DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
CHENNAI – 600036

Wavelength-Agnostic Design of Next-Generation 2D Photodetectors

A Thesis

Submitted by

AYUSH MUKUND JAMDAR

For the award of the degree

Of

BACHELOR OF TECHNOLOGY

May 2024

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
CHENNAI – 600036

Wavelength-Agnostic Design of Next-Generation 2D Photodetectors

A Thesis

Submitted by

AYUSH MUKUND JAMDAR

For the award of the degree

Of

BACHELOR OF TECHNOLOGY

May 2024

Somewhere, something incredible is waiting

to be known.

– Carl Sagan

To my parents, Dr. Mamta Jamdar and Dr. Mukund Jamdar, who inspire me to contribute to society.

THESIS CERTIFICATE

This is to undertake that the Thesis titled **WAVELENGTH-AGNOSTIC DESIGN OF NEXT-GENERATION 2D PHOTODETECTORS**, submitted by me to the Indian Institute of Technology Madras, for the award of **Bachelor of Technology**, is a bona fide record of the research work done by me under the supervision of **Dr. Sivarama Krishnan**. The contents of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Chennai 600036 **Ayush Mukund Jamdar**

Date: 17 May 2024

Dr. Sivarama Krishnan
Research advisor
Associate Professor
Department of Physics
IIT Madras

Dr. Praveen Bhallamudi
Research co-advisor
Assistant Professor
Department of Physics
IIT Madras

ACKNOWLEDGEMENTS

I thank my advisors Prof. Sivarama Krishnan and Prof. Praveen Bhallamudi (Physics, IIT Madras), and collaborators Prof. Srini Krishnamurthy (Physics, IIT Madras) and Prof. Rituraj (Electrical Engineering, IIT Kanpur) for their advice and support throughout this study.

I acknowledge the use of computing resources at HPCE, IITM.

ABSTRACT

KEYWORDS Photonic Crystals, Metasurfaces, Computational Electromagnetics, RCWA, Evolutionary Algorithms, Inverse Design.

This work presents an inverse design approach for creating two-dimensional (2D) photonic crystal metasurfaces with user-defined functionalities. Metasurfaces offer precise control over light-matter interactions due to their engineered nanostructures. Here, we employ Covariance Matrix Adaptation (CMA) optimization for inverse design, coupled with Rigorous Coupled Wave Analysis (RCWA) as an electromagnetic solver. This approach allows for the optimization of geometric patterns and dimensions within the metasurface to achieve desired resonances, enabling control over light interactions at specific wavelengths or optical modes. We demonstrate the method using a tri-layer structure comprising a 2D material (black phosphorus), a silicon metasurface, and a mirror. This design successfully creates photodetectors operating in the infrared (IR) range. However, the technique extends beyond this specific application and can be used to tailor reflectance and transmittance properties for any optical mode. This computationally efficient method paves the way for designing high-performance 2D metasurface-based devices across various fields, including quantum technologies, communication systems, and nonlinear light conversion.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 METASURFACES: A NEW FRONTIER IN OPTICAL DESIGN

Metasurfaces are emerging as a transformative class of flat optical components capable of manipulating light in novel ways. Unlike conventional lenses and gratings that operate over extended distances, these ultrathin structures achieve highly resolved control of light's phase, amplitude, and polarization within a single wavelength of propagation. This unique capability unlocks a vast potential for creating exotic optical phenomena, such as negative refraction, sub-wavelength microscopy, and broadband achromatic lenses, all in compact and flat devices. Metasurfaces typically consist of periodic arrangements of nanoresonators with precisely defined geometries. These resonators, often crafted from plasmonic materials or high-refractive-index dielectrics, are spaced at subwavelength scales to collectively control the incident light wavefront. This approach offers a significant advantage over conventional optics by achieving sophisticated light manipulation within an exceptionally short propagation distance. Moreover, fabrication techniques for metasurfaces are often more straightforward compared to bulk metamaterials, paving the way for a new era of miniaturized and versatile optical devices [7].

1.2 MOTIVATION AND INTUITION

In a seminal study [29], the authors demonstrated that a photonic crystal metasurface can significantly enhance light absorption in a graphene monolayer from a mere 2.3% [25] to total absorption 100% by inducing resonance at the telecommunication wavelength of 1550 nm. Traditional photodetectors often rely on thick absorber layers, typically hundreds of nanometres thick, for efficient light absorption. However, this comes at the cost of increased dark current compromising device sensitivity. By enabling efficient

absorption in ultra-thin layers, photonic crystals lead to photodetectors with substantially reduced dark current [21, 23, 34].

The study mentioned above [29] underscores a critical feature of 2D photonic crystal devices: their response depends solely on the geometric parameters of the unit cell (periodicity, thickness, and pattern type) for fixed material composition. This dependence offers a powerful tool for designing resonances at desired wavelengths, enabling the creation of photonic crystals with tailored optical responses.

Such a design method can aid the development of optical devices for various applications. Single wavelength total and broadband absorbers are often required in sensing and optical communications [24]. In pump-probe experiments, a device capable of double resonance at two specific wavelengths is essential. This requirement aligns with the concept of pump-probe spectroscopy, where a material is excited by a pump pulse at one wavelength and probed with a separate pulse at another wavelength to study its response [10]. Second-harmonic generation (SHG) is a nonlinear optical phenomenon where an intense light beam interacts with a material, and the material emits light at exactly half the wavelength (and, therefore, double the frequency) of the incoming light. An engineered photonic crystal device can achieve this strong light-matter interaction and exhibit double resonance for SHG [5]. Moreover, such a device will also be useful in Biphoton Generation, where a pump photon interacts with a material and subsequently decays into two entangled photons, each with a different wavelength.

In optical communication systems, efficient operation at both 1.3 μm and 1.55 μm is highly desirable for seamless channel switching and simplified system design [3]. A single device capable of operating at these two key wavelengths would offer significant advantages, including flexibility through dynamical switching and potentially improved performance. When photonic crystal metasurfaces are designed for light transmittance, one can design optical filters that selectively transmit specific light wavelengths and

reject others [27]. Such a filter, when integrated with a phase change material, becomes a 'tuneable filter' that can be applied to hyper-spectral imaging [33].

The ability to design resonances through metasurfaces is crucial to realize these applications that require a tailored optical response from a device. Methods to design these metasurfaces can be classified into forward design and inverse design. Traditionally, forward design either utilizes an exhaustive search through the parameter space (parameters of the metasurface geometry) or a priori intuitive knowledge. However, when complex designs with more dimensions (degrees of freedom) are required to attain highly constrained optical modes, complete exploration of the design space becomes computationally implausible and unrealistic. Moreover, intuitive methods cannot guarantee a globally optimal solution. On the other hand, inverse design methods include topology optimization, machine learning techniques, and evolutionary algorithms [22]. As discussed in [22], topology optimization uses gradient-based local optimizers that are not guaranteed to find the globally best solution since the metasurface parameter space typically involves several local optima. In machine learning techniques, a large dataset must be generated using electromagnetic simulations. The required training data for a problem scales up exponentially as device complexity increases through dimensions [16].

Interestingly, evolutionary algorithms provide a reliable solution to inverse design through stochastic numerical optimization. Methods like genetic algorithms [8], Ant-Colony Optimization (ACO) [20], and Particle Swarm Optimization (PSO) [18] have been applied to metasurface design. In particular, based on adaptive sampling, the Covariance Matrix Adaptation – Evolutionary Strategy (CMA-ES) provides faster convergence and more accurate results [7]. However, when the varied applications of plane 2D periodic photonic crystals discussed earlier are considered, a general design method is required to assist the development of metasurface-based opto-electronic devices. This work suggests that such a method can be developed by combining CMA-ES and Rigorous Coupled

Wave Analysis (RCWA) [19] - a highly efficient Maxwell's equation solver for layered structures invariant along the z direction. The rest of this paper will discuss this approach and experiments.

1.3 LITERATURE SURVEY

This literature survey delves into three key areas that underpin our work on developing novel metasurface functionalities. The first section explores coupling and guided resonance in metasurfaces, using an example of a graphene monolayer. We then discuss current methods in the inverse design of metasurfaces and nanophotonics in general. There on, the Covariance Matrix Adaptation (CMA) algorithm, a robust optimization technique crucial for efficiently navigating the vast parameter space during metasurface design. Finally, the survey examines the Rigorous Coupled-Wave Analysis (RCWA) method, a robust electromagnetic solver essential for simulating the light-matter interaction within complex metasurface structures.

The following sub-sections will describe the key research articles on which this work is based.

1.3.1 Total Absorption in Graphene Monolayer

Key Article: Piper et al., Total absorption in a graphene monolayer in the optical regime by critical coupling with a photonic crystal guided resonance. (2014) [29]

In this study, the authors present a numerical demonstration of achieving total absorption in graphene within the near-infrared and visible wavelength ranges. By employing critical coupling with guided resonances of a photonic crystal slab, the research showcases the ability to enhance absorption in graphene without the presence of plasmonic response in undoped graphene. The control of critical coupling is solely attributed to the properties of the photonic crystal resonance in this wavelength regime.

The paper delves into the theoretical framework and conditions necessary for absorption enhancement and critical coupling in thin films, offering valuable insights for designing a completely absorbing system. Using both a lossless metallic mirror and a realistic multilayer dielectric mirror, the study provides practical examples of achieving total absorption in graphene across the near-infrared and visible spectrum.

Overall, this research contributes to the understanding of enhancing light absorption in graphene through critical coupling with photonic crystal guided resonances, offering design guidelines for developing efficient optical systems in the near-infrared and visible wavelength ranges.

1.3.2 Current Methods in Inverse Design

Key Articles: Zhaoyi et al., Empowering metasurfaces with inverse design: principles and applications. (2021) [22] and Molesky, S et al., Inverse design in nanophotonics. (2018) [28]

Prominent methods in inverse design of metasurfaces fall into on of three categories -

- topology optimization
- machine learning techniques
- evolutionary (search) optimization

The above review articles discuss each of these methods in detail.

Topology Optimization

Topology optimization (TO) [4] is a mathematical method used in inverse design to determine the optimal layout or structure of a device, such as metasurfaces, by distributing material within a defined design space. In the context of metasurfaces, topology optimization involves discretizing the design space into small units and iteratively adjusting the distribution of meta-atoms to achieve specific electromagnetic properties [15]. The process aims to maximize performance metrics, such as light manipulation or

wavefront control, by exploring a wide range of design possibilities.

However, TO has certain drawbacks that need to be attended to. TO is a gradient-based method. This implies that the objective function is differentiated with respect to each input variable, computationally. Almost always, in electromagnetic simulations, when gradients cannot be explicitly computed, gradient computation is expensive. Thus, complex constraints on optical modes cannot be imposed as such designs would require higher degrees of freedom. Moreover, such objective functions have multiple local optima. A zero-gradient objective function does not necessarily find the global optimum.

Machine/Deep Learning

Machine learning algorithms, such as neural networks, are trained on simulation data to predict the performance of different metasurface designs, enabling rapid exploration of the design space and identification of optimal configurations. Deep neural networks consist of layers of artificial neurons connected in series, with different classes such as multilayer perceptrons (MLP) and convolutional neural networks (CNN) commonly used in metasurface design. Neural networks are trained through iterations of forward and backward propagations, adjusting weight and bias parameters to minimize prediction errors.

In the context of metasurfaces, machine learning algorithms can be trained on simulation data to predict the performance of different design configurations, enabling faster exploration of the design space. By leveraging neural networks or other machine learning models, researchers can efficiently search for optimal metasurface designs and identify patterns in the data to guide the design process. In recent years, DNNs have driven the advancement of nanophotonics and metasurfaces research [35, 30].

ML techniques have proven unique strengths in metasurface inverse design. However, their weakness cannot be ignored, and innovations are needed in future development. First, the required training data in ML sets scales up exponentially with the dimensionality

of the design space [16]. This data has to be generated through electromagnetic solvers. Owing to that, ML is difficult to apply to an inverse design of large-scale and complex devices. Second, ML is a data-driven technique in nature. It performs like a "black box" for data analysis without straightforward physical interpretation. The training difficulty arises because its accuracy and stability highly depend on the training set, and extensive hyperparameter tuning may also be required case by case. The lack of underlying physics understanding imposes another application bottleneck.

Evolutionary Optimization

Evolutionary algorithms (EAs) are population-based optimization techniques inspired by biological evolution [2]. These algorithms, such as genetic algorithms, simulate the process of natural selection to iteratively evolve a population of candidate solutions towards an optimal design. In the context of metasurface design, evolutionary algorithms can explore a diverse range of design possibilities without relying on gradient information, making them suitable for complex and non-convex optimization problems.

This method overcomes challenges faced by the previous two methods -

1. **Global Optimization**: Evolutionary algorithms (EAs), such as genetic algorithms, explore a diverse range of solutions and are less likely to get trapped in local optima, making them effective for non-convex optimization problems.

2. **Population-Based Search**: EAs maintain a population of candidate solutions, allowing for parallel exploration of the design space and increasing the likelihood of finding optimal solutions.

3. **Robustness**: Evolutionary algorithms do not rely on gradient information, making them robust to noisy or complex objective functions and suitable for exploring large and intricate design spaces.

4. **Not data-driven**: EAs are a family of search-based optimization methods. Unlike machine learning techniques, which aim to predict the behavior of a metasurface by analyzing a large number of samples, this method doesn't rely on large datasets to find the best solution.

5. **Adaptability**:EAs can adapt to changing design requirements or constraints during the optimization process, providing flexibility in handling dynamic optimization

problems.

In the next sections, we will look at one evolutionary algorithm in detail - Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES).

1.3.3 Global Optimization using Statistical Learning

Key Article: Elsawy et al, Global optimization of metasurface designs using statistical learning methods. (2019) [7]

The optimization of metasurfaces is essential for their integration into practical optical systems. However, existing design techniques often overlook near-field interactions that significantly impact device performance. In this study, the authors utilize advanced optimization methods based on statistical learning and evolutionary strategies in conjunction with a high-order Discontinuous Galerkin time-Domain (DGTD) solver to optimize phase gradient metasurfaces. By demonstrating the superiority of these techniques over existing designs, they showcase the effectiveness of statistical learning in handling complex optimization problems with multiple global minima/maxima. Their focus on GaN semiconductor phase gradient metasurfaces operating at visible wavelengths reveals that rectangular and cylindrical nanopillar arrays can achieve remarkable diffraction efficiencies of over 88% and 85% for TM polarization and both TM and TE polarizations.

Most importantly, this paper describes global optimization of designs using Genetic Algorithms, particularly, Covariance Matrix Adaptation - Evolutionary Strategies (CMA-ES) by applied it to a 3D metasurface of dielectric pillars. The authors explain how a stochastic numerical optimization technique is the most effective way to approach metasurface design, given the nature of the problem.

1.3.4 Covariance Matrix Adaptation

Key Article: Hansen, The CMA Evolution Strategy: A Tutorial. (2016) [11]

This tutorial introduces the CMA Evolution Strategy. The CMA-ES is a stochastic, or randomized, method for real-parameter (continuous domain) optimization of non-linear, non-convex functions. The author tries to motivate and derive the algorithm from intuitive concepts and from requirements of non-linear, non-convex search in a continuous domain.

1.3.5 RCWA

`grcwa` webpage

Key Articles: Jin et al., Inverse design of lightweight broadband reflector for relativistic lightsail propulsion (2020) [17] and Liu et al., A free electromagnetic solver for layered periodic structures (2012) [26]

The Rigorous Coupled-Wave Analysis (RCWA) method is a cornerstone for simulating light propagation and scattering in periodic nanostructures like metasurfaces. `grcwa` is a free and open-source software library specifically designed to implement the RCWA method efficiently.

Key Features of `grcwa`

- It offers a complete framework for performing RCWA simulations. It allows users to define the geometry and material properties of a periodic structure, specify the incident light conditions, and calculate the resulting electromagnetic fields within the structure.
- `grcwa` can handle complex metasurfaces composed of multiple layers with different materials. This is crucial for simulating realistic metasurface designs, which often involve a combination of dielectric and metallic components.
- `grcwa` is written in Python, a popular and user-friendly programming language. This makes it accessible to a broad scientific community familiar with Python scripting and scientific computing libraries.

CHAPTER 2

PROPOSED APPROACH

2.1 FRAMING THE PROBLEM MATHEMATICALLY

We propose a computational method to systematically optimize the design of two-dimensional (2D) photonic crystal slabs for controlling their absorbance, reflectance, and/or transmittance in the desired optical modes. This method is agnostic to the choice of materials and the targeted optical modes characterized by wavelength, polarization, and angle of incidence. Note that the process is applicable to tailoring absorbance, reflectance, and transmittance; however, we shall restrict this discussion to designing photodetectors by optimizing absorbance.

The proposed approach significantly improves the efficiency of photodetector design by eliminating the need for exhaustive searches in high-dimensional parameter spaces. The successful development of such a tool holds significant promise for various applications in optics, communication, and sensing, where tailored light absorbers are in high demand. Given the following information:

a. refractive indices of all materials denoted by Θ,

b. a set of geometrical shapes/patterns, and

c. a set of target wavelengths/optical modes $\Lambda = \lambda_1, \lambda_1, \ldots, \lambda_{|\Lambda|}$

we need to design a metasurface that will show a resonance, leading to high absorption at these optical modes. We will use a cylindrical hole with a circular type cross-section in silicon, i.e., a 2D periodic square array of air holes, for designing the photonic crystal slab (PCS). However, the same method can also be applied to any other cross-sectional pattern, like polygonal holes. We chose the circular-like cross-section as smooth and round holes are easier to fabricate than those with sharp corners/edges. Thus, the optimizable

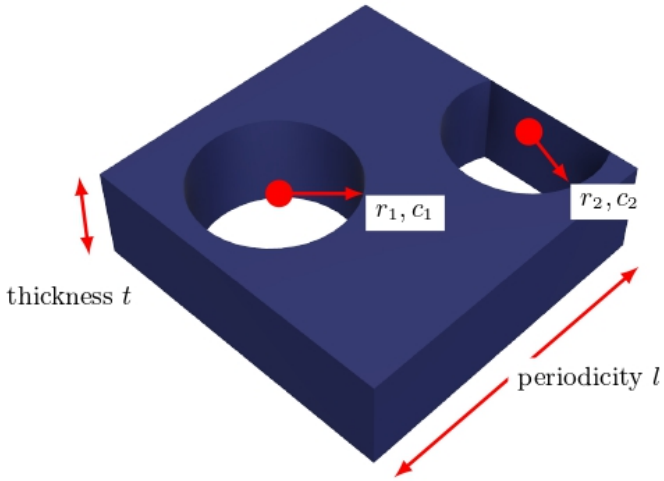


Figure 2.1: A schematic showing the metasurface unit cell and its optimizable geometric parameters – thickness t, periodicity l, hole radii (r_1, r_2), and hole centers (c_1, c_2) – a case with two holes $N_0 = 2$. The second hole is partial. When repeated in a plane, such a unit cell will create the metasurface. [14]

geometric parameters, as shown in Figure 2.1, are periodicity l of the square lattice, thickness of the metasurface/PCS t, centers of holes on the unit cell $\left(c_x, c_y\right)$ (Note that in this study, we allow partial holes too), and the hole radii r.

For optimization over multiple target wavelengths/optical modes, i.e., cases where $|\Lambda| \geq 2$, we allow for $N_0 > 1$ holes per unit cell to get more degrees of freedom and thus a higher dimensional search space to find the optimal structure. These holes may or may not be allowed to intersect, as required. Therefore, in general, we will have $3N_0 + 2$ optimization variables- center and radius of each hole, and periodicity and thickness of the PCS, each of which is bounded to fall in a feasible set per variable, Ω_i for $i = 1, 2, \ldots, 3N_0 + 2$, signifying box constraints. By specifying such a feasible set for each dimension, we can control and restrict the search space. Let $\mathbf{X}$ denote the vector of geometry variables. We have $\mathbf{X} \in \Omega \subset \mathbb{R}^{3N_0+2}$. Absorption would be a function of these geometric parameters and the material refractive indices. Formalizing this mathematically, we define the absorption function for the photonic crystal $\mathrm{A}{:}(\lambda, \Omega, \Theta) \rightarrow [0, 1]$ for wavelength $\lambda \in \Lambda$.

To find the best photodetector, we wish to find the optimal metasurface geometry $\mathbf{X}^*$ such that,

$$\mathbf{X}^* = \arg\max_{\mathbf{X}\in\Omega} \sum_{i=1}^{|\Lambda|} A(\lambda_i, \mathbf{X}, \Theta) \tag{2.1}$$

2.2 RIGOROUS COUPLED WAVE ANALYSIS

To simulate our metasurface designs, we employ Rigorous Coupled Wave Analysis (RCWA) [19]. This technique models the device in the frequency domain by discretizing it into a series of layers, each uniform in the direction normal to the layer. Within each layer, Maxwell's equations are solved analytically in the Fourier domain. RCWA is a well-established method favored for its high computational efficiency and speed. In this work, all simulations were performed using a Python implementation of RCWA known as GRWCA [17].

`grcwa` (autoGradable RCWA) [17] is a python implementation of RCWA for arbitrarily shaped photonic crystal slabs, supporting automatic differentiation with autograd. Each photonic crystal can have arbitrary dielectric profile on the 2D grids. Each layer is invariant along the transverse direction.

A typical simulation follows these steps -

a. Initialize an RCWA object with crystal dimensions and incident light frequency.

b. Add uniform layers, if any, to the crystal object with layer permittivity and thickness values.

c. Feed the epsilon profile of the patterned layer, create and add a grid of permittivities across the layer.

d. Define the polarization type and magnitude of the incident planewave.

e. Solve Maxwell's Equations for reflection and transmission using a function.

To plot a spectrum we discretize a continuous frequency range and loop these simulation

steps over it. For a pattern varying along z, like a frustum shaped hole or a pyramidal dielectric, for each frequency point we run another loop for permittivity grids along z that are necessary to define the crystal.

2.2.1 Units and Conventions

It is important to understand the units used for various physical quantities. As RCWA basically solves Maxwell's equations which are scale-invariant, normalized units are used. As a result, there is no intrinsic length unit. The base length unit, for example, can be determined while defining the unit cell dimensions. In our simulations we use 1 micron as the base length.

Speed of light, vacuum permittivity and permeability are normalized to unity. Hence, time is measured in units of length and frequency is measured in units of inverse length. A 700 nm length would be 0.7. For frequency in inverse length, a frequency of 1.1 corresponds to a wavelength of $(1/1.1) = 909$ nm [26].

2.2.2 Calculating Absorption

The method `Solve_FieldOnGrid(which_layer,z_offset)` on the `grcwa` object returns $\vec{E}$ and $\vec{H}$ at that depth in the layer (indices start with 1 for the top layer). We calculate $\vec{E} \times \vec{H}$ where $\vec{E}_x$ and $\vec{E}_y$ crossed and summed with $\vec{H}_y$ and $\vec{H}_x$ give the net energy flowing along z at the `z_offset` given as the function argument. Finding the difference between the energy at depths just under the top and just above the bottom gives the energy absorbed in that layer.

Note that this difference has to be normalized with total incident energy which in our case is $(\text{area}/2)$ (where *area* denotes the metasurface unit cell area) as incident light electric field amplitude is unity. It is also observed that absorption calculated in using energy is very close (error order = 10^{-6}) to $(1 - R - T)$ obtained from the simulation directly because no other layers absorb ($k = 0$ for dielectric silicon).

$$\vec{S}_z = \frac{1}{2 \cdot \text{area}} \text{Re}\left(E_x \cdot H_y - E_y \cdot H_x\right) \tag{2.2}$$

2.3 OPTIMIZATION: COVARIANCE MATRIX ADAPTATION - EVOLUTIONARY STRATEGY

The absorption function A, as defined above, is a complex, non-convex, non-separable, and rugged (multiple local optima) function, the direct gradient of which is not explicitly computable. Thus, we require a stochastic numerical optimization technique. As discussed in [7], to find global maxima in this case, the CMA-ES stochastic optimization algorithm proves to be a particularly well-suited tool due to the nature of the absorption function.

Essentially, the CMA-ES algorithm treats all the design parameters as random variables drawn from a multivariate Gaussian distribution. The core concept lies in iteratively evolving the mean and covariance matrix of this distribution to guide the search towards promising regions of the design space. During each iteration, CMA-ES updates the mean of the distribution to favor previously successful candidate solutions (those with better objective function values). Simultaneously, the covariance matrix is also adaptively updated to favour search directions that have led to past improvements. The algorithm is explained in Figure 2.2. Each metasurface parameter is subject to a box constraint specified by Ω_i introduced earlier, defining a continuous range of allowed values. We initialize the unit cell dimensions (length and thickness) within this search space and the hole radii with random values. Notably, in the case of a single hole ($N_0 = 1$), the hole center is initially positioned at the unit cell center, corresponding to complete holes within each unit cell (preferred solution). However, the optimization process can shift the hole center, potentially leading to partial holes depending on the specific problem and the optimal design. Altogether, this creates the mean vector m_0, an initial guess composed of initial values for all parameters.

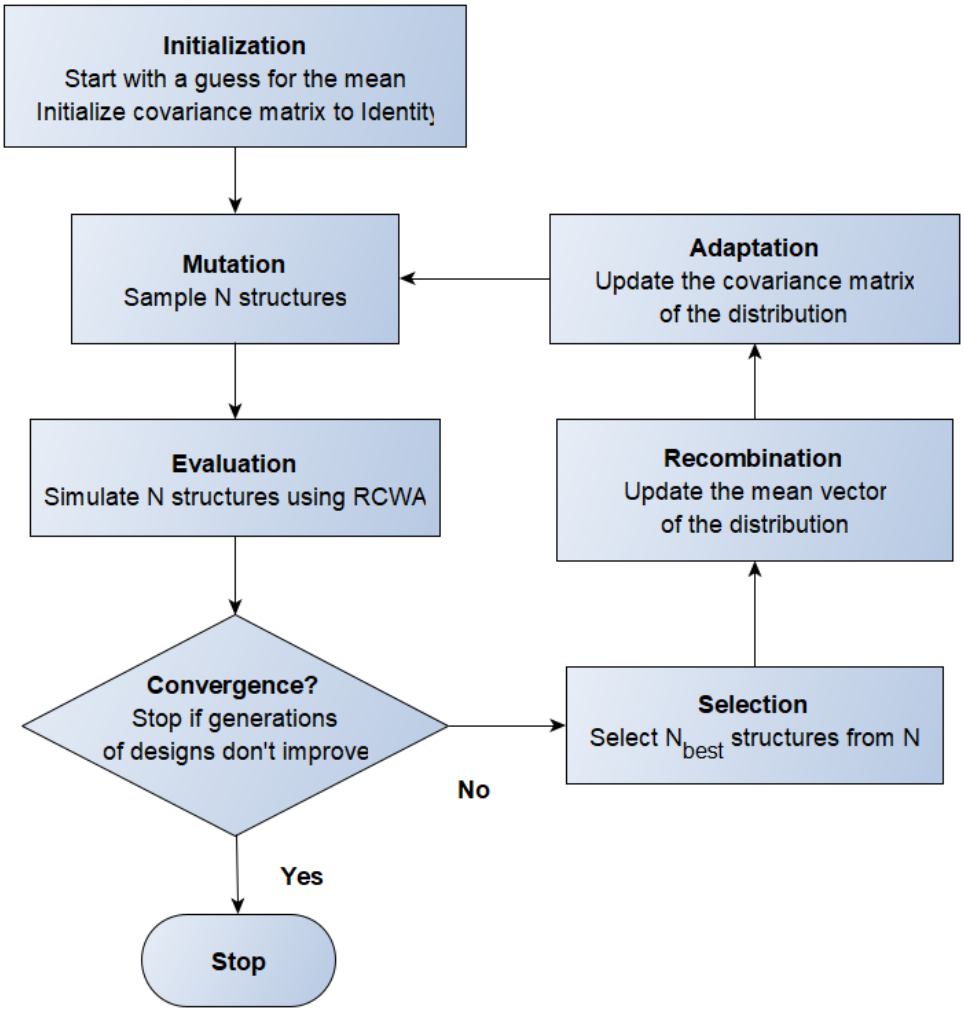


Figure 2.2: The CMA-ES Algorithm for photodetector design as explained in [7]. We start with an initial guess for the mean of all design variables. From a few samples drawn from the Gaussian and simulated through RCWA, the best few are selected to update the mean and covariance. The distribution keeps updating until no significant improvement is observed in sampled designs. [14]

In addition to m_0, the algorithm requires a hyperparameter σ_0 – the initial step-size. The CMA-ES algorithm dynamically adapts the step-size using cumulative step-size adaptation (CSA). This step-size controls the search 'reach' in each iteration, balancing the exploration of new regions in the initial stages with focused exploitation around promising areas as the search progresses. σ_0 and m_0 should be such that the optimum presumably lies within the initial cube $m_0 + 3\sigma_0\,(1, \ldots, 1)^T$. The complete algorithm and the role of all hyperparameters is explained in detail in [11]. It is suggested that given a search space $\Omega_i = [a, b]$, one may start with $\sigma_0 = 0.3\,(b - a)$. In our experiments, we typically used values in 0.5 to 1.0. In metasurface design, where all variables are on the nanoscale, the inherent uniformity in units (nanometers) eliminates the need for additional scaling, simplifying parameter selection. Overall, the hyperparameters and initializations required by the algorithm are m_0, σ_0, Ω_i for each variable, and the number of holes per unit cell N_0. However, an informative guess for m_0 would often lead to the best results faster than a random m_0. The convergence time depends on this starting point, too. For instance, if the initial hole radius is too small or too large for a small unit cell, finding the solution may be difficult as the cavity does not play its role. Due to the stochastic nature of CMA-ES and the random start, multiple optimization runs (5-10) might be necessary to achieve the best design.

The algorithm stops on a few termination criteria related to numerical stability described in [11]. The ones most relevant for this application are stagnation and function tolerance. The stagnation criterion checks if the designs have not improved over several generations by comparing the median of the last few values with the first. For a minimization problem, the tolerance criterion halts the process if the function value falls below a threshold, typically 10^{-10}. Exploiting the scale invariance of Maxwell's equations, we normalize all lengths used within the optimization process to a base unit length of 1 micron. This allows us to treat them as dimensionless quantities that can be processed for numerical optimization. For a more detailed explanation of the CMA-ES algorithm, we refer the reader to [11], [12] and [13]. In the following section, we shall describe some of our

results that demonstrate the application of this method using a three-layered structure for demonstrating the algorithm for photodetector design - Black Phosphorus [36] as the active 2D material for light absorption, Silicon (n=3.5) as the material for the photonic crystal slab/metasurface, and a reflector at the bottom, as shown in Figure 2.1.

CHAPTER 3

RESULTS AND DISCUSSION

The following simulations were performed on a single core of the Dual Intel Xeon Gold 6248, 20-core 2.5 GHz CPU Node at the High-Performance Computing Facility, IIT Madras.

3.1 SINGLE RESONANCE WITH METALLIC- AND BRAGG-REFLECTORS

Efficient light absorption at specific wavelengths in the infrared range is crucial for various applications in telecommunications and optoelectronics [24]. The wavelengths of 1.55 μm and 2.1 μm hold particular importance. In optical communication systems, 1.55 μm coincides with the minimum attenuation window in silica fibers, making it the dominant wavelength for long-distance data transmission [9]. Efficient absorbers at 1.55 μm are essential for developing critical components like optical modulators and terminators within these systems. Photodetectors operating at a wavelength of 2.1 μm are crucial for applications such as imaging [31], sensing [6], and high-speed communication [1].

Black phosphorus exhibits a highly tuneable bandgap, ranging from 0.3 to 2 eV [32]. This tunability allows for the customization of the bandgap to match specific infrared wavelengths, enabling efficient light absorption and detection. This makes it a good choice of an active absorber for infrared photodetectors since the bandgap allows generated electron-hole pairs to be separated to generate photocurrent. We use seven layers (4 nm thick) of Black Phosphorus using the experimental extinction coefficient data from [36].

To design an absorber for 1.55 μm, we used a three-layered structure illustrated in Figure 3.1 - Black Phosphorus, silicon metasurface, and metallic reflector. We use a 2D periodic

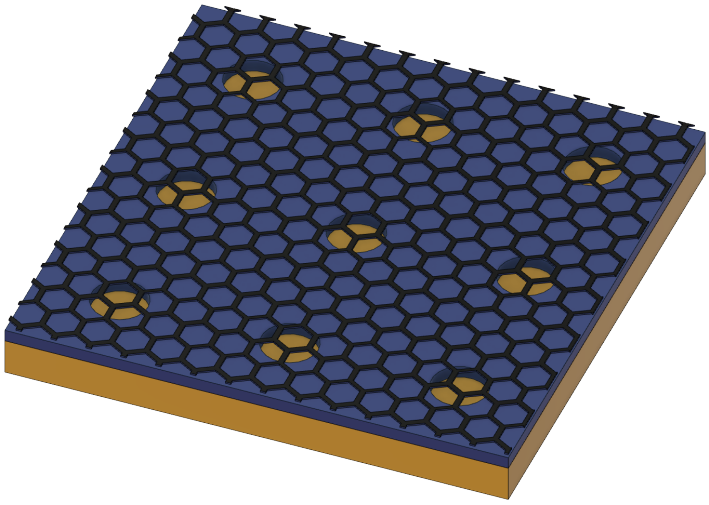
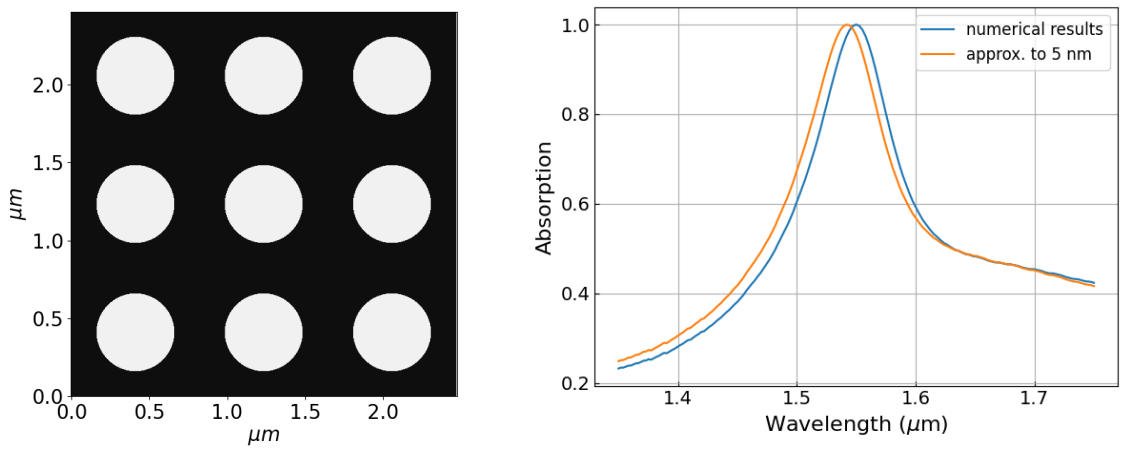


Figure 3.1: This schematic represents the three-layered structure used to design a photodetector for 1.55 μm. The top mesh represents the 4 nm thick 7-layered Black Phosphorus, the blue layer with holes is the patterned Silicon metasurface (see Figure 3.2), and the bottom yellow layer represents a lossless metallic mirror. [14]

metasurface with one hole per unit cell ($N_0 = 1$), thus its optimizable parameters are periodicity, thickness, hole radius, and hole center. Figure 3.2(a) shows the optimized metasurface design obtained from CMA-ES in 8 minutes of runtime. The absorption in Black Phosphorus is plotted against wavelength (Fig. 3.2(b)). One can observe a total absorption of 1.55 μm and broadband absorption of over 60% in a 100 nm range. Code used for this experiment is presented in Appendix B.

In another experiment, to design a photodetector for 2.1 μm, we replaced the metallic reflector with a DBR mirror of alternating SiO2 and Sb2S3 layers (Figure 3.3). Since a DBR is not reflective to all wavelengths, this experiment aimed to test if the algorithm remained effective without a perfect reflector. In this case too, the metasurface variables remain the same as before. Figure 3.4(a) shows the optimized metasurface design generated in just 20 minutes of runtime. Absorption in BP is plotted alongside. This time, we see a narrow peak at 2.1 μm with unintended resonances at a few nearby wavelengths.



(a) (b)

Figure 3.2: (a) The Silicon patterned metasurface design for a photodetector absorbing at 1550 nm. The holes are of radius 253 nm, the periodicity is 823 nm, and the metasurface is 131 nm thick. (b) Absorption in Black P. The algorithm provides real numbers for each design variable, plotted as numerical results. To check robustness, we round off these numerical results of lengths to the nearest 5 nm to consider fabrication accuracy, as shown in the orange curve. [14]

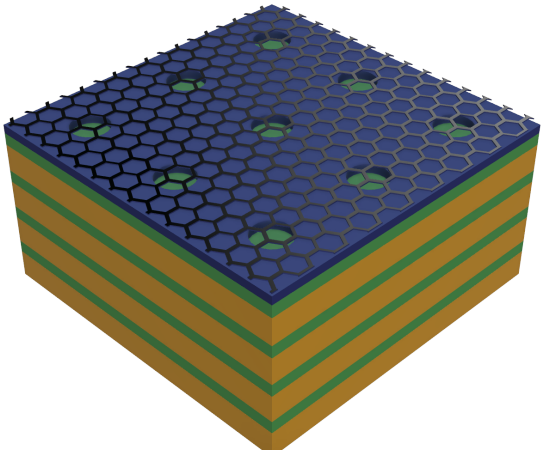
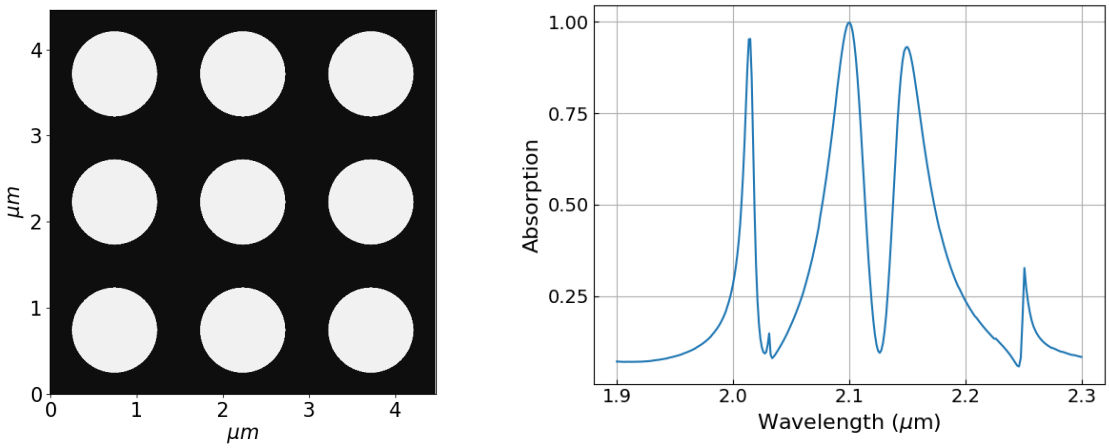


Figure 3.3: Schematic of the system used to design a photodetector for 2.1 μm. The top mesh represents Black Phosphorus; the blue layer with holes is the patterned silicon metasurface. The metal layer from the previous design (Fig. 3) is replaced with a DBR mirror of alternating SiO_2 green and Sb_2S_3 orange layers. [14]

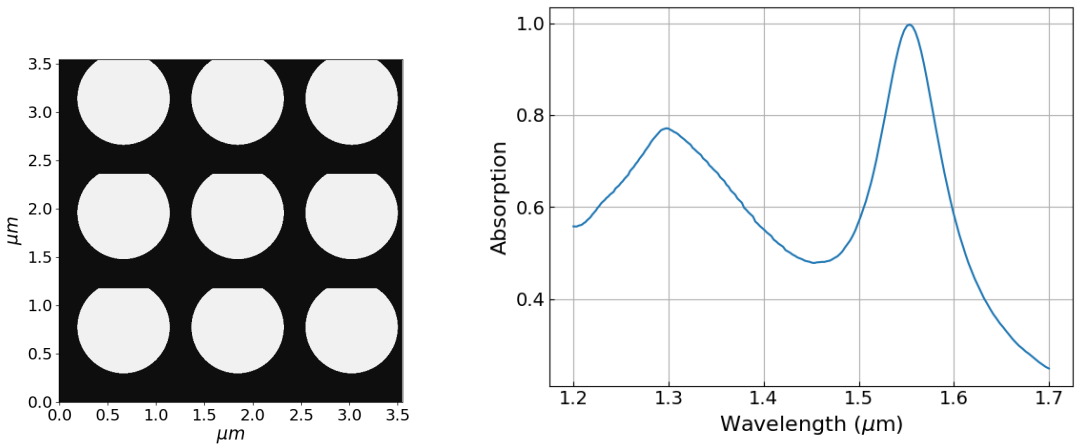


(a) (b)

Figure 3.4: (a) The Silicon patterned metasurface design for a photodetector absorbing at 2.1 μm. The holes are of radius 499 nm, the periodicity is 1.48 μm, and the metasurface is 113 nm thick. (b) Absorption in Black P, resonance designed at the target wavelength. Note that two other peaks appearing in the neighborhood are unintentional and naturally occurring. [14]

3.2 RESONANCE AT TWO WAVELENGTHS

As discussed in the introduction section, devices showing double resonance are desirable for a variety of applications. In particular, efficient operation at both 1.3 μm and 1.55 μm is highly desirable for seamless channel switching and simplified system design in optical communications [3]. Using the same three-layered structure (BP, silicon, metal) as in Fig. 3.1, we demonstrate how the technique can be applied to design double resonance. We kept $N_0 = 1$ to enable faster search in a lower dimensional space. However, one can also experiment with two holes per unit cell at the cost of computation time. The metasurface obtained through optimization is shown in Figure 3.5(a), and (b) shows the absorption resonances in BP at both wavelengths of interest. Note how the holes are sliced off from the top – this happens when the center shifts such that the hole is partially out of the unit cell. This creates an irregular cavity, compromising on smoothness and symmetry as a trade-off for a more demanding problem.



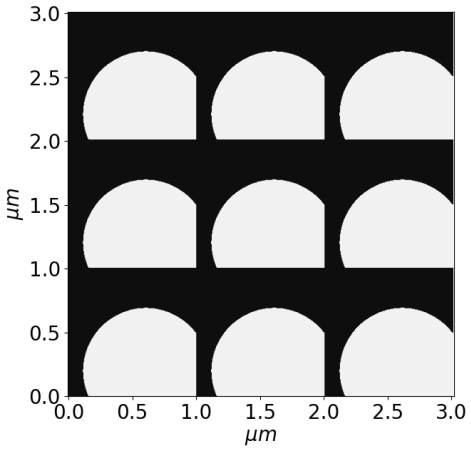
(a) Metasurface

(b) Absorption Spectrum

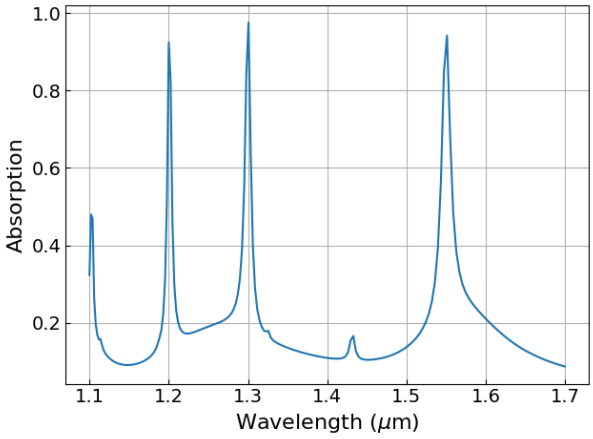
Figure 3.5: (a) The Silicon patterned metasurface design for a photodetector absorbing at 1.3 and 1.55 μm. As the number of wavelengths to design a resonance at increases, more degrees of freedom may be required. We allow the hole to move on the unit cell to enable that. During the CMA procedure, the hole may partially leave the unit cell, giving a metasurface as shown. The hole symmetry, thus, becomes a trade-off. The holes are of radius 482 nm, the periodicity is 1.18 μm, and the metasurface is 109 nm thick. The algorithm terminated in 53 minutes. (b) Absorption in Black P with resonance at both target wavelengths. [14]

The same double resonance inverse design was performed by replacing Black P with graphene and BAs in separate experiments just to test the material-agnostic nature of the complete process. Note that graphene is a zero bandgap material, rendering it practically useless as electron-hole pairs will recombine as soon as they are created. However, this inverse design procedure simply aims to create resonances in metasurfaces without practical considerations.

Figures 3.6 and 3.7 show the double-resonance result with graphene and BAs, respectively. Both these results re-emphasize that the process is material-agnostic. We use a single layer of graphene 0.34 nm thick. For BAs too, we use a 0.34 nm thick layer, for this experiment.

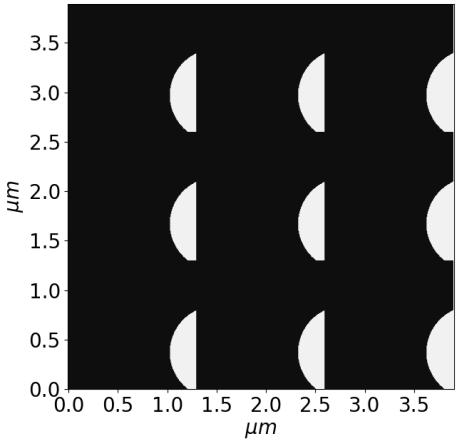


(a) Metasurface for graphene

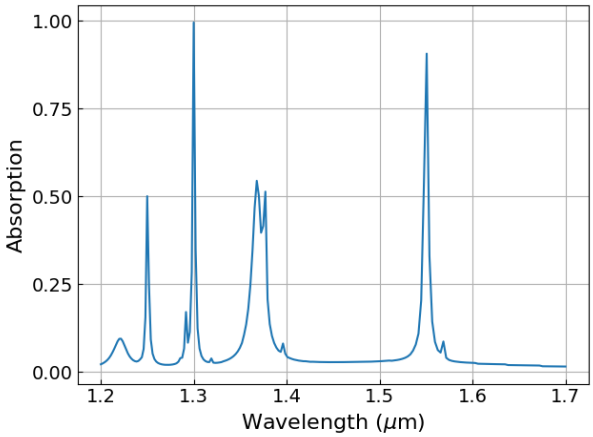


(b) Absorption Spectrum

Figure 3.6: Double resonance (1.3 and 1.55 μm) with graphene as the absorber (graphene monolayer, 0.34 nm) instead of Black P. (a) Metasurface for the graphene absorber and (b) shows the absorption spectrum. Both absorption peaks are $> 96\%$. Note that the extra peak at 1.2 μm is unintentional.



(a) Metasurface for BAs

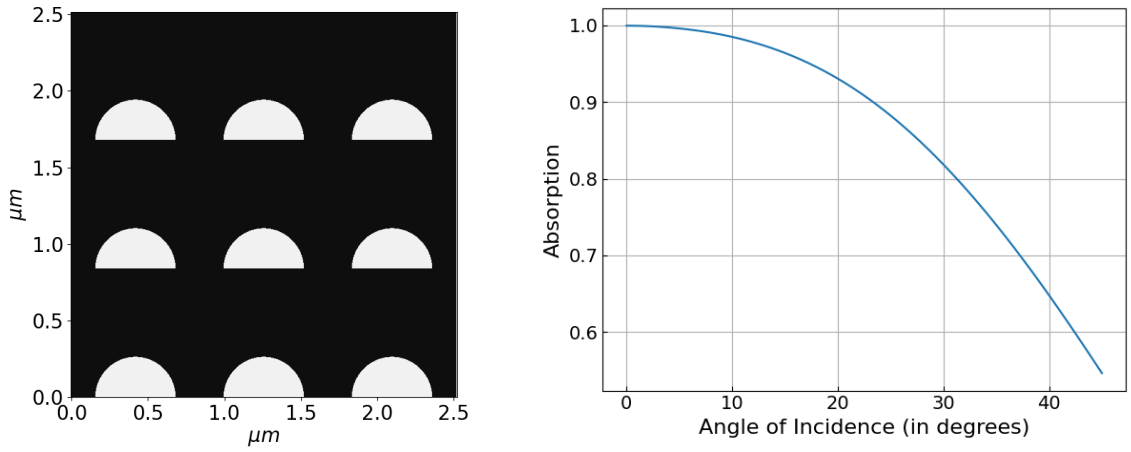


(b) Absorption Spectrum

Figure 3.7: Double resonance (1.3 and 1.55 μm) with BAs as the absorber (0.34 nm) instead of Black P. (a) Metasurface for the BAs absorber and (b) shows the absorption spectrum. Both absorption peaks are $> 96\%$.

3.3 WIDE-INCIDENT ANGLE ABSORPTION

So far, in all the simulations, we have simulated normal incidence. However, often for detectors in sensing, light may not always have a normal incidence. A device might be required to show high absorption in a cone of incidence. Here, we show that the same inverse design optimization procedure can be used not only to design and improve a spectral response but also an angular response. Such an experiment, performed for 1.55 μm, is illustrated in Figure 8. Using the three-layered structure as before, we inverse design a metasurface that will enable wide-incident angle absorption in Black P. To do so, we sample a few points in zero degrees to the required incident angle range and try to maximize absorption at each of these discrete wavelength samples. We observe that the designed structure exhibits at least 80% absorption in a 30-degree cone of incidence.



(a) Metasurface

(b) Absorption Spectrum

Figure 3.8: (a) The Silicon patterned metasurface design for a photodetector absorbing at 1.55 μm. The partial and almost semi-circular holes have a radius of 266 nm and a periodicity of 838 nm, and the metasurface is 131 nm thick. This experiment took around 90 minutes of runtime to complete. (b) Absorption in Black P is plotted as a function of the angle of incidence. We designed the device to absorb up to 30° of incident light. [14]

CHAPTER 4

CONCLUSION

As seen in the case of double resonance and wide-angle absorption, we traded off hole symmetry to design complex requirements. When more resonances are desired, such structures may be required. Furthermore, one may tune hyperparameters like N_0 to have two or more such holes per unit cell. This could help simulate a triangular crystal lattice. However, like any practical optimization problem, there will be a natural limit to the number of different resonances that can be designed. This heavily depends on the choice of materials and pattern type from a physics perspective. Nevertheless, limitations can be tackled by experimenting with hyperparameters, patterns, and the objective function used to frame the optimization problem.

Multiple avenues exist to improve this inverse design procedure through mathematics and physics. One can experiment with various metasurface layers by optimizing them together or sequentially. Furthermore, a staircase approximation is required to simulate conical holes or curved cavities as RCWA only deals with layers invariant along the z-axis. However, this can quickly become expensive. Mathematically, in equation (1), we can explore possibilities by changing the summation to a weighted sum or a product of individual objective functions. Interestingly, it might also be possible to draw inferences from the photonic band diagrams of such structures to gain deeper insights into how the shape of the cavity influences optical response. This analysis might be instrumental when an excellent initial guess is required; the algorithm can further improve upon that.

Overall, combining CMA and RCWA provides a general assistive method to design resonances in photonic crystals at desired optical modes.

4.1 RESEARCH OUTCOMES

As a result of this study, the following goals have been achieved –

- Development of a generalized computational framework to inverse design 2D metasurfaces. This includes ideation, writing, and organization of the codebase into clean importable modules that anybody can easily understand and use. Code used to experiment with Black Phosphorus for a 1.55 μm absorber is copied in Appendix B.

- The code will be hosted as an open-source repository on GitHub after the paper is submitted successfully. Until then, all figures, plot data, and code will be accessible through a drive link.

- Most importantly, this work is being written as a research paper to be **submitted to ACS Photonics** (tentatively / or Photonics Express) [14].

APPENDIX A

MATERIALS

Throughout this research, we have talked about three absorber 2D materials - graphene, BAs, and Black Phosphorus. For silicon, $n = 3.5$. For graphene in the infrared regime, we use $n = 3 + j5.446\lambda/3\ \mu\text{m}^{-1}$.

n and k for BAs are plotted in figure A.1.

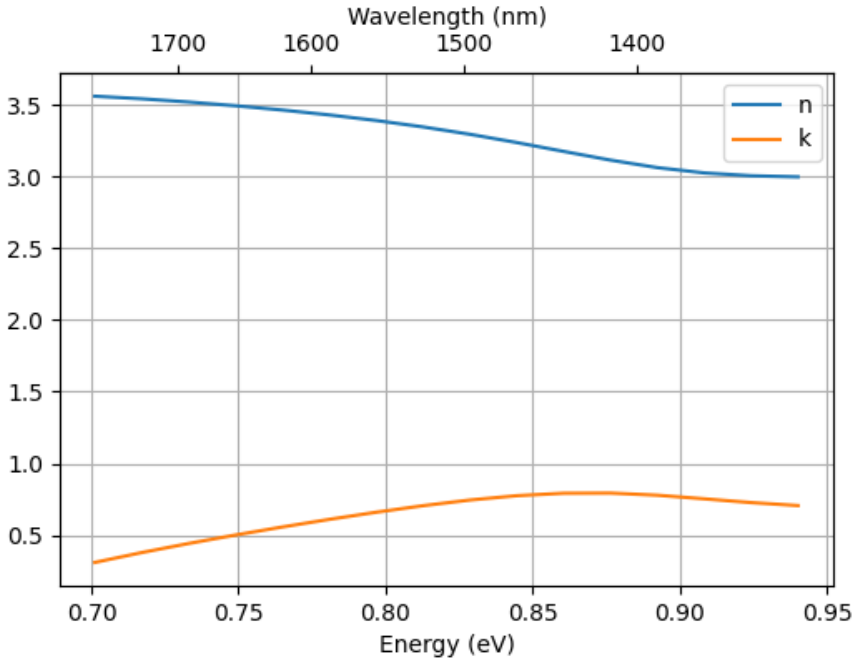


Figure A.1: Complex refractive index data for BAs.

For Black Phosphorus, k was obtained from extinction data from [36]. We use $n = 3.5$. See figure A.2.

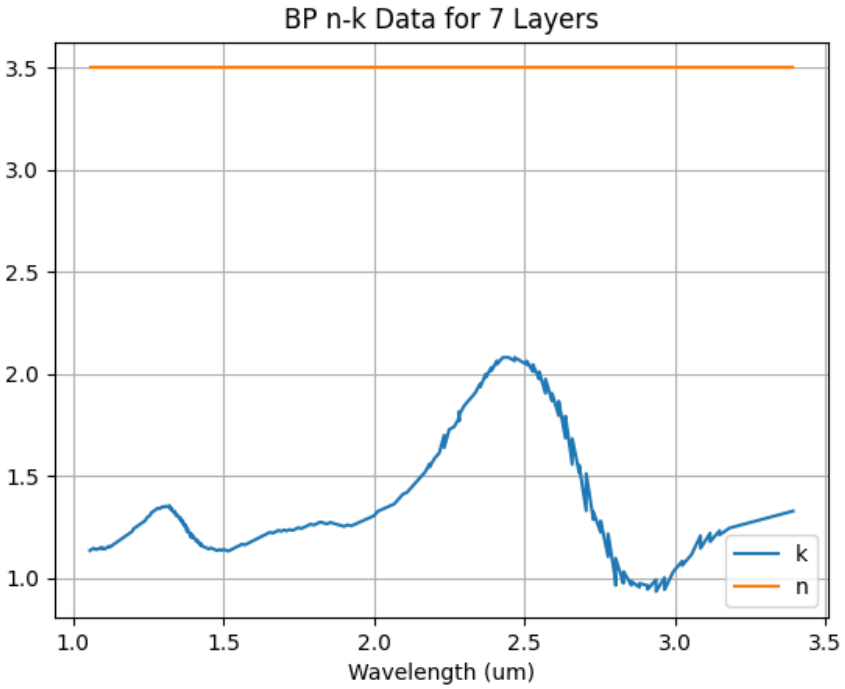


Figure A.2: Complex refractive index data for Black Phosphorus.

APPENDIX B

CODE

In this section, we present the code used in the simulation and optimization of using Black Phosphorus for designing an absorber for 1550 nm (see figure 3.2).

First, we have the file `materialspy` which includes all material refractive indices.

```python
from find_BPnk import get_BP_nk


def get_graphene_nk(f):
    """
    Calculates the complex refractive index of graphene for a given frequency.

    Args:
        f: Frequency in micrometers^-1.

    Returns:
        complex: The complex refractive index (n + ik).
    """

    ref_ind = 3 + 1j * (5.446 / (3 * f))
    return ref_ind


def get_BAs_nk(f):
    """
    Approximates the complex refractive index of BAs for a given frequency.

    Args:
        f: Frequency in micrometers^-1.

    Returns:
        complex: The complex refractive index (n + ik).
    """

    n = 3.1
    k = (0.25 + (5 / 8) * (2.5 - 1 / f)) * 2
    return n + k * 1j
```

```

# materials.py
refractive_indices = {
    "Silicon": 3.5,
    "Air": 1,
    "MoO2": 1.6 + 0.5j,
    "Graphene": get_graphene_nk,
    "PEC": (-1e8) ** 0.5,
    "TiO2": (2.5263),
    "BAs": get_BAs_nk,
    "BP": get_BP_nk,
    "Sb2S3": 3.2,
    "SiO2": 1.44,
}
```

Next, `config.py` sets up all hyperparameters and selects materials from available options in `materials.py`.

```
import numpy as np
from materials import refractive_indices

# specify the target wavelength (for single resonance)
CENTER_WAV = 1.55

# width around the center wavelength (for plotting)
WIDTH = 0.2

# thickness of the absorber layer in micron
absorber_thickness = 0.004  # 4 nm

# PEC / Metal thickness
pec_thickness = 0.1

# choose the materials from materials.py
absorber_material = "BP"
dielectric_material = "Silicon"
cavity_material = "Air"

# start to end wavelengths for plotting
start_wavelength = CENTER_WAV - WIDTH
end_wavelength = CENTER_WAV + WIDTH
```

```

# array of target wavelengths
f_sampled = np.array([1 / CENTER_WAV])

# number of holes per unit cell
N_HOLES = 1

# initial step-size for CMA-ES
SIGMA = 0.7

# grid size
Nx, Ny = 200, 200

# dielectric permitivitty
ep1_diel = refractive_indices[dielectric_material] ** 2

# cavity permittivity (generally air)
epbkg = refractive_indices[cavity_material]

# PEC / metal layer
epN = refractive_indices["PEC"] ** 2
```

All functions that are required to run the simulation and plot results are included in `helper.py`.

```
import numpy as np
import matplotlib.pyplot as plt
import grcwa
import seaborn as sns
from materials import refractive_indices
from config import absorber_material, absorber_thickness, pec_thickness


def get_layer_absorption(layer_number, layer_thickness, area, Nx, Ny,
rcwa_obj):
    """
    Calculates the fractional absorption of a specific layer in a structure.

    Args:
        layer_number: Layer index (starts from 1).
        layer_thickness: Thickness of the layer.
        area: Total area of the structure.
```

```
        Nx, Ny: Number of grid points in x and y directions.
        rcwa_obj: The grcwa object representing the structure.

    Returns:
        float: The fractional absorption of the specified layer.
    """

    tol = 1e-9
    E_top, H_top = rcwa_obj.Solve_FieldOnGrid(layer_number, tol *
    layer_thickness)
    E_bottom, H_bottom = rcwa_obj.Solve_FieldOnGrid(
        layer_number, (1 - tol) *
        layer_thickness
    )
    E_top = np.array(E_top)
    E_bottom = np.array(E_bottom)
    H_top = np.array(H_top)
    H_bottom = np.array(H_bottom)

    dA = area / (Nx * Ny)

    S_top = (
        np.sum(np.real(-E_top[1] * np.conj(H_top[0]) + E_top[0] *
        np.conj(H_top[1])))
        * dA
        / 2
    )

    S_bottom = (
        np.sum(
            np.real(
                -E_bottom[1] * np.conj(H_bottom[0]) + E_bottom[0] *
                np.conj(H_bottom[1])
            )
        )
        * dA
        / 2
    )

    absorption_fraction = (S_top - S_bottom) / (area / 2)

    return absorption_fraction


def get_pattern_epgrid(radii, centers, L1, L2, ep_dielectric, ep_hole, Nx, Ny):
    """
```

```
    Generates a permittivity grid representing a patterned hole structure.

    Args:
        radii: List of hole radii.
        centers: List of hole centers as tuples (x, y).
        L1, L2: Lengths of the unit cell in x and y directions.
        ep_dielectric: Permittivity of the dielectric material.
        ep_hole: Permittivity of the hole material.
        Nx, Ny: Number of grid points in x and y directions.

    Returns:
        ndarray: The permittivity grid for the patterned structure.
    """

    x0 = np.linspace(0, L1[0], Nx)
    y0 = np.linspace(0, L2[1], Ny)
    x, y = np.meshgrid(x0, y0, indexing="ij")

    epgrid = np.ones((Nx, Ny)) * ep_dielectric
    for i in range(len(radii)):
        ind = (x - centers[i][0]) ** 2 + (y - centers[i][1]) ** 2 <
        radii[i] ** 2
        epgrid[ind] = ep_hole

    return epgrid


def get_random_params(L_MIN, L_MAX, H_MIN, H_MAX, C_MIN, C_MAX, R_MIN, R_MAX,
N_HOLES):
    """
    Generates random parameters for the structure.

    Args:
        L_MIN, L_MAX, H_MIN, H_MAX, C_MIN, C_MAX, R_MIN, R_MAX: Minimum and
        maximum values for various parameters.
        N_HOLES: Number of holes in the structure.

    Returns:
        ndarray: Array containing the randomly generated parameters.
    """

    l = np.random.uniform(L_MIN, L_MAX)
    h = np.random.uniform(H_MIN, H_MAX)

    # Initialize hole centers at the center
    c_x = [l / 2 for i in range(N_HOLES)]
```

```
    c_y = [l / 2 for i in range(N_HOLES)]

    r = [np.random.uniform(R_MIN, R_MAX) for i in range(N_HOLES)]

    # initial parameter vector
    x = np.array(([l] + [h] + c_x + c_y + r))
    return x


def plot_metasurface(xopt, N_HOLES, ep1_diel, epbkg, Nx, Ny, tiles=(3, 3)):
  """
  Plots a repeating array of the unit cell structure.

  Args:
      xopt: Optimized parameters from the optimization process.
      N_HOLES: Number of holes in the structure.
      ep1_diel: Permittivity of the dielectric material.
      epbkg: Permittivity of the background material.
      Nx, Ny: Number of grid points in x and y directions for a single unit
      cell.
      tiles: Tuple specifying the number of unit cells to tile in x and y
      directions (default: (3, 3)).

  Returns:
      matplotlib.figure.Figure: The plot of the repeating unit cell structure.
  """

  # Extract hole radii and center coordinates
  radii = xopt[-N_HOLES:]
  c_x = xopt[2: 2 + N_HOLES]
  c_y = xopt[2 + N_HOLES: 2 + 2 * N_HOLES]
  centers = list(zip(c_x, c_y))  # Combine x and y center coordinates

  # Unit cell dimensions
  L1 = [xopt[0], 0]
  L2 = [0, xopt[0]]

  # Calculate permittivity grid for a single unit cell
  epgrid = get_pattern_epgrid(radii, centers, L1, L2, ep1_diel, epbkg, Nx, Ny)

  # Create a larger grid for tiling the unit cell
  x_tiles, y_tiles = tiles
  x_ = np.linspace(0, xopt[0] * x_tiles, Nx * x_tiles)
  y_ = np.linspace(0, xopt[0] * y_tiles, Ny * y_tiles)
  x, y = np.meshgrid(x_, y_, indexing="ij")

```

```
    # Tile the permittivity grid to match the larger grid
    plane_epgrid = np.tile(epgrid, (x_tiles, y_tiles))

    # Create the plot
    fig, ax = plt.subplots(figsize=(7, 7))
    ax.contourf(x, y, plane_epgrid, cmap="binary")
    ax.set_aspect("equal")
    ax.set_xlim(0, max(x_))
    # ax.set_xticks([0.0 + i * xopt[0] for i in range(x_tiles + 1)], fontsize=20)
    # ax.set_yticks([0.0 + i * xopt[0] for i in range(y_tiles + 1)], fontsize=20)
    ax.set_xlabel(r"$\mu m$", fontsize=20)
    ax.set_ylabel(r"$\mu m$", fontsize=20)
    # Optional title: ax.set_title("Metasurface")

    return fig


def get_epgrid_nk(f, epgrid2, epgrid3, Nx, Ny):
    """
    Constructs the permittivity grid considering varying refractive indices.
    NOTE:
    - This function is specific to the DBR case.
    - nk in the function name implies that refractive index is a function of
    wavelength
    - the same must be specified in materials.py and config.py

    Args:
        f: Frequency.
        epgrid2, epgrid3, epgrid4: Permittivity grids for different layers.
        Nx, Ny: Number of grid points in x and y directions.

    Returns:
        ndarray: The combined permittivity grid with nk dependence.
    """

    e_top_f = refractive_indices[absorber_material](f) ** 2
    epgrid_1 = np.ones((Nx, Ny)) * e_top_f
    epgrid = np.concatenate((epgrid_1.flatten(), epgrid2.flatten(),
    epgrid3.flatten()))
    return epgrid


def get_rt_nk(L1, L2, pthick, f, epgrid2, epgrid3, Nx, Ny):
  """
  Calculates reflection and transmission coefficients using GRCWA.

```

```
    Args:
        L1, L2: Lengths of the unit cell in x and y directions.
        pthick: List of thicknesses for each layer.
        f: Frequency.
        epgrid2, epgrid3, epgrid4: Permittivity grids for different layers.
        Nx, Ny: Number of grid points in x and y directions.

    Returns:
        tuple: A tuple containing reflection coefficient (R),
        transmission coefficient (T),
               and the GRCWA object for further analysis (optional).
    """

    # Truncation order (adjust as needed)
    nG = 101
    theta = 0.0  # Angle of incidence (normal incidence)
    phi = 0.0  # No in-plane polarization

    # Create a GRCWA object
    obj = grcwa.obj(nG, L1, L2, f, theta, phi, verbose=0)

    # Add vacuum layer
    obj.Add_LayerUniform(0.1, 1)

    # Add patterned layer(s) based on thickness list
    for layer_thickness in pthick:
      obj.Add_LayerGrid(layer_thickness, Nx, Ny)

    # Add another vacuum layer
    obj.Add_LayerUniform(0.1, 1)

    # Initialize the setup
    obj.Init_Setup()

    # Set permittivity grids for each layer
    obj.GridLayer_geteps(get_epgrid_nk(f, epgrid2, epgrid3, Nx, Ny))

    # Set plane wave excitation (adjust polarization as needed)
    planewave = {"p_amp": 0, "s_amp": 1, "p_phase": 0, "s_phase": 0}
    obj.MakeExcitationPlanewave(planewave["p_amp"], planewave["p_phase"],
    planewave["s_amp"], planewave["s_phase"], order=0)

    # Solve for reflection and transmission coefficients
    R, T = obj.RT_Solve(normalize=1)

    # Return reflection, transmission, and optionally the GRCWA object
```

```
  return R, T, obj


def save_spectrum_data(
    xopt,
    spectrum,
    N_HOLES,
    ep1_diel,
    epbkg,
    epN,
    Nx,
    Ny,
    spectrum_filename="spectrum.txt",
    abs_filename="absorption.txt",
):
  """
  Calculates and saves absorption spectra data.

  Args:
      xopt: Optimized parameters from the optimization process.
      spectrum: List of frequencies for which to calculate absorption.
      N_HOLES: Number of holes in the structure.
      ep1_diel: Permittivity of the dielectric material.
      epbkg: Permittivity of the background material.
      epSbS: Permittivity of the antimony sulfide (SbS) material.
      epSiO: Permittivity of the silicon dioxide (SiO2) material.
      thick_sbs: Thickness of the SbS layer.
      thick_sio: Thickness of the SiO2 layer.
      Nx, Ny: Number of grid points in x and y directions.
      spectrum_filename: Filename to save the spectrum data.
      abs_filename: Filename to save the absorption data.
  """

  L1 = [xopt[0], 0]
  L2 = [0, xopt[0]]
  pthick = [absorber_thickness, xopt[1], pec_thickness]

  radii = xopt[-N_HOLES:]  # Radii of cylinders
  c_x = xopt[2: 2 + N_HOLES]
  c_y = xopt[2 + N_HOLES: 2 + 2 * N_HOLES]
  centers = [(cx, cy) for cx in c_x for cy in c_y]

  epgrid2 = get_pattern_epgrid(radii, centers, L1, L2, ep1_diel, epbkg, Nx, Ny)
  epgrid3 = np.ones((Nx, Ny)) * epN

  absorptivity = []
```

```
    for f in spectrum:
      R, T, obj = get_rt_nk(L1, L2, pthick, f, epgrid2, epgrid3, Nx, Ny)
      a = get_layer_absorption(1, pthick[0], (L1[0] ** 2) * 1e-12, Nx, Ny, obj)
      absorptivity.append(a)

    absorptivity = np.array(absorptivity)

    # Save spectrum and absorption data
    np.savetxt(spectrum_filename, spectrum)
    np.savetxt(abs_filename, absorptivity)

    return


def plot_absorption_spectrum(spectrum_filename="spectrum.txt",
abs_filename="absorption.txt", save_fig=True):
    """
    Plots the absorption spectrum data from saved files.

    Args:
        spectrum_filename: Filename containing the spectrum data.
        abs_filename: Filename containing the absorption data.
        save_fig: Boolean flag to save the plot as an image (default: True).
    """

    spectrum = np.loadtxt(spectrum_filename)
    absorption = np.loadtxt(abs_filename)

    fig, ax = plt.subplots()
    ax = sns.lineplot(x=1 / spectrum, y=absorption)
    ax.set_xlabel(r"Wavelength ($\mu$m)", fontsize=16)
    ax.set_ylabel("Absorption", fontsize=16)
    ax.grid(True)
    ax.tick_params(which="both", direction="in")

    plt.locator_params(axis="y", nbins=5)
    plt.locator_params(axis="x", nbins=8)
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
    plt.tight_layout()

    if save_fig:
      plt.savefig("abs_plot.png")
```

```
  return fig, ax


def load_spectrum_data(spectrum_filename="spectrum.txt",
abs_filename="absorption.txt"):
  """
  Loads spectrum and absorption data from text files.

  Args:
      spectrum_filename: Name of the text file containing spectrum data.
      abs_filename: Name of the text file containing absorption data.

  Returns:
      tuple: A tuple containing two NumPy arrays -
      spectrum data and absorption data.
  """

  # Load spectrum data
  with open(spectrum_filename, "r") as f:
    spectrum_data = np.genfromtxt(f, delimiter=",")
    # Assuming comma-separated data

  # Load absorption data
  with open(abs_filename, "r") as f:
    absorption_data = np.genfromtxt(f, delimiter=",")
    # Assuming comma-separated data

  return spectrum_data, absorption_data
```

The optimization task is performed in `main.py`.

```
import numpy as np
import cma
from materials import refractive_indices
from config import *
from helper import *


# the objective function
def objective_function(params, Nx=200, Ny=200):
        """
        Objective function to minimize during optimization.

```

```
        Args:
            params: Array containing optimization parameters
            (l, h, center coordinates, radii).
                Nx, Ny: Grid points in x and y directions.

        Returns:
                float: Sum of reflection and transmission
            coefficients for all frequencies.
        """

        l = params[0]
        h = params[1]
        c_x = params[2 : 2 + N_HOLES]
        c_y = params[2 + N_HOLES : 2 + 2 * N_HOLES]
        centers = [(cx, cy) for cx in c_x for cy in c_y]

        L1 = [l, 0]
        L2 = [0, l]
        pthick = [absorber_thickness, h, pec_thickness]

        radii = params[-N_HOLES:]

        epgrid2 = get_pattern_epgrid(radii, centers, L1, L2, ep1_diel, epbkg,
Nx, Ny)
        epgrid3 = np.ones((Nx, Ny)) * epN

        R, T = [], []
        for f in f_sampled:
                R1, T1, _ = get_rt_nk(L1, L2, pthick, f, epgrid2, epgrid3,
  Nx, Ny)
                R.append(R1)
                T.append(T1)

        # Objective function: minimize sum of reflection and transmission
        return sum(R) + sum(T)


# Define optimization bounds
L_MIN = 0.5  # Minimum unit cell length (um)
L_MAX = 1.5  # Maximum unit cell length (um)
H_MIN = 0.1  # Minimum hole height (um)
H_MAX = 0.25  # Maximum hole height (um)
C_MIN = 0.0   # Minimum center coordinate (relative to unit cell)
C_MAX = 1.5   # Maximum center coordinate (relative to unit cell)
R_MIN = 0.05  # Minimum hole radius (um)
R_MAX = 0.5   # Maximum hole radius (um)
```

```

lower_bounds = [L_MIN, H_MIN] + [C_MIN] * 2 * N_HOLES + [R_MIN] * N_HOLES
upper_bounds = [L_MAX, H_MAX] + [C_MAX] * 2 * N_HOLES + [R_MAX] * N_HOLES


# Define initial guess for optimization parameters
x = get_random_params(
    L_MIN, L_MAX, H_MIN, H_MAX, C_MIN, C_MAX, R_MIN, R_MAX, N_HOLES
)

xopt, es = cma.fmin2(
    objective_function, x, SIGMA,
    options={"bounds": [lower_bounds, upper_bounds]}
)

# save the result to a file
with open("result.txt", "w") as f:
    f.write(str(es.result_pretty()))

# Save spectrum and absorption data
save_spectrum_data(
    xopt,
    f_sampled,
    N_HOLES,
    ep1_diel,
    epbkg,
    epN,
    Nx,
    Ny,
)
```

Once the data hence generated is saved, plotting becomes easy.

BIBLIOGRAPHY

[1] Govind P Agrawal. *Fiber-optic communication systems*. John Wiley & Sons, 2012.

[2] Thomas Bäck and Hans-Paul Schwefel. "An overview of evolutionary algorithms for parameter optimization". In: *Evolutionary computation* 1.1 (1993), pp. 1–23.

[3] PE Barnsley and PJ Chidgey. "All-optical wavelength switching from 1.3 mu m to a 1.55 mu m WDM wavelength routed network: system results". In: *IEEE photonics technology letters* 4.1 (1992), pp. 91–94.

[4] Martin Philip Bendsoe and Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.

[5] Jérémy Butet, Pierre-François Brevet, and Olivier JF Martin. "Optical second harmonic generation in plasmonic nanostructures: from fundamental principles to advanced applications". In: *ACS nano* 9.11 (2015), pp. 10545–10562.

[6] Mengyu Chen et al. "Mercury telluride quantum dot based phototransistor enabling high-sensitivity room-temperature photodetection at 2000 nm". In: *ACS nano* 11.6 (2017), pp. 5614–5622.

[7] Mahmoud MR Elsawy et al. "Global optimization of metasurface designs using statistical learning methods". In: *Scientific reports* 9.1 (2019), p. 17918.

[8] Thorsten Feichtner et al. "Evolutionary optimization of optical antennas". In: *Physical review letters* 109.12 (2012), p. 127701.

[9] Shou-fei Gao et al. "Conquering the Rayleigh scattering limit of silica glass fiber at visible wavelengths with a hollow-core fiber approach". In: *Laser & Photonics Reviews* 14.1 (2020), p. 1900241.

[10] James M Glownia et al. "Time-resolved pump-probe experiments at the LCLS". In: *Optics express* 18.17 (2010), pp. 17620–17630.

[11] Nikolaus Hansen. "The CMA evolution strategy: A tutorial". In: *arXiv preprint arXiv:1604.00772* (2016).

[12] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)". In: *Evolutionary computation* 11.1 (2003), pp. 1–18.

[13] Nikolaus Hansen and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies". In: *Evolutionary computation* 9.2 (2001), pp. 159–195.

[14] Ayush Jamdar et al. "Wavelength-agnostic metasurface design for next-generation photodetectors". In: *Acs Photonics (to be submitted)* (2024).

[15] Jakob Søndergaard Jensen and Ole Sigmund. "Topology optimization for nano-photonics". In: *Laser & Photonics Reviews* 5.2 (2011), pp. 308–321.

[16] Jiaqi Jiang, Mingkun Chen, and Jonathan A Fan. "Deep neural networks for the evaluation and design of photonic devices". In: *Nature Reviews Materials* 6.8 (2021), pp. 679–700.

[17] Weiliang Jin et al. “Inverse design of lightweight broadband reflector for relativistic lightsail propulsion”. In: *ACS Photonics* 7.9 (2020), pp. 2350–2355.

[18] Alexander V Kildishev et al. “Stochastic optimization of low-loss optical negative-index metamaterial”. In: *JOSA B* 24.10 (2007), A34–A39.

[19] Philippe Lalanne et al. “Design and fabrication of blazed binary diffractive elements with sampling periods smaller than the structural cutoff”. In: *JOSA A* 16.5 (1999), pp. 1143–1156.

[20] Andrew Lewis et al. “Optimising efficiency and gain of small meander line RFID antennas using ant colony system”. In: *2009 IEEE Congress on Evolutionary Computation*. IEEE. 2009, pp. 1486–1492.

[21] Chong Li et al. “Influence of photonic crystals on the performance parameters of GeSn vertical-structure photodiodes”. In: *Optics & Laser Technology* 163 (2023), p. 109375.

[22] Zhaoyi Li et al. “Empowering metasurfaces with inverse design: principles and applications”. In: *Acs Photonics* 9.7 (2022), pp. 2178–2192.

[23] Hongtao Lin et al. “Mid-infrared integrated photonics on silicon: a perspective”. In: *Nanophotonics* 7.2 (2017), pp. 393–420.

[24] Chaoyue Liu et al. “Silicon/2D-material photodetectors: from near-infrared to mid-infrared”. In: *Light: Science & Applications* 10.1 (2021), p. 123.

[25] Ming Liu et al. “A graphene-based broadband optical modulator”. In: *Nature* 474.7349 (2011), pp. 64–67.

[26] Victor Liu and Shanhui Fan. “S4: A free electromagnetic solver for layered periodic structures”. In: *Computer Physics Communications* 183.10 (2012), pp. 2233–2244.

[27] M Madhumitha, S Selvendran, et al. “Photonic crystal based narrowband optical filter: a brief analysis”. In: *Optik* 228 (2021), p. 166162.

[28] Sean Molesky et al. “Inverse design in nanophotonics”. In: *Nature Photonics* 12.11 (2018), pp. 659–670.

[29] Jessica R Piper and Shanhui Fan. “Total absorption in a graphene monolayer in the optical regime by critical coupling with a photonic crystal guided resonance”. In: *Acs Photonics* 1.4 (2014), pp. 347–353.

[30] Sunae So et al. “Deep learning enabled inverse design in nanophotonics”. In: *Nanophotonics* 9.5 (2020), pp. 1041–1057.

[31] Huong Tran et al. “Si-based GeSn photodetectors toward mid-infrared imaging applications”. In: *ACS Photonics* 6.11 (2019), pp. 2807–2815.

[32] Vy Tran et al. “Layer-controlled band gap and anisotropic excitons in few-layer black phosphorus”. In: *Physical Review B* 89.23 (2014), p. 235319.

[33] Qian Wang et al. “Optically reconfigurable metasurfaces and photonic devices based on phase change materials”. In: *Nature photonics* 10.1 (2016), pp. 60–65.

[34] Xiaoting Wang et al. “Recent advances in the functional 2D photonic and optoelectronic devices”. In: *Advanced Optical Materials* 7.3 (2019), p. 1801274.

[35] Kan Yao, Rohit Unni, and Yuebing Zheng. “Intelligent nanophotonics: merging photonics and artificial intelligence at the nanoscale”. In: *Nanophotonics* 8.3 (2019), pp. 339–366.

[36] Guowei Zhang et al. “Infrared fingerprints of few-layer black phosphorus”. In: *Nature communications* 8.1 (2017), p. 14071.