 or b_0 term, one sees that the equations become:

$$\begin{aligned} &\boldsymbol{J}_j^{t+1}\,(\Delta t\, b_{1,j}+2) && +\boldsymbol{J}_j^{t}\,(-4) && +\boldsymbol{J}_j^{t-1}\,(-\Delta t\, b_{1,j}+2) \\ =\,&\boldsymbol{E}_j^{t+1}\,(-\Delta t\, a_{0,j}) && && +\boldsymbol{E}_j^{t-1}\,(\Delta t\, a_{0,j}) \end{aligned}$$

Debye on the other hand does not have any term which becomes ω^2 before the transformation from the frequency domain to the time domain, and therefore no second derivatives are present. This allows one to use the trick of centering the equation around the time $t+1/2$, which results in a finite difference equation with only t and $t+1$ as its time steps.

$$\boldsymbol{J}_j^{t+1} = \frac{1-\Delta t/2\tau}{1+\Delta t/2\tau}\boldsymbol{J}_j^{t} + \frac{\Delta\chi/\tau}{1+\Delta t/2\tau}(\boldsymbol{E}^{t+1}-\boldsymbol{E}^{t}) \tag{3.16}$$

This shows why Debye especially, but Drude as well, benefits from having its own routine instead of being implemented with the Modified Lorentz Model.

Because the eqautions relate the polarization currents and electric field for the next time step at the same time, an implicit iteration scheme is used to solve these equations. First one estimates the next electric field value, then, using that estimate, one updates the polarization currents, and finally one updates the electric field again.

3.4.2 Implementing measurement data in FDTD

Now that the numerical handling of susceptibility has been shown, we need to obtain the susceptibility models. The normal method is use measurement data from real experiments and try to fit them to Drude, Debye, and

Modified Lorentz as best as possible. This is typically done by using curve fitting libraries in MatLab, SciPy or academic software kits, e.g. specialized ellipsometric software. In the case of Modified Lorentz terms, the simplest method is to simply look at the data and find the number of peaks, and use a curve fit with that many modified Lorentz poles as a first guess. If the fit is insufficient, one can try to add extra poles.

EMTL has a downloadable Matlab-script on their website which can be used to generate parameters for FDTD.

Gold model parameters

The FDTD solver used in this thesis, EMTL, offers a built in gold model based on Ref. [39], which used Drude + Critical Points. These values are implemented as terms in the modified Lorentz model. And, as discussed in section 2.2.2, the critical point model is a special case of the Modified Lorentz Model.

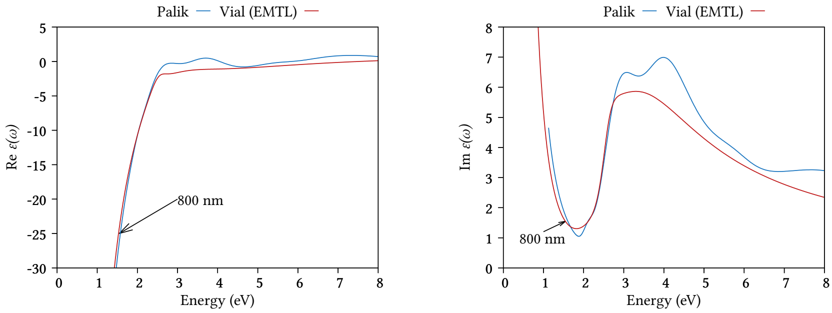


Figure 3.4: Plot of the susceptibility values of gold with respect to the energy of the incoming EM waves. EMTL's model is based on Vial's Drude + 2 critical points model [39], plotted in red, and is compared here with the litterature values measured by Palik [40] in blue. The 800 nm point, or 1.55 eV, is highlighted with arrows on both plots.

Figure 3.4 shows that the built in model is mostly accurate at the wavelength used in this thesis, 800 nm, but starts deviating from the values from Ref. [40]

at lower wavelengths. This is likely to be a error source in wide-frequency simulations.[5]

3.5 Boundary conditions

Having described how the inside of the simulation works, the next step is to describe the boundaries. Because the Yee cell grid updates values depending on the values of the neighbors, the boundaries have to satisfy certain conditions. First is that a wave should be able to traverse into the boundary and disappear. Second is that a wave travelling parallel to it should not be affected by it. As such, there are two types of boundaries.

3.5.1 Periodic boundary condition

The periodic boundary condition is mathematically straight-forward. Simply put, all edge cells are set to border the edge cells at the opposite boundary. One important consequence of this is that periodic boundaries are almost always planar, as one prefers a direct link between each boundary. It is theoretically possible to implement periodic boundaries on systems that are not describable by opposing parallel planes, but these will either represent unphysical systems, or require peculiar mathematics at the boundaries. Therefore, for all practical intents and purposes a periodic boundary condition is a set of paired planes with linked coordinates. Because EMTL only has rectangular cuboids available as the simulation environment, only this will be discussed. Expressed mathematically, if a uniform rectangular two dimensional grid has periodic boundaries, with points $F_{n,m}$ on the grid having indices $n \in \{0, 1...N\}$ and $m \in \{0, 1...M\}$, the boundary appears as the following equations:

$$F_{n,M+1} = F_{n,0} \qquad\qquad F_{N+1,m} = F_{0,m} \tag{3.17}$$

[5] Figure 2.1 has a curve much closer to the model based on Vial, so the question becomes which measurements one trusts the most. Since Palik's measurements are newer, I assume they're probably more correct.

However, in 3D simulations it is highly unlikely that one wants a periodic boundary in all directions, so one will normally only have that one or two of the principal axes are periodic.

3.5.2 Perfectly matched layer

Seeing that having periodic boundaries along all axes is undesirable, a second type of boundary condition is needed. The general term for this type of boundary condition is absorbing boundary condition. The first thought one could have would be to have a border that is just zero at all times. This does not work as it would make large reflections since the derivatives at the border would be as large as the fields themselves.

The solution was to create an anisotropic boundary that is meant to emulate the coordinate system stretching to infinity, which does not ruin the behaviour of waves running parallel to the boundary. This is what is called the Perfectly Matched Layer (PML). The derivation of the original PML is somewhat lengthy, so this section will go through the main points quickly and not in great depth. For those interested in a full derivation, I recommend the lecture slides put public by The University of Texas at El Paso..[6] Alternatively, one can see the derivation done in Ref. [37].

The first step of deriving a PML is to propose a uniaxially anisotropic boundary with a stretching tensor $[S]$.

$$\nabla \times \boldsymbol{H} = \frac{\partial}{\partial t}[S][\varepsilon]\boldsymbol{E} \tag{3.18}$$

Now, looking at a PML that is meant to cover the x-direction, and remembering that it should be uniaxial, one gets:

$$[S_x] = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & b \end{bmatrix} \tag{3.19}$$

[6] http://emlab.utep.edu/ee5390fdtd/Lecture%2013%20--%20The%20Perfectly%20Matched%20Layer.pdf

The next step is to minimize reflections from the boundary. This is done by inserting $[S_x]$ into the Fresnel equations and finding the result that gives 0 reflections, resulting in the following demands:

$$\sqrt{ab} = 1 \qquad\qquad \mathrm{Re}(a) = 1 \tag{3.20}$$

And so, we get

$$[S_x] = \begin{bmatrix} s_x^{-1} & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{bmatrix} \tag{3.21}$$

As values to satisfy these demands, Berenger proposed in Ref. [41] the following factor:

$$s_x = 1 + \frac{\sigma_x}{i\omega} \tag{3.22}$$

To generalize and compress the notation, it is common to write

$$[S] = [S_x][S_y][S_z] = \begin{bmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{bmatrix} \tag{3.23}$$

with the components s_x, s_y, $s_z \neq 1$ only when within the corresponding boundary. This is commonly known as the uniaxial PML, or UPML, and has the final form of:

$$\nabla \times \boldsymbol{H} = [S]\varepsilon\boldsymbol{E} \qquad\qquad \nabla \times \boldsymbol{E} = [S]\mu\boldsymbol{H} \tag{3.24}$$

Having established the simplest PML method, it now behooves us to explore it further. Continuing from equation (3.24), one can bring the PML tensor over to interact with the curl instead of the field.

$$[S]^{-1}\nabla \times \boldsymbol{H} = \varepsilon\boldsymbol{E}$$

This form has the distinct advantage of the PML tensor interacting with the curl alone, and not the susceptibility. This is valuable in case of an anisotropic susceptibility.

Now, multiplying the inverse PML tensor with the curl tensor yields

$$[S]^{-1}(\nabla\times) = \begin{bmatrix} 0 & -\frac{s_x}{s_y}\left(\frac{1}{s_z}\frac{\partial}{\partial z}\right) & \frac{s_x}{s_z}\left(\frac{1}{s_y}\frac{\partial}{\partial y}\right) \\ \frac{s_y}{s_x}\left(\frac{1}{s_z}\frac{\partial}{\partial z}\right) & 0 & -\frac{s_y}{s_z}\left(\frac{1}{s_x}\frac{\partial}{\partial x}\right) \\ -\frac{s_z}{s_x}\left(\frac{1}{s_y}\frac{\partial}{\partial y}\right) & \frac{s_z}{s_y}\left(\frac{1}{s_x}\frac{\partial}{\partial x}\right) & 0 \end{bmatrix} \tag{3.25}$$

One point of interest is that now, we consider only the terms directly connected to the directional derivative to be valid, i.e. the fractions before the parentheses can be dropped. Thus, one is left with

$$[S]^{-1}(\nabla\times) = \begin{bmatrix} 0 & -\frac{1}{s_z}\frac{\partial}{\partial z} & \frac{1}{s_y}\frac{\partial}{\partial y} \\ \frac{1}{s_z}\frac{\partial}{\partial z} & 0 & -\frac{1}{s_x}\frac{\partial}{\partial x} \\ -\frac{1}{s_y}\frac{\partial}{\partial y} & \frac{1}{s_x}\frac{\partial}{\partial x} & 0 \end{bmatrix} \tag{3.26}$$

This is often called stretched coordinate PML. This is because it can also be derived by introducing a stretched coordinate system as follows:

$$\tilde{x} = \int_0^x s_x(x')dx' \qquad \Rightarrow \qquad \frac{\partial}{\partial \tilde{x}} = \frac{1}{s_x}\frac{\partial}{\partial x} \tag{3.27}$$

with the y and z axes having the same transformation with the stretching parameters s_y and s_z respectively. This results in

$$\tilde{\nabla} = \hat{x}\frac{1}{s_x}\frac{\partial}{\partial x} + \hat{y}\frac{1}{s_y}\frac{\partial}{\partial y} + \hat{z}\frac{1}{s_z}\frac{\partial}{\partial z} \tag{3.28}$$

This nabla operator results in the same stretched curl operator as in equation (3.26).

Convolutional PML

The stretched coordinate PML is often called a convolutional PML (CPML), due to the transform from frequency into time domain manifesting as a convolution of the PML and the field.

The transformation from frequency to time domain can be done in the same manner as was done in section 2.2.3. It is then possible to implement this in FDTD with the same finite difference scheme used in section 3.4.1, resulting in a numerical approximation to the convolution of the media and the field, hence the name.

However, many choose to refer to CPML as the implementation pioneered in Ref. [42]. This implementation revisits the core proposition of Berenger's PML coordinate stretching [43]:

$$s_i = 1 + \frac{\sigma_i}{i\omega} \tag{3.29}$$

and replaces it with a simplified version of the more advanced stretching metrics suggested by Ref. [44]:

$$s_i = \kappa_i + \frac{\sigma_i}{\alpha_i + i\omega} \tag{3.30}$$

This modified coordinate stretching, while more expensive in runtime, results in a PML capable of bordering a material with dispersive behaviour as it will correctly dampen and extinguish all behaviours going into the PML without causing any feedback loops such as those that can happen with a UPML.

3.6 Signal waves

Having established the numerical system underlying the interactions of EM waves in the simulation, we need to specify how to stimulate the system in a manner to produce output. The most common solution is to produce an incoming EM wave.

3.6.1 Sine wave

A sine wave is simply that. It is a wave of a sine form, with a period and a phase. A wave travelling in the z direction and that is polarized along x-axis will have the form:

$$E_x(t) = E_0 \sin(\omega t + \omega_0) \tag{3.31}$$

$$\tag{3.32}$$

Adding another wave of equal intensity that has an orthogonal polarization and $\pi/2$ in phase difference, one can get a circularly polarized wave. If the amplitude of the secondary wave is less then it becomes an elliptical wave instead.

$$E_x(t) = E_0 \sin(\omega t + \omega_0) \tag{3.33}$$

$$E_y(t) = E_0 \sin(\omega t + \omega_0 \pm \pi/2) \tag{3.34}$$

3.6.2 Gaussian pulse

The core shape of a Gaussian pulse is as follows:

$$E(t) = E_0 \exp\left[-\left(\frac{t - t_0}{t_w}\right)^2\right]. \tag{3.35}$$

This pulse in of itself is not very useful for an FDTD scheme as it is highly unphysical as an EM wave will not only have a positive part, but also a negative. Therefore, Gaussian pulses are normally used as a modulation pulse in the context of EM waves, as is shown by the next two pulse types.

3.6.3 Berenger pulse

A Berenger pulse is a pulse that is described by the following equation:

$$E(t) = E_0(t - t_0) \exp\left[-\left(\frac{t - t_0}{t_\omega}\right)^2\right] \tag{3.36}$$

As can be seen, it is a Gaussian-modulated straight line that crosses 0 at t_0, which is also the same point as the peak of the Gaussian curve. It turns out that this is a stable waveform for an FDTD Yee cell scheme. And, as it is anti-symmetrical around 0, it is more physical than the pure Gaussian pulse, while retaining the wide phase spectrum that comes from being a pulse. In EMTL, the Berenger pulse is normally automatically set to cover the maximal width of frequencies, and has circular polarization disabled. These two properties are probably related, as the phase accuracy of the wave is likely to suffer at the highest frequencies.

3.6.4 Gaussian modulated sine wave

A different pulse type that uses a Gaussian pulse as its basis is the sine modulated Gaussian pulse.

$$E(t) = E_0 \exp\left[-\left(\frac{t - t_0}{t_w}\right)^2\right] \sin(\omega(t - t_0)). \tag{3.37}$$

A Gaussian-modulated sine wave can be circularly polarized as well. This pulse type has some advantages as it will contain several frequencies with the sine frequency being the dominant one. In other words, it can be considered a hybrid pulse between the Berenger and the Sine wave as it has the advantages and disadvantages of both, but at different scales.

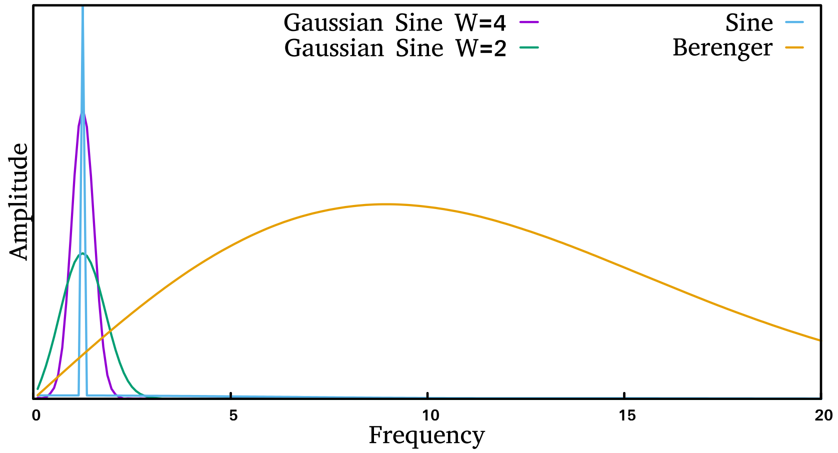


Figure 3.5: Frequency spectrum of the different waveforms. Amplitudes are arbitrary, and have been adjusted to illustrate the characteristics of the waveform. The Berenger pulse is the default for the resolution of 5 nm per mesh step.

3.6.5 Total field / scattered field

Now that the different signals have been established, it is necessary to find a way to generate the signal. A common method is to use the total field-scattered field (TF/SF) method. In this, one defines a region that is exempt from the signal wave, the Scattered Field region, and a region that is not, the total field region. At the borders between the regions, the calculated values of the wave as it should pass through the border undisturbed get implemented into the simulation. This counts both at the entry and exit of the total field region. As such, any deviation from the pattern one would get from an undisturbed wave will manifest itself at the TF/SF border and propagate into the scattered field.

A good example is if one has a mirror in the middle of a TF/SF region, which reflects the entire wave. In the direction of the original wave, the expected wave does not hit the border as expected, and a wave with opposite phase manifests at the border and propagates into the scattered field. Meanwhile, in the direction of the reflection, a wave that is unexpected at the border appears, and therefore passes through the border and propagates unhindered in the scattered field.

3.7 Detectors

With all that has been discussed, a complete overview of the FDTD EM scheme has been given. However, there is still one thing missing, and that is how one extracts data out of the simulation. For this, one sets points in the simulation space aptly called detectors. A detector is a single point that is meant to record all fields with a given sampling rate. An important distinction between a detector and a Yee cell is that the detector is a singular point, while the Yee cell consists of 6 points.

Thus, a detector in FDTD has to use equations (3.11) and (3.12) to re-localize the fields.

$$
\begin{aligned}
E_z(x_i, y_j, z_k) &= (E_z^{i,j,k-1} + E_z^{i,j,k})/2 \\
H_z(x_i, y_j, z_k) &= (H_z^{i,j,k} + H_z^{i,j-1,k} + H_z^{i-1,j,k} + H_z^{i-1,j-1,k})/4
\end{aligned}
$$

It is also possible to have detector positions not aligned to the grid, in which case one would interpolate the field values from the nearest surrounding data points.

The detector is in principle agnostic to its purpose afterwards, but it is useful to have categories that enable one to accurately use the detectors for what they are best suited for.

3.7.1 Planar detector sets

A planar detector set is a set of detectors that are specified only to cover one or several planes. This categorization is useful as one can set up a complete boundary, such as a plane above a structure which has a wave sent down on it, so that the detector plane picks up any reflections made.

This arrangement is most useful when the simulation has periodic boundaries.

3.7.2 Box detector sets

A box detector set is formed as a box surrounding an object in the simulation, with points uniformly spread on each surface of the box. This type is often ideal for simulations that have a particle suspended in an isotropic media, such as gold nanoparticles in water.

This type should only be used when the simulation is encapsulated by PMLs.

3.7.3 Nearfield detector set

A nearfield detector set is a freely defined set where one can fill a specified line, plane or volume with detectors. This detector type has the most flexibility, but also takes the most manual effort to produce worthwhile data. As a result, it is common that the nearfield detector set is the one that outputs the data in the most raw form compared to the others. Both planar and box detectors are normally used as flux detectors, which automatically FFTs and averages the data to find flux through the detector set's surfaces. A nearfield detector will however output its data in one of the following ways:

- Time domain output per point
- Frequency domain output per point.
- k-space domain for the set, in time values.
- k-space domain for the set, in frequencies.

The last two are of course very sensitive to the specific spacing of the detectors within the set, so it is often that one only considers the first two as the output types, and use manual post-processing to get the information one wants from the detector set.

3.7.4 Near to far transformation

There is another transformation that is useful for FDTD, and that is near to far transformation. If your simulation has the internal scale of 100 nm and you wish to observe something from a distance of e.g. 10 cm, extending the simulation environment to include this distance should be out of the question, so you need to transform the nearfield data into farfield results. To do this, the Stratton-Chu equations are typically used.

The Stratton-Chu equations are the application of Green's theorem to EM waves, originally derived in Ref. [45], but most modern applications use either Ref. [37] or [46] as their basis for modern implementations. The core principle is to have a plane of detectors recording all EM values over time. An FFT is then run on each point, and the results of said FFT are propagated onto each output point on the far field surface. After all detector outputs have propagated onto all points on the far field projection plane, one records the resulting phase and amplitude of every point.

3.8 Numerical accuracy

Numerical errors and other error sources are a crucial part in any simulation-based research as one needs to know if the results are reliable, and to which degree they are.

3.8.1 Causes of numerical inaccuracy

It is generally common to split numerical errors into categories depending on how the arise. One can roughly divide them accordingly:

- Accumulated floating-point errors
- Aliasing
- Algorithmic flaws

Accumulated floating-point error is an error type that is very hardware and implementation sensitive. It can best be explained as the sum of all rounding errors that arise from floating point arithmetic. In the simple case of summation, it can be understood as a Brownian walk with the rounding error intrinsic to the data structure being the step distance. In relation to FDTD, so long as one uses double-precision floating-point numbers, one would be hard pressed to experience any error. However, if you use a GPU-based solver, and do not have an Nvidia Tesla card[7] or equivalent, you will end up using single precision floats. These have a precision up to 7 decimals, which opens up for accumulated numerical errors. Therefore, it is the particular case that increased number of iterations can harm numerical accuracy instead of improving it. This is also linked to the exact method of implementation, as one can minimize the error by ensuring that all operations are between numbers of the same magnitude and other tricks. Especially critical is the floating-point error that can arise if a number is subtracted from an almost identically large number, in which the floating point precision error can become large relative to the result. If this number is then multiplied, or even worse, divided by, it can create enormous errors from just a few operations. In all modern solvers, double-precision floating-point numbers are used, unless one wants to use a GPU. A double-precision floating-point number has 15 decimals accuracy, which means that the accumulated floating-point error will roughly follow the biggest error estimate of

$$\mathrm{Err} \simeq \sqrt{N} \times 10^{-15} \times M_{\mathrm{diff}}, \tag{3.38}$$

where N is the number of iterations done (in the case of FDTD, per point is most suitable), 10^{-15} is the floating point accuracy of a double, and M_{diff} is the average difference in exponentials of the numbers being operated on. This calculation is mostly relevant for additions and subtractions, as multiplication is exponent-agnostic. All simulations in this paper were done with double-precision, and the FDTD iteration scheme has no algorithmic exposure to the extreme cases of floating-point errors, and so, floating-point errors are deemed irrelevant for this thesis. It is nevertheless briefly discussed here as EMTL has a CUDA version, which can be compiled to use single precision floating point numbers to be used on a standard gaming-

[7] These are the most common double-precision GPUs made specifically for the purpose of numerics.

oriented Nvidia GPU, which makes this relevant for others wanting to use the software.

Aliasing is when the resolution is too low to accurately represent the system. More specifically, the term is commonly used when the resolution introduces deformities. In FDTD, unless one uses a specialized Yee grid, one will have the problem that geometries will have a mismatch relative to the grid. The simplest example is that a sphere will have to be made by tiny blocks. There are algorithms to reduce the impact of the problem, such as subpixel smoothing [47], but these algorithms cannot eliminate the problem entirely. If one is at the limits of the capabilities of the hardware (or software), aliasing can end up being a problem one just has to accept.

Finally, algorithmic flaw is quite simply what it sounds like. One can generally split this category in two: bad numerical stability and bad/wrong implementation. The best example of the first kind is Euler's method. This method accumulates error as it progresses, and has no correction mechanisms. The second kind is simply that the code is wrong. Aside from just having made erroneous code, it can also be that assumptions were made that were invalid in the numerical scheme. One example from earlier is to mix finite difference approximations with different center-points. This flaw is particularly insidious as it will often not produce errors, and if it does, they are often small enough to go unnoticed for a long time.

In this thesis, all numerical error is assumed to be the consequence of aliasing and the general limitations of FDTD algorithms.

3.8.2 Methods of error estimation

As can be understood, it is possible to track down and quantify the different error types based on their behaviour.

The first step is simply to compare graphs to an already solved problem. For FDTD solvers, this is often Mie scattering. This is of course quite crude, but is often the most time efficient method as one can instinctively judge the magnitude of the error. If the error is qualitative, it normally points towards aliasing or an algorithmic flaw. Even if there is no algorithmic flaw it is normally still useful to do this as one gets a rough mental estimate of the error.

The next step is to start applying mathematical methods to quantify the error. The method I have chosen to use as the fundamental error metric is the L^2-Method (written here for time steps, but it is unit-agnostic).

$$L^2 = \frac{\int f(t) \cdot g(t)dt}{\sqrt{\left(\int f(t) \cdot f(t)dt\right) \cdot \left(\int g(t) \cdot g(t)dt\right)}} \tag{3.39}$$

This method checks the correlation factor between two functions. It has the property of being amplitude-agnostic, which is desirable for many systems. As can be deduced, the function has values ranging from -1, which is a perfect match with opposite signs, to +1 which is a perfect match with matching signs, and a value of 0 is a perfect mismatch with either partial anti-symmetry such as x vs $|x|$, or perfect mismatch such as $u(x)$ vs $u(-x)$ (where u is the unit step function), no values at all, or a combination of the three. If one has only positive values for both f and g, then the range is from perfect mismatch, 0, to perfect match, 1. The L^2 method is best suited for results where a curve shift, i.e. $f(x) = g(x + a)$ is either unlikely, or undesirable.

Of course, this is the analytical form, which is not particularly useful for a numerical data set. Therefore, one can use a discrete integration method such as the trapezoid method or Simpson's method. As is well-known, these methods have varying accuracy. In general, the accuracy is related to the complexity of the model. In this thesis, a simple trapezoid method will be used as it is easy to implement and can handle varying resolution along the variable axis. In the finite difference domain, the trapezoid method needed for the L^2 analysis has the following form:

$$\int f(t) \cdot g(t)dt \simeq \sum \Delta t_i^2 \cdot (f_i + \Delta f_i) \cdot (g_i + \Delta g_i) \tag{3.40}$$

$$= \sum (t_{i+1} - t_i)^2 \cdot [(f_{i+1} + f_i)/2] \cdot [(g_{i+1} + g_i)/2] \tag{3.41}$$

This method is often used to quantify error estimates from resolution comparisons. Resolution comparison is often the best method of estimating error when analytical solutions are not possible. In short, the method is to find a system that has the same properties as the system you wish to simulate, but is small enough to allow for much higher resolution than what you can use for the main system. Simulating this system at a range of resolutions will allow for finding both the breaking point resolution, i.e. when the results become qualitatively wrong, and the relative accuracy at different resolutions.

3.9 Runtime and memory scaling

Having gone through all the components of an FDTD EM solver, taking a look at the numerical efficiency is the final consideration. We know that a Yee cell contains 6 real numbers, and that for dispersive materials, the previous time step's values are needed as well. The detectors need to either record all time steps, or continuously do FFT calculations per time step and store the frequency spectra instead. Optimally, one can assume that N(detectors) << N(grid points), N being the total number of discrete points, and therefore detectors have a negligible impact on memory and runtime.

Thus, we have a memory demand that scales with the number of grid points, which in big-O notation becomes $\mathcal{O}(x^3)$, with x being number of points in each direction (assuming a cubic simulation environment). However, in the case of runtime, the time resolution is directly tied to the spatial resolution because of the CFL condition, resulting in $\mathcal{O}(t' \cdot x^4)$, with t' being the simulation time parameter, adjusted so that $t' \cdot x = N(\text{timesteps})$.

Thus, the expected scaling is that doubling the resolution along all dimensions should increase the simulation time by 16 times and memory usage by 8 times. This is however not necessarily the case as modern processor vectorization makes any repeated operation highly optimized, resulting in all calculations being streamlined at a processor instruction level, reducing the cost substantially in specific cases. This is especially true if the iteration scheme is optimized with this in mind.

3.9.1 FDTD vs. FEM

When considering whether to use FDTD or Finite Element Method (FEM), the primary concern should be the physics of the system.

Simply put, FDTD requires a lot less memory. This is due to the fact that FEM sets up every point as a row in an interaction matrix, and runs a matrix multiplication scheme to find the most accurate interaction pathway. In contrast, FDTD sets up each point as its own independent entity and uses its own and the nearest neighbors' states to calculate the time-propagated state. A good way to summarize it is that FEM will almost always be memory-bound, while FDTD will be compute-bound.

Of course, FDTD has the drawback that all data is in the time domain, and must therefore be transformed with an FFT algorithm to get phase information. As discussed earlier, because of the Yee cell, FDTD has the distinct disadvantage of superfluous iterations when simulating a system where the spatial resolution and the desired wavelengths are disproportionate.

With today's computers having higher computation capabilities than effective memory bandwidth, FDTD is often a suitable alternative even at a nanoscale level, but a good go-to rule is that once you require precise resolution at the single-digit nm-scale, FEM is often the preferred alternative. A reasonable cutoff is at 5 nm. Below 5 nm FDTD becomes far too costly for most uses.

Assuming a naive implementation of the structures, the Yee grid for FDTD scales as 6 real numbers per point, and the number of grid points scales with x^3, where x is the number of points in one direction. However, due to the leap-frogging method which is numerically stable, there is only need for one copy of the data, resulting in a total memory requirement of roughly $6x^3$ in principle. The real number is of course somewhat higher, as one needs to set up special rules for PMLs and dispersive materials, but the scale is correct. In comparison, a FEM solver having the same number of points has first off 3 complex numbers, which means 3 elements per point. But, since FEM is a matrix method, this means that one has the requirement of $(3x^3)^2$ to fill the matrix. This would of course be a naive method, as one would use sparse matrices and non-uniform grids to cut down the memory requirement substantially, but the core problem still remains that the basic scaling still goes as $\mathcal{O}(x^6)$.

In summary, FDTD has the distinct advantage of having memory scaling that goes as $\mathcal{O}(x^3)$ compared to FEM which has $\mathcal{O}(x^6)$. FDTD has the disadvantage of the frequency spectrum being locked to the resolution, which can lead to either superfluous iterations, or an undesired increase in resolution being needed.

3.10 EMTL

EMTL is a free and soon to be open source FDTD solver from Kintech labs, Moscow. Due to it being free, it is worthwhile to ascertain the capabilities of the software as specialized simulation software is normally expensive. Therefore, before and during the work on the thesis, I have done some testing to check scalability and stability. The software along with tutorials and papers detailing their tests can be found at http://fdtd.kintechlab.com/. The disadvantage of EMTL is that commercially available software, such as Lumerical,[8] has a more complete user interface compared to EMTL, which requires a lot of manual post-processing.

3.10.1 Algorithms

The more advanced algorithms of EMTL are normally accompanied by a paper written by the main developers. The most relevant features are the following:

- subpixel smoothing [47]
- oblique incidence re-iteration scheme [48]
- advanced PML structures [49]

While oblique incidence was a tempting prospect for this thesis, it requires re-runs of the simulation to correct for errors that come from the fact that the Yee-cell grid is best suited for the source EM waves to travel along

[8] https://www.lumerical.com

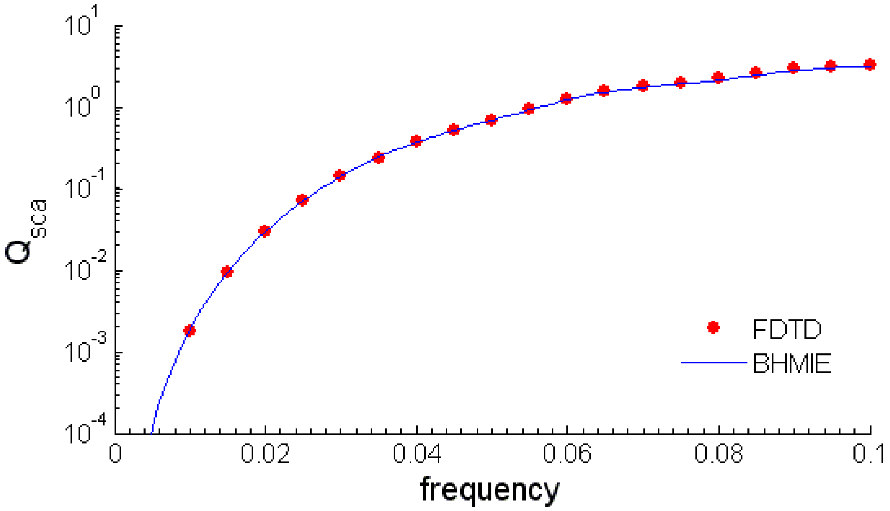


Figure 3.6: EMTL results (FDTD) for scattering intensity for Mie scattering, compared with a numerical solver (BHMIE) that uses the analytical equation for Mie scattering, taken from http://fdtd.kintechlab.com/en/tutorial.

its principal axes. The re-runs are therefore required to correct for border interactions for the source EM wave. It has therefore been deemed too costly for this thesis as the non-oblique simulation already pushes the sizes and resolutions of EMTL to the limit.

The subpixel smoothing is however relevant. It is meant to counteract the effects of poor resolution, and as is shown in Ref. [47], it succeeds to a decent degree. Considering that the resolution is close to the same order of magnitude as the structure being simulated, such an algorithm is essential for the simulation to work.

3.10.2 Accuracy

It is generally considered hard to find a good measure of accuracy for EM simulations, as the accuracy of the simulation will be heavily dependent on the complexity of the system that is simulated, and most analytically solvable problems are quite simple, estimation of absolute error is often unobtainable.

Having seen that the authors of EMTL have published results of their code corresponding with theory whenever they implement a new feature, such as in Ref. [50] and [51], doing similar tests would be redundant. As such, the primary benchmark for EMTL's numerical accuracy used in this thesis will be the capability EMTL has to reproduce the results underlying Ref. [1], which were obtained originally from the FEM solver COMSOL. As the original results were obtained with FEM, and tested in a lab, it should not have the same numerical vulnerabilities as FDTD, and is therefore the most viable criteria for whether simulating the system is within the scope of EMTL's capabilities.

Blazed gold nanogratings

4

In optics, getting accurate information from an EM wave is essential for characterizing a system's properties. One of the desired parameters is polarization. As an EM wave can be linearly, circularly or elliptically polarized, it is valuable to be able to accurately decompose an incoming wave into its constituent polarizations. Through the use of a nanograting which has periodic properties and polarization-dependent GSP interactions, Pors et al. have made a system that can redirect an incoming wave based on the Stokes vector components. This thesis' main body of work is to replicate the system from Ref. [1] in EMTL, starting with the phase diagrams of individual gold nanoblocks, and working up to simulate whole metasurfaces.

4.1 Mechanism

The purpose of the metasurface is to be able to redirect specific subcomponents of an incoming wave in specific directions. This is done by carefully arranging nanoscopic gold blocks on top of a glass plate, which in turn is on top of a gold substrate. There are two well established methods of analysing this, and later on I will propose a third method.

4.1.1 Phase-gradient pattern

The core mechanism of a directional refraction system lies in having a phase gradient. To analyze this in the time domain, the most ideal method is to consider Huygen's wavefront propagation theory.

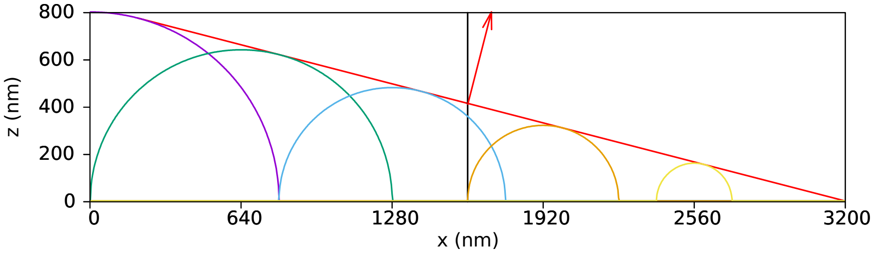


Figure 4.1: Illustration of a Huygens wavefront forming as a consequence of a phase gradient.

From this analysis, one finds that the resulting angle is

$$\theta = \operatorname{asin}\left(\frac{\lambda \cdot \Delta\tau}{D \cdot 2\pi}\right)$$

Where θ is the refraction angle, λ is the wavelength, $\Delta\tau$ is the phase difference between the start and end of the period, and D is the length of one period.

If the phase difference between the start and the end of the period is 2π, the distance of the short side should be one wavelength, which results in the following relation:

$$\theta = \operatorname{asin}\left(\frac{\lambda}{D}\right) \tag{4.1}$$

In figure 4.1 $D = 3200$ nm and $\lambda = 800$ nm, resulting in an angle $\theta = \operatorname{asin}(0.25) = 14.48^{\circ}$.

4.1.2 Repeating patterns

The secondary mechanism is one of the most commonly known mechanisms to physicists, which is the mechanism of an (approximately) infinite repeating grid. As long as there exists any net phase-altering behaviour within one period, one should be able to observe an effect at the angles $\theta = \operatorname{asin}(n \cdot \lambda/L)$. Simply put, you can treat each full period of the grating as one slit in a multiple slit experiment.

The combination of the two mechanisms has the benefit that any flaw in the phase gradient's angle naturally filters out due to the repeating pattern. In other words, even if the phase gradient turns out to be 2.1π per period instead of 2π, the angular error will only manifest as a small amplitude reduction, as the effect of having the pattern repeat itself will force any resulting refraction to go along one of the angles given by the multiple slit analogy. Of course, the more one perfects the primary mechanism, the reflected wave should get more unidirectional and sharper.

4.1.3 Fourier analysis method

The other analysis method commonly used to describe a directionally refracting system is to do Fourier analysis of the pattern. The first step is to express the phase gradient mathematically.

$$f(x) = e^{2\pi i\left(\frac{x}{D}+\frac{1}{2}\right)} \tag{4.2}$$

With f being the phase distortion at point x, and D is the length of one full period. This phase equation is periodic with a period every D along the x-axis. Because it is periodic, one can rewrite it to the following form:

$$f(x) = \sum_{n=-\infty}^{\infty} f_0(x - nD) \tag{4.3}$$

Here $f_0(x)$ is defined to only have values for one period centered on $x = 0$, i.e.:

$$f_0(x) = \begin{cases} f(x) & \text{if } |x| \leq D/2 \\ 0 & \text{if } |x| > D/2 \end{cases} \tag{4.4}$$

Now, the phase gradient pattern is expressed as a single period with displaced repetitions. Next we look at the shift theorem [52]:

$$\mathcal{F}\{g(x - x_0)\} = G(u)e^{-iux_0} \tag{4.5}$$

Combining equations (4.3) and (4.5), we end up with the following:

$$F(u) = F_0(u)\left(\sum_{n=-\infty}^{\infty} e^{-inuD}\right) \tag{4.6}$$

Since an infinitely repeating gradient is unrealistic, we restrict the sum to a discrete set of $2N+1$ repetitions. From Ref. [52] we have:

$$\sum_{n=-N}^{N} e^{-inuD} = \sum_{m=-\infty}^{\infty} (2N + 1)\,\mathrm{sinc}\,[(2N + 1)(uD/2 - m\pi)] \tag{4.7}$$

Going back to the single phase gradient, it can be found to be the following:

$$\mathcal{F}\{f_0(x)\} = D\,\mathrm{sinc}(u/2\pi - 1/D) = D\,\mathrm{sinc}(\pi/D \cdot (uD/2 - \pi) \tag{4.8}$$

If we now insert the results of equations (4.7) and (4.8) into equation (4.6), we end up with the final result:

$$F(u) = D(2N{+}1)\,\mathrm{sinc}(\pi/D{\cdot}(uD/2{-}\pi)) \sum_{m=-\infty}^{\infty} \mathrm{sinc}\,[(2N + 1)(uD/2 - m\pi)] \tag{4.9}$$

This equation has a clear maxima from the term with $m = 1$, with $u = 2\pi/D$. Using that $u = 2\pi z/D\lambda$ and $\sin\theta = z/D$, we end up with the final result for the maxima angle:

$$\sin\theta = \lambda/D \tag{4.10}$$

This is also the first maxima of Bragg's law, which makes sense as a repeating pattern should have maxima along the Bragg maxima angles.

4.1.4 The unit cell

Having gone through the ideal case of a phase gradient across a period, it is necessary to find a physical system that is able to produce the desired phase altering behaviour. One option is the rectangular gold block above a gold substrate with glass in between.

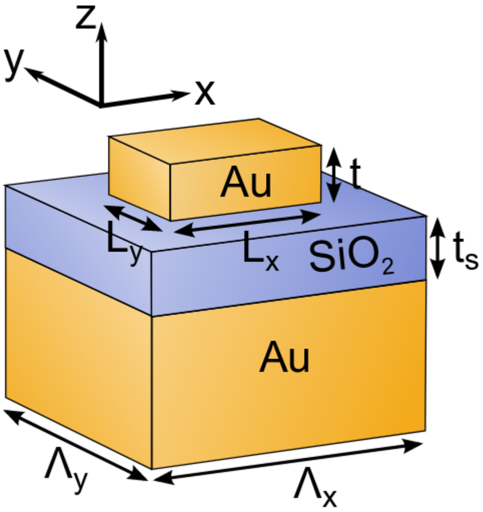


Figure 4.2: Model of a single unit cell with a gold block on top aligned with the principal directions of the unit cell. The incoming light would come from the z-direction, and travel downwards before hitting the cell. The dimensions are $t = 40$ nm, $t_s = 50$ nm, $\Lambda_x = 320$ nm, and $\Lambda_y = 250$ nm. Image was taken from Ref. [1].

The unit cell shown in figure 4.2 is the basis for all later patterns. When discussing any pattern later in the thesis, a block will refer to the gold block on top of the substrate which can have arbitrary length and width, and rotation around the z-axis. All blocks will always be centered in a unit cell, i.e. if a system consists of 10 blocks, it will consist of 10 unit cells.

This configuration allows for the gold block to have the surface currents induced by EM waves couple with the surface currents on the gold substrate, creating GSPs that act as makeshift induction loops, first absorbing and then re-emitting the EM waves, as shown in figure 4.3.

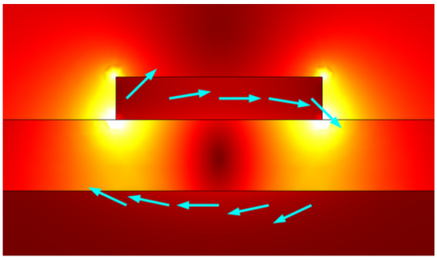


Figure 4.3: Illustration of a GSP formed by a gold block on a substrate. Brighter colors indicate a stronger electric field, and the arrows indicate the movement of the polarization currents constituting the GSP. Image was taken from Ref. [30].

Depending on the length of the block along the polarization direction, the GSP may absorb, retain and then re-emit the EM waves. The retention time, as well as absorption and emission rates, depend on the specific geometry of the gold block, as shown in figure 4.4. This can be related to quasi-static plasmon resonance theory, where an elongated sphere induces a red-shift relative to a regular sphere of the same volume [23].

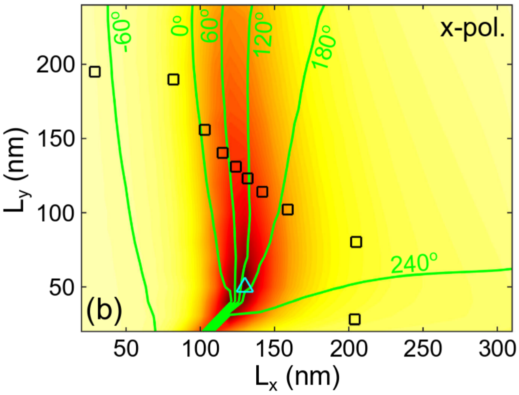


Figure 4.4: The amplitude and phase plot from Ref. [1] for a single gold block. The color scale shows reflectivity, with darker colors means less reflectivity, while the lines indicate constant phase contours.

One can also consider to decompose any incoming electrical field's effect into one component parallel to the surface of the block and one perpendicular to it. These two components will produce surface plasmons and volume plasmons respectively. For the gold box, since it is placed on top of a gold substrate with glass separating them, the surface plasmon will couple with the surface of the substrate and form a GSP.

The strength and duration of the GSP will be dependent on the specific geometry of the block. Thus, thanks to the right angles of the box, if the sides of the gold block on top of the glass substrate have different dimensions, the box will have separate GSP interactions for each electrical field component parallel to the side of the box. This allows for a box to have different phase alteration properties for two orthogonal linear polarizations.

Before going further, when discussing metasurfaces, a block refers to the gold block on top of the glass substrate, and an element refers to a unit cell. Blocks will generally be discussed in the context of being part of the structure shown in figure 4.2.

Linear polarization behaviour

Going back to the Jones vector from section 2.4.1, and the specific form of the vector in equation (2.54),

$$\begin{bmatrix} E_x(t) \\ E_y(t) \end{bmatrix} = E_0 \begin{bmatrix} \cos\psi \\ \sin\psi e^{i\Delta} \end{bmatrix} e^{i\varphi} \cdot e^{i(kz-\omega t)} \tag{4.11}$$

one can see that if two waves have the same polarization states with regards to ψ and Δ, but have a π phase shift in φ, complete extinction will appear. As such, the phase gradient that is desirable is a phase gradient in φ.

For linearly polarized light, this is quite straight-forward. The primary effect capable of delaying a wave is the formation of GSPs between the gold block and the gold substrate, and this will happen along the principal axes of the gold block.

Looking at figure 4.2, which illustrates a gold block that has its sides aligned with the axes of the unit cell, it is clear that it will have highly distinct behaviour for $|x\rangle$ and $|y\rangle$, as both of these polarization states are perfectly parallel to one side, and orthogonal to the other. Thus, to produce a system that is attuned to $|x\rangle$ and $|y\rangle$, one only needs to use blocks that are aligned with the unit cell axes. The same goes for $|a\rangle$ and $|b\rangle$, if one aligns the box to be rotated in the plane of the system at a 45 degree angle relative to the unit cell's x-axis.

Pancharatnam–Berry phase

The phase altering-behaviour turns out to also work for circular light, but in a different manner. Having a gold block with different behaviour along the two sides introduces a Pancharatnam–Berry phase, which is a geometric phase that occurs when the polarization state moves along the Poincaré sphere [53]. This can be used in the form of having a gold block with different width and length, and a rotation in the plane of the system. The geometric phase will depend on the rotation of the block, and the type of circular polarization, i.e. $|l\rangle$ and $|r\rangle$. Furthermore, if the phase difference between the two sides is π, the most optimal geometric phase is achieved [53], creating a pure directional output. This phase is cumbersome to explain mathematically, and will not be discussed as such in this thesis. Ref. [53] and [54] are good starting points for diving further into the topic with regards to metasurfaces using Pancharatnam–Berry phase related effects.

An intuitive method of understanding the geometric phase is to consider the fact that the GSP is strongest when the wave is parallel to the surface, and a circular wave will sweep over the surfaces in a specific order depending on the direction of the polarization. Thus, it is reasonable to suggest that the phase of the outgoing wave will be determined by the specific timing of when the $\boldsymbol{E}$-field vector was parallel to the surface.

The other way of describing it, which may be more mathematical, is to consider the interaction between the circular wave and the unit cell a movement along the Poincaré sphere. Any phase shift that results in the polarization state moving in such a manner as to enclose a surface area on the sphere will result in a geometric phase [53].

4.1.5 Designing birefringence

The foundational mechanism to the blazed gold nanograting is the formation of GSPs. With the structure of the gold blocks placed upon a continuous gold substrate with glass in between, the GSPs will form along the edges of the gold block, as shown in figure 4.3. Though, when designing a pattern that is supposed to redirect one polarization, e.g. $|x\rangle$, it is possible to have the pattern also have an effect on $|y\rangle$ by choosing the dimensions of each block such that $L_x(i) = L_y(M - i)$where M is the last element in a period, L_x is the length along the x-axis, and L_y is the length along the y-axis. This results in the system appearing mirrored, giving rise to birefringence for the output.

4.2 The Pors metagrating

As has been established, to get a bidirectional grating, a spatial periodicity with a corresponding phase gradient has to be designed. The Pors et al. grating [1] is relatively straight-forward as it targets one element of the

Stokes vector at a time. The three designs are called MS1, MS2 and MS3, with MS being an acronym for Metasurface. All patterns use the same unit cell dimensions shown in figure 4.2, Λ_x = 320 nm and Λ_y = 250 nm. The specific sizes of the blocks can be found in the enclosed code in the appendix.

MS1

MS1 is the metasurface attuned to $|x\rangle$ and $|y\rangle$, and is constructed by having all constituent blocks be aligned to the x and y axis, and have different length and width. In one end, the length starts long and gets shorter as you move along the period, and in the other end, the width starts long and gets shorter as one moves towards the start of the period. This pattern consists of 10 cells, and has therefore a length of $D = 10\Lambda_x = 3.2$ µm, resulting in an output angle of 14.5°.

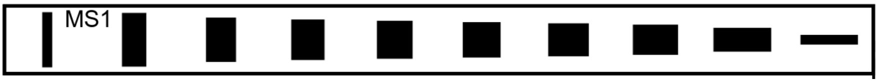


Figure 4.5: Illustration of MS1 taken from Ref. [1].

MS2

This grating follows the same principle as MS1, but with the blocks set at 45 degrees relative to the x axis. This gives them maximal GSP behaviour for $|a\rangle$ and $|b\rangle$ polarized light instead. This pattern consists of 8 cells, and has therefore a length of D = 2.56 µm, resulting in an output angle of 18.2°.

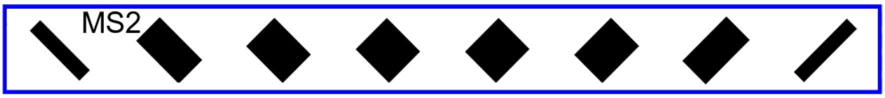


Figure 4.6: Illustration of MS2 taken from Ref. [1].

MS3

The MS3 grating, however, is based on a different mechanism, namely the alteration of the Pancharatnam–Berry phase. The Pancharatnam–Berry phase is dependent on the specific angle of the block. As such, this grating has the same basic block in all cells, but has each block rotated gradually along the period for a total rotation of π per period. This block has the specification of 130 × 60 nm. This pattern consists of 12 cells, and has therefore a length of $D = 3.84$ µm, resulting in an output angle of 12.0°.

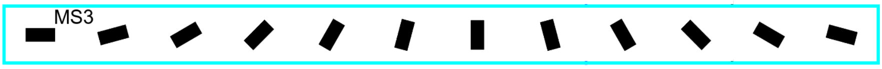


Figure 4.7: Illustration of MS3 taken from Ref. [1].

Full Grid

The full grid becomes a surface where all three patterns are present. This metasurface has two potential uses. The first is to classify the polarization state of incoming light. The second is to be an efficient method of generating every single polarization state from unpolarized monochromatic light.

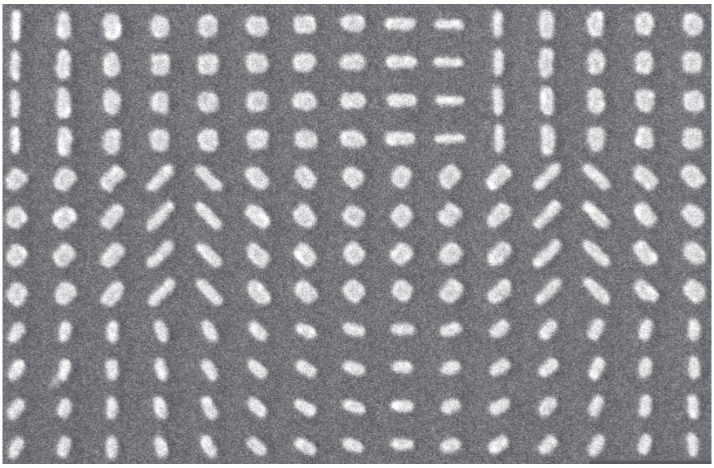


Figure 4.8: A scanning electron microscope image of a physical metasurface from Ref. [1]. As can be seen, 4 rows of the same type are clustered together in the y-direction. This image is only an excerpt of the larger system.

4.3 Alternative method of analyzing mechanics

A different method of conceptualizing the mechanism of the Pors grating, is to consider only one constituent block per repeating row at a time. Ignoring the phase properties of all other blocks, one can then consider each block unique to a period (e.g. if the pattern has a period of 8 blocks, then blocks 1, 9, 17 ...) as emitters with a synchronized phase, at a constant internal distance. In this analysis, instead of summing N repeating rows of M elements each with a directional emission, one sums a total of M sets of N equally spaced emittors.

4.3.1 Derivation of the diffraction condition

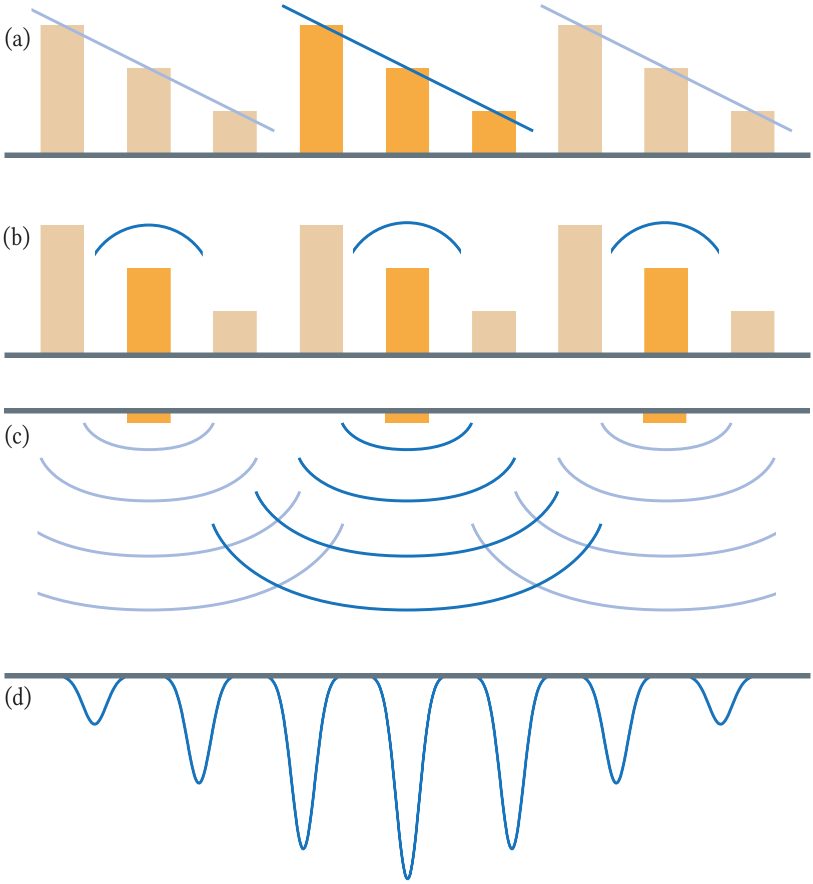


Figure 4.9: Illustration of the alternative analysis method. Fig. a) is the conventional method of analysis, while Fig. b) is the alternative method. Fig. c) is to illustrate the analogy of a multiple diffraction system, and Fig. d) is the typical diffraction pattern that arises from a finite repeating grid.

We start off by setting up the well-known diffraction pattern of multiple slits. The maxima will appear at angles that satisfy the following relation, known as Bragg's law [32]:

$$D \sin \theta = n\lambda \tag{4.12}$$

Here, D is the distance between the same type of emitters, i.e. the period length, λ is the wavelength, and θ is the diffracted angle. Since we're interested in the angle, we rewrite it to the following form:

$$\sin\theta = \frac{n\lambda}{D} \tag{4.13}$$

However, for the second block that gets analysed, two differences arise. The first is the difference in phase due to the blocks' GSPs having different retention times, and the second is the path difference between the blocks to the different maxima. Looking at the path difference one finds:

$$d\sin\theta = \frac{d \cdot n\lambda}{D} \qquad \Longrightarrow \qquad \beta = \frac{d}{D}n2\pi \tag{4.14}$$

Here d is the distance from the first block type analyzed and β is the resulting phase delay in radians from that displacement, resulting in a total phase difference of

$$\Delta\alpha_n = \tau + \frac{d}{D}n2\pi, \qquad n \in [-3, -2, -1, 0, 1, 2, 3], \tag{4.15}$$

where τ is the phase difference between the two block types. As shown in equation (4.15), if constructive interference is desired at the first maxima, the phase difference has to be

$$\Delta\alpha_1 = 0 = \tau + \frac{d}{D}2\pi, \qquad \Longrightarrow \qquad \tau = -\frac{d}{D}2\pi \tag{4.16}$$

This allows for another rewrite of equation (4.15):

$$\Delta\alpha_n = \frac{d}{D}2\pi(n-1), \qquad n \in [-3, -2, -1, 0, 1, 2, 3] \tag{4.17}$$

The next step is to ensure destructive interference at the other maxima. The simplest method of doing this is to take the cosine of each periodic block's contribution at one maxima at a time, and add them all together. Assuming all components satisfy equation (4.16), and using that cosine is an even function, one can additionally simplify equation (4.17) by defining a new variable $b = |n-1|$.

$$\sum_{i=0}^{M-1} \cos\left(d_i \frac{2\pi b}{D}\right) = 0, \qquad b \in \{1, 2, 3\} \tag{4.18}$$

Where i is the i'th block in the line, and M is the total number of blocks. I have elected to limit b to 3, as the metagrating systems in this thesis reflect at angles such that one does not even have to bother with second maxima.

If one further imposes a restriction of even spacing between the blocks, M equates to D/d, where d is the constant spacing. This simplifies the extinction condition further:

$$\sum_{i=0}^{M-1} \cos\left(i\frac{2\pi b}{M}\right) = 0, \qquad b \in \{1, 2, 3\} \tag{4.19}$$

Equations (4.18) and (4.19) have defined a metagrating functionality condition that is solely based on diffraction patterns and phase properties.

If one wants to relate this derivation to the Fourier formalism, equation (4.14) is equivalent to the shift theorem in Fourier analysis [52]:

$$\mathcal{F}\{g(x - x_0)\} = G(u)e^{-iux_0} \tag{4.20}$$

Here u is the Fourier variable, and x_0 is the displacement that in equation (4.14) is called d. One can see that the e^{-iux_0} term induces a phase shift that will vary with u, which gives rise to the possibility of aligning the phase shift of $G(u)$ with the displacement-induced shift.

The interesting thing about this analysis is that Pors et al. assumed that contiguousness was required for the grating to work, while this equation shows that as long as one can view the blocks as independent emittors when stimulated, no such condition is necessary.

There is of course the coherence concern, if the phase of the incoming wave starts differing drastically as one moves along the grating. It is conceivable that the contiguousness condition stated by Pors is a coherence condition instead. Equations (4.18) and (4.19) assume that the incoming light is coherent to a distance long enough that one can approximate the interference pattern to be similar to that of an infinitely repeating grid. It is possible to introduce a coherence correction at the earlier stages of the derivation, but it will severely diminish the simplicity of the equations, which defeats the purpose of this derivation.

4.3.2 Examples of using the diffraction condition

Having established the condition, the simplest method of visualizing it is to set up the results in tables, and see how the phases line up at the correct maxima.

Table 4.1: Phase table for a 3-block system. Here d is distance between block 0 and block n, D is the length of a period, τ is phase difference between block n and 0. Finally, -2, -1 ... 2 are the diffraction maxima.

Block #	d/D	τ	-2	-1	0	1	2
0	0	0	4π	2π	0	2π	4π
1	$1/3$	$-2\pi/3$	$4\pi - 6\pi/3$	$2\pi - 4\pi/3$	$0 - 2\pi/3$	2π	$4\pi + 2\pi/3$
2	$2/3$	$-4\pi/3$	$4\pi - 12\pi/3$	$2\pi - 8\pi/3$	$0 - 4\pi/3$	2π	$4\pi + 4\pi/3$
SUM			const.	ext.	ext.	const.	ext.

Table 4.2: Phase table for a 6-block system. Here d is distance between block 0 and block n, D is the length of a period, τ is phase difference between block 0 and n. Finally, -2, -1 ... 2 are the diffraction maxima.

Block #	d/D	τ	-2	-1	0	1	2
0	0	0	4π	$6\pi/3$	0	2π	$12\pi/3$
1	$1/6$	$-\pi/3$	3π	$4\pi/3$	$-\pi/3$	2π	$13\pi/3$
2	$2/6$	$-2\pi/3$	2π	$2\pi/3$	$-2\pi/3$	2π	$14\pi/3$
3	$3/6$	$-3\pi/3$	π	$0\pi/3$	$-3\pi/3$	2π	$15\pi/3$
4	$4/6$	$-4\pi/3$	0	$-2\pi/3$	$-4\pi/3$	2π	$16\pi/3$
5	$5/6$	$-5\pi/3$	-1π	$-4\pi/3$	$-5\pi/3$	2π	$17\pi/3$
SUM			ext.	ext.	ext.	const.	ext.

These results are also predictions to verify the method numerically, as one can see that the 3 block system has a peak at the -2 maxima.

4.3.3 Sparse lines

An interesting aspect of this method of analysis is that a row does not have to be filled to be analyzed. If one treats empty unit cells as "zero-blocks" with a phase τ_0, one can set up the same tables as before.

Table 4.3: Phase table for an evenly spaced 3-block system in a 6 unit cell long grid. d is distance between block 0 and block n, D is the length of a period, τ is phase difference between block n and 0. -2, -1 … 2 are the diffraction maxima. B_n signifies a filled position, Z_n a vacant position with a emission phase of τ_0 relative to B_0.

Block #	d/D	τ	-2	-1	0	1	2
B_0	0	0	4π	2π	0	2π	4π
B_1	2/6	$-2\pi/3$	2π	$2\pi/3$	$-2\pi/3$	2π	$14\pi/3$
B_2	4/6	$-4\pi/3$	0	$-2\pi/3$	$-4\pi/3$	2π	$16\pi/3$
Sum B			const.	ext.	ext.	const.	ext.
Zero #	d/D	τ	-2	-1	0	1	2
Z_0	1/6	τ_0	$10\pi/3 + \tau_0$	$5\pi/3 + \tau_0$	$0 + \tau_0$	$7\pi/3 + \tau_0$	$14\pi/3 + \tau_0$
Z_1	3/6	τ_0	$6\pi/3 + \tau_0$	$3\pi/3 + \tau_0$	$0 + \tau_0$	$9\pi/3 + \tau_0$	$18\pi/3 + \tau_0$
Z_2	5/6	τ_0	$2\pi/3 + \tau_0$	$\pi/3 + \tau_0$	$0 + \tau_0$	$11\pi/3 + \tau_0$	$22\pi/3 + \tau_0$
Sum Z			ext.	ext.	cont.	ext.	ext.

The interesting thing to note about table 4.3 is that the zero elements extinguish each other at all other maxima than the 0th, and the behaviour is symmetrical, as they all have the same phase. This is the general rule except for a 4 cell wide grid, i.e. 2 normal blocks and 2 zero blocks, where the second maxima will have constructive interference for the zero blocks.

The results of table 4.3 are also the basis for predictions to verify the method numerically, as one can see that the 3 block sparse system has a peak at the -2 maxima. This coupled with the aforementioned property that as long as it's periodic, the zero blocks will not contribute to any maxima other than the 0th.

4.3.4 Expanding the method

While this analysis is useful as it is, it still has room for improvement. Primarily, there are three possible additions that will improve its utility. The first is to expand the method to give a full two-angle output spectrum per repeating block type. The second is to incorporate coherence width into the calculation. The third improvement is to include reflection amplitude to the calculation. All of these are possible to implement and have no foreseeable conflicts with each other.

However, in expanding the method, one will naturally lose the simplicity of the current single angle, maxima only analysis that was shown in the tables earlier. As such, it is also a question of whether it is more beneficial to switch over to Fourier space when handling the patterns, in which case the only difference between the standard analysis is the idea of looking at each constituent block, instead of each period. This said, we should evaluate the pros and cons of this potential expansion of the method:

- Pro: Will treat any periodically appearing block type as one element computationally, and therefore be highly efficient.
- Pro: Will be efficient at incorporating loss from coherence width.
- Con: All blocks have to appear periodically.
- Con: Ignores secondary plasmon behaviour.

This raises the question of whether this method has the potential to create something that has any advantage of an analytically derived pattern. While this method is capable of analyzing patterns efficiently, the most probable usage of it would be to do randomized generating of patterns and analysing them to see if one gets sufficiently beneficial output patterns from it. But, since all secondary plasmonic behaviours are neglected, it is probable that it is incapable of generating any designs that are superior to human design.

4.4 Secondary plasmonic behaviour

The largest flaw when designing metagratings is that one has to account for unwanted plasmons. The two most obvious types are the GSPs formed with the nearest neighbors, which is an especially big factor for the gold blocks that have their principal axes aligned with the grid (MS1), and local surface plasmon resonances where the blocks momentarily have dipole attraction between them, changing the duration of which the plasmon-induced charge stays at the edge.

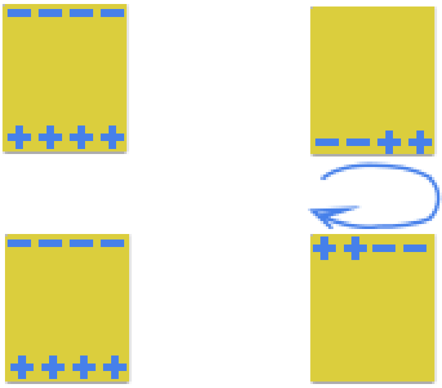


Figure 4.10: An illustration detailing how using a single gold block with periodic boundaries will get either coupled dipole interaction with the neighboring block (left pair), or a GSP between them (right pair).

When faced with a design problem, there are normally two approaches available. One can either take the problem into account and include the effects into the design, or one can attempt to remove, or at least minimize, the problem.

One of the problems when designing the metasurface is that the phase shift properties are calculated for a single gold block type at a time. Furthermore, it is calculated with periodic boundary conditions, which means that all neighboring blocks are identical, causing the phase information to include the effects of bordering identical neighbors.

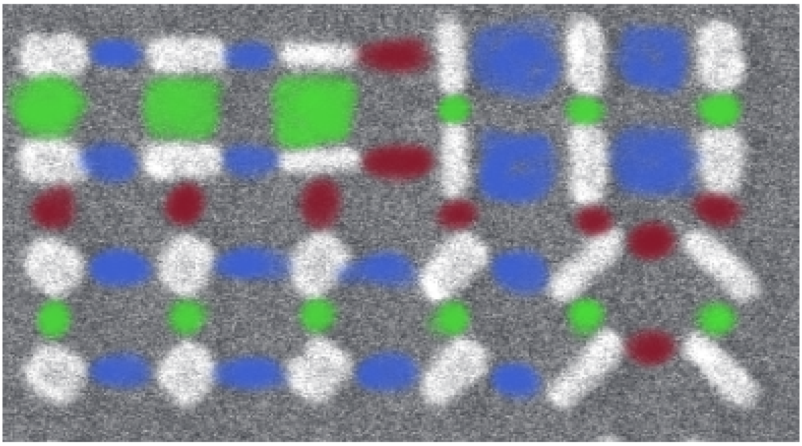


Figure 4.11: An illustration detailing how using a single gold block with periodic boundaries will result in incorrect compensation for secondary plasmons. This figure is an excerpt of figure 4.8, and has colors drawn on it to signify if a secondary plasmon is correctly compensated (green color), almost correctly compensated (blue), or wrongly compensated (red).

As can be seen in figure 4.11, the original design struggles with secondary plasmons being accounted for at varying degrees of accuracy. The blue colored areas appear because the elements vary along the x-axis, but the differences in dimension are not too large for the approximation of the effect being similar to that found by simulating a single box with periodic boundaries. This is of course a secondary and quite small effect when compared to the primary GSP behaviour, but it is interesting to study it as perfecting the design can be relevant in later designs.

It should also be considered that there are indications that Pors et al. did not simulate the blocks at a 45° angle, as the original $|a\rangle$ and $|b\rangle$ aligned pattern, MS2, had asymmetrical dimensions. This appears to be a mistake in directly applying results for $|x\rangle$ and $|y\rangle$ polarized light to the block configuration for $|a\rangle$ and $|b\rangle$ birefringent metasurface, MS2, without considering that the difference in lengths of the unit cell will not affect it. If this is the case, then only the MS1 pattern's neighbor interactions would be correctly labelled in figure 4.11, and all other patterns would suffer from this mistake.

4.4.1 Analyzing the plasmonic behavior

The first step is to isolate the primary GSP effect. If one is capable of getting highly accurate measurements of the primary GSP, one can accurately predict the contribution of the primary GSP, and ascertain that all other effects are from the secondary plasmons. The second step is naturally to use the primary GSP data to create an "ideal" data set for a system that is likely to experience secondary plasmons, and compare the ideal with the actual data.

Isolating the primary GSP

One possible method of doing this in a simulation would be to set up a simulation environment large enough to prevent secondary plasmonic behaviour to occur. By my estimation, increasing the size of the unit cell by three to five times while keeping the block size constant should be enough to remove the possibility of secondary plasmons to form. Then, one sets up a very high resolution nearfield detector mesh just above and around the gold box, create sub-domains at locations where one expects distinct plasmonic behaviour, such as above the box, along the sides, and in the corners. These sub-domains should start off small at locations where the effect is the strongest, and then be expanded until the sub-domain encapsulates the entire GSP. Done right, this should create a region map for collecting the different phase contributions, allowing for precise analysis and design. This would be the most accurate analysis as it could produce an accurate model of the primary GSP, and with an accurate model, one need only to run simulations of the full systems and compare it with the ideal scenario of only primary GSP contributions to quantify the effects of secondary plasmonic behaviour on the system. The downside is that this would be a complicated endeavor which require complicated numerical post-processing of the data.

The other approach I would propose is a brute-force method. Rather than getting accurate near-field data, one sticks with the total phase of the reflection. But, if one varies the size of the unit cell while keeping the block size constant, it should be possible to use the reflection phase of an empty cell, i.e. a cell with only glass and gold- substrate, and use regression analysis to isolate the contribution of the gold block and its plasmonic effects. This approach, if done right, should be able to isolate the primary GSP effect before the bordering cells get close enough for secondary plasmonic behaviour to occur.

Isolating secondary plasmons

Now, ideally, if one is able to do the first method for isolating the primary GSP, one could set up a similar system to find secondary plasmons. But, since this would be much the same type of approach as the previous step, it is more worthwhile to look at the second method's continuation.

After obtaining the primary GSP data for gold blocks, it should now be feasible to construct patterns and create an "ideal" reflection pattern. The simplest and perhaps most effective pattern would be a 2 × 2 supercell, with element 1,1 and 2,2 being the same, and 1,2 and 2,1 being the same, with periodic boundary conditions. This cell would have the advantage of each block being surrounded by the other type on all sides. This should be a well-suited environment for differences between ideal and actual behaviours to arise, and to be analyzed.

However, both of these methods would require a large amount of time to be done right, which may outweigh the benefit. Furthermore, this analysis is meant to improve potential designs, but if their accuracy far outweighs the physical production accuracy, having pinpoint precision in theory will not yield any significantly better results in the laboratory. So, it is therefore the recommendation of this thesis to carefully consider the cost and benefit of carrying out this analysis to the end. It might be that to incorporate production flaws, coherence width, and other factors relevant to the physical system into the analysis will yield far more worthwhile results than to get a complete understanding of the secondary plasmonic behaviour.

4.4.2 Reducing the prevalence of secondary plasmons

The second approach to unwanted plasmons is to design with the intention of minimizing the effect of unwanted plasmonic behaviour.

To obtain behaviour closer to an ideal phase retarder one would need the accurate GSP analysis discussed earlier. On the other hand, since the most probable error source is local surface plasmon resonances between the different blocks, it is possible to use the fact that these are dependent on the specific block types neighboring each other to our advantage. This leads to the idea of interleaved gratings.

Interleaved gratings

It was while simulating this system early on, that the idea of interleaving the patterns arose. The first idea was just to get the functionality of both patterns into one single line, which could reduce the size of the system. But, upon further investigation, there was a secondary benefit as well.

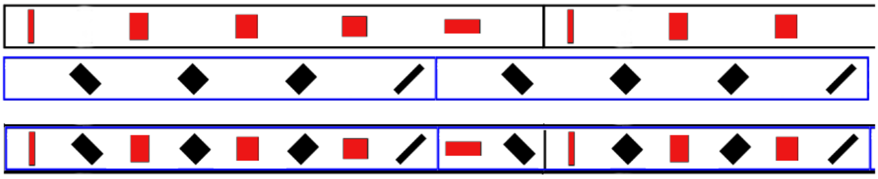


Figure 4.12: The top row with the red blocks is an MS1 pattern with every even numbered block removed, and the second row is an MS2 pattern with every odd numbered block removed. The third row is then the resulting MS1-MS2 interleave.

Looking at the arrangement in figure 4.12, one will naturally reduce the GSP formation between the elements in the x-direction, as the elements will always be angled in relation to each other. Further, if one alternates which row's elements start first, i.e. MS1 gets element 1, 3, 5 etc. in the first row, and 2, 4, 6 in the second, there should be no adjacent surfaces that are parallel, which should eliminate the GSP interactions almost completely. In other words, even if one chooses not to use an interleaving pattern to free space or compress the system, it would still be advantageous to interleave MS1 and MS2 merely for the benefit of removing unwanted GSPs between nearest neighbors.

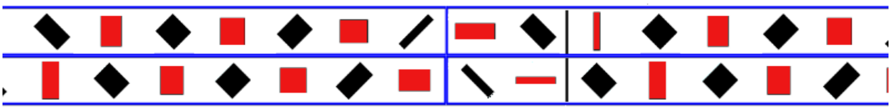


Figure 4.13: Illustration of an MS1-MS2 alternating interleave. The main difference is that this interleave alternates along the *y*-direction with regards to whether each pattern has even or odd positions. This results in a pattern that uses all the original blocks of the original patterns.

As can be seen, for the MS1-2 alternating interleave, there are no parallel surfaces between neighbors. This should severely reduce the influence of unwanted secondary GSPs, improving symmetry for the output. This also has low vulnerability from spatial incoherence, as the two types are uniformly spread across the surface. This is advantageous as it allows for the system to operate on optical systems with coherence widths far lower than the original system. This is a key demand in further miniaturisation of the metasurface, as well as lowering the system demands for interfacing with the metasurface effectively. However, the most significant quality of the alternating interleave is that the occurrence of two specific block types being neighbors will only happen at the overlap of the patterns' periods. This means that a pattern that has 10 blocks, and one that has 8, will only have identical occurrences every 40 blocks. In other words, any arbitrary combination of blocks that turned out to have some resonance of some kind, can only occur every 40'th block. Thus, by using the alternating interleave pattern, secondary plasmonic behaviour is made more chaotic, which should increase the probability of it only generating white noise instead of a focused emission.

Another design possibility would be to use the result from section 4.3.3, that the gratings work even if there are gaps in the pattern, and use that to make the MS3 pattern go along the y-axis instead. If done this way, the metasurface would have no discernible centerpoint, and can therefore handle spatial incoherence far better.

A final idea, if one wants complete azimuthal-angle separation of the reflections would be to have one of the interleaved patterns have a periodicity along the x-axis, and the other along the diagonal. This design would give full spatial separation of the reflected waves, which could hold potential benefits for a system design that wishes to incorporate a metasurface of this type.

Even if splitting the outputted reflections in the azimuthal angle is undesirable, the alternating interleave of MS1 and MS2 will open up the possibility for further miniaturization, which is always a positive.

5 Simulation results

This chapter will present all the numerically obtained results in simulating the Pors et al. metasurface from Ref. [1]. The goal was to reproduce the results of the paper, and then investigate the viability of the theoretical predictions made in the previous chapter.

The first step performed was to simulate a metasurface type from the Pors et al. metasurface paper [1], namely MS1, and see if the desired outcome would be produced. By simulating a full MS1 line the feasibility of simulating the metasurface effects was determined. This simulation was done by approximating the sizes based on Ref. [1] at a grid resolution of 10 nm. The output of the preliminary simulation corresponded sufficiently with the predicted angles of refraction for 800 nm light. Thus, thorough recreation of the Pors papers could begin.

The goal is to recreate the paper's results from the bottom up, but some slight alterations will be made to facilitate the testing of the alternating interleave. Therefore, this chapter will consist of the following, all done numerically:

- Recreate the phase information for a single gold box.
- Recreate each of the single polarization lines, but with symmetrical values.
- Recreate the full Pors system.
- Test the sparse metagrating predictions from section 4.3
- Test the alternating interleave predictions from section 4.4.2

The symmetrical values mentioned in the second point mean that for each pattern, the lengths of the gold boxes follow the relation $L_x(i) = L_y(M - i)$, i.e. that the first and last block of a pattern's period have the exact same dimensions, just with flipped x and y axis length, and similarly for the second and second last, etc. This was done in order to have the interleave results be independent on the accuracy of the compensation done towards secondary plasmonic behaviour, and have it be directly comparable to the other results.

5.1 Single gold box

The first step after confirming that EMTL was capable of simulating the macroscopic effects was to recreate the reflection phase diagram, shown earlier in figure 4.4, for a single gold block. This phase shift data is the basis of all the patterns later on.

5.1.1 Resolution comparison

Having confirmed the ability of EMTL to simulate the mechanics for the grating reflecting the first Stokes element at 10 nm, the next step was to investigate the behaviour of the system with regards to resolution. This was done by taking a single gold box, with dimensions 200 × 100 nm, rotated to an angle of 30 degrees, and stimulated by an $|a\rangle$ polarized Berenger pulse at normal incidence. The reflections were then by a planar detector set, and output in the frequency domain, which will from here on be called a "flux spectrum". The spectra were normalized to the amplitude of an unaltered Berenger pulse identical to the one used to stimulate the system. These results are only to uncover vulnerabilities related to resolution, and have no value beyond that.

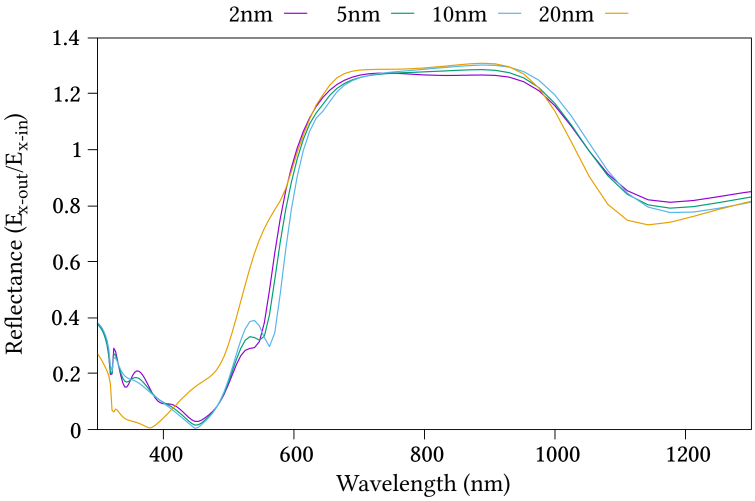


Figure 5.1: Normalized flux spectrum for E_x for a single gold box at certain resolutions.

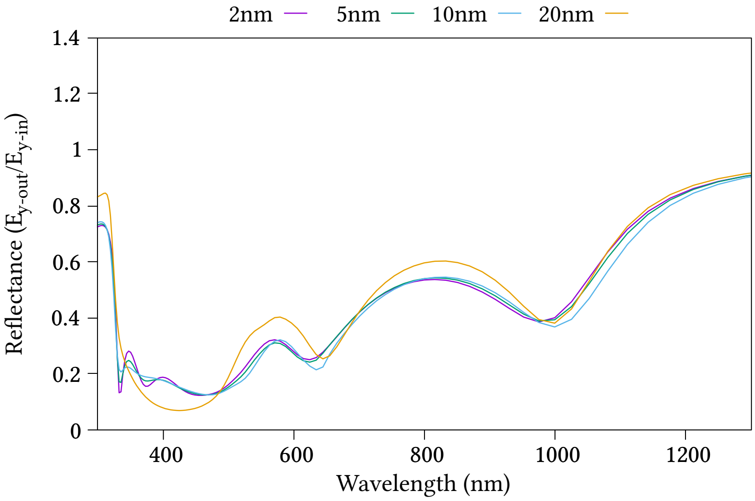


Figure 5.2: Normalized flux spectrum for E_y for a single gold box at certain resolutions.

Unsurprisingly, the results for 20 nm resolution were poor, showing that the resolution got past the resolution breaking point, i.e. it produced qualitatively different results than the others.

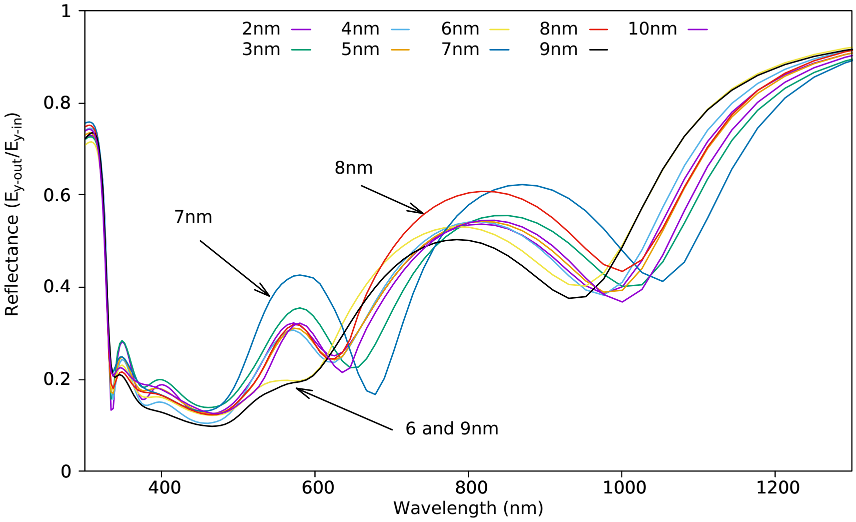


Figure 5.3: Normalized flux spectrum for E_y for a single gold box at varying resolutions at and below 10 nm. The clustered lines are the ones for 2, 4, and 5 nm resolutions, with 10 nm being the other resolution closest to matching it.

Looking closer at the resolutions at and below 10 nm, see figure 5.3, the most interesting result is that the resolutions of 9, 7, and 6 nm were the most deviant from the rest, even having qualitative differences in the case of 7 nm. Considering that the specifications of the gold box were at 200 × 100 nm, one can see that the resolutions with poor performance are inherently prime compared to the dimensions of the box. 3 nm is prime as well, but deviates to a lesser degree than the others due to it being a far finer resolution. It is also interesting that the 8 nm result had a correct shape, but too high amplitude around 800 nm. It is also worth considering that the simulation environment had dimensions of 320 × 250 nm, which is also relatively prime to the resolutions that performed poorly. Essentially, it could be that having a mismatch between resolution and simulation environment creates undesired effects on the simulation, in which case the coherence of the 2, 4, 5 and 10 nm simulations are due to the absence of grid error.

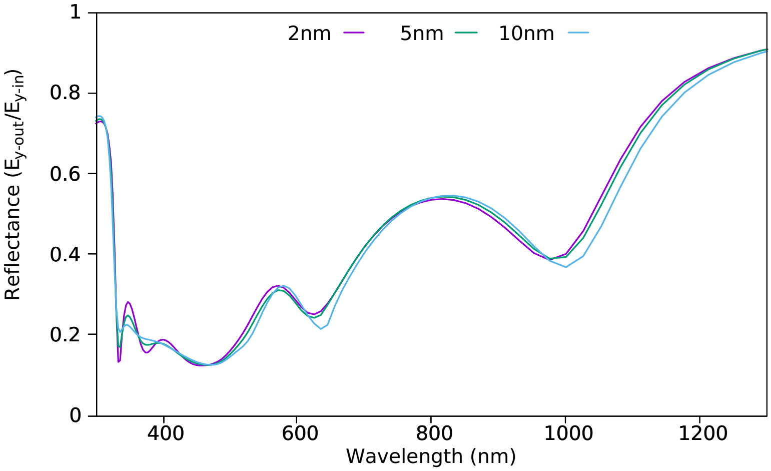


Figure 5.4: Normalized flux spectrum lines for E_y of the most suitable resolutions for the simulation.

The unfortunate result of the resolution testing showed that the errors are somewhat arbitrary in nature, especially if one considers the 3 nm case, which had more differences with 2, 4 and 5 nm than 10 nm did, despite being a far higher resolution. However, considering that the biggest predictor of the qualitative difference was whether or not the resolution was relatively prime with regards to the specifications of the unit cell and gold block, it seems reasonable to proceed under the assumption of both 5 and 10 nm being suitable resolutions, bearing in mind that it is highly likely that some inaccuracy will appear at these resolutions. Due to the combination of the somewhat arbitrary nature of the differences between the resolutions, and the fact that the thesis will only focus on one wavelength, 800 nm, quantifying the error is problematic. If comparing only at 800 nm, some of the deviant resolutions will get good results, but taking the whole spectrum into account may be as wrong. Furthermore, the certainty of 2 nm being a "more accurate" simulation is not good either.

In the end, the resolutions of 5 nm and 10 nm were selected for the simulations moving forward, as there's a factor 2 between them, both behaved in a similar fashion, and are the least costly resolutions available in runtime and memory of the tested resolutions.

5.1.2 Wave type selection criteria

A final consideration is the generated wave and the corresponding time used. Since the FFT algorithm is essentially a curve-fitting algorithm, a pulse will have intrinsic inaccuracy related to the resolution. Using a continuous sine wave, one can feed the FFT a signal that is completely filled, which will allow the fitting algorithm to pinpoint the phase to a higher degree than otherwise. This was confirmed when comparing the results of running the FFT on the first 400 time steps recorded, and contrasting it to the FFT of time steps 100 to 500. In the former, frequencies corresponding to wavelengths other than 800 nm had an amplitude of 3% or less relative to the value for 800 nm. For the delayed FFT, all other wavelengths had amplitudes of about 10^{-6} relative to 800 nm component. Thus, to minimize the error, a monochromatic sine wave with a delayed recording was employed rather than using a Berenger pulse to sweep all frequencies.

But, since Berenger pulse simulations require far less computation time, they will also be included in the simulations later to have a basis of comparison for simulations where circular polarization is irrelevant. Essentially, the metasurface's specialization for specific polarizations, especially circular polarization, will also work to highlight any vulnerability of a Berenger pulse simulation has towards chirality in the system.

The Gaussian sine pulse could have been used as well, but one could not assume that the circular polarization state of a Gaussian sine would produce correct results outside of the main frequency of the pulse. However, since the later simulations use the built-in far-field detector set which runs an FFT on the whole data set before projecting each frequency to far-field, obtaining a high-resolution frequency spectrum at frequencies close to 800 nm would require a long simulation time, reducing the possible benefit of a Gaussian sine pulse. This is due to the frequency resolution scaling linearly with the length of the data set.

5.1.3 Neglecting evanescence

In the simulation parameters I have chosen lengths at which evanescent waves have not necessarily decayed yet. This is due to the cost of increasing the simulation size. This does carry some risk, but the preliminary results, as well as the results later on indicated that it could be neglected. This does of course mean that some error could arise from this, but the increased computational cost would outweigh the benefit of eliminating evanescent waves.

5.1.4 Results

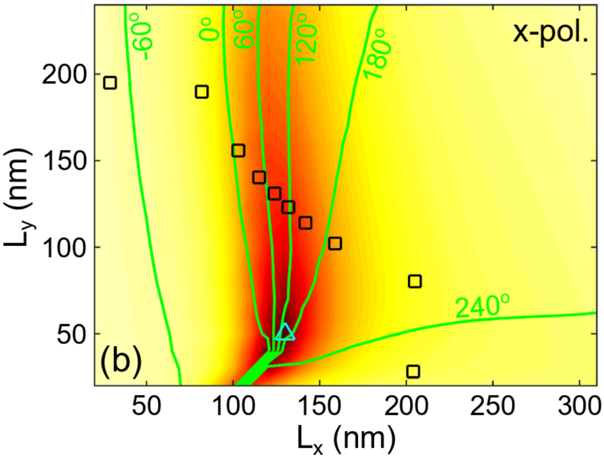


Figure 5.5: The amplitude and phase plot from Ref. [1] for a single gold block. The color scale shows reflectivity, with darker colors means less reflectivity, while the lines indicate constant phase contours. The blocks are the positions of the MS1 pattern.

The intention is to re-create the phase plot from Ref. [1] in order to either verify the results or to correct them. The simulation environment used was a 320 × 250 nm unit cell, with gold blocks varying in length and width from 20 nm to 250 nm, simulated at 5 nm resolution with PML above and below the system, and periodic boundaries to the sides.

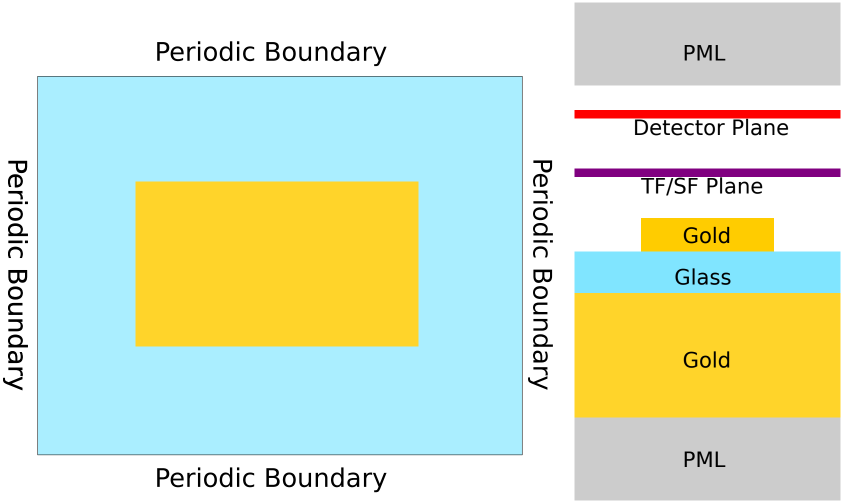


Figure 5.6: Illustrations detailing the simulation setup. The left image is seen from above, showing the central placement of the block and the periodic boundary along the borders in x and y direction, while the right image shows the structure from the side with PML boundaries encapsulating the environment.

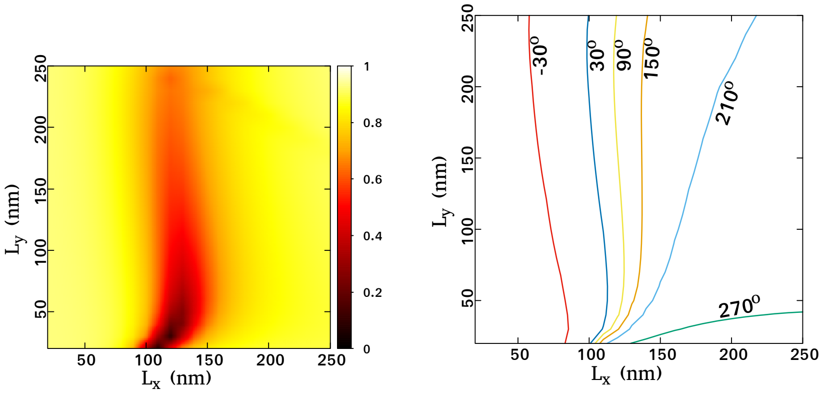


Figure 5.7: The amplitude (left) and phase plot (right) from my own simulations for a single gold block.

The results for the single gold box were positive. Figure 5.7 matches figure 5.5 very well. The amplitude of the reflected wave matches the data from Pors, and the phase gradient matches Pors as well. However, the exact value of the phase lines, differed a bit, indicating that there is a constant phase difference between the two plots.

Discrepancy with Pors

The most important result of this simulation is the phase gradient, as it is what drives the primary mechanism of the metagrating. And, as can be seen in section 5.1.4, this is in perfect agreement with the results by Pors et al. In other words, the single-cell simulations in EMTL with 5 nm resolution have, very accurately, reproduced the results of Ref. [1].

There is however a discrepancy in the exact values of the phase. The phase has here been calculated as follows:

- Set up a plane of detectors above the region generating the waves that travel downwards towards the system.
- Run a k-space decomposition on the plane for all time steps, extract the k_0-vector-value for each step.
- Run an FFT on the k_0 values from $t = 100$ to $t = 500$ to calculate a frequency spectrum with phase. Use the result for the frequency corresponding to 800 nm.
- Subtract the pure unaltered wave's phase at the plane on top of the system (i.e. where the gold blocks start).
- Subtract the phase shift from the difference in path.

This produced a different phase plot than that of Pors. As mentioned, the gradients are almost perfectly matched, and I was able to produce a near exact copy of the phase plot lines by changing the path compensation used. My hypothesis is that since Pors' system is only dependent on the gradient, the phase constant was neglected. This will give rise to problems should one wish to combine different designs where the other phase plot does not have this error in the constant.

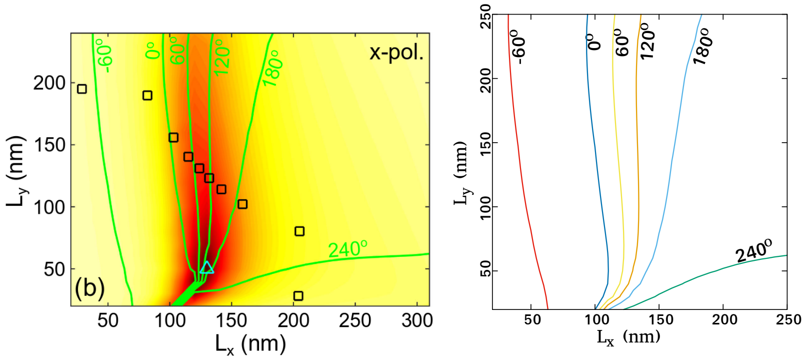


Figure 5.8: The Pors phase plot (left) and my own adjusted phase plot (right) from my own simulations for a single gold block.

As can be seen from figure 5.8, the two plots have almost identical behaviour on the phase lines. The correction was as follows:

$$\begin{aligned} \text{My path compensation equation} \qquad & \beta = \alpha + \frac{2\pi}{800\ \text{nm}} \cdot 120\ \text{nm} \\ \text{Added phase to fit with Pors' results:} \qquad & \Delta\beta = \frac{2\pi}{800\ \text{nm}} \cdot (-40\ \text{nm}) \end{aligned}$$

Here α is the phase measured at the top of the gold blocks from a non-interacting source wave. Incidentally, 40 nm is also the height of a gold block. This suggests that an error has happened in correcting the path. The most likely candidate is that the unaltered wave was measured at the glass substrate top surface, instead of the gold block top surface resulting in a small phase shift in the constant.

5.2 Single metasurface rows

Having established the traits of the unit cell, and confirmed them to be almost identical to the original paper, I moved on to simulate full patterns.

As this step was the last where it was viable to simulate at a resolution of 5 nm, additional simulations at 10 nm were also performed to give some measure of the difference in accuracy between the two resolutions on a complete grating. In order to project the results into far-field, a detector plane of sufficient size was specified. Since the least common multiple of the grating lengths, 8, 10, 12, is 120, and the full Pors system thus requires 120 × 3 unit cells, I used that as the basis for the near-to-farfield detector setup.[1] So, for the single rows (e.g. 12 × 1), 120 × 3 was used, for the systems later in the paper that are simulated at 10 nm resolution and at a dimension of 120 × Y, the near to farfield detector plane was just fitted to the simulation environment.

A further thing of note is to consider the coherence width of a physical system. In section 2.5 the coherence width was established, and if one inserts the specifications of the grid, with the wavelength being 2.5 times the unit cell length, one sees that the coherence width can be expressed as number of cells in the x-direction as follows:

$$\begin{aligned} l &= R \cdot 2.5 \cdot \Lambda_x \\ \text{using that } R = 50 \implies l &= 125\Lambda_x \end{aligned}$$

This shows that the farfield detector setup is coincidentally almost the same as the coherence width. In other words, by coincidence, the far field results will to some degree account for coherence width.

All simulations were run with periodic boundary conditions along x and y borders, and PML for the z-directional border, corresponding to the setup in figure 5.6. The patterns always advance in the x-direction, and the incident light always propagate with a k-vector direction of $-\hat{z}$

This section will first verify the function of the individual grating patterns, and then do an L^2 analysis using trapezoid integration of the intensity, comparing short low- and high-resolution systems.

As was shown in section 5.1.1, the numerical error is somewhat unpredictable in these simulations. Due to this fact, the choice of significant digits when

[1] This turned out to be the limit for what I could simulate on my computer, so I could not simulate the Pors system with 12 rows as shown in figure 4.8.

displaying results became contentious. The choice I landed on was to display as many digits as needed for the comparison of all results to each other. Essentially, because some results were down to the 5th decimal point, I elected to display 6 decimal points for all data sets, despite the accuracy probably not being this good. It is therefore important to keep in mind that all the results in the tables from here on out should be viewed mostly in a qualitative manner.

Finally, as the specifics of the different surfaces was established in another chapter, I will list them here as well.

- MS1: $D = 10\Lambda_x$, reflects $|x\rangle$ and $|y\rangle$ polarized light at angles ±14.5.
- MS2: $D = 8\Lambda_x$, reflects $|a\rangle$ and $|b\rangle$ polarized light at angles ±18.2.
- MS3: $D = 12\Lambda_x$, reflects $|r\rangle$ and $|l\rangle$ polarized light at angles ±12.0.

5.2.1 Visual comparison of resolutions

As discussed in section 3.8.2, one of the first comparison methods one should employ is a visual comparison. As can be seen in figure 5.9, other than the zero-angle reflection, there is almost perfect concordance between the two results. For MS2 and 3, the zero reflection had good concordance as well, and are therefore not shown as plots as they would be redundant.

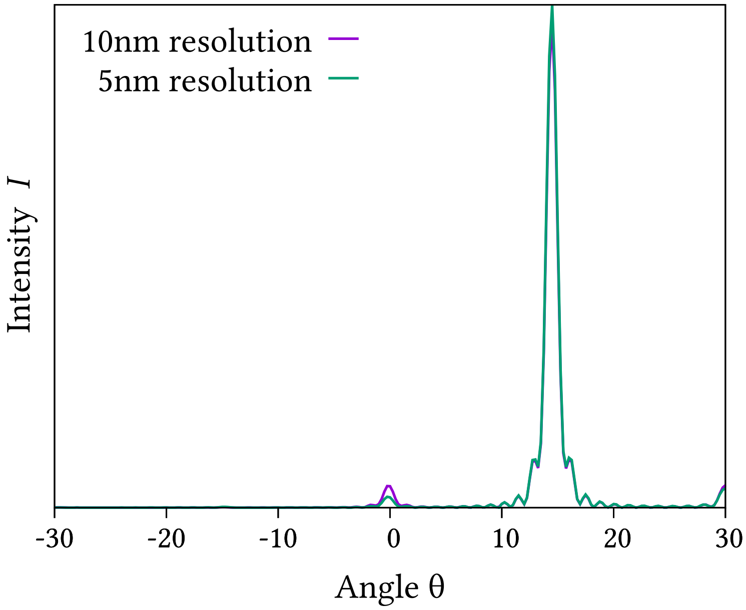


Figure 5.9: Intensity plot of the MS1 grating for $|x\rangle$ at high and low resolution. As can be seen,the only discernible difference is at the 0-angle reflection, which is a minor difference.

Figure 5.9 shows that the two resolutions produce nearly identical output. For the MS2 and MS3 patterns, even the 0-angle reflection was identical as well, so those plots have been left out.

5.2.2 High resolution grating results

Visual display

Before looking at the numbers, it is useful to do a quick visual comparison of the responses of each grating type. In figure 5.10 it is clear that each single grating type has close to identical output profiles, and are relatively symmetrical with regards to amplitude.

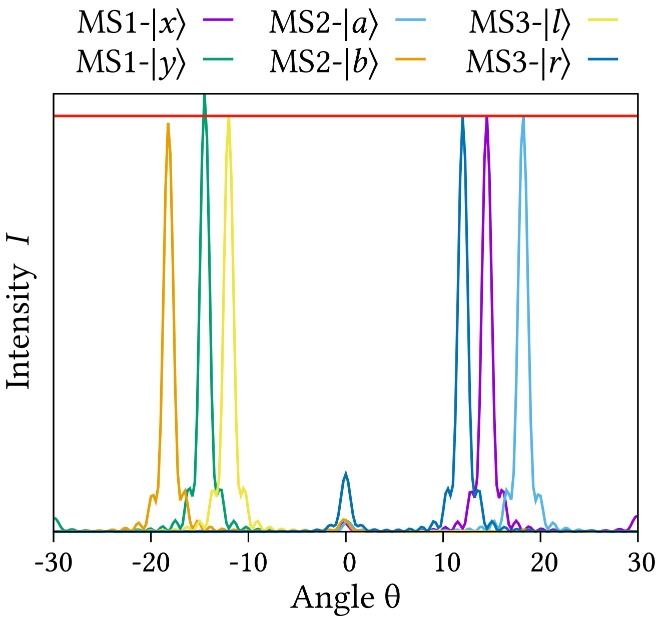


Figure 5.10: Overlapping intensity plots for all grating types' responses to their designated polarizations. All of them have been normalized to the peak intensity of the polarization that outputs in the positive angle direction.

Heat maps

The second way of visualizing the intensity that will be used in this thesis are heat maps. The benefit to this visualization method is that all polarization states of the incoming light can be shown in a single collection of heat maps. The heat maps of this subsection are not particularly interesting, but are kept as it is useful before discussing the combined patterns to see the behaviour of each individual pattern and how it is only birefringent for the intended orthogonal polarization states.

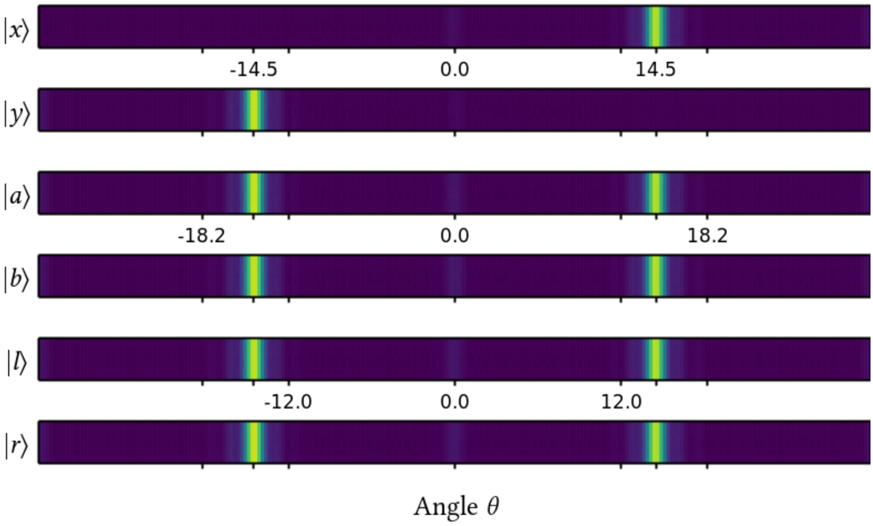


Figure 5.11: Intensity heat maps for all polarization states used on an MS1 grating.

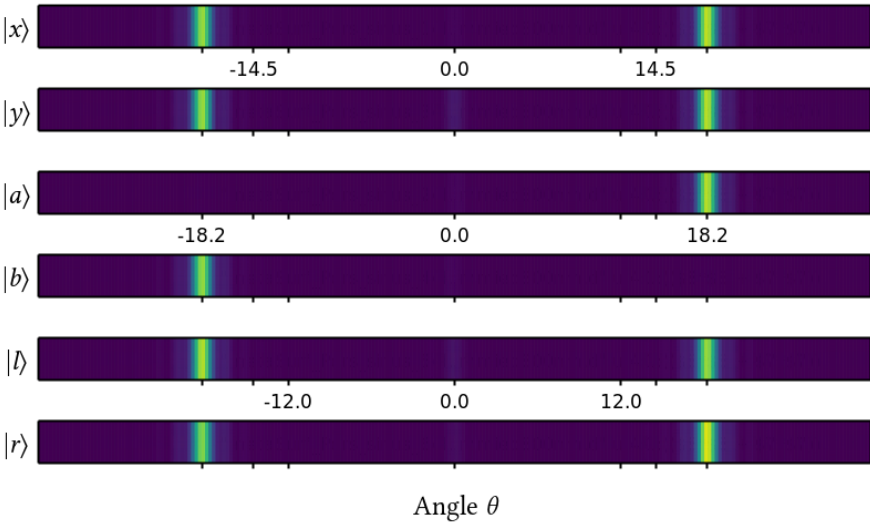


Figure 5.12: Intensity heat maps for all polarization states used on an MS2 grating.

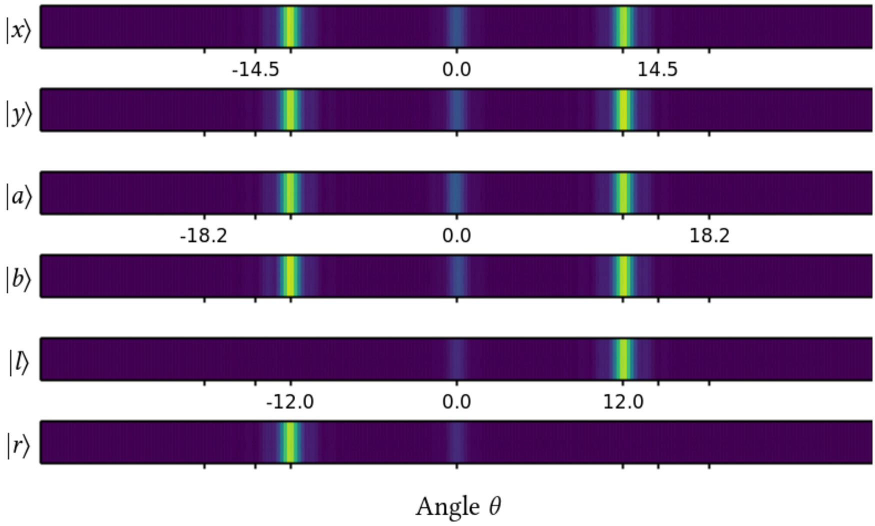


Figure 5.13: Intensity heat maps for all polarization states used on an MS3 grating.

Stokes vector output

As a Stokes vector has one intensity element, and 3 polarization elements, it would be overkill to write out the full Stokes vector for every output angle. Therefore, for brevity, only the relevant polarization elements of the Stokes vector will be displayed in the tables. These values are of course taken from the outputted angle closest to the reflection angle.

Table 5.1: Verbose Stokes parameters of output angles for the MS1 pattern simulated at high resolution. The values highlighted in bold are the values that are used in the simplified tables.

MS1	$I(14.5^{\circ})/I_{max} \times Q(14.5^{\circ})$	$I(-14.5^{\circ})/I_{max} \times Q(-14.5^{\circ})$
$\|x\rangle$	1.0 × **0.999984**	0.001012 × 0.991558
$\|y\rangle$	0.000814 × −0.991712	1.0 × −**0.999993**

While there is some information in the Stokes vector of the reflection of the angle opposite to the intended output angle, it would in most cases be excessive relative to the information that is desired. Therefore, a simplified table will be used in most cases, unless there's something worth mentioning in the data set other than the correlating Stokes elements.

Table 5.2: Simplified table of Stokes parameters of output angles for all single row patterns simulated at high resolution. The values highlighted in bold are the same values as in table 5.1. The results for $U(\pm 18.25^{\circ})$ are for the MS2 pattern, and likewise $V(\pm 12.0^{\circ})$ are for MS3.

$Q(14.5^{\circ})$	$Q(-14.5^{\circ})$	$U(18.25^{\circ})$	$U(-18.25^{\circ})$	$V(12.0^{\circ})$	$V(-12.0^{\circ})$
0.999984	**−0.999993**	0.999841	−0.999925	−0.999729	0.999216

As can be seen, when simulating the rows individually, they are able to reflect the incoming waves without altering the polarization state significantly.

5.2.3 Resolution error estimation

In this section, the error estimation of the single row simulations will be presented. What is done is for each polarization are comparisons between low and high resolution, and continuous sine wave and Berenger pulse. These are represented as lower and upper case, and s and b. Example: s-b is the comparison between low resolution (10 nm) sine and Berenger simulations, while S-B would be the same but with high resolution (5 nm). This analysis quantifies both the viability of the Berenger pulse analysis, and it quantifies the difference in aliasing sensitivity between the two modes.

As Berenger pulses have circular polarization disabled in EMTL,[2] only sine comparisons are available for $|l\rangle$ and $|r\rangle$.

[2] This is probably due to the Berenger pulse normally being used as a wide-frequency sweep, i.e. its frequency is set based on the simulation resolution as seen in figure 3.5.

Table 5.3: $1 - L^2$ values for MS1 grating.

MS1	$\|x\rangle$	$\|y\rangle$	$\|a\rangle$	$\|b\rangle$	$\|l\rangle$	$\|r\rangle$
s-S	0.000890	0.000574	0.000655	0.001369	0.000993	0.000983
b-B	0.002355	0.001830	0.002234	0.003216	N/A	N/A
b-s	0.002243	0.001848	0.002131	0.002164	N/A	N/A
B-S	0.002019	0.000752	0.001345	0.001221	N/A	N/A

Table 5.4: $1 - L^2$ values for MS2 grating

MS2	$\|x\rangle$	$\|y\rangle$	$\|a\rangle$	$\|b\rangle$	$\|l\rangle$	$\|r\rangle$
s-S	0.005591	0.005691	0.000025	0.000021	0.000340	0.000188
b-B	0.001110	0.001176	0.000176	0.000387	N/A	N/A
b-s	0.001135	0.000891	0.000083	0.001062	N/A	N/A
B-S	0.005625	0.004165	0.000537	0.000112	N/A	N/A

Table 5.5: $1 - L^2$ values for MS3 grating

MS3	$\|x\rangle$	$\|y\rangle$	$\|a\rangle$	$\|b\rangle$	$\|l\rangle$	$\|r\rangle$
s-S	0.008932	0.001166	0.001781	0.000624	0.000069	0.000085
b-B	0.003829	0.003689	0.004217	0.002255	N/A	N/A
b-s	0.111932	0.110463	0.103252	0.119072	N/A	N/A
B-S	0.079472	0.087593	0.088929	0.079889	N/A	N/A

Table 5.5 shows quite clearly that when analyzing chiral elements, the Berenger pulse deviates significantly from the sine wave.

5.2.4 L2 symmetry analysis

While the symmetry values have some qualitative bearings for the behaviour of the system, it is also worthwhile to see them in relation to the resolution and signal generation, to get a measure of how trustworthy the results are with regards to the simulation being able to correctly simulate all EM interactions.

Table 5.6: $1 - L^2$ symmetry values for MS1 grating.

MS1	$\lvert x\rangle$ vs. $\lvert y\rangle$	$\lvert a\rangle$ vs. $\lvert b\rangle$	$\lvert l\rangle$ vs. $\lvert r\rangle$
S	0.000039	0.001363	0.001306
s	0.000613	0.000664	0.000544
B	0.001000	0.002929	N/A
b	0.003137	0.002681	N/A

Table 5.7: $1 - L^2$ symmetry values for MS2 grating.

MS2	$\lvert x\rangle$ vs. $\lvert y\rangle$	$\lvert a\rangle$ vs. $\lvert b\rangle$	$\lvert l\rangle$ vs. $\lvert r\rangle$
S	0.012426	0.000051	0.000198
s	0.004880	0.000014	0.001997
B	0.000989	0.000044	N/A
b	0.006854	0.000669	N/A

Table 5.8: $1 - L^2$ symmetry values for MS3 grating.

MS3	$\lvert x\rangle$ vs. $\lvert y\rangle$	$\lvert a\rangle$ vs. $\lvert b\rangle$	$\lvert l\rangle$ vs. $\lvert r\rangle$
S	0.006485	0.000135	0.000005
s	0.000930	0.002635	0.000003
B	0.001024	0.000112	N/A
b	0.001384	0.000711	N/A

5.2.5 Remarks

The primary result is that the simulation using EMTL was successful.

As shown by the results and L^2 values, the error between high- and low-resolution is low enough to be considered negligible with most results for sine waves having values at third digit or below.

There is one result in particular that is noteworthy, and that is the Berenger pulse results for the MS3 grating in table 5.5. These results strongly indicate that the pulse is poorly suited to interact with configurations that are tuned to interact with circularly polarized light. Thus, it can be concluded that moving forward, sine waves should be used as they are more reliable.

Furthermore it is noteworthy how it is particularly $|x\rangle$ and $|y\rangle$ that has the highest asymmetry values for both MS2 and MS3 regardless of the method of generating the wave. This corresponds well with the hypothesis that the gold blocks get a dipole charge as it is affected by the incoming wave, and as the there is a length difference in the unit cell, the dipoles that are generated will couple with differing strength. Essentially, the dipoles made from $|x\rangle$ polarized light have greater distance between the blocks than the ones made from $|y\rangle$ polarized light.

5.3 Full Pors metasurface

Finally, the time comes to simulate the full system. Because of the size of the system, and the varying lengths of the individual rows, a very large simulation has to take place. Therefore, all simulations have been at 10 nm resolution.

In order to get all the metagrating elements to have the correct periodicity, a grid of 120 × 3 unit cells were simulated. This means that instead of the clustering of 4 rows at a time as in the original design, this simulation has only one row of each type. This will create some differences, but should still produce viable results.

5.3.1 Results

The desired results were obtained. Clear peaks at 12°, 14.5° and 18.25° shows that the system works as predicted by simpler theory.

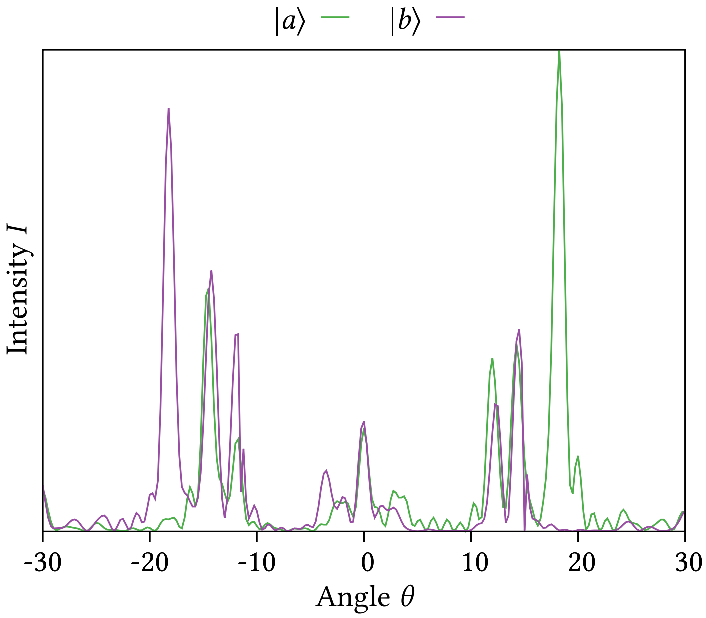


Figure 5.14: Intensity plot of $|a\rangle$ and $|b\rangle$ polarized light being reflected off the full Pors Metagrating

As can be seen from figure 5.14, there is quite a bit of asymmetry in the results. So, while there was near complete extinction where it should be, there is also the unfortunate result that the asymmetry imposes the requirement of calibration and post-processing in order to get accurate reads. This can be a limiting factor if one were to optimize the system for mass production. If there is good symmetry, it opens up for the possibility of making an analogue variant.

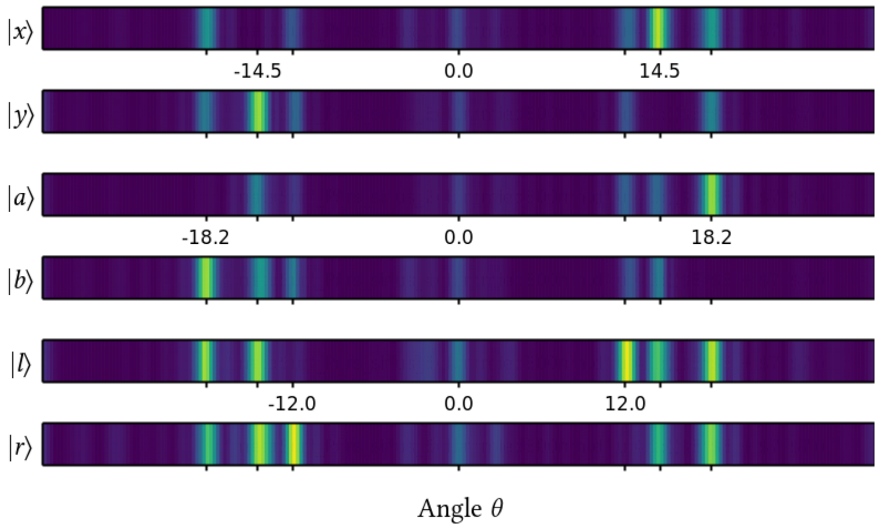


Figure 5.15: Intensity heatmap with regards to angle of all modes of polarized light being reflected off the full metagrating.

Table 5.9: Simplified table of Stokes parameters of output angles for a full Pors system simulation.

$Q(14.5^{\circ})$	$Q(-14.5^{\circ})$	$U(18.25^{\circ})$	$U(-18.25^{\circ})$	$V(12.0^{\circ})$	$V(-12.0^{\circ})$
0.998994	−0.960945	0.987713	−0.990696	−0.931817	0.905774

Table 5.10: $1 - L^2$ symmetry values for the full Pors metagrating

$|x\rangle$ vs. $|y\rangle$	$|a\rangle$ vs. $|b\rangle$	$|l\rangle$ vs. $|r\rangle$
0.024905	0.035851	0.047729

This shows that the Pors metagrating system is theoretically and computationally sound, and behaves as shown in their paper.

One thing to note however, which also affects the later systems, is that the MS3 pattern's contribution is less efficient than the others. Looking at figure 5.15, the MS1 and MS2 output angles have clearly higher intensity than all other patterns when affected by the intended polarization. The MS3 pattern's angles however output at an intensity comparable to the intensity of the of the other patterns' output when hit by polarizations unintended for them. Moreover, the symmetry values in table 5.10 showed much higher asymmetry than the results for the single pattern simulations. This shows that if you are to use this metagrating, it needs a proper calibration method which records the intensity of each pure polarization state and uses that as the fitting curves for any arbitrary wave that is to be measured.

5.4 Sparse metasurface

As mentioned in 4.4, having a full understanding of the metagrating mechanics is essential to produce new variations. Thus, the alternative analysis method proposed in section 4.3 needs to be tested numerically.

In order to verify the validity of equation (4.19), it is necessary to simulate the systems described in section 4.3.2. Because 3 and 6 blocks were the optimal testing candidates, the third pattern, which has a periodicity of 12 blocks was deemed the most suitable. Furthermore, it is the only grating where all blocks share the same dimensions and instead only vary by angle. This is relevant as the other blocks have varying degrees of reflectivity as seen in figure 5.7. Although the reflectivity difference has an effect no matter the number of grating elements, it is naturally more important the fewer blocks that are in play. It is also interesting as the MS3 system operates on a somewhat different principle than the others. So, having a test to see if the Pancharatnam–Berry phase will give constructive and destructive interference as if it was a normal phase shift will be interesting. This will be especially true for the three block pattern as it will have a separation distance of 4 unit cells, i.e. $320 \times 4 = 1280$ nm which is more than a wavelength between each block.

5.4.1 3-block sparse pattern

Table 4.3 predicted that a three block pattern should give constructive interference at the +1 and -2 interference maxima. For the 12-block system, the angles in question are 12.0° and 24.6°.

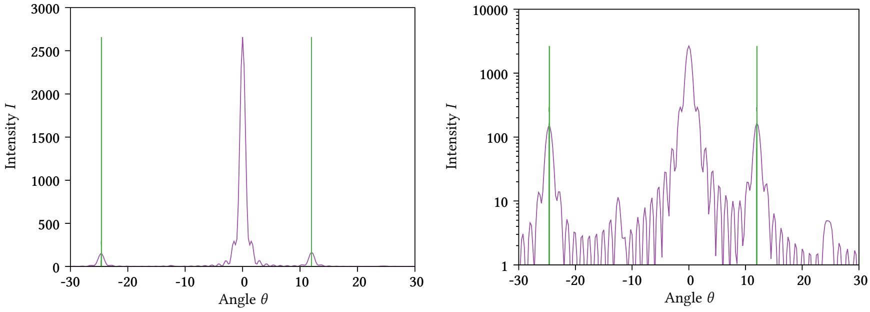


Figure 5.16: Intensity plot and logplot with regards to angle for 800 nm circular polarized light hitting a 3 elements per row sparse MS3 pattern with lines drawn at predicted maxima.

5.4.2 6-block sparse pattern

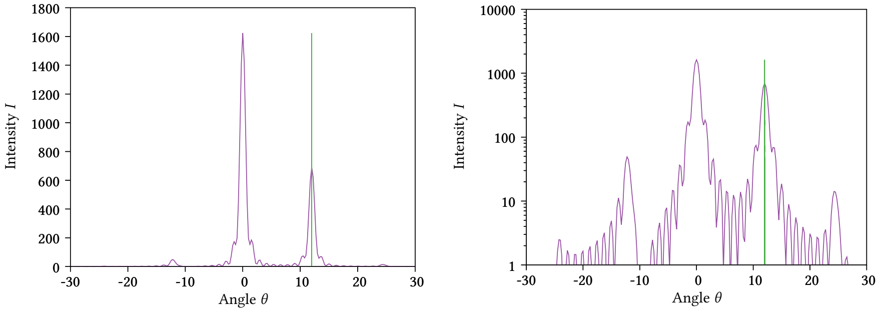


Figure 5.17: Intensity plot and log-plot with regards to angle for 800 nm circular polarized light hitting a 6 elements per row sparse MS3 pattern with lines drawn at predicted maxima.

Though there is a weak response at both -1 and +2, these responses are likely due to resolution error or nearest-neighbor interactions. With this consideration done, these barely visible responses are considered negligible compared to the main result.

5.4.3 Remarks

The prediction was correct for evenly spaced blocks. This proves the viability of the alternative analysis done in section 4.3. Further, the success of the three block system shows that the Pancharatnam–Berry phase operates in a similar fashion to regular phase in relation to adding up the phases, also when distances are above one wavelength. These results indicate therefore that it is safe to say that sub-wavelength separation is not a concern according to the EMTL simulations done here. The concern of the patterns having to have sub-wavelength separation seems to me to have come from the original classification of metamaterials having sub-wavelength properties that give a macroscopic material property. The error in interpretation comes from ”sub-wavelength” referring to the component size and not the pattern size, a distinction that appears to be lost to some.

5.5 Interleaved lines

Because of the near to far projection problem, extra long systems are desirable to reduce the error. Thus, every interleaved pattern got simulated at a combined length of 120 elements in the x-direction. On the positive side, this makes the results directly comparable to the full Pors metagrating results, as the potential error from the near to far projection should be equal.

5.5.1 Simple interleave vs alternating interleave

In this section, we look at whether the idea discussed with alternating the MS1 and MS2 pattern to reduce unwanted nearest neighbor GSPs.

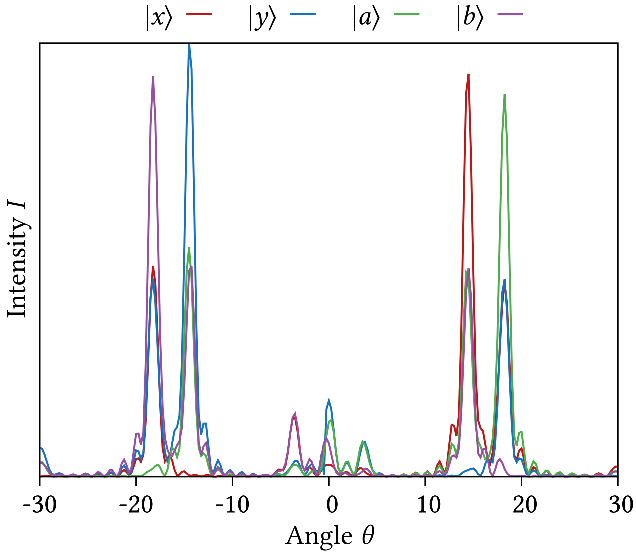


Figure 5.18: Intensity plot for MS1 and MS2 together without any interleaving as a function of angle for $|x\rangle, |y\rangle, |a\rangle,$ and $|b\rangle$ incoming waves.

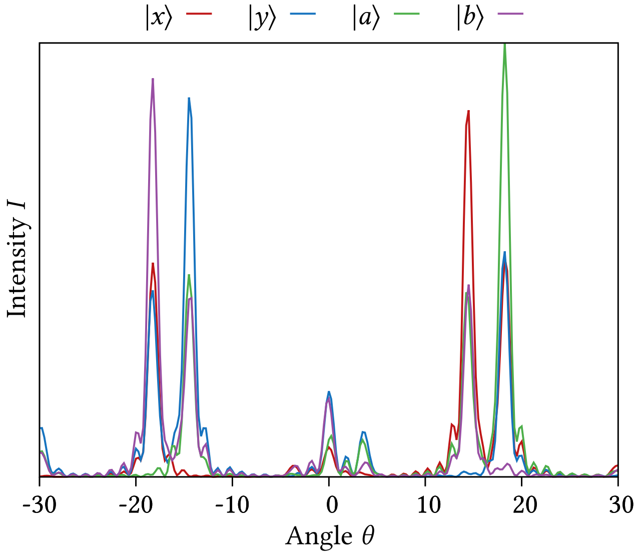


Figure 5.19: Intensity plot for a simple interleave of MS1 and MS2 as a function of angle for $|x\rangle, |y\rangle, |a\rangle,$ and $|b\rangle$ incoming waves.

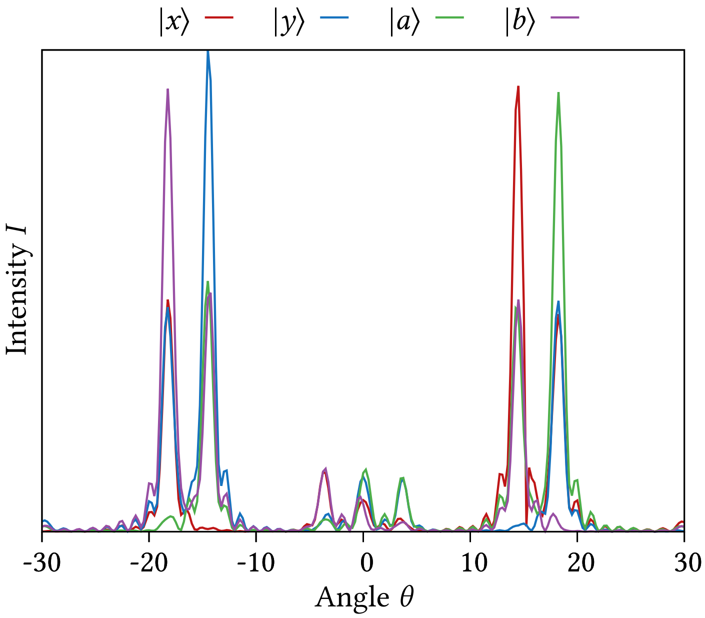


Figure 5.20: Intensity plot for an alternating MS1-MS2 interleave with regards to angle for $|x\rangle, |y\rangle, |a\rangle,$ and $|b\rangle$ incoming waves.

Table 5.11: $1 - L^2$ symmetry values for MS1 + MS2, MS1-2 interleave and MS1-2 alternating interleave.

Type	$\lvert x\rangle$ vs. $\lvert y\rangle$	$\lvert a\rangle$ vs. $\lvert b\rangle$	$\lvert l\rangle$ vs. $\lvert r\rangle$
No interleave	0.028113	0.008728	0.026415
Simple Int.	0.030500	0.018408	0.033391
Alternating Int.	0.006036	0.005873	0.021570

As can be seen, for the $|x\rangle$ vs. $|y\rangle$ and $|a\rangle$ vs. $|b\rangle$ especially, the symmetry values for the alternating interleave improved significantly. This is a clear indication that the hypothesis was correct, and the alternating pattern is able to reduce the effect of secondary plasmons. This indicates that the removal of directly parallel surfaces, and/or the chaotization of dipole interactions has led to a more symmetrical system.

Another interesting thing about figure 5.20 are the small peaks in the middle. These peaks lie close to the angle 3.6° which is the first maxima of a pattern with a period length of $D = 40 \cdot 320$ nm which is the common period length of the two interleaved patterns.

The simple interleave, however, did poorly. The symmetry value for $|x\rangle$ vs. $|y\rangle$ is understandable as the blocks align along the y-axis, which means that one has traded unwanted interactions along the x-axis for unwanted interactions along the y-axis. The simple interleave performed worse than the non-interleave on all fronts. Since the x-axis of the unit cell is longer than the y-axis, then it stands to reason that the coupling effect is stronger along the y-axis, resulting in more disturbances to the output.

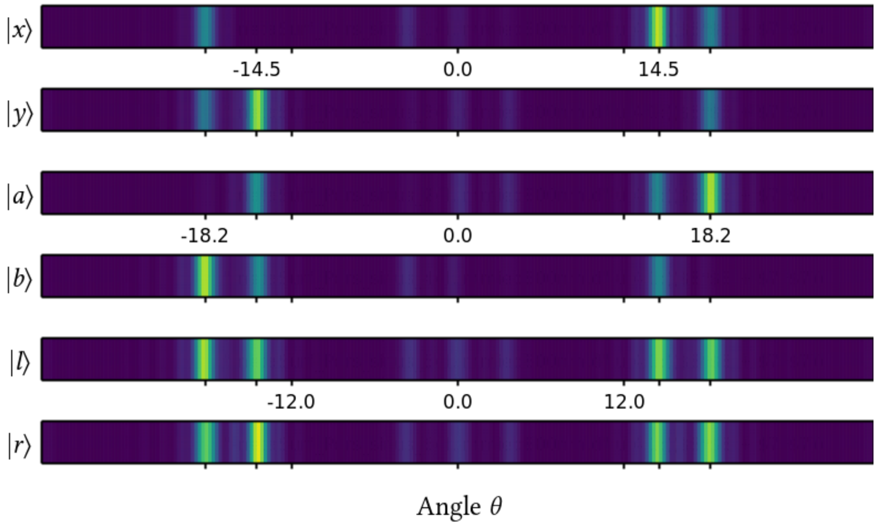


Figure 5.21: Intensity heatmap with regards to angle of all modes of polarized light being reflected off the MS1-2 alternating interleave

Table 5.12: Simplified table of Stokes parameters for varying setups using the MS1 and MS2 patterns.

MS1& 2	$Q(14.5^{\circ})$	$Q(-14.5^{\circ})$	$U(18.25^{\circ})$	$U(-18.25^{\circ})$
No Interleave	0.999999	−0.995814	0.999296	−0.999740
Simple Interleave	0.999604	−0.994806	0.999042	−0.998910
Alternating Interleave	0.998932	−0.995190	0.999962	−0.999316

As can be seen, there are no significant differences in the output with regards to Stokes vector elements.

5.5.2 Alternating interleaves with MS3 pattern

Having established the advantage of the alternating interleave, it is interesting to see how the MS3 pattern interacts as well.

MS1-3 alternating interleave

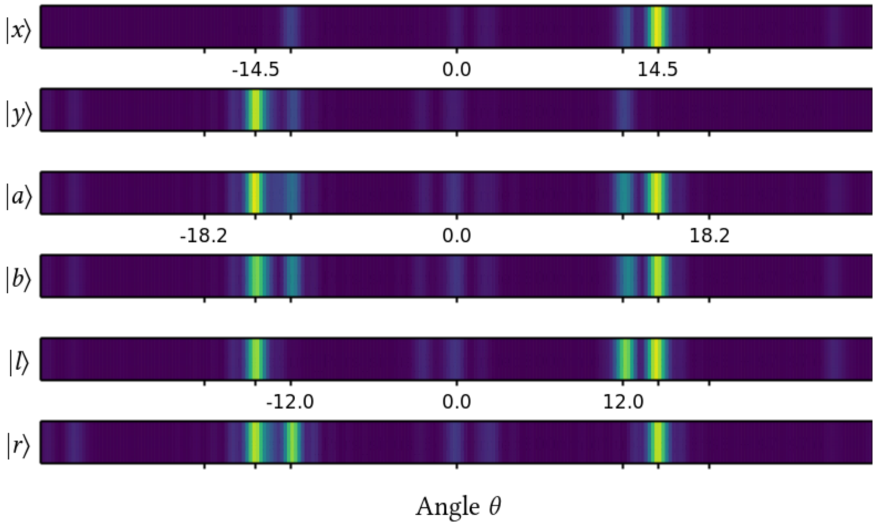


Figure 5.22: Intensity heatmap with regards to angle of all modes of polarized light being reflected off the MS1-3 alternating interleave.

Table 5.13: Simplified table of Stokes parameters for varying setups using the MS1 and MS3 patterns.

MS1& 3	$Q(14.5^{\circ})$	$Q(-14.5^{\circ})$	$V(12.0^{\circ})$	$V(-12.0^{\circ})$
No Interleave	0.996433	−0.996280	**−0.999111**	**0.994226**
Alternating Interleave	0.998417	−0.998628	**−0.994825**	**0.989744**

The highlighted table values are for a later discussion in section 5.5.3.

Table 5.14: $1 - L^2$ symmetry values for MS1 + MS3, and MS1-3 alternating interleave.

Type	$\lvert x\rangle$ vs. $\lvert y\rangle$	$\lvert a\rangle$ vs. $\lvert b\rangle$	$\lvert l\rangle$ vs. $\lvert r\rangle$
No Interleave	0.028489	0.040454	0.050337
Alternating Interleave	0.016285	0.023954	0.037389

As can be seen, the symmetry values were improved across the board.

MS2& 3 alternating interleave

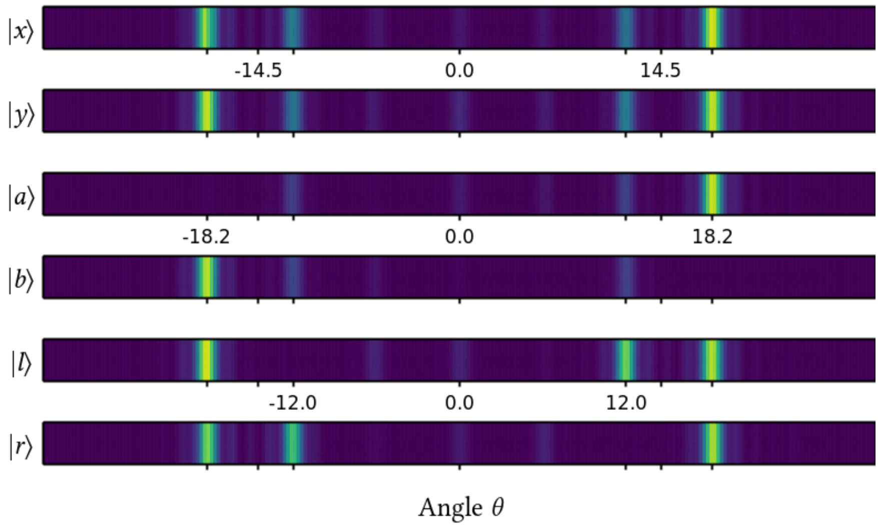


Figure 5.23: Intensity heatmap with regards to angle of all modes of polarized light being reflected off the MS2-3 alternating interleave.

Table 5.15: Simplified table of Stokes parameters for varying setups using the MS2 and MS3 patterns.

MS2& 3	$U(18.25^{\circ})$	$U(-18.25^{\circ})$	$V(12.0^{\circ})$	$V(-12.0^{\circ})$
No Interleave	0.998178	−0.960296	**−0.991484**	**0.999415**
Alternating Interleave	0.999897	−0.999725	**−0.997805**	**0.996788**

Table 5.16: $1 - L^2$ symmetry values for MS1 + MS3, and MS1-3 alternating interleave.

Type	$|x\rangle$ vs. $|y\rangle$	$|a\rangle$ vs. $|b\rangle$	$|l\rangle$ vs. $|r\rangle$
No interleave	0.011319	0.003971	0.001604
Alternating Interleave	0.006513	0.002240	0.002223

As can be seen, the symmetry values were improved for the linear polarization states, but not for the circular.

5.5.3 The low relative intensity for MS3 interleaves

The simulations for the two type comparisons revealed something interesting with regards to the MS3 pattern. The intensity of the directed refraction for circularly polarized light, highlighted with bold-font in tables 5.13 and 5.15, all had intensities at about 80% of that of the other pattern's directional output. This was shown to be consistent for all interleave options involving MS3, but there is a discrepancy between this result and that of the full three type metasurface where the MS3 pattern was weak compared to the others, but still had the highest intensity. This discrepancy calls into question the accuracy of the simulations as there should in principle not be any significant differences between the two-row results and the three row results with regards to relative intensity. Alternatively, an entirely different effect might be at play. The MS3 pattern has the highest absorption rate of incoming waves, but it might be that plasmon polaritons are formed, that travel along the substrate for a short distance, and then stimulates the other pattern with the energy originally "absorbed" by the MS3 pattern block. This hypothesis seems likely as the alternating interleave between MS1 and MS3 had the following intensities:

Table 5.17: Table of Intensities normalized against the highest intensity

Interleave	Pol.	$I(14.5^\circ)$	$I(-14.5^\circ)$	$I(12.0^\circ)$	$I(-12.0^\circ)$
No Int.	$\vert l\rangle$	1.0	0.787254	0.825933	0.006285
No Int.	$\vert r\rangle$	1.0	0.818892	0.002621	0.846501
Alt. Int.	$\vert l\rangle$	1.0	0.937813	0.832051	0.018276
Alt. Int.	$\vert r\rangle$	1.0	0.971209	0.003720	0.864405

Table 5.17 shows that it is consistently the angle 14.5° that has the highest intensity, which makes sense when considering that the MS1 pattern is the one that is the most affected by the unit cell being rectangular instead of square. However, table 5.17 also shows that the difference in intensity between the 14.5° and −14.5° was reduced drastically by the alternating interleave.[3] This supports the hypothesis of the MS3 blocks affecting their neighboring blocks. In the alternating interleave, all adjacent unit cells

[3] The same phenomena happened for the 18.2° degree reflection for the MS2-3 interleave, but with more symmetry between the outputs to begin with as the MS2 blocks are

are of the other type, which means that aside from the length difference between the two axes of the unit cell, there is no difference. Contrast this with the “no interleave” system, which is one row of MS1, and one row of MS3, with periodic boundaries, meaning that every MS3 block can only “spill plasmons” onto the MS1 pattern along the y-axis, which should result in the spillover affecting one side more than the other in the MS1 blocks. Furthermore, the sparse system shown in section 5.4 had the curious effect of the 3-block pattern having better behaviour than the 6-block system, which lends credence to the idea that the MS3 block specification exposes it more towards neighbor interference than the other types. In other words, the high absorption rate displayed in the single gold block amplitude plot in figure 5.7 may change behaviour when surrounded by different block types.

This result is interesting as it indicates that the MS3 blocks’ design has some significant drawbacks compared to the rest. Reducing the spillover from one cell to another in some way should improve the efficiency of the system. Going back to section 5.1.4, it is worthwhile to consider the difference between the MS3 blocks and the rest with regards to amplitude.

angled 45°, which cause any interaction along the x or y-axis to affect both output states more equally.

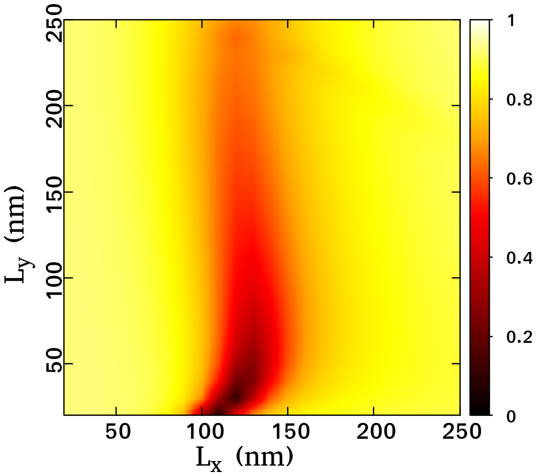


Figure 5.24: The amplitude plot for a single gold block. As can be seen, a gold block with the specifications of the MS3 base block, namely 130 × 60 nm, will put the block's reflection intensity right at the edge of the near-total absorption zone of the plot.

Figure 5.24 shows that the MS3 block will have extremely high absorption rates for certain polarization states. It therefore stands to reason that this could be the root cause, or at least a contributing factor, to why the neighboring blocks were affected to the degree they were, as well as explaining the poor performance of the MS3 grid to begin with, i.e. that while the other blocks may be in the low intensity region, they are not close to the near full extinction region of it. Interestingly, it can be that due to the poor choice of dimensions for the MS3 grid, an important effect was uncovered in how the full system behaves.

However, some of these effects can also be caused by aliasing, as the MS3 pattern has different angles than the other blocks, and the circularly polarized light will hit the blocks with polarization states ranging from anywhere between parallel to orthogonal to the sides of the block. The one indication that this might not be aliasing was the single row results for MS3, tables 5.5 and 5.8, which had almost no difference between the 10 nm and 5 nm results, as well as excellent symmetry values. Therefore, I will not exclude aliasing as a factor, but find it unlikely to be the primary cause.

5.5.4 Remarks

As is shown, the alternating interleave pattern works, indicating that the analysis of the mechanisms involved in the Pors metagrating were correct. This opens up the potential for expanding the complexity of the Pors metagrating without losing the pre-existing properties. It should be stressed that this simulation did not account for coherence width, which is undoubtedly the biggest detrimental factor to this design.

The MS3 pattern gave rise to some concern regarding the base design, the validity of the simulation, as well as the assumptions involved. These concerns could not be addressed due to limitations of time needed to analyze it properly.

6 Final remarks and outlook

In this thesis we have shown that large-scale FDTD simulation of metasurfaces are not only computationally feasible, but also produces physical predictions that are in line with known results produced by FEM solvers and verified in experiments.

The alternative analysis method from section 4.3 of using multiple diffraction pattern to predict the metasurface behaviour yielded predictions that were confirmed by simulations. This opens up new methods of viewing the metagratings which might produce more intricate designs over time. The most important information gained is that the contiguousness of a pattern is not necessarily required for the functionality of the pattern to persist, which allows for increased flexibility in design. This was further evidenced by the results for interleaved patterns, as the birefringence persisted without any significant detriments. Furthermore, the multiple diffraction analysis showed that the minimum number of elements needed for a directed reflection with only one maxima is four, assuming all elements have the same reflectance.

Another important result is that the metasurfaces had higher accuracy in outputting pure Stokes vector states, e.g. $Q \simeq 0,\ U \simeq 1,\ V \simeq 0$, if hit by the intended polarization state, than if it was hit by a polarization state other than the output state. This indicates strongly that the behaviour of the plasmons is not compatible with a linear combination approach, i.e. that $B(|x\rangle) \neq \frac{1}{\sqrt{2}}[B(|a\rangle) + B(|b\rangle)]$, where B is the plasmonic behaviour as a function of incoming polarization. This is a result that is important for further study into plasmonic metasurfaces as it complicates the design process if one wants absolute precision, if it does not directly preclude such a design to begin with.

There is also a concern about the MS3 pattern's gold block size. The pattern's blocks had far stronger interactions with their neighbors than the blocks of the two other patterns. This led to the MS3 pattern having bad results for interleaving, as the effect seemed to bleed over from the MS3 pattern blocks to the neighbors. Assuming that it is not aliasing causing the particular effect, it would be beneficial to look into a configuration that satisfies the required π-phase shift difference between the axes, but has far less absorbance of incoming light. Ref. [1] reported a large discrepancy in Stokes vector purity of the outputted waves compared to theoretical predictions. The MS3 spillover effect could be a contributing factor to this. However, the probable causes listed in Ref. [1], such as the approximation of the glass substrate having a constant dielectric susceptibility and production inaccuracies, such as the gold blocks ending up rounded at the edges because of the production method, should also be considered major contributing factors when considering the discrepancy between theory and practice.

A promising variant to the Pors Metagrating would be an alternating interleaved MS1-2 grating along the x-axis, but with 12 and 8 block periods, respectively, to separate the angles more, and a 12 block MS3 grating along the y-axis. This variant will have maximal separation and optimal intensity distribution, retain contiguousness for the MS3 pattern, while minimizing nearest neighbor GSP interactions for MS1 and 2. Due to the size of the system, and the requirement of angular resolution in both principal directions, this simulation was out of the scope of available hardware and time limitations.

In summary, it has been shown that the field of metagratings have incredible engineering potential as there are plenty of approaches available that might yield new innovations.

Bibliography

1. **A. Pors, M.G. Nielsen, S.I. Bozhevolnyi.**
   Plasmonic metagratings for simultaneous determination of Stokes parameters.
   Optica 2, 716 (2015).
   DOI: 10.1364/optica.2.000716
   arXiv: 1609.04691.

2. **T.W. Ebbesen, H.J. Lezec, H.F. Ghaemi, T. Thio, P.A. Wolff.**
   Extraordinary optical transmission through sub-wavelength hole arrays.
   Nature 391, 667–669 (Feb. 1998).
   DOI: 10.1038/35570

3. **J.B. Pendry.**
   Negative Refraction Makes a Perfect Lens.
   Physical Review Letters 85, 3966–3969 (Oct. 2000).
   DOI: 10.1103/physrevlett.85.3966

4. **W. Cai, V. Shalaev.**
   Optical Metamaterials (2010).
   ISBN: 978-1-4419-1150-6
   DOI: 10.1007/978-1-4419-1151-3

5. **A. Sihvola.**
   Metamaterials in electromagnetics.
   Metamaterials 1, 2–11 (Mar. 2007).
   DOI: 10.1016/j.metmat.2007.02.003

6. **C.R. Simovski.**
   On electromagnetic characterization and homogenization of nanostructured metamaterials.
   Journal of Optics 13, 013001 (Jan. 2011).
   DOI: 10.1088/2040-8978/13/1/013001

7. **J. Pendry, A. Holden, D. Robbins, W. Stewart.**
Magnetism from conductors and enhanced nonlinear phenomena.
IEEE Transactions on Microwave Theory and Techniques 47, 2075–2084 (1999).
DOI: 10.1109/22.798002

8. **S. Linden *et al.***
Magnetic Response of Metamaterials at 100 Terahertz.
Science 306, 1351–1353 (Nov. 2004).
DOI: 10.1126/science.1105371

9. **G. Dolling *et al.***
Cut-wire pairs and plate pairs as magnetic atoms for optical metamaterials.
Optics Letters 30, 3198 (Dec. 2005).
DOI: 10.1364/ol.30.003198

10. **V.M. Shalaev *et al.***
Negative index of refraction in optical metamaterials.
Optics Letters 30, 3356 (Dec. 2005).
DOI: 10.1364/ol.30.003356

11. **R.A. Shelby, D.R. Smith, S. Schultz.**
Experimental verification of a negative index of refraction.
Science (New York, N.Y.) 292, 77–9 (Apr. 2001).
DOI: 10.1126/science.1058847

12. **D.R. Smith, W.J. Padilla, D.C. Vier, S.C. Nemat-Nasser, S. Schultz.**
Composite Medium with Simultaneously Negative Permeability and Permittivity.
Physical Review Letters 84, 4184–4187 (May 2000).
DOI: 10.1103/physrevlett.84.4184

13. **A. Salandrino, N. Engheta.**
Far-field subdiffraction optical microscopy using metamaterial crystals: Theory and simulations.
Physical Review B 74, 075103 (Aug. 2006).
DOI: 10.1103/physrevb.74.075103

14. **Z. Liu, H. Lee, Y. Xiong, C. Sun, X. Zhang.**
Far-Field Optical Hyperlens Magnifying Sub-Diffraction-Limited Objects.
Science 315, 1686–1686 (Mar. 2007).
DOI: 10.1126/science.1137368

15. **Z. Jacob, L.V. Alekseyev, E. Narimanov.**
Optical Hyperlens: Far-field imaging beyond the diffraction limit.
Optics Express 14, 8247 (Sept. 2006).
DOI: 10.1364/oe.14.008247
arXiv: 0607277 [physics].

16. **Y. Gu, L. Zhang, J.K. Yang, S.P. Yeo, C.W. Qiu.**
Color generation via subwavelength plasmonic nanostructures.
Apr. 2015.
DOI: 10.1039/c5nr00578g
arXiv: 1232009.

17. **D. Schurig *et al.***
Metamaterial electromagnetic cloak at microwave frequencies.
Science 314, 977–980 (Nov. 2006).
DOI: 10.1126/science.1133628
arXiv: arXiv:1011.1669v3.

18. **N. Yu, F. Capasso.**
Optical metasurfaces and prospect of their applications including fiber optics.
Journal of Lightwave Technology 33, 2344–2358 (June 2015).
DOI: 10.1109/jlt.2015.2404860

19. **F. Ding, A. Pors, S.I. Bozhevolnyi.**
Gradient metasurfaces: a review of fundamentals and applications.
Reports on Progress in Physics 81, 026401 (Feb. 2018).
DOI: 10.1088/1361-6633/aa8732
arXiv: 1704.03032.

20. **N. Yu *et al.***
Light Propagation with Phase Discontinuities: Generalized Laws of Reflection and Refraction.
Science 334, 333–337 (Oct. 2011).
DOI: 10.1126/science.1210713

21. **D.J. Griffiths.**
Introduction to electromagnetism 4th editio (1999).
ISBN: 9780321856951

22. **L. Novotny, B. Hecht.**
Principles of nano-optics 2nd editio, 564 (2012).
ISBN: 9781107005464

23. **S.A. Maier.**
Plasmonics: Fundamentals and Applications (2007).
ISBN: 978-0-387-33150-8
DOI: 10.1007/0-387-37825-1

24. **P.B. Johnson, R.W. Christy.**
Optical constants of the noble metals.
Physical Review B 6, 4370–4379 (Dec. 1972).
DOI: 10.1103/physrevb.6.4370
arXiv: arXiv:1011.1669v3.

25. **L.J. Prokopeva, J.D. Borneman, A.V. Kildishev.**
Optical dispersion models for time-domain modeling of metal-dielectric nanostructures.
IEEE Transactions on Magnetics 47, 1150–1153 (May 2011).
DOI: 10.1109/tmag.2010.2091676

26. **P. Lautenschlager, M. Garriga, S. Logothetidis, M. Cardona.**
Interband critical points of CsaAs and their temperature dependence.
Phys. Rev. B 35, 9174–9189 (1987).
DOI: 10.1103/physrevb.35.9174

27. **A. Deinega, S. John.**
Effective optical response of silicon to sunlight in the finite-difference time-domain method.
Optics Letters 37, 112 (2012).
DOI: 10.1364/ol.37.000112

28. **S. Cruciani, V.D. Santis, M. Feliziani, F. Maradei.**
Cole-Cole vs Debye models for the assessment of electromagnetic fields inside biological tissues produced by wideband EMF sources.
cccc2012 Asia-Pacific Symposium on Electromagnetic Compatibility, APEMC 2012 - Proceedings, 685–688 (May 2012).
DOI: 10.1109/apemc.2012.6237915

29. **A.E. Rider, K. Ostrikov, S.A. Furman.**
Plasmas meet plasmonics.
The European Physical Journal D 66, 226 (Sept. 2012).
DOI: 10.1140/epjd/e2012-30273-3

30. **A. Pors, S.I. Bozhevolnyi.**
Plasmonic metasurfaces for efficient phase control in reflection.
Optics Express 21, 27438 (2013).
DOI: 10.1364/oe.21.027438

31. **C.M. Hurd.**
The magnetic susceptibility of silver and gold in the range 6-300°K.
Journal of Physics and Chemistry of Solids 27, 1371–1374 (Sept. 1966).
DOI: 10.1016/0022-3697(66)90130-2

32. **B.E.A. Saleh, M.C. Teich.**
Fundamentals of Photonics, 644–695 (Aug. 1991).
ISBN: 0471839655
DOI: 10.1002/0471213748
arXiv: 0408133v1 [arXiv:cond-mat].

33. **H. Fujiwara.**
Spectroscopic Ellipsometry: Principles and Applications (2007).
ISBN: 9780470060186

34. **J.W. Hovenier.**
Structure of a general pure Mueller matrix.
Applied optics 33, 8318–24 (Dec. 1994).
DOI: 10.1364/ao.33.008318

35. **K. Kim, L. Mandel, E. Wolf.**
Relationship between Jones and Mueller matrices for random media.
J. Opt. Soc. Am. A 4, 433–437 (Mar. 1987).
DOI: 10.1364/josaa.4.000433

36. **D. Gabor.**
Theory of communication. Part 1: The analysis of information.
Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering 93, 429–441 (Nov. 1946).
DOI: 10.1049/ji-3-2.1946.0074

37. **A. Taflove, S.C. Hagness, M. Piket-May.**
Computational Electromagnetics: The Finite-Difference Time-Domain Method, 629–670 (2005).
DOI: 10.1016/b978-012170960-0/50046-3

38. **R. Courant, K. Friedrichs, H. Lewy.**
On the Partial Difference Equations of Mathematical Physics.
IBM Journal of Research and Development 11, 215–234 (Mar. 1967).
DOI: 10.1147/rd.112.0215
arXiv: AD0832715.

39. **A. Vial, T. Laroche, M. Dridi, L. Le Cunff.**
A new model of dispersion for metals leading to a more accurate modeling of plasmonic structures using the FDTD method.
Applied Physics A 103, 849–853 (June 2011).
DOI: 10.1007/s00339-010-6224-9

40. **E.D. Palik.**
Handbook of Optical Constants of Solids, Author and Subject Indices for Volumes I, II, and III (1998).
ISBN: 9780080527710

41. **J.P. Berenger.**
A perfectly matched layer for the absorption of electromagnetic waves.
J. Comput. Phys. 114, 185–200 (Oct. 1994).
DOI: 10.1006/jcph.1994.1159
arXiv: 0021-9991.

42. **J.A. Roden, S.D. Gedney.**
Convolution PML (CPML): An efficient FDTD implementation of the CFS-PML for arbitrary media.
Microwave and Optical Technology Letters 27, 334–339 (Dec. 2000).
DOI: 10.1002/1098-2760(20001205)27:5<334::aid-mop14>3.0.co;2-a

43. **J.P. Bérenger.**
Improved PML for the FDTD solution of wave-structure interaction problems.
IEEE Transactions on Antennas and Propagation 45, 466–473 (Mar. 1997).
DOI: 10.1109/8.558661

44. **M. Kuzuoglu, R. Mittra.**
Frequency dependence of the constitutive parameters of causal perfectly matched anisotropie absorbers.
IEEE Microwave and Guided Wave Letters 6, 447–449 (1996).
DOI: 10.1109/75.544545

45. **J.A. Stratton, L.J. Chu.**
Diffraction theory of electromagnetic waves.
Physical Review 56, 99–107 (July 1939).
DOI: 10.1103/physrev.56.99

46. **P. Petre, T.K. Sarkar.**
Planar Near-Field to Far-Field Transformation Using an Equivalent Magnetic Current Approach.
IEEE Transactions on Antennas and Propagation 40, 1348–1356 (Nov. 1992).
DOI: 10.1109/8.202712
arXiv: arXiv:1011.1669v3.

47. **A. Deinega, I. Valuev.**
Subpixel smoothing for conductive and dispersive media in the finite-difference time-domain method.
Optics Letters 32, 3429 (Dec. 2007).
DOI: 10.1364/ol.32.003429

48. **I. Valuev, A. Deinega, S. Belousov.**
Iterative technique for analysis of periodic structures at oblique incidence in the finite-difference time-domain method.
Optics Letters 33, 1491 (July 2008).
DOI: 10.1364/ol.33.001491

49. **A. Deinega, I. Valuev.**
Long-time behavior of PML absorbing boundaries for layered periodic structures.
Computer Physics Communications 182, 149–151 (2011).
DOI: 10.1016/j.cpc.2010.06.006

50. **A. Deinega, T. Seideman.**
Self-interaction-free approaches for self-consistent solution of the Maxwell-Liouville equations.
Physical Review A - Atomic, Molecular, and Optical Physics 89, 022501 (Feb. 2014).
DOI: 10.1103/physreva.89.022501

51. **I. Valuev, A. Deinega, S. Belousov.**
Implementation of the iterative finite-difference time-domain technique for simulation of periodic structures at oblique incidence.
Computer Physics Communications 185, 1273–1281 (Apr. 2014).
DOI: 10.1016/j.cpc.2014.01.001

52. **H.M. Pedersen.**
TFY4195-Optics Compendium (2004).

53. **Z. Bomzon, G. Biener, V. Kleiner, E. Hasman.**
Space-variant Pancharatnam–Berry phase optical elements with computer-generated subwavelength gratings.
Optics Letters 27, 1141 (July 2002).
DOI: 10.1364/ol.27.001141

54. **M. Kang, T. Feng, H.-T. Wang, J. Li.**
Wave front engineering from an array of thin aperture antennas.
Optics Express 20, 15882 (July 2012).
DOI: 10.1364/oe.20.015882

Sample code

On the following pages, I have enclosed a sample code that demonstrates how to use the EMTL software to make a simulation of the MS1 and MS2 alternating interleave.

```
# include "uiexp.h"
# include "basis_3.hpp" // where the custom made rotation functions are.
                        // in the publicly available version only uiexp.h is needed.

int main(int argc,char **argv){

    //_CrtSetDbgFlag ( _CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF );
    emInit(argc,argv);
    uiExperiment task;
    task.SetPMLType(CPML_TYPE);  // Needed for dispersive materials near borders.
    //
    // NUMOFCELLS
    int typeWidth = 1; // how many of one type in y-direction
    int type = 0; // which type (0 = MS1&2, 1 = MS2&3, 2 = MS3&1)

    int typeSelector = type * typeWidth;

    int numOfXCells = 120;
    int numOfYCells = 2;

    // ex: set typeWidth to 2, type to 0, and numOfYCells to 6, and you get 2xMS1, 2xMS2, 2xMS3
    //     set typeWidth to 3, type to 1, and numOfYCells to 6, and you get 3xMS2, 3xMS3

    double scale = 1.0; // sets internal scale to 1 micrometer.
    double meshStep = 0.010; // 10 nm for resolution.
    double cflParam = 0.5; // CFL parameter set to 0.5, which implies a CFL value half of maximum.
    double simTime = 9.6; // Simulation time set to produce an FFT value at 1.25 freq -> 800 nm.

    double baseX = 0.320/scale;
    double baseY = 0.250/scale;
    double baseZ = 0.400/scale;

    double glassWidth = 0.050/scale;
    double prismHeight = 0.040/scale;

    // centering around 0,0,0, so coords go from -x to x, etc
    double x = baseX * (double)numOfXCells /2.0;
    double y = baseY * (double)numOfYCells /2.0;
    double z = baseZ/2.0;

    Vector_3 bounds(x, y, z);
    task.SetInternalSpace(-bounds,bounds);
    task.AddTFSFPlane(INF, INF, z*0.7);

    task.SetBC(BC_PER,BC_PER,BC_PML);
    task.SetResolution(meshStep, cflParam);

    double inputAng = 0*M_PI/4; // Hard-coded polarization angle for example code. 0 = x-pol.
        // This can be fed as a runtime argument if desirable.
    double wavelength = 800.0;
    int circularPol = 0;
    double circPolDirection = 1; // 1 for right hand, -1 for left hand.


    if (circularPol == 0){
        task.SetPlaneWave(Vector_3(0.0,0.0,-1.0),Vector_3(cos(inputAng),sin(inputAng),0.0));
    }else if (circularPol == 1){
        task.SetPlaneWave(Vector_3(0.0,0.0,-1.0),Vector_3(0.0,0.0,circPolDirection));
    }

    // Setting incoming pulse properties.
    double shapeFactor = 1.0;
    double meshFactor = 5.0*meshStep;


    int chooseSignalWave = 1; // 0 for Berenger, 1 for Sine.
    if (chooseSignalWave == 0){
        task.SetSignal(new Berenger(sqrt(2.0*M_E)/meshFactor,5.0*meshFactor,
                                    shapeFactor*meshFactor,5.0*meshFactor));
    }else if (chooseSignalWave == 1){
        double freqFactor = 1000*scale/wavelength; // flt7 is desired wavelength
        task.SetSignal(new PartOfSinus(sqrt(2.0*M_E)/meshFactor,2.0*M_PI*freqFactor,
                                       0, simTime, circPolDirection));
    }
    // I dug up the automatic settings for berenger pulses and recreated it here.
    // Not really necessary for Berenger, but the same parameters are useful for a Gaussine pulse.

    emMedium au = getAu(scale);
    emMedium glass(1.45*1.45);

    Vector_3 origin(0.0, 0.0, 0.0);

    // SETUP GEOMETRIES
    task.AddObject(au, GetHalfSpace(Vector_3(0,0,-1), Vector_3(0, 0, -glassWidth)));

    task.AddObject(glass, GetPlate(Vector_3(0,0,-1), origin, glassWidth));

    // steps: 1- initiate geometry at origin (z axis can be ignored as it's the rotational axis)
    //        2- rotate accordingly
    //        3- shift location to correct one

    // listed are the estimated dimensions of the original Pors metasurface.
    //      symm are the approximated averages that I deemed most suitable for own simulations.
    // accurate nums: x: {0.035, 0.085, 0.110, 0.120, 0.130, 0.140, 0.145, 0.165, 0.205, 0.205}
    //                y: {0.200, 0.195, 0.160, 0.145, 0.135, 0.130, 0.120, 0.110, 0.085, 0.035}
    //             symm: {0.035, 0.085, 0.110, 0.120, 0.130, 0.140, 0.145, 0.160, 0.200, 0.200}
    double boxDimsMS1[10] = {0.035, 0.085, 0.110, 0.120, 0.130, 0.140, 0.145, 0.160, 0.200, 0.200};

    //                205-40 ...  135-125, 125-135, 120-150, 100-160, 40-220
    double boxDimsMS2[8] = {0.040, 0.100, 0.120, 0.125, 0.135, 0.150, 0.160, 0.210};
    double boxDimsMS3[2] = {0.130, 0.060};

    // displacement constants for finding point to place first unit
    // have disp0y in case simulating larger grids gets relevant.
    double disp0x = -x + baseX/2.0;
    double disp0y = -y + baseY/2.0; // relevant if not single cell in y direction
    Vector_3 disp(disp0x, disp0y, 0.0);

    // IMPORTANT: This part uses personally made rotation functions.
    // These will have to be replaced by the user.

    // MS2 grid constant rotation 45 degrees
    double ang0 = M_PI/4;
    Basis_3 rot0 = MakeRotationalBasis_3(ang0, 0.0, 0.0);

    Polyhedron_3 *tmpBox;

//    int j = 0; // turn this into an outer loop iterator if making larger system
    // and add if-tests instead of commenting out the code.

    int j = type;
    for (int i=0; i<numOfXCells; i+=2){

        tmpBox = NULL;

        if ((j%(typeWidth*3))<typeWidth){

            int iMod = i%10;
            // MS1 version
            Vector_3 boxStart(-boxDimsMS1[iMod]/2.0, -boxDimsMS1[9-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS1[iMod]/2.0, boxDimsMS1[9-iMod]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
        }

        if (((j%(typeWidth*3))>=typeWidth) && ((j%(typeWidth*3))<(typeWidth*2))){
            int iMod = i%8;
            // MS2 version
            Vector_3 boxStart(-boxDimsMS2[iMod]/2.0, -boxDimsMS2[7-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS2[iMod]/2.0, boxDimsMS2[7-iMod]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);

            // constant rotation for MS2
            tmpBox->Rotate(rot0);
        }
        if ((j%(typeWidth*3))>=(typeWidth*2)){

            // MS3 version
            Vector_3 boxStart(-boxDimsMS3[0]/2.0, -boxDimsMS3[1]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS3[0]/2.0, boxDimsMS3[1]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);

            // step-dependent rotation for MS3
            Basis_3 rot = MakeRotationalBasis_3((double)i * M_PI/12.0, 0.0, 0.0);
            tmpBox->Rotate(rot);
        }
        Vector_3 displaceVec(disp0x + (double)i*baseX, disp0y + (double)(0.0)*baseY, 0.0);
        // need to use clone function to shift polyhedron.
        Region_3 *finBox = tmpBox->Clone(displaceVec);
        task.AddObject(au, finBox, 0); //initializes the block as a
                                       //physical gold block in the simulation.
    }

    j = type+1;
    for (int i=1; i<numOfXCells; i+=2){

        tmpBox = NULL;

        if ((j%(typeWidth*3))<typeWidth){
            int iMod = i%10;
            // MS1 version
            Vector_3 boxStart(-boxDimsMS1[iMod]/2.0, -boxDimsMS1[9-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS1[iMod]/2.0, boxDimsMS1[9-iMod]/2.0, prismHeight);
                tmpBox = GetBox(boxStart, boxEnd);

        }

        if (((j%(typeWidth*3))>=typeWidth) && ((j%(typeWidth*3))<(typeWidth*2))){
            int iMod = i%8;
            // MS2 version
            Vector_3 boxStart(-boxDimsMS2[iMod]/2.0, -boxDimsMS2[7-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS2[iMod]/2.0, boxDimsMS2[7-iMod]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
            // constant rotation for MS2
            tmpBox->Rotate(rot0);
        }

        if ((j%(typeWidth*3))>=(typeWidth*2)){
            // MS3 version
            Vector_3 boxStart(-boxDimsMS3[0]/2.0, -boxDimsMS3[1]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS3[0]/2.0, boxDimsMS3[1]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
            // step-dependent rotation for MS3
            Basis_3 rot = MakeRotationalBasis_3((double)i * M_PI/12.0, 0.0, 0.0);
            tmpBox->Rotate(rot);
        }
        Vector_3 displaceVec(disp0x + (double)i*baseX, disp0y + (double)(0.0)*baseY, 0.0);

        Region_3 *finBox = tmpBox->Clone(displaceVec);
        task.AddObject(au, finBox, 0);
    }

    j = type;
    for (int i=1; i<numOfXCells; i+=2){

        tmpBox = NULL;

        if ((j%(typeWidth*3))<typeWidth){
            int iMod = i%10;
            // MS1 version
            Vector_3 boxStart(-boxDimsMS1[iMod]/2.0, -boxDimsMS1[9-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS1[iMod]/2.0, boxDimsMS1[9-iMod]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
        }

        if (((j%(typeWidth*3))>=typeWidth) && ((j%(typeWidth*3))<(typeWidth*2))){
            int iMod = i%8;
            // MS2 version
            Vector_3 boxStart(-boxDimsMS2[iMod]/2.0, -boxDimsMS2[7-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS2[iMod]/2.0, boxDimsMS2[7-iMod]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
            // constant rotation for MS2
            tmpBox->Rotate(rot0);

        }

        if ((j%(typeWidth*3))>=(typeWidth*2)){
            // MS3 version
            Vector_3 boxStart(-boxDimsMS3[0]/2.0, -boxDimsMS3[1]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS3[0]/2.0, boxDimsMS3[1]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
            // step-dependent rotation for MS3
            Basis_3 rot = MakeRotationalBasis_3((double)i * M_PI/12.0, 0.0, 0.0);
            tmpBox->Rotate(rot);
        }

        Vector_3 displaceVec(disp0x + (double)i*baseX, disp0y + (double)(1.0)*baseY, 0.0);

        Region_3 *finBox = tmpBox->Clone(displaceVec);
        task.AddObject(au, finBox, 0);
    }

    j = type+1;
    for (int i=0; i<numOfXCells; i+=2){

        tmpBox = NULL;

        if ((j%(typeWidth*3))<typeWidth){
            int iMod = i%10;
            // MS1 version
            Vector_3 boxStart(-boxDimsMS1[iMod]/2.0, -boxDimsMS1[9-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS1[iMod]/2.0, boxDimsMS1[9-iMod]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
        }

        if (((j%(typeWidth*3))>=typeWidth) && ((j%(typeWidth*3))<(typeWidth*2))){
            int iMod = i%8;
            // MS2 version
            Vector_3 boxStart(-boxDimsMS2[iMod]/2.0, -boxDimsMS2[7-iMod]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS2[iMod]/2.0, boxDimsMS2[7-iMod]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
            // constant rotation for MS2
            tmpBox->Rotate(rot0);
        }

        if ((j%(typeWidth*3))>=(typeWidth*2)){
            // MS3 version
            Vector_3 boxStart(-boxDimsMS3[0]/2.0, -boxDimsMS3[1]/2.0, 0.0);
            Vector_3 boxEnd(boxDimsMS3[0]/2.0, boxDimsMS3[1]/2.0, prismHeight);
            tmpBox = GetBox(boxStart, boxEnd);
            // step-dependent rotation for MS3
            Basis_3 rot = MakeRotationalBasis_3((double)i * M_PI/12.0, 0.0, 0.0);
            tmpBox->Rotate(rot);
        }

        Vector_3 displaceVec(disp0x + (double)i*baseX, disp0y + (double)(1.0)*baseY, 0.0);

        Region_3 *finBox = tmpBox->Clone(displaceVec);
        task.AddObject(au, finBox, 0);
    }

    Vector_3 farPlaneMinus(-x, -y, -z);
    Vector_3 farPlanePlus(x, y, z-meshStep*4.0);

    task.AddNearToFarSet("far0to30",farPlaneMinus,farPlanePlus,Vector_3(),1000,
                         0,M_PI/6,121,0,M_PI,2,DET_F, min(1.0, 50*meshStep));

    task.GetDetectorSet("far0to30")->SetEdges(BOX_FRONT_Z);
    task.GetDetectorSet("far0to30")->SetComplexMode(outMOD|outRE|outIM);

    task.RemoveBinaries(); // this call deletes the raw simulation data in the detector sets.
            // if not used, simulations can end up taking gigabytes of storage at a time.

    task.Calculate(simTime);

    task.Analyze();

    return 0;
}
```