DISTRIBUTED SIMULATION AND OPTIMIZATION OF LARGE-AREA METASURFACES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Jinhie Skarda

August 2022

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jelena Vuckovic, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jonathan Fan

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

David Miller

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format.

Abstract

Emerging technologies such as augmented reality, lidar, and mobile imaging have opened a large market for complex, compact, and mass-producible optical systems. Metasurfaces are a promising building block for such next-generation systems. These flat optical elements can use subwavelength scatterers to control light, and can be mass-produced in the same advanced semiconductor foundries that have enabled successful scaling of consumer electronics. However, although metasurface functionality can be experimentally demonstrated, simulating metasurfaces is a central challenge in metasurface design.

This simulation challenge arises because metasurfaces typically span thousands of wavelengths in linear dimension, rendering traditional electromagnetic simulation techniques (e.g. Finite-Difference and Finite-Element methods) intractable. Here, we present a metasurface simulation distribution strategy that can preserve the simulation accuracy while allowing scalability to arbitrarily-large areas. Using this distribution strategy with a GPU-based implementation of the Transition-matrix (T-matrix) method, we show a record-size 3-dimensional metasurface simulation (over $600\lambda \times 600\lambda$) that accurately accounts for scatterer-scatterer interactions significantly beyond the commonly-used locally periodic approximation. We then demonstrate gradient-based optimization of single and multilayer metasurfaces using our distributed T-matrix method. Finally, we discuss using the distribution strategy with Finite-Difference Time-Domain solvers to handle arbitrary scatterer geometries.

Acknowledgments

Everything I have learned in my PhD is thanks to the incredible community that has supported me.

Thank you to my thesis committee. Thank you Jelena Vučković for being the best PhD advisor I could ask for - thank you for giving me the freedom to go where my interests led, and the support to explore widely. Thank you David Miller and Jonathan Fan for being on my reading committee, and for inspiring me to apply to Electrical Engineering graduate programs through your excellent quantum and EM courses - it truly did feel full-circle to have you on my committee. Thank you Olav Solgaard for being on my defense committee and for being a fantastic rotation advisor - I really enjoyed my first experience with experimental optics in your research group. Finally, thank you Mark Brongersma for chairing my thesis committee - I learned so much in your nanophotonics class.

Thank you to my labmates. Thank you to my main mentors, Rahul Trivedi and Logan Su - you are two of the smartest and most generous people I have ever met. Rahul has mentored me through all the work in this thesis - even after he graduated and moved to Germany to postdoc, he still took the time to meet with me and help me problem-solve and troubleshoot. Logan has taught me everything I know about structuring large codebases and smart debugging - I will forever strive to produce code he would merge. Thank you to my mentor and collaborator Hyounghan Kwon - your incredible attention to detail and encyclopedic metasurface knowledge has saved

me so many times this year. Thank you to my mentor Kiyoul Yang - I learned so much working on experimental projects with you in my first couple years. Thank you to Dries Vercruysse, Neil Sapra, Constantin Dory, Jesse Lu, Alyssa Cartwright, Geun Ho Ahn, Sattwik Deb Mishra, Alex White, Dominic Catanzaro, and Diego Ahmad-Stein - I have really enjoyed working on various projects with all of you.

At the end of my 2nd year, I suddenly lost my mom - it is, without a doubt, the hardest thing I have ever been through. Thank you, Priyanka Sekhar, for immediately dropping everything and coming to the East Coast, and for mobilizing my support network before I even knew how much I'd need it. Thank you Priyanka Sekhar, Janette Cheng, Michael Raitor, Sharon Newman, Kimberly Chang-Haines, my mom's amazing community and family, and everyone who made sure I never felt alone those two weeks. There is absolutely no way I would be finishing this degree three years later if it were not for my incredible support network of friends and family.

Thank you to my friends. Thank you Priyanka Sekhar, Michael Raitor, and Emily Cao for the best and the spiciest Thursday nights - JEMP forever. Thank you Kimberly Chang-Haines for all your texts and calls, and for being a sister to me. Thank you Sharon Newman for sharing almost every part of this PhD journey with me. Thank you Alison Rugar for being an incredible friend in my lab from year 1. Thank you Raj Paul for all your encouragement and empathy. Thank you William Cai for being a piece of home in California. Thank you Rahul Trivedi, Logan Su, Geun Ho Ahn, Sattwik Deb Mishra, and Alex White for all the camaraderie inside and outside of lab. Thank you Margalit Glasgow, Sharon Newman, Jade Thornton, Sara Shonkwiler, Chris Bernard, Lily Durkee, and Chris Coriz for helping me rediscover my love of the outdoors and being incredible adventure buddies. Thank you Margalit for also helping me get into harmony singing and guitar playing.

Thank you to my family. Thank you Grace and Bob Dowdell and the whole famiLEE for all the ways you have loved and cared for me. Thank you Ray Skarda

for all your support. Thank you to everyone in my mom's large and loving community, especially Elaine Ryan who has called me every week to check in. The outpouring of love and support from all of you constantly leaves me in awe. Thank you to Lisa and Gonzo Jaquez, who have walked daily through this grief journey with me these past two years - I could not have done this without your support.

Thank you to my mom, Haeok Lee. Thank you for showing me what it means to live a life so full of love and resilience that it can't help but fill everyone it touches. I love you and I will never stop missing you.

Contents

List of Figures

Chapter 1

Introduction

Being able to achieve full phase control of optical fields is a central challenge in optical engineering, with diverse applications in imaging, sensing, augmented, and virtual reality systems [50, 9]. Traditionally, free-space phase control has been accomplished via refractive optics, wherein the different phases are achieved by varying the path length of light propagating inside a high index material. However, the curved surfaces needed for phase control in this refraction-based approach result in bulky elements that cannot be mass-produced in semiconductor foundries. This large-footprint and expensive production is very problematic for many of the next-generation technologies, such as augmented reality, lidar, and mobile imaging, that require compact-footprint optical elements and are meant to be consumer devices.

The past decades have seen a rapid development of metasurface-based optical elements that exploit collective scattering properties of subwavelength structures for phase-shaping the incoming fields and are significantly more compact and integrable when compared to the conventional refractive optical elements [13, 100, 18, 39, 46, 49, 52]. Metasurfaces can provide phase control of electromagnetic fields in an utlrathin, flat footprint – this flat footprint allows mass-production via semiconductor foundries,

which dramatically reduces the price of manufacturing. Additionally, because metasurfaces can implement arbitrary phase-masks on the incoming light, they can be designed for multiple functionalities. This compact footprint, mass-producibility, and design flexibility make metasurfaces promising building blocks for next-generation optical technologies.

There have been many promising experimental demonstrations of implementing functionalities critical to next-generation technologies using metasurfaces. Metasurfaces have been used to create OLED displays with the ultra-high pixel density required to preserve resolution in the small footprint of augmented-reality glasses [44], and to implement a see-through display for augmented-reality [49]. Metasurfaces whose response can be actively tuned have been demonstrated for beam steering [68, 51, 38], for use in Lidar systems. Metasurfaces capable of full-Stokes polarimetry [7, 30] and edge detection [98, 100] have been demonstrated for mobile imaging and computation applications. There are even startup companies, such as metalenz (www.metalenz.com), producing commercial metasurfaces mass-produced in foundries for mobile imaging and facial recognition.

1.1 Metasurface simulation bottleneck

Electromagnetic simulation is a key element of optical device design, particularly because electromagnetic simulation is an excellent predictor for device experimental performance [72]. General-purpose electromagnetic solvers such as Finite-Difference Time-Domain (FDTD) [87], Finite-Difference Frequency-Domain (FDFD) [76], or Finite Element Method (FEM) [75] are the most commonly-used simulation approaches because they are widely available in both commercial and open-source form and they can handle a diverse range of applications. However, the simulation time and required

memory for these general-purpose solvers very rapidly increase as the simulation volume increases (Fig. 1.1). Indeed, these general-purpose solvers discretize the spatial domain of the simulation, giving a scaling of $O(l^2)$ where l is the linear dimension. The grid-size for this discretization depends on the wavelength and refractive index, and is typically smaller than $\frac{\lambda}{10n}$ where λ is the shortest simulation wavelength and n is the largest refractive index. This presents a challenge for simulating metasurfaces, since practical metasurfaces are 10^2 - 10^3 λ in the linear dimension, making it impractical to use these general-purpose solvers.

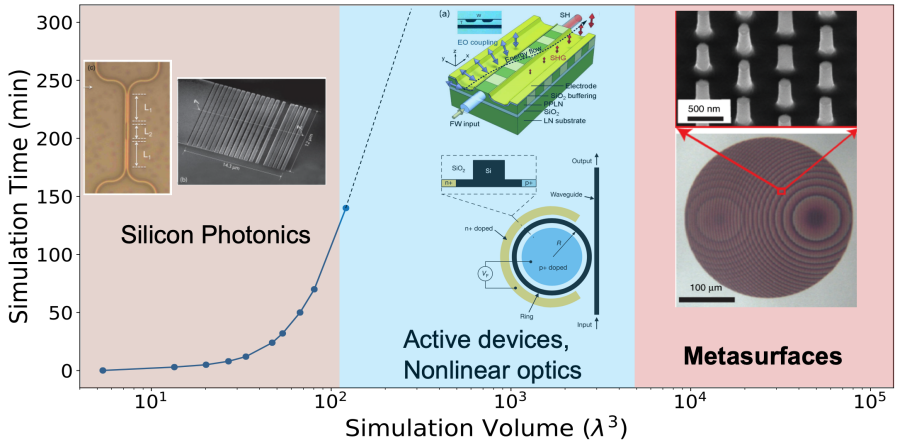


Figure 1.1: **Scalability challenge for general-purpose full-wave electromagnetic solvers.** Simulation time vs simulation volume for a general-purpose full-wave finite-difference solver, showing the extremely rapid increase in simulation time as the simulation volume increases. Although these general-purpose solvers can be useful for simulating passive silicon photonics devices and some active devices, the large simulation volumes necessary to simulate full metasurfaces are intractable for these general-purpose solvers. The inset images for the silicon photonics devices are from [57] (left) and [2] (right). The inset images for active and nonlinear devices are from [25] (Copyright Optica 2019 - reprinted with permission) (top) and [91] (Copyright Springer Nature 2005. Reprinted with permission.) (bottom). The inset image for metasurfaces is from [4] (Copyright Springer Nature 2015. Reprinted with permission.).

1.2 Locally-periodic metasurface design and simulation

The most commonly adopted metasurface-design strategy proceeds in two steps. *first*, a library of meta-atoms is generated by varying a few geometric paramters specifying the meta-atom, and a library response curve for transmission and phase is computed from periodic simulations of these meta-atoms (Fig. 1.2). These periodic simulations are very efficient because the meta-atom dimensions are typically sub-wavelength. Since metasurfaces are often 10^2 - 10^3 λ in the linear dimension, the field variations within a single meta-atom cell are ignored and instead a single value for the transmission and phase response is computed for the meta-atom by integrating over the cell.

Next, an aperiodic meta-surface is generated by laying out the periodic meta-atoms corresponding to the target spatially-varying phase profile [61, 3, 69, 93, 1, 6, 23, 31]. Fig. 1.3 illustrates this process for implementing a focusing phase-profile to design a metalens.

Then, the goal is to estimate the performance of the metasurface design. The most common performance metric used for metalenses is focusing efficiency, which is defined as the ratio of the power within a circle of radius 3 $\times$ FWHM in the focal plane to the power incident on the metasurface. FWHM refers to the Full-Width at Half-Maximum, which is defined as $\frac{\lambda}{2NA}$ – λ is the operating wavelength and NA is the Numerical Aperture defined as $\frac{\frac{L}{2}n_{bg}}{\sqrt{f^2+\left(\frac{L}{2}\right)^2}}$, where L is the metasurface side-length, f is the focal length of the desired lens, and n_{bg} is the index of the material in which the scatterers are embedded. In order to compute the desired performance metric, such as the focusing efficiency, the we must compute the optical fields after they pass through the metasurface. In the following two sections (Sec. 1.2.1 and Sec. 1.2.2), we discuss the two most common methods for this computation.

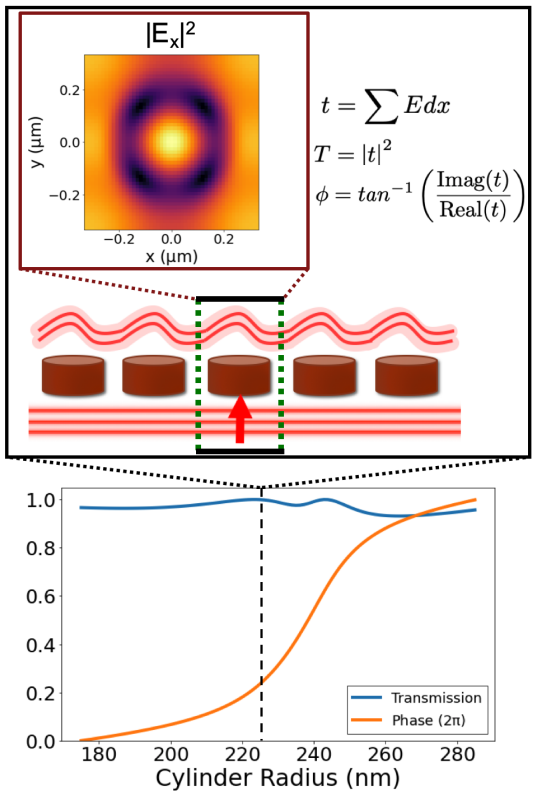


Figure 1.2: **Meta-atom library response curve and generation process.** Meta-atom library response curve of transmission and phase vs radius for meta-atom cells consisting of silicon cylinders with height with height 220 nm, radii range of 175-280 nm, lattice period of 666 nm, and background refractive index 1.66; source wavelength of 1340 nm (from [33]). The upper panel depicts the computation of the transmission and phase values for the cylinder radius indicated by the vertical dashed line in the response curve – the unit-cell of the cylinder is simulated with periodic boundary conditions to give the field that would be obtained from an infinite array of this cylinder. From this unit-cell field, a single complex value for the transmission response is computed – the transmission is the absolute value and the phase is the angle of this complex number. This process is repeated for a set of cylinder radii and the transmission and phase values are interpolated to arrive at the library response curve.

1.2.1 Phase sampling

This approach is extremely computationally efficient, making it the most popular method for metasurface simulation today. Here, we simply sample the library response

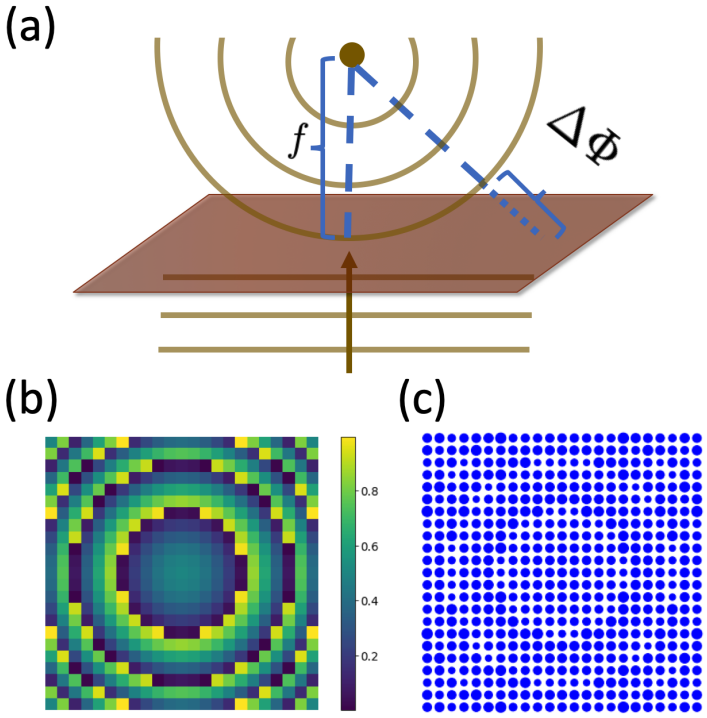


Figure 1.3: **Aperiodic metalens design process from periodic scatterer library.** (**a**) Schematic of the phase difference required for a metasurface to transform an incoming plane wavefront to a spherical wavefront converging to a point at a distance f away from the metasurface – this metasurface is referred to as a metalens with focal length f. (**b**) Required phase change to implement a $30 \times 30\mu m$ metalens with focal length $20\mu m$ in a background with index 1.66. (**c**) Schematic of the aperiodic metalens assembled by sampling the scatterer response library from Fig. 1.2 to determine the cylinder radius for each scatterer.

curve similar to the one in Fig. 1.2 to obtain the transmission and phase response for each meta-atom in the metasurface. We use the transmission and phase to obtain a field value $te^{i\phi}$ just above each meta-atom – this is called the near-field. Then, through multiplication of the free-space propagator in k-space, we propagate this near-field to the focal-plane and use this focal-plane field to compute the focusing efficiency performance metric. For the metalens example in Fig. 1.3, we arrive at a focusing efficiency of 87% using this phase sampling approach (Fig. 1.4). However,

in computing this focusing efficiency, we have relied on two key assumptions in this approach to reduce the computational complexity – approximating the scatterer-scatterer interaction through periodic simulation of each meta-atom, and ignoring the field variations within a single meta-atom cell.

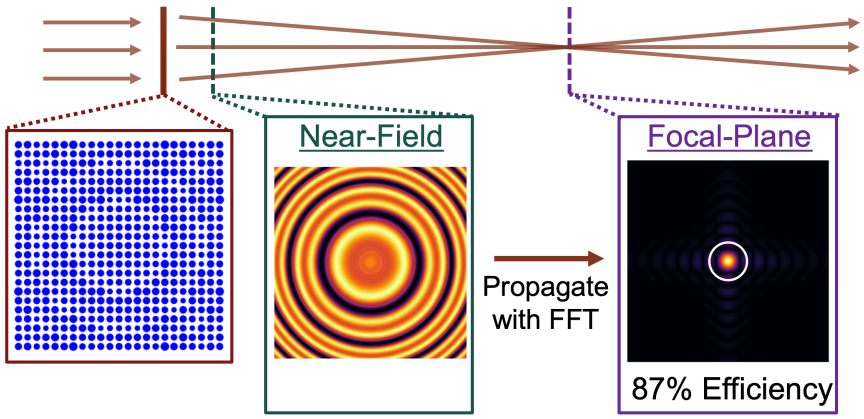


Figure 1.4: **Example computation of metalens focusing efficiency using the phase-sampling LPA approach.** Simulation of the metalens from Fig. 1.3 (left box) by computing the metasurface near-field by sampling the periodic library response curve from Fig. 1.2 (center box), and propagating this near-field to the focal-plane using an FFT (right box). The metalens efficiency is obtained by calculating the fraction of the incident power is contained within the circle of radius 3 × FWHM depicted in white in the right box.

1.2.2 Field stitching

This approach removes one of the two key assumptions we made in the simple phase sampling approach discussed in Section 1.2.1. Here, we still rely on the locally-periodic approximation in which we attempt to account for scatterer-scatterer interaction through periodic simulation of each meta-atom. However, we no longer ignore the field variations within a single meta-atom cell – instead, we perform the periodic simulation for each meta-atom cell and store the full field above the cell. Then, we

obtain the near-field by tiling together these meta-atom cell fields. Once again, we propagate this near-field to the focal-plane to compute the focusing efficiency. For the metalens example in Fig. 1.3, we arrive at a focusing efficiency of 60% using this field stitching approach (Fig. 1.5). Thus, when we remove the single-value approximation in Section 1.2.1, our estimated focusing efficiency drops from 87% to 60%.

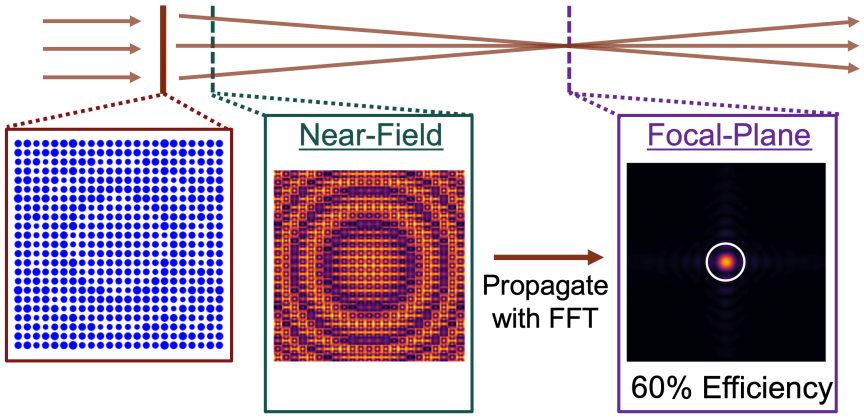


Figure 1.5: **Example computation of metalens focusing efficiency using the field-stitching LPA approach.** Simulation of the metalens from Fig. 1.3 (left box) by computing the metasurface near-field by stitching together the unit-cell periodic simulation fields (as done in the top box of Fig. 1.2) (center box), and propagating this near-field to the focal-plane using an FFT (right box). The metalens efficiency is obtained by calculating the fraction of the incident power is contained within the circle of radius 3 × FWHM depicted in white in the right box.

The metalens from Fig. 1.3 that we have been simulating in Fig. 1.4 and Fig. 1.5 is much smaller than a metasurface of practical interest (linear dimension of $30\mu m$, as opposed to the mm to cm scale linear dimension of practical metasurfaces). This allows us to remove the locally-periodic approximation as well by applying the general-purpose electromagnetic solver FDTD to this metalens – when we do so, we obtain a focusing efficiency of only 16% (Fig. 1.6). However, as we have noted in Fig. 1.1, it is impossible to apply a general-purpose electromagnetic solver like FDTD to

a practical metasurface due to the large simulation volume required.

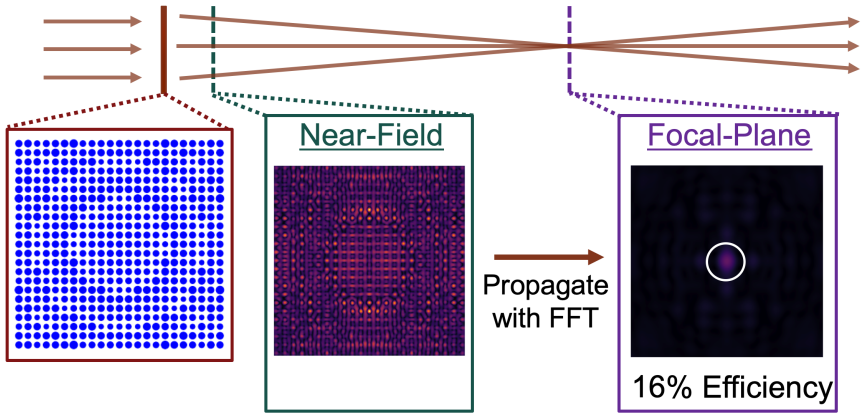


Figure 1.6: **Example computation of metalens focusing efficiency using a full-wave FDTD simulator.** Simulation of the metalens from Fig. 1.3 (left box) by computing the metasurface near-field through an FDTD simulation of the full metasurface (center box), and propagating this near-field to the focal-plane using an FFT (right box). The metalens efficiency is obtained by calculating the fraction of the incident power is contained within the circle of radius 3 × FWHM depicted in white in the right box.

Thus, current metasurface simulation faces a tradeoff between computational efficiency and accuracy – techniques that rely on the locally-periodic approximation are scalable but not accurate in cases where the phase profile is rapidly-varying (for example, high-NA lens profiles) or sensitive to broken periodicity (for example, metasurfaces based on resonant effects like the metasurface in Fig. 1.3, 3.5b, and 3.11). General-purpose electromagnetic solvers provide accurate simulation results but are not scalable to the large-areas required for practical metasurfaces.

1.3 Other metasurface design approaches

Generating the metasurface library for the commonly-used design approach discussed in Section 1.2 becomes increasingly difficult for multi-functional design problems. For

instance, while it is usually not difficult to generate a library for designing a simple phase-mask operating at a few operating modes [81, 6, 45], it becomes increasingly difficult to scale up the number of modes since the same metasurface is required to simultaneously satisfy multiple design conditions corresponding to the different input modes. Fully automating design of metasurfaces can provide a potential solution to this problem. Gradient-based optimization has been successful in designing integrated optical elements that are more compact, robust and high performing than their classical counterparts [63, 92, 90, 41, 77, 27, 72, 85].

However, due to the large size-scale of practically-interesting metasurfaces (Fig. 1.1), inverse-design approaches that use discrete general-purpose electromagnetic solvers to simulate and design the full surface are limited to small design areas or a small number of optimization iterations [12, 59], or restrict the parameter space through a specific symmetry that allows for fast simulations [15, 56, 16]. Consequently, nearly all the current methods for inverse-designing large-scale 3D metasurfaces rely on approximate electromagnetic simulations of the metasurface locally using either periodic or radiation boundary conditions [52, 55, 70, 17, 78, 71, 54, 79, 8, 96, 42, 43, 10], which do not accurately account for interactions between different meta-atoms. These approaches are thus fundamentally limited to designing metasurfaces with slow phase variations due to the implicit local approximation. A coupled-mode formalism can also be applied for metasurface simulation and optimization [99] but this approach is not guaranteed to yield exact fields, particularly for metasurfaces with multiple low quality-factor modes.

1.4 Thesis overview

In this thesis, we propose and demonstrate a numerically accurate simulation strategy that can be used to design and analyze large-area metasurfaces (Chapter 2). Our

strategy relies on a distribution of the simulation method where the simulation time scales linearly with the compute resources. This is achieved by a Nyquist-sampling decomposition of the fields incident on the metasurface, similar to that used recently to characterize the discrete impulse response of aperiodic metasurfaces [89]. Our distribution strategy, by ensuring minimal communication between compute nodes, allows for a linear reduction in the simulation time with the number of compute nodes, indicating that arbitrarily large metasurfaces can be simulated in reasonable time with sufficiently large number of compute nodes. On each compute node, we implement a GPU-based transition-matrix (T-matrix) simulation[95, 94, 97] (Chapter 3). Though there are GPU-optimized FDTD implementations that allow fast simulation of unit-cells up to 100 $\lambda \times 100\ \lambda$ [40], these approaches do not currently provide a low-overhead means of parallel simulation distribution. We demonstrate numerically accurate simulations of metasurfaces of size 1mm $\times$ 1 mm at a wavelength of 1.55μm (about $645\lambda \times 645\lambda$) on a cluster of 48 GPU nodes. With this distributed T-matrix solver, we demonstrate the ability to efficiently compute the gradients with respect to both the geometry and the positions of the meta-atoms for both single- and multi- layer metasurfaces, thus enabling the application of optimization-based design to large-scale multi-layer metasurfaces. Finally, we demonstrate our distribution strategy with an FDTD subregion solver, enabling large-area simulation and optimization of arbitrary metasurface configurations (Chapter 4).

Chapter 2

Low-overhead distribution method for large-area simulation

2.1 Nyquist Sampling of Propagating Incident Field

To simulate millimeter-scale metasurfaces, it is essential to parallelize the simulation method across multiple compute nodes. In order to be scalable, however, this parallelization scheme should introduce only a modest communication overhead in the simulation as this communication overhead can potentially offset any time savings achieved due to the parallelization ([88, 37, 28]).

For metasurface simulations, however, by utilizing the property that the incident fields generated by far-field sources will be within the light-cone in the $\mathbf{k}-$space, a parallelization strategy can be devised that requires minimal communication between the compute nodes. The fundamental principle behind this parallelization is to represent the bandlimited incident field by its samples using the Nyquist sampling theorem [48]. More precisely, consider an incident field propagating along the $z-$direction —

the transverse polarization of this field, $\mathbf{E}^T_{inc}(x, y, z)$ at any z can be expressed as

$$\mathbf{E}^T_{inc}(x, y, z) = \sum_{i,j} \mathbf{E}^T_{inc}(x_i, y_j, z) f_{i,j}(x, y) \tag{2.1}$$

where $x_i, y_j = i\lambda/2, j\lambda/2$ with λ being the wavelength in the background medium, and $f_{i,j}(x, y)$ is a *jinc* function [34] centered at (x_i, y_j). The *jinc* function [34] centered at (x_i, y_j) is defined in Eq. 2.2, where $j_1(\cdot)$ is the first order spherical bessel function and k_0 is the wavenumber for the background medium.

$$jinc_{i,j}(x, y) = \frac{j_1\left(k_0\sqrt{(x - x_i)^2 + (y - y_j)^2}\right)}{k_0\sqrt{(x - x_i)^2 + (y - y_j)^2}} \tag{2.2}$$

Each term in the Nyquist decomposition can be considered to be an independent source, which falls off to zero with distance (Fig. 2.1), and the response of a metasurface to these individual sources can be obtained by considering only a spatially-truncated portion of the metasurface in the simulation.

2.2 Locality of Incident Field Samples

The spatial locality of the incident field Nyquist samples is numerically demonstrated in Fig. 2.2, in which we consider the scattered power obtained on exciting a metasurface with a single jinc source as a function of the size of the metasurface included in the simulation. As the size of the metasurface is increased, the scattered power converges, indicating that a local simulation is sufficient to capture the metasurface response. The size of the metasurface to achieve a particular accuracy in the simulation is governed by diffraction of the jinc source by the time it reaches the metasurface.

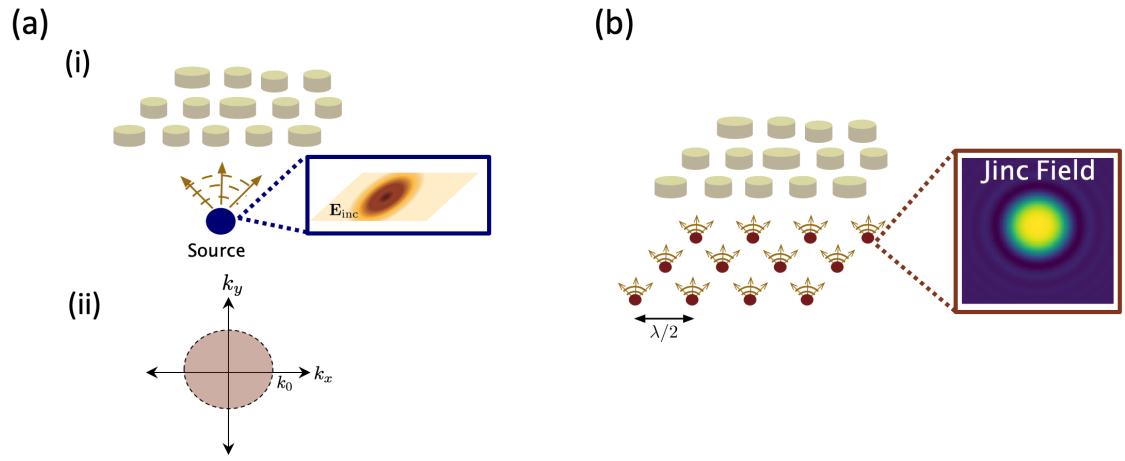


Figure 2.1: **Nyquist sampling of bandlimited incident field.** Schematic of Nyquist sampling of the incident electric field, which is bandlimited to the light-cone because it is propagating.

2.3 Parallelization using Nyquist Samples

To parallelize the simulation, we can then divide up the jinc sources that compose the incident electric field into smaller groups, and simulate the local response of the metasurface for each source group by performing an independent solve on a single compute node. Since the jinc sources are spatially-localized, we only need to consider a spatially-truncated subregion of the metasurface to compute its response to a given group of jinc sources. The total size of the subregion simulation is determined by two factors: the chosen subregion size and the required padding. The subregion size determines how many jinc sources will be included in each subregion simulation, and is chosen with the available computational resources in mind. Choosing a smaller subregion size will result in a larger number of faster and cheaper subregion simulations. Choosing a larger subregion size will result in a smaller number of more expensive subregion simulations. The padding size is the extra area included around the subregion to ensure that the effect of the jinc sources on the outer edge of the subregion is

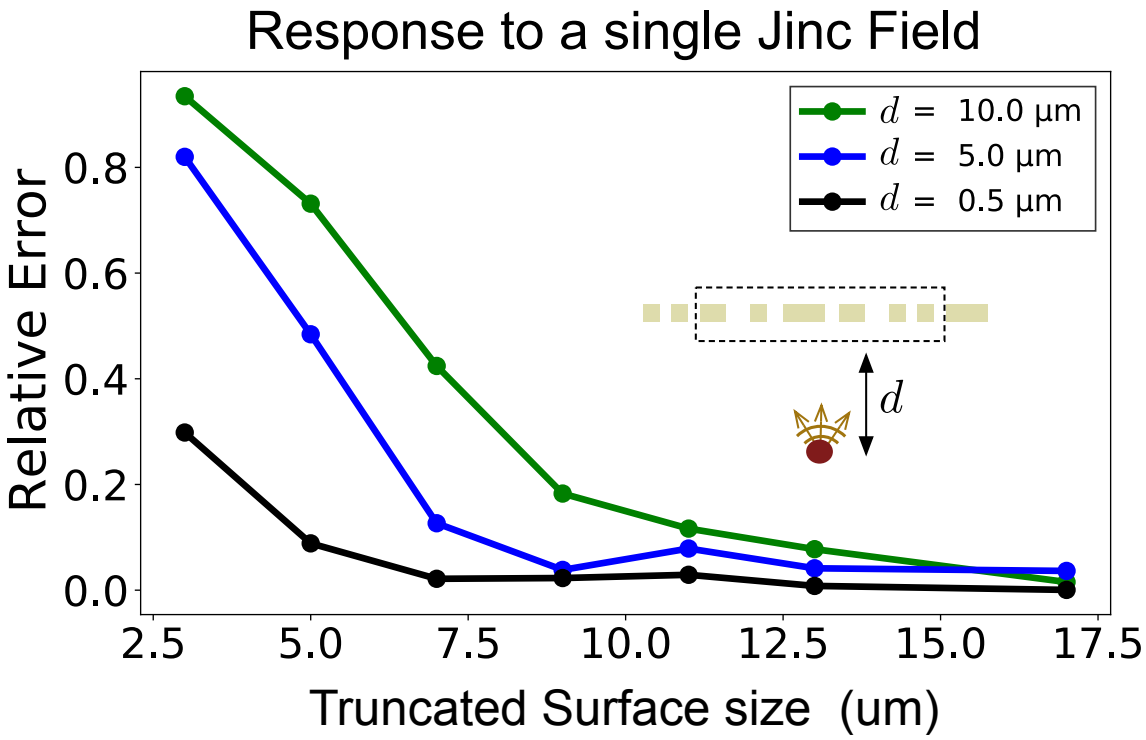


Figure 2.2: **Padding study showing spatial locality of the jinc source.** Percent error in scattered field power versus spatial-extent of metasurface included in the simulation for a single jinc source placed 10 μm (green), 5 μm (blue), and 0.5 μm (black) from the metasurface. The full metasurface is a 25 μm $\times$ 25 μm metalens with focal length of 10 μm, and the surface size on the x-axis of this convergence plot refers to the spatial-extent around the center of this metasurface that is included in the simulation – the metalens scatterer library is shown in Fig. B.1b. The y-axis relative error is computed assuming the simulation including the full metasurface is the converged result.

fully captured. This padding area overlaps with the adjacent subregions, and its size is determined by performing the jinc source spatial-localization study illustrated in Fig. 2.2. Fig. 2.3 depicts the subregion size and padding for a group of jinc sources.

The subregion simulations performed on each compute node are of linear dimension $subregion_{size} + 2 \times padding$ – a smaller, computationally-tractable size. After having performed all the subregion simulations, the electric fields obtained can be added together to compute the total electric field due to the linearity of Maxwell's

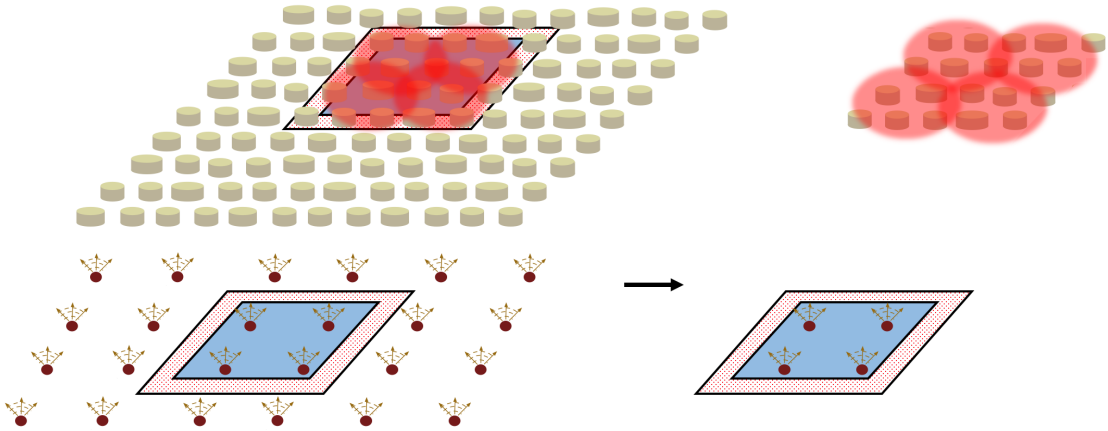


Figure 2.3: **Subregion size and padding.** Schematic of the definition of subregion size and padding for the subregion simulations. The subregion size (depicted by the blue rectangle) defines the number of jinc sources included in a given subregion simulation. This size is chosen based on available computational resources. The padding size (depicted by the dotted-red rectangle) is additional area included in the subregion simulation to account for the full extent of the subregion's outermost jinc sources. This padding area overlaps with adjacent subregions and is determined by a jinc source localization study like the one performed in Fig. 2.2. The total linear dimension of each subregion simulation is then $subregion_{size} + 2 \times padding$.

equations. Importantly, this parallelization strategy is very low-overhead in that it only requires communication between the compute nodes once at the start and once at the end of the simulation – the parallelization scheme is depicted in Fig. 2.4.

We use RabbitMQ (https://www.rabbitmq.com) to create a queue of the meta-surface subregion simulations and manage the distribution of these simulations to the available GPU compute nodes in a fault-tolerant manner. The RabbitMQ message-passing interface handles both the distribution of the subregion simulations and the collection and summation of the resulting fields. In the event that a subregion simulation does not successfully complete (for example, if a GPU compute node dies during computation), the subregion simulation remains in the queue to be executed by the

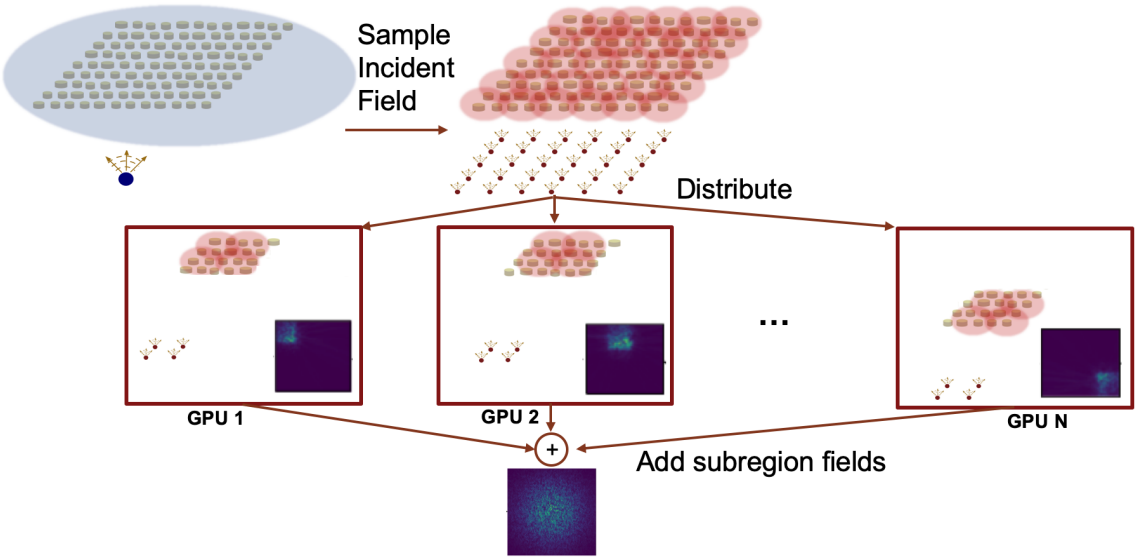


Figure 2.4: **Low-overhead parallelization scheme to allow simulation of arbitrarily large metasurfaces.** Schematic of the simulation distribution scheme — the incident field is first sampled and represented as a superposition of jinc sources, and then smaller groups of jinc sources and the locally surrounding metasurface regions are simulated on independent GPUs. This scheme is very low-overhead because the GPU compute-nodes only need to communicate once when the subregion information is distributed and once when the subregion simulations complete.

next available compute-node worker.

2.4 Distribution method performance

In order to simulate millimeter and centimeter scale metasurfaces, we needed a simulation distribution method that is both scalable and accurate. In this section, we assess our distribution method's performance. We study the scalability of our method by benchmarking the simulation time as a function of the number of GPU's used, and find that the total simulation time scales as $1/N_{nodes}$ (Fig. 2.5). This is because the distribution method is low-overhead and because each compute node performs

roughly the same amount of compute. This $1/N_{nodes}$ scaling implies that, given sufficient compute, this method can be used to simulate arbitrarily-large metasurfaces.

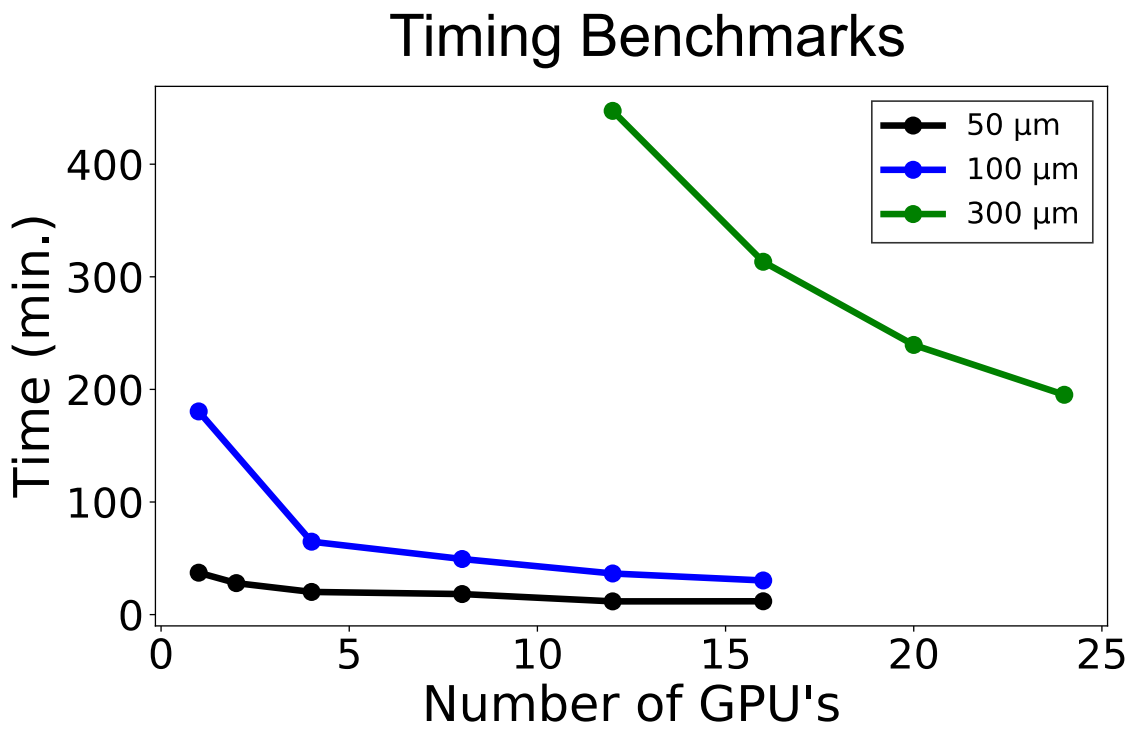


Figure 2.5: **Parallelization method scales linearly with number of compute nodes.** Total simulation time versus number of V100 GPU's used for simulation for a 50 μm (black), 100 μm (blue), and 300 μm (green) metasurface. All metasurfaces have focal length of 25μm and are designed from a library of silicon cylinders with height 940 nm, radii range of 50-250 nm, lattice period of 1070 nm, air background, and source wavelength of 1550 nm (based on scatterer library from[5]) – the metalens scatterer library is shown in Fig. B.1b.

As shown in Fig. 2.2, the accuracy of the single-node subregion simulation can be preserved by using an appropriate truncated surface size because of the spatial-locality of the jinc sources. The appropriate truncated surface size depends on the scatterer library, the number of layers, and the jinc source distance from the metasurface – it can be efficiently determined by performing the study in Fig. 2.2 for the metasurface

design parameters of interest.

Now that we have verified that our distribution method is scalable and preserves the accuracy of the single-node simulator, we must deploy a single-node simulator that is efficient and that accurately captures the scatterer-scatterer interactions. In Chapter 3, we discuss using the transition-matrix method on spherical harmonic basis functions as the single-node simulator. In Chapter 4, we discuss using FDTD as the single-node simulator.

Chapter 3

Spherical harmonic transition-matrix method as single-node simulator

3.1 Motivation

Most of the general-purpose electromagnetic solvers like FDTD and FDFD work on the position basis, meaning that the Maxwell operator needs an element for each spatial point (Fig. 3.1a). For the case of metasurfaces, however, this is not the optimal basis. Since metasurfaces are composed of subwavelength scatterers that cannot introduce large angular momenta, working on the spherical harmonic basis functions provides a more compact description of the system (Fig. 3.1b). A more detailed mathematical description of the spherical harmonic basis functions can be found in Appendix A.1.

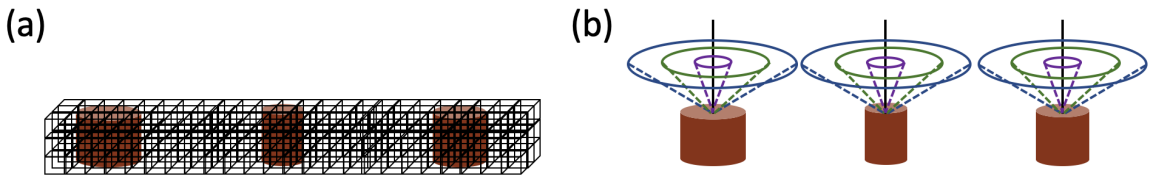


Figure 3.1: **Basis functions for metasurface simulations.** (a) Depiction of the position-domain discretization used when applying most general-purpose electromagnetic solvers to metasurface simulation. (b) Depiction of spherical harmonic basis for metasurface simulation. Since the subwavelength scatterers composing a metasurface do not introduce large angular momenta, this basis provides a more compact system of equations.

3.2 Compact system of equations

The transition-matrix (T-matrix method) approach is depicted in Fig. 3.2. We begin by expressing the incident and scattered fields from the ith single scatterer on the spherical harmonic basis functions:

$$\begin{aligned} \mathbf{E}_{\text{inc}}^{(i)}(\mathbf{x}-\mathbf{x}^{(i)}) &= \sum_j a_j^{(i)} \mathcal{R}\Phi_j(\mathbf{x}-\mathbf{x}^{(i)}) \\ \mathbf{E}_{\text{inc}}^{(i)}(\mathbf{x}-\mathbf{x}^{(i)}) &= \mathbf{a}^{(i)^T} \boldsymbol{\mathcal{R}}\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(i)}) \end{aligned} \tag{3.1}$$

$$\begin{aligned} \mathbf{E}_{\text{sca}}^{(i)}(\mathbf{x}-\mathbf{x}^{(i)}) &= \sum_j s_j^{(i)} \Phi_j(\mathbf{x}-\mathbf{x}^{(i)}) \\ \mathbf{E}_{\text{sca}}^{(i)}(\mathbf{x}-\mathbf{x}^{(i)}) &= \mathbf{s}^{(i)^T} \boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(i)}) \end{aligned} \tag{3.2}$$

The incident field coefficients from Eq. 3.1 can be related to the scattered field coefficients from Eq. 3.2 by the T-matrix:

$$\mathbf{s}^{(i)} = \mathbf{T}^{(i)}\mathbf{a}^{(i)} \tag{3.3}$$

The T-matrix depends only on the scatterer permittivity and geometry, and is computed analytically using the Null-field method detailed in [26].

Collective scattering from multiple scatterers can then be addressed by including the field scattered by all other scatterers in the incident field for a given scatterer:

$$\mathbf{E}_{\text{inc}}^{(i)}(\mathbf{x}-\mathbf{x}^{(i)}) = \mathbf{a}^{(i)^T}\boldsymbol{\mathcal{R}}\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(i)}) + \sum_{n\neq i}\mathbf{s}^{(n)^T}\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(n)}) \tag{3.4}$$

However, the basis functions in Eq. 3.4 are now written with respect to different coordinate system centers. Translation coefficients for the spherical harmonic basis functions ($\boldsymbol{\xi}$), the details of which can be found in Appendix A.1.2, remedy this by allowing a basis function written with respect to one coordinate origin to be shifted to another coordinate origin:

$$\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(n)}) = \boldsymbol{\xi}^{(i)}(\mathbf{x}^{(n)},\mathbf{x}^{(i)})\boldsymbol{\mathcal{R}}\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(i)}) \tag{3.5}$$

Using the translation coefficients defined in Eq. 3.5, we can rewrite Eq. 3.4 as:

$$\begin{aligned}\mathbf{E}_{\text{inc}}^{(i)}(\mathbf{x}-\mathbf{x}^{(i)}) &= \mathbf{a}^{(i)^T}\boldsymbol{\mathcal{R}}\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(i)}) + \sum_{n\neq i}\mathbf{s}^{(n)^T}\boldsymbol{\xi}^{(i)}(\mathbf{x}^{(n)},\mathbf{x}^{(i)})\boldsymbol{\mathcal{R}}\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(i)}) \\ \mathbf{E}_{\text{inc}}^{(i)}(\mathbf{x}-\mathbf{x}^{(i)}) &= \left(\mathbf{a}^{(i)} + \sum_{n\neq i}\boldsymbol{\xi}^{(i)}(\mathbf{x}^{(n)},\mathbf{x}^{(i)})^T\mathbf{s}^{(n)}\right)^T \boldsymbol{\mathcal{R}}\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^{(i)})\end{aligned} \tag{3.6}$$

Now that we have expressed the total incident field on the ith scatterer all in terms of basis functions centered on the ith scatterer in Eq. 3.6, we can apply Eq.

3.3 to obtain the coefficients of the field scattered from this ith scatterer:

$$\begin{aligned}\mathbf{s}^{(i)} &= \mathbf{T}^{(i)}\left(\mathbf{a}^{(i)} + \sum_{n\neq i}\boldsymbol{\xi}^{(i)}(\mathbf{x}^{(n)},\mathbf{x}^{(i)})^T\mathbf{s}^{(n)}\right)\\ \mathbf{T}^{(i)^{-1}}\mathbf{s}^{(i)} &- \sum_{n\neq i}\boldsymbol{\xi}^{(i)}(\mathbf{x}^{(n)},\mathbf{x}^{(i)})^T\mathbf{s}^{(n)} = \mathbf{a}^{(i)}\end{aligned} \tag{3.7}$$

We can then write Eq. 3.7 in matrix form as:

$$\underbrace{\begin{bmatrix} \mathbf{T}_1^{-1} & -\boldsymbol{\xi}^T(\mathbf{x}_1,\mathbf{x}_2) & -\boldsymbol{\xi}^T(\mathbf{x}_1,\mathbf{x}_3) & \dots & -\boldsymbol{\xi}^T(\mathbf{x}_1,\mathbf{x}_N)\\ -\boldsymbol{\xi}^T(\mathbf{x}_2,\mathbf{x}_1) & \mathbf{T}_2^{-1} & -\boldsymbol{\xi}^T(\mathbf{x}_2,\mathbf{x}_3) & \dots & -\boldsymbol{\xi}^T(\mathbf{x}_2,\mathbf{x}_N)\\ -\boldsymbol{\xi}^T(\mathbf{x}_3,\mathbf{x}_1) & -\boldsymbol{\xi}^T(\mathbf{x}_3,\mathbf{x}_2) & \mathbf{T}_3^{-1} & \dots & -\boldsymbol{\xi}^T(\mathbf{x}_3,\mathbf{x}_N)\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ -\boldsymbol{\xi}^T(\mathbf{x}_N,\mathbf{x}_1) & -\boldsymbol{\xi}^T(\mathbf{x}_N,\mathbf{x}_2) & -\boldsymbol{\xi}^T(\mathbf{x}_N,\mathbf{x}_3) & \dots & \mathbf{T}_N^{-1}\end{bmatrix}}_{\boldsymbol{\Omega}}\underbrace{\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\\ \mathbf{s}_3\\ \vdots\\ \mathbf{s}_N\end{bmatrix}}_{\mathbf{s}} = \underbrace{\begin{bmatrix}\mathbf{a}_1\\ \mathbf{a}_2\\ \mathbf{a}_3\\ \vdots\\ \mathbf{a}_N\end{bmatrix}}_{\mathbf{a}} \tag{3.8}$$

The matrix $\boldsymbol{\Omega}$ in the above system of equations is referred to as the 'Maxwell operator', since it is equivalent to expressing the frequency domain Maxwell's equations on the vector spherical wavefunction basis. Appendix A.2 details the expansion of the jinc source from Eq. 2.1 on the spherical harmonic basis functions to compute the vector $\mathbf{a}$ on the right-hand side of Eq. 3.8.

As can be seen in Appendix A.1.1, the spherical harmonic basis functions on which the incident and scattered fields are expressed are indexed by the orbital number l and the magnetic number m, with $l \in \{0, 1, 2\dots\}$, $m \in \{-l, -l+1, \dots l-1, l\}$. Since the subwavelength scatterers composing a metasurface cannot introduce large angular momenta, the expansions of the incident and scattered fields Eqs. 3.1 and 3.2 can be truncated to a finite number of terms by ignoring contributions of basis functions with $l > l_{\max}$ – this corresponds to using $2l_{\max}(l_{\max}+2)$ basis functions.

Indeed, we find that a fairly small $l_{\max} = 6$ yields accurate results for metasurface simulations. Thus, with only 96 basis functions for each scatterer, we can accurately capture the scattering properties.

To summarize, the problem of solving the Maxwell's equations has been reduced to the problem of solving the linear system:

$$\mathbf{\Omega s} = \mathbf{a} \tag{3.9}$$

3.3 Numerical implementation

From a numerical perspective, the linear system in Eq. 3.8 is severely ill conditioned due to the transition matrices being close to singular for small scatterers. This issue can be mitigated using the following block diagonal preconditioner $\mathbf{M}$:

$$\mathbf{M} = \begin{bmatrix} \mathbf{T}_1 & & & & \\ & \mathbf{T}_2 & & & \\ & & \mathbf{T}_3 & & \\ & & & \ddots & \\ & & & & \mathbf{T}_N \end{bmatrix} \tag{3.10}$$

Instead of solving $\mathbf{\Omega s} = \mathbf{a}^{(0)}$, we solve $\mathbf{M\Omega s} = \mathbf{Ma}^{(0)}$ (a system of equations similar to those found in [60]). Validation simulations for our T-matrix simulation method implementation are shown in Appendix Fig. A.1 — we see good agreement between the T-matrix method, FDTD, and FDFD simulations.

We compute the T-matrices for each scatterer using the null-field method from [26]. We use a numerical discretization of 0.0025 microns for evaluating the surface integrals to compute the T-matrices, and we use $l_{\max} = 6$ for truncating the spherical

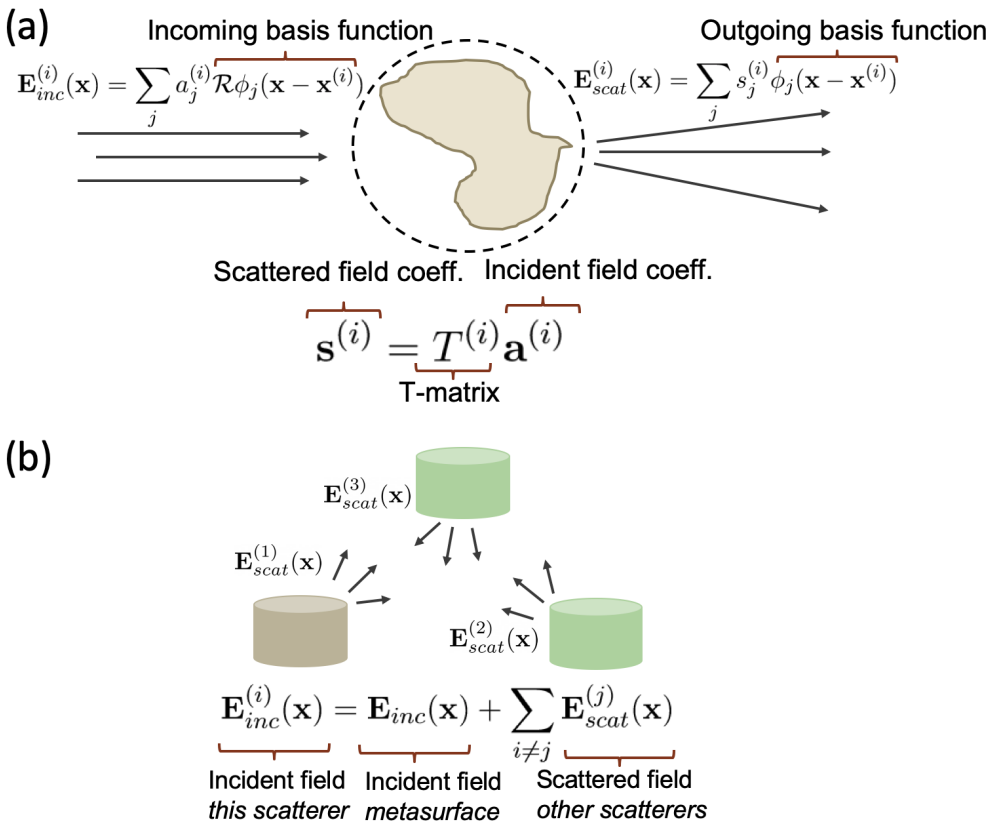


Figure 3.2: **Application of T-matrix method on spherical harmonic basis functions.** (a) Schematic of T-matrix method applied to a single scatterer. The incident and scattered fields are expressed on the spherical harmonic basis functions, and the T-matrix is analytically computed from the permittivity and geometry of the scatterer. (b) Schematic of T-matrix method applied to multiple scatterers. The scattered field from all other scatterers is incorporated into the incident field for a given scatterer, yielding a linear system of equations for the scattered field coefficients.

harmonic basis functions on which we express the incident and scattered fields.

To obtain a more computationally-efficient implementation of the T-matrix method, we implement several optimizations often used in the field of computational engineering. First, we use an iterative algorithm (GMRES) to solve the system of equations resulting from the T-matrix formalism — a major advantage of using iterative algorithms is that they only require matrix-vector products which we program in a fully

parallelized manner on GPUs. Furthermore, the computation of the T-matrix of the individual meta-atoms as well as of the coupling coefficients requires computing special functions which can be time consuming. In our implementation, we precompute lookup tables for these special functions before the start of the T-matrix simulation which significantly cuts down the run-time of the full simulation.

Accelerating the solution of the resulting system of linear equations given in Eq. 3.9 using hardware accelerators such as GPUs has been an immensely successful approach in scaling up partial differential equation solvers. In this section, we describe a simple approach to implement the transition matrix simulation on GPUs — in particular, there are two issues that we address using common approaches from the field of computational engineering:

1. *Speeding up matrix solve*: The matrix solve using GMRES relies on the ability to perform matrix vector products (i.e. if we are solving $\mathbf{Ax} = \mathbf{b}$ using GMRES, we need to be able to compute the product of the matrix $\mathbf{A}$ with an arbitrary vector $\mathbf{x}$). For a $N \times N$ dense matrix, this computation is an $O(N^2)$ operation, and consequently speeding up this operation is a key component of scaling up the simulator to larger systems.

2. *Memory obstacle in precomputing the full matrix*: The implementation described in the previous section constructs the full Maxwell operator explicitly as a matrix, and then performs the matrix vector products. While this might work for small scale simulations (≈ 15 μm in linear dimension for subwavelength silicon scatterers while using a machine with 8 GM RAM), for larger simulations the matrix would become too large to store in memory. So as to obviate this issue, we would like to be able to perform matrix-vector products without explicitly constructing the matrix.

To this end, we implement the matrix vector product with the Maxwell operator as

a GPU operation (Fig. 3.3). The matrix elements are computed while performing the matrix-vector product and discarded once the computation involving those elements have been performed. While computing the k^{th} element of the matrix-vector product, the inner product of the k^{th} row of the matrix with the vector needs to be computed — in our implementation, we assign one GPU thread to handle one such inner product so as to parallelize the matrix-vector product operation. This matrix-vector product operation can then be used along with GMRES to fully solve the system of equations — we implement the operations in GMRES other than the matrix-vector product operation using the CUBLAS library [64].

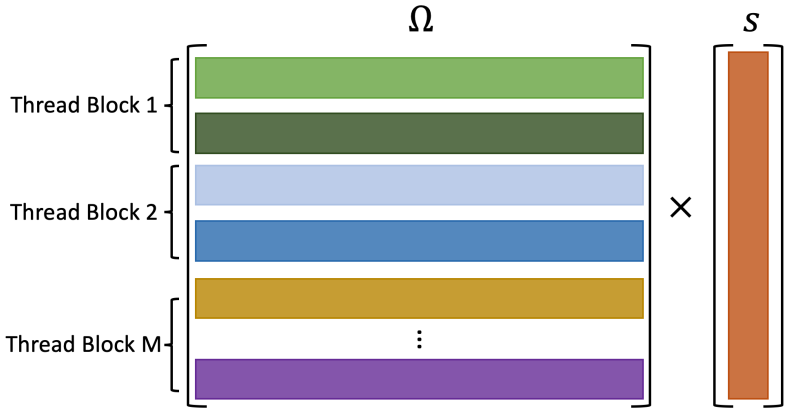


Figure 3.3: **GPU Parallelization for T-matrix simulation.** Schematic of the distribution of the matrix-vector product on the left side of Eq. 3.9 across a GPU, such that this linear system of equations can be solved implicitly using GMRES for better efficiency. Each thread (grouped as thread blocks) is assigned to perform the computation of the product between one row and the vector with the thread computing any matrix-elements required and discarding them once the computation is done. The matrix Ω is of size $NF \times NF$ and the vector s is of size $NF \times 1$, where N is the number of scatterers and F is the number of spherical basis functions – a total of $N \times F$ GPU threads are launched.

Fig. 3.4 shows a comparison between the matrix solve time between a CPU implementation of the solution of Eq. 3.9 and a GPU implementation of the solution of Eq. 3.9. Both the implementations use lookup tables and interpolation — note

that in the CPU solve time, we include the time taken for constructing the Maxwell operator and then solving Eq. 3.9 using GMRES. We observe a 10× speedup on using the GPU implementation over the CPU implementation. We also do a rough comparison between our GPU implementation of the T-matrix method and the commercially available FDTD software Lumerical [84] in Appendix Fig. A.2 and find our T-matrix method solver is several orders of magnitude faster.

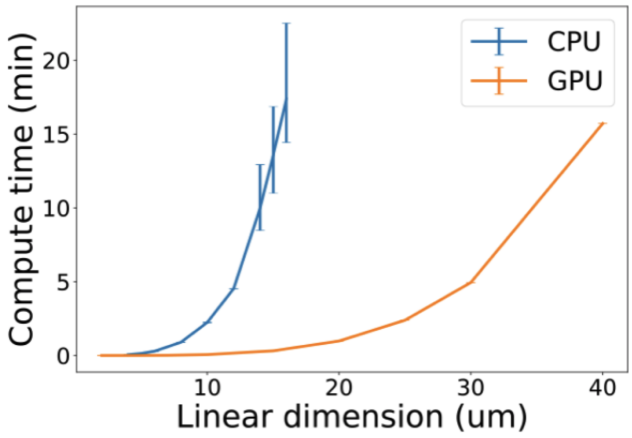


Figure 3.4: **CPU vs GPU Implementation of T-matrix simulation.** Comparison of the solve time between the CPU and GPU implementations for a 2D array of cylinders located at randomly chosen positions within a rectangle of the specified linear dimensions. The error bars indicate the spread in the solve time in between 10 different randomly chosen configurations of the cylinders for the same linear dimension. Note that both GPU and CPU simulations are performed with GMRES with a residual of 10^{-6}. All the GPU simulations were performed on GTX Titan Black with 6GB memory. This figure is reproduced here from [86].

3.4 Comparison with locally-periodic approximation

To demonstrate that our full metasurface simulation approach captures meta-atom interactions beyond the Locally-Periodic Approximation (LPA), we compare the T-matrix simulation method with the two commonly-used LPA approaches from Section 1.2. Fig. 3.5 compares the T-matrix simulation method with the LPA phase-sampling approach from Section 1.2.1 and the LPA field-stitching approach from Section 1.2.2. For high aspect ratio scatterers, we find that while the simple phase-sampling method significantly deviates from the T-matrix method, the field stitching method does not. However, for small aspect-ratio scatterers, which are expected to have larger inter meta-atom interactions, both the LPA approximations significantly deviate from the T-matrix method [33]. These results are a strong indication of the ability of the T-matrix method to capture meta-atom interactions and accurately simulate the metasurface response.

3.5 Very large-area metasurface simulations

Given a sufficiently large number of compute nodes, we expect our simulation strategy to be able to handle arbitrarily-large problems. Here, we present the simulation of a metasurface of size $1mm \times 1mm$ (about $645\lambda \times 645\lambda$) (Fig.3.6a-b) performed on a compute cluster of 48 V100 GPU nodes in about 10 hours. This total time is broken down into the compute times for the key simulation parts in Fig. 3.6c. To our knowledge, this is the largest 3-dimensional metasurface simulation that accurately captures scatterer-scatterer interaction and makes no symmetry assumptions regarding the scatterer distribution. To perform the distributed simulation, we used a subregion size of 20 μm $\times$ 20 μm and selected a padding of 6.5 μm based on the jinc

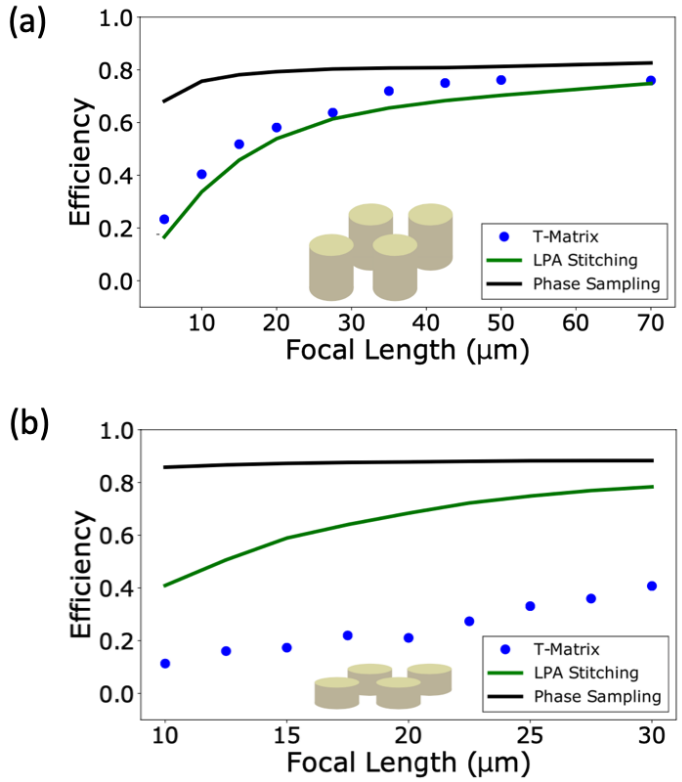
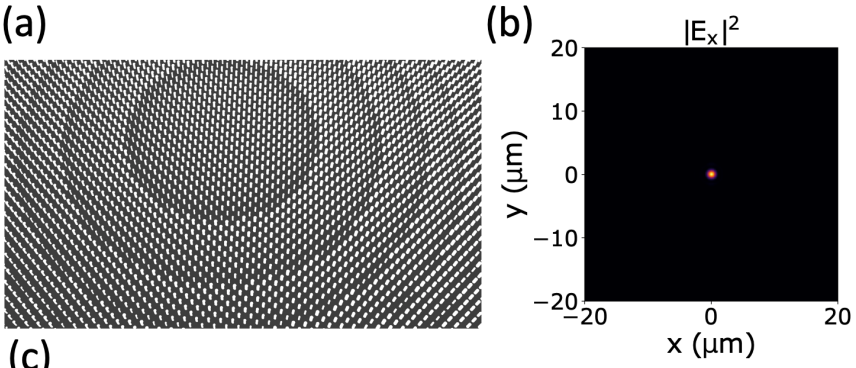


Figure 3.5: **Comparison of T-matrix method simulations with locally-periodic assumption (LPA) simulations.** (**a**) Efficiency versus focal length for 25 μm $\times$ 25 μm metasurfaces designed from a library of high-aspect ratio scatterers with a large period (silicon cylinders with height 940 nm, radii range of 50-250 nm, lattice period of 1070 nm, and air background; source wavelength of 1550 nm – based on scatterer library from [5]) — efficiencies are computed using the T-matrix approach (blue dots), the commonly-used LPA phase sampling approach (black curve), and the LPA field-stitching method (green curve). The metalens efficiency is defined as the ratio of the power within a circle of radius 3 $\times$ FWHM in the focal plane to the power incident on the metasurface. The T-matrix and LPA-stitching methods agree fairly well here because the scatterers are high-aspect ratio and the lattice constant is large, hence the interactions between neighboring scatterers is negligible. (**b**) Efficiency versus focal length for 15 μm $\times$ 15 μm metasurfaces designed from a library of low-aspect ratio scatterers with a small period (silicon cylinders with height 220 nm, radii range of 175-280 nm, lattice period of 666 nm, and background refractive index 1.66; source wavelength of 1340 nm – using scatterer library from [33]) — efficiencies are computed using the T-matrix approach (blue dots), the commonly-used LPA phase sampling approach (black curve), and the LPA field-stitching method (green curve). The metalens efficiency is defined as the ratio of the power within a circle of radius 3 $\times$ FWHM in the focal plane to the power incident on the metasurface. The T-matrix and LPA-stitching methods do not agree here because the scatterers are low-aspect ratio and the lattice constant is small, hence the interaction between neighboring scatterers is significant.

source spatial-locality study in Fig. 2.2 (d = 0.5 μm). This resulted in 2601 subregion simulations, which we distributed across the 48 V100 GPUs using the RabbitMQ system described in Section 2.3.



(c)

LUT Computation (GPU-min.)	1357.9
T-Matrix Solve (GPU-min.)	19078.93
Field Computation (GPU-min. / Detector)	0.0013

Figure 3.6: **Large-area 1mm × 1mm ($645\lambda \times 645\lambda$) metalens simulation.** (**a**) Center portion of the simulated $1mm \times 1mm$ metalens with focal length 0.4mm (NA = 0.78) designed from a library of silicon cylinders with height 940 nm, radii range of 50-250 nm, lattice period of 1070 nm, air background, and source wavelength of 1550 nm (based on scatterer library from[5]). The metasurface consists of 874,225 scatterers. (**b**) X-component of the electric field in the focal plane of the large-area metalens. The lens efficiency is calculated to be 52%, where lens efficiency is defined as the ratio of the power within a circle of radius 3 × FWHM in the focal plane to the power incident on the metasurface. (**c**) Computation time for the key stages of the large-area 1 mm × 1 mm metasurface simulation: *top row* – computing the Look-Up Tables (LUT) used to efficiently perform T-matrix simulation (Section 3.3); *middle row* – computing the T-matrices (Section 3.3) and solving the resulting linear system of equations for the scattered field coefficients (Eq. 3.8); *bottom row* – computing the E and H fields from the scattered field coefficients for each desired detector point (Eq. 3.2). The simulation is performed on 48 V100 GPUs and is distributed between these compute nodes using a subregion size of 20 μm × 20 μm and a padding of 6.5 μm, resulting in 2601 subregion simulations.

3.6 Optimization framework

As discussed in Section 1.3, gradient-based optimization is a promising approach to efficiently explore the extremely vast number of design degrees of freedom afforded by large-area metasurfaces like the metalens in Section 3.5. For example, the problem of designing a metalens could be posed in an optimization framework as depicted in Fig. 3.7.

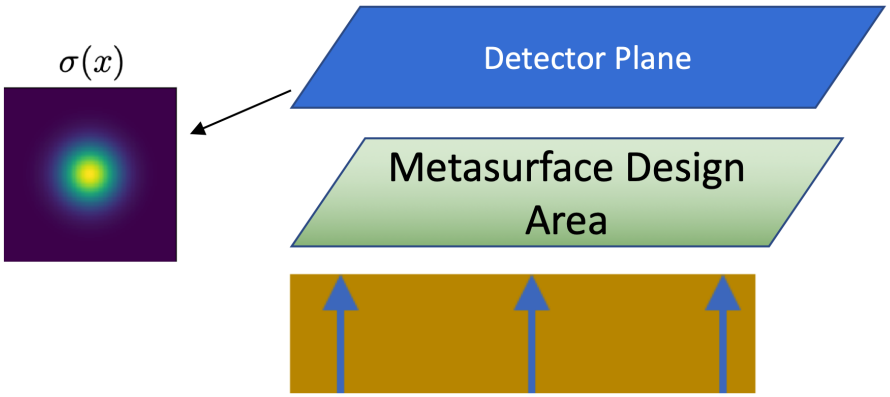


Figure 3.7: **Metalens design as an optimization problem.** Schematic of the metalens design problem posed as an optimization problem. We start with an incident field, a design area for the metasurface, and a target field at the detector plane. We then iteratively optimize the design in the metasurface design area to improve its ability to generate our desired target detector field profile. For the metalens design problem, this target detector field profile ($\sigma(x)$ could be the gaussian field depicted here.

An essential ingredient for optimization-based design of metasurfaces is an efficient evaluation of the gradient of the figure of merit with respect to the design parameters. A particularly useful method to evaluate gradients is based on adjoint-sensitivity analysis [47, 73] which analytically differentiates through Maxwell's equations and computes the gradients with respect to all the design parameters with a cost proportional to only two electromagnetic simulations (Fig. 3.8).

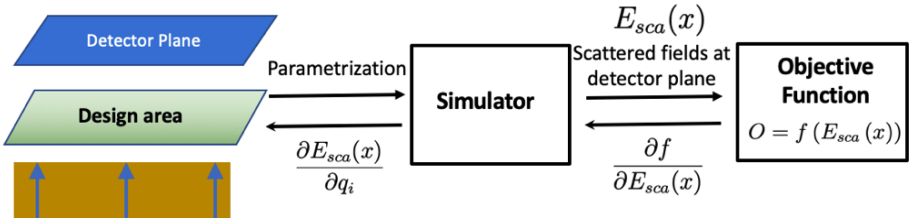


Figure 3.8: **Optimization-based metasurface design procedure.** Schematic of a single iteration in the iterative procedure for optimization-based metasurface design. In the forward pass of this process, the parametrization of the metasurface design area is passed into the metasurface simulator, which passes its computed fields to the objective function for evaluation. In the backward pass of this process, the gradient of the objective function with respect to the field is passed back to the simulator, which computes the gradient of the field with respect to the metasurface parameters. Using the chain rule, the final result of this backward pass is the gradient of the objective function with respect to the metasurface parameters. This gradient is used to update the metasurface design such that the objective function is improved.

The distributed T-matrix simulation method is also amenable to distributed adjoint sensitivity analysis and can allow for scalable evaluation of the gradient of a performance metric defined on the electric fields scattered from the metasurface with respect to both the meta-atom shape and positions. An efficient implementation of the gradient computation allows us to use gradient-based optimization algorithms to optimize the performance of metasurfaces much like that done with inverse-design of silicon photonics devices. In the following subsection, we derive the adjoint expression for the gradient with respect to the meta-atom position and geometry.

3.6.1 Adjoint for position and geometry gradients.

Suppose that we have N scatterers located at $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_N$. Moreover, the geometry of the scatterers are dependent on parameters $p_1, p_2 \ldots p_M$ (e.g. these parameters can be the radii of the cylindrical meta-atoms, or the lengths and breadths rectangular

meta-atoms). Consider a performance metric $\mathcal{O}$ which takes the following form:

$$\mathcal{O} = f\left(\mathbf{E}_{sca}(p_i, q_i)\right) \tag{3.11}$$

Our goal will be to derive the expressions for $\frac{\partial O}{\partial q_i}$ and $\frac{\partial O}{\partial p_i}$. To begin, we note that q_i and p_i are real values, E_{sca} is complex-valued, and we assume f is real-valued. We can then apply the concept of the Wirtinger derivative, using $\frac{\partial O}{\partial q_i}$ as an example (the same applies to $\frac{\partial O}{\partial p_i}$).

$$\begin{aligned} \frac{\partial O}{\partial q_i} &= \frac{\partial f}{\partial \mathbf{E}_{sca}} \frac{\partial \mathbf{E}_{sca}}{\partial q_i} + \frac{\partial f^*}{\partial \mathbf{E}_{sca}{}^*} \frac{\partial \mathbf{E}_{sca}{}^*}{\partial q_i{}^*} \\ \frac{\partial O}{\partial q_i} &= 2Re\left(\frac{\partial f}{\partial \mathbf{E}_{sca}} \frac{\partial \mathbf{E}_{sca}}{\partial q_i}\right) \end{aligned} \tag{3.12}$$

As a concrete example of a practical performance metric $\mathcal{O}$, we can consider the commonly-used overlap objective. Here, the function f would be defined as:

$$f\left(\mathbf{E}_{sca}\right) = \left|\hat{\boldsymbol{\sigma}}^*(\mathbf{x}) \cdot \mathbf{E}_{sca}(\mathbf{x})\right|^2 \tag{3.13}$$

where $\hat{\boldsymbol{\sigma}}^*(x) = \boldsymbol{\sigma}^*(x)dxdy$, $\boldsymbol{\sigma}(x)$ is a vector function of space that has information of the desired field profile (for example, this could be a gaussian field profile for a metalens design problem as depicted in Fig. 3.7), dx and dy are the detector point discretizations in x and y respectively, and $\mathbf{E}_{sca}$ is the field scattered by the metasurface. Then for this overlap objective function, we have:

$$\frac{\partial f}{\partial \mathbf{E}_{sca}} = \left(\hat{\boldsymbol{\sigma}}^*(\mathbf{x}) \cdot \mathbf{E}_{sca}(\mathbf{x})\right)^* \hat{\boldsymbol{\sigma}}^*(\mathbf{x}) \tag{3.14}$$

Position gradient.

The goal of this section is to compute the gradient of the objective function with respect to the scatterer positions in the metasurface. From Eq. 3.12, we begin with:

$$\frac{\partial O}{\partial q_i} = 2Re\left(\frac{\partial f}{\partial \mathbf{E}_{sca}}\frac{\partial \mathbf{E}_{sca}}{\partial q_i}\right) \tag{3.15}$$

Since we have the expression for $\frac{\partial f}{\partial \mathbf{E}_{sca}}$ in Eq. 3.14, we now need to derive an expression for $\frac{\partial \mathbf{E}_{sca}}{\partial q_i}$.

$$\begin{aligned} \mathbf{E}_{sca} &= \mathbf{s}\cdot\boldsymbol{\phi} \\ \frac{\partial \mathbf{E}_{sca}}{\partial q_i} &= \frac{\partial \mathbf{s}}{\partial q_i}\cdot\boldsymbol{\phi} + \mathbf{s}\cdot\frac{\partial \boldsymbol{\phi}}{\partial q_i} \end{aligned} \tag{3.16}$$

We can simplify the first term on the right side of the Eq. 3.16:

$$\begin{aligned} \frac{\partial}{\partial q_i}(\boldsymbol{\Omega}\mathbf{s} = \mathbf{a}) \\ \frac{\partial \boldsymbol{\Omega}}{\partial q_i}\mathbf{s} + \boldsymbol{\Omega}\frac{\partial \mathbf{s}}{\partial q_i} = \frac{\partial \mathbf{a}}{\partial q_i} \\ \frac{\partial \mathbf{s}}{\partial q_i} = \boldsymbol{\Omega}^{-1}\left(\frac{\partial \mathbf{a}}{\partial q_i} - \frac{\partial \boldsymbol{\Omega}}{\partial q_i}\mathbf{s}\right) \end{aligned} \tag{3.17}$$

Then inserting Eq. 3.16 and 3.17 into Eq. 3.15, we have:

$$\frac{\partial O}{\partial q_i} = 2Re\left(\frac{\partial f}{\partial \mathbf{E}_{sca}} \odot \boldsymbol{\phi}^T\boldsymbol{\Omega}^{-1}\left(\frac{\partial \mathbf{a}}{\partial q_i} - \frac{\partial \boldsymbol{\Omega}}{\partial q_i}\mathbf{s}\right) + \mathbf{s}_i^T\frac{\partial f}{\partial \mathbf{E}_{sca}} \odot \frac{\partial \boldsymbol{\phi}_i}{\partial q_i}\right) \tag{3.18}$$

where $\odot$ denotes element-wise vector multiplication.

It seems that computing the gradient requires computation of $\boldsymbol{\Omega}^{-1}$ — this would be a prohibitively expensive simulation that would render the gradient computation

impractical. Note however, in the gradient computation, we only require the computation of $\mathbf{v}^{\mathrm{T}}\boldsymbol{\Omega}^{-1} = [\boldsymbol{\Omega}^{-\mathrm{T}}\mathbf{v}]^{\mathrm{T}}$ and not the full inverse $\boldsymbol{\Omega}^{-1}$. This is equivalent to solving the following system of equations:

$$\boldsymbol{\Omega}^{\mathrm{T}}\boldsymbol{\alpha} = \mathbf{v} \tag{3.19}$$

Eq. 3.19 is labelled as the adjoint simulation, and it needs to be performed *once* at each step of the optimization (importantly, note that this simulation does *not* depend on the variable with respect to which the gradient is being computed). Here, $\mathbf{v}$ is called the adjoint source and is defined as:

$$\mathbf{v} = \frac{\partial f}{\partial \mathbf{E}_{sca}}^{T} \odot \boldsymbol{\phi} \tag{3.20}$$

Thus, our final expression for the position gradient is:

$$\frac{\partial O}{\partial q_i} = 2Re\left(\boldsymbol{\alpha}^T\left(\frac{\partial \mathbf{a}}{\partial q_i} - \frac{\partial \boldsymbol{\Omega}}{\partial q_i}\mathbf{s}\right) + \mathbf{s}_i^T\frac{\partial f}{\partial \mathbf{E}_{sca}} \odot \frac{\partial \boldsymbol{\phi}_i}{\partial q_i}\right) \tag{3.21}$$

Radius gradient.

The goal of this section is to compute the gradient of the objective function with respect to the geometry of the scatterers in the metasurface. From Eq. 3.12, we begin with:

$$\frac{\partial O}{\partial q_i} = 2Re\left(\frac{\partial f}{\partial \mathbf{E}_{sca}}\frac{\partial \mathbf{E}_{sca}}{\partial p_i}\right) \tag{3.22}$$

Once again, since we have the expression for $\frac{\partial f}{\partial \mathbf{E}_{sca}}$ in Eq. 3.14, we now need to derive an expression for $\frac{\partial \mathbf{E}_{sca}}{\partial p_i}$.

$$\begin{aligned} \mathbf{E}_{sca} &= \mathbf{s} \cdot \boldsymbol{\phi} \\ \frac{\partial \mathbf{E}_{sca}}{\partial p_i} &= \frac{\partial \mathbf{s}}{\partial p_i} \cdot \boldsymbol{\phi} + \mathbf{s} \cdot \frac{\partial \boldsymbol{\phi}}{\partial p_i} \end{aligned} \tag{3.23}$$

Since the basis function vector $\boldsymbol{\phi}$ does not depend on the scatterer geometry, $\mathbf{s} \cdot \frac{\partial \boldsymbol{\phi}}{\partial p_i} = 0$. Thus, we only need to simplify the first term on the right-hand side of Eq. 3.23:

$$\begin{aligned} \frac{\partial}{\partial p_i} (\boldsymbol{\Omega}\mathbf{s} = \mathbf{a}) \\ \frac{\partial \boldsymbol{\Omega}}{\partial p_i}\mathbf{s} + \boldsymbol{\Omega}\frac{\partial \mathbf{s}}{\partial p_i} = \frac{\partial \mathbf{a}}{\partial p_i} \\ \frac{\partial \mathbf{s}}{\partial p_i} = \boldsymbol{\Omega}^{-1}\left(\frac{\partial \mathbf{a}}{\partial p_i} - \frac{\partial \boldsymbol{\Omega}}{\partial p_i}\mathbf{s}\right) \end{aligned} \tag{3.24}$$

Since the vector of incident field coefficients $\mathbf{a}$ does not depend on the scatterer geometry, $\frac{\partial \mathbf{a}}{\partial p_i} = 0$. Inserting Eq. 3.23 into Eq. 3.22, we have:

$$\frac{\partial O}{\partial q_i} = 2Re\left(-\frac{\partial f}{\partial \mathbf{E}_{sca}} \odot \boldsymbol{\phi}^T \boldsymbol{\Omega}^{-1} \frac{\partial \boldsymbol{\Omega}}{\partial p_i}\mathbf{s}\right) \tag{3.25}$$

Applying the same adjoint method in Eq. 3.19 with the same adjoint source from Eq. 3.20, we can rewrite Eq. 3.25 in its final form:

$$\frac{\partial O}{\partial q_i} = 2Re\left(-\boldsymbol{\alpha}^T \frac{\partial \boldsymbol{\Omega}}{\partial p_i}\mathbf{s}\right) \tag{3.26}$$

Computational implementation of gradients.

We use central-difference finite-difference to numerically compute all required partial derivatives for the gradient expressions in Eq. 3.21 and Eq. 3.26. For example, $\frac{\partial \mathbf{a}}{\partial q_i}$ is

computed as follows:

$$\frac{\partial \mathbf{a}}{\partial q_i} = \frac{\mathbf{a}(\mathbf{x}[q] + dl) - \mathbf{a}(\mathbf{x}[q] - dl)}{2dl} \tag{3.27}$$

$\frac{\partial \mathbf{s}}{\partial q_i}$ and $\frac{\partial \boldsymbol{\phi}_i}{\partial q_i}$ are computed in the same way as shown in Eq. 3.27.

Recalling the definition of $\boldsymbol{\Omega}$ from Eq. 3.7, we can see that computing its derivatives requires taking the derivatives of $\mathbf{T}^{(i)}$ and $\boldsymbol{\xi}^{(i)}(\mathbf{x}^{(n)}, \mathbf{x}^{(i)})$.

For $\frac{\partial \boldsymbol{\Omega}}{\partial q_i}$, we have:

$$\frac{\partial \mathbf{T}^{(i)}}{\partial q_i} = 0 \tag{3.28}$$

$$\frac{\partial \boldsymbol{\xi}^{(i)}(\mathbf{x}^{(j)}, \mathbf{x}^{(k)})}{\partial q_i} = \begin{cases} 0 & j, k \neq i \\ \frac{\boldsymbol{\xi}^{(i)}(\mathbf{x}^{(j)}[q]+dl, \mathbf{x}^{(k)}) - \boldsymbol{\xi}^{(i)}(\mathbf{x}^{(j)}[q]-dl, \mathbf{x}^{(k)})}{2dl} & j = i \\ \frac{\boldsymbol{\xi}^{(i)}(\mathbf{x}^{(j)}, \mathbf{x}^{(k)}[q]+dl) - \boldsymbol{\xi}^{(i)}(\mathbf{x}^{(j)}, \mathbf{x}^{(k)}[q]-dl)}{2dl} & k = i \end{cases} \tag{3.29}$$

For $\frac{\partial \boldsymbol{\Omega}}{\partial p_i}$, we have:

$$\frac{\partial \boldsymbol{\xi}^{(i)}(\mathbf{x}^{(j)}, \mathbf{x}^{(k)})}{\partial q_i} = 0 \tag{3.30}$$

$$\frac{\partial \mathbf{T}^{(i)}}{\partial p_i} = \begin{cases} 0 & j \neq i \\ \frac{\mathbf{T}^{(i)}(\mathbf{p}[i]+dl) - \mathbf{T}^{(i)}(\mathbf{p}[i]-dl)}{2dl} & j = i \end{cases} \tag{3.31}$$

We parallelize all these finite-difference derivative computations in Eq. 3.27, 3.28, and 3.30 across GPU threads due to memory and speed considerations. These finite-difference derivatives must be computed for each design degree of freedom for each scatterer for each spherical basis function. However, because the matrices and vectors involved in these computations are sparse, we can slightly modify the GPU thread-allocation scheme depicted in Fig. 2.4 that parallelizes over spherical basis function

to also parallelize with respect to design degree of freedom and scatterer. Fig. 3.9a depicts the GPU thread-allocation scheme used for $\frac{\partial \mathbf{a}}{\partial q_i}$, $\frac{\partial \mathbf{s}}{\partial q_i}$, and $\frac{\partial \boldsymbol{\phi}_i}{\partial q_i}$. Fig. 3.9b depicts the GPU thread-allocation scheme used for $\frac{\partial \boldsymbol{\Omega}}{\partial q_i}\mathbf{s}$. Finally, Fig. 3.9c depicts the GPU thread-allocation scheme used for $\frac{\partial \boldsymbol{\Omega}}{\partial p_i}\mathbf{s}$.

Combining this multi-GPU gradient computation with the multi-GPU forward simulation, we have opened the door to gradient-based optimization over the many degrees of freedom afforded by arbitrarily large metasurfaces. In particular, our method allows optimizing both the shape and position of the scatterers composing the large-area metasurface — optimizing the scatterer positions is very difficult for any inverse-design approach that relies on a periodicity assumption. We interface our distributed metasurface solver with the photonic optimization framework software SPINS (https://github.com/stanfordnqp/spins-b) [85] to perform the inverse-design.

3.7 Distributed optimization of low aspect-ratio metalens

Although thin low-aspect ratio metasurfaces (Huygens metasurfaces) are of interest because they are more amenable to large-scale fabrication, they have not found widespread adoption due to their very limited efficiencies and angular responses [33]. These thin metasurfaces rely on overlapping electric and magnetic resonances to create phase-shifts spanning the required 2π range. As such, they are very sensitive to scatterer-scatterer interactions and are difficult to design and model using traditional periodic-approximation approaches. Our ability to accurately model the scatterer-scatterer effects in our metasurface inverse-design may allow discovery of more practical Huygens metasurfaces [65, 11].

As a proof-of-concept for this Huygens metasurface design use-case, we perform

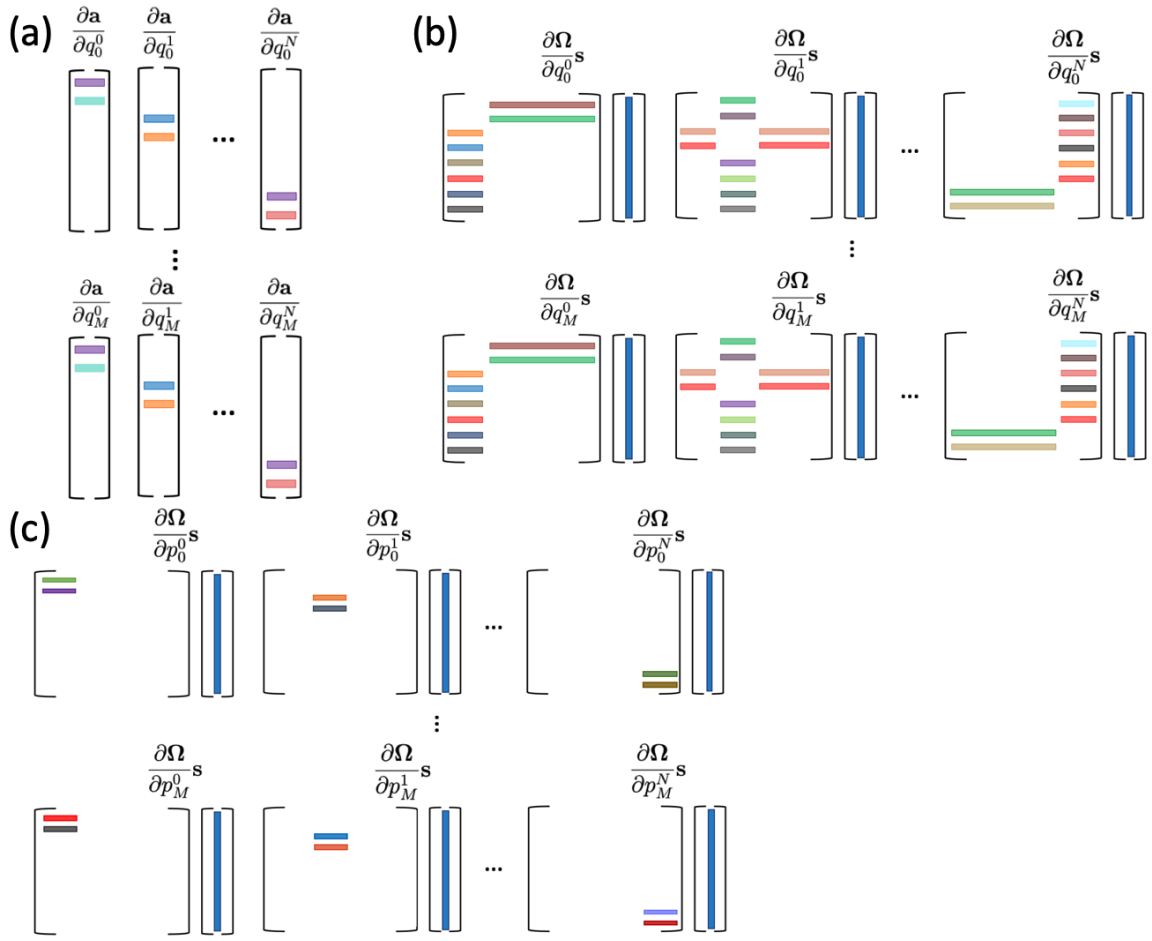


Figure 3.9: **GPU thread-allocation schemes for adjoint gradient computation.** Schematic of the GPU thread allocation schemes, in which each colored rectangle depicts a GPU thread. M is the number of position components, N is the number of scatterers, and F is the number of spherical basis functions. (**a**) Scheme for $\frac{\partial \mathbf{a}}{\partial q_i}$, $\frac{\partial \mathbf{s}}{\partial q_i}$, and $\frac{\partial \boldsymbol{\phi}_i}{\partial q_i}$ – a total of $M \times N \times F$ GPU threads are launched. (**b**) Scheme for $\frac{\partial \boldsymbol{\phi}_i}{\partial q_i}\mathbf{s}$ – a total of $M \times N \times F$ GPU threads are launched. The banded structure of $\frac{\partial \boldsymbol{\phi}_i}{\partial q_i}$ is a result of Eq. 3.28. (**c**) Scheme for $\frac{\partial \boldsymbol{\phi}_i}{\partial p_i}\mathbf{s}$ – a total of $M \times N \times F$ GPU threads are launched. The banded structure of $\frac{\partial \boldsymbol{\phi}_i}{\partial p_i}$ is a result of Eq. 3.30.

a distributed gradient-based optimization with respect to both the positions and the radii of the cylindrical meta-atoms composing a metalens initially designed with the same scatterer library used in Fig. 3.5(b). In order to properly define the subregions

for the distributed optimization, we first perform a jinc spatial localization study (padding study) in Fig. 3.10 for the Huygens metasurface library following a similar approach to Fig. 2.2. From this study, we determine that a total truncated surface size of $12\mu m$ (corresponding to a padding size of $6\mu m$ - see Fig. 2.3)is sufficient to capture the response of the Huygens scatterer library to a single jinc source placed $0.01\mu m$ from the metasurface.

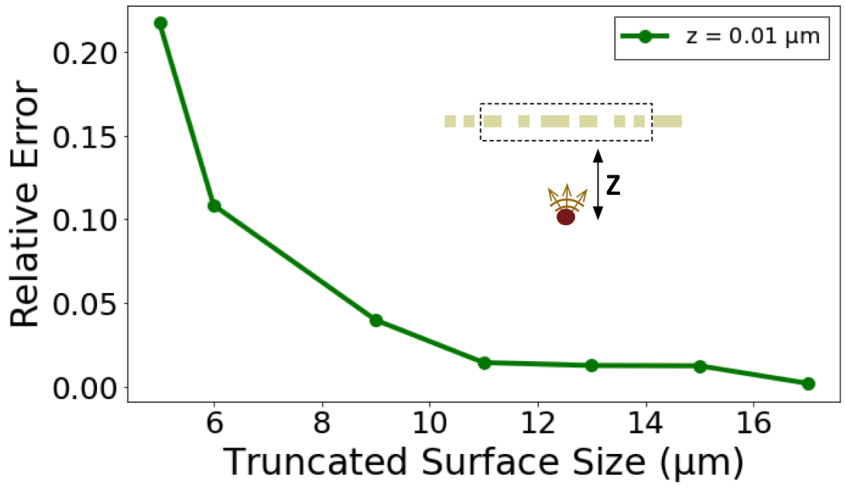


Figure 3.10: **Padding study for Huygens metasurface scatterer library.** Percent error in scattered field power versus spatial-extent of metasurface included in the simulation for a single jinc source placed 0.1 μm from the metasurface. The full metasurface is a 20 μm $\times$ 20 μm metalens with focal length of 14 μm composed of scatterers from the low aspect-ratio library in Fig. 3.5b (silicon cylinders with height 220 nm, radii range of 175-280 nm, lattice period of 666 nm, and background refractive index 1.66; source wavelength of 1340 nm – using scatterer library from [33]). The surface size on the x-axis of this convergence plot refers to the spatial-extent around the center of this metasurface that is included in the simulation. The y-axis relative error is computed assuming the simulation including the full metasurface is the converged result.

Then, in Fig. 3.11, we use our distributed gradient-based optimization to iteratively improve a cost function evaluating the amount of power within a spot at the focal plane for a 30 μ m $\times$ 30 μm Huygens metalens with focal-length 20 μm. The

distributed optimization was performed on 9 T4 GPUs with the metalens divided into 9 subregions (subregion size of 10 μm $\times$ 10 μ m, and padding size of 6 μ m). The metalens has a very high NA of 0.996 and the optimization improves the efficiency of the metalens by about 2 $\times$, giving a final efficiency of about 24%.

The forward simulations performed took an average of about 120 GPU-min and the gradient computations with respect to radius and position took an average of 150 GPU-min. Fig. 3.12 shows the timing of key stages for the forward simulation (a) and the adjoint gradient computation (b). The computation time remains constant across iterations, except for one peak in which 4 of the 9 T4 GPU workers had died and needed to be restarted. As a result, the computation time increased by about 40% until the workers were restarted. However, the optimization was still able to proceed when the workers died and the workers were able to be restarted without interrupting the optimization – this illustrates the fault tolerance of the RabbitMQ subregion distribution queue we use, as mentioned in Section 2.3. Additionally, Fig. 3.12b shows that the gradient with respect to scatterer position degrees of freedom is slower to compute than the gradient with respect to scatterer geometry degrees of freedom – this is due to the fact that a larger number of finite-difference derivatives are needed for the position gradient (Eq. 3.21 vs. Eq. 3.26).

3.8 Distributed optimization of multi-layered metalens

Designing multi-layered metasurfaces through the traditional periodic-approximation library approach is quite challenging due to the increased complexity of the unit-cells. The conventional design approach based on the lookup table causes significant degradation in the multilayered devices' performance due to the multiple scattering between

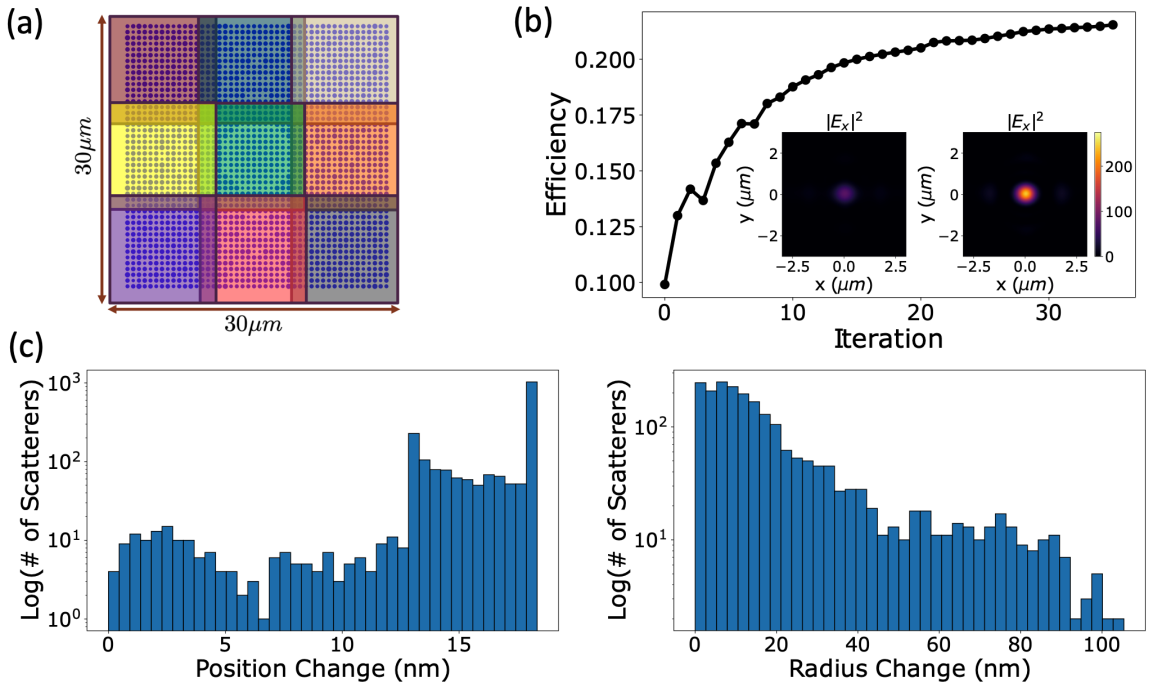


Figure 3.11: **Distributed Gradient-based optimization improvement of metalens design.** (**a**) Schematic of the cylindrical metasurface scatterers in the initial metalens design and the 9 subregions used for the distributed optimization (subregion size of 10 μm $\times$ 10 μ m, and padding size of 6 μ m). Fig. 3.10 is used to determine the padding size of 6 μ m. This initial metasurface is a 30 μm $\times$ 30 μm metalens with focal-length 20 μm designed from the low-aspect ratio scatterer library in Fig. 3.5(b) using the traditional metasurface design approach.(**b**) Lens efficiency versus optimization iteration, where lens efficiency is defined as the ratio of the power within a circle of radius 3 $\times$ FWHM in the focal plane to the power incident on the metasurface. The metalens is 15 μm $\times$ 15 μm in size, and is optimized for x-polarized light only. In 35 optimization iterations, the metalens efficiency is almost doubled. The inset shows the X-component of the electric field in the focal plane before optimization (left) and after optimization (right). (**c**) Histograms of the distance between the final scatterer positions and the initial scatterer positions (left) and the absolute radius difference between the final scatterer cylinders and the initial scatterer cylinders (right). As can be seen in these histograms, both the scatterer positions and radii change as a result of the optimization.

the layers and near-field interactions [59]. As a result, the design of multi-layered metasurfaces requires advanced optimization techniques to achieve high-performance

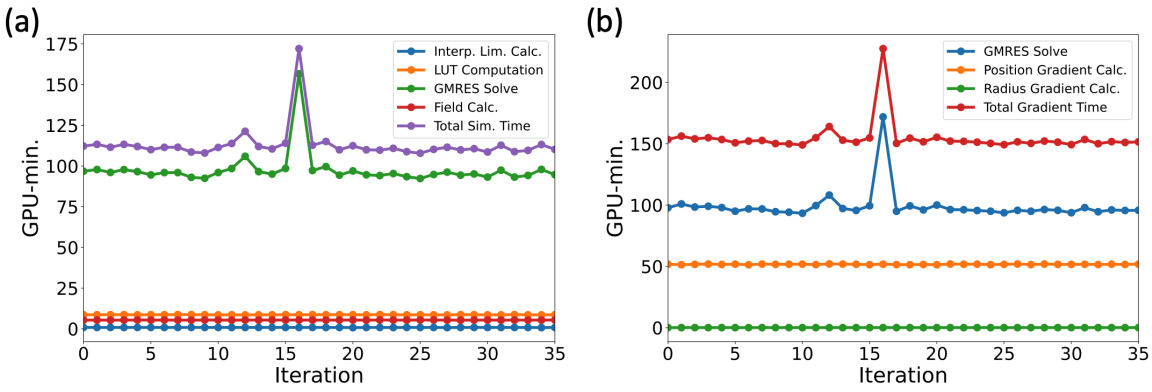


Figure 3.12: **Compute-time for the distributed gradient-based metalens optimization iterations.** (**a**) Computation time vs optimization iteration for the key stages of the forward simulation: *blue, orange* – computing the limits and values (respectively) for the Look-Up Tables (LUT) used to efficiently perform T-matrix simulation (Section 3.3); *green* – solving the linear system of equations for the scattered field coefficients (Eq. 3.8); *red* – computing the E and H fields from the scattered field coefficients for each desired detector point (Eq. 3.2); *purple* – total time for forward simulation.(**b**) Computation time vs optimization iteration for the key stages of the gradient calculation: *blue* – solving the linear system of equations for the adjoint vector (Eq. 3.19); *orange* – computing the required finite-difference derivatives and using the adjoint vector to calculate the gradient with respect to the x and y position of each scatterer center (Eq. 3.21); *green* – computing the required finite-difference derivatives and using the adjoint vector to compute the gradient with respect to the radius of each scatterer (Eq. 3.26); *red* – total time for all required gradient computation.

beyond single-layer metasurfaces.

Here, we demonstrate applying our distributed optimization to a multi-layered metasurface composed of higher aspect-ratio Si cylindrical scatterers embedded in SiO2. Since this scatterer library does not rely on overlapping electric and magnetic resonances to achieve full 2π phase coverage like the Huygens library from Section 3.7, a thicker set of scatterers (730 nm thickness, 670 nm square lattice constant, radii between 50-250 nm, source wavelength 1340 nm) is required because the phase coverage is now obtained non-resonantly through propagation in the scatterer.

To perform the distributed optimization, we first use the single jinc source spatial-locality study for this double-layer higher-aspect ratio silicon in SiO2 cylinder scatterer library to determine the padding size of $7.5\mu m$ (Fig. 3.13).

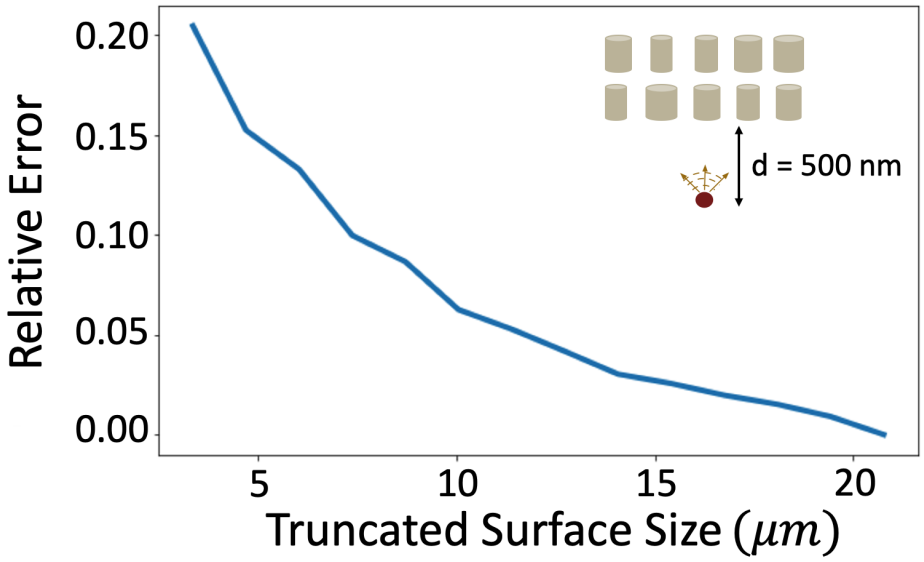


Figure 3.13: **Padding study showing spatial locality of the jinc source for the double-layer scatterer library.** Percent error in scattered field power versus spatial-extent of metasurface included in the simulation for a single jinc source placed 500 nm from the metasurface. The full metasurface is a 20 μm $\times$ 20 μm metasurface with focal length of 10 μm, and the surface size on the x-axis of this convergence plot refers to the spatial-extent around the center of this metasurface that is included in the simulation. The y-axis relative error is computed assuming the simulation including the full metasurface is the converged result.

We generated the initial condition for the optimization by adding a uniform array of nanoposts $1.5\mu m$ above a conventionally-designed metalens, with side length of $40\mu m$ and NA of 0.5 (Fig. 3.14a). We choose a subregion size of $7.5\mu m \times 7.5\mu m$ and use the padding size of $7.5\mu m$ determined in Fig. 3.13 – this results in 16 subregions that we distribute across 8 NVIDA T4 GPUs. The forward and adjoint simulations took an average of $\sim$960 GPU-min per iteration and we performed 18 iterations in total. One can boost the simulation speed by using the high-end GPUs and increasing

the number of the GPUs. In Fig. 3.14b, we can see the optimization improves the efficiency of the metalens by ∼12% in absolute value (∼ 20% in relative value), giving a final efficiency of about ∼ 77%.

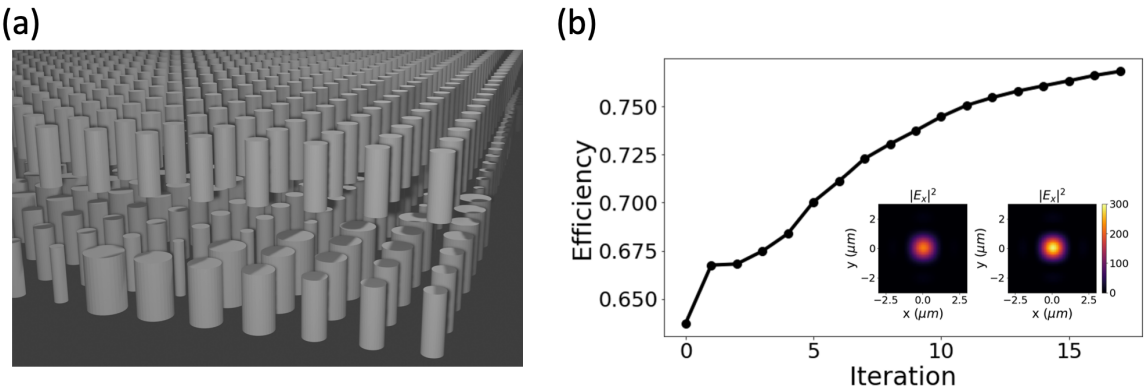


Figure 3.14: **Distributed gradient-based double-layer metalens optimization.** (**a**) Schematic of the initial double-layer metalens design. The metalens scatterers are Si cylinders with radii between 50-250 nm, and thickness and lattice constant of 730 nm and 670 nm, respectively. The illumination wavelength is 1340 nm and the background material is glass. The side length of the device and the NA of the metalens were fixed at 40 μm and 0.5, respectively. We created the bottom layer using the conventional metasurface lens design. Then, we added the upper layer composed of uniform array of nanoposts having a radius of 130 nm. The distance between the centers of the layers was 1.5 μm. (**b**) Absolute efficiency of the double-layered metalens versus optimization iteration, where the efficiency is defined as the ratio of the power within a circle of radius 3 × FWHM in the focal plane to the power incident on the metasurface. All positions and radii of scatterers in both layers are simultaneously optimized using the gradients in order to improve the efficiency. The inset shows the intensity of the x-component of the electric field in the focal plane before optimization (left) and after optimization (right).

Combining the multi-GPU gradient computation with the multi-GPU forward simulation, this distributed optimizer uniquely enables gradient-based optimization of the large scale devices over the many degrees of freedom afforded by arbitrarily large multi-layered metasurfaces.

3.9 Limitations

Although implementing the T-matrix approach on the spherical harmonic functions results in a compact linear system of equations (Sec. 3.1) that accurately captures scatterer-scatterer interactions (Sec. 3.4), it does impose two main limitations on the types of metasurface systems that can be simulated. The first limitation is that, because we are working on the spherical harmonic basis functions, the bounding spheres of adjacent scatterers must not intersect. For low aspect-ratio scatterers like the Huygens library in Section 3.7, neighboring scatterers may be put very close together without their bounding spheres intersecting (Fig. 3.15a). However, for high aspect-ratio scatterers like the silicon scatterers in Section 3.8, neighboring scatterers must be kept far away from each other to keep the bounding spheres separate (Fig. 3.15b). This scatterer-separation constraint greatly restricts the degrees of freedom, hence limiting the efficiencies that can be obtained – for example, this is part of the reason the higher aspect-ratio metalens in Sec. 3.8 only saw a focusing efficiency improvement of about 1.2× after optimization, while the focusing efficiency of the low aspect-ratio Huygens metalens in Sec. 3.7 was doubled after optimization.

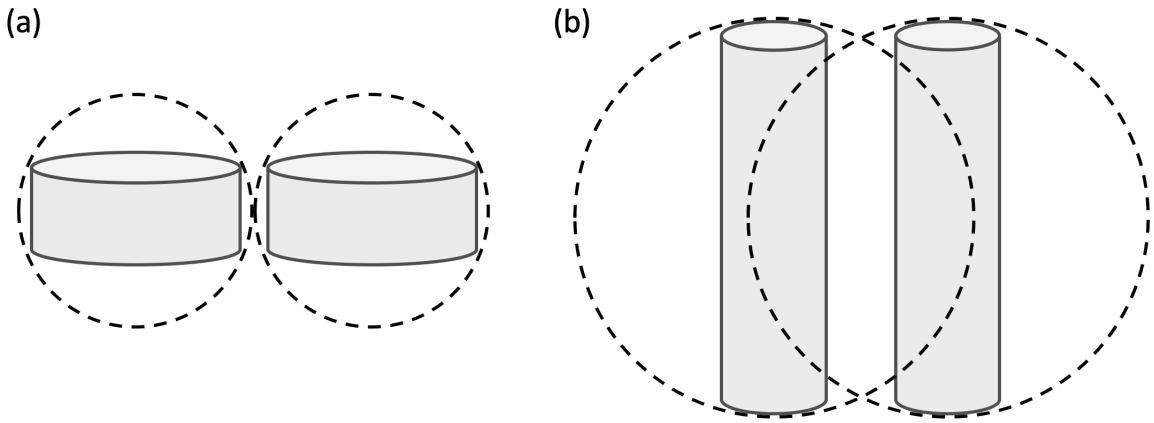


Figure 3.15: **Aspect-ratio limitation resulting from spherical harmonic basis functions.** (**a**) Schematic of the bounding spheres of two adjacent low aspect-ratio scatterers. These scatterers can be placed arbitrarily close together without their bounding spheres overlapping. (**b**) Schematic of the bounding spheres of two adjacent high aspect-ratio scatterers. These scatterers must maintain a large separation to ensure their bounding spheres do not overlap.

One way of addressing this limitation is to move from working on the vector spherical basis functions to working on the vector spheroidal basis functions. The advantage of the vector spheroidal basis functions is that the focal length of the coordinate system can be chosen such that the bounding spheroid allows high aspect-ratio scatterers to be placed close together (Fig. 3.16). An approach for implementing the T-matrix method on the vector spheroidal basis functions is detailed in Appendix A.4. However, as can be seen in Appendix A.4.3 and A.4.4, implementing efficient computation of the vector spheroidal basis functions is a major undertaking.

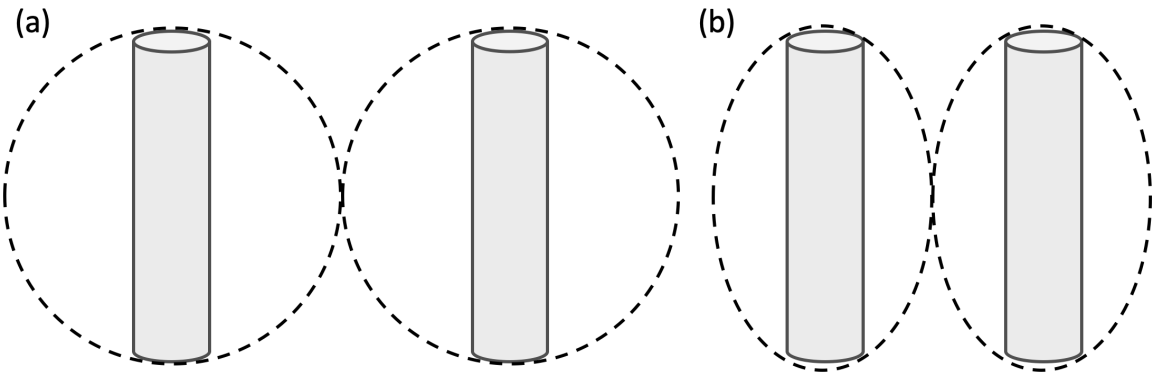


Figure 3.16: **Changing from spherical to spheroidal basis functions to address aspect-ratio limitation.** (**a**) Schematic showing the minimum distance between two adjacent high aspect-ratio scatterers when using spherical harmonic basis functions. The scatterers must be far enough apart that their bounding spheres do not overlap. (**b**) Schematic of the bounding spheroids of two adjacent high aspect-ratio scatterers. With spheroidal basis functions, the focal length of the coordinate system can be chosen per the scatterer geometry such that arbitrary aspect-ratios and scatterer separations can be handled.

In addition to the aspect-ratio limitation, substrates and any other kind of infinite surface are extremely challenging to handle with the spherical or the spheroidal basis functions. One approach to handling substrates is to use Sommerfield integrals and associated layer Green's functions [14] but this is very computationally complex and expensive. In order to retain the benefit of the computational efficiency of our T-matrix implementation, this single-node solver should be applied to lower aspect-ratio scatterer libraries embedded in a homogeneous background material. For a more general-purpose single-node solver, we also investigate using FDTD in Chapter 4.

Chapter 4

FDTD as single-node simulator

4.1 Motivation

We explore FDTD as a single-node simulator because it is a flexible, general-purpose solver that can handle arbitrary geometries and substrates – hence addressing the limitations that come with the T-matrix method on spherical harmonic functions in Chapter 3. As it is a very general simulation approach, there are a number of commercial FDTD implementations (including Lumerical [84], Flexcompute [40], and Comsol [24]) and some open-source FDTD implementations (including meep [66]). These implementations vary as far as ease-of-use, flexibility, and computational efficiency. We choose to use meep [66] for this work because of its open-source nature, which is helpful for debugging purposes.

4.2 Low-overhead distribution with FDTD

Most existing FDTD implementations do offer parallelization across CPU or GPU compute nodes. This parallelization is done by separating the simulation spatial domain into regions, each of which are handled by a given compute-node. However,

because computation of finite-difference derivatives for spatial points on the boundary of each region requires information from spatial points in adjacent regions, the compute-nodes must communicate information at every iteration of the simulation (Fig. 4.1a). Hence, this standard approach is not low-overhead, which can result in the problem depicted in Fig. 4.1b in which adding more compute nodes eventually increases the compute time because the node communication becomes the dominant cost. Additionally, FDTD has simulation size limitations imposed by the memory necessary for the spatial and temporal evolution of the fields. Thus, scaling FDTD to the mm and cm scales required for simulation and optimization of practical metasurfaces necessitates a different approach to parallelization.

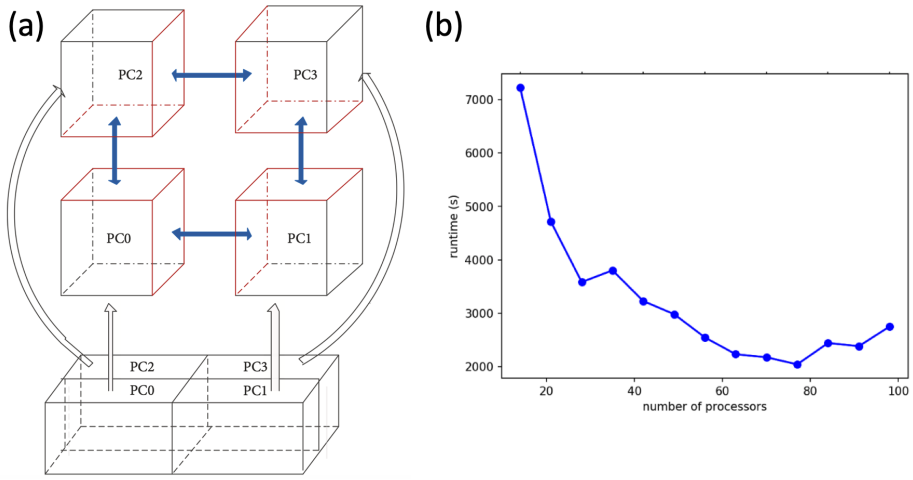


Figure 4.1: **Standard FDTD spatial parallelization approach.** (**a**) Schematic of the typical spatial region parallelization approach employed with FDTD (adapated from [80]). The red edges of each spatial region indicate that these boundary points need information from the adjacent spatial regions in order to compute the necessary finite differences. Thus, the compute nodes must all communicate with their adjacent nodes at every simulation iteration. (**b**) Example of the phenomenon where adding more compute nodes eventually results in an increase in the simulation runtime because the cost of the node communications required at every iteration becomes dominant
(https://meep.readthedocs.io/en/latest/Parallel_Meep).

In order to enable the FDTD solver to scale to very large-area metasurface simulations, we propose and demonstrate using FDTD as the subregion solver in our distribution approach detailed in Chapter 2 (Fig. 4.2). This allows the FDTD simulations to remain at the size scales that are tractable from a memory standpoint and that can be efficiently further parallelized through the standard spatial parallelization approach.

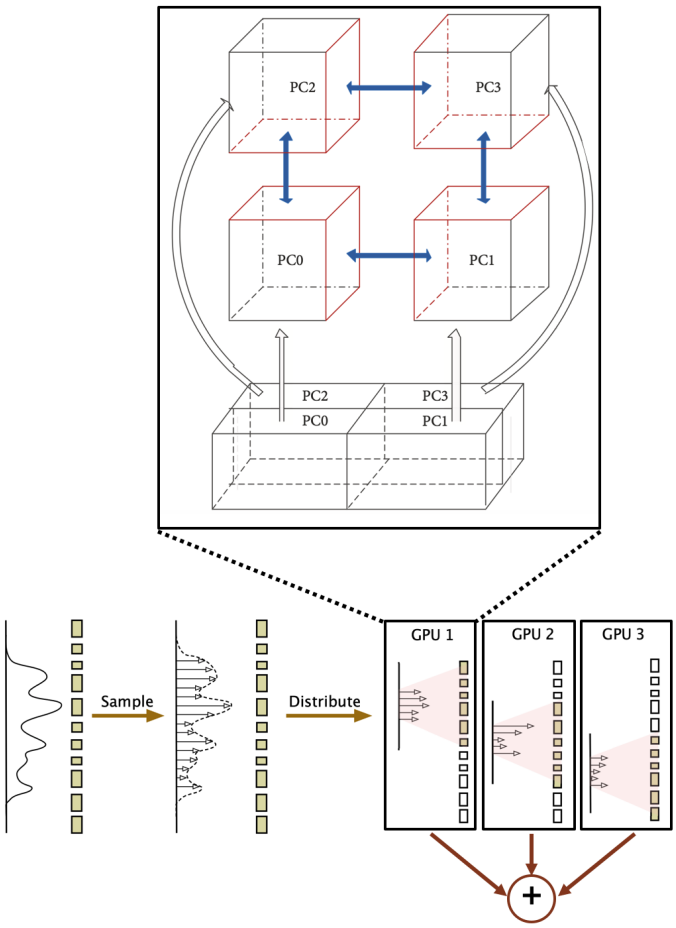


Figure 4.2: **FDTD as a subregion solver.** Schematic of using FDTD as the subregion solver in our low-overhead distribution approach detailed in Chapter 2. The incident field is sampled at the Nyquist sampling rate, the resulting jinc sources are grouped, and the metasurface is split into subregions based on the spatial-locality of the jinc sources. These subregions are simulated in parallel using FDTD with the standard spatial parallelization also applied.

4.3 Jinc source as a Total-Field Scattered-Field

In order to employ FDTD as the subregion solver in our low-overhead distribution scheme, we must formulate the jinc source as an FDTD source.

4.3.1 Nyquist Sampling of Time-dependent Transverse Incident Field

Consider an incident field propagating along the $+z$ axis and described by its transverse components as a function of transverse coordinate (x, y) at $z = z_t$. Then, at $z = z_t$, we have:

$$\mathbf{E}^T_{\text{inc}}(x, y, z = z_t, t) = \frac{1}{(2\pi)^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{E}^T_{\text{inc}}(k_x, k_y, \omega) e^{i(k_x x + k_y y - \omega t)} dk_x dk_y d\omega \quad (4.1)$$

with dispersion relation:

$$k_x^2 + k_y^2 + k_z^2 = \frac{\omega^2}{c^2} \quad (4.2)$$

If we assume that this incident field is produced by a source that is either far from the metasurface or paraxial, we know it is only made up of propagating components. Since there are no evanescent components, we know:

$$k_x^2 + k_y^2 \leq \frac{\omega^2}{c^2} \quad (4.3)$$

Assume this incident field is a pulse, so it has a finite frequency range decided by the pulse duration:

$$\mathbf{E}^T_{\text{inc}}(k_x, k_y, \omega) = 0 \text{ if } \omega \notin [\omega_{min}, \omega_{max}] \quad (4.4)$$

Then we know all k_x and k_y satisfy $k_x^2 + k_y^2 \leq \omega_{max}^2/c^2$, and Eq. 4.1 becomes:

$$\mathbf{E}^T_{\text{inc}}(x, y, z = z_t, t) = \frac{1}{(2\pi)^3} \int_{k_x^2 + k_y^2 \leq \frac{\omega_{max}^2}{c^2}} \int_{\omega_{min}}^{\omega_{max}} \mathbf{E}^T_{\text{inc}}(k_x, k_y, \omega) e^{i(k_x x + k_y y - \omega t)} dk_x dk_y d\omega \quad (4.5)$$

So, at every time t, this incident field is bandlimited in k-space. Since the incident field is bandlimited in k-space, we can apply the Nyquist sampling theorem to perfectly reconstruct the field from samples of the field as long as we sample finely enough. For a given ω, the Nyquist sampling rate would be $\frac{\lambda}{2}$. So, with our range of frequencies $\omega \in [\omega_{min}, \omega_{max}]$, the finest sampling rate will be determined by ω_{max} since that corresponds to the smallest wavelength. Since the Nyquist sampling theorem allows for sampling finer than the Nyquist rate, we will sample everything at this finest sampling rate. Applying the Nyquist sampling theorem with this sampling rate, we can write our field in terms of its samples as:

$$\mathbf{E}^T_{\text{inc}}(x, y, z = z_t, t) = \sum_{i,j=-\infty}^{\infty} \mathbf{E}^T_{\text{inc}}(x_i, y_j, z = z_t, t) \frac{j_1(k_{max}\rho_{i,j})}{k_{max}\rho_{i,j}} \tag{4.6}$$

where $x_i = \frac{i\lambda_{min}}{2}$, $y_j = \frac{j\lambda_{min}}{2}$, $\rho_{i,j} = \sqrt{(x - x_i)^2 + (y - y_j)^2}$, and $\mathbf{E}^T_{\text{inc}}(x_i, y_j, z = z_t, t)$ are the field samples.

4.3.2 TFSF Source with Jinc Formulation for FDTD

Using the equivalence principle in Eq. 4.7, we can formulate the Nyquist-sampled incident field from Eq. 4.6 as a Total-Field Scattered-Field (TFSF) source for FDTD.

$$\begin{bmatrix} \mathbf{J} \\ \mathbf{K} \end{bmatrix} = \delta(\partial\Omega) \begin{bmatrix} \mathbf{n} \times \mathbf{H} \\ -\mathbf{n} \times \mathbf{E} \end{bmatrix} \tag{4.7}$$

Eq. 4.7 means that by specifying the electric and magnetic currents given by the tangential components of the desired electric and magnetic fields on the planes of a box surrounding the metasurface (Fig 4.3), we can exactly produce the desired electric and magnetic fields of the jinc source inside the box with no diffraction effects.

In Eq. 4.6, we expressed $\mathbf{E}^T_{\text{inc}}(x, y, z = z_t, t)$ for the jinc source formulation. We

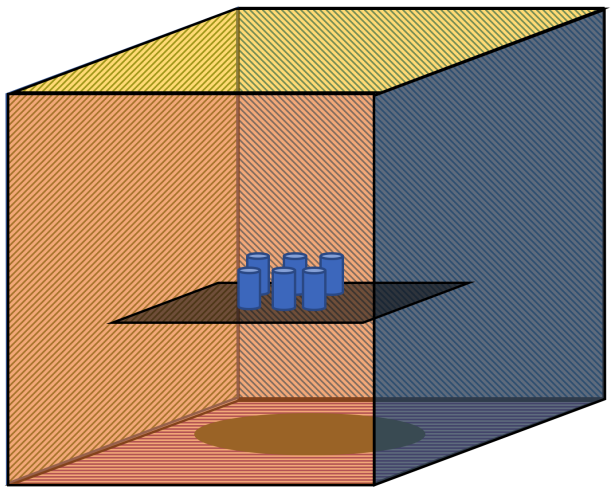


Figure 4.3: **Total-field scattered-field (TFSF) FDTD source schematic.** By the equivalence principle in Eq. 4.7, the tangential components of desired electric and magnetic fields on a given plane can be used to specify a current source that will produce the desired fields. However, if only a single-plane is specified, diffraction effects will occur at the edges of the plane. By instead specifying 6 TFSF source planes, these diffraction effects can be avoided because the fields cancel out everywhere except inside the box. Thus, by specifying a TFSF source box enclosing a given metasurface subregion, the desired incident field for a single or group of jinc sources can be generated.

now need to determine full expressions for $\mathbf{E}_{\text{inc}}(x, y, z, t)$ and $\mathbf{H}_{\text{inc}}(x, y, z, t)$ in order to supply the required field components on all 6 planes of the TFSF box.

4.3.3 Determining full vector expression for E and H Fields

In Eq. 4.6, we know how to compute $\mathbf{E}^T_{\text{inc}}(x, y, z = z_t, t)$ because we sample our desired incident-field description on the tangential plane located at $z = z_t$. As can be seen in Fig. 4.3, we must have expressions for the $\hat{x}$, $\hat{y}$, and $\hat{z}$ incident-field components as functions of (x, y, z) – thus, we must compute the normal component of the field from this known tangential component and generalize the tangential component expression

so it can be computed at any z in the domain.

General Expression for $\hat{x}$ and $\hat{y}$ E-field Components

Writing the jinc source in its Fourier representation (which is a sum of plane waves), Equation 4.6 becomes:

$$\mathbf{E}^T_{\text{inc}}(x,y,z=z_t,t) = \sum_{i,j=-\infty}^{\infty} \mathbf{E}^T_{\text{inc}}(x_i,y_j,z=z_t,t)\frac{1}{2\pi}\left(\frac{\pi}{k_{max}}\right)^2 \int_{k_x^2+k_y^2\leq k_{max}^2} e^{i(k_x(x-x_i)+k_y(y-y_j))}dk_xdk_y \tag{4.8}$$

Fourier transforming $\mathbf{E}^T_{\text{inc}}(x_i,y_j,z=z_t,t)$, we have:

$$\mathbf{E}^T_{\text{inc}}(x,y,z=z_t,t) = \sum_{i,j=-\infty}^{\infty}\left[\frac{1}{2\pi}\int_{\omega_{min}}^{\omega_{max}}\left(\mathbf{E}^T_{\text{inc}}(x_i,y_j,z=z_t,\omega)e^{-i\omega t}\left(\frac{\pi}{2{k_{max}}^2}\int_{k_x^2+k_y^2\leq k_{max}^2} e^{i(k_x(x-x_i)+k_y(y-y_j))}dk_xdk_y\right)d\omega\right)\right] \tag{4.9}$$

We can then rewrite Eq. 4.9 so it can be evaluated at arbitrary z locations (rather than just at $z=z_t$) by introducing a z-propagation term on the plane wave components:

$$\mathbf{E}^T_{\text{inc}}(x,y,z,t) = \sum_{i,j=-\infty}^{\infty}\left[\frac{1}{2\pi}\int_{\omega_{min}}^{\omega_{max}}\left(\mathbf{E}^T_{\text{inc}}(x_i,y_j,z=z_t,\omega)e^{-i\omega t}\left(\frac{\pi}{2{k_{max}}^2}\int_{k_x^2+k_y^2\leq k_{max}^2} e^{i(k_x(x-x_i)+k_y(y-y_j)+k_z(z-z_t))}dk_xdk_y\right)d\omega\right)\right] \tag{4.10}$$

Now, we make the following coordinate transformation to polar coordinates in order to simplify the integral over k.

$$k_x = kcos\theta \ , \ k_y = ksin\theta \ , \ dk_xdk_y = kdkd\theta \tag{4.11}$$

Applying this coordinate transformation from Eq. 4.11, Eq. 4.9 becomes:

$$\mathbf{E}^T_{\text{inc}}(x,y,z,t) = \sum_{i,j=-\infty}^{\infty} \left[\frac{1}{2\pi} \int_{\omega_{min}}^{\omega_{max}} \left(\mathbf{E}^T_{\text{inc}}(x_i, y_j, z = z_t, \omega) e^{-i\omega t} \right. \right. \tag{4.12}$$
$$\left. \left. \left(\frac{\pi}{2{k_{max}}^2} \int_0^{k_{max}} \int_0^{2\pi} e^{i\left(kcos\theta(x-x_i)+ksin\theta(y-y_j)+\sqrt{\left(\frac{\omega}{c}\right)^2-k^2}(z-z_t)\right)} k d\theta dk \right) d\omega \right) \right]$$

Focusing first on the integral over θ, let us define I_θ:

$$I_\theta = \int_0^{2\pi} e^{i(kcos\theta(x-x_i)+ksin\theta(y-y_j))} d\theta \tag{4.13}$$

We rewrite the $x - x_i$ and $y - y_j$ as follows:

$$x - x_i = rcos\alpha \ , \ y - y_j = rsin\alpha \tag{4.14}$$
$$r = \sqrt{(x-x_i)^2 + (y-y_j)^2} \ , \ \alpha = tan^{-1}\left(\frac{y-y_j}{x-x_i}\right) \tag{4.15}$$

Plugging Eq. 4.14 into Eq. 4.13, we have:

$$I_\theta = \int_0^{2\pi} e^{i(krcos\theta cos\alpha + krsinsin\alpha)} d\theta = \int_0^{2\pi} e^{ikr(cos(\theta-\alpha))} d\theta \tag{4.16}$$

To solve this integral, we use the following integral formula for Bessel functions of the first kind:

$$J_n(z) = \frac{1}{\pi}\int_0^{\pi} cos(zsin\theta - n\theta) d\theta = \frac{1}{2\pi i^n}\int_0^{2\pi} e^{izcos\phi} e^{in\phi} d\phi \text{ and } J_{-n}(x) = (-1)^n J_n(x) \tag{4.17}$$

Matching the integral in Eq. 4.16 with the Bessel integral form in Eq. 4.17, we

can write:

$$I_\theta = 2\pi J_0(kr) \tag{4.18}$$

Plugging the result in Eq. 4.18 into Eq. 4.12, we have:

$$\mathbf{E}^T_{\text{inc}}(x,y,z,t) = \frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty} \tag{4.19}$$
$$\left[\int_{\omega_{min}}^{\omega_{max}} \left(\mathbf{E}^T_{\text{inc}}(x_i,y_j,z=z_t,\omega)e^{-i\omega t}\left(\frac{\pi^2}{{k_{max}}^2}\int_0^{k_{max}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega}{c}\right)^2-k^2}(z-z_t)\right)}kdk\right)d\omega\right)\right]$$

General Expression for $\hat{z}$ E-field Components

Now, we want to calculate $\mathbf{E}^n_{\text{inc}}$ from $\mathbf{E}^T_{\text{inc}}$. Since we have written $\mathbf{E}^T_{\text{inc}}$ as a superposition of plane waves (Eq. 4.10), we can use the divergence theorem to determine the coefficients of $\mathbf{E}^n_{\text{inc}}$ from $\mathbf{E}^T_{\text{inc}}$ (because plane waves have no normal component, since they are propagating planes):

$$\mathbf{k}\cdot\mathbf{A} = 0 \text{ , so, } A_n = -\frac{\mathbf{k}_t\cdot\mathbf{A}_t}{k_n} \tag{4.20}$$

The normal component for us is the z direction, so we have:

$$A_z = -\frac{(k_xA_x+k_yA_y)}{k_z} = -\frac{(k_xA_x+k_yA_y)}{\sqrt{\frac{\omega^2}{c^2}-k_x^2-k_y^2}} \tag{4.21}$$

Plugging the expression for $\mathbf{E}^T_{\text{inc}}$ from Eq. 4.10 into Eq. 4.21 as $\mathbf{A}_t$, we have:

$$\mathbf{E}^Z_{\text{inc}}(x,y,z,t) = \frac{1}{(2\pi)^2}\sum_{i,j=-\infty}^{\infty}\Bigg[\int_{\omega_{min}}^{\omega_{max}} e^{-i\omega t}\int_{k_x^2+k_y^2\leq k_{max}^2} \tag{4.22}$$
$$-\frac{(k_x E^T_{inc,x}(x_i,y_i,z=z_t,\omega)+k_y E^T_{inc,y}(x_i,y_i,z=z_t,\omega))}{\sqrt{\frac{\omega^2}{c^2}-k_x^2-k_y^2}} e^{i\left(k_x(x-x_i)+k_y(y-y_j)+\sqrt{\frac{\omega^2}{c^2}-k_x^2-k_y^2}(z-z_t)\right)}$$
$$dk_x dk_y d\omega\Bigg]\hat{z}$$

Applying the coordinate transformation from Eq. 4.11, Eq. 4.22 becomes:

$$\mathbf{E}^Z_{\text{inc}}(x,y,z,t) = \frac{1}{(2\pi)^2}\sum_{i,j=-\infty}^{\infty}\Bigg[\int_{\omega_{min}}^{\omega_{max}} e^{-i\omega t}\int_0^{k_{max}} -\frac{k^2}{\sqrt{\frac{\omega^2}{c^2}-k^2}} \tag{4.23}$$
$$\int_0^{2\pi}\left(cos\theta E^T_{inc,x}(x_i,y_i,z=z_t,\omega)+sin\theta E^T_{inc,y}(x_i,y_i,z=z_t,\omega)\right)$$
$$e^{i\left(kcos\theta(x-x_i)+ksin\theta(y-y_j)+\sqrt{\frac{\omega^2}{c^2}-k^2}(z-z_t)\right)} d\theta dk d\omega\Bigg]\hat{z}$$

Focusing first on the integral over θ, let us define $I_{\theta,x}$ and $I_{\theta,y}$ as:

$$I_{\theta,x} = \int_0^{2\pi} cos\theta e^{i(kcos\theta(x-x_i)+ksin\theta(y-y_j))}d\theta \tag{4.24}$$
$$I_{\theta,y} = \int_0^{2\pi} sin\theta e^{i(kcos\theta(x-x_i)+ksin\theta(y-y_j))}d\theta \tag{4.25}$$

Plugging Eq. 4.14 into Eq. 4.24, we now have:

$$I_{\theta,x} = \int_0^{2\pi} cos\theta e^{i(krcos\theta cos\alpha+krsinsin\alpha)}d\theta = \int_0^{2\pi} cos\theta e^{ikr(cos(\theta-\alpha))}d\theta \tag{4.26}$$
$$I_{\theta,y} = \int_0^{2\pi} sin\theta e^{i(krcos\theta cos\alpha+krsinsin\alpha)}d\theta = \int_0^{2\pi} sin\theta e^{ikr(cos(\theta-\alpha))}d\theta \tag{4.27}$$

Defining $\theta' = \theta - \alpha$ and using the formulas $cos\theta = \frac{1}{2}\left(e^{i\theta}+e^{-i\theta}\right)$ and $sin\theta =$

$\frac{1}{2i}\left(e^{i\theta}-e^{-i\theta}\right)$, we can write Eq. 4.26 as:

$$I_{\theta,x}=\int_{-\alpha}^{2\pi-\alpha}\frac{1}{2}\left(e^{i(\theta'+\alpha)}+e^{-i(\theta'+\alpha)}\right)e^{ikrcos(\theta')}d\theta'= \tag{4.28}$$
$$\frac{1}{2}\left[\left(e^{i\alpha}\int_{-\alpha}^{2\pi-\alpha}e^{i\theta'}e^{ikrcos(\theta')}d\theta'\right)+\left(e^{-i\alpha}\int_{-\alpha}^{2\pi-\alpha}e^{-i\theta'}e^{ikrcos(\theta')}d\theta'\right)\right]$$
$$I_{\theta,y}=\int_{-\alpha}^{2\pi-\alpha}\frac{1}{2i}\left(e^{i(\theta'+\alpha)}-e^{-i(\theta'+\alpha)}\right)e^{ikrcos(\theta')}d\theta'= \tag{4.29}$$
$$\frac{1}{2i}\left[\left(e^{i\alpha}\int_{-\alpha}^{2\pi-\alpha}e^{i\theta'}e^{ikrcos(\theta')}d\theta'\right)-\left(e^{-i\alpha}\int_{-\alpha}^{2\pi-\alpha}e^{-i\theta'}e^{ikrcos(\theta')}d\theta'\right)\right]$$

Matching the integrals in Eq. 4.28 with the Bessel integral form in Eq. 4.17, we can write:

$$I_{\theta,x}=\frac{1}{2}\left[\left(e^{i\alpha}2\pi iJ_1(kr)\right)+\left(e^{-i\alpha}\frac{2\pi}{i}J_{-1}(kr)\right)\right] \tag{4.30}$$
$$=2\pi iJ_1(kr)\left[\frac{1}{2}\left(e^{i\alpha}+e^{-i\alpha}\right)\right]=2\pi icos(\alpha)J_1(kr)$$
$$I_{\theta,y}=\frac{1}{2i}\left[\left(e^{i\alpha}2\pi iJ_1(kr)\right)-\left(e^{-i\alpha}\frac{2\pi}{i}J_{-1}(kr)\right)\right] \tag{4.31}$$
$$=2\pi iJ_1(kr)\left[\frac{1}{2i}\left(e^{i\alpha}-e^{-i\alpha}\right)\right]=2\pi isin(\alpha)J_1(kr)$$

Plugging the expressions from Eq. 4.30 into Eq. 4.23, we now have:

$$\mathbf{E}^Z_{\text{inc}}(x,y,z,t)=\frac{i}{(2\pi)}\sum_{i,j=-\infty}^{\infty}\left[\int_{\omega_{min}}^{\omega_{max}}e^{-i\omega t}\int_0^{k_{max}}\right. \tag{4.32}$$
$$\left.-\frac{J_1(kr)k^2}{\sqrt{\frac{\omega^2}{c^2}-k^2}}e^{i\sqrt{\frac{\omega^2}{c^2}-k^2}(z-z_t)}\left(cos\alpha E^T_{inc,x}(x_i,y_i,z=z_t,\omega)+sin\alpha E^T_{inc,y}(x_i,y_i,z=z_t,\omega)\right)dkd\omega\right]\hat{z}$$

$$\tag{4.33}$$

The integrand in Eq. 4.30 becomes singular when $k=\frac{\omega}{c}$. To remove this singularity, we will eventually do a variable transformation. We begin by first splitting the

k-integral into two integrals:

$$\mathbf{E}^{Z}_{\text{inc}}(x,y,z,t) = \frac{i}{(2\pi)}\sum_{i,j=-\infty}^{\infty}\Bigg[\int_{\omega_{min}}^{\omega_{max}} e^{-i\omega t}\left(cos\alpha E^{T}_{inc,x}(x_i,y_i,z=z_t,\omega)+sin\alpha E^{T}_{inc,y}(x_i,y_i,z=z_t,\omega)\right) \tag{4.34}$$

$$\left(\int_0^{\frac{\omega}{c}} -\frac{J_1(kr)k^2}{\sqrt{\frac{\omega^2}{c^2}-k^2}}e^{i\sqrt{\left(\frac{\omega}{c}\right)^2-k^2}(z-z_t)}dk + \int_{\frac{\omega}{c}}^{k_{max}} \frac{iJ_1(kr)k^2}{\sqrt{k^2-\frac{\omega^2}{c^2}}}e^{-\sqrt{k^2-\left(\frac{\omega}{c}\right)^2}(z-z_t)}dk\right)d\omega\Bigg]\hat{z}$$

Now, we introduce the following variables that we will use to transform the two k-integrals in Eq. 4.34 such that we have no discontinuity:

$$k = \frac{\omega}{c}sin\phi \text{ and } dk = \frac{\omega}{c}cos\phi d\phi \tag{4.35}$$

$$k = \frac{\omega}{c}sec\psi \text{ and } dk = \frac{\omega}{c}sec\psi tan\psi d\psi \tag{4.36}$$

Plugging the trig substitutions from Eq. 4.35 into Eq. 4.34, we have:

$$\mathbf{E}^{Z}_{\text{inc}}(x,y,z,t) = \frac{i}{(2\pi)}\sum_{i,j=-\infty}^{\infty}\Bigg[\int_{\omega_{min}}^{\omega_{max}}\left(\left(\frac{\omega}{c}\right)^2 e^{-i\omega t}\right. \tag{4.37}$$

$$\left(cos\alpha E^{T}_{inc,x}(x_i,y_i,z=z_t,\omega)+sin\alpha E^{T}_{inc,y}(x_i,y_i,z=z_t,\omega)\right)\Big(\tag{4.38}$$

$$\int_0^{\frac{\pi}{2}} -J_1\left(\frac{\omega r sin\phi}{c}\right)e^{i\frac{\omega}{c}cos\phi(z-z_t)}sin^2\phi d\phi +$$

$$\left.\left.\int_0^{arcsec\left(\frac{\omega_{max}}{\omega}\right)} iJ_1\left(\frac{\omega r sec\psi}{c}\right)e^{-\frac{\omega}{c}tan\psi(z-z_t)}sec^3\psi d\psi\right)d\omega\right)\Bigg]\hat{z}$$

Computation Approach for E-field Components

As we have derived in Eq. 4.19 and Eq. 4.37, the full expressions for our E-field jinc source as a function of (x,y,z,t) are:

$$
\begin{aligned}
&\mathbf{E}_{\text{inc}}(x,y,z,t) = \\
&\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\Bigg[\int_{\omega_{min}}^{\omega_{max}}\Bigg(E_{\text{inc,x}}(x_i,y_j,z=z_t,\omega)e^{-i\omega t}\Big(\\
&\frac{\pi^2}{{k_{max}}^2}\int_0^{k_{max}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega}{c}\right)^2-k^2}(z-z_t)\right)}kdk\Big)d\omega\Bigg)\Bigg]\hat{x} \\
&+\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\Bigg[\int_{\omega_{min}}^{\omega_{max}}\Bigg(E_{\text{inc,y}}(x_i,y_j,z=z_t,\omega)e^{-i\omega t}\Big(\\
&\frac{\pi^2}{{k_{max}}^2}\int_0^{k_{max}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega}{c}\right)^2-k^2}(z-z_t)\right)}kdk\Big)d\omega\Bigg)\Bigg]\hat{y} \\
&+\frac{i}{(2\pi)}\sum_{i,j=-\infty}^{\infty}\Bigg[\int_{\omega_{min}}^{\omega_{max}}\Bigg(\left(\frac{\omega}{c}\right)^2 e^{-i\omega t} \\
&\Big(cos\alpha E^T_{inc,x}(x_i,y_i,z=z_t,\omega)+sin\alpha E^T_{inc,y}(x_i,y_i,z=z_t,\omega)\Big)\Big(\\
&\int_0^{\frac{\pi}{2}} -J_1\left(\frac{\omega r sin\phi}{c}\right)e^{i\frac{\omega}{c}cos\phi(z-z_t)}sin^2\phi d\phi + \\
&\int_0^{arcsec\left(\frac{\omega_{max}}{\omega}\right)} iJ_1\left(\frac{\omega r sec\psi}{c}\right)e^{-\frac{\omega}{c}tan\psi(z-z_t)}sec^3\psi d\psi\Big)d\omega\Bigg)\Bigg]\hat{z}
\end{aligned}
\tag{4.39}
$$

For simpler and more efficient implementation, we will make an approximation where we only evaluate the k-integral in Eq. 4.39 at $\omega = \omega_{cen}$, so that we can pull it out of the integral over ω and have the remaining ω-integral be the inverse Fourier transform of the incident field samples. This approximation should be valid for narrow-band sources or when we only care about the response at a single frequency. With this approximation, the full expressions for our E-field jinc source as a function

of (x, y, z, t) become:

$$\mathbf{E}_{\text{inc}}(x,y,z,t) = \tag{4.40}$$
$$\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\left[\left(\frac{\pi^2}{{k_{max}}^2}\int_0^{k_{max}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega_{cen}}{c}\right)^2-k^2}(z-z_t)\right)}kdk\right)\right.$$
$$\left.\int_{\omega_{min}}^{\omega_{max}}\left(E_{\text{inc,x}}(x_i,y_j,z=z_t,\omega)e^{-i\omega t}d\omega\right)\right]\hat{x}$$
$$+\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\left[\left(\frac{\pi^2}{{k_{max}}^2}\int_0^{k_{max}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega_{cen}}{c}\right)^2-k^2}(z-z_t)\right)}kdk\right)\right.$$
$$\left.\int_{\omega_{min}}^{\omega_{max}}\left(E_{\text{inc,y}}(x_i,y_j,z=z_t,\omega)e^{-i\omega t}d\omega\right)\right]\hat{y}$$
$$+\frac{i}{(2\pi)}\sum_{i,j=-\infty}^{\infty}\left[\left(\frac{\omega_{cen}}{c}\right)^2\left(\int_0^{\frac{\pi}{2}} -J_1\left(\frac{\omega_{cen}rsin\phi}{c}\right)e^{i\frac{\omega_{cen}}{c}cos\phi(z-z_t)}sin^2\phi d\phi\right)\right.$$
$$\left.\left(cos\alpha\int_{\omega_{min}}^{\omega_{max}} e^{-i\omega t}E_{inc,x}(x_i,y_i,z=z_t,\omega)d\omega + sin\alpha\int_{\omega_{min}}^{\omega_{max}} e^{-i\omega t}E_{inc,y}(x_i,y_i,z=z_t,\omega)d\omega\right)\right]\hat{z}$$
$$=\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\left[\left(\frac{\pi^2}{{k_{max}}^2}\int_0^{k_{max}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega_{cen}}{c}\right)^2-k^2}(z-z_t)\right)}kdk\right)E_{\text{inc,x}}(x_i,y_j,z=z_t,t)\right]\hat{x} \tag{4.41}$$
$$+\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\left[\left(\frac{\pi^2}{{k_{max}}^2}\int_0^{k_{max}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega_{cen}}{c}\right)^2-k^2}(z-z_t)\right)}kdk\right)E_{\text{inc,y}}(x_i,y_j,z=z_t,t)\right]\hat{y}$$
$$+\frac{i}{(2\pi)}\sum_{i,j=-\infty}^{\infty}\left[\left(\frac{\omega_{cen}}{c}\right)^2\left(\int_0^{\frac{\pi}{2}} -J_1\left(\frac{\omega_{cen}rsin\phi}{c}\right)e^{i\frac{\omega_{cen}}{c}cos\phi(z-z_t)}sin^2\phi d\phi\right)\right.$$
$$\left.(cos\alpha E_{inc,x}(x_i,y_i,z=z_t,t)+sin\alpha E_{inc,y}(x_i,y_i,z=z_t,t))\right]\hat{z}$$

where $r=\sqrt{(x-x_i)^2+(y-y_j)^2}$, $\alpha = tan^{-1}\left(\frac{y-y_j}{x-x_i}\right)$, $\phi = arcsin\left(\frac{kc}{\omega_{cen}}\right)$.

We note that the k-integrals in Eq. 4.40 have an evanescent component when $k > \frac{\omega_{cen}}{c}$. To make this explicit and give the final expression we compute, we split

the k-integrals into two parts:

$$
\begin{aligned}
&\mathbf{E}_{\text{inc}}(x,y,z,t) = \\
&\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\Bigg[\frac{\pi^2}{{k_{max}}^2}\Bigg(\int_0^{\frac{\omega_{cen}}{c}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega_{cen}}{c}\right)^2-k^2}(z-z_t)\right)}dk+ \\
&\int_{\frac{\omega_{cen}}{c}}^{k_{max}} J_0(kr)e^{-\left(\sqrt{k^2-\left(\frac{\omega_{cen}}{c}\right)^2}(z-z_t)\right)}kdk\Bigg)E_{\text{inc,x}}(x_i,y_j,z=z_t,t)\Bigg]\hat{x} \\
&+\frac{1}{2\pi}\sum_{i,j=-\infty}^{\infty}\Bigg[\frac{\pi^2}{{k_{max}}^2}\Bigg(\int_0^{\frac{\omega_{cen}}{c}} J_0(kr)e^{i\left(\sqrt{\left(\frac{\omega_{cen}}{c}\right)^2-k^2}(z-z_t)\right)}dk+ \\
&\int_{\frac{\omega_{cen}}{c}}^{k_{max}} J_0(kr)e^{-\left(\sqrt{k^2-\left(\frac{\omega_{cen}}{c}\right)^2}(z-z_t)\right)}kdk\Bigg)E_{\text{inc,y}}(x_i,y_j,z=z_t,t)\Bigg]\hat{y} \\
&+\frac{i}{(2\pi)}\sum_{i,j=-\infty}^{\infty}\Bigg[\left(\frac{\omega_{cen}}{c}\right)^2(cos\alpha E_{inc,x}(x_i,y_i,z=z_t,t)+sin\alpha E_{inc,y}(x_i,y_i,z=z_t,t)) \\
&\Bigg(\int_0^{\frac{\pi}{2}} -J_1\left(\frac{\omega_{cen}rsin\phi}{c}\right)e^{i\frac{\omega_{cen}}{c}cos\phi(z-z_t)}sin^2\phi d\phi+ \\
&\int_0^{arcsec\left(\frac{\omega_{max}}{\omega}\right)} iJ_1\left(\frac{\omega_{cen}rsec\psi}{c}\right)e^{-\frac{\omega_{cen}}{c}tan\psi(z-z_t)}sec^3\psi d\psi\Bigg)\Bigg]\hat{z}
\end{aligned}
\tag{4.42}
$$

where $r=\sqrt{(x-x_i)^2+(y-y_j)^2}$, $\alpha=tan^{-1}\left(\frac{y-y_j}{x-x_i}\right)$, $\phi=arcsin\left(\frac{kc}{\omega}\right)$, $\psi=arcsec\left(\frac{kc}{\omega}\right)$.

Determining full vector expression for H-Field

Up until this point, we have written all expressions for the electric-field – however, we must also provide tangential components for the magnetic-field to create the TFSF planar currents. All the previous derivation is actually not specific to electric-field, so one approach we can take is to also Nyquist-sample the desired magnetic-field of the source in our given transverse plane and use Eq. 4.42 to compute the necessary tangential components for the TFSF source.

An alternative approach is to computing the required H-field components is to use the expression for $\mathbf{E}_{\text{inc}}(x,y,z,t)$ in Eq. 4.40 to compute $\mathbf{H}_{\text{inc}}(x,y,z,t)$ starting with

Maxwell's equations:

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \text{ , assuming time dependence } e^{-i\omega t} \text{: } FFT[\mathbf{H}]_j = -\frac{i}{\omega_j \mu} FFT\left[\nabla \times \mathbf{E}\right]_j \tag{4.43}$$

$$\mathbf{H}_{\text{inc}}(x,y,z,t) = IFFT\left[-\frac{i}{\boldsymbol{\omega}\mu} \odot FFT\left[\nabla \times \mathbf{E}\right]\right] \tag{4.44}$$

where $\odot$ denotes element-wise product between the elements of the vector $-\frac{i}{\boldsymbol{\omega}\mu}$ and $FFT\left[\nabla \times \mathbf{E}\right]$.

Another approach to numerically calculating $\mathbf{H}_{\text{inc}}(x,y,z,t)$ from $\mathbf{E}_{\text{inc}}(x,y,z,t)$ is to do a cumulative numerical integral of $\mathbf{E}_{\text{inc}}(x,y,z,t)$ over time:

$$\mathbf{H}_{\text{inc}}(x,y,z,t) = -\frac{1}{\mu}\int_0^t (\nabla \times \mathbf{E}_{\text{inc}}(x,y,z,t)dt) \tag{4.45}$$

This numerical integration approach in Eq. 4.45 should be faster in O(n) rather than the O(nlogn) for the FFT approach in Eq. 4.43.

Finally, the fastest approach to computing $\mathbf{H}_{\text{inc}}(x,y,z,t)$ from $\mathbf{E}_{\text{inc}}(x,y,z,t)$ is to approximate only at w_{cen} - however, this is an approximation that will only work when the frequency bandwidth is small:

$$\mathbf{H}_{\text{inc}}(x,y,z,t) = -\frac{i}{\omega_{cen}\mu}\nabla \times \mathbf{E}_{\text{inc}}(x,y,z,t) \tag{4.46}$$

4.3.4 Jinc TFSF: Single-source fields

We implement the TFSF source corresponding to the jinc field using Eq. 4.42 to obtain the required E and H field tangential components. Fig. 4.4 shows a TFSF source box generating the field of a single-jinc source within the box and no field everywhere else.

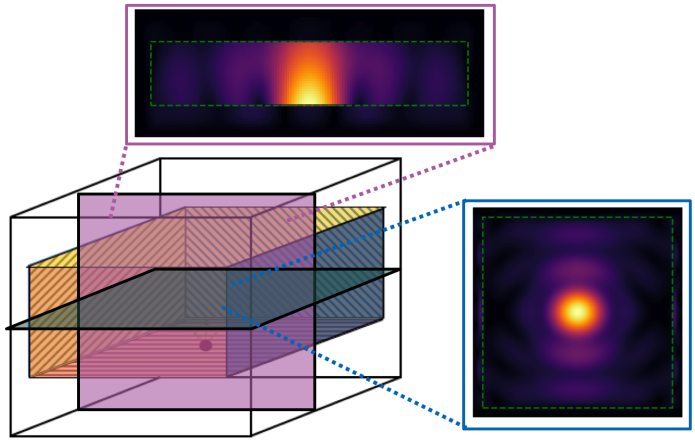


Figure 4.4: **Spatial-locality study with TFSF jinc source.** Schematic of the FDTD simulation setup for a TFSF source that generates the field for a single jinc source. The TFSF source box is completely contained within the FDTD simulation region – PML boundary conditions surround the FDTD simulation region. The field from a horizontal slice (blue plane) and vertical slice (magenta plane) through the simulation center is shown, and the familiar jinc spatially-localized field pattern is observed inside the TFSF box, and no field is observed outside the box.

Using this single-jinc TFSF source, we can then perform a spatial-locality padding study similar to that of Fig. 2.2, Fig. 3.13, and Fig. 3.10. As shown in Fig. 4.5a-b, we choose a scatterer library consisting of high aspect-ratio silicon rectangular posts placed close together in a background of air – we would not have been able to simulate surfaces composed of these scatterers using the T-matrix method on spherical harmonic functions from Chapter 3 because of the intersection of their bounding spheres. We choose to work with this scatterer library to highlight the flexibility we gain from working with FDTD as the single-node simulator. Once again, we see a convergence in the relative error between the simulation of a full metasurfaces and the simulation of a truncated metasurface when excited by the field of a single jinc

source (Fig. 4.5c).

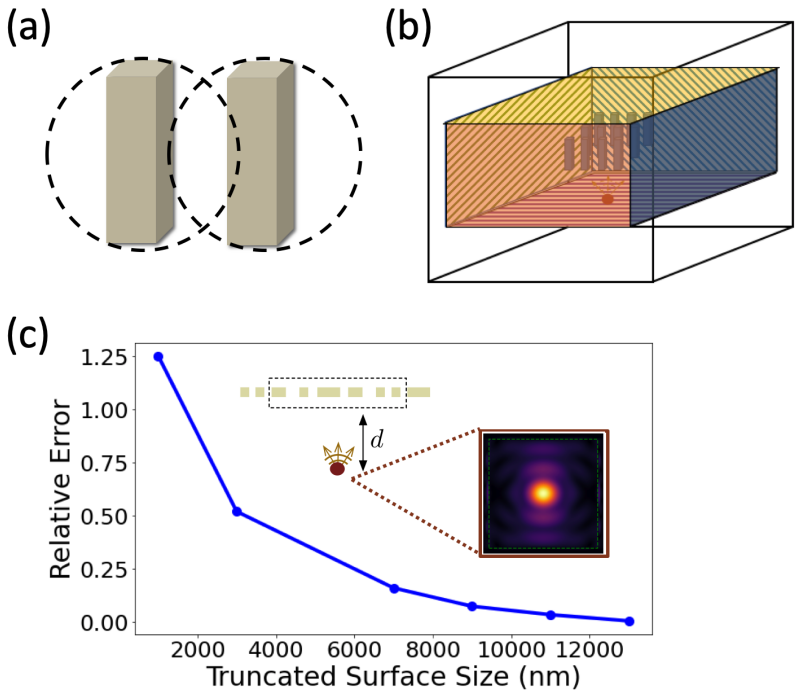


Figure 4.5: **FDTD simulation fields for a TFSF source corresponding to a single jinc source.** (**a**) Schematic of the rectangular pillar scatterers (3.68 refractive index, height 1050 nm, lattice constant 700 nm, side length 50-200 nm) used in this spatial-locality study. The dashed lines show the bounding spheres of the scatterers to highlight that this scatterer library violates the bounding sphere separation constraint from the spherical harmonic T-matrix method in Section 3.9. (**b**) Schematic of the spatial-locality study setup – the metasurface is enclosed in a TFSF source box that generates the field of a single jinc source as in Fig. 4.4. (**c**) Percent error in scattered field power versus spatial-extent of metasurface included in the simulation for a single jinc source placed 0.1 μm from the metasurface. The full metasurface is a 15 μm $\times$ 15 μm metalens with focal length of 20 μm composed of scatterers described in (a), excited with wavelength 1550 nm in background index of 1. The surface size on the x-axis of this convergence plot refers to the spatial-extent around the center of this metasurface that is included in the simulation. The y-axis relative error is computed assuming the simulation including the full metasurface is the converged result.

4.4 Distributed metalens simulation

Now that we are able to generate jinc fields for an FDTD simulation using the TFSF formulation and we can perform spatial-locality studies, we are ready to perform a distributed simulation using FDTD as the single-node solver. We use the same high aspect-ratio scatterer library from Fig. 4.5 and select a padding of $4\mu m$ based on this study. Using a subregion size of $8\mu m \times 8\mu m$, we perform a distributed simulation of a $30\mu m \times 30\mu m$ metalens with focal length $20\mu m$ – this results in 16 subregions, which we distributed across 64 CPU's to perform a parallelized meep FDTD simulation across 4 CPU's (Fig. 4.6a). We compute a final focusing efficiency for this metalens of 72% (Fig. 4.6b), showing good agreement with the 75% focusing efficiency we compute with a lumerical FDTD simulation of the same metalens. This successful distributed simulation using the FDTD jinc TFSF source opens the door to performing full simulation and optimization of a wide range of large-area metasurfaces.

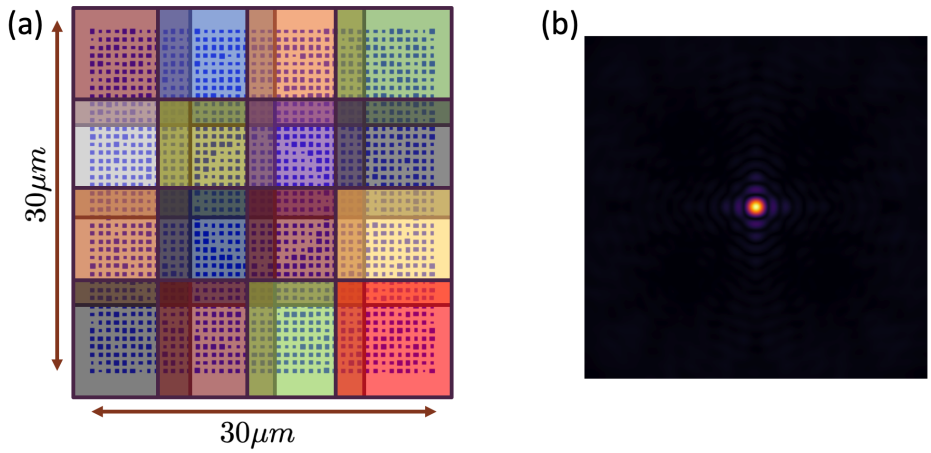


Figure 4.6: **Distributed metalens simulation.** (**a**) Schematic of the metalens simulated. The metalens is $30\mu m \times 30\mu m$ with focal length $20\mu m$ (NA = 0.6), and is composed of scatterers from the library described in Fig. 4.5a. The subregions for the distributed simulation are of size $8\mu m \times 8\mu m$ with padding of $4\mu m$, resulting in 16 subregions. (**b**) X-component of the electric field at the focal plane of the metalens – the efficiency is calculated to be 72%.

Chapter 5

Conclusion and Outlook

We have demonstrated a scalable distribution method to accurately simulate arbitrarily large-area metasurfaces. Our method uses the Nyquist sampling theorem to allow parallel distribution of compute across multiple compute nodes, on which a T-matrix method with spherical harmonic functions solver or an FDTD solver is used to simulate the subregion. With our distribution approach applied with the T-matrix method subregion solver, we show a roughly $\frac{1}{N_{GPU}}$ scaling of the total simulation time and demonstrate that our method accurately accounts for all scatterer interactions. We demonstrate our ability to apply our distribution method to the computation of the gradient with respect to all design parameters, and use this gradient information to optimize a performance metric such as the focusing efficiency. With our distribution method applied with an FDTD subregion solver, we demonstrate the ability handle a far wider range of metasurface geometries (for example, high aspect-ratio scatterers with small lattice constants and substrates). Additionally, the recent developments of ultra-efficient hardware-specialized FDTD solvers such as [40] make the use of FDTD as the subregion solver in our distribution method extremely promising for scaling to very large-areas. Our simulation distribution method provides a solution to the long-standing problem of simulating large-area metasurfaces and opens the door to

gradient-based optimization of the full metasurface, taking advantage of all the design degrees of freedom. This is especially important as we consider very high-NA metasurfaces and moving to thick, multi-layer metasurfaces.

High-NA metasurfaces are very sought-after for augmented-reality displays, microscopy, and mobile imaging. However, as discussed in [16], the commonly-used unit-cell metasurface design approach from Section 1.2 does not provide the number of degrees of freedom necessary for high-NA lenses. This can be empirically observed in the limited efficiencies achieved to-date in experimental demonstrations of high-NA metalenses [53, 5, 67, 13]. Instead, we must push the "unit-cell" to become the whole metasurface to open the door to efficient and high-NA lenses – thus, the ability to simulate and optimize the full metasurface opens the possibility of discovering high-NA metalens designs with efficiencies high enough for practical use. Our method for full metasurface optimization is likely to also benefit the discovery of metasurfaces with highly efficient operation at several discrete modes - for example, metasurfaces that focus a few discrete wavelengths to different points [70].

For applications such as multi-color imaging and aberration correction, metasurface designs are required to operate efficiently at a high number of modes. However, as discussed in [62, 74], time bandwidth product bounds imply that there is a trade-off between bandwidth of a design and the minimum required thickness. As a result, moving to multi-layer metasurfaces [59] or fully-3D thick metasurfaces [12] is likely necessary for achieving high efficiency across many modes or for broadband operation. However, moving to thick metasurfaces introduces many more degrees of freedom to an already very large design problem. The efficient parameter space exploration provided by gradient-based optimization with the adjoint method will be critical for design of practical thick metasurfaces [55].

Metasurfaces offer a compact and cost-effective platform for implementing the complex optical functionalities demanded by next-generation technologies such as

augmented reality, lidar, and mobile imaging. By applying our distributed adjoint optimization approach using a fast FDTD subregion solver such as [40], novel free-form multilayer or thick metasurface structures could be designed to provide functionality and performance beyond the current unit-cell-based state of the art. We expect that optimizing large-area and large-volume metasurfaces as a whole using our distribution method will produce devices that help realize technologies such as augmented-reality glasses that actually do fit in the footprint of standard glasses, and complex structured-light systems for mobile phone facial-recognition or robotics. The full-volume metasurface design space is unexplored territory with much promise for the next generation of compact optical devices.

Appendix A

Transition-matrix simulation method

A.1 Vector spherical wavefunctions

A.1.1 Definition

Here, we define the vector spherical harmonic functions that we use to provide a compact description of the incident and scattered fields for the T-matrix method. We follow appendix B of [26] and appendix D of [86], which may be referenced for further details.

We begin with the frequency-domain scalar helmholtz equation for homogenous media in Eq. A.1, where $k = \omega\sqrt{\varepsilon}/c$ is the wavenumber of the medium with permittivity ε.

$$\nabla^2\phi(\mathbf{x}) + k^2\phi(\mathbf{x}) = 0 \tag{A.1}$$

Applying separation of variables, we find that the spherical wavefunctions $\phi_{l,m}(k_b\mathbf{x})$

and $\mathcal{R}\phi_{l,m}(k_b\mathbf{x})$ defined below are solutions to Eq. A.1:

$$\phi_{l,m}(k\mathbf{x}) = h_l^{(1)}(kr)P_l^{|m|}(\cos\theta)\exp(im\varphi) \tag{A.2a}$$

$$\mathcal{R}\phi_{l,m}(k\mathbf{x}) = j_l(kr)P_l^{|m|}(\cos\theta)\exp(im\varphi) \tag{A.2b}$$

where $l \in \{0, 1, 2\dots\}$, $m \in \{-l, -l+1, \dots l-1, l\}$, (r, θ, φ) are the spherical coordinates of the point $\mathbf{x}$, $h_l^{(1)}(x)$ is the spherical hankel function of the first kind, $j_l(x)$ is the spherical bessel function and $P_l^{|m|}(x)$ is the normalized associated legendre polynomial. Note that each solution of the scalar wave function is indexed by two numbers — the orbital index l and the magnetic index m. $\mathcal{R}$ indicates the wavefunction's behavior at the origin — $\mathcal{R}\phi_{l,m}(k\mathbf{x})$ evaluates to 0 at the origin while $\phi_{l,m}(\mathbf{x})$ diverges at the origin. Additionally, $\phi_{l,m}(k\mathbf{x})$ captures spherical waves radiating to infinity (used to express the scattered field in the T-matrix formalism), while $\mathcal{R}\phi_{l,m}(k\mathbf{x})$ captures spherical standing waves (used to express the incident field in the T-matrix formalism).

Next, we can use each of the scalar spherical harmonic functions from Eq. A.2 to compute two vector spherical harmonic functions that correspond to two independent polarizations that satisfy the frequency-domain Maxwell's equations in homogenous media A.3.

$$\begin{aligned} \nabla \times \nabla \times \mathbf{E}(\mathbf{x}) - k^2\mathbf{E}(\mathbf{x}) &= 0 \\ \nabla \cdot \mathbf{E}(\mathbf{x}) &= 0 \end{aligned} \tag{A.3}$$

These vector spherical harmonic functions, denoted by $\Phi_{l,m,p}(k_b\mathbf{x})$ and $\mathcal{R}\Phi_{l,m,p}(k_b\mathbf{x})$

(where $p \in \{0, 1\}$ is the polarization index), are expressed as:

$$\begin{bmatrix} \Phi_{l,m,p=0}(k\mathbf{x}) \\ \mathcal{R}\Phi_{l,m,p=0}(k\mathbf{x}) \end{bmatrix} = \frac{1}{\sqrt{2l(l+1)}} \nabla \begin{bmatrix} \phi_{l,m}(k\mathbf{x}) \\ \mathcal{R}\phi_{l,m}(k\mathbf{x}) \end{bmatrix} \times \mathbf{x} \tag{A.4a}$$

$$\begin{bmatrix} \Phi_{l,m,p=1}(k\mathbf{x}) \\ \mathcal{R}\Phi_{l,m,p=1}(k\mathbf{x}) \end{bmatrix} = \frac{1}{k} \nabla \times \begin{bmatrix} \Phi_{l,m,p=0}(k\mathbf{x}) \\ \mathcal{R}\Phi_{l,m,p=0}(k\mathbf{x}) \end{bmatrix} \tag{A.4b}$$

where $l \in \{1, 2, 3 \dots\}$, $m \in \{-l, -l+1 \dots l-1, l\}$ and $p \in \{0, 1\}$. The vector spherical harmonic functions form a basis for the solutions of the vector helmholtz equation (Eq. A.3), and an orthogonality condition can be constructed for them on the surface of an arbitrarily chosen sphere as detailed in [26].

We can express the incident field for the T-matrix method on the regular vector spherical wavefunctions:

$$\mathbf{E}_{\text{inc}}(\mathbf{x}) = \sum_{l,m,p} a_{l,m,p} \mathcal{R}\Phi_{l,m,p}(k_b \mathbf{x}) \tag{A.5}$$

We can express the scattered field for the T-matrix method on the radiating vector spherical wavefunctions:

$$\mathbf{E}_{\text{sca}}(\mathbf{x}) = \sum_{l,m,p} s_{l,m,p} \Phi_{l,m,p}(k_b \mathbf{x}) \tag{A.6}$$

A.1.2 Translation coefficients

The spherical harmonic translation theorem relates vector spherical wavefunctions defined with respect to two different origins to each other [26, 29]. This is necessary for including the field scattered from all other scatterers in the incident field for a

given scatterer (Section 3.2):

$$\mathbf{E}_{\mathrm{inc},j}(\mathbf{x}) = \mathbf{E}_{\mathrm{inc}}(\mathbf{x}) + \sum_{\substack{i=1 \\ i\neq j}}^{N} \sum_{l,m,p} s_{l,m,p;i} \Phi_{l,m,p}(k_b(\mathbf{x} - \mathbf{x}_i)) \tag{A.7}$$

The translation theorem is expressed as:

$$\Phi_{l,m,p}(k(\mathbf{x} - \mathbf{x}_a)) = \sum_{l',m',p'} \xi_{l,m,p;l',m',p'}(k(\mathbf{x}_a - \mathbf{x}_b)) \Phi_{l',m',p'}(k(\mathbf{x} - \mathbf{x}_b)) \tag{A.8}$$

where

$$\xi_{l,m,p;l',m',p'}(\mathbf{x}) = \delta_{p,p'} \alpha_{l,m;l',m'}(\mathbf{x}) + (1 - \delta_{p,p'}) \beta_{l,m;l',m'}(\mathbf{x}) \tag{A.9}$$

and with (r, φ, θ) as the spherical coordinates of the vector $\mathbf{x}$:

$$\alpha_{l,m;l',m'}(\mathbf{x}) = \exp(i(m - m')\varphi) \sum_{q=|l-l'|}^{l+l'} a_5(l, m|l', m'|q) h_q^{(1)}(r) P_q^{|m-m'|}(\cos\theta) \tag{A.10a}$$

$$\beta_{l,m;l',m'}(\mathbf{x}) = \exp(i(m - m')\varphi) \sum_{q=|l-l'|+1}^{l+l'} b_5(l, m|l', m'|q) h_q^{(1)}(r) P_q^{|m-m'|}(\cos\theta) \tag{A.10b}$$

where:

$$\begin{aligned}\alpha(l,m|l',m'|p) =& i^{|m-m'|-|m|-|m'|+l'-l+p}(-1)^{m-m'} \\ &\times [l(l+1)+l'(l'+1)-p(p+1)]\sqrt{2p+1} \\ &\times \sqrt{\frac{(2l+1)(2l'+1)}{2l(l+1)(l'+1)}} \begin{pmatrix} l & l' & p \\ m & -m' & m'-m \end{pmatrix} \begin{pmatrix} l & l' & p \\ 0 & 0 & 0 \end{pmatrix}\end{aligned} \tag{A.11a}$$

$$\begin{aligned}\beta(l,m|l',m'|p) =& i^{|m-m'|-|m|-|m'|+l'-l+p}(-1)^{m-m'} \\ &\times \sqrt{(l+l'+1+p)(l+l'+1-p)(p+l-l')(p-l+l')(2p+1)} \\ &\times \sqrt{\frac{(2l+1)(2l'+1)}{2l(l+1)(l'+1)}} \begin{pmatrix} l & l' & p \\ m & -m' & m'-m \end{pmatrix} \begin{pmatrix} l & l' & p \\ 0 & 0 & 0 \end{pmatrix}\end{aligned} \tag{A.11b}$$

with

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \tag{A.12}$$

is the Wigner-3j symbol [58]. With this addition theorem, $\mathbf{E}_{\mathrm{inc},j}(\mathbf{x})$ can be rewritten as:

$$\mathbf{E}_{\mathrm{inc},j}(\mathbf{x}) = \sum_{l,m,p} a_{l,m,p;j} \mathcal{R}\Phi_{l,m,p}(k_b(\mathbf{x}-\mathbf{x}_j)) \tag{A.13}$$

where

$$a_{l,m,p;j} = a^{(0)}_{l,m,p;j} + \sum_{\substack{i=1 \\ i\neq j}}^{N} \sum_{l',m',p'} \xi_{l',m',p';l,m,p}(k_b(\mathbf{x}_j-\mathbf{x}_i)) s_{l',m',p';i} \tag{A.14}$$

A.2 Implementing jinc source on vector spherical harmonic basis

In order to implement the Nyquist-sampling-based-parellelization scheme described above, it is necessary to be able to simulate the response of the metasurface to a collection of jinc sources using the T-matrix method. This requires the ability to expand the jinc sources on the vector spherical wavefunctions (Eq. 3.8). In this section, we mathematically develop such an expansion.

Consider a jinc source at $z = z_0$ propagating in the $+z$ direction and centered at $\mathbf{x}_0^t = (x_0, y_0)$ in the transverse plane. The transverse electric field at $z = z_0$ is given by:

$$\mathbf{E}^T(\mathbf{x}^T, z = z_0) = \mathbf{A}^T \frac{j_1(k_0|\mathbf{x}^T - \mathbf{x}_0^T|)}{k_0|\mathbf{x}^T - \mathbf{x}_0^T|} \tag{A.15}$$

where $j_1(\cdot)$ is the first order spherical bessel function and $\mathbf{A}^T = A_x\hat{x} + A_y\hat{y}$ is the transverse polarization of the jinc field. Alternatively, in the fourier representation,

$$\mathbf{E}^T(x, y, z = z_0) = \frac{\mathbf{A}^T}{2\pi} \int_{|\mathbf{k}^T|<k_0} \exp[i\mathbf{k}^T \cdot (\mathbf{x}^T - \mathbf{x}_0^T)] d^2\mathbf{k}^T \tag{A.16}$$

Propagating each plane-wave component, we obtain:

$$\mathbf{E}(x, y, z) = \frac{1}{2\pi} \int_{|\mathbf{k}^T|<k_0} \mathbf{A}(\mathbf{k}^T) \exp(i\mathbf{k}^T \cdot (\mathbf{x}^T - \mathbf{x}_0^T)) \exp(ik_z(z - z_0)) d^2\mathbf{k}^T \tag{A.17}$$

where $k_z = \sqrt{k_0^2 - \mathbf{k}^T \cdot \mathbf{k}^T}$ and

$$\mathbf{A}(\mathbf{k}^T) = \mathbf{A}^T - \hat{z}\frac{\mathbf{k}^T \cdot \mathbf{A}^T}{k_z(\mathbf{k}^T)} \tag{A.18}$$

Changing the integration variable to α, β ($0 \leq \alpha \leq 2\pi$ and $0 \leq \beta \leq \pi/2$) where

$\mathbf{k}^T(\beta,\alpha) = k_0 \sin\beta(\hat{x}\cos\alpha + \hat{y}\sin\alpha)$ and therefore $k_z(\mathbf{k}^T) \equiv k_z(\beta,\alpha) = k_0\cos\beta$, $\mathbf{A}(\mathbf{k}^T) \equiv \mathbf{A}(\beta,\alpha) = \hat{x}A_x + \hat{y}A_y - \hat{z}\tan\beta(A_x\cos\alpha + A_y\sin\alpha)$ and $d^2\mathbf{k}^T = k_0^2\sin\beta\cos\beta d\alpha d\beta$. Moreover, each plane wave can be expanded into a sum of vector spherical harmonic wavefunctions [26]:

$$\begin{aligned}&\mathbf{A}(\beta,\alpha)\exp(i\mathbf{k}^T(\beta,\alpha)\cdot(\mathbf{x}^T - \mathbf{x}_0^T))\exp(ik_z(\beta,\alpha)(z-z_0))\\ &= \exp[i\mathbf{k}(\beta,\alpha)\cdot(\mathbf{x}_0' - \mathbf{x}_0)]\sum_{l,m,p} a_{l,m,p}(\beta,\alpha)\mathcal{R}\Phi_{l,m,p}(k_0(\mathbf{x}-\mathbf{x}_0'))\end{aligned} \tag{A.19}$$

where

$$a_{l,m,p=0}(\beta,\alpha) = -\frac{4i^l\exp(-im\alpha)}{\sqrt{2l(l+1)}}\left[im\pi_l^{|m|}(\beta)(\hat{\beta}\cdot\mathbf{A}(\beta,\alpha)) + \tau_l^{|m|}(\beta)(\hat{\alpha}\cdot\mathbf{A}(\beta,\alpha))\right] \tag{A.20a}$$

$$a_{l,m,p=1}(\beta,\alpha) = -\frac{4i^{l+1}\exp(-im\alpha)}{\sqrt{2l(l+1)}}\left[\tau_l^{|m|}(\beta)(\hat{\beta}\cdot\mathbf{A}(\beta,\alpha)) - im\pi_l^{|m|}(\beta)(\hat{\alpha}\cdot\mathbf{A}(\beta,\alpha))\right] \tag{A.20b}$$

where $\mathbf{m}_{l,m}(\cdot,\cdot)$ and $\mathbf{n}_{l,m}(\cdot,\cdot)$ are vector spherical harmonics with orbital index l and azimuthal index m. Therefore,

$$\mathbf{E}(\mathbf{x}) = \sum_{l,m,p} a_{l,m,p}\mathcal{R}\Phi_{l,m,p}(k_0(\mathbf{x}-\mathbf{x}_0')) \tag{A.21}$$

where $a_{l,m,p}$ are given by:

$$a_{l,m,p} = \frac{1}{2\pi}\int_{\beta=0}^{\pi/2}\int_{\alpha=0}^{2\pi} a_{l,m,p}(\beta,\alpha)\exp[i\mathbf{k}(\beta,\alpha)\cdot(\mathbf{x}_0' - \mathbf{x}_0)]\sin\beta\cos\beta d\alpha d\beta \tag{A.22}$$

In the remainder of this section, we will evaluate the integral over α analytically, and the resulting expression can then be numerically integrated over β. To do so, it will

be convenient to define a function $\Gamma_m(\xi, \eta, \rho)$ by:

$$\Gamma_m(\xi, \eta, \rho) = \frac{1}{2\pi} \int_0^{2\pi} \exp(-im\alpha) \exp[i\rho \cos(\alpha - \xi)] \cos(\alpha - \eta) d\alpha \tag{A.23}$$

$\Gamma_m(\xi, \eta, \rho)$ can be evaluated analytically: making a change of variables to $\alpha' = \alpha - \xi + \pi/2$, we obtain:

$$\begin{aligned}
\Gamma_m(\xi, \eta, \rho) &= \frac{i^m \exp(-im\xi)}{2\pi} \int_{\pi/2-\xi}^{5\pi/2-\xi} \exp(-im\alpha') \exp(i\rho \sin \alpha') \sin(\alpha' + \xi - \eta) d\alpha' \\
&= \frac{i^{m-1} \exp(-im\xi)}{4\pi} \int_0^{2\pi} \exp(-im\alpha') \exp(i\rho \sin \alpha') \\
&\quad (\exp(i(\alpha' + \xi - \eta)) - \exp(-i(\alpha' + \xi - \eta))) d\alpha' \\
&= \frac{i^{m-1} \exp(-im\xi)}{2} \Big[\exp\{i(\xi - \eta)\} J_{m-1}(\rho) - \exp\{-i(\xi - \eta)\} J_{m+1}(\rho) \Big]
\end{aligned} \tag{A.24}$$

wherein we have used the identity:

$$J_m(\rho) = \frac{1}{2\pi} \int_0^{2\pi} \exp[i\rho \sin \theta - m\theta] d\theta \tag{A.25}$$

Additionally, note that $\hat{\beta} = (\hat{x} \cos \alpha + \hat{y} \sin \alpha) \cos \beta - \hat{z} \sin \beta$ and $\hat{\alpha} = -\hat{x} \sin \alpha + \hat{y} \cos \alpha$. Therefore:

$$\hat{\beta} \cdot \mathbf{A}(\beta, \alpha) = \sec \beta (A_x \cos \alpha + A_y \sin \alpha) \tag{A.26a}$$

$$\hat{\alpha} \cdot \mathbf{A}(\beta, \alpha) = -A_x \sin \alpha + A_y \cos \alpha \tag{A.26b}$$

Finally, let $\mathbf{x}_0' - \mathbf{x}_0 \equiv (r_0, \theta_0, \varphi_0)$, and therefore $\mathbf{k}(\beta, \alpha) \cdot (\mathbf{x}_0' - \mathbf{x}_0) = k_0 r_0 (\cos \beta \cos \theta_0 + \sin \beta \sin \theta_0 \cos(\alpha - \varphi_0))$.

1. Consider the computation of $a_{l,m,p=0}$. Using Eq. A.20a, we obtain:

$$a_{l,m,p=0}(\beta,\alpha)\sin\beta\cos\beta = -\frac{4i^l\exp(-im\alpha)}{\sqrt{2l(l+1)}}\big[(im\pi_l^{|m|}(\beta)A_x+\tau_l^{|m|}(\beta)\cos\beta A_y)\cos\alpha + (im\pi_l^{|m|}(\beta)A_y-\tau_l^{|m|}(\beta)\cos\beta A_x)\sin\alpha\big]\sin\beta \tag{A.27}$$

and therefore

$$\frac{1}{2\pi}\int_0^{2\pi} a_{l,m,p=0}(\beta,\alpha)\exp[i\mathbf{k}(\beta,\alpha)\cdot(\mathbf{x}_0-\mathbf{x}_0')]\sin\beta\cos\beta d\alpha = -\frac{4i^l\exp(ik_0r_0\cos\beta\cos\theta_0)\sin\beta}{\sqrt{2l(l+1)}} \times\Big[(im\pi_l^{|m|}(\beta)A_x+\tau_l^{|m|}(\beta)\cos\beta A_y)\Gamma_m(\phi_0,0,k_0r_0\sin\beta\sin\theta_0)+ (im\pi_l^{|m|}(\beta)A_y-\tau_l^{|m|}(\beta)\cos\beta A_x)\Gamma_m(\phi_0,\pi/2,k_0r_0\sin\beta\sin\theta_0)\Big] \tag{A.28}$$

2. Consider the computation of $a_{l,m,p=1}$. Using Eq. A.20b, we obtain:

$$a_{l,m,p=1}(\beta,\alpha)\sin\beta\cos\beta = -\frac{4i^{l+1}\exp(-im\alpha)}{\sqrt{2l(l+1)}}\big[(\tau_l^{|m|}(\beta)A_x-im\pi_l^{|m|}(\beta)A_y\cos\beta)\cos\alpha +(\tau_l^{|m|}(\beta)A_y+im\pi_l^{|m|}(\beta)A_x\cos\beta)\sin\alpha\big] \tag{A.29}$$

and therefore

$$\begin{aligned}&\frac{1}{2\pi}\int_0^{2\pi} a_{l,m,p=1}(\beta,\alpha)\exp[i\mathbf{k}(\beta,\alpha)\cdot(\mathbf{x}_0-\mathbf{x}_0')]\sin\beta\cos\beta d\alpha = \\ &-\frac{4i^{l+1}\exp(ik_0r_0\cos\beta\cos\theta_0)\sin\beta}{\sqrt{2l(l+1)}} \\ &\times\big[(\tau_l^{|m|}(\beta)A_x-im\pi_l^{|m|}(\beta)A_y\cos\beta)\Gamma_m(\phi_0,0,k_0r_0\sin\beta\sin\theta_0) \\ &+(\tau_l^{|m|}(\beta)A_y+im\pi_l^{|m|}(\beta)A_x\cos\beta)\Gamma_m(\phi_0,\pi/2,k_0r_0\sin\beta\sin\theta_0)\big]\end{aligned} \tag{A.30}$$

It can be noted that the numerical integration over β can be accelerated by using lookup tables for the various special functions involved in the computation. Furthermore, we also parallelize the computation of the jinc sources on GPU to accelerate it with one thread being assigned to compute $a_{l,m,p}$ for a single choice of (l, m, p) with respect to a chosen scatterer.

Finally, we remark that since the jinc sources are spatially limited, we only need to simulate the metasurface locally to compute its response to the jinc source. In order to estimate how local this simulation needs to be, a padding study like the one in Figure 2.2 should be performed.

A.3 Spherical T-matrix validation and benchmark

In this Appendix, we validate and benchmark our implementation of the T-matrix method against the common FDTD software Lumerical [84].

Validation simulations for our T-matrix simulation method implementation are shown in Supplementary Figure A.1 — we see good agreement between the T-matrix method, FDTD, and FDFD simulations.

Fig. A.2 shows a comparison of the simulation time of our implementation when compared to commercially available FDTD solvers [84]. We note that this is not a strictly rigorous benchmark, as the parallelization settings are not exactly the same between our software and Lumerical – the purpose of this comparison is only to roughly put our simulation times in context of a simulation tool the reader is likely familiar with. Our software is able to simulate a 60×60 μm^2 large metasurface in 1-1.5 hours on a single GPU core (Nvidia V100). As seen in the inset in Fig. A.2, the bulk of the simulation time is spent in the linear system solve.

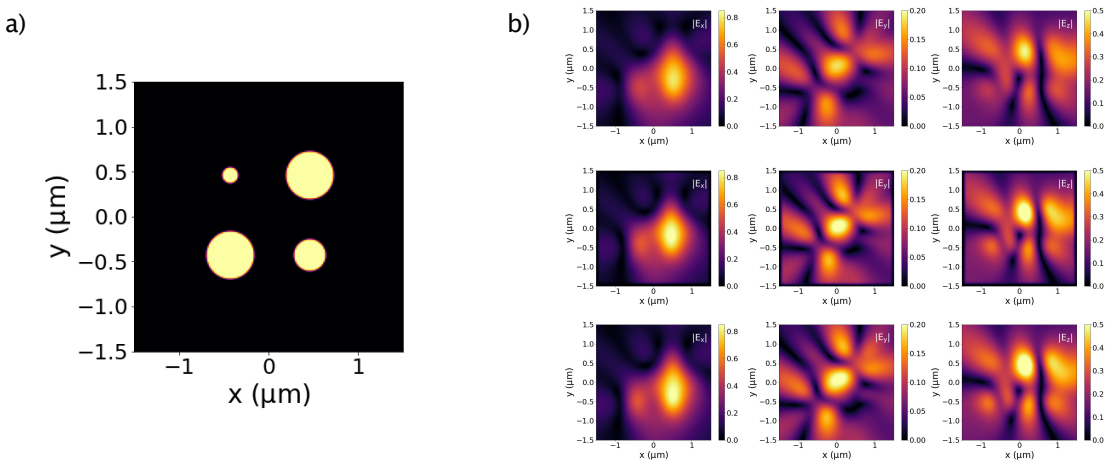


Figure A.1: **Validation of our T-matrix implementation.** (**a**) Scatterers used for the validation simulation — the scatterers are illuminated with a plane wave and have a refractive index of 3.5. (**b**) Electric field magnitudes for the x, y, and z components of the scattered fields from the T-matrix method simulation (top), FDFD simulation (middle), and FDTD simulation (bottom).

A.4 Vector spheroidal wavefunctions

While the vector spherical wavefunctions can be used for an efficient description of the scattered field for subwavelength scatterers, there are several drawbacks of this choice of basis functions. Firstly, the computation of the transition matrices for high aspect ratio scatterers is ill conditioned when the vector spherical wavefunctions are employed. Furthermore, the solution to the multiple-scattering problem in spherical coordinates is no longer valid if two scatterers are close enough for the bounding sphere of one scatterer to contain the origin of the second scatterer. Using vector spheroidal wavefunctions for the representing the scattering properties of the individual scatterers resolved both of these problems. The derivations in Sections A.4.1 and A.4.2, and the implementation in Section A.4.3 were done by Rahul Trivedi.

Unlike their spherical counterparts, vector spheroidal harmonics have not been significantly used for the solution of Maxwell's equations. Previous works have either

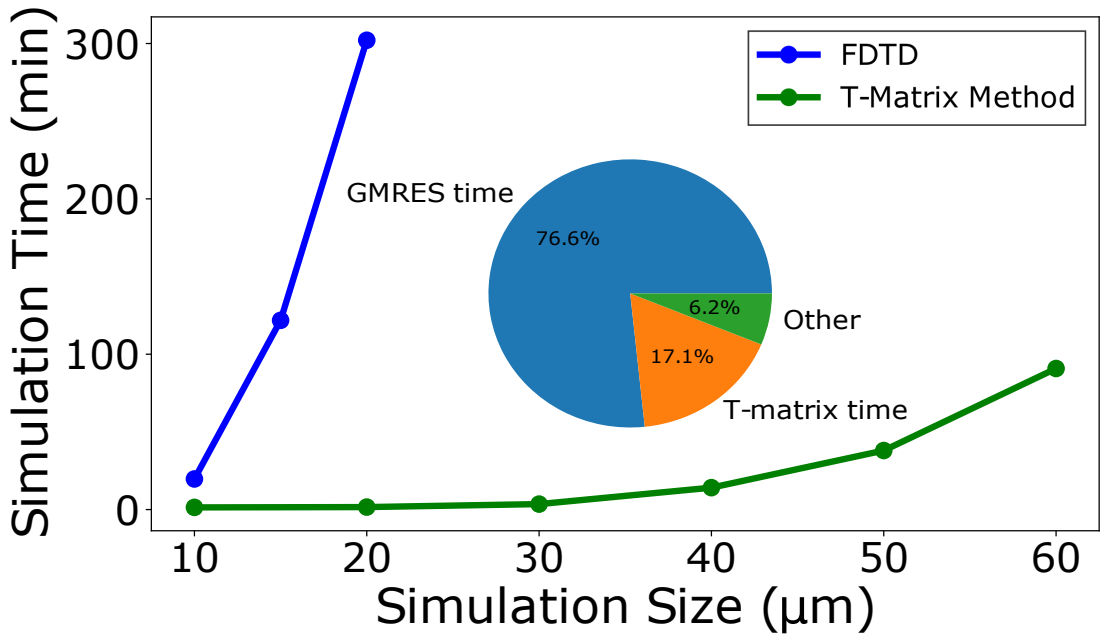


Figure A.2: **Timing benchmarks for our T-matrix implementation.** Simulation time versus simulation size for this single-GPU T-matrix method and for FDTD. The T-matrix method simulation was performed on a single V100 GPU, while the FDTD simulation was performed with Lumerical FDTD (Lumerical FDTD solutions, www.lumerical.com) [84] on 8 CPUs with 32 GB RAM and mesh accuracy level 3. The inset breaks down the total simulation time for the 20×20 μm surface into the GMRES solve time (76.6%), the time to compute the T-matrices (17.1%), and all other computation (e.g. computing the incident field coefficients on the spherical harmonic basis functions and expanding the scattered field coefficients on the basis functions to compute the scattered electric field; 6.2%).

been specialized to solving scattering from spheroidal scatterers [83, 19, 82, 20], or have not considered multiple scattering problems [35, 36]. In this section, we extend the formalism presented in [35], which deals with scattering from a single arbitrary shaped scatterer in vector spheroidal wavefunctions, to multiple scattering problems that are key to enable metasurface simulations using vector spheroidal wavefunctions. Subsection A.4.1 introduces the solutions of the scalar and vector Helmholtz equations in the spheroidal coordinates and their important properties, and subsection A.4.2

provides a solution to the multiple scattering problem in spheroidal coordinates.

A.4.1 Solution to vector Helmholtz equation in spheroidal coordinates

The prolate spheroidal coordinates describe the coordinates of a point with parameters (ξ, η, ϕ) where $\xi \in [1, \infty)$, $\eta \in (-1, 1)$ and $\phi \in [0, 2\pi)$. These are related to the cartesian coordinates via:

$$x = f\sqrt{(\xi^2 - 1)(1 - \eta^2)}\cos\phi \tag{A.31a}$$

$$y = f\sqrt{(\xi^2 - 1)(1 - \eta^2)}\sin\phi \tag{A.31b}$$

$$z = f\xi\eta \tag{A.31c}$$

Here we have introduced a parameter f which will be the focal length of the spheroidal coordinate system. Note that ϕ is identical to the azimuthal angle used in spherical and cylindrical coordinate systems. A few simple observations can easily be made about the spheroidal coordinate systems from Eq. A.31. Surface of constant ξ form ellipsoids with major axis being along the 'z' direction:

$$\frac{z^2}{\xi^2} + \frac{x^2 + y^2}{\xi^2 - 1} = f^2 \tag{A.32}$$

Furthermore, in the limit of $\xi \to \infty$, this ellipsoid tends to a sphere of radius $f\xi$. Similarly, surfaces of constant η form hyperboloids with major axis being along the 'z' direction:

$$\frac{z^2}{\eta^2} - \frac{x^2 + y^2}{1 - \eta^2} = f^2 \tag{A.33}$$

Note that η and $-\eta$ correspond to the same hyperboloid — the sign of η merely determines whether the point lies in the region $z > 0$ or $z < 0$.

The scalar helmholtz equation at wavenumber k is given by:

$$\nabla^2\psi(\mathbf{x}) + k^2\psi(\mathbf{x}) = 0 \tag{A.34}$$

The scalar helmholtz equation is separable in the prolate spheroidal coordinates. Similar to its solutions in the spherical coordinates, its solutions are indexed by two quantum numbers, l and m, and are given by:

$$\psi_{l,m}(\mathbf{x}) = R_{l,m}(\xi)S_{l,m}(\eta)\exp(\mathrm{i}m\phi) \tag{A.35}$$

$R_{l,m}(\xi)$ and $S_{l,m}(\eta)$ correspond to the 'radial' and 'polar' parts of the wavefunction and satisfy the following differential equations:

$$(\xi^2-1)\frac{d^2}{d\xi^2}R_{l,m}(\xi) + 2\xi\frac{d}{d\xi}R_{l,m}(\xi) + \left(h^2\xi^2 - \frac{m^2}{\xi^2-1}\right)R_{l,m}(\xi) = A_{l,m}R_{l,m}(\xi)\ \forall\ \xi \in (1,\infty) \tag{A.36a}$$

$$(1-\eta^2)\frac{d^2}{d\eta^2}S_{l,m}(\eta) - 2\eta\frac{d}{d\eta}S_{l,m}(\eta) - \left(h^2\eta^2 + \frac{m^2}{1-\eta^2}\right)S_{l,m}(\eta) = A_{l,m}S_{l,m}(\eta)\ \forall\ \eta \in (-1,1) \tag{A.36b}$$

where $h = kf$ and $A_{l,m}$ are separation constants that are introduced while applying the separation of variables method on the scalar helmholtz equation. Note that in general $A_{l,m}, S_{l,m}(\eta)$ and $R_{l,m}(\xi)$ depend on h. Several general properties of $S_{l,m}(\eta)$ and $R_{l,m}(\eta)$ to aid the following sections can be immediately deduced:

1. From Eq. A.36 that $S_{l,m}$ and $R_{l,m}$ can be chosen to depend only on $|m|$.

2. From Eq. A.36b, it follows that $S_{l,m}$ can be chosen so as to satisfy:

$$\int_{-1}^{1} S_{l,m}(\eta) S_{l',m}(\eta) d\eta = \delta_{l,l'} \tag{A.37}$$

3. There are two linearly independent solutions to Eq. A.36a — we will label them by $je_{l,m}(\xi)$ and $ne_{l,m}(\xi)$. Out of these two solutions, $je_{l,m}(\xi)$ is regular at $\xi = 1$ while $ne_{l,m}(\xi)$ is singular at $\xi = 1$. Furthermore, they have the following asymptotic forms:

$$\lim_{\xi\to\infty} je_{l,m}(\xi) \to \frac{1}{h\xi} \sin\left[h\xi - (l+1)\frac{\pi}{2}\right] \tag{A.38a}$$

$$\lim_{\xi\to\infty} ne_{l,m}(\xi) \to \frac{1}{h\xi} \cos\left[h\xi - (l+1)\frac{\pi}{2}\right] \tag{A.38b}$$

Furthermore, we can construct an "outgoing" solution $he^{(1)}_{l,m}(\xi) = ne_{l,m}(\xi) + \mathrm{i} je_{l,m}(\xi)$ which has the asymptotic form:

$$\lim_{\xi\to\infty} he_{l,m}(\xi) \to \frac{\exp(\mathrm{i}h\xi)}{h\xi} \tag{A.39}$$

From the solutions of the scalar Helmholtz equation, we can construct the solutions of the vector Helmholtz equation, which is given by:

$$\nabla \times \nabla \times \vec{F}(\vec{x}) = k^2 \vec{F}(\vec{x}) \tag{A.40}$$

There are many ways of this construction, here we use the one which bears closest resemblance to the construction of vector spherical harmonics.

$$\Phi_{l,m,p=0}(\vec{x}) = \nabla \times (\vec{x}\psi_{l,m}(\vec{x})) = \vec{x} \times \nabla\psi_{l,m}(\vec{x}) \tag{A.41a}$$

$$\Phi_{l,m,p=1}(\vec{x}) = \frac{\nabla \times \Phi_{l,m,p=0}(\mathbf{x})}{k} \tag{A.41b}$$

It can also be seen that:

$$\Phi_{l,m,p=0}(\mathbf{x}) = -\frac{\nabla \times \Phi_{l,m,p=1}(\mathbf{x})}{k} \tag{A.42}$$

Furthermore, for each (l, m, p), there will be a regular and singular (outgoing) solution for $\Phi_{l,m,p}$. We will denote the regular solution by a prefix 'Re' and the singular solution by a prefix 'Si' — consequently, the solutions to Helmholtz equation are $\mathcal{S}\Phi_{l,m,p}(\mathbf{x})$ and $\mathcal{R}\Phi_{l,m,p}(\mathbf{x})$. We point out that the singular solutions to Helmholtz equation satisfy Eq. A.40 except for at $\mathbf{x} \neq 0$. Of particular interest are the asymptotic forms for $\Phi_{l,m,p}$. These can be evaluated using Eq. A.41 and are given by:

$$\lim_{\xi\to\infty} \Phi_{l,m,p=0}(\mathbf{x}) = \left(-\sqrt{1-\eta^2}\frac{d}{d\eta}S_{l,m}(\eta)\hat{\phi} + \frac{\mathrm{i}mS_{l,m}(\eta)}{\sqrt{1-\eta^2}}\hat{\eta}\right) R_{l,m}(\xi)e^{\mathrm{i}m\phi} + \mathcal{O}\left(\frac{1}{\xi^2}\right) \tag{A.43a}$$

$$\lim_{\xi\to\infty} \Phi_{l,m,p=1}(\mathbf{x}) = \left(\frac{\mathrm{i}mS_{l,m}(\eta)}{\sqrt{1-\eta^2}}\hat{\phi} + \sqrt{1-\eta^2}\frac{d}{d\eta}S_{l,m}(\eta)\hat{\eta}\right) \frac{d}{d\xi}R_{l,m}(\xi)e^{\mathrm{i}m\phi} + \mathcal{O}\left(\frac{1}{\xi^2}\right) \tag{A.43b}$$

where $R_{l,m}(\xi)$ is $je_{l,m}(\xi)$ for the regular solution and $he_{l,m}(\xi)$ for the singular solution. We immediately notice that as $\xi \to \infty$, the two vector spheroidal wavefunctions become orthogonal to each other at each point in space.

A.4.2 Multiple scattering problem in spheroidal coordinates

As a starting point for the solution of the multiple-scattering problem, we establish some relationships between the vector spheroidal wavefunctions.Suppose $\vec{F}_1(\mathbf{x})$ and $\vec{F}_2(\mathbf{x})$ are two vector fields and Γ is some volume in space, then we can define the

following functional between the two vector fields:

$$\mathfrak{F}_\Gamma[\vec{F}_1, \vec{F}_2] = \int_{\partial\Gamma} \left[\vec{F}_1 \times (\nabla \times \vec{F}_2) - \vec{F}_2 \times (\nabla \times \vec{F}_1)\right] \cdot d\vec{S} \tag{A.44}$$

If $\vec{F}_1$ and $\vec{F}_2$ satisfy Eq. A.40 at all points inside the volume Γ, then a straightforward application of Gauss' law gives $\mathfrak{F}(\vec{F}_1, \vec{F}_2) = 0$. The following relationships between the vector spheroidal wavefunctions can then be deduced:

(a) We thus immediately deduce that along any closed volume Γ:

$$\mathfrak{F}[\mathcal{R}\Phi_{l,m,p}(\mathbf{x} - \mathbf{x}_0), \mathcal{R}\Phi_{l',m',p'}(\mathbf{x} - \mathbf{x}'_0)] = 0 \tag{A.45}$$

for any vectors $\mathbf{x}_0$ and $\mathbf{x}'_0$. Note that we have used the fact that if $\vec{F}(\mathbf{x})$ satisfies Eq. A.40 at all points in space, so does $\vec{F}(\mathbf{x} - \mathbf{x}_0)\ \forall\ \mathbf{x}_0$.

(b) Consider next the evaluation of $\mathfrak{F}_\Gamma[\Phi_{l,m,p}(\mathbf{x}), \Phi_{l',m',p'}(\mathbf{x})]$ in a volume Γ that contains the origin. Note that if Γ does not contain the origin, then the functional would evaluate to 0. To do so, we construct a volume Γ_∞ as shown in Fig. A.3a. Clearly, since the volume $\Gamma_\infty - \Gamma$ does not contain the origin, $\mathfrak{F}_{\Gamma_\infty - \Gamma}[\Phi_{l,m,p}(\mathbf{x}), \Phi_{l',m',p'}(\mathbf{x})] = 0$. Furthermore,

$$\mathfrak{F}_{\Gamma_\infty - \Gamma}[\Phi_{l,m,p}(\mathbf{x}), \Phi_{l',m',p'}(\mathbf{x})] = \mathfrak{F}_{\Gamma_\infty}[\Phi_{l,m,p}(\mathbf{x}), \Phi_{l',m',p'}(\mathbf{x})] - \mathfrak{F}_\Gamma[\Phi_{l,m,p}(\mathbf{x}), \Phi_{l',m',p'}(\mathbf{x})] \tag{A.46}$$

from which it immediately follows that:

$$\mathfrak{F}_\Gamma[\Phi_{l,m,p}(\mathbf{x}), \Phi_{l',m',p'}(\mathbf{x})] = \mathfrak{F}_{\Gamma_\infty}[\Phi_{l,m,p}(\mathbf{x}), \Phi_{l',m',p'}(\mathbf{x})] \tag{A.47}$$

While evaluating the functional on Γ_∞, we can greatly simplify our calculation and use the far field forms in Eq. A.43. Furthermore, as $\xi \to \infty$,

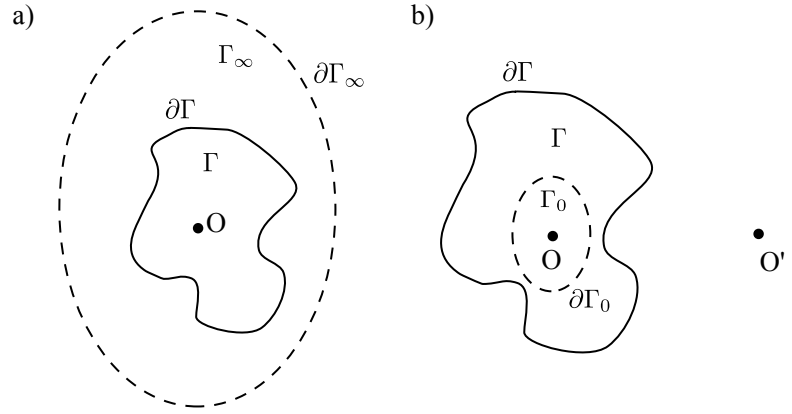


Figure A.3: Schematics for computing the functional $\mathfrak{F}_{\partial\Gamma}[\cdot,\cdot]$ — Setup for establishing the relations the functional between vector spheroidal wavefunctions with the a) same center and b) displaced centers.

$d\vec{S} = \hat{\xi} f^2 \xi^2 d\eta d\phi$. We now consider the various combinations of the values of p and p':

- $p = p' = 0$: In this case,

$$\begin{aligned}
&\mathfrak{F}_{\Gamma_\infty}[\Phi_{l,m,0}(\mathbf{x}), \Phi_{l',m',0}(\mathbf{x})] \\
&= \lim_{\xi_0\to\infty} kf^2\xi_0^2 \int_{\partial\Gamma_\infty} \left[\Phi_{l,m,0}(\mathbf{x}) \times \Phi_{l',m',1}(\mathbf{x}) - \Phi_{l',m',0}(\mathbf{x}) \times \Phi_{l,m,1}(\mathbf{x})\right] \cdot \hat{\xi} d\eta d\phi \\
&= kf^2 \delta_{m,-m'} \Omega_{l,l';m} \left[\lim_{\xi_0\to\infty} \xi_0^2 \left(R_{l,m}(\xi_0) \frac{d}{d\xi_0} R_{l',m}(\xi_0) - R_{l',m}(\xi_0) \frac{d}{d\xi_0} R_{l,m}(\xi_0) \right)\right]
\end{aligned} \tag{A.48}$$

where

$$\begin{aligned}
\Omega^m_{l,l'} &= -\int_{-1}^{1} \left[(1-\eta^2) \frac{d}{d\eta} S_{l,m}(\eta) \frac{d}{d\eta} S_{l',m}(\eta) + \frac{m^2}{1-\eta^2} S_{l,m}(\eta) S_{l',m}(\eta) \right] \mathrm{d}\eta \\
&= A_{l,m}\delta_{l,l'} + \int_{-1}^{1} h^2 \eta^2 S_{l,m}(\eta) S_{l',m}(\eta) d\eta
\end{aligned} \tag{A.49}$$

From Eq. A.48, it immediately follows that:

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,0}(\mathbf{x}), \mathcal{S}\Phi_{l,m,0}(\mathbf{x})] = 0 \tag{A.50a}$$

$$\mathfrak{F}_\Gamma[\mathcal{R}\Phi_{l,m,0}(\mathbf{x}), \mathcal{S}\Phi_{l,m,0}(\mathbf{x})] = -f\delta_{m,-m}\Omega^m_{l,l'} \tag{A.50b}$$

- $p = p' = 1$: This case is very similar to the previous one, with the final result being:

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,0}(\mathbf{x}), \mathcal{S}\Phi_{l,m,0}(\mathbf{x})] = 0 \tag{A.51a}$$

$$\mathfrak{F}_\Gamma[\mathcal{R}\Phi_{l,m,0}(\mathbf{x}), \mathcal{S}\Phi_{l,m,0}(\mathbf{x})] = -f\delta_{m,-m}\Omega^m_{l,l'} \tag{A.51b}$$

- $p = 0, p' = 1$: In this case, using the asymptotic forms for the vector wavefunctions, we immediately see that:

$$\begin{aligned}
&\mathfrak{F}_{\Gamma_\infty}[\Phi_{l,m,0}(\mathbf{x}), \Phi_{l,m,1}(\mathbf{x})] \\
&= \lim_{\xi_0\to\infty} -kf^2\xi_0^2 \int_{\partial\Gamma_\infty} \Big[\Phi_{l,m,0}(\mathbf{x}) \times \Phi_{l',m',0}(\mathbf{x}) + \Phi_{l,m,1}(\mathbf{x}) \times \Phi_{l',m',1}(\mathbf{x})\Big] \cdot \hat{\xi} d\eta d\phi \\
&= \lim_{\xi_0\to\infty} -kf^2\xi_0^2\delta_{m,-m'} \Big[im \int_{-1}^{1} \frac{d}{d\eta}(S_{l,m}(\eta)S_{l',m}(\eta))d\eta\Big)\Big] \\
&\Big[\frac{d}{d\xi_0}R_{l,m}(\xi_0)\frac{d}{d\xi_0}R_{l',m}(\xi_0) + R_{l,m}(\xi_0)R_{l',m}(\xi_0)\Big]
\end{aligned} \tag{A.52a}$$

It can be noted from Eq. A.36b that if $m \neq 0$, then the differential equation has a singularity at $\eta = \pm 1$ and consequently $S_{l,m}(\pm 1) = 0$. Therefore, we immediately conclude that the RHS of Eq. A.52 evaluates to 0. Therefore,

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,0}(\mathbf{x}), \mathcal{S}\Phi_{l',m',1}(\mathbf{x})] = 0 \tag{A.53a}$$

$$\mathfrak{F}_\Gamma[\mathcal{R}\Phi_{l,m,0}(\mathbf{x}), \mathcal{S}\Phi_{l',m',1}(\mathbf{x})] = 0 \tag{A.53b}$$

- $p = 1, p' = 0$: Following a procedure similar to that of $p = 0, p' = 1$, it is

immediately evident that

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,1}(\mathbf{x}), \mathcal{S}\Phi_{l',m',1}(\mathbf{x})] = 0 \tag{A.54a}$$

$$\mathfrak{F}_\Gamma[\mathcal{R}\Phi_{l,m,1}(\mathbf{x}), \mathcal{S}\Phi_{l',m',1}(\mathbf{x})] = 0 \tag{A.54b}$$

(c) We finally consider the situation shown in Fig. A.3b — we wish to compute $\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0), \mathcal{R}\Phi_{l',m',p'}(\mathbf{x})]$ and $\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0), \mathcal{R}\Phi_{l',m',p'}(\mathbf{x})]$ where O is assumed to be at the origin and O' has coordinates given by the vector $\mathbf{x}'_0$. Note that since Γ does not enclose O', it follows that $\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0)$ is not singular within Γ. Consequently, it immediately follows that

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0), \mathcal{R}\Phi_{l',m',p'}(\mathbf{x})] = 0 \tag{A.55}$$

Consider now the evaluation of $\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0), \mathcal{S}\Phi_{l',m',p'}(\mathbf{x})]$: To do so, we consider an ellipsoid Γ_0 chosen such that $\forall \mathbf{x} \in \partial\Gamma_0, |\mathbf{x}| < |\mathbf{x}'_0|$. Furthermore, both $\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0)$ and $\mathcal{S}\Phi_{l',m',p'}(\mathbf{x})$ are regular within $\Gamma - \Gamma_0$. Therefore:

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0), \mathcal{S}\Phi_{l',m',p'}(\mathbf{x})] = \mathfrak{F}_{\Gamma_0}[\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0), \mathcal{S}\Phi_{l',m',p'}(\mathbf{x})] \tag{A.56}$$

Now, $\forall \mathbf{x} \in \partial\Gamma_0$, from translation theorem [21]:

$$\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0) = \sum_{l^*,m^*} \tau^p_{l,m;l^*,m^*}(\mathbf{x}'_0)\mathcal{R}\Phi_{l^*,m^*,p}(\mathbf{x}) \tag{A.57}$$

Therefore, from Eqs. A.56, A.50 and A.51, it follows that:

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_{l,m,p}(\mathbf{x}-\mathbf{x}'_0), \mathcal{S}\Phi_{l',m',p'}(\mathbf{x})] = -f\delta_{p,p'} \sum_{l^*} \tau^p_{l,m;l^*,-m'}\Omega^{m'}_{l^*,l'} \tag{A.58}$$

For ease of notation, the above results can be summarized as:

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_\lambda(\mathbf{x};h), \mathcal{R}\Phi_{\lambda'}(\mathbf{x};h)] = M_{\lambda,\lambda'}(h) \tag{A.59a}$$

$$\mathfrak{F}_\Gamma[\mathcal{S}\Phi_\lambda(\mathbf{x};h), \mathcal{S}\Phi_{\lambda'}(\mathbf{x}+\vec{d};h)] = D_{\lambda,\lambda'}(\vec{d};h) \tag{A.59b}$$

where we have collected the indices (l, m, p) into λ, and also displayed the dependence of the spheroidal wavefunctions on $h = kf$. Furthermore, if we compactly express the translation coefficients via $T_{\lambda',\lambda}(\vec{d})$ where:

$$\mathcal{S}\Phi_\lambda(\mathbf{x}+\vec{d};h) = \sum_{\lambda'} T_{\lambda',\lambda}(\vec{d};h)\mathcal{R}\Phi_{\lambda'}(\mathbf{x};h) \tag{A.60}$$

then we have the relationship

$$D_{\lambda,\lambda'}(\vec{d};h) = \sum_{\mu} M_{\lambda,\mu}(h)T_{\mu,\lambda'}(\vec{d};h) \tag{A.61}$$

For truncation with a finite number of spheroidal basis function, it will be convenient to introduce matrices $\mathbf{M}(h)$, $\mathbf{D}(\vec{d};h)$ and $\mathbf{T}(\vec{d};h)$ formed from $M_{\lambda,\lambda'}(h), D_{\lambda,\lambda'}(\vec{d};h)$ and $T_{\lambda,\lambda'}(\vec{d};h)$ which would then satisfy:

$$\mathbf{D}(\vec{d};h) = \mathbf{M}(h)\mathbf{T}(\vec{d};h) \tag{A.62}$$

Finally, we consider N scatterers centered at $\vec{x}_1, \vec{x}_2 \ldots \vec{x}_N$. We assume that the focal length f of the ellipsoidal coordinate system is chosen so that the bounding ellipsoids of the scatterers do not intersect with each other. We will denote by $\partial\Gamma_i$ the surface of the i^{th} scatterer and by $\partial\Gamma_i^+$ the surface of the bounding ellipsoid for the i^{th} scatterer. Without proof, we assume that the scattered field in the region

outside all the bounding ellipsoids can be expressed as the following superposition:

$$\vec{E}_{\rm sca}(\vec{x}) = \sum_{i=1}^{N} \sum_{\lambda} s_{\lambda}^{i} \mathcal{S}\Phi_{\lambda}(\vec{x} - \vec{x}_i) \tag{A.63}$$

The scatterers are illuminated by a plane wave $\mathbf{E}_{\rm inc}(\mathbf{x})$ which can be described by:

$$\mathbf{E}_{\rm inc}(\mathbf{x}) = \sum_{\lambda} a_{\lambda}^{i} \mathcal{R}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i) \tag{A.64}$$

Furthermore, we will denote by $\mathbf{E}(\mathbf{x})$ the solution to Maxwell's equations in all space. Clearly, in the region outside the bounding ellipsoids of all the scatterers, $\mathbf{E}(\mathbf{x}) = \mathbf{E}_{\rm sca}(\mathbf{x}) + \mathbf{E}_{\rm inc}(\mathbf{x})$. Furthermore, since both $\mathbf{E}(\mathbf{x})$ and $\mathbf{E}_{\rm sca}(\mathbf{x}) + \mathbf{E}_{\rm inc}(\mathbf{x})$ do not have any singularities in the region between $\partial\Gamma_i$ and $\partial\Gamma_i^+$, it follows that:

$$\mathfrak{F}_{\partial\Gamma_i}[\mathbf{E}(\mathbf{x}), \mathcal{R}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i)] = \mathfrak{F}_{\partial\Gamma_i^+}[\mathbf{E}_{\rm sca}(\mathbf{x}) + \mathbf{E}_{\rm inc}(\mathbf{x}), \mathcal{R}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i)] \tag{A.65a}$$

$$\mathfrak{F}_{\partial\Gamma_i}[\mathbf{E}(\mathbf{x}), \mathcal{S}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i)] = \mathfrak{F}_{\partial\Gamma_i^+}[\mathbf{E}_{\rm sca}(\mathbf{x}) + \mathbf{E}_{\rm inc}(\mathbf{x}), \mathcal{S}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i)] \tag{A.65b}$$

Substituting the expressions for $\mathbf{E}(\mathbf{x})$ and $\mathbf{E}_{\rm inc}(\mathbf{x})$, we obtain:

$$\mathfrak{F}_{\partial\Gamma_i}[\mathbf{E}(\mathbf{x}), \mathcal{R}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i)] = \sum_{\mu} M_{\lambda,\mu}(h_b) s_{\mu}^{i} \tag{A.66a}$$

$$\mathfrak{F}_{\partial\Gamma_i^+}[\mathbf{E}(\mathbf{x}), \mathcal{S}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i)] = -\sum_{\mu} M_{\lambda,\mu}(h_b) a_{\mu}^{i} - \sum_{j \neq i} \sum_{\mu} s_{\mu}^{j} D_{\lambda,\mu}(\mathbf{x}_i - \mathbf{x}_j; h_b) \tag{A.66b}$$

Furthermore, we expand $\mathbf{E}(\mathbf{x})$ at the surface $\partial\Gamma_i$ in terms of $\mathcal{R}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i; k_i f)$:

$$\mathbf{E}(\mathbf{x}) = \sum_{\lambda} b_{\lambda}^{i} \mathcal{R}\Phi_{\lambda}(\mathbf{x} - \mathbf{x}_i; k_i f) \tag{A.67}$$

Introducing $Q^i_{\lambda,\mu}$ and $\hat{Q}^i_{\lambda,\mu}$ via:

$$Q^i_{\lambda,\mu} = \mathfrak{F}_{\partial\Gamma_i}[\mathcal{R}\Phi_\mu(\mathbf{x}-\mathbf{x}_i;k_i f), \mathcal{R}\Phi_\lambda(\mathbf{x}-\mathbf{x}_i;k_b f)] \text{ and}$$
$$\hat{Q}^i_{\lambda,\mu} = \mathfrak{F}_{\partial\Gamma_i}[\mathcal{R}\Phi_\mu(\mathbf{x}-\mathbf{x}_i;k_i f), \mathcal{S}\Phi_\lambda(\mathbf{x}-\mathbf{x}_i;k_b f)] \tag{A.68}$$

we obtain:

$$\mathbf{Q}^i\mathbf{b}^i = \mathbf{M}\mathbf{s}^i \tag{A.69a}$$
$$\hat{\mathbf{Q}}^i\mathbf{b}^i = -\mathbf{M}\vec{A}^i - \sum_{j\neq i}\mathbf{M}\mathbf{T}^i(\mathbf{x}_i-\mathbf{x}_j)\mathbf{s}^j \tag{A.69b}$$

which are the desired transition matrix equations. Note that these are identical to the transition matrix equation for spherical coordinates, with the transition matrix given by $\mathbf{T}^i = -\mathbf{M}^{-1}\mathbf{Q}^i[\hat{\mathbf{Q}}^i]^{-1}\mathbf{M}$.

A.4.3 Implementation of Spheroidal Scalar Wavefunctions

Separation of variables on the scalar Helmholtz equation yields the following differential equation with respect to the coordinate η:

$$\frac{d}{dx}\left[(1-x^2)\frac{d}{dx}S^m(x)\right] - \left(h^2x^2 + \frac{m^2}{1-x^2}\right)S^m(x) = \lambda^m S^m(x) \text{ for } x\in[-1,1] \tag{A.70}$$

Here, λ^m and $S^m(x)$ are both to be determined. Note that the differential equation only depends on the magnitude of m. Consequently, we will choose $S^{-m}(x) = S^m(x)$ and $\lambda^m = \lambda^{-m}$ and henceforth in this section, we will assume m to be non-negative. In order to solve this differential equation, we expand $S^m(x)$ into a sum of the associated

Legendre polynomials $P_l^m(x)$:

$$S^m(x) = \sum_{r=0}^{\infty} d^e_{m,r} P^m_{m+2r}(x) + \sum_{r=0}^{\infty} d^o_{m,r} P^m_{m+2r+1}(x) \tag{A.71}$$

Note that the $P_l^m(x)$ satisfy:

$$\frac{d}{dx}\left[(1-x^2)\frac{d}{dx}P_l^m(x)\right] - \frac{m^2}{1-x^2}P_l^m(x) = l(l+1)P_l^m(x) \text{ for } x \in [-1,1] \tag{A.72}$$

Next, we would like to evaluate the coefficients $d^{e/o}_{m,r}$. To obtain a relationship between these coefficients from Eqs. A.71 and A.70, it is necessary to express $x^2 P_l^m(x)$ in terms of $P_l^m(x)$. The starting point for this is the following recursion for the associated legendre polynomials:

$$xP_l^m(x) = \frac{l-m+1}{2l+1}P^m_{l+1}(x) + \frac{l+m}{2l+1}P^m_{l-1}(x) \tag{A.73}$$

where it is assumed that $P_l^m(x) = 0$ for $l < m$. Repeated application of this recursion allows us obtain:

$$\begin{aligned} &x^2 P_l^m(x) = \\ &\frac{(l-m+1)(l+2-m)}{(2l+1)(2l+3)}P^m_{l+2}(x) + \left[\frac{(l-m+1)(l+1+m)}{(2l+1)(2l+3)} + \frac{(l+m)(l-m)}{(2l+1)(2l-1)}\right]P_l^m(x) + \\ &\frac{(l+m)(l-1+m)}{(2l+1)(2l-1)}P^m_{l-2}(x) \end{aligned} \tag{A.74}$$

Using this recursion along with Eqs. A.70 and A.71, we obtain the following recursions:

$$\beta^m_r d^e_{m,r-1} + (\alpha^m_r - \lambda^m) d^e_{m,r} + \gamma^m_r d^e_{m,r+1} = 0 \tag{A.75a}$$

$$\beta^m_{r+1/2} d^o_{m,r-1} + (\alpha^m_{r+1/2} - \lambda^m) d^o_{m,r} + \gamma^m_{r+1/2} d^o_{m,r+1} = 0 \tag{A.75b}$$

where

$$\alpha_x^m = (m+2x)(m+2x+1) + h^2\left[\frac{(2x+1)(2m+2x+1)}{(2m+4x+1)(2m+4x+3)} + \frac{2x(2m+2x)}{(2m+4x+1)(2m+4x-1)}\right] \tag{A.76a}$$

$$\beta_x^m = h^2\frac{2x(2x-1)}{(2m+4x-3)(2m+4x-1)} \tag{A.76b}$$

$$\gamma_x^m = h^2\frac{(2m+2x+2)(2m+2x+1)}{(2m+4x+5)(2m+4x+3)} \tag{A.76c}$$

To compute λ^m from the recursion in Eq. A.75, we rewrite it as an eigenvalue equation. Noting from definition of β_x^m that $\beta_0^m = \beta_{1/2}^m = 0$, it follows that

$$\begin{bmatrix} \alpha_0^m & \gamma_0^m & 0 & 0 & \dots & 0 \\ \beta_1^m & \alpha_1^m & \gamma_1^m & 0 & \dots & 0 \\ 0 & \beta_2^m & \alpha_2^m & \gamma_2^m & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} d_{m,0}^e \\ d_{m,1}^e \\ d_{m,2}^e \\ \vdots \end{bmatrix} = \lambda^m \begin{bmatrix} d_{m,0}^e \\ d_{m,1}^e \\ d_{m,2}^e \\ \vdots \end{bmatrix} \tag{A.77a}$$

$$\begin{bmatrix} \alpha_{1/2}^m & \gamma_{1/2}^m & 0 & 0 & \dots & 0 \\ \beta_{3/2}^m & \alpha_{3/2}^m & \gamma_{3/2}^m & 0 & \dots & 0 \\ 0 & \beta_{5/2}^m & \alpha_{5/2}^m & \gamma_{5/2}^m & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} d_{m,0}^o \\ d_{m,1}^o \\ d_{m,2}^o \\ \vdots \end{bmatrix} = \lambda^m \begin{bmatrix} d_{m,0}^o \\ d_{m,1}^o \\ d_{m,2}^o \\ \vdots \end{bmatrix} \tag{A.77b}$$

These are the two eigenvalue problems that need to be solved (after truncation) in order to setup the spheroidal wavefunctions. We will refer to the first one as the even eigenvalue problem and the second one as the odd eigenvalue problem. The eigenvectors and eigenvalues obtained will be indexed by the orbital number $l \geq m$ — the scheme of assignment of the orbital number is as follows: the solution corresponding to the smallest eigenvalue of the even problem will be assigned $l = m$, that corresponding to the smallest eigenvalue of the odd problem will be assigned $l = m + 1$, that corresponding to the second smallest eigenvalue of the even problem

$l = m + 1$ and so on. Note that with this assignment scheme, a solution for which $l - m$ is even corresponds to the first eigenproblem and a solution for which $l - m$ is odd corresponds to the second eigenproblem.

For the purposes of computation of the hankel-like radial wavefunction in spheroidal coordinates, it is necessary to be able to compute the values of the coefficients $d^{o/e}_{m,r}$ for negative values of r. Consider extending $d^e_{m,r}$ to negative r. Noting from Eq. A.76 that $\gamma^m_{-m-1} = 0$. This immediately implies that $d^e_{m,r} = 0 \; \forall \; r < -m$. To evaluate $d^e_{m,r}$ for $-m \leq r \leq -1$, we use Eq. A.75 to obtain a system of equations for $d^e_{m,r}$:

$$\begin{bmatrix} \alpha^m_{-1} - \lambda^m & \beta^m_{-1} & 0 & 0 & \dots & 0 & 0 \\ \gamma^m_{-2} & \alpha^m_{-2} - \lambda^m & \beta^m_{-2} & 0 & \dots & 0 & 0 \\ 0 & \gamma^m_{-3} & \alpha^m_{-3} - \lambda^m & \beta^m_{-3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \gamma^m_{-m} & \alpha^m_{-m} - \lambda^m \end{bmatrix} \begin{bmatrix} d^e_{m,-1} \\ d^e_{m,-2} \\ d^e_{m,-3} \\ d^e_{m,-4} \\ \vdots \\ d^e_{m,-(m-1)} \\ d^e_{m,-m} \end{bmatrix} = \begin{bmatrix} -\gamma^m_{-1} d^e_{m,0} \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{A.78}$$

Here, $d^e_{m,0}$ has already been evaluated via the solution of the eigenvalue problem in Eq. A.77. Similarly, noting from Eq. A.76 that $\gamma^m_{-m-1/2} = 0$ and therefore $d^o_{m,r} = 0 \; \forall \; r < -m$. To evaluate $d^o_{m,r}$ for $-m \leq r \leq -1$, we use Eq. A.75 to obtain a system

of equations for $d^o_{m,r}$:

$$\begin{bmatrix} \alpha^m_{-1/2}-\lambda^m & \beta^m_{-1/2} & 0 & 0 & \dots & 0 & 0 \\ \gamma^m_{-3/2} & \alpha^m_{-3/2}-\lambda^m & \beta^m_{-3/2} & 0 & \dots & 0 & 0 \\ 0 & \gamma^m_{-5/2} & \alpha^m_{-5/2}-\lambda^m & \beta^m_{-5/2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \gamma^m_{-m+1/2} & \alpha^m_{-m+1/2}-\lambda^m \end{bmatrix} \begin{bmatrix} d^o_{m,-1} \\ d^o_{m,-2} \\ d^o_{m,-3} \\ d^o_{m,-4} \\ \vdots \\ d^o_{m,-(m-1)} \\ d^o_{m,-m} \end{bmatrix} = \begin{bmatrix} -\gamma^m_{-1/2} d^o_{m,0} \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{bmatrix} \tag{A.79}$$

For the purpose of computing the hankel-like radial function, it is necessary to evaluate a complementary eigenvector components. This is defined by

$$\Delta^{o/e}_{m,p} = \lim_{x\to 0} \frac{d^{o/e}_{m,-m-p+x}}{x} \;\forall\; p \geq 1 \tag{A.80}$$

where the limit is understood to be taken of an extension of the closed form solution to $d^{o/e}_{m,r}$ over all real numbers.

Consider first the evaluation of $\Delta^e_{m,p}$: Note from Eq. A.75 for $r = -m-1+x$

that:

$$\beta^m_{-m-1+x} d^e_{m,-m-2+x} + (\alpha^m_{-m-1-x} - \lambda^m) d^e_{m,-m-1+x} + \gamma^m_{-m-1+x} d^e_{m,-m+x} = 0 \quad \text{(A.81)}$$

It is also worthwhile noting from Eq. A.76 that

$$\gamma^m_{-m-1+x} = -\frac{2xh^2}{(2m-1)(2m+1)} \quad \text{(A.82)}$$

Noting that $\lim_{x\to 0} d^e_{m,-m+x} = d^e_{m,-m} \neq 0$, it follows from Eq. A.81 that:

$$\beta^m_{-m-1} \Delta^e_{m,2} + (\alpha^m_{-m-1} - \lambda^m) \Delta^e_{m,1} - \frac{2h^2}{(2m-1)(2m+1)} d^e_{m,-m} = 0 \quad \text{(A.83)}$$

Similarly, using Eq. A.75 for $r = -m - p + x$ for $p > 1$, dividing by x and taking limit of $x \to 0$ we obtain:

$$\beta^m_{-m-p} \Delta^e_{m,p+1} + (\alpha^m_{-m-p} - \lambda^m) \Delta^e_{m,p} + \gamma^m_{-m-p} \Delta^e_{m,p-1} = 0 \quad \text{(A.84)}$$

Eqs. A.81 and A.84 can be collected into a system of equation that can be truncated and solved:

$$\begin{bmatrix} \alpha^m_{-m-1} - \lambda^m & \beta^m_{-m-1} & 0 & 0 & \dots \\ \gamma^m_{-m-2} & \alpha^m_{-m-2} - \lambda^m & \beta^m_{-m-2} & 0 & \dots \\ 0 & \gamma^m_{-m-3} & \alpha^m_{-m-3} - \lambda^m & \beta^m_{-m-3} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \Delta^e_{m,1} \\ \Delta^e_{m,2} \\ \Delta^e_{m,3} \\ \Delta^e_{m,4} \\ \vdots \end{bmatrix} = \frac{2h^2 d^e_{m,-m}}{(2m-1)(2m+1)} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad \text{(A.85)}$$

Similarly, consider the evaluation of $\Delta^o_{m,p}$: Note from Eq. A.75 for $r = -m-1+x$

that

$$\beta^m_{-m-1/2+x} d^o_{m,-m-2+x} + (\alpha^m_{-m-1/2+x} - \lambda^m) d^o_{m,-m-1+x} + \gamma^m_{-m-1/2+x} d^o_{m,-m+x} = 0 \tag{A.86}$$

It is also worthwhile noting from Eq. A.76 that

$$\gamma^m_{-m-1/2+x} = \frac{2xh^2}{(2m-3)(2m-1)} \tag{A.87}$$

Therefore,

$$\beta^m_{-m-1/2} \Delta^o_{m,2} + (\alpha^m_{-m-1/2} - \lambda^m) \Delta^o_{m,1} + \frac{2h^2}{(2m-1)(2m-3)} d^o_{m,-m} = 0 \tag{A.88}$$

It also follows from Eq. A.75 that for $p > 1$ that

$$\beta^m_{-m-p+1/2} \Delta^o_{m,p+1} + (\alpha^m_{-m-p+1/2} - \lambda^m) \Delta^o_{m,p} + \gamma^m_{-m-p+1/2} \Delta^o_{m,p-1} = 0 \tag{A.89}$$

Collective these into a system of equations, which can be truncated and solved:

$$\begin{bmatrix} \alpha^m_{-m-1/2} - \lambda^m & \beta^m_{-m-1/2} & 0 & 0 & \dots \\ \gamma^m_{-m-3/2} & \alpha^m_{-m-3/2} - \lambda^m & \beta^m_{-m-3/2} & 0 & \dots \\ 0 & \gamma^m_{-m-5/2} & \alpha^m_{-m-5/2} - \lambda^m & \beta^m_{-m-5/2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \Delta^o_{m,1} \\ \Delta^o_{m,2} \\ \Delta^o_{m,3} \\ \Delta^o_{m,4} \\ \vdots \end{bmatrix} = -\frac{2h^2 d^o_{m,-m}}{(2m-1)(2m-3)} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{A.90}$$

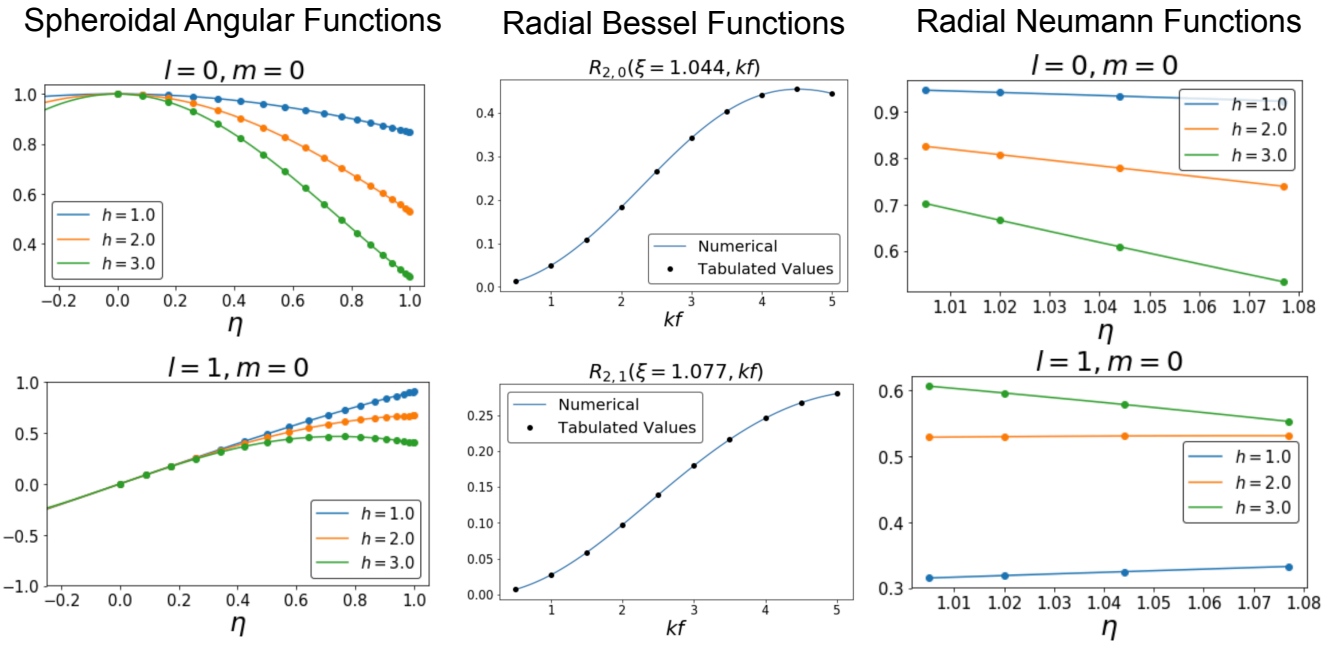


Figure A.4: Implementation of spheroidal special functions and their comparison against available lookup tables.

Once the eigenproblem is known, we can construct the 'bessel-like' radial wavefunctions as well using the following formula: If $l - m$ is even, then

$$R_l^m(\xi) = \frac{1}{N_l^m}\left(\frac{\xi^2 - 1}{\xi^2}\right)^{m/2} \sum_{k=0}^{\infty} (-1)^{(m-l)/2+k} \delta_k^{l,m} j_{m+2k}(h\xi) \tag{A.91a}$$

with

$$\delta_k^{l,m} = d_k^{l,m} \sqrt{2m + 4k + 1} \left[\frac{(2m + 2k)!}{(2k)!}\right]^{1/2} \text{ and } N_l^m = \sum_{k=0}^{\infty} \delta_k^{l,m} \tag{A.91b}$$

and if $l - m$ is odd, then

$$R_l^m(\xi) = \frac{1}{N_l^m}\left(\frac{\xi^2 - 1}{\xi^2}\right)^{m/2} \sum_{k=0}^{\infty} (-1)^{(m-l+1)/2+k} \delta_k^{l,m} j_{m+2k+1}(h\xi) \tag{A.91c}$$

with

$$\delta_k^{l,m} = c_k^{l,m}\sqrt{2m+4k+3}\left[\frac{(2m+2k+1)!}{(2k+1)!}\right]^{1/2} \text{ and } N_l^m = \sum_{k=0}^{\infty}\delta_k^{l,m} \tag{A.91d}$$

The construction of the 'neumann-like' radial wavefunctions, which are required for computing the outgoing solutions of the wave-equation, we perform an analytical continuation on $S_l^m(\eta)$ to η being more than 1. Fig. A.4 shows our implementation of these special functions along with comparison against documented lookup tables of function values available in literature.

A.4.4 Implementation of Spheroidal Vector Wavefunctions

Here, we describe a method for computing the vector spheroidal wavefunctions, $\Phi_{p=0}$ and $\Phi_{p=1}$, from the scalar spheroidal wavefunctions (Eq. A.41) using the spheroidal special functions implemented in Fig. A.4. Then, we discuss computing the coefficients for the incident field and the translation coefficients on the spheroidal basis functions using the incident field and translation coefficients on the spherical vector wavefunctions (as described in Section A.1). Since the implementation of the vector spherical wavefunctions is more straightforward and can be built on pre-existing efficient special function libraries, it is advantageous to harness as much of this infrastructure as possible for the vector spheroidal wavefunction coefficient computation.

Expanding Eq. A.41, we have:

$$\begin{aligned}
\Phi^{\xi}_{p=0} &= -\frac{f^2 m\eta \mathrm{i}\psi}{g_\phi g_\eta} \\
\Phi^{\phi}_{p=0} &= \frac{f^2}{g_\xi g_\eta}\left(\eta\frac{\partial\psi}{\partial\xi} - \xi\frac{\partial\psi}{\partial\eta}\right) \\
\Phi^{\eta}_{p=0} &= \frac{f^2 m\xi \mathrm{i}\psi}{g_\phi g_\xi} \qquad \text{(A.92)} \\
\Phi^{\xi}_{p=1} &= \frac{1}{k g_\eta g_\phi}\left[\left(-\frac{g_\eta f^2 m^2 \xi\psi}{g_\phi g_\xi}\right) - \frac{\partial}{\partial\eta}\left(\frac{g_\phi f^2}{g_\xi g_\eta}\left(\eta\frac{\partial\psi}{\partial\xi} - \xi\frac{\partial\psi}{\partial\eta}\right)\right)\right] \\
\Phi^{\phi}_{p=1} &= \frac{1}{k g_\xi g_\eta}\left[\frac{\partial}{\partial\eta}\left(-\frac{g_\xi m \mathrm{i} f^2 \eta\psi}{g_\eta g_\phi}\right) - \frac{\partial}{\partial\xi}\left(\frac{g_\eta f^2 m \mathrm{i}\xi\psi}{g_\xi g_\phi}\right)\right] \\
\Phi^{\phi}_{p=1} &= \frac{1}{k g_\xi g_\phi}\left[\frac{\partial}{\partial\xi}\left(\frac{g_\phi f^2}{g_\xi g_\eta}\left(\eta\frac{\partial\psi}{\partial\xi} - \xi\frac{\partial\psi}{\partial\eta}\right)\right) - \left(\frac{g_\xi f^2 m^2 \eta\psi}{g_\eta g_\phi}\right)\right] \qquad \text{(A.93)}
\end{aligned}$$

where $\vec{g}$ is the gradient of $\mathbf{x}$ with components:

$$\begin{aligned}
g_\xi &= \left|\frac{\partial\mathbf{x}}{\partial\xi}\right| = f\left(\frac{\xi^2-\eta^2}{\xi^2-1}\right)^{1/2} \\
g_\phi &= \left|\frac{\partial\mathbf{x}}{\partial\phi}\right| = f\left(\left(\xi^2-1\right)\left(1-\eta^2\right)\right)^{1/2} \\
g_\eta &= \left|\frac{\partial\mathbf{x}}{\partial\eta}\right| = f\left(\frac{\xi^2-\eta^2}{1-\eta^2}\right)^{1/2} \qquad \text{(A.94)}
\end{aligned}$$

The scalar spherical wavefunction ψ in Eq. A.92 is defined in terms of the spheroidal special functions from Fig. A.4 for standing basis (basis for incident field) as:

$$\psi_{l,m}(\mathbf{x}) = RadialBessel_{l,m}(\xi)Angular_{l,m}(\eta)\exp(\mathrm{i}m\phi) \qquad \text{(A.95)}$$

The scalar spherical wavefunction ψ in Eq. A.92 is defined in terms of the spheroidal special functions from Fig. A.4 for propagating basis (basis for scattered

field) as:

$$\psi_{l,m}(\mathbf{x}) = [RadialBessel_{l,m}(\xi) + \mathrm{i} * RadialNeumann_{l,m}(\xi)]\, Angular_{l,m}(\eta) \exp(\mathrm{i}m\phi) \tag{A.96}$$

As defined in [22], we can express spherical wavefunctions ${}^{\bullet}\vec{R\Phi}_{l,m}(r,\theta,\phi)$ as a sum of spheroidal wavefunctions $\vec{R\Phi}_{l,m}(h;\eta,\xi,\phi)$ as follows:

$${}^{\bullet}\vec{R\Phi}_{l,m}(r,\theta,\phi) = \sum_{b=|m|,|m|+1}^{\infty'} \Gamma_{b,m}^{l,m}(h)\vec{R\Phi}_{l,m}(h;\eta,\xi,\phi) \tag{A.97}$$

where $\Gamma_{b,m}^{l,m}(h)$ is defined as:

$$\Gamma_{b,m}^{l,m}(h) = \mathrm{i}^{b-l} \frac{{}^{\bullet}N_{l,m}}{N_{b,m}(h)} d_{l-|m|}^{b,m}(h) \tag{A.98}$$

${}^{\bullet}N_{l,m}$ and $N_{b,m}$ are the normalization factors of the associated Legendre functions for the spherical wavefunction computation and the normalization factors of the spheroidal angle function $S_{ml}^{(1)}$ for the spheroidal wavefunction computation [32], respectively:

$$\begin{aligned} {}^{\bullet}N_{l,m} &= \left(\frac{2l+1}{2}\frac{(l-m)!}{(l+m)!}\right)^{1/2} \\ N_{b,m}(h) &= 2\sum_{r=0,1}^{\infty'} \frac{(r+2b)!\left(d_r^{b,m}(h)\right)^2}{(2r+2b+1)r!} \end{aligned} \tag{A.99}$$

The conversion in Eq. A.97 allows us to compute the spheroidal basis incident field coefficients from the incident field coefficients on the spherical basis functions

(whose computation procedure is detailed in Section A.2):

$$a_{l,m} = \sum_{b=|m|,|m|+1}^{\infty'} \left({}^{\bullet}a_{b,m} \Gamma_{l,m}^{b,m}(h) \right) \tag{A.100}$$

A similar conversion can be applied to compute the translation coefficients on the spheroidal basis from the spherical basis translation coefficients. Then, with the incident field and translation coefficients, the spheroidal vector wavefunctions, and the transition matrices computed on the spheroidal basis, T-matrix simulations can be performed on the spheroidal basis.

Appendix B

Scatterer libraries

Fig. B.1 shows the response curves for all the scatterer libraries used in the main text. The low aspect-ratio Huygens library in Fig. B.1a was used for the introduction example demonstrating the limitations of the periodic approximation metasurface simulation approach (Fig. 1.2, 1.3, 1.4, 1.5, and 1.6), the comparison of the T-matrix method with locally-periodic simulations for many different NA metalenses (Fig. 3.5b), and the distributed single-layer metalens optimization (Fig. 3.10, 3.11, and 3.12). The higher aspect-ratio silicon-in-air with larger lattice separation in Fig. B.1b was used for the comparison of the T-matrix method with locally-periodic simulations for many different NA metalenses (Fig. 3.5a), the jinc source spatial-locality demonstration (Fig. 2.2), the scalability study of the low-overhead distribution method (Fig. 2.5), and the very large-area metalens simulation (Fig. 3.6). The higher aspect-ratio silicon-in-sapphire with smaller lattice separation in Fig. B.1c was used for the double-layer metalens optimization (Fig. 3.13 and 3.14). The high aspect-ratio square post library in Fig. B.1d was used for the distributed FDTD metalens simulation (Fig. 4.5 and 4.6).

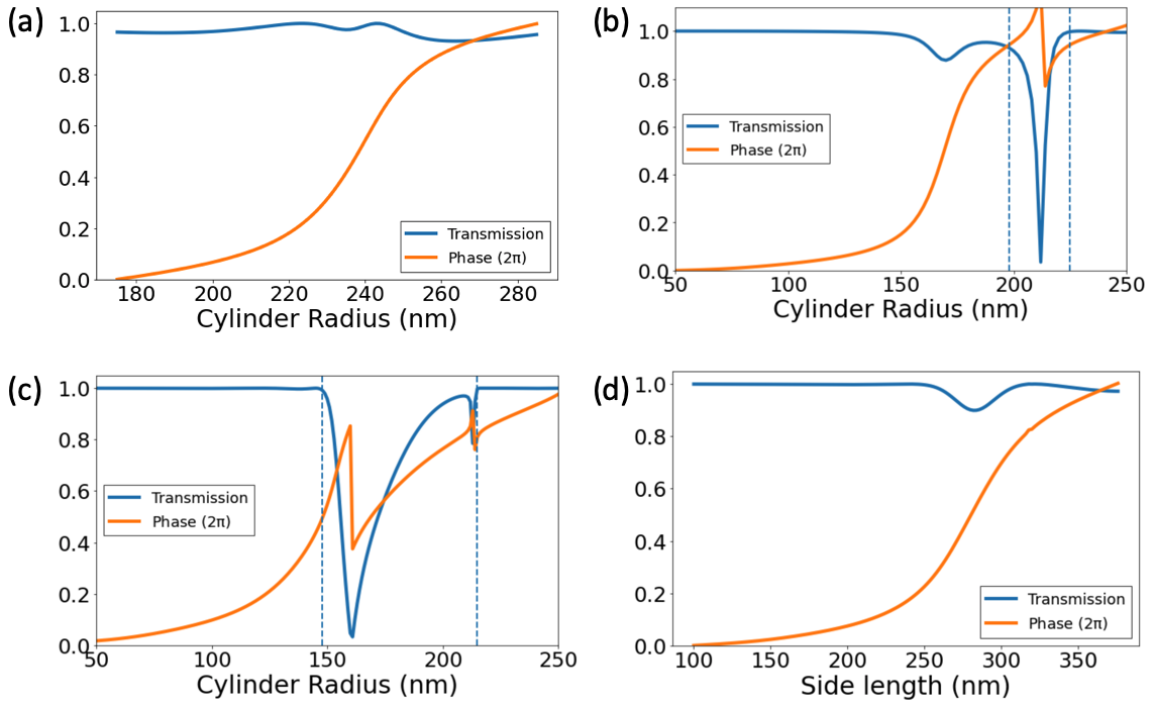


Figure B.1: **Response curves for all scatterer libraries.** (**a**) Transmission and phase response for the scatterer library based on[5], consisting of silicon cylinders with height 940 nm, radii range of 50-250 nm, square lattice period of 1070 nm, air background, and plane wave source wavelength of 1550 nm. (**b**) Transmission and phase response for the scatterer library based on [33], consisting of silicon cylinders with height 220 nm, radii range of 175-280nm, square lattice period of 666 nm, background refractive index of 1.66, and plane wave source wavelength of 1340 nm. (**c**) Transmission and phase response for a higher aspect-ratio library based on the scatterer library from [33], consisting of silicon cylinders with height 730 nm, radii range of 50-250 nm, square lattice period of 666 nm, background refractive index of 1.66, and plane wave source wavelength of 1340 nm. (**d**) Transmission and phase response for a high aspect-ratio library, consisting of square posts with refractive index 3.68, with height 1050 nm, side length range of 100-400 nm, square lattice period of 800 nm, background refractive index of 1, and plane wave source wavelength of 1550 nm.

Bibliography

[1] Francesco Aieta, Patrice Genevet, Mikhail A Kats, Nanfang Yu, Romain Blanchard, Zeno Gaburro, and Federico Capasso. Aberration-free ultrathin flat lenses and axicons at telecom wavelengths based on plasmonic metasurfaces. *Nano letters*, 12(9):4932–4936, 2012.

[2] Mikael Antelius, Kristinn B Gylfason, and Hans Sohlström. An apodized soi waveguide-to-fiber surface grating coupler for single lithography silicon photonics. *Optics express*, 19(4):3592–3598, 2011.

[3] Amir Arbabi, Ehsan Arbabi, Mahdad Mansouree, Seunghoon Han, Seyedeh Mahsa Kamali, Yu Horie, and Andrei Faraon. Increasing efficiency of high numerical aperture metasurfaces using the grating averaging technique. *Scientific reports*, 10(1):1–10, 2020.

[4] Amir Arbabi, Yu Horie, Mahmood Bagheri, and Andrei Faraon. Dielectric metasurfaces for complete control of phase and polarization with subwavelength spatial resolution and high transmission. *Nature nanotechnology*, 10(11):937–943, 2015.

[5] Amir Arbabi, Yu Horie, Alexander J Ball, Mahmood Bagheri, and Andrei

Faraon. Subwavelength-thick lenses with high numerical apertures and large efficiency based on high-contrast transmitarrays. *Nature communications*, 6(1):1–6, 2015.

[6] Ehsan Arbabi, Amir Arbabi, Seyedeh Mahsa Kamali, Yu Horie, and Andrei Faraon. Multiwavelength polarization-insensitive lenses based on dielectric metasurfaces with meta-molecules. *Optica*, 3(6):628–633, 2016.

[7] Ehsan Arbabi, Seyedeh Mahsa Kamali, Amir Arbabi, and Andrei Faraon. Full-stokes imaging polarimetry using dielectric metasurfaces. *Acs Photonics*, 5(8):3132–3140, 2018.

[8] Elyas Bayati, Raphael Pestourie, Shane Colburn, Zin Lin, Steven G Johnson, and Arka Majumdar. Inverse designed extended depth of focus meta-optics for broadband imaging in the visible. *arXiv preprint arXiv:2105.00160*, 2021.

[9] Garry Berkovic and Ehud Shafir. Optical methods for distance and displacement measurements. *Advances in Optics and Photonics*, 4(4):441–471, 2012.

[10] Steven J Byrnes, Alan Lenef, Francesco Aieta, and Federico Capasso. Designing large, high-efficiency, high-numerical-aperture, transmissive meta-lenses for visible light. *Optics express*, 24(5):5110–5124, 2016.

[11] Haogang Cai, Srilok Srinivasan, David A Czaplewski, Alex BF Martinson, David J Gosztola, Liliana Stan, Troy Loeffler, Subramanian KRS Sankaranarayanan, and Daniel López. Inverse design of metasurfaces with non-local interactions. *npj Computational Materials*, 6(1):1–8, 2020.

[12] Philip Camayd-Muñoz, Conner Ballew, Gregory Roberts, and Andrei Faraon. Multifunctional volumetric meta-optics for color and polarization image sensors. *Optica*, 7(4):280–283, 2020.

[13] Wei Ting Chen, Alexander Y Zhu, and Federico Capasso. Flat optics with dispersion-engineered metasurfaces. *Nature Reviews Materials*, 5(8):604–620, 2020.

[14] WC Chew, JL Xiong, and MA Saville. A matrix-friendly formulation of layered medium green's function. *IEEE antennas and wireless propagation letters*, 5:490–494, 2006.

[15] Rasmus E Christiansen, Zin Lin, Charles Roques-Carmes, Yannick Salamin, Steven E Kooi, John D Joannopoulos, Marin Soljačić, and Steven G Johnson. Fullwave maxwell inverse design of axisymmetric, tunable, and multi-scale multi-wavelength metalenses. *Optics Express*, 28(23):33854–33868, 2020.

[16] Haejun Chung and Owen D Miller. High-na achromatic metalenses by inverse design. *Optics express*, 28(5):6945–6965, 2020.

[17] Haejun Chung and Owen D Miller. Tunable metasurface inverse design for 80% switching efficiencies and 144 angular deflection. *ACS Photonics*, 7(8):2236–2243, 2020.

[18] Shane Colburn, Alan Zhan, and Arka Majumdar. Metasurface optics for full-color computational imaging. *Science advances*, 4(2):eaar2114, 2018.

[19] MFR Cooray and IR Ciric. Scattering of electromagnetic waves by a coated dielectric spheroid. *Journal of electromagnetic waves and applications*, 6(7):1491–1507, 1992.

[20] MFR Cooray, IR Ciric, and BP Sinha. Electromagnetic scattering by a system of two parallel dielectric prolate spheroids. *Canadian journal of physics*, 68(4-5):376–384, 1990.

[21] Jeannine Dalmas and Roger Deleuil. Translational addition theorems for prolate spheroidal vector wave functions ??ˆ{??} and ??ˆ{??}. *Quarterly of applied mathematics*, 44(2):213–222, 1986.

[22] Jeannine Dalmas, Roger Deleuil, and RH MacPhie. Rotational-translational addition theorems for spheroidal vector wave functions. *Quarterly of applied mathematics*, 47(2):351–364, 1989.

[23] Robert C Devlin, Mohammadreza Khorasaninejad, Wei Ting Chen, Jaewon Oh, and Federico Capasso. Broadband high-efficiency dielectric metasurfaces for the visible spectrum. *Proceedings of the National Academy of Sciences*, 113(38):10473–10478, 2016.

[24] Edmund JF Dickinson, Henrik Ekström, and Ed Fontes. Comsol multiphysics®: Finite element software for electrochemical analysis. a mini-review. *Electrochemistry communications*, 40:71–74, 2014.

[25] Tingting Ding, Yuanlin Zheng, and Xianfeng Chen. Integration of cascaded electro-optic and nonlinear processes on a lithium niobate on insulator chip. *Optics letters*, 44(6):1524–1527, 2019.

[26] Adrian Doicu, Thomas Wriedt, and Yuri A Eremin. *Light scattering by systems of particles: null-field method with discrete sources: theory and programs*, volume 124. Springer, 2006.

[27] Constantin Dory, Dries Vercruysse, Ki Youl Yang, Neil V Sapra, Alison E Rugar, Shuo Sun, Daniil M Lukin, Alexander Y Piggott, Jingyuan L Zhang, Marina Radulaski, et al. Inverse-designed diamond photonics. *Nature communications*, 10(1):1–7, 2019.

[28] Adam Dziekonski, Piotr Sypek, Adam Lamecki, and Michal Mrozowski. Communication and load balancing optimization for finite element electromagnetic simulations using multi-gpu workstation. *IEEE Transactions on Microwave Theory and Techniques*, 65(8):2661–2671, 2017.

[29] Amos Egel, Yuri Eremin, Thomas Wriedt, Dominik Theobald, Uli Lemmer, and Guillaume Gomard. Extending the applicability of the t-matrix method to light scattering by flat particles on a substrate via truncation of sommerfeld integrals. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 202:279–285, 2017.

[30] Einstom Engay, Dewang Huo, Radu Malureanu, Ada-Ioana Bunea, and Andrei Lavrinenko. Polarization-dependent all-dielectric metasurface for single-shot quantitative phase imaging. *Nano Letters*, 21(9):3820–3826, 2021.

[31] Zhi-Bin Fan, Zeng-Kai Shao, Ming-Yuan Xie, Xiao-Ning Pang, Wen-Sheng Ruan, Fu-Li Zhao, Yu-Jie Chen, Si-Yuan Yu, and Jian-Wen Dong. Silicon nitride metalenses for close-to-one numerical aperture and wide-angle visible imaging. *Physical Review Applied*, 10(1):014005, 2018.

[32] C Flammer and Spheroidal Wave Functions. Stanford univ. press, 1957.

[33] Carlo Gigli, Qitong Li, Pierre Chavel, Giuseppe Leo, Mark Brongersma, and Philippe Lalanne. Fundamental limitations of huygens' metasurfaces for optical beam shaping. *arXiv e-prints*, pages arXiv–2011, 2020.

[34] Joseph W Goodman. Introduction to fourier optics, roberts & co. *Publishers, Englewood, Colorado*, 2005.

[35] Roger H Hackman. The transition matrix for acoustic and elastic wave scattering in prolate spheroidal coordinates. *The Journal of the Acoustical Society of America*, 75(1):35–45, 1984.

[36] Roger H Hackman and Douglas G Todoroff. An application of the spheroidal-coordinate-based transition matrix: The acoustic scattering from high aspect ratio solids. *The Journal of the Acoustical Society of America*, 78(3):1058–1071, 1985.

[37] Everton Hermann, Bruno Raffin, François Faure, Thierry Gautier, and Jérémie Allard. Multi-gpu and multi-cpu parallelization for interactive physics simulations. In *European Conference on Parallel Processing*, pages 235–246. Springer, 2010.

[38] Aaron L Holsteen, Ahmet Fatih Cihan, and Mark L Brongersma. Temporal color mixing and dynamic beam shaping with silicon metasurfaces. *Science*, 365(6450):257–260, 2019.

[39] Yu Horie, Amir Arbabi, Ehsan Arbabi, Seyedeh Mahsa Kamali, and Andrei Faraon. Wide bandwidth and high resolution planar filter array based on dbr-metasurface-dbr structures. *Optics express*, 24(11):11677–11682, 2016.

[40] Tyler W Hughes, Momchil Minkov, Victor Liu, Zongfu Yu, and Shanhui Fan. Full wave simulation and optimization of large area metalens. *OSA Optical Design and Fabrication Congress 2021*, 2021.

[41] Tyler W Hughes, Momchil Minkov, Ian AD Williamson, and Shanhui Fan. Adjoint method and inverse design for nonlinear nanophotonic devices. *ACS Photonics*, 5(12):4781–4787, 2018.

[42] Jiaqi Jiang and Jonathan A Fan. Global optimization of dielectric metasurfaces using a physics-driven neural network. *Nano letters*, 19(8):5366–5372, 2019.

[43] Jiaqi Jiang, David Sell, Stephan Hoyer, Jason Hickey, Jianji Yang, and Jonathan A Fan. Free-form diffractive metagrating design based on generative adversarial networks. *ACS nano*, 13(8):8872–8878, 2019.

[44] Won-Jae Joo, Jisoo Kyoung, Majid Esfandyarpour, Sung-Hoon Lee, Hyun Koo, Sunjin Song, Young-Nam Kwon, Seok Ho Song, Jun Cheol Bae, Ara Jo, et al. Metasurface-driven oled displays beyond 10,000 pixels per inch. *Science*, 370(6515):459–463, 2020.

[45] Mohammadreza Khorasaninejad, Alexander Yutong Zhu, Charles Roques-Carmes, Wei Ting Chen, Jaewon Oh, Ishan Mishra, Robert C Devlin, and Federico Capasso. Polarization-insensitive metalenses at visible wavelengths. *Nano letters*, 16(11):7229–7234, 2016.

[46] Hyounghan Kwon, Ehsan Arbabi, Seyedeh Mahsa Kamali, MohammadSadegh Faraji-Dana, and Andrei Faraon. Single-shot quantitative phase gradient microscopy using a system of multifunctional metasurfaces. *Nature Photonics*, 14(2):109–114, 2020.

[47] Christopher M Lalau-Keraly, Samarth Bhargava, Owen D Miller, and Eli Yablonovitch. Adjoint shape optimization applied to electromagnetic design. *Optics express*, 21(18):21693–21701, 2013.

[48] HJ Landau. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706, 1967.

[49] Gun-Yeal Lee, Jong-Young Hong, SoonHyoung Hwang, Seokil Moon, Hyeokjung Kang, Sohee Jeon, Hwi Kim, Jun-Ho Jeong, and Byoungho Lee. Metasurface eyepiece for augmented reality. *Nature communications*, 9(1):1–10, 2018.

[50] Yun-Han Lee, Tao Zhan, and Shin-Tson Wu. Prospects and challenges in augmented reality displays. *Virtual Real. Intell. Hardw.*, 1(1):10–20, 2019.

[51] Shi-Qiang Li, Xuewu Xu, Rasna Maruthiyodan Veetil, Vytautas Valuckas, Ramón Paniagua-Domínguez, and Arseniy I Kuznetsov. Phase-only transmissive spatial light modulator based on tunable dielectric metasurface. *Science*, 364(6445):1087–1090, 2019.

[52] Zhaoyi Li, Raphaël Pestourie, Joon-Suh Park, Yao-Wei Huang, Steven G Johnson, and Federico Capasso. Inverse design enables large-scale high-performance meta-optics reshaping virtual reality. *arXiv preprint arXiv:2104.09702*, 2021.

[53] Haowen Liang, Qiaoling Lin, Xiangsheng Xie, Qian Sun, Yin Wang, Lidan Zhou, Lin Liu, Xiangyang Yu, Jianying Zhou, Thomas F Krauss, et al. Ultrahigh numerical aperture metalens at visible wavelengths. *Nano letters*, 18(7):4460–4466, 2018.

[54] Zin Lin and Steven G Johnson. Overlapping domains for topology optimization of large-area metasurfaces. *Optics express*, 27(22):32445–32453, 2019.

[55] Zin Lin, Victor Liu, Raphaël Pestourie, and Steven G Johnson. Topology optimization of freeform large-area metasurfaces. *Optics express*, 27(11):15765–15775, 2019.

[56] Zin Lin, Charles Roques-Carmes, Rasmus E Christiansen, Marin Soljačić, and Steven G Johnson. Computational inverse design for ultra-compact single-piece

metalenses free of chromatic and angular aberration. *Applied Physics Letters*, 118(4):041104, 2021.

[57] Zeqin Lu, Han Yun, Yun Wang, Zhitian Chen, Fan Zhang, Nicolas AF Jaeger, and Lukas Chrostowski. Broadband silicon photonic directional coupler using asymmetric-waveguide based phase control. *Optics express*, 23(3):3795–3808, 2015.

[58] James H Luscombe and Marshall Luban. Simplified recursive algorithm for wigner 3 j and 6 j symbols. *Physical Review E*, 57(6):7274, 1998.

[59] Mahdad Mansouree, Hyounghan Kwon, Ehsan Arbabi, Andrew McClung, Andrei Faraon, and Amir Arbabi. Multifunctional 2.5 d metastructures enabled by adjoint optimization. *Optica*, 7(1):77–84, 2020.

[60] Johannes Markkanen and Alex J Yuffa. Fast superposition t-matrix solution for clusters with arbitrarily-shaped constituent particles. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 189:181–188, 2017.

[61] Andrew McClung, Mahdad Mansouree, and Amir Arbabi. At-will chromatic dispersion by prescribing light trajectories with cascaded metasurfaces. *Light: Science & Applications*, 9(1):1–9, 2020.

[62] David AB Miller. Fundamental limit to linear one-dimensional slow light structures. *Physical review letters*, 99(20):203903, 2007.

[63] Sean Molesky, Alexander Y Piggott, Jin Weiliang, Jelena Vučković, and Alejandro W Rodriguez. Inverse design in nanophotonics. *Nature Photonics*, 9(11):659, 2018.

[64] CUDA Nvidia. Cublas library. *NVIDIA Corporation, Santa Clara, California*, 15(27):31, 2008.

[65] Adam J Ollanik, Jake A Smith, Mason J Belue, and Matthew D Escarra. High-efficiency all-dielectric huygens metasurfaces from the ultraviolet to the infrared. *ACS photonics*, 5(4):1351–1358, 2018.

[66] Ardavan F Oskooi, David Roundy, Mihai Ibanescu, Peter Bermel, John D Joannopoulos, and Steven G Johnson. Meep: A flexible free-software package for electromagnetic simulations by the fdtd method. *Computer Physics Communications*, 181(3):687–702, 2010.

[67] Ramon Paniagua-Dominguez, Ye Feng Yu, Egor Khaidarov, Sumin Choi, Victor Leong, Reuben M Bakker, Xinan Liang, Yuan Hsing Fu, Vytautas Valuckas, Leonid A Krivitsky, et al. A metalens with a near-unity numerical aperture. *Nano letters*, 18(3):2124–2132, 2018.

[68] Junghyun Park, Byung Gil Jeong, Sun Il Kim, Duhyun Lee, Jungwoo Kim, Changgyun Shin, Chang Bum Lee, Tatsuhiro Otsuka, Jisoo Kyoung, Sangwook Kim, et al. All-solid-state spatial light modulator with independent phase and amplitude control for three-dimensional lidar applications. *Nature nanotechnology*, 16(1):69–76, 2021.

[69] Raphaël Pestourie, Youssef Mroueh, Thanh V Nguyen, Payel Das, and Steven G Johnson. Active learning of deep surrogates for pdes: Application to metasurface design. *npj Computational Materials*, 6(1):1–7, 2020.

[70] Raphaël Pestourie, Carlos Pérez-Arancibia, Zin Lin, Wonseok Shin, Federico Capasso, and Steven G Johnson. Inverse design of large-area metasurfaces. *Optics express*, 26(26):33732–33747, 2018.

[71] Thaibao Phan, David Sell, Evan W Wang, Sage Doshay, Kofi Edee, Jianji Yang, and Jonathan A Fan. High-efficiency, large-area, topology-optimized metasurfaces. *Light: Science & Applications*, 8(1):1–9, 2019.

[72] Alexander Y Piggott, Jesse Lu, Konstantinos G Lagoudakis, Jan Petykiewicz, Thomas M Babinec, and Jelena Vučković. Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer. *Nature Photonics*, 9(6):374–377, 2015.

[73] Alexander Y Piggott, Jan Petykiewicz, Logan Su, and Jelena Vučković. Fabrication-constrained nanophotonic inverse design. *Scientific reports*, 7(1):1–7, 2017.

[74] Federico Presutti and Francesco Monticone. Focusing on bandwidth: achromatic metalens limits. *Optica*, 7(6):624–631, 2020.

[75] Junuthula Narasimha Reddy. *Introduction to the finite element method*. McGraw-Hill Education, 2019.

[76] Raymond C Rumpf. Simple implementation of arbitrarily shaped total-field/scattered-field regions in finite-difference frequency-domain. *Progress In Electromagnetics Research*, 36:221–248, 2012.

[77] Neil V Sapra, Ki Youl Yang, Dries Vercruysse, Kenneth J Leedle, Dylan S Black, R Joel England, Logan Su, Rahul Trivedi, Yu Miao, Olav Solgaard, et al. On-chip integrated laser-driven particle accelerator. *Science*, 367(6473):79–83, 2020.

[78] David Sell, Jianji Yang, Sage Doshay, and Jonathan A Fan. Periodic dielectric metasurfaces with high-efficiency, multiwavelength functionalities. *Advanced Optical Materials*, 5(23):1700645, 2017.

[79] David Sell, Jianji Yang, Sage Doshay, Rui Yang, and Jonathan A Fan. Large-angle, multifunctional metagratings based on freeform multimode geometries. *Nano letters*, 17(6):3752–3757, 2017.

[80] Qingwu Shi, Bin Zou, Lamei Zhang, and Desheng Liu. Hybrid parallel fdtd calculation method based on mpi for electrically large objects. *Wireless Communications and Mobile Computing*, 2019, 2019.

[81] Zhujun Shi, Mohammadreza Khorasaninejad, Yao-Wei Huang, Charles Roques-Carmes, Alexander Y Zhu, Wei Ting Chen, Vyshakh Sanjeev, Zhao-Wei Ding, Michele Tamagnone, Kundan Chaudhary, et al. Single-layer metasurface with controllable multiwavelength functions. *Nano letters*, 18(4):2420–2427, 2018.

[82] B Sinha and R MacPhie. Electromagnetic plane wave scattering by a system of two parallel conducting prolate spheroids. *IEEE Transactions on Antennas and Propagation*, 31(2):294–304, 1983.

[83] Bateshwar P Sinha and Robert H MacPhie. Electromagnetic scattering by prolate spheroids for plane waves with arbitrary polarization and angle of incidence. *Radio Science*, 12(2):171–184, 1977.

[84] FDTD Solutions. Lumerical solutions. *Inc., http://www. lumerical. com*, 2003.

[85] Logan Su, Dries Vercruysse, Jinhie Skarda, Neil V Sapra, Jan A Petykiewicz, and Jelena Vučković. Nanophotonic inverse design with spins: Software architecture and practical considerations. *Applied Physics Reviews*, 7(1):011407, 2020.

[86] Logan Wang Su. *Computational nanophotonic design: frameworks and applications*. Stanford University, 2020.

[87] Allen Taflove and Susan C Hagness. *Computational electrodynamics*, volume 28. Artech house publishers Norwood, MA, 2000.

[88] Jingzhe Tang, Yanfeng Zheng, Chao Yang, Wei Wang, and Yaozhi Luo. Parallelized implementation of the finite particle method for explicit dynamics in gpu. *Computer Modeling in Engineering & Sciences*, 122(1):5–31, 2020.

[89] Mahsa Torfeh and Amir Arbabi. Modeling metasurfaces using discrete-space impulse response technique. *ACS Photonics*, 7(4):941–950, 2020.

[90] Fengwen Wang, Jakob S Jensen, and Ole Sigmund. Robust topology optimization of photonic crystal waveguides with tailored dispersion properties. *JOSA B*, 28(3):387–397, 2011.

[91] Qianfan Xu, Bradley Schmidt, Sameer Pradhan, and Michal Lipson. Micrometre-scale silicon electro-optic modulator. *nature*, 435(7040):325–327, 2005.

[92] Ki Youl Yang, Jinhie Skarda, Michele Cotrufo, Avik Dutt, Geun Ho Ahn, Mahmoud Sawaby, Dries Vercruysse, Amin Arbabian, Shanhui Fan, Andrea Alù, et al. Inverse-designed non-reciprocal pulse router for chip-based lidar. *Nature Photonics*, 14(6):369–374, 2020.

[93] Alan Zhan, Shane Colburn, Rahul Trivedi, Taylor K Fryett, Christopher M Dodson, and Arka Majumdar. Low-contrast dielectric metasurface optics. *ACS Photonics*, 3(2):209–214, 2016.

[94] Alan Zhan, Taylor K Fryett, Shane Colburn, and Arka Majumdar. Inverse design of optical elements based on arrays of dielectric spheres. *Applied optics*, 57(6):1437–1446, 2018.

[95] Alan Zhan, Ricky Gibson, James Whitehead, Evan Smith, Joshua R Hendrickson, and Arka Majumdar. Controlling three-dimensional optical fields via inverse mie scattering. *Science advances*, 5(10):eaax4769, 2019.

[96] Maksym V Zhelyeznyakov, Steve Brunton, and Arka Majumdar. Deep learning to accelerate scatterer-to-field mapping for inverse design of dielectric metasurfaces. *ACS Photonics*, 8(2):481–488, 2021.

[97] Maksym V Zhelyeznyakov, Alan Zhan, and Arka Majumdar. Design and optimization of ellipsoid scatterer-based metasurfaces via the inverse t-matrix method. *OSA Continuum*, 3(1):89–103, 2020.

[98] Junxiao Zhou, Haoliang Qian, Ching-Fu Chen, Junxiang Zhao, Guangru Li, Qianyi Wu, Hailu Luo, Shuangchun Wen, and Zhaowei Liu. Optical edge detection based on high-efficiency dielectric metasurface. *Proceedings of the National Academy of Sciences*, 116(23):11137–11140, 2019.

[99] Ming Zhou, Dianjing Liu, Samuel W Belling, Haotian Cheng, Mikhail A Kats, Shanhui Fan, Michelle L Povinelli, and Zongfu Yu. Inverse design of metasurfaces based on coupled-mode theory and adjoint optimization. *ACS Photonics*, 8(8):2265–2273, 2021.

[100] You Zhou, Hanyu Zheng, Ivan I Kravchenko, and Jason Valentine. Flat optics for image differentiation. *Nature Photonics*, 14(5):316–323, 2020.