Delft University of Technology

LITERATURE REVIEW
COSSE PROGRAMME

Physics-Informed Neural Networks and Topology Optimization for the Design of Flat Metamaterial Space Optics

Dylan Everingham
(5034027)

March 21st, 2022

Contents

1 Introduction

In this paper I present my preliminary work towards my MSc thesis project titled "Physics-Informed Neural Networks and Topology Optimization for the Design of Flat Metamaterial Space Optics". The goals of my project are to investigate and implement methods for the geometric design of flat optical metasurfaces, with several possible configuration cases and a real-world deployment application in space optics.

My work so far has included a review of both literature on useful background material and projects with similar goals to mine, deliberation with my advisors on the formulation of the design problem, and planning towards software implementation of possible solutions.

I begin with an overview of relevant background material on metamaterials, methods in computational electromagnetics, and algorithmic differentiation. Then I discuss several similar, previously-completed projects which I found during my review of the literature, discussing in depth their methods, including those in physics-based machine learning and topology optimization. Finally, I overview the framing of my project, list my research questions, asses the applicability of discussed methods, and propose some implementation plans.

This project is part of my MSc in Applied Mathematics at TU Delft, itself part of the Computer Simulations for Science and Engineering (COSSE) MSc programme jointly administered by TU Delft, TU Berlin, and KTH.

2 Background Material

In this section, I present introductory discussions on some fields of background material necessary to construct solutions to my design problem. Because the lenses I am concerned with are to be made of an all-dielectric metamaterial, I start by answering what exactly a metamaterial is. Then I formulate and discuss rigorous coupled-wave analysis, or RCWA, a numerical solver method which I will be using to simulate wave scattering as part of my design process. Finally, I introduce algorithmic differentiation, or AD, a mathematical programming technique which will be essential to the physics-based machine learning and optimization methods I hope to apply.

2.1 Terahertz Metamaterials

There is no universally accepted definition of the term **metamaterial**, but it refers in general to engineered composite materials which may exhibit physical properties, most notably electromagnetic response, which are not observed in nature or in their constituent materials. Usually, these properties arise due primarily to some sub-wavelength-scale lattice structure of the material rather than directly from its chemical properties.

Metamaterials may possibly exhibit unusual properties such as having a negative index of refraction for particular wavelengths[1]. They might also be "left-handed" - that is, inside such a material, electromagnetic waves obey a "left-hand rule" and can convey energy in the direction opposite to their phase velocity[2]. This has led to great interest in recent years in their use for developing novel optical devices, such as ultra-thin flat lenses and collimators[3][4][5] and hybrid systems for correcting chromatic and spherical aberrations in traditional refractive lenses[6]. Dielectric metamaterial lenses, or **metalenses**, have been developed in recent years with favorable properties such as full phase coverage[7][8], polarization insensitivity[9][10], high numerical aperture[10], and dynamically controllable focal length and intensity[11]. Applications for such devices exist in many fields and include sub-diffraction limit super lenses, cloaking devices, medical imaging, and flat space optics.

In general, the periodicity of the sub-wavelength-scale structure of a metamaterial is too small for incident electromagnetic waves to scatter between adjacent elements, and so the wave itself is not resonant. However, when the metamaterial contains a metallic element, an incident wave may induce an oscillating current which itself is resonant and emits some response. Such metamaterials

are called **resonant**. Other materials, lacking conductive components and resonant currents, are called **nonresonant**, derive their properties simply from how their structure scatters incident power, and can have arbitrarily small periodicity. Metamaterials can broadly be divided into these two classes.

In this project, I am primarily concerned with nonresonant terahertz metamaterials - those which exhibit desirable properties in the terahertz range, usually defined as 0.1 to 10 THz. The lack of naturally occurring materials and practical technologies which interact with waves in this frequency band is called the **terahertz gap** and has led to great interest in techniques for the development of devices which function in this range. In particular I consider the notion of a dielectric **metasurface**, a surface of sub-wavelength dielectric structures whose optical properties are governed by a large number of geometrical parameters.

Terahertz metasurfaces are of particular concern to my application in space optics due to their ability to replace traditional refractive optics. Because their surface structure is of sub wavelength-scale, they are essentially flat, which can be favorable over bulky silicon refractive lenslets. Recent advances in microfabrication have not only increased the possible geometric degrees of freedom for metasurfaces, but have also made it possible to manufacture and test such devices relatively quickly and to high accuracy, opening up potential for tight-looped design processes[12].

My project deals with the question of how to effectively and efficiently choose good parameters of a metasurface for a particular application. In general, the forward problem, that is, finding how light is scattered by a particular metasurface, is well understood. In simple cases it admits analytical solutions, and is otherwise solved commonly via a number of numerical solver methods for Maxwell's Equations. However, the reverse problem of finding an appropriate lens topology given a desired scattering pattern becomes intrinsically ill-posed in the presence of multiple light scattering, which is the case when dealing with metamaterials[13]. This makes normal numerical solver methods for the reverse problem computationally intractable and necessitates the investigation of more powerful computational techniques from optimization and machine learning.

Several papers which I discuss later in this review [13][14][15][16][17][18] have already proposed and implemented such techniques for similar, but not identical, applications to mine. In order to understand how these techniques work and decide which to apply, I first need to understand methods to solve the forward problem.

2.2 Computational Electromagnetics

A key component of most techniques for metasurface design is a method for predicting a device's electromagnetic response to some incident wave, that is, a **forward solver**. Normally this amounts to solving Maxwell's equations numerically for approximations of the electric and magnetic field values in the space around the device. In order to do so, some understanding of computational electromagnetics is required.

In this section I briefly reintroduce Maxwell's equations and use them as a motivation for rigorous coupled-wave analysis, a semi-analytical fast solver method which I will be taking advantage of in my project.

2.2.1 Maxwell's Equations

In the following table I summarize **Maxwell's equations**, a set of partial differential equations which constitute a model of classical electromagnetics and underlie all solver methods used to analyze and design metasurfaces. We consider them here in differential form.

Name	Statement	Explanation
Gauss's Law	$\boldsymbol{\nabla} \cdot \mathbf{E} = \rho/\epsilon_0$	Electric flux through a closed surface is proportional to the charge enclosed by that surface.
Gauss's Law for Magnetism	$\boldsymbol{\nabla} \cdot \mathbf{H} = 0$	Magnetic monopoles do not exist.
Faraday's Law of Induction	$\boldsymbol{\nabla} \times \mathbf{E} = -\frac{\mathrm{d}\mathbf{B}}{\mathrm{d}t}$	A time-varying magnetic field always accompanies a spatially varying electric field.
Ampere's Law (with Maxwell's Addition)	$\boldsymbol{\nabla} \times \mathbf{H} = \mathbf{J} + \epsilon_0 \frac{\mathrm{d}\mathbf{E}}{\mathrm{d}t}$	The magnetic field induced around a closed loop is related to the electric current through that loop.

Table 1: Statement of Maxwell's Equations in a vacuum, differential form. Here, $\mathbf{E}$ is the electric field intensity (V/m), $\mathbf{H}$ the magnetic field intensity (A/m), ρ a charge density (C/m^3), $\mathbf{J}$ a current density (A/m^2), ϵ_0 the permittivity of free space (F/m), μ_0 the permeability of free space (H/m), and $\boldsymbol{\nabla}\cdot, \boldsymbol{\nabla}\times$ the divergence and curl differential operators.

In the remainder of this section, I consider only propagating $\mathbf{E}$ and $\mathbf{H}$ fields with no charge or current sources, so $\rho = 0$ and $\mathbf{J} = 0$.

A number of numerical simulation methods exist and are widely adopted for finding approximate solutions to these equations for general incident waves and devices, including the finite element method (FEM), the finite difference time domain method (FDTD), and the finite difference frequency domain method (FDFD). Other methods, such as the one discussed immediately below, rely on assumptions about the geometry of a scattering device but can gain increased performance as a trade-off.

2.2.2 Formulation of RCWA

Rigorous coupled-wave analysis, or RCWA, is a semi-analytical method used to solve Maxwell's equations for scattering from layered, periodic dielectric structures, a class of objects which include some metasurfaces. It is considered the dominant method in the analysis of such structures, owing to its high computational efficiency[19]. Consider a 3d dielectric structure such as the one depicted below.

The structure is uniform in the z (longitudinal) direction, and periodic in the x and y (transverse) directions. A single spatial period of the structure is called a **unit cell**. As a "semi-analytical" method, RCWA seeks to solve for the response analytically in the longitudinal direction, while solving numerically in

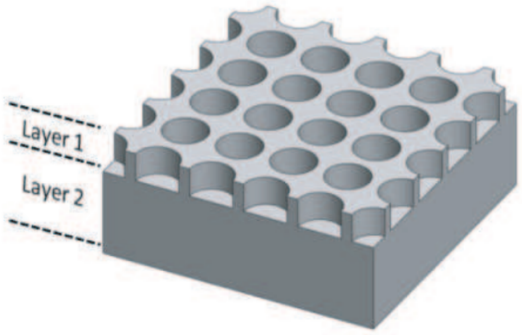

Figure 1: A layered, periodic dielectric structure.

the transverse directions via discretization in Fourier space.

In order to derive this method, Maxwell's equations must first be framed in their time-harmonic form. I will also now consider the solution inside one of the device layers, with some space-dependent relative permittivity and permeability $\epsilon_r(x, y)$ and $\mu_r(x, y)$. Note that these parameters do not depend on z because each of the layers is regular in the longitudinal direction. Because the first two equations are not time-dependent, they remain unchanged, and Faraday's Law and Ampere's Law become:

$$\boldsymbol{\nabla} \times \mathbf{E} = k_0 \mu_r(x, y) \tilde{\mathbf{H}} \text{ and } \boldsymbol{\nabla} \times \tilde{\mathbf{H}} = k_0 \epsilon_r(x, y) \mathbf{E},$$
$$\text{where } k_0 = \omega \sqrt{\mu_0 \epsilon_0}, \text{ and } \tilde{\mathbf{H}} = -i \sqrt{\mu_0 / \epsilon_0} \mathbf{H}$$

Here, k_0 is the wavenumber corresponding to each harmonic of frequency ω, and $\tilde{\mathbf{H}}$ is the normalized magnetic field intensity. This normalization is helpful because it eliminates any sign inconsistency introduced with the complex values and equalizes the amplitudes of $\mathbf{E}$ and $\mathbf{H}$.

A complex Fourier series expansion of the material properties ϵ_r and μ_r in the transverse directions can now be taken, leaving the z direction alone:

$$\epsilon_r(x, y) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} a_{m,n} e^{i(m\mathbf{T_1} + n\mathbf{T_2})\mathbf{r}}$$

$$a_{m,n} = \frac{1}{\Lambda_x \Lambda_y} \int_{-\Lambda_x/2}^{\Lambda_x/2} \int_{-\Lambda_y/2}^{\Lambda_y/2} \epsilon_r(x, y) e^{-i(m\mathbf{T_1} + n\mathbf{T_2})\mathbf{r}} \, dy \, dx$$

$$\mu_r(x, y) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} b_{m,n} e^{i(m\mathbf{T_1} + n\mathbf{T_2})\mathbf{r}}$$

$$b_{m,n} = \frac{1}{\Lambda_x \Lambda_y} \int_{-\Lambda_x/2}^{\Lambda_x/2} \int_{-\Lambda_y/2}^{\Lambda_y/2} \mu_r(x, y) e^{-i(m\mathbf{T_1} + n\mathbf{T_2})\mathbf{r}} \, dy \, dx$$

$\mathbf{T_1} = \frac{2\pi}{\Lambda_x}\hat{x}$ and $\mathbf{T_2} = \frac{2\pi}{\Lambda_y}\hat{y}$ are called **reciprocal lattice vectors**, and provide a simpler way to write the expansion terms. M and N are the number of

spatial harmonics in the expansion in the x and y directions, respectively. They should both be odd and rounded down when divided by 2 so that the expansion is centered on $(m, n) = (0, 0)$. Λ_x and Λ_y are the lengths of the spatial interval of the expansion in x and y, or equivalently the transverse dimensions of the unit cell. Following this expansion, all information about the permittivity and permeability distributions is contained in the Fourier coefficients $a_{m,n}$ and $b_{m,n}$.

The fields can then be expanded as

$$\mathbf{E}(x, y, z) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} \mathbf{S}(m, n; z) e^{-i(k_x(m,n)x + k_y(m,n)y)}$$

$$\tilde{\mathbf{H}}(x, y, z) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} \mathbf{U}(m, n; z) e^{-i(k_x(m,n)x + k_y(m,n)y)}$$

$$\text{where } \mathbf{k}_{xy}(m, n) = \boldsymbol{\beta} - m\mathbf{T_1} - n\mathbf{T_2}, \text{ and}$$

$$k_x(m, n) = \beta_x - mT_{1,x} - nT_{2,x},\ k_y(m, n) = \beta_y - mT_{1,y} - nT_{2,y}$$

This expansion is a representation of the $\mathbf{E}$ and $\mathbf{H}$ fields as a sum of infinite 2d plane waves or spatial harmonics. m and n parameterize the angle and period of each plane wave, and $\mathbf{S}$ and $\mathbf{U}$ are their (z-dependent) complex amplitudes. I adopt the convention that e^{-ikz} represents a wave travelling in the $+z$ direction.

$\mathbf{k}_{xy}$ is the **wave vector** which characterizes each plane wave in our expansion, pointing the direction normal to the wave and with magnitude equal to its wavenumber. Solutions to Maxwell's equations must obey **Blotch's theorem**, which states that waves in a periodic dielectric structure take the form of a plane wave modulated by a periodic function. $\boldsymbol{\beta}$ is the wave vector of this aperiodic plane wave component, which can be thought of equivalently as the wave number of an incident wave to the device. As an example, assuming free space outside of our layer, for an incident plane wave of frequency ω at incident polar angle θ and azimuthal angle ϕ, there is simply

$$\boldsymbol{\beta} = \omega\epsilon_0\mu_0 \begin{bmatrix} sin(\theta)cos(\phi) \\ sin(\theta)sin(\phi) \\ cos(\theta) \end{bmatrix}$$

The goal is to find the complex amplitudes $\mathbf{S}$ and $\mathbf{U}$, which can then be converted back into field values in cartesian space. To do so, the curl operators in our time-harmonic Maxwell equations can be expanded and then the Fourier expansions for $\epsilon_r, \rho_r, \mathbf{E}$ and $\tilde{\mathbf{H}}$ can be substituted in. Consider for Ampere's Law:

Equation		Curl Expanded	
$\boldsymbol{\nabla} \times \mathbf{E} = k_0\mu_r(x, y)\tilde{\mathbf{H}}$	$\implies$	$\begin{cases} \frac{\mathrm{d}E_z}{\mathrm{d}y} - \frac{\mathrm{d}E_y}{\mathrm{d}z} = k_0\mu_r(x, y)\tilde{H}_x \\ \frac{\mathrm{d}E_x}{\mathrm{d}z} - \frac{\mathrm{d}E_z}{\mathrm{d}x} = k_0\mu_r(x, y)\tilde{H}_y \\ \frac{\mathrm{d}E_z}{\mathrm{d}x} - \frac{\mathrm{d}E_x}{\mathrm{d}y} = k_0\mu_r(x, y)\tilde{H}_z \end{cases}$	$\implies$

Semi-Analytical Fourier Space

$$\begin{cases} -i\tilde{k_y}(m,n)S_z(m,n;\tilde{z}) - \dfrac{\mathrm{d}S_y(m,n;\tilde{z})}{\mathrm{d}\tilde{z}} = \displaystyle\sum_{q=-M/2}^{M/2}\sum_{r=-N/2}^{N/2} b_{m-q,n-r}U_x(q,r;\tilde{z}) \\ \dfrac{\mathrm{d}S_x(m,n;\tilde{z})}{\mathrm{d}\tilde{z}} + i\tilde{k_x}(m,n)S_z(m,n;\tilde{z}) = \displaystyle\sum_{q=-M/2}^{M/2}\sum_{r=-N/2}^{N/2} b_{m-q,n-r}U_y(q,r;\tilde{z}) \\ -i\tilde{k_x}(m,n)S_y(m,n;\tilde{z}) + i\tilde{k_y}(m,n)S_x(m,n;\tilde{z}) = \displaystyle\sum_{q=-M/2}^{M/2}\sum_{r=-N/2}^{N/2} b_{m-q,n-r}U_z(q,r;\tilde{z}) \end{cases}$$

Here the normalized wave vector components $\tilde{k}_x = k_x/k_0, \tilde{k}_y = k_y/k_0$ and $\tilde{k}_z = k_z/k_0$ are used, as well as the normalized longitudinal coordinate $\tilde{z} = k_0 z$. These equations hold true for all choices of $m \in [-M/2, M/2]$ and $n \in [-N/2, N/2]$, and so the combined set of all equations for each choice can be written as one matrix equation. To do so, some matrices and column vectors are introduced:

$$\mathbf{s}_x = \begin{bmatrix} S_x(1,1), & S_x(1,2), & \cdots & S_x(M,N) \end{bmatrix}^T$$

and equivalently for $\mathbf{s}_y, \mathbf{s}_z, \mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z$,

$$\tilde{\mathbf{K}}_x = \begin{bmatrix} \tilde{k}_x(1,1) & & & 0 \\ & \tilde{k}_x(1,2) & & \\ & & \ddots & \\ 0 & & & \tilde{k}_x(M,N) \end{bmatrix}, \text{ and equivalently for } \tilde{\mathbf{K}}_y, \tilde{\mathbf{K}}_z.$$

In addition, there are some useful definitions for the material properties:

$$[\![\epsilon_r]\!] = [\epsilon_{r(m,n)}], \text{ for } \epsilon_{r(m,n)} = \sum_{q=-M/2}^{M/2}\sum_{r=-N/2}^{N/2} a_{m-q,n-r} \text{ and}$$

$$[\![\mu_r]\!] = [\mu_{r(m,n)}], \text{ for } \mu_{r(m,n)} = \sum_{q=-M/2}^{M/2}\sum_{r=-N/2}^{N/2} b_{m-q,n-r}$$

These $[\epsilon_r]$ and $[\mu_r]$ are symmetric convolution matrices which contain all information about the permittivity and permeability distributions in the layer. Ampere's Law in semi-analytical Fourier space is then

$$\begin{cases} -i\tilde{\mathbf{K}}_y\mathbf{s}_z - \dfrac{\mathrm{d}}{\mathrm{d}\tilde{z}}\mathbf{s}_y = [\mu_r]\mathbf{u}_x \\ \dfrac{\mathrm{d}}{\mathrm{d}\tilde{z}}\mathbf{s}_x + i\tilde{\mathbf{K}}_x\mathbf{s}_z = [\mu_r]\mathbf{u}_y \\ \mathbf{K}_x\mathbf{s}_y - \tilde{\mathbf{K}}_y\mathbf{s}_x = [\mu_r]\mathbf{u}_z \end{cases}$$

The longitudinal term $\mathbf{u}_z$ can be eliminated by solving for it in the third equation. After expanding the result and simplifying, there is the block matrix form

$$\frac{\mathrm{d}}{\mathrm{d}\tilde{z}}\begin{bmatrix}\mathbf{s}_x \\ \mathbf{s}_y\end{bmatrix} = \mathbf{P}\begin{bmatrix}\mathbf{u}_x \\ \mathbf{u}_y\end{bmatrix}, \quad (1)$$

$$\text{where } \mathbf{P} = \begin{bmatrix} \tilde{\mathbf{K}}_x[\epsilon_r]^{-1}\tilde{\mathbf{K}}_y & [\mu_r] - \tilde{\mathbf{K}}_x[\epsilon_r]^{-1}\tilde{\mathbf{K}}_x \\ \tilde{\mathbf{K}}_y[\epsilon_r]^{-1}\tilde{\mathbf{K}}_y - [\mu_r] & -\tilde{\mathbf{K}}_y[\epsilon_r]^{-1}\tilde{\mathbf{K}}_x \end{bmatrix}$$

Applying the same steps for Faraday's Law, there is

$$\frac{\mathrm{d}}{\mathrm{d}\tilde{z}} \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \mathbf{s}_x \\ \mathbf{s}_y \end{bmatrix}, \quad (2)$$

$$\text{where } \mathbf{Q} = \begin{bmatrix} \tilde{\mathbf{K}}_x[\mu_r]^{-1}\tilde{\mathbf{K}}_y & [\epsilon_r] - \tilde{\mathbf{K}}_x[\mu_r]^{-1}\tilde{\mathbf{K}}_x \\ \tilde{\mathbf{K}}_y[\mu_r]^{-1}\tilde{\mathbf{K}}_y - [\epsilon_r] & -\tilde{\mathbf{K}}_y[\mu_r]^{-1}\tilde{\mathbf{K}}_x \end{bmatrix}$$

Finally, taking the derivative of (2) with respect to $\tilde{z}$ and substituting the result into (1), the **matrix wave equation** for $\mathbf{s}_x$ and $\mathbf{s}_y$ is obtained:

$$\frac{\mathrm{d}^2}{\mathrm{d}\tilde{z}^2} \begin{bmatrix} \mathbf{s}_x \\ \mathbf{s}_y \end{bmatrix} - \mathbf{\Omega}^2 \begin{bmatrix} \mathbf{s}_x \\ \mathbf{s}_y \end{bmatrix} = 0, \text{where } \mathbf{\Omega}^2 = \mathbf{PQ}$$

$$(\textit{Standard PQ Form})$$

This is simply a large number of ODEs which each admit an analytical solution. The solutions are of the form

$$\begin{bmatrix} \mathbf{s}_x(\tilde{z}) \\ \mathbf{s}_y(\tilde{z}) \end{bmatrix} = e^{-\mathbf{\Omega}\tilde{z}}\mathbf{s}^+(0) + e^{\mathbf{\Omega}\tilde{z}}\mathbf{s}^-(0)$$

where $\mathbf{s}^+(0)$ and $\mathbf{s}^-(0)$ are initial values of the problem for forward and backwards propagating waves, respectively.

Now all that is required are the matrices $e^{-\mathbf{\Omega}\tilde{z}}$ and $e^{\mathbf{\Omega}\tilde{z}}$, which can be found using the property

$$f(\mathbf{A}) = \mathbf{W}f(\boldsymbol{\lambda})\mathbf{W}^{-1}, \text{ where}$$

$\mathbf{A}$ is an arbitrary full rank matrix, f an arbitrary matrix function, $\mathbf{W}$ the matrix of A's eigenvectors, and $\boldsymbol{\lambda}$ the matrix of corresponding eigenvalues. Applying this to $\mathbf{\Omega}^2$ with each $f(A) = e^{A\tilde{z}}$ and $f(A) = e^{-A(\tilde{z})}$ gives

$$e^{-\mathbf{\Omega}\tilde{z}} = \mathbf{W}e^{-\boldsymbol{\lambda}\tilde{z}}\mathbf{W}^{-1} \text{ and } e^{\mathbf{\Omega}\tilde{z}} = \mathbf{W}e^{\boldsymbol{\lambda}\tilde{z}}, \text{ where}$$

$$e^{\boldsymbol{\lambda}\tilde{z}} = \begin{bmatrix} e^{\sqrt{\lambda_1^2}\tilde{z}} & & 0 \\ & \ddots & \\ 0 & & e^{\sqrt{\lambda_{N_\lambda^2}}\tilde{z}} \end{bmatrix}$$

Here, $\mathbf{W}$ is the matrix of $\mathbf{\Omega}^2$'s eigenvectors, and $\lambda_1 ... \lambda_{N_\lambda}$ are its corresponding eigenvalues. This gives a final solution for $\mathbf{s}$

$$\begin{bmatrix} \mathbf{s}_x(\tilde{z}) \\ \mathbf{s}_y(\tilde{z}) \end{bmatrix} = \mathbf{W}e^{-\boldsymbol{\lambda}\tilde{z}}\mathbf{c}^+ + \mathbf{W}e^{\boldsymbol{\lambda}\tilde{z}}\mathbf{c}^-, \text{ where}$$

$$\mathbf{c}^+ = \mathbf{W}^{-1}\mathbf{s}^+(0) \text{ and } \mathbf{c}^- = \mathbf{W}^{-1}\mathbf{s}^-(0)$$

Writing a similar expression for $\mathbf{u}$ and combining in order to write the final solution for both s and u inside a single layer, there is

$$\boldsymbol{\psi}(\tilde{z}) = \begin{bmatrix} \mathbf{s}_x(\tilde{z}) \\ \mathbf{s}_y(\tilde{z}) \\ \mathbf{u}_x(\tilde{z}) \\ \mathbf{u}_y(\tilde{z}) \end{bmatrix} = \begin{bmatrix} \mathbf{W} & \mathbf{W} \\ -\mathbf{V} & \mathbf{V} \end{bmatrix} \begin{bmatrix} e^{-\boldsymbol{\lambda}\tilde{z}} & 0 \\ 0 & e^{\boldsymbol{\lambda}\tilde{z}} \end{bmatrix} \begin{bmatrix} \mathbf{c}^+ \\ \mathbf{c}^- \end{bmatrix}, \text{ where}$$

$$\mathbf{V} = \mathbf{Q}\mathbf{W}\boldsymbol{\lambda}^{-1}$$

Therefore, finding the complex amplitudes $\mathbf{s}$ and $\mathbf{u}$, and thus the solution to Maxwell's equations for wave scattering inside a single dielectric layer, amounts to solving the eigenvalue problem for $\boldsymbol{\Omega}^2$ to obtain $\mathbf{W}$ and $\boldsymbol{\lambda}$. There are no strong assumptions about $\boldsymbol{\Omega}^2$, meaning that this is in general a complex-valued degenerate eigenproblem. The implications of this for use of the method in my project are discussed in the section on optimization methods.

However, if the layer is homogeneous, meaning that the material is isotropic and ϵ_r and μ_r are constant values which no longer depend on x and y, some simplifying assumptions can be make in order to obtain the solution

$$\mathbf{W} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \ \boldsymbol{\lambda} = \begin{bmatrix} i\tilde{\mathbf{K}}_z & 0 \\ 0 & i\tilde{\mathbf{K}}_z \end{bmatrix}, \ \tilde{\mathbf{K}}_z = (\sqrt{\mu_r^*\epsilon_r^*\mathbf{I} - \tilde{\mathbf{K}}_x^2 - \tilde{\mathbf{K}}_y^2})^*, \ \mathbf{V} = \mathbf{Q}\boldsymbol{\lambda}^{-1}$$

This means that for homogeneous layers, the eigenvalue problem does not actually need to be solved.

Now consider the problem for a device consisting of multiple non-homogeneous periodic layers. At each layer, $[\![\epsilon_r]\!]$ and $[\![\mu_r]\!]$ are different, and therefore also $\mathbf{P}$, $\mathbf{Q}$, $\boldsymbol{\Omega}^2$, $\mathbf{W}$, $\mathbf{V}$, $\boldsymbol{\lambda}$, $\mathbf{c}^+$ and $\mathbf{c}^-$. However, boundary conditions dictate that $\tilde{\mathbf{K}}_x$ and $\tilde{\mathbf{K}}_y$ remain the same between layers. This means that, for some layer i of thickness L_i, sandwiched by two homogeneous layers 1 and 2 of zero thickness each with constant ϵ_0 and μ_0, we obtain boundary conditions

$$\boldsymbol{\psi}_1 = \boldsymbol{\psi}_i(0) \Rightarrow \begin{bmatrix} \mathbf{W}_0 & \mathbf{W}_0 \\ -\mathbf{V}_0 & \mathbf{V}_0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_0^+ \\ \mathbf{c}_0^- \end{bmatrix} = \begin{bmatrix} \mathbf{W}_i & \mathbf{W}_i \\ -\mathbf{V}_i & \mathbf{V}_i \end{bmatrix} \begin{bmatrix} \mathbf{c}_i^+ \\ \mathbf{c}_i^- \end{bmatrix}, \text{ and}$$

$$\boldsymbol{\psi}_i(k_0 L_i) = \boldsymbol{\psi}_2 \Rightarrow \begin{bmatrix} \mathbf{W}_i & \mathbf{W}_i \\ -\mathbf{V}_i & \mathbf{V}_i \end{bmatrix} \begin{bmatrix} e^{-\boldsymbol{\lambda}_i k_0 L_i} & 0 \\ e^{\boldsymbol{\lambda}_i k_0 L_i} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_i^+ \\ \mathbf{c}_i^- \end{bmatrix} = \begin{bmatrix} \mathbf{W}_2 & \mathbf{W}_2 \\ -\mathbf{V}_0 & \mathbf{V}_0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_0^+ \\ \mathbf{c}_0^- \end{bmatrix}$$

Here, $\mathbf{W}_0$, $\mathbf{V}_0$, $\mathbf{c}_0^+$ and $\mathbf{c}_0^-$ are calculated using the solution for homogeneous layers. For any such layer i a **scattering matrix** $\mathbf{S}^{(i)}$ can be defined such that

$$\begin{bmatrix} \mathbf{c}_1^- \\ \mathbf{c}_2^+ \end{bmatrix} = \mathbf{S}^{(i)} \begin{bmatrix} \mathbf{c}_1^+ \\ \mathbf{c}_2^- \end{bmatrix}, \mathbf{S}^{(i)} = \begin{bmatrix} \mathbf{S}_{11}^{(i)} & \mathbf{S}_{12}^{(i)} \\ \mathbf{S}_{21}^{(i)} & \mathbf{S}_{22}^{(i)} \end{bmatrix}$$

$$\mathbf{S}_{11}^{(i)} = (\mathbf{A}_i - \mathbf{X}_i\mathbf{B}_\mathbf{i}\mathbf{A}_i^{-1}\mathbf{X}_i\mathbf{B}_i)^{-1}(\mathbf{X}_i\mathbf{B}_i\mathbf{A}_i^{-1}\mathbf{X}_i\mathbf{A}_i - \mathbf{B}_i)$$

$$\mathbf{S}_{12}^{(i)} = (\mathbf{A}_i - \mathbf{X}_i\mathbf{B}_i\mathbf{A}_i^{-1}\mathbf{X}_i\mathbf{B}_i)^{-1}\mathbf{X}_i(\mathbf{A}_i - \mathbf{B}_i\mathbf{A}_i^{-1}\mathbf{B}_i)$$

$$\mathbf{S}_{21}^{(i)} = \mathbf{S}_{12}^{(i)}, \ \mathbf{S}_{22}^{(i)} = \mathbf{S}_{11}^{(i)}$$

$$\mathbf{A}_i = \mathbf{W}_i^{-1}\mathbf{W}_0 + \mathbf{V}_i^{-1}\mathbf{V}_0, \ \mathbf{B}_i = \mathbf{W}_1^{-1}\mathbf{W}_0 - \mathbf{V}_i^{-1}\mathbf{V}_0, \ \mathbf{X}_i = e^{-\boldsymbol{\lambda}_i k_0 L_i}$$

Here, c_1 and c_2 correspond to the reflection and transmission sides of the layer, respectively. We can then model an entire device as a series of such layers, separated by homogeneous gap layers of zero thickness with constant

ϵ_0 and μ_0. Such zero thickness gaps have no effect on the physical validity of the model and simplify the calculation of each scattering matrix by ensuring that each depends on only the physical parameters of its corresponding layer and not those of adjacent ones. Thus if our device contains multiple identical layers in different positions, the corresponding scattering matrix only needs to be constructed once.

This convention for the scattering matrix ensures that it is symmetric and thus highly efficient to calculate, as well as consistent with convention[19]. In addition, it allows the matrices of individual layers to be easily combined into a **global scattering matrix** as

$$\mathbf{S}^{(global)} = \mathbf{S}^{(ref)} \otimes \mathbf{S}^{(device)} \otimes \mathbf{S}^{(trn)},$$
$$\mathbf{S}^{(device)} = \mathbf{S}^{(1)} \otimes \mathbf{S}^{(2)} \otimes \cdots \otimes \mathbf{S}^{(N_L)} \text{ for } N_L \text{ layers}$$

Here, $\otimes$ is the **Radheffer star product** defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{B}_{11}(\mathbf{I} - \mathbf{A}_{12}\mathbf{B}_{21})^{-1}\mathbf{A}_{11} & \mathbf{B}_{12} + \mathbf{B}_{11}(\mathbf{I} - \mathbf{A}_{12}\mathbf{B}_{21})^{-1}\mathbf{A}_{12}\mathbf{B}_{22} \\ \mathbf{A}_{21} + \mathbf{A}_{22}(\mathbf{I} - \mathbf{B}_{21}\mathbf{A}_{12}^{-1}\mathbf{B}_{21}\mathbf{A}_{11} & \mathbf{A}_{22}(\mathbf{I} - \mathbf{B}_{21}\mathbf{A}_{12})^{-1}\mathbf{B}_{22} \end{bmatrix}$$

The scattering matrices $\mathbf{S}^{(ref)}$ and $\mathbf{S}^{(trn)}$ represent the device's reflection (i.e. $-z$) and transmission ($+z$) regions, respectively, and are calculated using some constant physical parameters in those regions $\epsilon_{r,ref}, \epsilon_{r,trn}, \mu_{r,ref}$ and $\mu_{r,trn}$ as

$$\mathbf{S}_{11}^{(ref)} = -\mathbf{A}_{ref}^{-1}\mathbf{B}_{ref},\ \mathbf{S}_{12}^{(ref)} = 2\mathbf{A}_{ref}^{-1},$$
$$\mathbf{S}_{21}^{(ref)} = \frac{1}{2}(\mathbf{A}_{ref} - \mathbf{B}_{ref}\mathbf{A}_{ref}^{-1}\mathbf{B}_{ref}),\ \mathbf{S}_{22}^{(ref)} = \mathbf{B}_{ref}\mathbf{A}_{ref}^{-1},$$
$$\mathbf{A}_{ref} = \mathbf{W}_0^{-1}\mathbf{W}_{ref} + \mathbf{V}_0^{-1}\mathbf{V}_{ref},\ \mathbf{B}_{ref} = \mathbf{W}_0^{-1}\mathbf{W}_{ref} - \mathbf{V}_0^{-1}\mathbf{V}_{ref},$$
$$\text{with equivalent expressions for } \mathbf{A}_{trn}, \mathbf{B}_{trn}, \text{ and } \mathbf{S}^{(trn)}$$

Knowing $\mathbf{S}^{(global)}$, an approximation of the transmitted and reflected fields for some incident wave and and arbitrary periodic, layered device can finally be obtained. For some normalized incident polarization vector $\mathbf{P} = [p_x, p_y, p_x]^T, \|\mathbf{P}\| = 1$, and vector $\boldsymbol{\delta}_{0,pq}$ of length $M \times N$, which has 0s in all positions except for a 1 in the p, q^{th} position indicating the mode of the incident wave (normally the center position), there is

$$\mathbf{c}_{inc} = \mathbf{W}_{inc}^{-1} \begin{bmatrix} p_x\boldsymbol{\delta}_{0,pq} \\ p_y\boldsymbol{\delta}_{0,pq} \end{bmatrix}, \text{ and}$$
$$\mathbf{r}_T = \begin{bmatrix} \mathbf{r}_x \\ \mathbf{r}_y \end{bmatrix} = \mathbf{W}_{ref}\mathbf{S}_{11}\mathbf{c}_{inc},\ \mathbf{t}_T = \begin{bmatrix} \mathbf{t}_x \\ \mathbf{t}_y \end{bmatrix} = \mathbf{W}_{ref}\mathbf{S}_{21}\mathbf{c}_{inc}$$

These are the transverse components of the complex wave amplitudes in the reflected and transmitted regions. The longitudinal components are

$$\mathbf{r}_z = -\tilde{\mathbf{K}}_{z,ref}^{-1}(\tilde{\mathbf{K}}_x\mathbf{r}_x + \tilde{\mathbf{K}}_y\mathbf{r}_y),\ \mathbf{t}_z = -\tilde{\mathbf{K}}_{z,trm}^{-1}(\tilde{\mathbf{K}}_x\mathbf{t}_x + \tilde{\mathbf{K}}_y\mathbf{t}_y), \text{ with}$$

$$\tilde{\mathbf{K}}_{z,ref} = -(\sqrt{\mu^*_{r,ref}\epsilon^*_{r,ref}\mathbf{I} - \tilde{\mathbf{K}}_x^2 - \tilde{\mathbf{K}}_y^2})^*, \text{ and equivalently for } \tilde{\mathbf{K}}_{z,trn}$$

To get the final fields in both regions, these plane wave amplitudes can be transformed back to cartesian space, and then propagated any distance z away from the transmission or reflection surface of the device. Several methods for calculating this propagation exist, such as the band-limited angular spectrum method[20]. An implementation of this propagation method is provided in the code of Colburn and Majumdar[17].

2.3 Algorithmic Differentiation

In optimization and deep learning problems it is often very advantageous if the process to be modeled is differentiable with respect to its inputs. Having access to the derivatives of a forward solver enables direct optimization algorithms such as gradient descent for the inverse problem. Access to derivatives is also usually required when modeling a process via a PINN, because their values are needed to evaluate the loss function at each training step.

However, for an arbitrary function implemented as computer program, such as a numerical solver, the derivative is not in general available. There are several techniques which attempt to address this.

In **numerical differentiation**, the method of finite differences is used to compute an approximation to the derivative using several evaluated function values. While simple to implement, this technique can encounter problems with floating-point round-off errors due to the discretization introduced, and its performance does not scale well to higher-order derivatives or derivatives with respect to many inputs. These problems make it ill-suited to an optimization problem in which the target function is highly nonlinear and has many inputs, such as scattering from a metasurface.

In **symbolic differentiation**, one attempts to represent the function as a single mathematical object or expression, which can be used to analytically compute the exact value of a derivative. However, in practice, this is a complicated task which requires the function to be rewritten in a number of fundamental ways - for example, normally numbers in the program must be converted to arbitrary-precision representation, so that the precision limits of standard floating point types do not impact the exactness of the computation. This is highly inefficient for programs with many inputs and operations, which includes most numerical solvers for physical problems.

A third option, **algorithmic differentiation** (also called automatic differentiation, computational differentiation, autodiff, or simply **AD**), attempts to solve all of these complications. AD takes advantage of the fact that any sequential computer program, no matter its complexity, is essentially a series of elementary operations. As long as the derivative of each elementary operation is known, the chain rule can be applied to evaluate the derivative of such a composed function.

AD has two operating modes, each of which computes the same values but with derivatives of the composing operations computed in a different order. In **forward accumulation**, derivatives of the first intermediate value (i.e. that after the first operation has been applied) with respect to the inputs are evaluated first, and the products of each operation's derivative are accumulated as we progress through the operations towards the output. For a function

$y = f^n(f^n - 1(...(f^1(x))$ with intermediate values $w_0 = x$, $w_i = f^{i-1}(w_{i-1})$, $w_n = y$, this computes

$$\frac{\mathrm{d}w_i}{\mathrm{d}x} = \frac{\mathrm{d}w_i}{\mathrm{d}w_{i-1}} \frac{\mathrm{d}w_{i-1}}{\mathrm{d}x} \text{ for } i = 1, 2...n$$

The alternative is **reverse accumulation**, which instead starts from the output and progresses backwards towards the input, computing

$$\frac{\mathrm{d}y}{\mathrm{d}w_i} = \frac{\mathrm{d}y}{\mathrm{d}w_{i+1}} \frac{\mathrm{d}w_{i+1}}{\mathrm{d}w_i} \text{ for } i = n, n-1...1$$

Each of these modes has performance advantages in certain applications. For computing the gradient of a function with a large number of inputs relative to the number of outputs, such as a loss function for a neural network, the reverse mode is far more computationally efficient[17].

2.3.1 Correctness of AD

Standard autodiff frameworks following this framework can be formally shown to be correct, in the sense of correctly computing the derivative at almost all points in the domain for a large class of possibly non-differentiable functions, given that the function is differentiable almost everywhere and that some assumptions hold on the set of elementary operations[21]. Specifically, each of the elementary operations utilized by the program must be **piecewise analytic under analytic partition** (or **PAP**).

A PAP function $f : X \to Y$ is a piecewise function made up of partitions $\{A^i, f^i\}_{i \in \{1,2...\}}$ such that all f^i are analytic functions over their domains $X^i = \{x \in A^i | f^i(x)\}$, and A^i are analytic partitions.

Recall that an analytic function is infinitely differentiable and equal to its Taylor expansion. An analytic partition is any set $A \in \mathbb{R}^n$ for which some analytic functions $g_j^+ : X_j^+ \to \mathbb{R}$ and $g_k^- : X_k^- \to \mathbb{R}$ on open domains $X_j^+, X_k^- \subseteq \mathbb{R}^n$ for $j \in J$, $k \in K$, $J, K \in \mathbb{Z}_{>0}$ exist such that $A = \{x \in X_j^+ | (\forall j \in J) g_j^+(x) > 0\} \cup \{x \in X_k^- | (\forall k \in K) g_k^-(x) \leq 0\}$. In other words, an analytic partition can be broken up into a number of regions on each of which some analytic function exists such that for all points in that region, the analytic function is strictly positive or less than or equal to zero.

Most of the elementary operations used in common implementations of AD, such as PyTorch and TensorFlow, hold to this condition, and so these autodiff systems can be shown to be correct for all programs written using these operations. An example is the *relu* function $\phi(x) = max(0, x)$, a common activation function for neural networks. This function is not differentiable at 0, but the TensorFlow implementation returns 0 for its derivative evaluated at 0 anyway. This choice makes the TensorFlow implementation of *relu* in fact a PAP function and leads to AD correctness for programs using this operation.

2.3.2 Implementation of AD

There are several approaches to the implementation of programs as to utilize algorithmic differentiation, a programming paradigm known as **differentiable**

programming. A straightforward approach might be to overload basic arithmetic operators so that they return information about their derivatives in addition to their results. This can be accomplished efficiently in many languages - for example, C++.

Another approach is to formulate the function as a symbolic computational graph, with nodes representing operations and edges representing data dependency between them. This is the strategy employed by TensorFlow, a widely-used, flexible, open-source software framework for machine learning[22]. First released in 2015, TensorFlow provides a high-level scripting interface which can be used to compose such program graphs from elementary operations such as matrix multiplications and convolutions, and to execute these programs on powerful heterogeneous computer architectures including GPUs. This has enabled many applications of differential programming and machine learning across sectors.

Implementations of differentiable versions of common solver methods from computational electromagnets using AD have already been shown, including FDTD[23], FDFD[24], and RCWA[17].

3 Previous Methods and Results

In this section, I present a number of previous papers which address the use of AI and optimization techniques for the design of metamaterial optical devices and metasurfaces, discussing their methods in depth. I have categorized them into three groups based on their general approach.

3.1 Scientific Machine Learning

The first class of solutions belongs to what can be described as broadly as **scientific machine learning**. This refers simply to the application of standard methods in machine learning, such as deep neural networks, to scientific data sets. This leads to a neural network which replicates the results of a numerical solver method and can efficiently predict EM response for a device given the parameters of a metasurface, which can be called a **forward network**. This leads in turn to the training of an **inverse network** which optimizes metasurface parameters given some desired response.

3.1.1 Jiang et al.

Jiang et al.[15] applies this idea in order to design metalenses for phase manipulation, with the goal of training a DNN which can predict metasurface geometric parameters for a desired phase spectra. Six geometric parameters are sought: the transverse dimensions of three small TiO_2 structures called nanofins of fixed height, which are tiled across a substrate surface.

First, an FDTD solver is used to generate phase spectra, sampled at 81 frequency points over the 380nm-780nm band, for a sample of 7680 parameter combinations. The phase coefficients can be calculated from the scattering matrices used by FDTD, which resemble those from RCWA. Care is taken to cluster more samples around discontinuities in the phase, so that these discontinuities

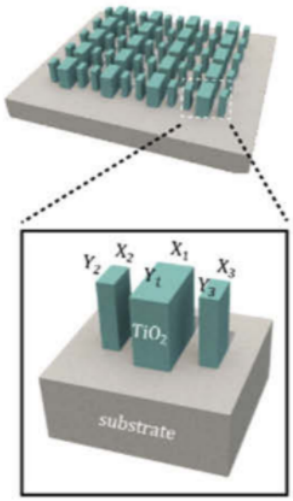

Figure 2: Metasurface design sought by Jiang et al. composed of repeating nanofin structures.

may be properly learned. A DNN with six hidden layers composed of approximately 600 neurons each is implemented using TensorFlow[22] and trained on the majority of the generated data, using a MAE loss function which compares predicted spectra to generated spectra. Network hyperparameters such as batch size and learning rate are varied and compared in terms of network performance, with the best out of those tried selected, but no sophisticated hyperparameter search method is employed. This results in a forward network which models the FTFD solver with an error of <0.2 on 93% of the test data.

Next, the inverse network is initialized. The structure is the same as the forward network except that the loss function now compares predicted spectra to some desired spectra, and that the input layer of geometric parameters is now treated as a hidden layer and the parameters as trainable weights. The rest of the network weights are reused from the forward network and fixed.

In the reverse network optimization process, some arbitrary initial guesses for geometric parameters are first chosen. The network output is calculated, resulting in a loss, which can then be backpropagated in order to update the parameter guess. This is repeated until the loss is small enough, and then the result is checked for correctness with a FDTD simulation. this optimization process is repeated to find other ideal parameters for any additional desired phase spectra. The authors also extend their process to predic and design for other phase properties, including group delay and group delay dispersion, in order to demonstrate the generalizable power of the DNN approach.

3.1.2 An et al.

An et al.[14] attempts a similar goal with a similar approach. The authors consider arrays of small dialectric structures of varying shape with 3-5 geometric parameters and possibly a material parameter (permittivity), called **meta-atoms**, arranged on top of a substrate material of lower refractive index, as

seen below.

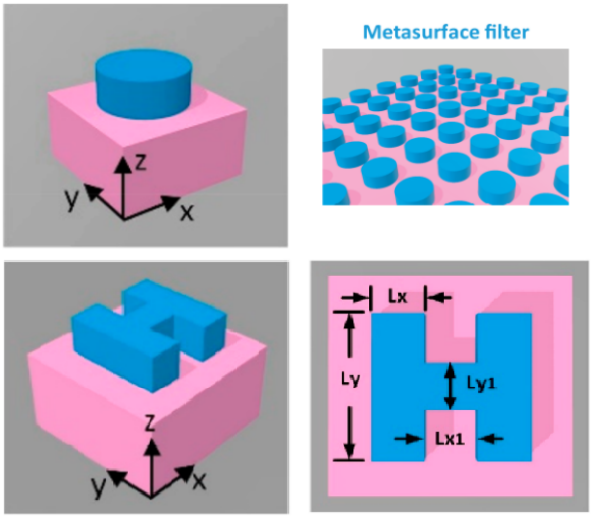

Figure 3: Metasurface design sought by An al. composed of arrays of dielectric "meta-atoms". Separate DNNs are employed for parameter selection of each meta-atom structure, such as the cylindrical and H-shaped ones seen here.

As in Jiang et al., a large amount of simulation-generated amplitude and phase spectra training data is created - over 50000 combinations of geometric parameters, and 31 frequency samples over the 30-60THz (5-10μm wavelength) band. Again, the problem of how to ensure that phase discontinuities are properly captured by the DNN model arises. Finding that these discontinuities occur in the amplitude and phase but that the real and imaginary parts of the complex transmission coefficient, from which the amplitude and phase can be caluculated, are continuous, the authors decide to construct two independent networks which predict these parts. The outputs of these two DNNs can then be combined to predict amplitude and phase according to the formulas

$$amplitude = \sqrt{Imag(T)^2 + Real(T)^2}, \text{ and}$$
$$phase = tan^{-1}\,\frac{Imag(T)}{Real(T)}$$

Here, the transmission coefficient T is the ratio of the amplitude of the incident wave to that of the transmitted wave. T can be calculated directly from the results of any full-wave simulation such as FDTD and therefore can be used in the loss function for these DNNs. Each network is again implemented with TensorFlow[22] has three hidden layers consisting of 500, 500, and 200 neurons each. For a simple cylindrical meta-atom design, the resulting trained networks achieve an average MSE error on the test set of 0.00035 for the real part and 0.00023 for the imaginary part of T. The authors go on to repeat this procress, training networks to predict transmission coefficient components for over 30000 meta-atom structures, including the H-shaped one shown above.

These authors take a slightly different approach to the design of the inverse

network, which the call the mete-filter design network. For any one meta-atom structure, the meta-filter design network is a separate DNN with four hidden layers of widths 500, 500, 500, and 50 neurons each which takes as input a desired phase spectra and produces design parameters as output. The resulting parameters are fed to the forward network, which produces a predicted spectra which can be compared to the target to produce a loss. This is then backpropagated in order to update the many weights and biases of the design network, while the forward network parameters remain unchanged. In this way, the trained forward network is used as part of the loss function, effectively a more computationally efficient stand in for a standard numerical solver.

Because the meta-filter design network is a proper DNN of its own here rather than Jiang's one layer added to the forward network, it can be trained once on a data set consisting of many desired phase spectra and afterwards be used to efficiently predict meta-atom parameters for many other desired spectra. This is in contrast to Jiang, wherein the inverse network must be re-optimized for each desired spectra.

3.1.3 Advantages and Disadvantages

Prior to the introduction of neural network-based approaches to these design problems, they would have been solved via more traditional optimization approaches. One mentioned by both Jiang and Al is the practice of first building a large "library" of possible metasurface designs and simulating the response of each with a numerical solver, then using some classical search method to traverse the library and find desired metasurface parameters. Both authors criticize this approach as being inefficient and time-consuming, both due to the time cost of running many numerical simulations as well as the common over-reliance on trial and error or empirical reasoning in the search method which is not sound when applied to the highly nonlinear problem of multiple scattering in metamaterials. They criticize adjoint methods for the same dependency on slow solver methods.

The scientific machine learning approaches discussed in this section still rely on some numerical solver in order to generate training data, but this is a one-time cost and the trained networks are thereafter able to predict metasurface parameters without any further need of the numerical solver.

However, despite these papers' attempts to address this, the highly nonlinear and possibly discontinuous nature of multiple-scattering solutions can lead to the accuracy of these DNN's predictions strongly depending on how well the training data achieves coverage of the design space. In order to maintain a reasonable model accuracy, then, either a prohibitively large corpus of training data must be produced or the design space must be restricted by reducing the number of geometrical metasurface parameters. Both papers do this, allowing for only 6 or less parameters. While allowing the method to be viable, it greatly restricts the possible metasurface designs, which may perhaps lead to highly effective designs being missed.

Another issue is that the general neural network frameworks applied in these papers do not strictly enforce the physical correctness of solutions or the manufacturing admissibility of discovered metasurface parameters. For example, Jiang et al. find that sometimes the network will return negative parameter values which have no physical meaning. This problem is more coherently han-

dled via classical constrained optimization frameworks. As long as nonphysical results remain technically possible, they must be weeded out via validation with numerical solvers.

Because my project seeks to explore many geometrical parameters and retain high physical accuracy, it will be important to overcome both of these issues. The next methods discussed are focused on doing just that.

3.2 Physics-Informed Neural Networks

While the methods discussed above may fall under the category of scientific machine learning, they do not actually take advantage of any physical knowledge known about the system (aside from using numerical solvers based on it to generate training data). A more clever approach would be to embed information about the physical laws, that is, Maxwell's equations or some other set of PDEs, into the learning process itself. This can be achieved by incorporating the PDEs that govern the data set into the network's loss function. The network is then called a **physics-informed neural network**, or a **PINN**.

Say that, for a function $u(\mathbf{x})$ with $\mathbf{x} = (x, y)$ in a domain $\Omega \in \mathbb{R}$, there is a PDE of the form

$$f\left(\mathbf{x}; \frac{\mathrm{d}u}{\mathrm{d}x}, \frac{\mathrm{d}u}{\mathrm{d}y}, \frac{\partial^2 u}{\partial x \partial y}; \lambda\right) = 0, \text{ for } \mathbf{x} \in \Omega,$$

with some mixed boundary conditions

$$u(\mathbf{x}) = g_D(\mathbf{x}) \text{ on } \Gamma_D \subset \partial\Omega \text{ and } \frac{\mathrm{d}u(\mathbf{x})}{\mathrm{d}\mathbf{x}} = \mathbf{g}_R(u, x, y) \text{ on } \Gamma_R \subset \partial\Omega$$

Here λ is an unknown parameter in the PDE system, which could represent, for example, some metasurface parameters. The goal of the PINN is to recover λ.

The PINN then consists of a DNN with an output $\hat{u}(x, y; \theta)$ which approximates u, where θ is a vector of all DNN weights and biases to be trained. The goal is then to train the DNN and optimize θ and λ so as to minimize the error between u and $\hat{u}$. This is achieved with a loss function of the form

$$\mathcal{L}(\theta, \lambda) = w_f \mathcal{L}_f(\theta, \lambda; \mathcal{T}_f) + w_i \mathcal{L}_i(\theta, \lambda; \mathcal{T}_i) + w_b \mathcal{L}_b(\theta, \lambda; \mathcal{T}_b), \text{ where}$$

$$\mathcal{L}_f(\theta, \lambda; \mathcal{T}_f) = \frac{1}{|\mathcal{T}_f|} \sum_{\mathbf{x} \in \mathcal{T}_f} \left\| f\left(\mathbf{x}; \frac{\mathrm{d}\hat{u}}{\mathrm{d}x}, \frac{\mathrm{d}\hat{u}}{\mathrm{d}y}, \frac{\partial^2 \hat{u}}{\partial x \partial y}; \lambda\right) \right\|_2^2,$$

$$\mathcal{L}_i(\theta, \lambda; \mathcal{T}_i) = \frac{1}{|\mathcal{T}_i|} \sum_{\mathbf{x} \in \mathcal{T}_i} \|\hat{u}(\mathbf{x}) - u(\mathbf{x})\|_2^2,$$

$$\mathcal{L}_b(\theta, \lambda; \mathcal{T}_b) = \frac{1}{|\mathcal{T}_b|} \sum_{\mathbf{x} \in \mathcal{T}_b} \left\| (\hat{u}(\mathbf{x}) - g_D(\mathbf{x})) + \left(\frac{\mathrm{d}\hat{u}(\mathbf{x})}{\mathrm{d}\mathbf{x}} - g_R(u, \mathbf{x})\right) \right\|_2^2$$

Here, $\mathcal{L}_f$, $\mathcal{L}_i$ and $\mathcal{L}_b$ are loss terms representing adherence to the PDE, to a desired solution $u(\mathbf{x})$, and to the boundary conditions, respectively, and w_f, w_i and w_b are their weights. $\mathcal{T}_f$, $\mathcal{T}_i$, $\mathcal{T}_b$ are some set of points sampled from Ω at which these conditions are checked, which may be on a grid or chosen randomly.

If the PINN is implemented with an AD system such as TensorFlow, then the derivatives $\frac{\mathrm{d}\hat{u}(\mathbf{x})}{\mathrm{d}\mathbf{x}}$ are easily available at each training step.

The inclusion of the $\mathcal{L}_i$ term makes this an **inverse** problem which finds λ, but requires some known solution $u(\mathbf{x})$. In other words, given some known solution to the PDE, this network can find the PDE parameter λ such that the PDE admits such a solution. A remarkable fact is that with the minimal change of omitting $\mathcal{L}_i$, the PINN functions in a forward direction instead, finding an admissible solution $\hat{u}(\mathbf{x})$ given some λ. Adding this term is of insignificant computational cost, meaning that this PINN framework can solve forward and inverse problems on equal footing[13].

To train the PINN, normal forward evaluation and backpropagation is iterated until the loss is smaller than some σ. Afterwards, the obtained λ and $\hat{u}(\mathbf{x})$ can be verified using a numerical simulation method. The entire architecture is summarized in diagram below:

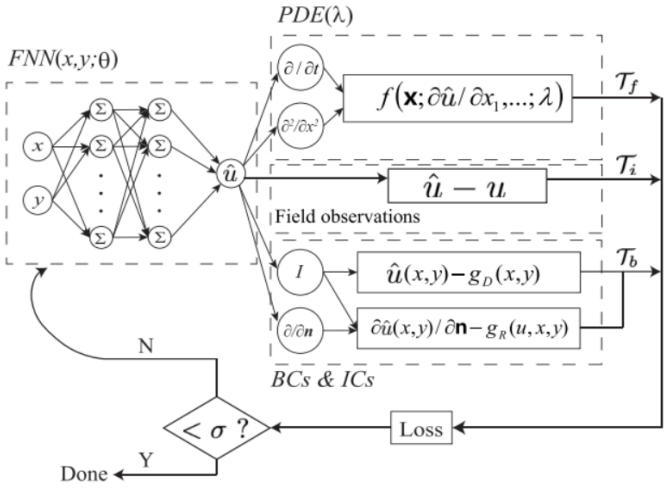

Figure 4: Framework of a general PINN, for the inverse problem to obtain PDE parameter λ. The left FNN (feed-forward neural network) is a model of the PDE solution, and the right side shows the terms of the loss function.

3.2.1 Chen et al.

Chen et al.[13] applies this general framework to the problem of homogenizing finite-size metamaterials. That is, they seek to replace a lattice of cylindrical dielectric meta-atoms each having some constant permittivity ϵ with a single dielectric cylinder having a permittivity profile $\epsilon(x, y)$. Both the lattice of meta-atoms and the singly scatterer should produce the same scattering pattern. For this problem, the relevant PDE is the Helmholtz equation for weakly inhomogeneous 2d media under TM polarization excitation, for the unknown electric field z component $\mathbf{E}_z$:

$$\nabla^2 \mathbf{E}_z(x, y) + \epsilon_r(x, y) k_0^2 \mathbf{E}_z = 0$$

Here k_0 is the wavenumber of the incident wave in free space and $\epsilon_r(x, y)$ is the sought permittivity profile. In terms of the general PINN framework, $u(x, y)$ is generated from a single numerical simulation of the original meta-atom lattice, and the sought parameter λ is ϵ_r. The situation and results can be seen below.

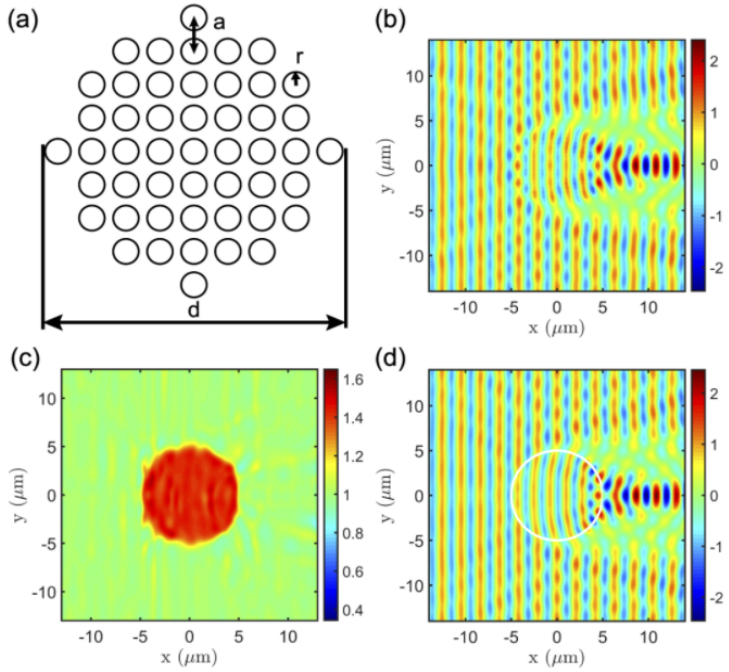

Figure 5: Use of a PINN for retrieval of an effective permittivity profile $\epsilon(x, y)$. a) Arrangement of original meta-atoms whose scattering pattern is desired, with $a = 500nm, r = 125nm, d = 10\mu m, N = 317, \epsilon = 3$. b) Simulated scattering pattern $u(x, y)$ of the meta-atom lattice with incident plane wave wavelength $\lambda_0 = 2.1\mu m$. c) Permittivity profile $\epsilon_r(x, y)$ for a single cylinder, predicted by the PINN. d) Scattering pattern for predicted permittivity profile, with 2.82% error compared to the desired pattern in b).

Chen et al. use DeepXDE, a Python deep learning library, to implement their PINN with 4 layers of 64 neurons each. An FEM simulation, sampling both $\mathbf{E}_z$ and ϵ_r at a 150x150 square lattice spatial resolution, is used both to generate the desired scattering pattern as well as verify the result. The hyperbolic tangent activation function is used, and the training rate is set to 10^{-3}. 150000 iterations of training were run, after which a loss value of 10^{-2} was reached. The retrieved permittivity profile $\epsilon_r(x, y)$ produced a scattering pattern with only 2.82% error in the L^2 norm when compared to the desired pattern.

The authors go on to show that their PINN implementation is valid for solving the homogenization problem for other arrangements of meta-atoms, such as a Vogel spiral. By using the framework to recover effective permittivities for coated cylinders and arrangements of multiple cylinders, they show that their method can be extended to deal with multiple materials and objects. Fi-

nally, they apply their method to the problem of invisible cloaking design, that is, finding the permittivity of a cloaking material which, when applied to a nanocylinder, cancels its scattering for a given incident plane wave.

3.2.2 Advantages and Disadvantages

The magic of the PINN framework is that, by penalizing non-physical solutions, it restricts the space of admissible solutions to a manageable size. Compared to normal machine learning methods applied to the same problem, this loss function acts as a regularization agent and amplifies the information content of used data, allowing the PINN to generalize well even when using much less training data. Only one piece of training data is generated and used, the desired scattering pattern u, compared to the prior approaches in which thousands or hundreds of thousands of simulations are required to generate a massive training corpus. In this small-data regime, most other general machine learning techniques fail to provide any sort of convergence guarantees[25].

In fact, because the inverse PINN requires no data on the parameters λ which it predicts, it belongs to the class of methods called **unsupervised learning**, to which reinforcement learning also belongs[13].

These features of PINNs enable them to overcome the training data limitations of the methods discussed in the previous section and allows for the prediction of greater numbers of metasurface parameters. This is clearly seen in the results of Chen et al., in which the permittivity profile ϵ_r is predicted at every point on a 150x150 square spatial lattice across the domain - 22500 parameters, if considered independently. This is in contrast to the previous methods, which dealt with only 3-6 metasurface parameters[15][14].

Furthermore, this framework can be readily applied in general to the problem of recovering geometrical metasurface parameters for some desired electromagnetic response. In this case, the PDE will be a form of Maxwell's equations such as the formulation seen in RCWA, $u(\mathbf{x})$ will be the desired field distribution, and λ will be a vector of metasurface parameters.

A downside to the PINN framework is that a single trained PINN does not function as a general solution to the inverse problem for multiple problem parameters, and needs to be retrained for each problem statement. In An et al., the authors are able to predict metasurface parameters for many desired phase profiles with repeated evaluations of the same meta-filter design network, without retraining needed in between each. This is highly computationally efficient, if many of these calls are made. In Chen at al., by contrast, the PINN must be retained to predict metasurface parameters for each desired scattering pattern. This is of minimal concern if the number of desired scattering patterns is low, but is nonetheless a performance consideration.

Because my project concerns only a small number of desired scattering patterns, PINNs present a leading and likely effective solution to the problem at hand. However, as discussed in the next section, there are methods which may be just as effective and possibly more efficient without needing to rely on machine learning techniques whatsoever.

3.3 Topology Optimization

The use of classical optimization methods for metasurface design has been criticized due to the high performance cost of running many forward simulations[15][16][13]. This criticism has been leveled in particular at the use of adjoint methods, a class of general methods for numerically computing gradients which can be used for optimization in metamaterial design. This has motivated the application of machine learning techniques to the inverse design problem, as discussed in the previous two sections.

However, papers in recent years have demonstrated the recovery of many metasurface parameters at similar levels of accuracy and efficiency to the deep learning solutions through the use of more traditional optimization methods, enabled through the use of more powerful forward solvers such as RCWA and efficient software implementation with strong low-level optimization in C, parallelism, and AD using TensorFlow[18][17]. These techniques are able to determine unique metalens shapes from an extensive design space consisting of a high number of geometric degrees of freedom, and can be said to belong to the field of **topology optimization**.

3.3.1 Colburn and Majumdar

As discussed in detail in the section on RCWA, despite offering a very high standard of performance for solving the multiple scattering problem for periodic, layered devices, it has issues with respect to differentiability. Specifically, at its core, one must find solutions to the matrix wave equation

$$\frac{\mathrm{d}^2}{\mathrm{d}\tilde{z}^2}\begin{bmatrix}\mathbf{s}_x\\ \mathbf{s}_y\end{bmatrix} - \boldsymbol{\Omega}^2\begin{bmatrix}\mathbf{s}_x\\ \mathbf{s}_y\end{bmatrix} = 0$$

which amounts to solving the eigenproblem for the matrix $\boldsymbol{\Omega}^2$. In order for the entire method to be differentiable and for an AD implementation to be possible, access to the derivatives of these eigenvectors and eigenvalues is required. For certain scatterer geometries, $\boldsymbol{\Omega}^2$ is Hermitian and has no repeated eigenvalues. However, for general scatterers, this does not hold, leading to a complex-valued, degenerate eigenproblem wherein the eigenvector gradients are undefined[26].

Without differentiability, standard optimization techniques for nonlinear functions are not applicable to RCWA. But it so happens that it is possible to make some small modifications to the method, analogous to the modification made to the *relu* function by TensorFlow in order to ensure it is a PAP function, which make an AD implementation of RCWA possible.

Colburn and Majumdar[17] formulate and apply these corrections, resulting in a TensorFlow implementation which can be directly optimized for metasurface parameters using backpropagation. Two innovations are combined. First, they consider an eigenequation of the form $\boldsymbol{\Omega}^2\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}$, where the columns of $\mathbf{W}$ are the eigenvectors of $\boldsymbol{\Omega}^2$ and $\boldsymbol{\Lambda}$ is the diagonal matrix of its eignvalues, and some real scalar function $J = f(\mathbf{W}, \boldsymbol{\Lambda})$ which depends on the eigendecomposition. Using Wirtinger derivatives, operators from complex analysis which enable a differential calculus for complex functions completely analogous to ordinary calculus for real functions, the authors derive an expression for the sensitivity of J to changes in $\boldsymbol{\Omega}^2$. This is written as

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{\Omega}^2} = \mathbf{W}^{-H}\left(\frac{\mathrm{d}J}{\mathrm{d}\mathbf{\Lambda}^*} + \mathbf{F}^* \circ \left(\mathbf{W}^H \frac{\mathrm{d}J}{\mathrm{d}\mathbf{W}^*}\right)\right)\mathbf{W}^H$$

Here, $\mathbf{F}$ is defined as $\mathbf{F}_{ij} = 1/(\lambda_i - \lambda_j)$ if $i \neq j$ and $\mathbf{F}_{ij} = 0$ otherwise, with $\lambda_0, \lambda_1 ... \lambda_n$ the eigenvalues of $\mathbf{\Omega}^2$. $\circ$ is the Hadamard product defined on two matrices A and B of the same dimensions as $(A \circ B)_{ij} = (A) *_{ij} (B)_{ij}$.

This effectively removes any differentiability problems caused by the presence of complex eigenvalues. However, if any eigenvalues are repeated, i.e. are degenerate, $\mathbf{F}_{ij}$ remains undefined. to avoid this problem, the authors redefine $\mathbf{F}$ as

$$\mathbf{F}_{ij} = \frac{\lambda_i - \lambda_j}{(\lambda_i - \lambda_j)^2 + \varepsilon}$$

where ε is a small real number. The introduction of ε introduces also some small error in $\mathbf{F}_{ij}$ but ensures that the reverse sensitivity of J is defined. The result is a slightly modified RCWA which can be implemented correctly with AD, using TensorFlow. The authors present such an implementation, the output of which the validate against and existing RCWA solver, the commonly used S4.

Having this implementation, Colburn and Majumdar now cleverly treat the reverse scattering problem of solving inverse RCWA for some metasurface parameters a_i as a neural network which can be optimized via backpropagation. The scheme is illustrated below:

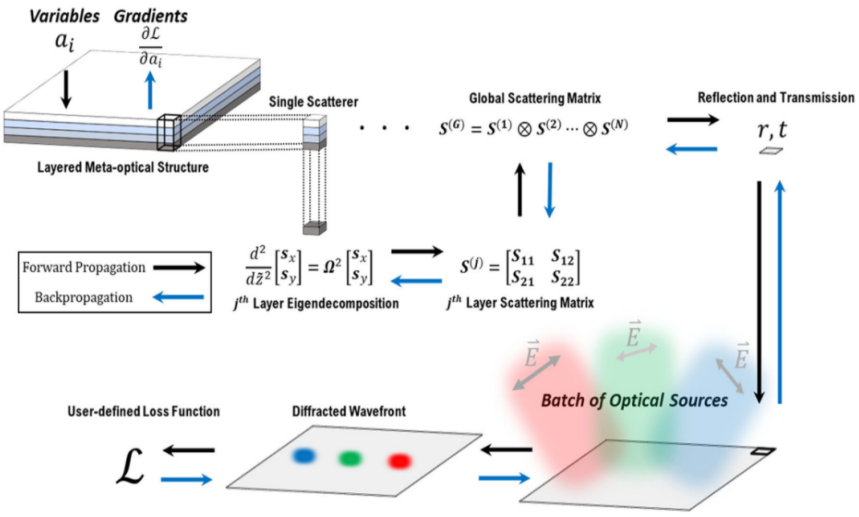

Figure 6: Colburn and Majumdar's scheme for metamaterial design with RCWA and algorithmic differentiation. The figure follows all notation I have used in my section on RCWA.

The authors treat the metamaterial parameters a_i like trainable weights of a neural network. After initial guesses for the a_i are provided, RCWA is used to calculate the $\mathbf{E}$ field in the reflection and transmission regions, following the black arrows. This result is passed to some scalar loss function $\mathcal{L}$, which

defines an optimization criteria - for example, maximizing intensity on some focal plane. Because this has all been implemented using TensorFlow, the gradients $\frac{d\mathcal{L}}{da_i}$ are readily calculated using the reverse mode of AD, following the blue arrows. Then, updates can be applied to the a_i in order to produce new best guesses. This process is iterated until the loss is sufficiently small, at which point some optimized a_i have been obtained. All code is available at `https://github.com/scolburn54/rcwa_tf`.

The approach turns out to be both flexible and powerful. This optimization scheme is shown to find effective parameters for a broad range of periodic, layered metasurface structures, including arrays of independently sized nanopost resonators and multilayer gratings. In addition, a number of optimization objectives and loss functions are demonstrated, such as minimizing reflection and focusing red, blue and green light at different points. For a relatively simple case of optimizing reflectivity for an array of nanocylinders on a substrate, the scheme is able to find a design maximizing reflectivity at 99.8% of the incident light.

Great care is taken by the authors to ensure high performance. In addition to the inherent speed of RCWA, the scheme benefits from the efficiency and parrallelizability of backpropagation. Implementation in TensorFlow allows for easy usage of dedicated parallel architecture including GPUs[22], which is demonstrated to increase performance by 14x to 24x for certain problems.

Compared specifically to adjoint methods, a speedup is demonstrated in terms of time per optimization iteration. Iterations of an adjoint method normally cost twice the forward simulation time because they require two simulations, the forward and adjoint simulations, to compute a gradient. The authors demonstrate that their scheme, which performs one forward simulation and a backpropagation update per iteration, results in a 1.4x speedup over the adjoint standard of two forward simulations for multiple problems. This result and the GPU speedup are shown in the figure below.

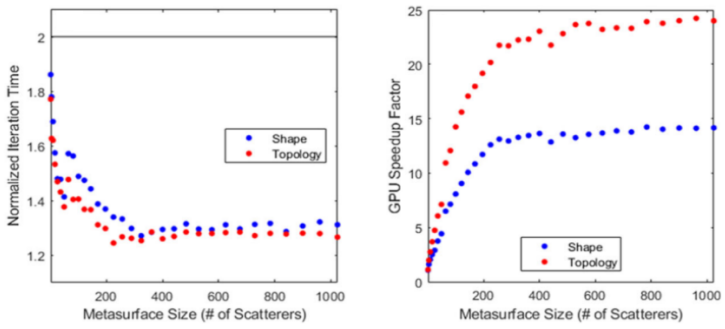

Figure 7: Performance results for the Colburn-Majumdar metaoptics optimization scheme. Both simple optimization of scatterer shape and full topology optimization are considered. Left: optimization iteration time, normalized to the time of a single forward simulation, are compared to the adjoint method (black horizontal line). Right: speedup factor from GPU parallelization vs. problem complexity.

3.3.2 Lin et al.

The paper of Lin et al.[18] adds additional support to the efficiency of topology optimization schemes for periodic, layered metasurface design. The authors use RCWA as their primary forward solver method for each unit cell of the periodic device, with FDFD used to perform a full-device simulation after results are obtained for verification. Unlike Colburn and Majumdar, however, Lin et al. do not leverage AD or backpropagation for performance gains, instead using the basic adjoint method and relying just on efficient implementations of the forward solvers in C and massive parallelization using the message-passing interface (MPI) library.

Surprisingly, this choice turns out to be effective. The authors claim that their straightforward topology optimization approach allows them to handle thousands of degrees of freedom per unit cell, far exceeding the number of parameters optimized by the machine learning approaches. This is demonstrated through the recovery of multi-layer aperiodic topologies for monochromatic and multi-wavelength focusing metalenses, in both 2d and 3d. One example is given below.

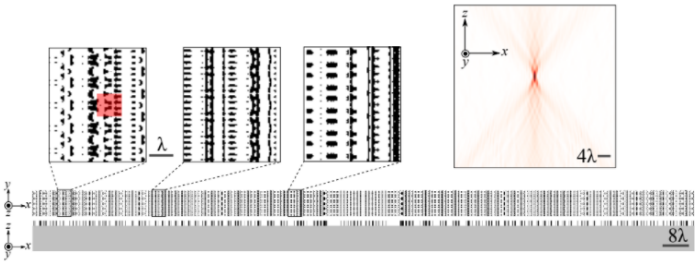

Figure 8: Monochrmatic 3d cylcindrical metalens topography achieved through adjoint method optimization by Lin et al. Red section shows a single λxλ unit cell. Plot on the right shows simulation of the final focusing behavior.

This lens topology contains 4x10^4 degrees of freedom in total, and was computed in parallel on 25 CPUs. The final FDTF validation simulation gives a transmission efficiency of 75%.

Optimization of arbitrary aperiodic topologies using RCWA is achieved though approximating the aperiodic structures as a set of periodic scattering problems, one for each unit cell. In other words, to solve for the near field for each unit cell, a separate problem is solved in which the single unit cell is tiled across an entire device. Each of the problems may be solved entirely in parallel. Afterwards, the final near field is approximated via the periodic near field solutions.

The authors conclude by positing that they expect methods such as theirs to become indispensable for tackling design problems with large design spaces and multiple layers - certainly an expectation I'd like to test.

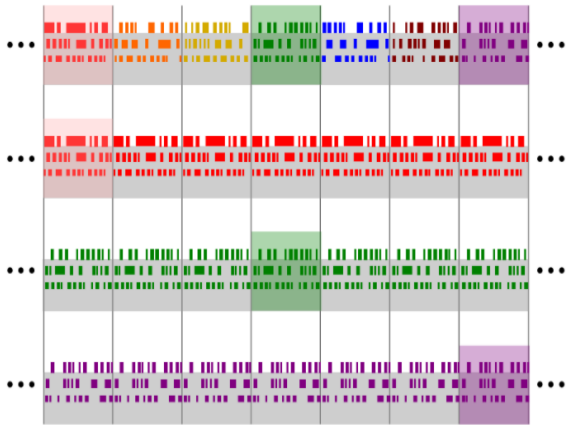

Figure 9: Lin et al. approximate the near field scattering pattern of an arbitrary aperiodic layered metasurface, shown at top, by solving a separate periodic scattering problem for each unit cell. These solutions, shown in the lower three rows, are combined to approximate the near field on the entire surface.

3.3.3 Advantages and Disadvantages

Based on these papers, it seems that, in recent years, the strongest strategy for handling metasurface optimization has been via direct optimization methods either leveraging clever AD formulations or straightforwardly applying high performance computing techniques. They are highly flexible, performative, and accurate.

This is not to say that the PINN approach is strictly worse, as I may simply have failed to find a paper which attempts to optimize as many geometrical parameters with a PINN as these topology optimization paper attempt to with their methods. Probably, which strategy is dominant strongly depends on the application. However, it seems likely that in order to handle highly complex metasurfaces, a PINN would need to adopt some ideas from these topology optimization approaches, such as taking advantage of topology periodicy in order to utilize the fast RCWA method, or utilizing high-performance parallel architecture for training.

The comparison of the PINN approach to the topology optimization approach when applied to my specific application will be one of my central research questions, as discussed in the following section.

4 Project Proposal

In this section, I discuss the framing of my specific project. I first describe the metalens at hand, including its parameters, optimization goals, and application. I propose the research questions which I will explore over the course of the project. Finally, I suggest some possible directions to explore, and discuss what they require as far as software implementation is concerned.

4.1 Lens Design

4.1.1 Physical Description

The existing conventional refractive lens which I seek to replace with a metalens consists of a array of silicon lenslets, as seen below.

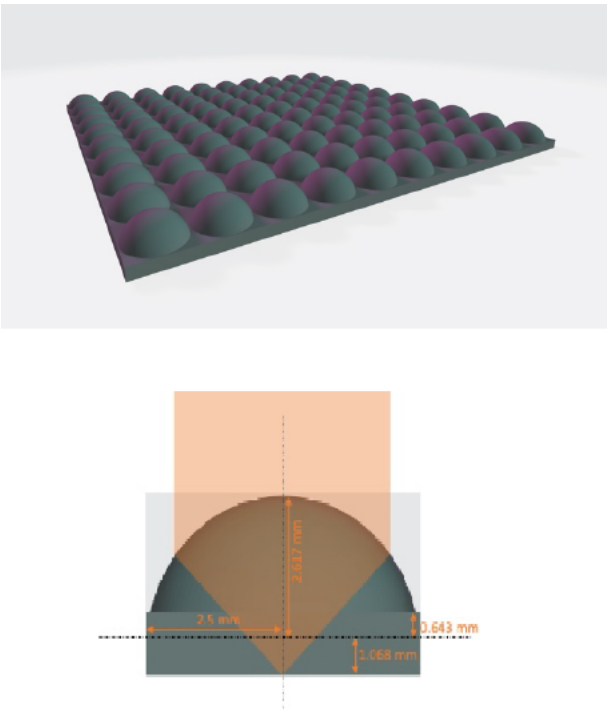

Figure 10: Existing lens design to be replaced, consisting of an array of traditional refractive lenslets.

The high-level goal of the project is to replace this design, considered bulky, with an array of flat metasurface lenslets or unit cells, which are favorable in space optics applications due to their compactness.

For both designs, I consider incident linearly polarized plane waves at $f = 2.5THz$.

The entirety of the device, both the metasurface topology itself as well as the underlying substrate, will consist of high-resistivity, float-zone silicon, which is a sort of highly pure silicon obtained via a process called vertical zone melting. This silicon has been found to be one of the most transparent dielectric materials in the THz domain, making it well suited for construction of nonresonant THz metasurfaces[27]. It displays a permittivity of $\epsilon = 11.4$, and displays several favorable optical properties including very little dispersion over the 0.5-4.5 THz band.

Each lenslet will measure 5mm x 5mm, with a thickness which is left to be determined as a design parameter. They will have a layered, discrete, stepped surface topology consisting of a grid of square regions which I call **pixels**, each

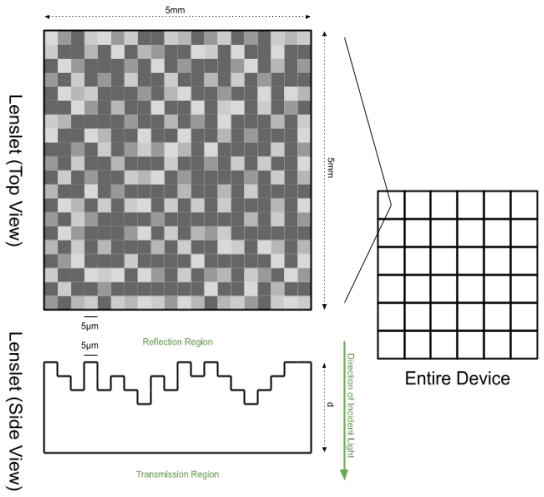

Figure 11: Basic proposed design of metasurface lens for COPILOT, consisting of an array of flat metasurface lenslets. This diagram shows 20x20 pixels per lenslet, which will in reality be 1000x1000. Lenslet thickness d and dimensions of lenslet array to be determined.

approximately $5\mu m$ on a side. This gives a 1000x1000 square grid of pixels on each lenslet. Each pixel will have a discrete height which can take on one of 5 different values. The result is a structure with 5 layers, which can be manufactures via silicon etching[12].

These pixel heights are the primary geometrical parameters required to be optimized in my lens design process. The result of any optimization process should be an array λ of discrete pixel heights, describing the entire topology of a lenslet.

4.1.2 Design Goals

I am concerned with two separate design goals: a device for wave focusing in the **near field**, and one for focusing in the **far field**.

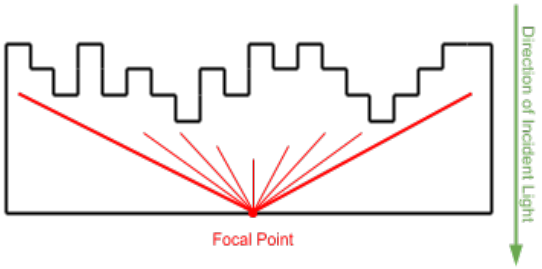

Figure 12: Near field design case. The focal point of each lenslet is the center of its transmission side surface, and all lenslets are identical.

The near field design demands wave focusing at the transmission side of each lenslet. In many metaoptics design processes, the design goal is specified in terms of the phase and direction of a transmitted wave. For the near field case, however, the focal plane is less than one wavelength away from the transmission surface of the device, meaning that there is not enough space to construct a coherent plane wave. This necessitates formulation of the goal in terms of field intensities instead, which are calculable using RCWA.

The near field goal also makes trivial the requirement of RCWA that the device be periodic, because each lenslet will be identical. The periodicity of the device is then simply 5mm.

The far field design demands that all lenslets focus instead on a single point some farther distance away, as shown below.

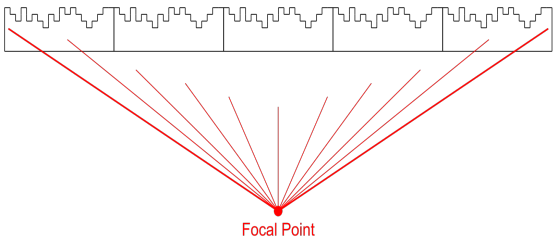

Figure 13: Far field design case. All lenslets focus on a single faraway point, meaning that now they must be not identical.

The training goal can be specified identically to the near field case, except that the field must be propagated further into the transmitted region. However, because now not all lenslets will be identical, a problem is presented to the required device periodicity.

I propose to solve this by introducing "periodicity groups" - groups of lenslets which will each have the same training goal. Each lenslet in a periodicity group will focus on a point at the same relative location to itself.

This is likely to introduce some error in the final design, as the lenslets are not in fact trained to all focus on a single point. However, it allows the enforcement of the devices periodicity required for the use of RCWA, enabling all the same fast design processes as could be used in the near field problem. This speed is especially important in the far field problem, because one lenslet design must be produced per periodicity group, rather than one total as in the near-field problem. Therefore, increasing the size of the periodicity groups also reduces the total amount of computation required. However, each group could be optimized in parallel, as in the paper of Lin et al.[18].

The size of the periodicity groups is left to be determined as a design parameter. If the size is chosen to be 1, then we can apply the idea of Lin et al. of optimizing a single lenslet by considering it as a periodic element tiled across an entire device. This is the most expensive case, but admits the largest design space, allowing arbitrary, aperiodic devices.

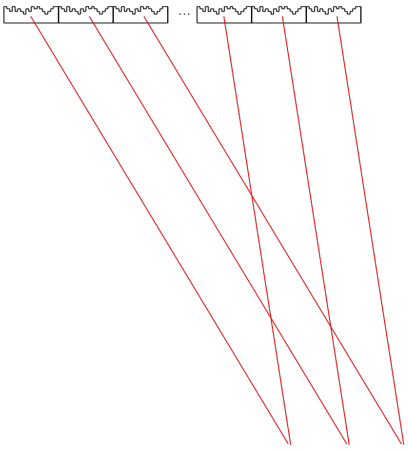

Figure 14: Periodicity groups for far field design case. Groups of lenslets are trained with the same objective, but the total effect of all groups is still to focus on one region. Some error is introduced because lenslets in the same group do not focus on the same point.

4.1.3 Design Parameters

There are a few parameters left unspecified which may be chosen in order to maximize the effectiveness of the metasurface parameter selection. Choosing what to do with these is a core question of the project.

- **Lenslet Thickness** d
  The thickness of each lenslet, labled as d in Figure 11, may be specified before optimization of left as one additional optimizable parameter. In either case, the thickness of the lenslet below the surface etching can be treated as a single extra homogeneous layer, easily handled by RCWA.

- **Single-Sided or Double-Sided Device**
  It is possible for two separate topologies to be etched on the transmission and reflection sides of the device. Doing so would double the number of geometrical metasurface parameters, meaning that the optimization process is more computationally expensive, but may expand the design space to include more effective designs. It is to be determined whether this trade-off is worth it.

- **Size of Periodicity Groups**
  If the periodicity groups are larger, the cost of optimization is reduced, because less individual designs must be produced. However, larger periodicity groups likely lead to greater error in the solution. This is a parameter which will need to be tuned after an optimization strategy for the far field problem is implemented.

- **Single or Multiple Materials**
  The use of multiple materials, perhaps one per layer, with different indices of refraction, is a possibility. This is unlikely to affect the cost of optimization, as RCWA already assumes the possibility of different layer materials.

Whether this actually leads to more effective designs is a question to be answered after an optimization strategy is implemented.

- **Anti-reflective Coating**
  It is a common practice in optics design to incorporate thin-film anti-reflective coatings on the surface of devices in order to eliminate stray light. The simulation of this should be relatively straightforward, as it can be treated as another layer in the RCWA formulation. By doing so, I can investigate if this is worthwhile in this application, or if the metasurface topology alone is sufficient in reducing reflectivity.

4.1.4 Manufacturing

After I have found effective lens designs, the actual lenses will be manufactured by the Kavli Nanolab Delft[28], a nanofabrication research facility at TU Delft which uses the Van Leeuwenhoek Laboratory (VLL) cleanroom facilities. This will enable experimental verification of my results.

There are a few constraints on the design, deriving from manufacturing constraints. Most importantly, only continuous surfaces are allowed, meaning that "bridges" and "tunnels" of material may not be made. Each lenslet pixel will be specified by exactly one height, rather than by which layers are filled or not filled with material.

4.1.5 Deployment

The purpose of this project is to provide viable metamaterial lens designs for a high-altitude balloon mission named COPILOT. The mission is proposed by the French Space agency (CNES), in partnership with the National Research Council of Canada Industrial Research Assistance Program (IRAP), the French Alternative Energies and Atomic Energy Commission (CEA) and the Netherlands Institute for Space Research (SRON).

COPILOT will observe low-density interstellar gas and map the intensity of the C+ fine structure line at 158 micron, otherwise known as the **ionized carbon forbidden line**. This spectral line is produced by ionized carbon that undergoes a **forbidden transition**, which occurs only in low density plasmas and even then at a low rate. It is an indicator of star formation, and information about its angular distribution and redshift can be used to map the milky way galaxy. COPILOT will achieve this mapping at unprecedented speed, covering the entire galaxy up to high galactic latitude. The experiment will improve the angular resolution of the most sensitive C+ map to date by a factor of 200.

4.2 Research Questions

The primary research questions I am concerned with in this project are as follows:

1. What technique is the most effective and computationally efficient for determining metasurface parameters in this application?

The primary goal of the project is to determine effective metasurface parameters and demonstrate a functioning metalens via fabrication. But this will not be possible unless I build an optimization pipeline which is efficient enough to handle my many geometric parameters. This will be a matter of trying different options and comparing their computational efficiencies as well as prediction accuracies.

Of the methods discussed, the two I have identified as most promising are textbftopology optimization with AD and RCWA and a **PINN architecture**. The direct analytical performance comparison of these two options is nontrivial, and will need to be verified in practice. To that end, I aim to produce an implementation of both of these methods and compare their performance and accuracy across a broad range of cases, including the near and far field cases as well as various choices of design parameters.

For the topology optimization solution, I plan to modify the code of Colburn and Majumdar[17] for use with my application. This requires the definition of a loss function for both the near and far field cases. In both cases, the loss function must reward field intensity near the focus point while penalizing intensity in all other directions, in order to minimize stray light which could interfere unfavorably with other parts of the optical system.

This loss function will form one term of the larger PINN loss function. Using the notation from my section on general PINN architectures, this corresponds to $\mathcal{L}_i$. The PDE term $\mathcal{L}_f$ can be formulated using the derived equation for the reflected and transmitted field complex amplitudes in terms of the global scattering matrix from RCWA, with something like

$$\mathcal{L}_f(\mathbf{S}, \mathbf{W}_{ref}, \mathbf{c}_{inc}) = (\mathbf{r}_T - \mathbf{W}_{ref}\mathbf{S}_{11}\mathbf{c}_{inc}) + (\mathbf{t}_T - \mathbf{W}_{ref}\mathbf{S}_{21}\mathbf{c}_{inc})$$

where $\mathbf{S}$ and $\mathbf{W}_{ref}$ can be calculated based on the surface topology and $\mathbf{c}_{inc}$ just depends on the choice of incident wave. The final term, $\mathcal{L}_b$, which deals with boundary conditions, can penalize differences between lenslets which should be periodically identical.

The PINN will be implemented in TensorFlow in order to easily enable parallelization and direct comparison to the topology optimization solution.

There is also the question as to how to deal with the discrete nature of the metasurface topology. As I see it, there are two options - discretizing the pixel heights after each training or optimization step, or discretizing them after continuous optimization has completely finished. By discretize, I mean round a continuous pixel height value off to the nearest of the 5 allowed values.

Without a doubt, both strategies impose some error, but it is unclear at the outset which one minimizes this. With the backpropagation framework of both the TensorFlow RCWA and the PINN, it may be possible to model and track these errors.

2. Which choice of design parameters admits the most effective metasurface designs while maintaining computational feasibility?

As detailed above, there are a number of design parameters available which allow a trade-off between design space complexity and computational cost of optimizing solutions. I want to establish which of these choices most strongly affects the accuracy of designed metasurfaces, and if these choices depend on

whether the near or far field case is being considered.

3. How sensitive are the resulting designs to manufacturing inaccuracies and to the angles of incidence of source waves, and can these sensitivities be minimized?

A final concern is how robust the performance of designed devices is to slight changes in the incident waves or geometrical parameters. Because both of the design approaches take advantage of AD via TensorFlow, I should be able to gain access to the derivative of the loss with respect to these changes relatively easily.

A first step will be to characterize the sensitivities by determining the geometric parameters and incident wave directions in which these derivatives are greatest, for parameters trained with both optimization approaches and different choices of design parameters. If some of these sensitivities turn out to be significant, it may be necessary to penalize them the the loss function. It will then be interesting to investigate if such a sensitivity-penalizing loss function can lead to a good balance between low sensitivity and high accuracy.

For the PINN architecture, incentivizing the good performance for multiple angles of incident waves would probably involve including the incidence angle as an input to the forward solver, and then including some randomly selected angles as part of the loss function sample point sets $mathcalT_f$, $mathcalT_i$ and $mathcalT_b$. The loss function should also take into account this angle, and penalize solutions less when the angle from normal is high.

For the direct topology optimization method, doing the same thing probably requires multiple runs of the forward solver at each iteration. This might remove much of the theoretical performance gain that this scheme has over the PINN, leading to an interesting performance comparison in this case.

Bibliography

[1] D. R. Smith et al. "Composite Medium with Simultaneously Negative Permeability and Permittivity". In: *Phys. Rev. Lett.* 84 (18 May 2000), pp. 4184–4187. DOI: 10.1103/PhysRevLett.84.4184. URL: https://link.aps.org/doi/10.1103/PhysRevLett.84.4184.

[2] D. R. Smith et al. "Left-Handed Metamaterials". In: *Photonic Crystals and Light Localization in the 21st Century*. Ed. by Costas M. Soukoulis. Dordrecht: Springer Netherlands, 2001, pp. 351–371. ISBN: 978-94-010-0738-2. DOI: 10.1007/978-94-010-0738-2_25. URL: https://doi.org/10.1007/978-94-010-0738-2_25.

[3] Xue Jiang et al. "All-dielectric metalens for terahertz wave imaging". In: *Opt. Express* 26.11 (May 2018), pp. 14132–14142. DOI: 10.1364/OE.26.014132. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-26-11-14132.

[4] Takehito Suzuki, Kota Endo, and Satoshi Kondoh. "Terahertz metasurface ultra-thin collimator for power enhancement". In: *Opt. Express* 28.15 (July 2020), pp. 22165–22178. DOI: 10.1364/OE.392814. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-28-15-22165.

[5] Takehito Suzuki et al. “Metalens mounted on a resonant tunneling diode for collimated and directed terahertz waves”. In: *Opt. Express* 29.12 (June 2021), pp. 18988–19000. DOI: 10.1364/OE.427135. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-29-12-18988.

[6] Rajath Sawant et al. “Aberration-corrected large-scale hybrid metalenses”. In: *Optica* 8.11 (Nov. 2021), pp. 1405–1411. DOI: 10.1364/OPTICA.434040. URL: http://opg.optica.org/optica/abstract.cfm?URI=optica-8-11-1405.

[7] Delin Jia et al. “Transmissive terahertz metalens with full phase control based on a dielectric metasurface”. In: *Opt. Lett.* 42.21 (Nov. 2017), pp. 4494–4497. DOI: 10.1364/OL.42.004494. URL: http://opg.optica.org/ol/abstract.cfm?URI=ol-42-21-4494.

[8] Chun-Chieh Chang et al. “Demonstration of a highly efficient terahertz flat lens employing tri-layer metasurfaces”. In: *Opt. Lett.* 42.9 (May 2017), pp. 1867–1870. DOI: 10.1364/OL.42.001867. URL: http://opg.optica.org/ol/abstract.cfm?URI=ol-42-9-1867.

[9] Yufei Gao et al. “Polarization Independent Achromatic Meta-Lens Designed for the Terahertz Domain”. In: *Frontiers in Physics* 8 (2020). ISSN: 2296-424X. DOI: 10.3389/fphy.2020.606693. URL: https://www.frontiersin.org/article/10.3389/fphy.2020.606693.

[10] Hao Chen et al. “Sub-wavelength tight-focusing of terahertz waves by polarization-independent high-numerical-aperture dielectric metalens”. In: *Opt. Express* 26.23 (Nov. 2018), pp. 29817–29825. DOI: 10.1364/OE.26.029817. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-26-23-29817.

[11] Pei Ding et al. “Graphene aperture-based metalens for dynamic focusing of terahertz waves”. In: *Opt. Express* 26.21 (Oct. 2018), pp. 28038–28050. DOI: 10.1364/OE.26.028038. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-26-21-28038.

[12] Kimmo Solehmainen et al. “Development of multi-step processing in silicon-on-insulator for optical waveguide applications”. In: *Journal of Optics A: Pure and Applied Optics J. Opt. A: Pure Appl. Opt* 8 (July 2006), pp. 455–460. DOI: 10.1088/1464-4258/8/7/S22.

[13] Yuyao Chen et al. “Physics-informed neural networks for inverse problems in nano-optics and metamaterials”. In: *Opt. Express* 28.8 (Apr. 2020), pp. 11618–11633. DOI: 10.1364/OE.384875. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-28-8-11618.

[14] Sensong An et al. “A Deep Learning Approach for Objective-Driven All-Dielectric Metasurface Design”. In: *ACS Photonics* 6.12 (2019), pp. 3196–3207. DOI: 10.1021/acsphotonics.9b00966. eprint: https://doi.org/10.1021/acsphotonics.9b00966. URL: https://doi.org/10.1021/acsphotonics.9b00966.

[15] Li Jiang et al. “Neural network enabled metasurface design for phase manipulation”. In: *Opt. Express* 29.2 (Jan. 2021), pp. 2521–2528. DOI: 10.1364/OE.413079. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-29-2-2521.

[16] Fardin Ghorbani et al. “Deep neural network-based automatic metasurface design with a wide frequency range”. In: *Scientific Reports* 11 (Mar. 2021). DOI: 10.1038/s41598-021-86588-2.

[17] Shane Colburn and Arka Majumdar. “Inverse design and flexible parameterization of meta-optics using algorithmic differentiation”. In: *Communications Physics* 4 (Mar. 2021). DOI: 10.1038/s42005-021-00568-6.

[18] Zin Lin et al. “Topology optimization of freeform large-area metasurfaces”. In: *Opt. Express* 27.11 (May 2019), pp. 15765–15775. DOI: 10.1364/OE.27.015765. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-27-11-15765.

[19] Raymond Rumpf. “Improved formulation of scattering matrices for semi-analytical methods that is consistent with convention”. In: *Progress In Electromagnetics Research B* 35 (Aug. 2011), pp. 241–261. DOI: 10.2528/PIERB11083107.

[20] Kyoji Matsushima and Tomoyoshi Shimobaba. “Band-Limited Angular Spectrum Method for Numerical Simulation of Free-Space Propagation in Far and Near Fields”. In: *Opt. Express* 17.22 (Oct. 2009), pp. 19662–19673. DOI: 10.1364/OE.17.019662. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-17-22-19662.

[21] Wonyeol Lee et al. “On Correctness of Automatic Differentiation for Non-Differentiable Functions”. In: *NeurIPS 2020 - 34th Conference on Neural Information Processing Systems*. Vancouver / Virtual, Canada, Dec. 2020. URL: https://hal.inria.fr/hal-03081582.

[22] Martın Abadi et al. “TensorFlow: A System for Large-Scale Machine Learning”. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, Nov. 2016, pp. 265–283. ISBN: 978-1-931971-33-1. URL: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[23] Tyler W. Hughes et al. “Forward-Mode Differentiation of Maxwell’s Equations”. In: *ACS Photonics* 6.11 (Oct. 2019), pp. 3010–3016. ISSN: 2330-4022. DOI: 10.1021/acsphotonics.9b01238. URL: http://dx.doi.org/10.1021/acsphotonics.9b01238.

[24] Logan Su et al. “Nanophotonic inverse design with SPINS: Software architecture and practical considerations”. In: *Applied Physics Reviews* 7.1 (2020), p. 011407. DOI: 10.1063/1.5131263. eprint: https://doi.org/10.1063/1.5131263. URL: https://doi.org/10.1063/1.5131263.

[25] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[26] Matthias Seeger et al. “Auto-Differentiating Linear Algebra”. In: (Oct. 2017).

[27] Jianming Dai et al. “Terahertz time-domain spectroscopy characterization of the far-infrared absorption and index of refraction of high-resistivity, float-zone silicon”. In: *Journal of The Optical Society of America B-optical Physics - J OPT SOC AM B-OPT PHYSICS* 21 (July 2004). DOI: 10.1364/JOSAB.21.001379.

[28] TU Delft. *Kavli Nanolab Description*. 2022. URL: https://www.tudelft.nl/tnw/over-faculteit/afdelingen/quantum-nanoscience/kavli-nanolab-delft (visited on 03/15/2022).