POLITECNICO DI TORINO

Master's Degree in Communications and Computer Networks Engineering

Master's Thesis

Advanced Numerical Strategies for the Inverse Design of Metasurface Antennas

In collaboration with **Link's Foundation**

Supervisors
Prof. Giuseppe Vecchi
Dr. Marcello Zucchi
Dr. Lucia Teodorani

Candidate
Shaghayegh Samadzadehghezelghayeh

June 2025

Abstract

Metasurface antennas have emerged as a promising alternative to conventional reflectors and phased arrays for high-gain, low-profile, beam-steerable applications in both terrestrial and space-borne platforms. By sculpting electromagnetic wavefronts with sub-wavelength surface patterns, they deliver unprecedented control over radiation characteristics while minimizing volume, weight, and cost. In this thesis, we develop a fully automated, physics-informed inverse-design framework capable of synthesizing three-dimensional, fabrication-ready metasurface antenna architectures that meet stringent performance and realizability constraints.

Building upon a surface-integral-equation formulation, our methodology directly optimizes the equivalent surface current, which allows us to impose arbitrary far-field masks (e.g., beam-shaping and sidelobe suppression) while strictly enforcing passivity, losslessness, and other physical realizability conditions on the synthesized surface impedance. To tame the multiscale complexity and ill-conditioning inherent in large metasurface designs, we introduce three complementary innovations: a hierarchical multiresolution RWG basis that compresses the system's degrees of freedom in early iterations without degrading final accuracy; unit-flux diagonal preconditioning and Hessian-based diagonal gradient scaling to equalize variable sensitivities and dramatically speed convergence; and the use of modern adaptive optimizers (Adam and AdaHessian) drawn from the machine-learning community, which further enhance robustness and iteration efficiency in highly nonconvex, high-dimensional design landscapes.

The effectiveness of these strategies varies across different metasurface structures, each requiring delicate tuning of involved parameters to balance accuracy, convergence, and realizability. Among all, Unit-Flux Diagonal Preconditioning consistently demonstrates stable results and a marked enhancement in convergence, making it a particularly reliable component of the proposed framework.

The practical effectiveness of adaptive optimizers such as Adam and AdaHessian, however, depends strongly on careful hyperparameter tuning. Step sizes, and decay rates must be balanced with respect to the physical characteristics of the design space, and poorly chosen values can lead to stagnation or large fluctuations in convergence. This sensitivity motivates the exploration of hybrid optimization strategies combining the fast early progress of adaptive methods with the stability of classical techniques like conjugate gradient which is proposed as a promising direction for future work.

Contents

List of Tables 4

List of Figures 5

1 Introduction 7
- 1.1 State of the Art . . . 8
- 1.2 Aim of the Thesis . . . 11
- 1.3 Thesis Outline . . . 11

2 Background and Literature Review 13
- 2.1 Metasurface Antenna Fundamentals . . . 14
  - 2.1.1 IBC &Surface Homogenization . . . 14
  - 2.1.2 Surface Integral Equation (SIE) Formulation . . . 16
  - 2.1.3 Discretization via Method of Moments . . . 18
- 2.2 Automated Inverse-Design Technique . . . 21
  - 2.2.1 Geometry . . . 21
  - 2.2.2 Constraint Definition . . . 23
- 2.3 Current-based Optimization Algorithm . . . 26
  - 2.3.1 Algorithm Formulation . . . 26
  - 2.3.2 Constraints Enforcement . . . 30

3 Multiscale Basis and Preconditioning for efficient MTS Inverse Design 33
- 3.1 Multi-resolution RWG Basis . . . 34
  - 3.1.1 Construction of MR-RWG Basis . . . 37
  - 3.1.2 Multilevel Optimization in the MR–RWG Basis . . . 40
  - 3.1.3 Optimization in MR–RWG Basis . . . 41
- 3.2 Preconditioning Strategies . . . 42
  - 3.2.1 Mesh-Level (Unit Edge-Flux) Preconditioning . . . 43
  - 3.2.2 Gradient Preconditioning via Hessian Diagonal . . . 45

4 Advanced Optimization Strategies Tailored to MTS Design 48
- 4.1 Problem Formulation and Challenges . . . 49
- 4.2 First-Order: Adam Optimizer . . . 51
  - 4.2.1 Algorithm Overview and Update Rules . . . 52

4.2.2 Adam : Choice of Hyperparameters . . . 53
4.2.3 Convergence Behavior and Step-Size Control . . . 54
4.3 Second-Order: Ada-Hessian . . . 55
4.3.1 Algorithm Overview and Update Rules . . . 56
4.3.2 AdaHessian : Choice of Hyperparameters . . . 58
4.3.3 Hyperparameter Tuning . . . 59

5 Results and Analysis 63
5.1 Simulation Setup . . . 64
5.2 Mesh and Geometry Definition . . . 65
5.2.1 Multi-Level Optimization in MR-RWG Basis . . . 66
5.2.2 Effect of Unit Edge-Flux Preconditioning on Convergence . . . 68
5.2.3 MR-RWG Basis Change . . . 70
5.3 Second Order Information . . . 71
5.3.1 Hutchinson's Based Diagonal Estimation . . . 71
5.4 New Optimizers . . . 72
5.4.1 Adam Optimizer . . . 73
5.4.2 AdaHessian . . . 73
5.5 Results of the Impedance Realization for DP-NLCG . . . 74

6 Conclusion and Future Works 76

Appendix 78
I Antenna Generalities . . . 78
I.1 Field Polarization . . . 78
I.2 Directivity, Gain, and Efficiency . . . 79
II RWG Basis Definition . . . 80
III Surface Equivalence Theorem . . . 81
IV Definition of the Complex Gradient . . . 83
V Power Objective, Gradient, and Hessian . . . 84
VI Far-Field Objective, Gradient, and Hessian . . . 87

List of Tables

2.1 Equivalent surface currents 17
2.2 Key design constraints for inverse metasurface synthesis. 25
4.3 Adam Hyperparameters 53
4.1 Summary of radiation penalties applied in the power objective $f_{\text{rlz}}(\mathbf{I})$, each promoting physical consistency or bounding key quantities. 60
4.2 Summary of radiation penalties applied in the far-field objective $f_{\text{rad}}(\mathbf{I})$, each enforcing directional gain requirements. 61
4.4 AdaHessian Hyperparameters 62
5.1 Far-Field Mask Specifications 64
5.2 Angular Mask Regions for Pencil-Beam Design 65
5.3 Geometric parameters of small-scale, and separated square metasurface antenna 66
5.4 Geometric parameters of mesh levels 67
5.5 Comparison of optimization results NLCG, DP-NLCG 74
1 List of unique gradient and mixed Hessian terms appearing in $H_{I,I^*}[f_{\text{rad}}]$. . 90

List of Figures

1.1 Metasurface-antenna technologies: (a) square metallic patches with size modulation, (b) circular patches with radius modulation, (c) metal or dielectric pillars with height modulation, (d) perforated slots with radius modulation. . . . 9
1.2 Multiscale structure of a 20λ-diameter metasurface antenna: (a) overall aperture layout (macroscale), (b) periodic lattice arrangement (mesoscale), and (c) individual scattering elements (microscale). . . . 9
1.3 (a) Actual metasurface and (b) homogenized metasurface with the surface impedance distribution. . . . 10
2.1 Example of geometry for a metasurface antenna layout: (a) 3D view showing the configuration with an on-surface source for the incident field; (b) cross-sectional view illustrating the equivalent surface current **J**. . . . 22
2.2 Example of mesh employed in the design. Only the impedance surface is discretized, and the effect of the grounded substrate (background medium) is taken into account by the multilayer Green's function. . . . 23
3.1 Example triangular mesh for a metallic surface. RWG basis functions are supported on each pair of adjacent triangles. . . . 36
3.2 Illustration of an RWG basis function on two adjacent triangles. Arrows indicate the RWG basis with direction of surface current. (a) Coarse mesh $j = 0$ (b) Refined mesh $j = 1$ . . . 36
3.3 Detail functions at level $j = 1$. (a) Solenoidal loop function; (b)–(g) non-solenoidal edge (RWG) functions[24]. . . . 39
5.1 Masks for radiation pattern . . . 64
5.2 Small Scale mesh . . . 65
5.3 Separated Square Mesh . . . 65
5.4 Coarse Mesh . . . 66
5.5 Mid-Level Mesh . . . 66
5.6 Fine Mesh . . . 66
5.7 Convergence Comparison : NLCG, ML-NLCG . . . 67
5.8 Current Optimized: NLCG . . . 68
5.9 Current Optimized: ML-NLCG . . . 68
5.11 Current Optimized: NLCG . . . 68
5.12 Directivity: NLCG . . . 68
5.10 Convergance Comparison : NLCG, DP-NLCG . . . 69

5.13 Current Optimized: DP-NLCG 69
5.14 Directivity: DP-NLCG 69
5.15 Convergence Comparison 70
5.16 Current Optimized: MR-NLCG 70
5.17 Directivity: MR-NLCG 70
5.18 Diagonal Estimation 72
5.19 Convergence Comparison : NLCG vs Hutch-NLCG 72
5.20 Convergence Comparison : NLCG vs Adam 73
5.21 Convergence Comparison : NLCG vs Adam 74
5.22 Realization of Impedance for DP-NLCG 75
1 Graphical representation of the RWG basis function. 80
2 Surface equivalence theorem illustration: (a) actual and (b) equivalent problem. 82

Chapter 1

Introduction

1.1 State of the Art

Emerging platforms such as automotive radar, low-Earth-orbit satellite constellations, ground terminals for high-throughput satellites, massive-MIMO 5G/6G base stations, and radio-astronomy arrays all demand antennas that deliver high gain and beam shaping within ever-tighter volume, weight, and power budgets. These requirements are especially acute at millimeter-wave frequencies, where free-space attenuation is severe and each additional decibel of aperture efficiency translates directly into longer link range or smaller power amplifiers [23].

Classical high-directive apertures—parabolic reflectors and printed/slotted phased arrays—each meet the gain requirement but introduce prohibitive trade-offs. Reflectors achieve narrow beams by scaling their physical aperture. However, exceeding the gain of 25 dBi[1] at centimeter wavelengths entails dishes tens of centimeters across, with bulky supports and mechanical gimbals for scanning, driving up weight, cost, and maintenance [6]. Conversely, phased arrays collapse the profile to a few millimeters and enable electronic beam steering, yet each element requires a phase shifter, transmitter/receiver module, and bias network. As element counts rise into the hundreds or thousands, the RF-front-end bill of materials, ohmic losses, and DC power draw escalate, often dominating the link budget at Ka-band and above [17, 20]. Together, these limitations leave a design gap for apertures that are flat and low-cost like printed circuits yet capable of shaping and steering beams without a forest of active components.

In recent years, metasurface antennas have emerged as a compelling high-gain alternative that sits between bulky reflectors and complex phased arrays. Rather than relying on a curved dish or thick lens, a metasurface antenna uses a single, ultrathin sheet patterned with sub-wavelength metallic or dielectric elements. By carefully tuning each inclusion's geometry and arrangement, the flat panel collectively redirects and focuses incoming or outgoing waves exactly as a curved reflector would—yet without any physical curvature or added depth. This "flat form factor" not only reduces the antenna's overall thickness and weight, but also simplifies fabrication and integration into modern devices, enabling high-performance beam shaping in a low-profile, easily manufactured package. Indeed, metasurface-based reflectarray antennas are less bulky than parabolic dishes and can be fabricated with standard printed-circuit techniques, leading to lower production costs [3]. Like phased arrays, they can achieve precise beam shaping and even beam steering, but without needing discrete phase shifters at each element [21]. As a result, reflectarray-type metasurface antennas can provide high gain, low loss, and low cross-polarization levels, while remaining lightweight and conformal [21]. For example, planar metasurfaces have demonstrated aperture efficiencies up to $\sim$ 70% and bandwidths around 30% in experimental prototypes [8], approaching the performance of conventional antennas but in a much flatter package. Moreover, metasurface antennas offer design flexibility: by tailoring the pattern of sub-wavelength elements, one can convert an incoming wave into

[1] At centimeter wavelengths (e.g., 10 GHz), a 25 dBi gain corresponds to an aperture diameter of tens of centimeters (approximately 30 cm), beyond which size, weight, and mechanical complexity become impractical.

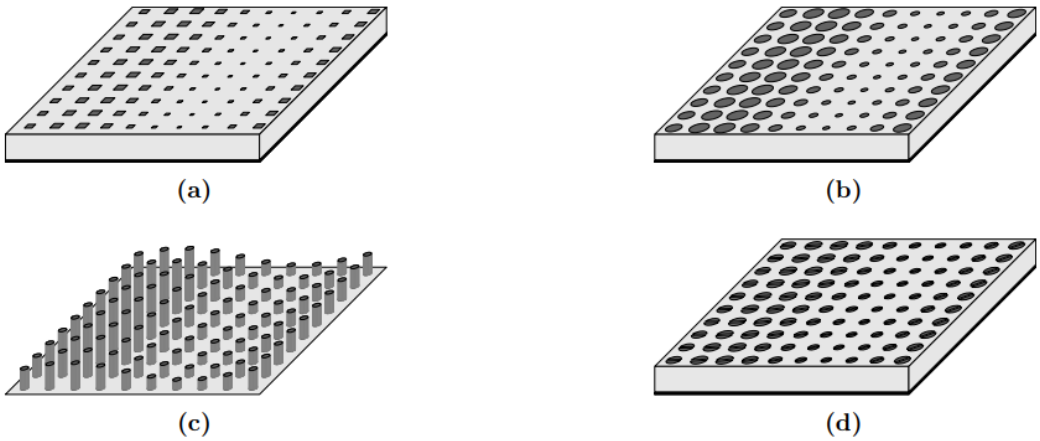

Figure 1.1. Metasurface-antenna technologies: (a) square metallic patches with size modulation, (b) circular patches with radius modulation, (c) metal or dielectric pillars with height modulation, (d) perforated slots with radius modulation.

almost any desired radiation pattern.

Feeding schemes for metasurface antennas fall into two main classes. In the *reflectarray* configuration, an external feed (horn or patch) illuminates the metasurface, which reflects energy into a collimated or shaped beam. By contrast, *leaky-wave* or *holographic* metasurfaces launch a guided surface wave via an embedded feed; sub-wavelength elements then continuously couple energy into free space, producing a directive radiation pattern without any out-of-plane feed [15]. Both approaches exploit the metasurface's ability to locally tailor the boundary conditions and thereby sculpt the far-field pattern.

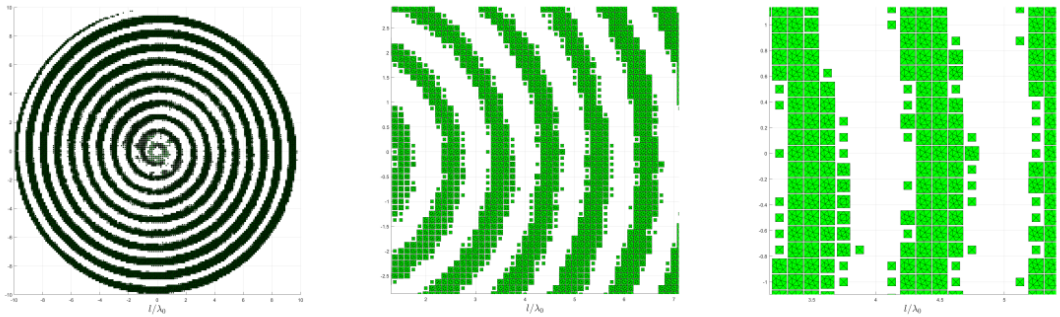

Figure 1.2. Multiscale structure of a 20λ-diameter metasurface antenna: (a) overall aperture layout (macroscale), (b) periodic lattice arrangement (mesoscale), and (c) individual scattering elements (microscale).

Designing metasurface antennas is a challenging multi-scale problem. On the one hand, the metasurface as a whole may span large wavelengths in size to achieve a high directivity (for example, an aperture tens of wavelengths across is needed for a narrow beamwidth). On the other hand, the constituent scatterers are sub-wavelength in size and spacing to ensure a nearly continuous electromagnetic response is challenging. A brute-force electromagnetic simulation of the entire antenna, resolving every tiny element over a large aperture, would involve an enormous number of unknowns and is generally impractical. To make the problem tractable, homogenized surface models are used. In particular, a metasurface can be modeled macroscopically by an equivalent surface impedance or impedance boundary condition (IBC) that relates the tangential electric and magnetic fields on the surface. Instead of simulating each element in detail, one treats the metasurface as a continuous sheet characterized by a position-dependent impedance $Z_s(x, y)$. This impedance represents the aggregate effect of the sub-wavelength elements on the currents and fields, analogous to a "skin" covering the surface. The IBC model dramatically reduces the computational complexity: the antenna can be analyzed with surface integral equations or other methods using the unknown continuous impedance or the equivalent surface current, rather than millions of element-level unknowns.

Figure 1.3. (a) Actual metasurface and (b) homogenized metasurface with the surface impedance distribution.

This homogenization approach is valid when the metasurface elements are electrically small and spaced sufficiently tightly to behave like a quasi-continuous sheet [11]. An important constraint is that the synthesized impedance profile must be physically realizable – in practice it should correspond to passive, lossless elements with reactance values within achievable ranges. In other words, the macroscopic design must ultimately map to actual meta-atom geometries (e.g. capacitor-loaded patches, variable-length resonators, etc.) that can be fabricated. Ensuring the impedance stays within these bounds (no extreme values that would require unrealizable materials) is part of the design challenge.

With homogenization in hand, modern metasurface design proceeds in two phases (see Figure 1.2). Phase 1 is the *macroscopic synthesis* of $Z_s(x, y)$ (or equivalently the continuous surface current) to satisfy far-field objectives—mask-type gain constraints or prescribed patterns—formulated as an inverse electromagnetic problem. Phase 2 is the *unit-cell realization*, where each local impedance value is implemented by tailoring a meta-atom's geometry or loading under a local-periodicity assumption [8]. This separation allows one to leverage high-level optimization for pattern synthesis, then apply standard full-wave or

circuit-model methods to translate impedance into geometry.

Although early metasurface designs relied on closed-form wave-physics or holographic rules, these analytical methods struggle with arbitrary field masks, broadband requirements, and fabrication limits. They typically assume idealized illumination and neglect inter-element coupling and edge effects, making them unsuitable for complex coverage specifications or conformal surfaces. In contrast, *numerical inverse-design* methods cast metasurface synthesis as an optimization problem—minimizing pattern error or enforcing mask constraints—using techniques [18, 27]. These advanced strategies can accommodate 2D vs 3D formulations, pattern-matching vs mask-type objectives, and multi-layer configurations, offering a powerful toolkit for crafting high-performance metasurface antennas.

1.2 Aim of the Thesis

The primary goal of this work is to enhance the performance of the fully automated, three-dimensional inverse-design framework for metasurface antennas in computational efficiency. Building on recent advances in current-based synthesis, the scattering problem is represented as a surface-integral equation in which the sole unknown is the equivalent electric current on the metasurface. By driving this current directly through an optimization routine, one can enforce arbitrary mask-type constraints on the far-field pattern—such as prescribed main-beam gain and sidelobe levels while simultaneously guaranteeing that the resulting surface impedance remains passive, lossless, and within realistic fabrication limits [27].

To accelerate convergence and manage electrically large apertures, our methodology integrates three key numerical enhancements. First, developing a hierarchical Rao–Wilton–Glisson (RWG) basis that captures both global and local current variations with fewer degrees of freedom. Second, an edge-preconditioning scheme normalizes mesh edges to achieve unit flux across the edges of the mesh, substantially improving the conditioning of the Method-of-Moments matrix and yielding stable per-iteration performance. Third, we incorporate advanced optimization algorithms—specifically adaptive first-order and and second-order solvers (e.g. ADAM and AdaHessian) to exploit both gradient and curvature information, aiming for accelerating convergence and enhancing robustness on ill-conditioned metasurface design problems.

Collectively, these contributions establish a general-purpose toolchain capable of translating high-level radiation specifications into physically realizable metasurface designs, thereby bridging the gap between theoretical synthesis and practical antenna fabrication.

1.3 Thesis Outline

Chapter 1. This chapter outlines the motivation for metasurface antennas and the limitations of conventional aperture technologies. It introduces metasurfaces as a flat, efficient alternative and frames their design as a multi-scale inverse problem.

Chapter 2. This chapter provides the theoretical and methodological foundations underpinning the metasurface antenna design process. It begins by reviewing the principles of metasurface antennas, including surface homogenization and impedance boundary conditions. The chapter then introduces the surface integral equation (SIE) formulation and its discretization using the Method of Moments. Next, it explores automated inverse-design strategies, detailing geometric modeling and various constraint definitions such as passivity, impedance bounds, and radiation pattern requirements. Finally, the chapter outlines the current-based optimization framework, including algorithm formulation and the enforcement of physical and performance constraints.

Chapter 3. This chapter presents advanced numerical methods that enhance the efficiency of metasurface antenna inverse design. It begins by introducing the MR-RWG basis, which hierarchically captures surface currents across multiple scales using nested mesh refinements and wavelet decompositions. This structure promises for faster convergence and matrix sparsification without sacrificing accuracy. The chapter then explores two complementary preconditioning strategies—unit edge-flux normalization to correct for geometric imbalances, and Hessian-based diagonal gradient scaling to adapt step sizes according to local curvature—both of which theoretically promise to accelerate iterative solvers in large-scale, ill-conditioned design problems.

Chapter 4. This chapter presents advanced optimization strategies specifically designed for large-scale, non-convex optimization problems. We begin by introducing the existing optimization method for inverse-design problem that enforces both physical realizability (passive, bounded impedance) and radiation-pattern masks—and highlight the numerical challenges posed by its nonconvex, ill-conditioned landscape. We then introduce two modern solver families first, the Adam optimizer, which combines momentum and per-parameter adaptive learning rates to navigate rugged cost surfaces without costly line searches; and second, AdaHessian, which augments Adam's variance tracking with lightweight Hessian-diagonal estimates (via Hutchinson's method) to scale steps according to local curvature. For each method we present the algorithmic updates, and explain how these strategies could overcome the stiffness and anisotropy inherent in metasurface design.

Chapter 5. This chapter presents a comprehensive evaluation of our numerical methods on different metasurface geometries. First, we assess the multiresolution RWG basis by comparing convergence rates, solve times, and far-field accuracy against the standard RWG discretization on a prototypical small aperture. Next, we scale up to a full-aperture design, analyzing how the hierarchical basis and preconditioners affect convergence, and beam-forming performance. Finally, we asses the differences between three optimizer strategies—nonlinear conjugate-gradient, Adam, and AdaHessian.

Chapter 6. This chapter outlines key avenues for further improving both the accuracy and convergence efficiency of MTS inverse-design optimization.

Chapter 2

Background and Literature Review

2.1 Metasurface Antenna Fundamentals

Metasurface (MTS) antennas are a class of antennas formed by thin planar layers of sub-wavelength scattering elements that collectively manipulate electromagnetic waves as discussed in 1. They are essentially the two-dimensional analog of metamaterials, engineered to transform an incident wave into a desired radiated wavefront. Figure1.1 illustrates several practical implementations of metasurfaces using different element technologies (e.g., printed metallic patches, dielectric posts, slots), all of which can be modeled in a homogenized sense as continuous impedance sheets. The multi-scale nature of metasurface antennas[1] makes direct full-wave design challenging. To enable tractable analysis and design, a common approach is to replace the detailed array of elements with an equivalent IBC defined on an abstract surface representing the metasurface. This impedance boundary provides a macroscopic description of how the metasurface interacts with fields, without modeling each unit cell in detail.

2.1.1 IBC & Surface Homogenization

When a metasurface is homogenized, it is modeled as a continuous surface characterized by a position-dependent impedance $Z(\mathbf{r})$. The impedance relates the tangential electric field $\mathbf{E}_{\tan}$ to the discontinuity in the tangential magnetic field across the surface. In general form, the IBC is expressed as:

$$\mathbf{E}_{\tan}(\mathbf{r}) = \overline{Z}(\mathbf{r}) \cdot [\hat{n} \times (\mathbf{H}^{+} - \mathbf{H}^{-})] \tag{2.1}$$

for any point $\mathbf{r}$ on the metasurface. Here, $\hat{n}$ is the unit normal to the surface[2], and $\mathbf{H}^{+}$ and $\mathbf{H}^{-}$ are the magnetic fields just above and just below the sheet, respectively. The quantity $\hat{n} \times (\mathbf{H}^{+} - \mathbf{H}^{-})$ is effectively the surface electric current density $\mathbf{J}$ flowing on the metasurface.[3]

Thus, in a simpler form, we may write:

$$\mathbf{E}_{\tan} = \overline{Z}\,\mathbf{J} \tag{2.2}$$

on the metasurface. The impedance $\overline{Z}$ can be a scalar for isotropic surfaces affecting one polarization or a tensor for anisotropic metasurfaces that couple polarizations. This IBC encapsulates the homogenized response of the myriad sub-wavelength elements—it tells us that the metasurface imparts a position-dependent boundary condition relating tangential fields, without needing to model each element individually.

Because the IBC is a macroscopic model, any impedance function $Z(\mathbf{r})$ obtained from design must correspond to a physically realizable arrangement of meta-atoms. In practice, this means the impedance profile must be passive and lossless, and also within practical

[1] Involving fine-featured unit cells across electrically large apertures

[2] Oriented, say, from side − to side +

[3] by Maxwell's surface boundary conditions

limits of achievable reactance values. A passive, lossless impedance surface is one that does not generate or absorb net energy, it only stores and redirects electromagnetic energy. In energetic terms, the time-average power absorbed by the metasurface must be zero everywhere on the surface. This condition can be derived from the complex power density at the surface.

The complex power per unit area associated with the fields on the sheet is given by:

$$\tilde{p}(\mathbf{r}) = \mathbf{E}_{\tan}(\mathbf{r}) \cdot \mathbf{J}^*(\mathbf{r}) \tag{2.3}$$

where $(\cdot)^*$ denotes the complex conjugate, and

$$\mathbf{J} = \hat{n} \times (\mathbf{H}^+ - \mathbf{H}^-) \tag{2.4}$$

For a passive, lossless metasurface, the real part of $\tilde{p}$ must vanish everywhere: $\Re\{\tilde{p}(\mathbf{r})\} = 0$. In the simple case of a scalar impedance ($\overline{Z} = ZI$), according to Equation (2.2) this reduces to:

$$\tilde{p} = \Re\left\{Z\,|\mathbf{J}|^2\right\} = |\mathbf{J}|^2\,\Re\{Z\} = 0, \tag{2.5}$$

which can only be satisfied for all nonzero currents $\mathbf{J}$ if $\Re\{Z\} = 0$. In other words, the impedance must be purely reactive (imaginary-valued). This is the fundamental requirement for a passive, lossless metasurface that the surface impedance at every point must have zero real part.

Physically, this means the meta-elements store but do not dissipate energy and they do not provide gain. For a general tensor impedance[4], the passivity/losslessness conditions are more involved. Essentially, each independent component of the impedance tensor must be purely imaginary as well. In fact, it can be shown that for a 2×2 impedance tensor (describing two orthogonal polarizations on the surface), all symmetric components must have zero real part, and any antisymmetric coupling terms must also be purely imaginary (or zero). The general form of a passive, lossless impedance tensor $\overline{Z}$ can be written as:

$$\overline{Z} = jX_{II}\,\hat{\mathbf{u}}_I\hat{\mathbf{u}}_I + R_{NN}\,\hat{\mathbf{u}}_N\hat{\mathbf{u}}_N + jX_{KK}\,\hat{\mathbf{u}}_K\hat{\mathbf{u}}_K + jX_{LL}\,\hat{\mathbf{u}}_L\hat{\mathbf{u}}_L, \tag{2.6}$$

where $\{\hat{\mathbf{u}}_I, \hat{\mathbf{u}}_N, \hat{\mathbf{u}}_K, \hat{\mathbf{u}}_L\}$ denote a suitable orthonormal basis for decomposing the tensor—typically with I, K corresponding to polarization directions and N, L to cross-coupling components.

In this expression, any term with a real coefficient (e.g., R_{NN}) represents instantaneous energy exchange between orthogonal polarizations and is generally not desired. A nonzero R_{NN} corresponds to a bianisotropic response which couples electric and magnetic fields. This kind of coupling complicates metasurface design and is beyond the scope.

Consequently, we restrict our designs to impedance surfaces with no resistive part and no nonreciprocal coupling ($R_{NN} = 0$). Under these conditions, the impedance at each point can be treated as a purely imaginary scalar for single-polarization designs or a

[4] e.g. if the metasurface exhibits anisotropy or bianisotropy

purely imaginary diagonal tensor for dual-polarized designs, which greatly simplifies both analysis and implementation [26].

To summarize, through homogenization we replace the metasurface by an ideal continuous boundary condition characterized by $Z(\mathbf{r})$. This model is valid under the assumption that the unit cells are electrically small and can be represented by an averaged response. It allows us to perform analysis and design at the surface level.

However, any synthesized $Z(\mathbf{r})$ must be checked against physical constraints: it must be purely reactive everywhere (passive/lossless), and its reactance values $X(\mathbf{r}) = \Im\{Z(\mathbf{r})\}$ must lie within practical bounds determined by realizable unit-cell geometries. For example, extremely large inductive or capacitive reactances might require unrealistically small or large features to implement. Therefore, an upper and lower bound, $X_{\min}$ and $X_{\max}$, is typically imposed on the impedance profile.

The benefit of the IBC approach is that it provides a design parameter—the spatial impedance distribution—that we can manipulate to achieve desired electromagnetic behavior, without explicitly dealing with the fine details of each meta-atom during the initial design stage.

2.1.2 Surface Integral Equation (SIE) Formulation

With the impedance boundary condition established, we can formulate the electromagnetic Surface Integral Equation (SIE) that governs the metasurface antenna. Essentially, the SIE is a statement of Maxwell's equations, specifically the appropriate form of the Electric Field Integral Equation (EFIE) applied to the impedance boundary. It provides a way to solve for the unknown surface currents or fields on the metasurface due to a given excitation. Two main formulations exist for single-layer metasurfaces [26]:

Opaque impedance formulation. Assumes the metasurface is backed by a perfect conductor or otherwise impenetrable, so fields exist only on one side of the surface. In this case, any currents induced on the metasurface can be considered as living on that illuminated side only (typical of, e.g., a reflectarray with a ground plane).

Transparent impedance formulation. Allows fields to exist on both sides of the metasurface (the sheet is not a perfect conductor). This is the more general case, appropriate for a free-standing metasurface or one on a substrate without a continuous ground plane [26]. Here, the metasurface may be illuminated from one side or excited by sources embedded within the structure (e.g., a surface-wave feed), and it can radiate to both sides.

In this work, we adopt the transparent formulation, as it covers the broadest range of scenarios including those where the source is on the metasurface itself. Under the transparent formulation, we consider the metasurface to be a permeable interface in a background medium. The theoretical tool to handle this is the surface equivalence principle. According to the equivalence theorem, the effect of the metasurface in an otherwise homogeneous domain can be modeled by placing equivalent currents on the surface that reproduce the fields discontinuities. In particular, if we conceptually enclose the sources

(feed) and any material interior in a volume and replace the actual metasurface with free space, we can enforce the same external fields by introducing equivalent surface currents on the boundary. For an impedance surface, the relevant equivalence conditions applied on the surface S_{IBC} are mentioned at Table 2.1

In electromagnetic scattering and boundary-integral formulations, we decompose the fields into their "incident" and "scattered" parts so that the actual fields present are simply their sum:

$$\mathbf{E}_{\text{tot}}(\mathbf{r}) = \mathbf{E}_{\text{inc}}(\mathbf{r}) + \mathbf{E}_{\text{s}}(\mathbf{r}) \tag{2.7}$$

$$\mathbf{H}_{\text{tot}}(\mathbf{r}) = \mathbf{H}_{\text{inc}}(\mathbf{r}) + \mathbf{H}_{\text{s}}(\mathbf{r}). \tag{2.8}$$

Here, $\mathbf{E}_{\text{inc}}$ and $\mathbf{H}_{\text{inc}}$ are the known fields launched by the source (for example, a plane wave or embedded feed) before any interaction, while $\mathbf{E}_{\text{s}}$ and $\mathbf{H}_{\text{s}}$ are the fields scattered by the object (antenna, metasurface, scatterer) in response to that illumination. The superposition principle guarantees that a measurement at any point sees the total field, which is exactly the sum of what was sent in and what has been scattered back or re-radiated.

Table 2.1. Equivalent surface currents

Type	**Expression**
Equivalent electric current	$\mathbf{J}_{\text{eq}} = \hat{n} \times \mathbf{H}_{\text{tot}}$
Equivalent magnetic current	$\mathbf{M}_{\text{eq}} = -\hat{n} \times \mathbf{E}_{\text{tot}}$

These arise from enforcing the continuity of tangential fields except for the known jumps (the impedance relation provides the jump condition for $\mathbf{E}$ in terms of $\mathbf{H}$). In practice, for a thin printed metasurface of conducting patches, there is negligible tangential E-field discontinuity (no significant magnetic surface current), so one can often set $M_{eq} = 0$ and focus on the electric surface current J_{eq}. This is the case here, as the metasurface is a single conducting layer with no magnetically active response; thus $\mathbf{M}_{eq} = 0$ and the problem unknown reduces to the electric current $\mathbf{J}(\mathbf{r})$ flowing on the metasurface [10].

Using the above considerations, we can derive the electric field integral equation with IBC (often called the EFIE-IBC). The total tangential electric field on the metasurface is the sum of the incident field (due to the source excitation alone) and the scattered field (radiated by the induced surface currents). By enforcing the IBC, we require that this total field equals $Z(\mathbf{r})J(\mathbf{r})$ at every point on S_{IBC}. Mathematically:

$$\left[\mathbf{E}_{inc}(\mathbf{r}) + \int_{S_{IBC}} \overline{\mathbf{G}}_{EJ}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dS'\right]_{\text{tan}} = Z(\mathbf{r})\, \mathbf{J}(\mathbf{r}), \quad \forall\, \mathbf{r} \in S_{IBC}. \tag{2.9}$$

This is the surface integral equation for the metasurface [26]. In this expression, $\mathbf{E}^{inc}$ is the known incident field (for example, a surface wave launched by an embedded source

or a plane wave illumination), and the integral term represents the scattered electric field produced by the unknown current distribution $\mathbf{J}$ on the surface. $\overline{\mathbf{G}}_{EJ}(\mathbf{r}, \mathbf{r}')$ is the dyadic Green's function that gives the electric field at observation point $\mathbf{r}$ due to a unit electric current source at $\mathbf{r}'$; it encapsulates the electromagnetic coupling between current elements on the surface. In general, $\overline{\mathbf{G}}_{EJ}$ may be the free-space Green's function or, if the metasurface is on a layered medium (e.g., a substrate), a multilayer Green's function appropriate to that stackup [10]. The subscript 'tan' indicates we take the component tangent to the surface, since the IBC relates tangential fields.

Equation (III) enforces that the tangential electric field consistent with a given surface current J (LHS of the equation) matches the required field dictated by the impedance boundary (RHS). It is a single integral equation for the unknown current $J(\mathbf{r})$ distributed on the surface. In operator notation, one can write this compactly as:

$$(\mathcal{L}J)(\mathbf{r}) + \mathbf{E}^{\text{inc}}_{\text{tan}}(\mathbf{r}) = Z(\mathbf{r})\,\mathbf{J}(\mathbf{r}), \tag{2.10}$$

where $\mathcal{L}$ is the electric field integral operator (EFIO) acting on J [10]. This formulation is very powerful in the sense that it reduces the full 3D field problem to an equation on a 2D surface. Moreover, it is well-suited to numerical solution via the Method of Moments (MoM), as will be discussed in the next section.

It's worth noting that (III) is a second-kind integral equation[5], which is generally well-behaved for iterative solution.

2.1.3 Discretization via Method of Moments

The integral equation (III) generally does not admit a closed-form solution for arbitrary shapes and impedance profiles, so we resort to numerical solution. The Method of Moments (MoM) is a well-established technique to solve such integral equations in electromagnetics [10]. The MoM approach involves discretization of the continuous current $J(\mathbf{r})$ into a finite set of basis functions and testing the integral equation against a corresponding set of testing functions. There are various numerical methods to choose basis functions and testing functions. If testing functions is chosen to be the same as the basis functions, this approach is called Galerkin method. This discretization converts the continuous equation into a matrix equation that can be solved on a computer.

In practice, the metasurface surface S_{IBC} is first meshed into small elements. These elements are often chosen to be triangles to cover arbitrary shaped surfaces. Over this mesh the unknown surface current is expanded as:

$$J(\mathbf{r}) \approx \sum_{n=1}^{N} I_n\,\mathbf{\Lambda}_n(\mathbf{r}), \tag{2.11}$$

[5] A "second-kind" Fredholm integral equation has the form $\phi(x) + \int_S K(x,y)\,\phi(y)\,dy = f(x)$, i.e. the unknown ϕ appears both outside and inside the integral. Thanks to the identity term, the operator $\mathbb{I} + \mathcal{K}$ is better conditioned (it admits a convergent Neumann series for $\|\mathcal{K}\| < 1$), so discretizations lead to stable, efficiently solvable linear systems.

where $\mathbf{\Lambda}_n$ are a set of known vector basis functions and I_n are the unknown expansion coefficients to be determined [10]. A common choice for metallic surfaces is to use the Rao-Wilton-Glisson (RWG) basis functions, defined on pairs of adjacent triangular mesh elements, which ensure current continuity across the mesh and are divergence-conforming. Each RWG basis Λ_n is associated with an interior edge of the mesh and represents a linearly varying current distribution on the two triangles sharing that edge. Using such bases, the surface current is fully described by the coefficients I_n on the mesh edges [10].

We can recast the continuous integral-differential equation (III)

$$\mathcal{L}[f](\mathbf{r}) \;=\; g(\mathbf{r}) \tag{2.12}$$

where $\mathcal{L}$ is our linear boundary-integral operator, g is the known excitation (e.g. the incident field), and f is the unknown surface current—into a finite, algebraic system by expanding f in a suitable basis. Concretely, we write

$$f(\mathbf{r}) \;=\; \sum_{n=1}^{N} I_n \, \varphi_n(\mathbf{r}), \tag{2.13}$$

where $\{\varphi_n\}_{n=1}^{N}$ are chosen RWG basis functions, and the coefficients $\{I_n\}$ become the unknowns in our Method of Moments discretization. Substituting this expansion into $\mathcal{L}[f] = g$ and enforcing a testing procedure (Galerkin, collocation, etc.) then yields an $N \times N$ linear system for the vector $\mathbf{I} = [I_1, \ldots, I_N]^T$.

Substituting (2.13) into (2.12) and using linearity yields

$$\sum_{n=1}^{N} I_n \, \mathcal{L}[\varphi_n](\mathbf{r}) \approx g(\mathbf{r}). \tag{2.14}$$

We define the residual function as

$$R(\mathbf{r}) = g(\mathbf{r}) - \sum_{n=1}^{N} I_n \, \mathcal{L}[\varphi_n](\mathbf{r}), \tag{2.15}$$

and require that the residual vanish in a weighted (weak) sense.

To enforce this, we introduce a set of test functions $\{w_m(\mathbf{r})\}_{m=1}^{N}$ and an inner product

$$\langle w_m, \varphi_n \rangle = \int_D w_m(\mathbf{r}) \cdot \varphi_n(\mathbf{r}) \, dD. \tag{2.16}$$

Here, D denotes the common support over which both sets of functions live.

$$D = D_f \cap D_w, \tag{2.17}$$

where D_f is the region on which the expansion basis $\{\varphi_n\}$ is defined, and D_w is the region on which the testing functions $\{w_m\}$ are defined. In practice this ensures that (2.16) only integrates over points where both w_m and φ_n are nonzero.

We then enforce

$$\langle w_m, R \rangle = 0, \qquad m = 1, \ldots, N, \tag{2.18}$$

substituting (2.18) into (2.15) gives the system of equations

$$\sum_{n=1}^{N} I_n \langle w_m, \mathcal{L}[\varphi_n] \rangle = \langle w_m, g \rangle, \qquad m = 1, \ldots, N. \tag{2.19}$$

The equation (2.19) can be written in compact form as

$$\mathbf{Z}\,\mathbf{I} = \mathbf{V}, \tag{2.20}$$

thus, the generic EFIE can be discretized. In (2.20),

$$\mathbf{Z} = \begin{bmatrix} \langle w_1,\, \mathcal{L}(\varphi_1) \rangle & \langle w_1,\, \mathcal{L}(\varphi_2) \rangle & \cdots & \langle w_1,\, \mathcal{L}(\varphi_N) \rangle \\ \langle w_2,\, \mathcal{L}(\varphi_1) \rangle & \langle w_2,\, \mathcal{L}(\varphi_2) \rangle & \cdots & \langle w_2,\, \mathcal{L}(\varphi_N) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle w_N,\, \mathcal{L}(\varphi_1) \rangle & \langle w_N,\, \mathcal{L}(\varphi_2) \rangle & \cdots & \langle w_N,\, \mathcal{L}(\varphi_N) \rangle \end{bmatrix} \tag{2.21}$$

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{bmatrix} \tag{2.22}$$

$$\mathbf{V} = \begin{bmatrix} \langle w_1,\, g \rangle \\ \langle w_2,\, g \rangle \\ \vdots \\ \langle w_N,\, g \rangle \end{bmatrix} \tag{2.23}$$

In a compact format we can write

$$Z_{mn} = \langle w_m, \mathcal{L}[\varphi_n] \rangle, \qquad V_m = \langle w_m, g \rangle, \qquad \mathbf{I} = [I_1, \ldots, I_N]^T, \tag{2.24}$$

yielding the Method-of-Moments system indicated in (2.20).

In the Galerkin method, one sets $w_m = \varphi_m$, resulting in a symmetric formulation where the test and basis functions are the same.

In summary, through MoM discretization we obtain a tool to compute, for a given impedance profile $Z(\mathbf{r})$ (hence a given $\mathbf{Z}$ matrix), what surface current and radiated fields will result. These analysis tools—the IBC modeling, SIE formulation, and MoM solution—form the foundation upon which we can build an inverse design methodology, as described next.

Before moving on, it is important to note that the accuracy of this model has been validated extensively in literature for metasurface antennas. By comparing MoM-IBC solutions to full-wave simulations or measurements, one finds that a well-chosen homogenized model can predict the antenna performance accurately while being far more computationally efficient[16, 22]. These methods will be used not only to design the metasurface (in inverse design context) but also later to verify the final design by a forward solve of the EFIE-IBC system.

2.2 Automated Inverse-Design Technique

With the analytical framework in place, we turn to the core topic of automated inverse design for metasurface antennas. The goal of inverse design is essentially the reverse of analysis: instead of determining the radiation from a given impedance profile, we seek to find the impedance profile (or equivalently, the surface currents) that will produce a desired radiation pattern, given a specified excitation. In other words, we want to solve the "inverse problem" of metasurface synthesis—automatically computing the required spatial impedance $Z(\mathbf{r})$ (or $J(\mathbf{r})$) to meet design objectives.

2.2.1 Geometry

We first define the metasurface antenna geometry and the design specifications. The techniques discussed are general, but for concreteness we consider a representative configuration: a finite planar metasurface of radius/width d (on the order of several wavelengths) that is excited by an integrated feed source. In many low-profile metasurface antennas, the feed is a surface-wave source embedded at or near the metasurface plane [26]. For example, the feed could be a small coaxially-fed monopole or a launch structure that generates a cylindrical surface wave that propagates outward across the metasurface.

This in-plane source configuration is a distinguishing feature of metasurface antennas compared to, say, conventional reflectarrays (which are illuminated by an external horn). It enables very compact, low-profile designs but also introduces a strong coupling between the source and the metasurface that must be accounted for [26].

For generality, we do not assume the presence of a perfectPerfect Electric Conductor (PEC) ground plane immediately behind the metasurface, i.e., we allow a transparent metasurface formulation (fields on both sides) as discussed in Section 2.1.2. In practice, many designs do include a ground plane a short distance behind the metasurface (especially if using a dielectric substrate) to prevent backside radiation and to support the surface wave. This effectively makes the metasurface opaque in terms of radiation. However, even such cases can be handled by the transparent formulation by including the ground plane in the Green's function of the background medium [26].

The important aspect of the geometry is that the metasurface is a single continuous sheet (possibly backed by layers) on which we can impose an impedance profile. We partition the metasurface aperture into a number of unit cells or subdomains for the purpose of design. In a physical device, each unit cell corresponds to one metamaterial element (e.g., one patch or one slot) whose geometry can be tuned to realize a specific impedance value. For the inverse design algorithm, it is convenient to define these cells such that the impedance is considered uniform within each cell. For instance, one might use a grid of small identical shapes (pixels) or simply leverage the MoM mesh elements as "cells".

In our approach, we effectively use the MoM discretization itself as the canvas for design: the surface is modeled by many small triangular patches and we can assign an impedance value to each patch (or group of patches) [26]. Thus, the continuous function $Z(\mathbf{r})$ is expanded in a piecewise-constant manner:

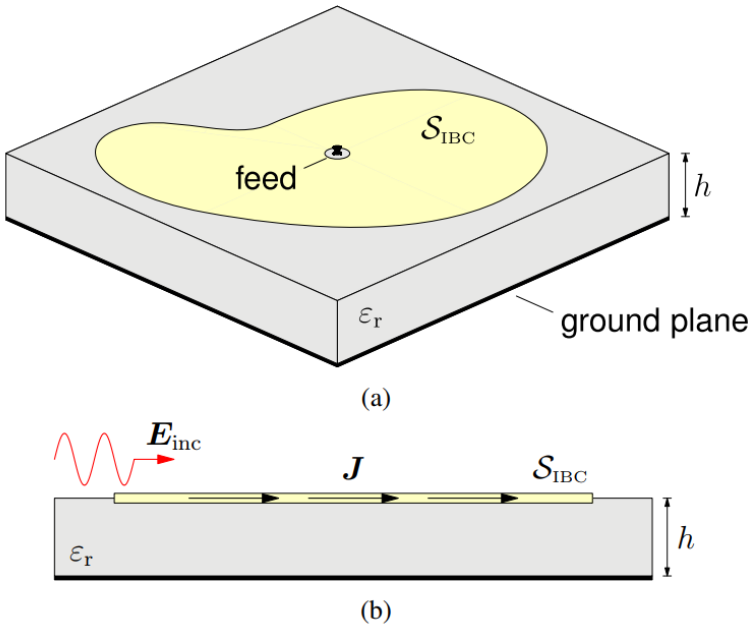

Figure 2.1. Example of geometry for a metasurface antenna layout: (a) 3D view showing the configuration with an on-surface source for the incident field; (b) cross-sectional view illustrating the equivalent surface current $\mathbf{J}$.

$$Z(\mathbf{r}) = \sum_{\ell=1}^{N_c} z_\ell \, \Psi_\ell(\mathbf{r}), \tag{2.25}$$

where N_c is the number of chosen cells and $\Psi_\ell(\mathbf{r})$ is 1 on cell ℓ and 0 elsewhere [26]. Each coefficient z_ℓ represents the impedance value on cell ℓ. This expansion is analogous to what we saw for currents in Eq. (2.11); here it is the impedance profile that is discretized.

In essence, the metasurface is divided into N_c elements, each assumed to have an approximately uniform impedance that we need to determine. Typically, the cell size is on the order of the unit cell period of the physical metasurface (often a small fraction of the wavelength). By choosing N_c sufficiently large, one can approximate a smoothly varying ideal impedance profile. In many design cases, N_c will equal the number of physical meta-elements. For example, if the metasurface uses a periodic lattice of patches, each patch (or each cluster of sub-cells for fine resolution) would correspond to one impedance variable.

It is important to note the coordinate system and polarization definitions in this geometry. We consider the metasurface lying in the xy-plane (at $z = 0$ for instance). The radiation pattern is observed in the far field characterized by spherical angles (θ, ϕ): θ is the polar angle from broadside ($+z$ axis) and ϕ the azimuth angle. For a single-feed design, we often aim for an axial beam (along $+z$) or a shaped beam within some angular sector.

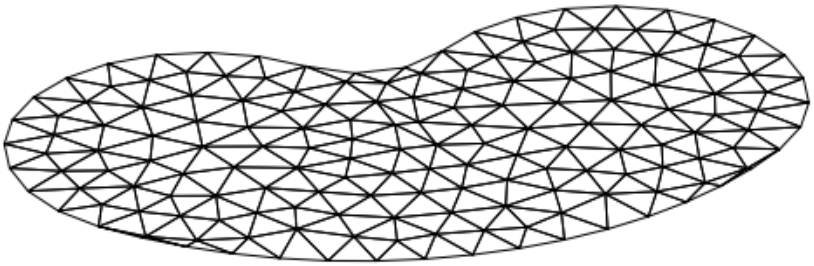

Figure 2.2. Example of mesh employed in the design. Only the impedance surface is discretized, and the effect of the grounded substrate (background medium) is taken into account by the multilayer Green's function.

The metasurface can be designed for a specific polarization of radiation (e.g., linear polarization oriented along some axis, or circular polarization). For linearly polarized designs, $Z(\mathbf{r})$ can be considered a scalar or at most a diagonal tensor (one value for the electric field component along the polarization direction of interest). For circularly polarized designs, anisotropic impedance patterns might be required, or one may design two orthogonal linear polarization distributions with a quadrature phase difference to synthesize circular polarization. In the scope of our discussion, we will focus on the impedance as a scalar function, keeping in mind that extension to tensorial impedance is possible if needed for polarization control.

2.2.2 Constraint Definition

Automating the design requires formalizing the design constraints and objectives. We have already touched upon the critical physical constraint which is the impedance must remain passive and lossless everywhere. Now we list all the major constraints that need to be considered in the inverse design problem:

Passivity & Losslessness. As derived in Section 2.1.1, this means $\Re\{Z(\mathbf{r})\} = 0$ for all points (no resistive component) [26]. The metasurface cannot locally absorb power; all incident power must be either re-radiated or guided away, and none can be lost as heat. In practice, a small resistive part might be tolerated if one designs a deliberately lossy matching layer, but here we focus on fully reactive surfaces (no intentional loss). This condition ensures the designed surface can be realized with purely reactive elements like metal patches and does not require active components or lumped loads that provide gain or dissipation [26].

Impedance Value Bounds. There are practical limits to the reactive impedance values that can be achieved by a meta-atom of given size and technology. Extremely high or low reactances may correspond to geometries that are too small or too large to fabricate, or they may push the homogenization assumption too far. We thus enforce $X_{\min} \leq X(\mathbf{r}) \leq X_{\max}$ for the reactance at every point. For example, $X_{\min}$ might be related to a capacitance limit when patches approach touching, and $X_{\max}$ might correspond to an inductance limit when patches shrink to small dots. These bounds ensure the synthesized impedance profile is realizable with actual structures. Typically, $X_{\min}$ and $X_{\max}$ are determined either by prior full-wave simulations of unit cells or by experimental considerations (such as avoiding element self-resonance frequency bands). In practice, every metasurface implementation imposes a finite "reactance window" dictated by the chosen unit-cell technology and substrate. For example, suppose our design uses printed square-patch elements on a given dielectric—after simulating or measuring those patches at the operating frequency, we find that the smallest feasible patch still exhibits about $+10\,\Omega$ of surface reactance, while the largest patch we can fit into a single unit cell provides up to $+300\,\Omega$. Any attempt to push below $10\,\Omega$ would require unrealistically large patch areas (or vanishingly small gaps), and any attempt above $300\,\Omega$ would force features so small and closely spaced that the etching process fails or losses become excessive.

Consequently, during optimization we impose a hard bound on the local reactance:

$$X_{min}\,\Omega \;\leq\; X(\mathbf{r}) \;\leq\; X_{max}\,\Omega, \quad \forall\,\mathbf{r} \in S_{\mathrm{IBC}}. \tag{2.26}$$

By constraining $X(\mathbf{r})$ to this interval, we guarantee that every synthesized impedance value corresponds to a patch geometry that our fabrication process can actually produce. In other words, the design algorithm never proposes $X(\mathbf{r})$ outside $[10,\ 300]\,\Omega$, because anything beyond this range falls into "unrealizable" territory for our chosen patch-technology-on-substrate.

Radiation Pattern Constraints: The raison d'être of the design is to meet certain far-field radiation criteria. Usually, the specifications are given in terms of a desired radiation pattern or a mask. A mask-type amplitude constraint means that at each angle (θ, ϕ) in the far field, the radiated field amplitude $F(\theta, \phi)$ should either exceed a minimum value (in the main lobe region) or stay below a maximum value (in sidelobe or forbidden regions) [26]. For example, a typical requirement might be: produce a pencil beam toward boresight with gain G_0, while ensuring sidelobe levels are at least 20 dB down in specified angular regions. This can be translated into an upper mask $F(\theta, \phi) \leq F_{\mathrm{mask}}(\theta, \phi)$ for those angles (sidelobe region) and an equality or lower bound in the main beam direction. Other pattern constraints might include null placement (forcing radiation to near-zero in certain directions), symmetry constraints, or polarization purity (e.g., cross-polarization levels to be kept low). For our formulation, we can denote the target far-field amplitude pattern as $F_d(\theta, \phi)$ (which could be piecewise defined by masks), and require $|F(\theta, \phi) - F_d(\theta, \phi)| \leq \epsilon$ or similar for all angles, where ϵ is a tolerance. In practice, one incorporates these pattern requirements into the objective function to be minimized (as discussed in Section 2.3.1), rather than as hard constraints, to allow some trade-offs.

Surface-Wave Power Conservation: In the case of surface-wave-fed metasurface antennas, there is an implicit constraint related to the guided wave: we want the incident surface-wave power to be gradually converted to radiation with minimal back-reflection. If the metasurface impedance profile is poorly designed, the surface wave may reflect significantly (like encountering a mismatch), resulting in standing waves on the surface and reduced efficiency. For a passive lossless surface, any local mismatch doesn't cause absorption but does cause power to be reflected back toward the source. Thus, a design strategy is to ensure local impedance matching between the surface wave and free-space radiation at each point, a concept sometimes termed local impedance equalization [7]. In more concrete terms, the impedance must be such that the forward traveling wave continuously leaks power out without any portion needing to reflect backward. While this condition is hard to enforce exactly in a global optimization, it can be encouraged by penalizing any residual guided wave at the aperture edge or by using analytical guidance (e.g., choosing an initial impedance profile that roughly follows the Marcuvitz leaky-wave solution for a given illumination). In some state-of-the-art design methods, the introduction of carefully tuned auxiliary surface waves is used to redistribute power and achieve local power conservation across the aperture [25, 4].

In our method, we will implicitly handle this by including the appropriate terms in the cost function (for instance, a term representing un-radiated power or reflected power can be minimized).

The above points can be summarized in Table 2.2 for clarity. These constraints must all be kept in mind when setting up the inverse design algorithm. Some of them (like passivity and impedance bounds) are hard constraints that must be satisfied in the final design. Others (like pattern fidelity or minimized reflection) are typically handled as soft constraints or objectives that the algorithm seeks to optimize.

Table 2.2. Key design constraints for inverse metasurface synthesis.

Constraint	**Description**
Passive & Lossless Surface	The metasurface must not absorb or generate net power. This requires a purely reactive local impedance: $\Re\{Z(\mathbf{r})\} = 0$ for all $\mathbf{r}$ [26].
Impedance Bounds	The local surface reactance $X(\mathbf{r}) = \Im\{Z(\mathbf{r})\}$ must lie within realizable bounds: $X_{\min} \leq X(\mathbf{r}) \leq X_{\max}$. These limits reflect fabrication constraints and available meta-atom geometries.
Radiation Pattern Mask	The far-field pattern must meet design specifications, such as achieving desired gain in the main lobe and suppressing sidelobes. Constraints can be expressed via an amplitude mask $F(\theta, \phi) \leq F_{\text{mask}}(\theta, \phi)$ in angular regions of interest [26].

It is worth noting that satisfying all these constraints simultaneously is non-trivial.

There is often a tension between pattern requirements and physical constraints. For instance, extremely low sidelobes might only be achievable if the aperture illumination is tapered, but a strong taper could imply regions of very low power flow which might conflict with using only passive elements (as they cannot amplify). Or, ensuring a very broad bandwidth might conflict with maintaining purely reactive impedance over frequency.

In this chapter, however, we focus on the single-frequency design problem (at the center frequency of operation), and the constraints above apply at that design frequency.

Having established what needs to be enforced, the next step is to cast the design task as a solvable mathematical problem—typically an optimization problem—and devise an algorithm that navigates the design space to find an optimal solution that meets the above criteria.

2.3 Current-based Optimization Algorithm

In this section, we describe in detail the proposed method for synthesizing metasurface antennas. The primary objective of the design procedure is to determine the spatial variation of the surface impedance. However, rather than optimizing impedance values directly, the algorithm is formulated to work exclusively with the equivalent surface current. At the conclusion of the optimization, the optimized current distribution is used to compute the corresponding impedance.

By focusing on the current instead of solving the forward problem at every iteration, the method achieves significant reductions in computational complexity. This strategy requires that the sought-after current can be constrained to correspond to a passive, lossless surface while also producing a radiated field that meets the specified performance criteria. Additionally, bounds on the reactance implied by the current must respect practical realizability constraints, all without explicitly evaluating that reactance during the optimization.

2.3.1 Algorithm Formulation

The optimal current is defined as the minimizer of an unconstrained cost function, expressed purely in terms of the current coefficients:

$$\mathbf{I}^\star = \arg\min_{\mathbf{I}\in\mathbb{C}^N} f(\mathbf{I})\,. \tag{2.27}$$

The total cost is split into two terms:

$$f(\mathbf{I}) = f_{\text{rlz}}(\mathbf{I}) + f_{\text{rad}}(\mathbf{I})\,, \tag{2.28}$$

where f_{rlz} enforces physical realizability (e.g., passivity, reactance bounds), and f_{rad} encodes electromagnetic performance (e.g., pattern matching).

Whenever inequality-type constraints appear—such as field masks or bounds on the reactance—we impose them using a ramp function:

$$r(x) = \max(x, 0)\,. \tag{2.29}$$

For example, the condition $a \leq b$ is satisfied when $r(a-b) = 0$.

To improve numerical behavior and avoid local minima, each term in the cost is constructed as a fourth-degree polynomial in $\mathbf{I}$, with ramp functions used for inequalities. The general cost takes the form:

$$f(\mathbf{I}) = \sum_d q_d(\mathbf{I})\, s_d(\mathbf{I}) + \sum_d \left[r(t_d(\mathbf{I}))\right]^2 = \mathbf{q}^T(\mathbf{I})\, \mathbf{s}(\mathbf{I}) + \mathbf{r}^T(t(\mathbf{I}))\, \mathbf{r}(t(\mathbf{I}))\,, \tag{2.30}$$

where

$$\mathbf{q}(\mathbf{I}) = [q_1(\mathbf{I}) \quad q_2(\mathbf{I}) \quad \cdots]^T\,, \tag{2.31}$$

$$\mathbf{s}(\mathbf{I}) = [s_1(\mathbf{I}) \quad s_2(\mathbf{I}) \quad \cdots]^T\,, \tag{2.32}$$

$$\mathbf{t}(\mathbf{I}) = [t_1(\mathbf{I}) \quad t_2(\mathbf{I}) \quad \cdots]^T\,. \tag{2.33}$$

Each q_d, s_d, and t_d is a quadratic form in the current vector:

$$q_d(\mathbf{I}) = \Phi\, \mathbf{I}^\dagger A_d \mathbf{I} + \mathbf{I}^\dagger \mathbf{b}_d + c_d\,, \qquad \Phi \in \{\Re, \Im\}\,, \tag{2.34}$$

where $A_d \in \mathbb{C}^{N\times N}$ is positive-definite, $\mathbf{b}_d \in \mathbb{C}^N$, and $c_d \in \mathbb{C}$. Restricting to even-degree polynomials guarantees boundedness below and ensures cost divergence as $\|\mathbf{I}\| \to \infty$. The matrices A_d are chosen to allow efficient evaluation, often using sparse structures or FFT-based matrix–vector products.

To minimize the cost (2.27), we employ a nonlinear Conjugate Gradient (CG) method, as summarized in Algorithm 1. The algorithm leverages the efficiency of MoM-based field evaluation and allows large problems to be solved with $\mathcal{O}(N \log N)$ complexity and $\mathcal{O}(N)$ memory.

Algorithm 1 Nonlinear Conjugate Gradient Method

Require: Initial guess $\mathbf{I}_0$
Ensure: Optimized current $\mathbf{I}^\star$
1: Compute initial gradient: $\mathbf{g}_0 \leftarrow \nabla f(\mathbf{I}_0)$
2: Set initial direction: $\mathbf{p}_0 \leftarrow -\mathbf{g}_0$
3: **for** $k = 0$ to $K_{\max} - 1$ **do**
4: **Line search:** find step size α_k that minimizes $f(\mathbf{I}_k + \alpha\, \mathbf{p}_k)$
5: Update current: $\mathbf{I}_{k+1} \leftarrow \mathbf{I}_k + \alpha_k\, \mathbf{p}_k$
6: Compute new gradient: $\mathbf{g}_{k+1} \leftarrow \nabla f(\mathbf{I}_{k+1})$
7: Compute CG coefficient: $\beta_k \leftarrow \frac{\|\mathbf{g}_{k+1}\|^2}{\|\mathbf{g}_k\|^2}$
8: Update search direction: $\mathbf{p}_{k+1} \leftarrow -\mathbf{g}_{k+1} + \beta_k\, \mathbf{p}_k$
9: **end forreturn** $\mathbf{I}^\star \leftarrow \mathbf{I}_{K_{\max}}$

A key step is the line search (step 4), which identifies the optimal step size α_k along the direction $\mathbf{p}_k$. Since each component of $f(\mathbf{I})$ is a low-degree polynomial in $\mathbf{I}$, we can efficiently derive or bracket the optimal α_k analytically, ensuring robust convergence.

After convergence, one may apply smoothing to the current before converting to an impedance distribution. However, in this work, no smoothing is applied so as to preserve the native structure of the optimized solution. The impedance profile is directly computed from the current using the inverse EFIE-IBC formula (Eq. (2.20)).

This formulation is somewhat hybrid, as it includes a condition that couples J and Z via the EFIE. In practice, we do not explicitly enforce EFIE at every step of the optimization; instead, we ensure that the J we work with is one that could exist on the surface for some Z. One way to do this is to incorporate the EFIE into the objective function as a penalty (ensuring J and E maintain consistency), or more directly, to always compute E from J using the forward operator (which automatically satisfies Maxwell's equations) and then use E to derive Z.

The strategy we adopt is to allow J to freely vary (subject to some regularization) and focus on matching the far-field pattern. The physical feasibility of J in terms of an impedance will be enforced by post-processing or by separate penalty terms. In simpler terms, our objective function can be written as a sum of terms:

- $f_{\text{rad}}(\mathbf{J})$: quantifies error in the radiated field relative to the target pattern (e.g., mean square error in far-field amplitude where the pattern has to match, plus large penalties if any sidelobe exceeds mask).
- $f_{\text{rlz}}(\mathbf{J})$: penalizes any violation of impedance constraints and power constraints. This could be further split into, for example, f_{act} (penalizing any real part in the effective impedance), f_{impL} and f_{impU} (penalizing if impedance needed falls below $X_{\min}$ or above $X_{\max}$), etc.

These can be combined into a single scalar cost function, for instance:

$$F_{\text{total}}(J) = f_{\text{rad}} + f_{\text{rlz}} = w_1 f_{\text{pattern}} + w_2 f_{\text{act}} + w_3 f_{\text{impL}} + w_4 f_{\text{impU}} + \cdots \tag{2.35}$$

where w_i are weights that balance the relative importance of each term. This approach is a penalized optimization or augmented Lagrangian method: rather than strictly enforcing the constraints at every iteration, we guide the solution by making violations costly in the objective [26]. In practice, we might start with certain weights and adjust them during the optimization (this is sometimes done via a homotopy or continuation method, where constraints are gradually enforced more strictly) [26].

For example, one could begin optimizing for the pattern with a relatively low penalty on impedance reactance bounds to allow the algorithm freedom, then progressively increase those penalties to push the solution into the feasible impedance range. The optimization is performed over the space of possible current distributions J, which (after MoM discretization) becomes an N-dimensional space of the current coefficients I_n. This is typically a very high-dimensional space (thousands of unknowns or more). We therefore employ a gradient-based optimization algorithm, which can handle high dimensions by using gradient (first-order) information rather than brute-force search. In particular, methods like nonlinear conjugate gradient (NLCG) or quasi-Newton (like L-BFGS) are suitable.

In this work, a gradient-based iterative algorithm with line search is used [26]. Each iteration involves computing the objective function and its gradient with respect to all current coefficients, then updating the currents in a direction that reduces the cost.

A crucial part of this is the ability to compute the gradient efficiently. Fortunately, the problem structure allows us to do so. The relationship between J and the radiated field is linear (through the radiation integral or MoM matrix), and many of our constraints are quadratic or at least differentiable with respect to J.

In essence, to find the sensitivity of the far-field pattern to changes in current, we can simulate a back-propagation: for example, inject the field error in the far field as sources and compute the resulting fields on the surface, which gives the gradient direction for adjusting J.

The mathematical details aside, the end result is that we can obtain $\nabla_{\mathbf{J}} f_{\text{pattern}}$ with a computational cost similar to one forward field computation. Likewise, the gradients of the impedance penalty terms can be derived from the current and field values at the surface (e.g., for the loss penalty, $\partial f_{\text{loss}}/\partial I_n \propto \Re\{Z(\mathbf{r})\}$ in the support of basis n, which in turn relates to $\Re\{E_{\tan}\}$ if J is known). All these gradient computations are implemented in our solver, allowing almost-linear complexity per iteration [26]. In fact, as noted earlier, using fast methods for the forward problem, the entire objective and gradient can be evaluated in $\mathcal{O}(N \log N)$ time [26], which is critical for large N.

The optimization proceeds iteratively: starting from an initial guess for $J(\mathbf{r})$, we compute the gradient of F_{total} and move J in the negative gradient direction (or a conjugate direction) by a certain step size that is determined by a line search. A specialized line search is employed to speed convergence: in our case, a polynomial approximation of the cost along the search direction is built (using, say, a few samples) and minimized analytically to find the optimal step [26]. This has been found to be faster than a simple backtracking or fixed-step approach, because it uses curvature information of the cost function along that line. In the implemented algorithm, a 4th-order polynomial fit to the objective is constructed at each line search using consecutive iterations, and the minimum of that polynomial is taken as the step [26]. This yields an efficient update without requiring many trial evaluations.

To put it succinctly, the algorithm can be outlined as:

1. **Initialize $\mathbf{J}^{(0)}$:** The starting current distribution. This could be, for example, the current that would produce the target pattern if unconstrained (sometimes obtained by solving an inverse source problem without the impedance constraint), or simply the incident field distribution (as a trivial initial guess).

2. **Evaluate cost and gradient:** Compute $F_{\text{total}}(\mathbf{J}^{(k)})$ and $\nabla F_{\text{total}}(\mathbf{J}^{(k)})$ using the forward solver and adjoint techniques.

3. **Update direction:** Determine a search direction $\mathbf{p}^{(k)}$ (for steepest descent, $\mathbf{p}^{(k)} = -\nabla F$, for conjugate gradient, combine with previous gradient, etc.).

4. **Line search:** Find an optimal step α along $\mathbf{p}^{(k)}$ that minimizes $F_{\text{total}}(\mathbf{J}^{(k)} + \alpha \mathbf{p}^{(k)})$. Use the polynomial approximation method to do this quickly [26].

5. **Update solution:** $\mathbf{J}^{(k+1)} = \mathbf{J}^{(k)} + \alpha \mathbf{p}^{(k)}$.

6. **Apply any intermediate constraints (optional):** One might at this point enforce any simple constraints. For example, if during the iterations we want to keep things physical, we could project out any component of J that causes negative radiation resistance, etc. However, in our approach we largely allow J to evolve freely and only enforce constraints strictly at the end or via penalties.

7. **Convergence check:** If F_{total} is below a threshold or the change in J is small, stop. Otherwise, increment k and repeat from step 2.

Because multiple terms are being optimized together, typically one monitors the individual components of the cost as well. For instance, one might see the far-field error drop initially, then as penalties ramp up, the impedance term gets addressed. In some implementations, the weights w_i are dynamically adjusted based on intermediate results (as was done in the reference design, which re-weighted the penalties in stages) [26]. This ensures that at the end, all constraints are satisfied without having excessively hindered the initial convergence toward the desired pattern.

2.3.2 Constraints Enforcement

After (or during) the optimization, the design must be rendered physically feasible by enforcing the constraints from Section 2.2.2 The primary constraints to enforce are the passivity ($\Re\{Z\} = 0$) and the impedance bounds. There are a couple of strategies to achieve this.

One simple approach is to perform the optimization allowing Z (or J) to momentarily violate the constraints if it helps find a good solution, and then project the solution onto the feasible set at the end. For example, once an optimal current distribution $\mathbf{J}_{\text{opt}}(\mathbf{r})$ is obtained, we calculate the corresponding unconstrained impedance profile:

$$Z_{\text{opt}}(\mathbf{r}) = \frac{E_{\tan}(\mathbf{r})}{J_{\text{opt}}(\mathbf{r})} \tag{2.36}$$

Using the final fields and currents. This Z_{opt} might have a small real part or some values out of bounds. We then modify it as follows: set $\Re\{Z_{\text{opt}}(\mathbf{r})\} = 0$ everywhere (simply drop the resistive part) and clip the imaginary part $\Im\{Z_{\text{opt}}(\mathbf{r})\}$ to lie within $[X_{\min}, X_{\max}]$ [26]. The result is an adjusted impedance $Z_{\text{phys}}(\mathbf{r})$ that is strictly passive and within limits. One can then do a final forward simulation (analysis) with Z_{phys} to verify how the pattern turned out. This approach is straightforward and guarantees the final design meets the physical constraints, although it might slightly degrade the achieved pattern if the adjustments were significant.

Alternatively, one can enforce the constraints at each iteration or as the solution progresses. For instance, after each update of $\mathbf{J}$, we could compute the instantaneous required $Z(\mathbf{r})$ and immediately zero out its real part and clamp its values. However, naively doing so can stall the optimization or make the problem non-differentiable. A smoother way

is to include the constraint terms as penalties (as we did) and gradually increase their weight. In the later iterations of the optimization, the algorithm naturally drives the solution into the feasible region because any residual real part of impedance or out-of-range reactance incurs a heavy cost. By the final iteration, $\Re\{Z\}$ is nearly zero and bounds are almost satisfied; a final projection as described above then causes minimal change to the solution [26]. This combined approach was effectively used in prior work and yielded good results with only negligible adjustment needed at the end.

In addition to the local impedance values, we also want to ensure the solution does not implicitly require non-passive behavior in terms of power. One check is to compute the active power balance: the total power radiated by the metasurface should not exceed (or fall too short of) the power provided by the source. If our solution for J implied that more power is radiated than provided, it would mean the metasurface is acting as an antenna with gain $> 100\%$ (impossible without active sources). Typically, including the passivity constraint $\Re\{Z\} = 0$ everywhere inherently prevents net power creation, but it is possible for an intermediate solution to momentarily act like it would need amplification (if, say, we heavily overweight pattern over power conservation).

In practice, we can monitor the active power functional

$$f_{\text{act}} = \int_S \Re\{\tilde{p}(\mathbf{r})\}\, \mathrm{d}S \tag{2.37}$$

which should be zero for a passive lossless design, and include it in the cost or at least verify that it approaches zero in the final design [26]. In the reference implementation, this was indeed done: the functional f_{act} was one of the terms tracked and its weight w_{act} was adjusted to drive it to nearly zero by the end of optimization, ensuring no active power was needed.

The enforcement of constraints is thus a combination of analytic adjustments and optimization-driven penalties. The end result of the current-based optimization algorithm is twofold:

1. an optimized current distribution $\mathbf{J}_{\text{opt}}(\mathbf{r})$ that yields the desired far-field pattern, and
2. a synthesized impedance profile $Z_{\text{phys}}(\mathbf{r})$ derived from that current which is guaranteed to be passive and realizable.

The design process deliberately does not assume any specific initial shape for $Z(\mathbf{r})$—it is discovered via the current synthesis. This is a key advantage: it avoids getting trapped in suboptimal designs due to preconceived phase profiles or analytically guessed impedance patterns. The algorithm can explore unconventional impedance variations that a human designer might not anticipate, all while respecting Maxwell's laws through the integral equation framework.

Finally, it is worth emphasizing that after obtaining $Z_{\text{phys}}(\mathbf{r})$, a final verification is usually performed. This involves plugging Z_{phys} back into a full-wave solver or our MoM model and analyzing the metasurface's performance. In Chapter 4, we will see examples of this verification step, where the EFIE-IBC is solved with the designed impedance to confirm that the radiated pattern meets the mask and that the device operates as expected [26].

The numerical results demonstrate that the proposed current-based inverse design strategy is effective: it can handle complex pattern specifications and large apertures, producing metasurface antenna designs with excellent performance (e.g., high directivity and controlled sidelobes) while inherently satisfying the passivity and physicality requirements. Recent literature further validates this approach—for instance, Xu *et al.* achieved near-100% aperture efficiency in a passive metasurface by optimizing the surface currents and impedance profile with a similar integral-equation-based method, and Budhu *et al.* demonstrated passive beamforming on conformal surfaces using a multi-phase current and impedance optimization procedure. These successes underscore the power of computational inverse design in the metasurface realm, as it marries rigorous electromagnetic modeling with modern optimization to push antenna performance to new heights.

Chapter 3

Multiscale Basis and Preconditioning for efficient MTS Inverse Design

3.1 Multi-resolution RWG Basis

In this section we introduce Multiresolution Rao–Wilton–Glisson (MR-RWG) basis to accelerate and stabilize large-scale Method of Moments (MoM) solves for metasurface antennas by hierarchically capturing both global and local current variations. Starting from a very coarse triangular mesh and repeatedly subdividing edges, we generate a nested sequence of meshes whose finest "pixel" level recovers full accuracy, while intermediate levels provide progressively coarser representations. At each level, we split the current into solenoidal (loop) and non-solenoidal (edge) components: the coarsest-level loops (or alternative tree-loop functions) serve as scaling functions, and the new loops and RWG functions introduced by each refinement act as wavelets (detail functions). Crucially, every MR-RWG function can be written as a sparse linear combination of the finest-level RWGs via simple inter-mesh reconstruction relations, so that the method plugs directly into existing MoM codes. By dramatically reducing the number of effective unknowns at early iterations, improving the conditioning of the system matrix, and enabling aggressive sparsification of negligible interactions, the MR-RWG scheme delivers orders-of-magnitude speedups in iterative convergence while never compromising the final full-resolution solution.

In multiresolution analysis, we build a nested sequence of function spaces by successively refining the mesh. Concretely, let

$$X_{\mathrm{RWG}_j} \tag{3.1}$$

denote the span of all RWG basis functions defined on the level-j mesh. Refining each triangle[1] produces a finer mesh whose RWG space contains the coarser one:

$$X_{\mathrm{RWG}_j} \subset X_{\mathrm{RWG}_{j+1}}. \tag{3.2}$$

This is the essence of equation (3.2).

Next, hierarchical decomposition theory tells us that the additional functions gained when moving from level j to $j+1$ themselves form a complementary "detail" subspace, which we call

$$W_{\mathrm{RWG}_j}. \tag{3.3}$$

By construction, every function in the finer space can be uniquely split into a part that lives in the coarser space plus a part in the detail space:

$$X_{\mathrm{RWG}_{j+1}} = X_{\mathrm{RWG}_j} \oplus W_{\mathrm{RWG}_j}, \tag{3.4}$$

as in equation (3.4). Here, the direct-sum symbol $\oplus$ indicates that $X_{\mathrm{RWG}_j} \cap W_{\mathrm{RWG}_j} = \{0\}$.

Chaining these decompositions over all levels $j = 0,1,\ldots,L-1$ yields the full multiresolution expansion of the finest-level space:

$$X_{\mathrm{RWG}_{\mathrm{finest}}} = X_{\mathrm{RWG}_0} \oplus W_{\mathrm{RWG}_0} \oplus W_{\mathrm{RWG}_1} \oplus \cdots \oplus W_{\mathrm{RWG}_{L-1}}, \tag{3.5}$$

[1] e.g. by bisecting its edges.

as in equation (3.5). Where L is the finest level.

By working directly in the multiresolution basis[2] we can carry out the entire matrix solve at reduced cost. Coarse-level scaling functions capture the bulk of the solution's energy, so the iterative solver converges quickly on those large-scale modes. Only the smaller wavelet coefficients remain to be refined, and because they live in progressively smaller subspaces, their contributions can be updated efficiently. Once the solver has found all scaling and wavelet coefficients to the desired tolerance, the true full-resolution current distribution is obtained simply by summing every component. It is exactly a linear combination—that is, a weighted sum—of all the scaling and wavelet basis functions. If we denote by $\{\phi_i^{(0)}\}$ the scaling functions defined on the coarsest mesh level, with corresponding coefficients $c_i^{(0)}$, and by $\{\psi_k^{(j)}\}$ the wavelet functions at refinement level j, with detail coefficients $d_k^{(j)}$, then the full high-resolution current distribution is reconstructed as:

$$J(\mathbf{r}) \approx \sum_i c_i^{(0)}\, \phi_i^{(0)}(r) + \sum_{j=0}^{L-1} \sum_k d_k^{(j)}\, \psi_k^{(j)}(r). \tag{3.6}$$

In matrix form, we can group all coarse-level coefficients into a vector $\mathbf{C}_0$, and detail coefficients at level j into vectors $\mathbf{D}_j$. The corresponding basis function matrices are $\mathbf{\Phi}_0$ and $\mathbf{\Psi}_j$. The full-resolution current vector is then given by:

$$\mathbf{I} = \mathbf{\Phi}_0\, \mathbf{C}_0 + \sum_{j=0}^{L-1} \mathbf{\Psi}_j\, \mathbf{D}_j. \tag{3.7}$$

Because each of these sets of coefficients is computed in its own much smaller subspace, the total solve remains computationally efficient. Yet the final answer—the complete current distribution—is the same linear combination of basis functions that would have been obtained by solving directly on the fine mesh.

[2]Each unknown is either a coarse *scaling* coefficient or a finer *wavelet* detail coefficient.

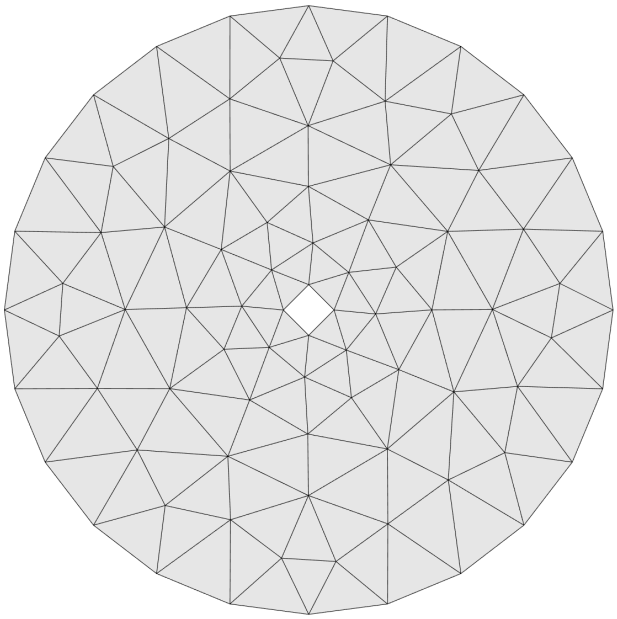

Figure 3.1. Example triangular mesh for a metallic surface. RWG basis functions are supported on each pair of adjacent triangles.

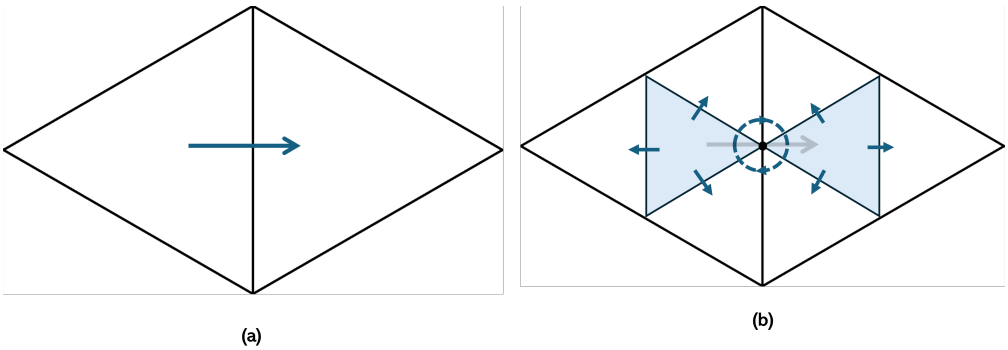

Figure 3.2. Illustration of an RWG basis function on two adjacent triangles. Arrows indicate the RWG basis with direction of surface current. (a) Coarse mesh $j = 0$ (b) Refined mesh $j = 1$

Figure 3.2 (a) illustrates a single RWG basis function defined over two adjacent triangles, where the arrow indicates the direction of current flow. Each RWG function is divergence-conforming and spans a pair of triangles. In the hierarchical construction, scaling functions are formed by combining multiple RWG functions over larger mesh patches. For instance, a group of RWG functions around a loop can be summed to form a coarse-level function, while subtracting adjacent basis functions results in a wavelet with zero mean. This preserves the divergence-conforming (edge-continuity) property essential for physical

accuracy.

Because the subspaces are nested (as in Equation (3.2)), the hierarchical transformation introduces no new degrees of freedom beyond those of the finest mesh.[3] These basis functions can be constructed algorithmically via graph-based refinement or edge-merging techniques.The key advantage of this approach is that it explicitly encodes multi-scale features. Low-frequency (coarse) current modes reside in X_{RWG_0}, while high-frequency (fine) variations are captured by successive W_{RWG_j} spaces. The hierarchical basis often exhibits vanishing-moment properties, meaning many off-diagonal entries in the MoM matrix become negligible and can be truncated. As reported in literature, this wavelet-like representation enables matrix sparsification with expected minimal impact on accuracy [2].

3.1.1 Construction of MR-RWG Basis

To build our MR-RWG basis, we start by generating a sequence of triangular surface discretizations of the metasurface, each one a uniform refinement of the previous. At the very coarsest level $j = 0$, the surface is represented by just a handful of large triangles; at each subsequent level $j = 1,2,\ldots$ we split every triangle until, at level $j = L$, we recover the finest "pixel" mesh used in the standard RWG Method-of-Moments.

On each level j, we collect all of the usual Rao–Wilton–Glisson basis functions defined on that mesh into a single row vector:

$$\varphi^{(j)} = [\, \varphi_1^{(j)},\ \varphi_2^{(j)},\ \ldots, \varphi_{N_j}^{(j)}]. \tag{3.8}$$

Here, N_j is the number of edges in the level-j mesh, and each $\varphi_n^{(j)}$ lives on exactly two adjacent triangles. The space of all possible currents that can be represented at resolution j is therefore:

$$X_{\mathrm{RWG}_j} = \mathrm{span}\left\{\varphi_1^{(j)}, \ldots, \varphi_{N_j}^{(j)}\right\}, \tag{3.9}$$

meaning any surface-current distribution on that mesh can be written as a linear combination of those N_j basis functions. As j increases, the mesh refines, N_j grows, and X_{RWG_j} becomes a strictly larger subspace, eventually matching the full high-resolution RWG space at $j = L$.

The construction of the multiresolution RWG (MR-RWG) basis proceeds in two tightly coupled phases. First, in the **inter-mesh reconstruction** step, we compute a sequence of sparse change-of-basis matrices $C^{(j)}$ that exactly express each level-j RWG basis function as a linear combination of the finer level-$(j + 1)$ RWGs. By chaining these matrices from the coarsest mesh ($j = 0$) up to the finest "pixel" mesh ($j = L$), we obtain a single block-triangular transformation matrix $T_{\mathrm{MR}\leftarrow\mathrm{RWG}}$ that maps all MR-basis functions directly into the standard fine-mesh RWG basis.
Second, in the **scaling/wavelet decomposition** step, we exploit the nested structure of the RWG function spaces.

[3] It merely represents a change of basis in the same functional space.

Inter-Mesh Reconstruction A fundamental property (see [24]) is that every level-j RWG can be written exactly in the next finer space:

$$\varphi^{(j)} = C^{(j)}\,\varphi^{(j+1)}, \qquad j = 0,1,\ldots,L-1, \tag{3.10}$$

where $C^{(j)} \in \mathbb{R}^{N_j \times N_{j+1}}$ is sparse and encodes the exact linear combination of level-$(j+1)$ RWGs that reproduces each coarser function. Each coarse-level basis can be expressed in terms of the next finer-level basis via its own sparse matrix $C^{(j)}$. Symbolically,

$$\varphi^{(0)} = C^{(0)}\,\varphi^{(1)}, \quad \varphi^{(1)} = C^{(1)}\,\varphi^{(2)}, \quad \ldots, \quad \varphi^{(L-1)} = C^{(L-1)}\,\varphi^{(L)}. \tag{3.11}$$

If we recursively substitute each equation into the one before it, we obtain a single relation that expresses the coarsest-level basis directly in terms of the finest-level basis:

$$\varphi^{(0)} = \left(C^{(0)}C^{(1)}\cdots C^{(L-1)}\right)\,\varphi^{(L)}. \tag{3.12}$$

So, by chaining these relations from $j = 0$ up to $L-1$, one obtains a single MR-to-pixel change-of-basis matrix

$$T_{\mathrm{MR}\leftarrow\mathrm{RWG}} = \begin{pmatrix} I & & & \\ C^{(0)} & I & & \\ C^{(0)}C^{(1)} & C^{(1)} & I & \\ \vdots & \vdots & \ddots & \ddots \end{pmatrix}, \tag{3.13}$$

which maps the full multiresolution basis—spanning all levels—into the pixel-level RWG basis $\varphi^{(L)}$. Rather than applying each interpolation matrix $C^{(j)}$ step-by-step, this construction allows us to apply their cumulative product in a single operation.

Scaling / Wavelet Decomposition Once the nested RWG spaces

$$X_{\mathrm{RWG}_0} \subset X_{\mathrm{RWG}_1} \subset \cdots \subset X_{\mathrm{RWG}_L} \tag{3.14}$$

have been established via inter-mesh reconstruction, each finer-level space splits into a coarser "scaling" subspace and a complementary "detail" (wavelet) subspace as in equation (3.4).

Here, X_{RWG_j} denotes the span of RWG functions on mesh level j, capturing all current modes that can be represented at that resolution. The detail space W_{RWG_j} supplies the new, finer-scale features required to augment X_{RWG_j} up to $X_{\mathrm{RWG}_{j+1}}$.

Detailed functions can be constructed as a combination of two distinct types of wavelets.

Solenoidal (loop) wavelets. Each refinement introduces new interior nodes on the mesh. Around each such node, one constructs a localized loop basis function supported on the immediately surrounding triangle fan. These level-j loop functions lie entirely in W_{RWG_j} and capture divergence-free (solenoidal) current patterns not representable at coarser resolutions.

Non-solenoidal (edge) wavelets. Simultaneously, every new edge[4] gives rise to a standard RWG basis function. Since refining a triangle pair introduces six new edges, each such refinement produces six new non-solenoidal wavelets. Excluding any RWGs inherited from the coarser level ensures these functions span the non-solenoidal portion of W_{RWG_j} and, together with the loops, form a complete detail basis.

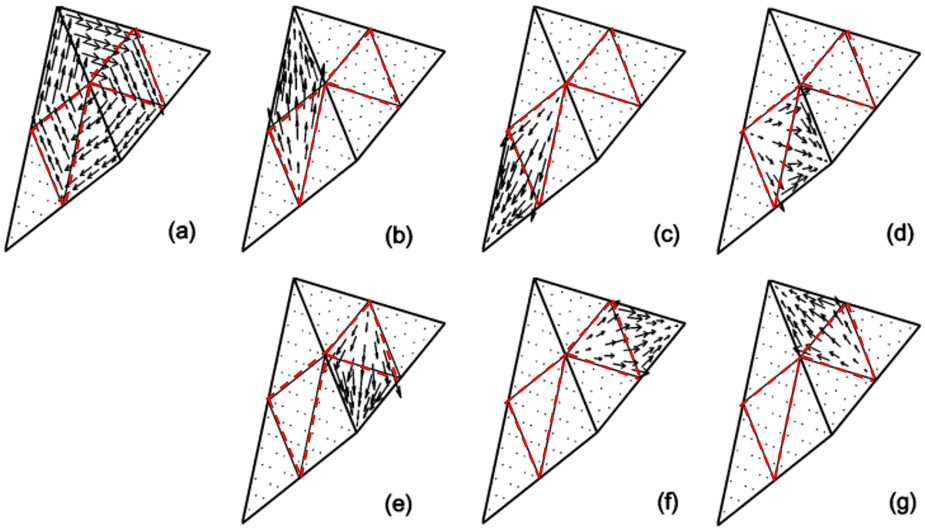

Figure 3.3. Detail functions at level $j = 1$. (a) Solenoidal loop function; (b)–(g) non-solenoidal edge (RWG) functions[24].

Finally, by chaining together the inter-mesh expansions and the scaling–wavelet splits across all levels, every MR-RWG basis function—whether a level-0 scaling function or a level-j detail wavelet—admits an exact representation in terms of the finest ("pixel") RWGs. Collecting the level-0 scalings $\{\phi_i^{(0)}\}$ and all detail wavelets $\{\psi_k^{(j)} : 0 \le j < L\}$ into a single stacked vector,

$$\mathbf{\Phi}_{\mathrm{MR}} = \begin{bmatrix} \phi^{(0)} \\ \psi^{(0)} \\ \psi^{(1)} \\ \vdots \\ \psi^{(L-1)} \end{bmatrix}, \tag{3.15}$$

and denoting by $\varphi^{(L)}$ the row-vector of the N_L pixel-level RWGs, this change of basis is expressed succinctly as

$$\mathbf{\Phi}_{\mathrm{MR}} = T_{\mathrm{MR}\leftarrow\mathrm{RWG}}\, \varphi^{(L)}, \tag{3.16}$$

[4]Defined as an edge present on level $j+1$ but absent from level j

where $T_{\mathrm{MR}\leftarrow\mathrm{RWG}}$ is a block-triangular matrix composed of the sparse inter-mesh reconstruction matrices $C^{(j)}$ and identity blocks.

In practice, this means that once the Method–of–Moments system has been assembled and solved for the MR-coefficients $\{c_i^{(0)}, d_k^{(j)}\}$, the full high-resolution current (see Eq. (3.6))is recovered via a single sparse matrix–vector product using $T_{\mathrm{MR}\leftarrow\mathrm{RWG}}$. No additional fine–mesh solves are required, yet the resulting current lies exactly in the original pixel-level RWG space.

3.1.2 Multilevel Optimization in the MR–RWG Basis

In the MR–RWG framework, multilevel optimization is implemented by recasting our original fine-mesh design variables $I \in \mathbb{R}^N$ into a hierarchy of coarse- and fine-scale coordinates $c \in \mathbb{R}^M$, and then solving progressively richer subproblems in these nested subspaces. Concretely, we start from the block-triangular change–of–basis matrix

$$T_{\mathrm{MR}\leftarrow\mathrm{RWG}} \;\in\; \mathbb{R}^{M\times N}, \tag{3.17}$$

whose rows collect all scaling and wavelet basis functions expressed in the fine-mesh RWG basis. Denoting by $I \in \mathbb{R}^N$ the vector of pixel-level currents and by

$$c \;=\; \begin{bmatrix} c_0 \\ d_0 \\ \vdots \\ d_{L-1} \end{bmatrix} \;\in\; \mathbb{R}^M \tag{3.18}$$

the concatenated MR–RWG coefficient vector (with c_0 in the coarsest scaling space and each d_j spanning the detail subspace W_{RWG_j}), the forward and inverse transforms read

$$c \;=\; T_{\mathrm{MR}\leftarrow\mathrm{RWG}}\, I, \qquad I \;=\; T^{-1}_{\mathrm{MR}\leftarrow\mathrm{RWG}}\, c, \tag{3.19}$$

where T^{-1} denotes the sparse inverse (or pseudoinverse if $M = N$).

We then rewrite our original objective $f(I)$ in these coordinates as

$$\widetilde{f}(c) \;=\; f(T^{-1}_{\mathrm{MR}\leftarrow\mathrm{RWG}}c), \tag{3.20}$$

and minimize $\widetilde{f}(c)$ not all at once but block by block. First, we restrict to the coarsest subspace by holding every detail block d_j at zero and optimizing only c_0. The reduced problem

$$c_0^\star \;=\; \arg\min_{c_0}\; \widetilde{f}(c_0,0,\ldots,0) \tag{3.21}$$

lives in a very low-dimensional, well-conditioned space, and converges rapidly to $c_0^\star$. Next we "unfreeze" the first detail block d_0, keep all higher d_j zero, and solve

$$d_0^\star \;=\; \arg\min_{d_0}\; \widetilde{f}(c_0^\star, d_0,0,\ldots,0). \tag{3.22}$$

We continue level by level—at each stage introducing only the new detail coefficients d_j—until all blocks $\{c_0^\star, d_0^\star, \ldots, d_{L-1}^\star\}$ have been optimized. Finally, we map back to the full fine-mesh solution via

$$I^\star \;=\; T_{\mathrm{MR}\leftarrow\mathrm{RWG}}^{-1}\, c^\star. \tag{3.23}$$

This multilevel strategy delivers several additional benefits. By capturing the dominant low-frequency current modes first, the coarse-level solve provides an excellent warm start for subsequent refinements, reducing the total number of gradient iterations required. Each detail subproblem involves only a few hundred variables at most, allowing per-level preconditioning tailored to that scale—which further improves conditioning and reduces function-evaluation cost. And since the inter-mesh transforms are sparse, the overhead of switching between representations is negligible compared to the gains in convergence speed. In practice, this approach can reduce overall runtime by one to two orders of magnitude compared to a blind fine-mesh optimization, while guaranteeing that the final full-resolution solution is identical to what one would obtain by solving directly on the pixel mesh.

3.1.3 Optimization in MR–RWG Basis

We can "lift" the entire optimization into the MR–RWG coordinate system by changing variables from the fine-mesh currents I to the multiresolution coefficients c. Concretely, we can define

$$c \;=\; T_{\mathrm{MR}\leftarrow\mathrm{RWG}}\, I, \qquad I \;=\; T_{\mathrm{MR}\leftarrow\mathrm{RWG}}^{-1}\, c, \tag{3.24}$$

and rewrite your objective $f(I)$ as

$$\widetilde{f}(c) \;=\; f\big(T_{\mathrm{MR}\leftarrow\mathrm{RWG}}^{-1} c\big). \tag{3.25}$$

Then run your optimizer (NLCG, Adam, AdaHessian, ...) on the variable c, never touching I directly. At each iteration:

1. **Evaluate cost:**
$$f \;=\; f\big(T_{\mathrm{MR}\leftarrow\mathrm{RWG}}^{-1} c\big).$$

2. **Compute gradient in I-space:**
$$g_I \;=\; \nabla_I f\big(T_{\mathrm{MR}\leftarrow\mathrm{RWG}}^{-1} c\big).$$

3. **Push gradient into c-space:**
$$g_c \;=\; T_{\mathrm{MR}\leftarrow\mathrm{RWG}}\, g_I.$$
Feed g_c to your optimizer.

4. **Take an update step:**
$$c \;\leftarrow\; c + \alpha\, p(c, g_c),$$
and if you ever need the physical currents, recover $I = T_{\mathrm{MR}\leftarrow\mathrm{RWG}}^{-1} c$.

Because $T_{\mathrm{MR}\leftarrow\mathrm{RWG}}$ is sparse and block-triangular, all these transforms cost only $\mathcal{O}(N)$. By optimizing directly in c-space you retain full conjugacy and momentum across scales—no repeated restarts—while still recovering the same fine-mesh solution $I^\star$ (often with faster convergence than a one-shot fine-mesh solve).“‘

3.2 Preconditioning Strategies

The metasurface radiation operator is inherently dense in the far-field, the electric field is given by an integral over the entire current distribution (e.g., via a Stratton–Chu or Green's function formulation), such that each basis-function current I_n contributes to the field at all observation angles. see Equation (III). Equivalently, the Method of Moments (MoM) matrix for an EFIE-IBC system is fully populated [1]. Thus, a change in a single current coefficient alters every far-field and near-field residual. This global coupling means the system is sensitive to any parameter disparity. For example, multiscale geometry (mixtures of sub-wavelength and larger features) leads to a very wide range of element sizes and reactances, causing ill-conditioned matrices. For example, multiscale geometry (mixtures of sub-wavelength and larger features) leads to a very wide range of element sizes and reactances, causing ill-conditioned matrices [14]. Similarly, reactive resonances (elements operating near internal resonant frequencies) introduce nearly singular behavior: at resonance the EFIE operator develops null-spaces (internal resonances) and the system becomes ill-conditioned. Anisotropic sampling or weighting of the response—such as non-uniform far-field angular sampling or highly disparate objective weights—results in certain current variations producing large changes in the cost function, while others have negligible effect. This imbalance stretches the optimization landscape into long, narrow “ravines,” making convergence more difficult for gradient-based methods [9]. Additional factors contributing to ill-conditioning include:

- **Feed coupling:** A strong dependence of many surface currents on the driven element results in localized overexcitation and imbalanced influence across the system.
- **Constraint ramping:** Nonlinear “activation” of constraints during optimization—such as gradually applied penalties or activation schedules—alters the local curvature of the design space dynamically, often leading to abrupt changes in sensitivity.
- **Finite-precision effects:** Round-off errors and numerical cancellations become significant in high-dynamic-range problems, particularly when matrix elements differ by several orders of magnitude.

These mechanisms collectively produce Jacobian or Hessian matrices with extremely large condition numbers, amplifying numerical instability and degrading both solver convergence and optimization performance.
In the inverse design of a metasurface antenna, one must repeatedly solve the Method-of-Moments (MoM) system (2.20). We can Interpret this formula as:

$$(\mathbf{Z}^d - \mathbf{L})\,\mathbf{I} = \mathbf{V}_{\mathrm{inc}}, \tag{3.26}$$

where $\mathbf{Z}^d = \mathrm{diag}(jX_1, \ldots, jX_N)$ contains the individual element reactances, and $\mathbf{L}$ represents the mutual coupling matrix. When the layout includes both deeply sub-wavelength inclusions (with small $|X_n|$) and larger phase-shifting elements (with large $|X_n|$), the diagonal entries of $\mathbf{Z}^d$ can span several orders of magnitude. Meanwhile, $\mathbf{L}$ introduces strong cross-scale interactions via its off-diagonal terms. This disparity drives the condition number $\kappa(\mathbf{Z}^d - \mathbf{L})$ to very large values, which in turn slows down the convergence of iterative solvers and amplifies the impact of finite-precision round-off errors. This motivates the use of preconditioning strategies to accelerate convergence in metasurface (MTS) design optimization, particularly in the presence of severe ill-conditioning.

3.2.1 Mesh-Level (Unit Edge-Flux) Preconditioning

In the RWG-based Method of Moments (MoM) formulation, each unknown current coefficient I_n multiplies a basis function $\varphi_n(\mathbf{r})$, which is defined over the two triangles that share edge n. This edge has length ℓ_n, and the associated RWG function represents a surface current whose total flux across the shared edge is proportional to $I_n\,\ell_n$. In other words, for fixed I_n, a longer edge injects more net current into the geometry than a shorter one, due to the linear dependence of the RWG flux on edge length. The key point comes directly from how the RWG basis functions are defined.

By construction, the RWG basis function $\varphi_n(\mathbf{r})$ associated with edge n satisfies the integral identity:

$$\iint_{\mathrm{supp}(\varphi_n)} \varphi_n(\mathbf{r})\, dS = \ell_n, \tag{3.27}$$

where ℓ_n is the length of the edge shared by the two supporting triangles.

When the total surface current is expressed as a linear combination of RWG functions:

$$\mathbf{J}(\mathbf{r}) = \sum_n I_n\, \varphi_n(\mathbf{r}), \tag{3.28}$$

the net current flux flowing across the shared edge n is given by:

$$\Phi_n = \iint \mathbf{J}(\mathbf{r}) \cdot \hat{\mathbf{t}}_n\, dS = \iint I_n\, \varphi_n(\mathbf{r})\, dS = I_n\, \ell_n, \tag{3.29}$$

where $\hat{\mathbf{t}}_n$ is a unit tangent vector aligned with the direction of current flow across the edge. Thus, for a fixed coefficient I_n, a longer edge length ℓ_n results in proportionally larger total current flux across that edge. Each entry of the Method of Moments (MoM) impedance matrix involves a double surface integral of the form:

$$Z_{mn} \propto \iiiint G(\mathbf{r}, \mathbf{r}')\, \varphi_n(\mathbf{r}')\, \varphi_m(\mathbf{r})\, dS'\, dS, \tag{3.30}$$

where $G(\mathbf{r}, \mathbf{r}')$ is the Green's function, and $\varphi_n(\mathbf{r}')$, $\varphi_m(\mathbf{r})$ are RWG basis and testing functions, respectively[5].

[5] Galerkin's method

Since each RWG basis function φ_n integrates to its associated edge length ℓ_n, the overall magnitude of column $Z_{:n}$ in the impedance matrix (and similarly the corresponding column in the far-field projection matrix) scales with ℓ_n.

Therefore, a longer edge effectively "injects" more current into all testing functions φ_m, resulting in proportionally larger values in column n of the matrix.

We can interpret the coefficient I_n as a current per unit length only if it is scaled by $1/\ell_n$, where ℓ_n is the length of the RWG edge. Without this scaling, a fixed value of I_n on a longer edge results in more total charge movement—and therefore stronger radiated or scattered fields—simply because a longer conductor supports more current flow. In other words, the same amplitude I_n produces more physical current, and thus couples more strongly into the integral equations, when the associated edge length ℓ_n is larger.

Because the columns of the system matrix grow in norm approximately proportional to ℓ_n, the ratio

$$\kappa \;\propto\; \frac{\max_n \ell_n}{\min_n \ell_n} \tag{3.31}$$

directly contributes to the condition number of the resulting linear system.

Without compensating for this disparity—such as through unit-flux scaling—basis functions on longer edges dominate the system behavior, leading to a badly unbalanced and ill-conditioned matrix.

To enforce unit-flux per coefficient, we introduce a simple diagonal scaling. Let

$$\ell_n \;=\; \text{length of edge } n, \qquad D \;=\; \operatorname{diag}(\ell_1, \ell_2, \ldots, \ell_N). \tag{3.32}$$

We then rescale the RWG basis (or equivalently the unknowns) so that the new basis functions

$$\psi_n(\mathbf{r}) \;=\; \frac{\varphi_n(\mathbf{r})}{\ell_n} \tag{3.33}$$

satisfy the unit-flux condition:

$$\iint \psi_n(\mathbf{r})\, dS = \frac{1}{\ell_n} \iint \varphi_n(\mathbf{r})\, dS = 1. \tag{3.34}$$

Equivalently, if $\mathbf{I}$ collects the original coefficients in (3.28), we define scaled coefficients $\widetilde{\mathbf{I}}$ by

$$\widetilde{\mathbf{I}} = D\,\mathbf{I}, \quad \Longrightarrow \quad \mathbf{J}(\mathbf{r}) = \sum_n \widetilde{I}_n\, \psi_n(\mathbf{r}). \tag{3.35}$$

Substituting into the Method-of-Moments system $\mathbf{Z}\,\mathbf{I} = \mathbf{V}$ and multiplying both sides by D, we obtain

$$D\,\mathbf{Z}\,D^{-1}\,\widetilde{\mathbf{I}} = D\,\mathbf{V}. \tag{3.36}$$

Thus, the preconditioned matrix

$$\widetilde{\mathbf{Z}} = D\,\mathbf{Z}\,D^{-1} \tag{3.37}$$

has columns (and rows) with norms independent of edge length ℓ_n. We solve

$$\widetilde{\mathbf{Z}}\,\widetilde{\mathbf{I}} = D\,\mathbf{V} \tag{3.38}$$

and recover the original coefficients via

$$\mathbf{I} = D^{-1}\widetilde{\mathbf{I}}. \tag{3.39}$$

By enforcing unit edge-flux in this way, we eliminate the large dynamic range in column norms (and in gradient components), thereby dramatically improving the conditioning and convergence speed of both direct and gradient-based solvers.

3.2.2 Gradient Preconditioning via Hessian Diagonal

Previously we introduced unit-flux preconditioning to balance every current coefficient by its edge length so that no basis function overpowers the update simply because it spans a longer mesh edge. Building on that foundation, Hessian preconditioning goes a step further by measuring how sharply the cost function curves along each current coefficient and then rescaling every gradient component by the inverse of its own second derivative. In practice we compute or approximate only the diagonal entries of the Hessian so that sharp dimensions are automatically damped and flat dimensions receive larger nudges. This per-coordinate scaling turns the long narrow "ravines" of the metasurface design landscape into more uniform basins and unleashes much faster, more balanced progress without ever assembling or inverting the full Hessian.

Because our metasurface-antenna objective function combines explicit quadratic power-balance terms (see Section 2.3) with smooth ramp penalties, its second derivatives admit closed-form expressions. Since the cost is a real-valued function of complex currents, we employ Wirtinger (CR) calculus to compute its Hessian exactly [13].

A convenient way to define the complex Hessian for a real-valued function

$$f : \mathbb{C}^N \to \mathbb{R}, \quad I \mapsto f(I), \tag{3.40}$$

is via the Wirtinger-type second derivatives. In particular, by treating I and its complex conjugate I^* as independent variables, the complex Hessian is given as a $2N \times 2N$ block matrix of all second derivatives:

$$\nabla^2_{CR} f(I) = \begin{pmatrix} \frac{\partial^2 f}{\partial I\,\partial I^*} & \frac{\partial^2 f}{\partial I\,\partial I} \\ \frac{\partial^2 f}{\partial I^*\,\partial I^*} & \frac{\partial^2 f}{\partial I^*\,\partial I} \end{pmatrix}. \tag{3.41}$$

Of the four second-derivative blocks in the complex Hessian, the block that directly captures the real curvature along each complex-coefficient direction is the mixed derivative

$$H_{mixed} \;=\; \frac{\partial^2 f}{\partial I\,\partial I^*}, \tag{3.42}$$

which belongs to the upper-left part of the full complex Hessian matrix $\nabla^2_{CR} f$. This block is Hermitian by construction and, for a real-valued cost function, it contains the precise information about how small variations in each complex coefficient I_n influence

the gradient norm. In practical implementations, we extract only the diagonal entries of H_{mixed} to estimate local curvature. That is, we define

$$H_{ii} \;=\; \frac{\partial^2 f}{\partial I_i \, \partial I_i^*}, \tag{3.43}$$

and use this diagonal as the foundation of our curvature-based preconditioner. This Jacobi-style approximation is efficient, requiring only $\mathcal{O}(N)$ operations to compute and store, and proves sufficient for rebalancing step sizes across the stiff and flat directions encountered in metasurface optimization problems.

H_{ii} reflects the sensitivity of the cost function to perturbations in the i^{th} design variable. Large variations in curvature across coordinates result in ill-conditioning, which hampers the efficiency of gradient-based solvers. In another words:

In our inverse-design problem, the diagonal entry H_{ii} of the Hessian quantifies how sharply the cost function changes when the ith current coefficient I_i is perturbed. When these diagonal curvatures vary significantly across coordinates, the resulting ill-conditioning severely hampers the convergence rate of any gradient-based solver. Broadly, two limiting regimes emerge:

Stiff directions ($|H_{ii}| \gg 1$). These are associated with basis functions whose currents exert a disproportionately large influence on the far-field pattern—such as elements near the main-beam steering zone or those operating near reactive resonances. If left unscaled, these stiff directions force the global step size to be extremely small to maintain stability, effectively stalling progress in all other directions. By dividing the update ΔI_i by $|H_{ii}|$, we attenuate these overly sensitive coordinates and allow a larger overall step size without risking divergence.

Flat directions ($|H_{ii}| \ll 1$). These correspond to basis functions located in weakly radiating zones, electromagnetically shadowed regions, or areas where penalties are inactive. The gradient magnitudes here are small, so uniform step sizes result in negligible updates—causing stagnation. Rescaling by $1/|H_{ii}|$ boosts these directions, helping them evolve on the same footing as the stiff components.

This simple Jacobi (diagonal) preconditioning therefore balances the effective step lengths across all coordinates, significantly accelerating convergence in our non-convex metasurface-antenna optimization landscape.

Using a single scalar step size η implicitly assumes that every design variable I_i experiences the same local curvature and gradient scale. In practice, Some currents lie in *stiff* regions—such as those near resonances or tightly coupled to the main lobe—where even a small perturbation leads to a large change in the objective function. Others lie in *flat* regions, where they are weakly radiating, masked out, or otherwise have little influence on the cost. As a result, a single global step size η must be chosen small enough to ensure stability in the stiffest direction, which effectively freezes progress along all flat directions.

By introducing a diagonal preconditioner

$$\mathbf{P} = \mathrm{diag}(|\mathbf{H}_{11}|, \ldots, |\mathbf{H}_{NN}|), \tag{3.44}$$

Each update is then computed as:

$$\Delta I_i = -\eta \, \frac{g_i}{P_{ii}}, \tag{3.45}$$

is scaled inversely to its local curvature. Stiff directions (large P_{ii}) are damped, while flat directions (small P_{ii}) are boosted.

This per-coordinate adaptation equalizes the effective step length across all variables, allowing faster and more balanced convergence than a single global step size. In a non-convex optimization landscape with many active ramp constraints and piecewise changes, coordinate-wise scaling helps prevent the line-search from repeatedly backing off to satisfy descent conditions in a few stiff directions. This saves numerous costly function and gradient evaluations, improving overall efficiency. Each preconditioned step now respects the underlying physics: currents that strongly influence the fields are adjusted conservatively, while weakly coupled currents are still allowed to evolve meaningfully. Diagonal scaling replaces an overly cautious, one-size-fits-all step size η with a locally calibrated step for each basis function—dramatically accelerating convergence by exploiting the natural curvature variations inherent in metasurface designs.

Unit-flux scaling and Hessian-based diagonal preconditioning each tackle a different root cause of slow convergence in metasurface antenna design. First, normalizing every basis coefficient so that a unit value always represents the same total current removes the purely geometric bias that comes from uneven edge lengths. Once that imbalance is gone, Hessian-informed scaling steps in to recognize which currents really drive rapid changes in the cost and which barely move the needle, damping updates in highly sensitive directions and amplifying those in nearly flat regions. Together these two strategies form a powerful one-two punch: edge-length normalization lays a level playing field, and curvature-aware adjustment then steers each update by true physical sensitivity rather than mesh quirks. The result is an optimizer that can push boldly where it is safe and inch forward where it must, all without wasting time on needless backtracking or tiny, one-size-fits-all step sizes.

Chapter 4

Advanced Optimization Strategies Tailored to MTS Design

4.1 Problem Formulation and Challenges

As detailed in Section 2.3.1, the optimization problem is narrowed down to find the current coefficients minimizing the objective function.The optimal current is obtained by solving the unconstrained optimization problem

$$\mathbf{I}^* \;=\; \operatorname*{argmin}_{\mathbf{I}\in\mathcal{N}} f(\mathbf{I}), \tag{4.1}$$

where the cost function f is designed so that, for the minimizer $\mathbf{I}_{opt}$, the resulting metasurface antenna satisfies all the constraints defined in Section 2.2.2. In particular,

$$f(\mathbf{I}) \;=\; f_{\mathrm{rlz}}(\mathbf{I}) \;+\; f_{\mathrm{rad}}(\mathbf{I}), \tag{4.2}$$

with the two terms defined below.

Impedance and passivity term. The first contribution in (4.2) is

$$f_{\mathrm{rlz}}(\mathbf{I}) = \sum_{i=1}^{N_c} w_i^{\mathrm{act}}\,\rho_i^{\mathrm{act}}(\mathbf{I}) + \sum_{i=1}^{N_c} w_{Li}^{\mathrm{imp}}\,\rho_{Li}^{\mathrm{imp}}(\mathbf{I}) + \sum_{i=1}^{N_c} w_{Ui}^{\mathrm{imp}}\,\rho_{Ui}^{\mathrm{imp}}(\mathbf{I}) + \sum_{i=1}^{N_c} w_i^{\mathrm{scal}}\,\rho_i^{\mathrm{scal}}(\mathbf{I}) \tag{4.3}$$

ensuring that the current obtained from (4.1) yields a *scalar* impedance , a *passive and lossless* structure , and a reactance that remains within the technological bounds. See table 4.1.

Radiation mask term. The second contribution in (4.2),

$$f_{\mathrm{rad}}(\mathbf{I}) \;=\; w^{\mathrm{ref}}\,\rho^{\mathrm{ref}}(\mathbf{I}) + \sum_{j=1}^{N_c} w_j^{\mathrm{co,L}}\,\rho_j^{\mathrm{co,L}}(\mathbf{I}) + \sum_{j=1}^{N_c} w_j^{\mathrm{co,U}}\,\rho_j^{\mathrm{co,U}}(\mathbf{I}) + \sum_{j=1}^{N_c} w_j^{\mathrm{cx}}\,\rho_j^{\mathrm{cx}}(\mathbf{I}) + \sum_{j=1}^{N_c} w_j^{\mathrm{tot}}\,\rho_j^{\mathrm{tot}}(\mathbf{I}), \tag{4.4}$$

guarantees that the radiated field lies inside the prescribed masks (see Figure 5.1). See table 4.2.
Equations (4.3) and (4.4) together make $f(\mathbf{I})$ a scalar objective formed by the weighted sum of all relevant penalties.

The general algebraic expression of the objective function is given by

$$f(\mathbf{I}) = \sum_i q_i(\mathbf{I})\,s_i(\mathbf{I}) + \sum_i \mathrm{ramp}^2\big(t_i(\mathbf{I})\big), \tag{4.5}$$

where $q_i(\mathbf{I})$, $s_i(\mathbf{I})$, and $t_i(\mathbf{I})$ are multivariable quadratic functions of the current coefficients of the form:

$$q_i(\mathbf{I}) = \Re\left\{\mathbf{I}^H\mathbf{A}_i\mathbf{I} + \mathbf{I}^H\mathbf{b}_i + c_i\right\}, \tag{4.6}$$

where $\mathbf{A}_i \in \mathbb{C}^{N\times N}$ is a Hermitian matrix, $\mathbf{b}_i \in \mathbb{C}^N$, $c_i \in \mathbb{C}$, and $\mathbf{I} \in \mathbb{C}^N$ is the current vector. The definition of the objective function and constraints are explained in the section 2.3.1. The objective function (4.2) is minimized using Nonlinear Conjugate Gradient algorithm.

The current vector is updated iteratively according to

$$\mathbf{I}_{k+1} \;=\; \mathbf{I}_k \;+\; \alpha_k\,\mathbf{p}_k, \tag{4.7}$$

where $\mathbf{I}_k$ is the current iterate, $\mathbf{p}_k$ is the search (update) direction, and α_k is a scalar step length to be determined. α_k is obtained by a one-dimensional minimization of the objective function along $\mathbf{p}_k$ which is the so called line-search.

$$\alpha^* \;=\; \operatorname*{argmin}_{\alpha\in\mathbb{R}} f(\mathbf{I}_k + \alpha\,\mathbf{p}_k). \tag{4.8}$$

Because f becomes a piece-wise fourth-degree polynomial when expressed solely as a function of α (due to the ramp functions), its derivative is a cubic polynomial possessing two local maxima and a single local minimum. These stationary points can be located analytically, making the line-search relatively straightforward and inexpensive.

Although the fourth-order structure is computationally convenient, it renders the global *non-convex* optimization . Indeed, fixing a starting current $\mathbf{I}_k$ and a direction $\mathbf{p}_k$ turns f into a sum of quartic terms, which is generally non-convex. Although the Nonlinear Conjugate Gradient method with exact line search (NLCG + LS) has served as a reliable baseline for current-based metasurface synthesis, it suffers from several inherent limitations when applied to the non-convex and high-dimensional nature of our optimization landscape.

Nonlinear conjugate-gradient with exact line-search remains attractive for metasurface-antenna design because it requires minimal memory—storing only a handful of past search directions—and its deterministic update rules guarantee descent and affine-invariance on quadratic problems. However, in our highly ill-conditioned, nonconvex setting these strengths become weaknesses: the method can stall in poor basins when its fixed conjugacy relations fail to navigate complex loss landscapes, and its multi-point line searches dramatically increase per-iteration cost by repeatedly evaluating both the objective and its gradient along each search direction. Moreover, beyond the implicit curvature captured by conjugacy, NLCG does not adapt step sizes to local stiffness, so "stiff" directions force tiny global steps while "flat" directions remain quasi-stationary. In contrast, first-order methods like Adam eliminate line-search overhead by using momentum and adaptive per-parameter rates, and second-order schemes such as AdaHessian further employ lightweight Hessian-diagonal estimates to automatically rescale updates. These modern optimizers thus deliver far fewer iterations, lower wall-clock time, and reduced risk of entrapment compared to NLCG in challenging metasurface inverse-design problems.

For a strictly convex quadratic objective, the nonlinear conjugate–gradient (NLCG) method converges in[1]

$$O\big(\kappa\,\log(1/\varepsilon)\big)$$

iterations, where κ denotes the condition number and ε the target relative error. In the present ill-conditioned metasurface problem, we have

[1] See, e.g., [19, Eq. (8.12)].

$$\kappa \sim 10^6\text{–}10^8,$$

which implies *hundreds* of iterations to reach even moderate accuracy.[2]

Each NLCG step requires solving the one-dimensional stationarity condition

$$\phi'(\alpha) = 0,$$

along the search direction. Although the derivative is analytic, the procedure evaluates the cost functional and its derivative at several trial points. If a single cost evaluation (field solve + far-field penalty assembly) takes T_f, one line-search costs

$$O(m\,T_f), \quad m \approx 3\text{–}5,$$

yielding an overall per-iteration expense of

$$O((1+m)\,T_f),$$

and further slowing convergence.

Nonlinear conjugate-gradient with exact line-search remains attractive for metasurface-antenna design because its deterministic update rules guarantee descent and affine-invariance on quadratic problems. However, in highly ill-conditioned, nonconvex setting, this strength become weaknesses. The method can stall in poor basins when its fixed conjugacy relations fail to navigate complex loss landscapes, and its multi-point line searches dramatically increase per-iteration cost by repeatedly evaluating both the objective and its gradient along each search direction. Moreover, beyond the implicit curvature captured by conjugacy, NLCG does not adapt step sizes to local stiffness, so "stiff" directions force tiny global steps while "flat" directions remain quasi-stationary. In contrast, first-order methods like Adam eliminate line-search overhead by using momentum and adaptive per-parameter rates, and second-order schemes such as AdaHessian further employ lightweight Hessian-diagonal estimates to automatically rescale updates. These modern optimizers thus deliver far fewer iterations, lower wall-clock time, and reduced risk of entrapment compared to NLCG in challenging metasurface inverse-design problems.

4.2 First-Order: Adam Optimizer

The Adam optimizer sits at the intersection of momentum-based and adaptive-learning-rate methods, maintaining two running averages at each iteration: a first moment (the exponentially weighted mean of past gradients) and a second moment (the exponentially weighted mean of past squared gradients). By bias-correcting these estimates, Adam

[2]This range was confirmed both by theoretical near-resonance behavior of the continuous EFIE operator and by direct measurements using MATLAB's `condest`—a fast, iterative function for estimating a matrix's 1-norm condition number without explicit inversion—on discretized MoM matrices for representative metasurface meshes, which consistently yielded values between 10^6 and 10^8.

computes an update direction that both smooths noisy descent paths and automatically scales each parameter's step size according to its recent gradient variability. In effect, "flat" directions (with small or erratic gradients) are allowed larger moves, while "stiff" directions (with large, consistent gradients) are tamed to prevent overshoot—all without any costly line-search or manual learning-rate schedule.

In the inverse design of metasurface antennas, ADAM's self-tuning behavior offers clear benefits. The highly nonconvex, ill-conditioned landscape—where some current coefficients couple strongly to the fields and others barely at all—naturally produces gradient magnitudes spanning many orders of magnitude. ADAM's per-parameter scaling automatically compensates for these discrepancies, making early iterations both stable and aggressive, and reducing the need for painstaking step-size tuning. By eliminating line searches and gracefully handling noisy penalty ramps (from far-field mask constraints and reactive-bounds enforcement), ADAM accelerates convergence on large, finely discretized metasurface grids, often slashing iteration counts by an order of magnitude compared to conjugate-gradient methods and providing a robust foundation for further curvature-aware refinements like those in ADAHESSIAN.

4.2.1 Algorithm Overview and Update Rules

Adam tracks two exponential moving averages for each current coefficient. The first moment, $\mathbf{m}_k$, accumulates a decaying history of recent gradients and provides a momentum effect that smooths out noisy fluctuations. The second moment, $\mathbf{v}_k$, accumulates a decaying history of squared gradients and estimates the variance in each direction. To correct the inherent bias from initializing both averages at zero, Adam computes

$$\widehat{\mathbf{m}}_{k+1} = \frac{\mathbf{m}_{k+1}}{1-\beta_1^{k+1}}, \quad \widehat{\mathbf{v}}_{k+1} = \frac{\mathbf{v}_{k+1}}{1-\beta_2^{k+1}}.$$

The actual parameter update then divides the bias-corrected momentum by the square root of the bias-corrected variance (plus a small ε for numerical stability), yielding

$$\mathbf{I}_{k+1} = \mathbf{I}_k - \alpha \frac{\widehat{\mathbf{m}}_{k+1}}{\sqrt{\widehat{\mathbf{v}}_{k+1}} + \varepsilon},$$

so that each coefficient receives its own adaptive step size.

Adam offers several key advantages in the context of metasurface optimization. First, it requires only a single gradient evaluation per iteration, avoiding costly line searches and reducing the computational burden compared to NLCG. Second, Adam applies automatic per-parameter scaling: parameters in stiff directions (with consistently large gradients) receive smaller steps, while those in flat directions (with small or erratic gradients) receive larger ones. This property is essential when current-basis sensitivities vary by orders of magnitude. Finally, Adam's momentum averaging smooths out oscillations introduced by non-smooth penalties such as squared-ramp far-field masks, improving stability in rugged, nonconvex regions.

Algorithm 2 Adam Optimizer

Require: Initial guess $\mathbf{I}_0$, learning rate α, decay rates $0 < \beta_1 < 1$, $0 < \beta_2 < 1$, stability constant ε

Ensure: Optimized current $\mathbf{I}^\star$

1: Initialize first moment $\mathbf{m}_0 \leftarrow \mathbf{0}$, second moment $\mathbf{v}_0 \leftarrow \mathbf{0}$
2: **for** $k = 0$ to $K_{\max} - 1$ **do**
3: Compute gradient: $\mathbf{g}_k \leftarrow \nabla f(\mathbf{I}_k)$
4: Update first moment:

$$\mathbf{m}_{k+1} \;=\; \beta_1\,\mathbf{m}_k \;+\; (1-\beta_1)\,\mathbf{g}_k$$

5: Update second moment:

$$\mathbf{v}_{k+1} \;=\; \beta_2\,\mathbf{v}_k \;+\; (1-\beta_2)\,\mathbf{g}_k^2$$

6: Compute bias-corrected moments:

$$\widehat{\mathbf{m}}_{k+1} = \frac{\mathbf{m}_{k+1}}{1-\beta_1^{k+1}} \quad , \quad \widehat{\mathbf{v}}_{k+1} = \frac{\mathbf{v}_{k+1}}{1-\beta_2^{k+1}}$$

7: Update parameters:

$$\mathbf{I}_{k+1} = \mathbf{I}_k \;-\; \alpha\,\frac{\widehat{\mathbf{m}}_{k+1}}{\sqrt{\widehat{\mathbf{v}}_{k+1}} + \varepsilon}$$

8: **end for**
9: **return** $\mathbf{I}^\star = \mathbf{I}_{K_{\max}}$

4.2.2 Adam : Choice of Hyperparameters

Adam has four primary hyperparameters indicated in table 4.3.

Table 4.3. Adam Hyperparameters

Hyperparameter	**Default**	**Role in Optimization**
α	10^{-3} (tuning is needed)	Global learning rate
β_1	0.9 (range $[0.8, 0.99]$)	First-moment decay
β_2	0.999 (tune 0.99–0.9999)	Second-moment decay
ε	10^{-8}	Numerical stabilizer

Global Learning Rate α is a single scalar that multiplies every parameter update and therefore controls the overall step size. If α is too large, Adam may overshoot steep regions of the cost—such as near mask–violation ramps or resonant-reactance transitions—and exhibit oscillations or outright divergence. If α is too small, progress along flat directions—like weakly coupled basis functions or inactive ramps—becomes vanishingly slow,

stretching convergence times out by orders of magnitude. In metasurface inverse design, where the cost landscape combines sharp cliffs and broad plateaus, starting at the conventional $\alpha = 10^{-3}$ often works well. From there, one can refine via a logarithmic grid search over $[10^{-4}, 10^{-2}]$, or apply a "warm-up" schedule (ramping α from a small value) followed by annealing to fine-tune convergence without sacrificing early stability.

First-Moment Decay β_1 governs how quickly the exponential moving average of past gradients "forgets" older information. Large values (0.9–0.99) impart strong momentum, smoothing high-frequency noise and carrying the iterate through narrow ravines. However, in highly non-stationary regimes—such as when squared-ramp penalties suddenly activate or deactivate—excessive momentum can "lock in" a poor direction and delay recovery. Reducing β_1 toward 0.8 makes the average more reactive to new gradient information but also more sensitive to erratic variations.

Second-Moment Decay β_2 controls how quickly the moving average of squared gradients adapts to new curvature. High values (0.999–0.9999) produce stable per-parameter scalings but may lag behind rapid changes in the loss landscape. Lowering β_2 to around 0.99 or 0.9 makes the scaling more agile and responsive, but at the cost of increased variance in the update magnitudes. Stabilizer ε, a small constant (typically $\varepsilon = 10^{-8}$) added to the denominator $\sqrt{\widehat{\mathbf{v}}_k} + \varepsilon$ ensures numerical stability by preventing division by zero. If ε is too small, then in regions with vanishing gradients the effective step size can become unbounded and destabilize updates. Conversely, if ε is too large, it acts as an artificial lower bound on the denominator, capping all updates and slowing convergence. In practice, ε may even be adapted dynamically—raised when $\widehat{\mathbf{v}}_k$ falls below a threshold, and lowered again once gradient activity resumes—to maintain well-conditioned step sizes across the wide curvature spectrum typical of metasurface inverse-design problems.

4.2.3 Convergence Behavior and Step-Size Control

Adam's success lies in two key mechanisms: **momentum** and **adaptive rescaling**. The momentum effect, captured by the bias-corrected first-moment estimate $\hat{m}_t$, acts as an exponential moving average of past gradients. By accumulating directional information over multiple steps, momentum smooths out high-frequency noise—such as that introduced by numerical errors in field solves or abrupt ramp activations—and builds "velocity" along consistently descending directions. This enables the optimizer to push through shallow local oscillations and maintain progress even when individual gradient evaluations fluctuate. Meanwhile, the second-moment estimate $\hat{v}_t$ tracks the uncentered variance of the gradients. Dividing the step by $\sqrt{\hat{v}_t} + \epsilon$ automatically *tempers* updates in directions exhibiting large or volatile gradients, effectively acting as a per-coordinate trust-region that prevents overshooting in steep corners of the cost surface.

Over the course of optimization, as $\hat{v}_t$ accumulates squared gradients, the *effective step size*

$$\frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \tag{4.9}$$

Naturally *decays*, providing an implicit learning-rate annealing that transitions from large exploratory moves in the early stages to fine-grained adjustments near a minimum. This

dynamic balance—bold exploration when gradients are uniformly small, cautious refinement when gradients spike—allows Adam to navigate highly non-convex landscapes with stiff directions. In contrast, traditional NLCG with line-search may spend many iterations coping with ill-conditioning or may oscillate when faced with sudden curvature changes. By self-regulating its per-parameter step sizes, Adam remains robust in the presence of constraint-violating gradients (e.g., from squared-ramp penalties) and avoids wasting evaluations on overly conservative global step-lengths [12, 5].

In the inverse design of metasurface antennas, the cost combines far-field mask penalties—which can switch on and off sharply—and physical-realizability constraints that create steep local ridges in the objective. These features yield a gradient landscape that is both *rugged* (with many local minima from piecewise penalties) and *anisotropic* (exhibiting stiff coupling near feeds and flat behavior in shadowed regions). Adam's momentum effectively carries the design out of small, shallow basins caused by inactive masks, while its adaptive rescaling immediately dampens steps when a new mask ramp activates and produces large gradient spikes. Consequently, the optimizer can traverse narrow feasible corridors—such as those defining allowable sidelobe levels—without repeated line searches or manual step-size tuning. Empirical tests on metasurface problems show that Adam converges reliably to high-quality solutions in fewer wall-clock hours than NLCG, particularly when constraints tighten dynamically during optimization.

4.3 Second-Order: Ada-Hessian

At its core, AdaHessian augments the familiar momentum-and-variance framework of Adam with lightweight second-order (Hessian) information, yielding an optimizer that dynamically adapts not only to the scale of recent gradients but also to the local curvature of the objective. Rather than forming or inverting the full Hessian—which would be prohibitive for the hundreds of thousands of current coefficients in a metasurface design—AdaHessian uses a single randomized Hessian–vector product (via a Hutchinson estimator) per iteration to approximate the diagonal of the true Hessian. This diagonal is then smoothed both spatially (by averaging across small blocks of parameters that share physical locality) and temporally (via an exponential moving average), producing a per-parameter curvature estimate that automatically stretches or contracts the update step in each direction. AdaHessian offers several key advantages for metasurface antenna inverse design. It provides balanced steps in an ill-conditioned landscape, where the influence of basis coefficients on far-field constraints and boundary conditions may vary by several orders of magnitude. By adapting its update size according to local curvature, AdaHessian reduces steps in "stiff" directions (with large second derivatives) while allowing larger progress in "flat" ones, avoiding the overshooting or stagnation typical of standard first-order methods. Its robustness to penalty ramps and reactive singularities is another critical benefit: the optimizer gracefully handles sharp "kinks" in the loss landscape—introduced by squared-ramp penalties and reactive bounds—thanks to momentum-based smoothing and a stable Hessian-diagonal approximation, thereby preventing stalls near non-critical saddles.

4.3.1 Algorithm Overview and Update Rules

AdaHessian extends the Adam optimizer by incorporating explicit curvature information into each update step. At every iteration t, it begins by computing the gradient $\mathbf{g}_t$ as usual. Rather than a single probe vector, we can employ a mini-batch of Rademacher vectors to build an accurate Hessian diagonal. Concretely, at iteration t we draw m independent Rademacher vectors $\{z^{(i)}\}_{i=1}^{n} v$, with

$$nv \approx 0.3\,N, \tag{4.10}$$

In our implementation, the full Hessian H is available to compute analytically (or in a sparse form), so we avoid any costly dense matrix–matrix operations by computing only the required Hessian–vector products. Concretely, at iteration t we draw $m \approx 0.3\,N$ independent Rademacher vectors $\{z^{(i)}\}_{i=1}^{m}$, where N is the number of degrees of freedom. For each probe vector $z^{(i)}$, we perform the structured matrix–vector multiply $H\,z^{(i)}$. The diagonal estimate is then formed as

$$d_t \;\approx\; \frac{1}{m}\sum_{i=1}^{m} z^{(i)} \odot \big(H\,z^{(i)}\big), \tag{4.11}$$

which converges to $\mathrm{diag}(H)$ with high precision once m reaches roughly 30% of N. Empirically, this proportional sampling reproduces the true Hessian diagonal to within numerical tolerance, capturing essential curvature information without inflating the per-iteration cost.

In practice, the raw Hutchinson estimate d_t of the Hessian diagonal can be quite noisy, both because each probe only captures partial information and because nearby parameters often share similar curvature. To suppress this noise and improve stability, AdaHessian filters d_t in two ways. First, it performs **spatial smoothing** in which the entries of d_t are averaged over small contiguous blocks of size b [3]. This yields a locally "flat" version $d_t^{(s)}$ whose values vary slowly across neighboring parameters, reflecting the fact that nearby elements typically experience similar second-derivative behavior. Second, it applies **temporal smoothing** via an exponential moving average: the smoothed diagonal $\overline{d}_{t+1}$ is updated as

$$\overline{d}_{t+1} = \beta_2\,\overline{d}_t \;+\; (1-\beta_2)\,\big(d_t^{(s)}\big)^2, \tag{4.12}$$

When we initialize the temporal average $\overline{d}_0$ to zero, the early estimates $\overline{d}_t$ are biased low simply because they have "seen" fewer terms of the true sequence $\big(d_0^{(s)}\big)^2, \big(d_1^{(s)}\big)^2, \ldots$. Dividing by the factor

$$1-\beta_2^{t+1} \tag{4.13}$$

corrects for that startup bias, guaranteeing that

$$\widehat{d}_{t+1} = \frac{\overline{d}_{t+1}}{1-\beta_2^{t+1}} \tag{4.14}$$

[3] e.g., grouping basis coefficients that occupy the same physical patch of the metasurface

is an unbiased estimate of the exponentially weighted average of $\left(d_i^{(s)}\right)^2$.

By first smoothing each raw diagonal estimate d_t across local spatial blocks of size b, and then applying a bias-corrected exponential moving average in time, AdaHessian yields a Hessian-diagonal approximation that is both **locally coherent**—reflecting the fact that neighboring basis functions share similar curvature—and **adaptively responsive** to genuine changes in the function landscape as the optimization proceeds.

Under our settings, we skip both temporal and spatial averaging and feed the raw Hutchinson estimate d_t directly into the update. In other words, at each iteration we form equation 4.11 and use those per-parameter curvature estimates "as is" in the adaptive step-sizes.

Empirically we found that for our metasurface discretizations this avoids the extra cost and delay of any explicit filtering yet still delivers stable, well-conditioned updates.

At the same time that AdaHessian gathers curvature information, it also tracks a running "momentum" of the gradients exactly as Adam does. Concretely, the first-moment vector $\mathbf{m}_{t+1}$ blends the previous momentum $\mathbf{m}_t$ with the current gradient $\mathbf{g}_t$ via

$$\mathbf{m}_{t+1} = \beta_1 \, \mathbf{m}_t + (1 - \beta_1) \, \mathbf{g}_t, \tag{4.15}$$

where β_1 (typically 0.9) controls how quickly old gradient information is "forgotten." Because $\mathbf{m}_0$ starts at zero, the early $\mathbf{m}_{t+1}$ values are biased toward zero. To correct for this startup bias, AdaHessian divides by $\left(1 - \beta_1^{t+1}\right)$, yielding the unbiased estimate.

$$\widehat{\mathbf{m}}_{t+1} = \frac{\mathbf{m}_{t+1}}{1 - \beta_1^{t+1}}. \tag{4.16}$$

This bias-corrected momentum $\widehat{\mathbf{m}}_{t+1}$ then serves as the numerator in the curvature-scaled update, ensuring that the optimizer benefits from smooth, low-variance descent directions from the very first iteration.

The final parameter update is given by

$$\Delta \mathbf{I} = -\alpha \, \frac{\hat{\mathbf{m}}_{t+1}}{\left(\hat{\mathbf{d}}_{t+1}\right)^{k/2} + \varepsilon}, \tag{4.17}$$

where α is the global learning rate, ε is a small stabilizer to prevent division by zero, and the exponent $k \in [0,1]$ controls the degree of curvature influence. Setting $k = 0$ recovers a momentum-like update, while $k = 1$ approximates Newton's method.

This architecture yields several key benefits for metasurface antenna inverse design. First, adaptive curvature scaling automatically balances stiff and flat directions in the ill-conditioned optimization landscape, enabling AdaHessian to make near-Newton updates without costly matrix factorizations. Second, spatial averaging reinforces physical coherence. Nearby metasurface elements that experience similar boundary conditions produce harmonized curvature estimates, promoting stability across the grid. Finally, because AdaHessian requires only a few Hessian-vector product with respect to the full matrix-matrix product per iteration, its total per-step complexity remains reasonable. This efficiency makes it well-suited to large, high-resolution metasurface synthesis problems where full second-order methods would be computationally intractable.

4.3.2 AdaHessian : Choice of Hyperparameters

AdaHessian's behavior hinges on seven key hyperparameters as summarized in Table 4.4.
The global learning rate α uniformly scales each update, balancing convergence speed against stability; the first-moment decay rate β_1 controls how aggressively past gradients are retained as momentum—higher β_1 smooths noise but slows responsiveness, while lower β_1 adapts more quickly to changing descent directions; the Hessian-power exponent $k \in [0,1]$ interpolates between first-order Adam-style steps ($k = 0$) and full Newton-style curvature correction ($k = 1$); the spatial block size b controls how many neighboring parameters share a single curvature estimate, smoothing out local noise; the number of probes $nv \approx 0.3\,N$ determines how many randomized Hessian–vector products are averaged to approximate the true diagonal; the decay rate β_2 sets the temporal inertia for the exponential moving average of squared, block-averaged curvature, filtering out rapid fluctuations; and the stability constant ε prevents division by zero in very low-curvature regions, capping the maximum per-parameter step to avoid runaway updates.

Algorithm 3 AdaHessian Optimizer (Raw Diagonal)

Require: Initial guess $\mathbf{I}_0$, base learning rate α, decay rate $0 < \beta_1 < 1$, Hessian power k, number of probes m, stability constant ε
Ensure: Optimized current $\mathbf{I}^\star$

Initialize momentum $\mathbf{m}_0 \leftarrow \mathbf{0}$
for $t = 0$ to $T - 1$ **do**
  Draw m Rademacher probes $z^{(i)} \sim \{\pm 1\}^N$
  Compute raw Hessian diagonal estimate:

$$\mathbf{d}_t \;=\; \frac{1}{m}\sum_{i=1}^{m} z^{(i)} \odot \left(H\, z^{(i)}\right)$$

  Compute gradient $\mathbf{g}_t \leftarrow \nabla f(\mathbf{I}_t)$
  Update first moment (momentum):

$$\mathbf{m}_{t+1} \;=\; \beta_1\, \mathbf{m}_t \;+\; (1-\beta_1)\, \mathbf{g}_t$$

  Correct bias in momentum:

$$\widehat{\mathbf{m}}_{t+1} = \frac{\mathbf{m}_{t+1}}{1-\beta_1^{t+1}}$$

  Update parameters with raw curvature scaling:

$$\mathbf{I}_{t+1} = \mathbf{I}_t \;-\; \alpha\, \frac{\widehat{\mathbf{m}}_{t+1}}{(\mathbf{d}_t)^{\,k/2} + \varepsilon}$$

end for
return $\mathbf{I}^\star = \mathbf{I}_T$

4.3.3 Hyperparameter Tuning

Because metasurface-antenna designs span a wide variety of geometries, frequency bands, and far-field mask specifications, no single set of optimizer hyperparameters is universally optimal. Systematic tuning is therefore essential: different structures demand different values of α, $\beta_{1,2}$, k, b, nv, and ε to achieve the fastest and most reliable convergence.

To minimize manual effort and ensure robust performance across varied metasurface structures, it is invaluable to embed automated hyperparameter search into the workflow:

Bayesian Optimization. This approach constructs a probabilistic surrogate model—often a Gaussian process or Tree-structured Parzen Estimator—that predicts the optimizer's convergence metric (for example, final objective value or total iteration count) as a function of the hyperparameters. At each iteration, it uses an *acquisition function* (such as Expected Improvement) to balance exploring poorly understood regions of the hyperparameter space against exploiting areas known to perform well. By iteratively updating this model with real run results, Bayesian optimization typically finds near-optimal settings in far fewer trials than grid or random search.

Population-Based Training (PBT). PBT maintains a cohort of parallel training runs, each with its own hyperparameter configuration. Periodically—say every few hundred iterations—it evaluates all members, replaces the worst performers with perturbed clones of the best, and optionally mutates their hyperparameters (for instance, increasing α by 10% or toggling k). This *evolutionary* process allows both the model state (the current vector) and the hyperparameters to co-adapt, quickly driving the entire population toward robust regions of the search space while continuously refining promising candidates.

Hypergradient Methods. Here, one treats hyperparameters themselves (such as α, β_1, β_2, k) as differentiable variables in an outer optimization loop. By unrolling a short horizon of parameter updates and computing the gradient of the final loss with respect to these hyperparameters, one can apply standard gradient-descent or Adam updates to them directly. Though more complex to implement—requiring careful truncation or checkpointing to manage memory—hypergradient techniques can adapt learning rates and decay schedules on the fly, yielding finely tuned hyperparameter trajectories that evolve alongside the current vector.

By layering these automated strategies—first isolating α, then refining decay rates, and finally tuning curvature-specific parameters—practitioners can swiftly discover hyperparameter configurations that generalize across different metasurface geometries, frequency bands, and far-field mask constraints, all while dramatically reducing the need for manual experimentation.

Penalty Term	Purpose	Definition / Equation
Active Power Penalty	Enforces reactive behavior (no net real power flow on cell i).	$P_i = \frac{1}{2}\Re\langle E, J\rangle_{S_i}$ $\rho_i^{\text{act}}(\mathbf{I}) = P_i^2$
Lower-Bound Reactance	Ensures that $\Im\{Z_i\} \geq X_{L,i}$ by penalizing insufficient reactive power.	$Q_i = \frac{1}{2}\Im\langle E, J\rangle_{S_i}$ $J_i = \langle J, J\rangle_{S_i}$ $\Psi_{L,i}^{\text{imp}} = X_{L,i}J_i - Q_i$ $\rho_{L,i}^{\text{imp}}(\mathbf{I}) = \left[\max(\Psi_{L,i}^{\text{imp}}, 0)\right]^2$
Upper-Bound Reactance	Prevents reactance from exceeding $X_{U,i}$.	$\Psi_{U,i}^{\text{imp}} = Q_i - X_{U,i}J_i$ $\rho_{U,i}^{\text{imp}}(\mathbf{I}) = \left[\max(\Psi_{U,i}^{\text{imp}}, 0)\right]^2$
Scaling Penalty	Prevents mismatch between $\lvert E\rvert^2$ and $\lvert J\rvert^2$ magnitudes on each cell.	$E_i = \langle E, E\rangle_{S_i}$ $\rho_i^{\text{scal}}(\mathbf{I}) = E_iJ_i - (P_i^2 + Q_i^2)$

Table 4.1. Summary of radiation penalties applied in the power objective $f_{\text{rlz}}(\mathbf{I})$, each promoting physical consistency or bounding key quantities.

Penalty Term	Purpose	Definition / Equation
Reference-Pattern	Ensures that peak gain in the reference direction $\hat{r}_{\text{ref}}$ meets or exceeds target G_{ref}.	$F_{\text{ref}}(\hat{r}_{\text{ref}}) = \text{gain in direction } \hat{r}_{\text{ref}}$ $\rho^{\text{ref}}(\mathbf{I}) = \left[\max\left(G_{\text{ref}} - F_{\text{ref}}(\hat{r}_{\text{ref}}),\, 0\right)\right]^2$
Co-Polar Lower Mask	Enforces minimum co-pol gain $M^{\text{L}}_{\text{co},j}$ in direction $\hat{r}_j$ along polarization vector p_j.	$F_{\text{co}}(\hat{r}_j) = \left\lvert E(\hat{r}_j) \cdot p_j^* \right\rvert^2$ $\rho_j^{\text{co,L}}(\mathbf{I}) = \left[\max\left(M^{\text{L}}_{\text{co},j} - F_{\text{co}}(\hat{r}_j),\, 0\right)\right]^2$
Co-Polar Upper Mask	Limits co-pol sidelobes to maximum $M^{\text{U}}_{\text{co},j}$ in direction $\hat{r}_j$.	$F_{\text{co}}(\hat{r}_j)$ same as above $\rho_j^{\text{co,U}}(\mathbf{I}) = \left[\max\left(F_{\text{co}}(\hat{r}_j) - M^{\text{U}}_{\text{co},j},\, 0\right)\right]^2$
Cross-Polar Mask	Suppresses cross-pol leakage below mask $M^{\text{U}}_{\text{cx},j}$.	$F_{\text{cx}}(\hat{r}_j) = \left\lvert E(\hat{r}_j) \cdot x_j^* \right\rvert^2$ $\rho_j^{\text{cx}}(\mathbf{I}) = \left[\max\left(F_{\text{cx}}(\hat{r}_j) - M^{\text{U}}_{\text{cx},j},\, 0\right)\right]^2$
Total-Gain Upper Mask	Ensures combined co- and cross-pol gain stays under mask $M^{\text{U}}_{\text{tot},j}$.	$F_{\text{tot}}(\hat{r}_j) = F_{\text{co}}(\hat{r}_j) + F_{\text{cx}}(\hat{r}_j)$ $\rho_j^{\text{tot}}(\mathbf{I}) = \left[\max\left(F_{\text{tot}}(\hat{r}_j) - M^{\text{U}}_{\text{tot},j},\, 0\right)\right]^2$

Table 4.2. Summary of radiation penalties applied in the far-field objective $f_{\text{rad}}(\mathbf{I})$, each enforcing directional gain requirements.

Table 4.4. AdaHessian Hyperparameters

Hyperparameter	**Default / Range**	**Role**
α	10^{-3} (tuning is needed)	Global learning rate
β_1	0.9 (range $[0.8, 0.99]$)	First-moment (momentum) decay
k	0.5 (range $[0, 1]$)	Curvature-scaling exponent
nv	$\approx 0.3\,N$	Number of Hessian-vector probes
β_2	0.999 (tune 0.99–0.9999)	Temporal smoothing decay
ε	10^{-8} (optional 10^{-6})	Numerical stability constant

Chapter 5

Results and Analysis

5.1 Simulation Setup

To initialize the optimization, we first solve the problem using only the far-field mask constraints. In this stage, the only active penalty terms are those listed in Table 4.2. This far-field–only initialization provides a well-conditioned starting point that satisfies the radiation specifications before introducing the full set of physical realizability penalties.

The specifications of the masks for the radiation pattern are present in the Figure 5.1 .

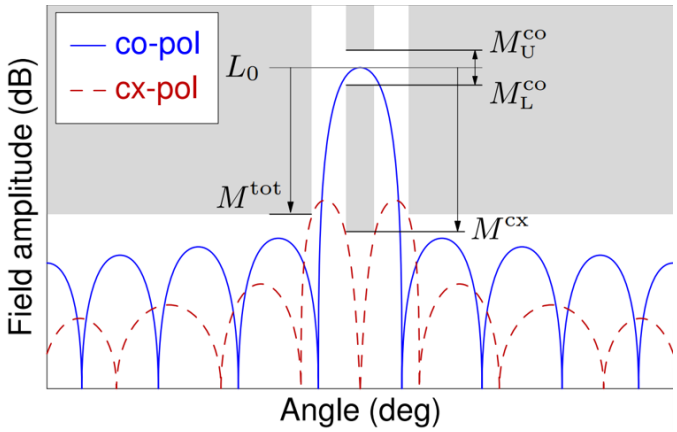

Figure 5.1. Masks for radiation pattern

Table 5.1. Far-Field Mask Specifications

Specification	**Value (dB rel. to L_0)**
Co-polar upper bound, M_U^{co}	$+3$
Co-polar lower bound, M_L^{co}	-5
Cross-polar upper bound, M_U^{cx}	-20
Total-gain upper bound, M_U^{tot}	-25

For the pencil-beam target, we partition the elevation angle θ into several regions to enforce our far-field and polarization constraints. We define the main-beam aperture as

$$|\theta| \leq 6^\circ,$$

over which we apply both the co-polarization mask M_{co} and the cross-polarization mask M_{cx} to ensure the desired polarization purity. Outside the main beam, for

$$20^\circ < |\theta| \leq 90^\circ,$$

we activate the total-gain upper mask M_{tot} to keep sidelobe levels below the prescribed threshold. Additionally, we specify a narrow reference window

$$|\theta| \leq 1^\circ$$

via the reference mask M_{ref}, which serves as a calibration zone for phase and magnitude alignment. By combining these angular masks, we achieve a narrow, high-purity pencil beam with suppressed side lobes and a well-defined reference region for robust performance. The polarization under test is circular.

Table 5.2. Angular Mask Regions for Pencil-Beam Design

Angular Region	**Active Masks**	**Purpose**
$\lvert\theta\rvert \leq 1^\circ$	M_{ref}	Reference window for phase/magnitude calibration
$\lvert\theta\rvert \leq 6^\circ$	M_{co}^{L}, M_{cx}^{U}	Enforce co- and cross-polarization purity over the main beam
$6^\circ < \lvert\theta\rvert \leq 20^\circ$	—	Transition region (no mask)
$20^\circ < \lvert\theta\rvert \leq 90^\circ$	$M_{\text{tot}}^{\text{U}}$	Limit total sidelobe level outside main beam

5.2 Mesh and Geometry Definition

The optimizations are done across various structure, the results in this section are reported for the the two structure shown in Figure 5.2, and Figure 5.3.

The parameters of each geometry is indicated at Table 5.3.

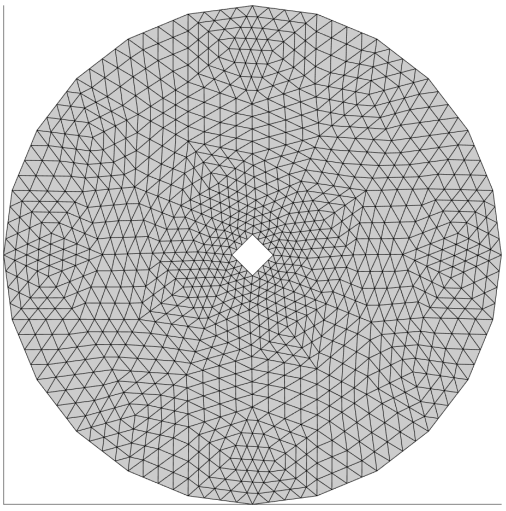

Figure 5.2. Small Scale mesh

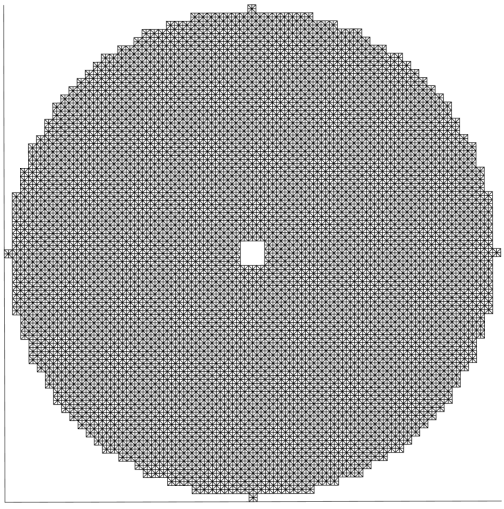

Figure 5.3. Separated Square Mesh

The test structure here "Separated Square" consists of a grid of individual square patches. RWG basis functions are defined on each square, while no RWGs span the gaps between

Table 5.3. Geometric parameters of small-scale, and separated square metasurface antenna

Parameter	**Value for small-scale**	**Value for separated-square**
Operating frequency, f	23 GHz	23 GHz
Relative permittivity, ε_r	3	3.34
Relative permeability, μ_r	1	1
Substrate thickness, h	6.35×10^{-4} m	5.08×10^{-4} m
Maximum radius, R_{max}	3λ	3λ
Maximum radius, R_{min}	$\lambda/4$	$\lambda/6$
Maximum mesh element size	$\lambda/6$	$\lambda/10$
Number of mesh nodes	1176	25308
Number of mesh cells (triangles)	2240	22496
Number of RWG basis functions	3304	22496

adjacent patches.

5.2.1 Multi-Level Optimization in MR-RWG Basis

In this section, we use the small scale antenna for testing the multi-level optimization using MR-RWG basis. We implement multiresolution by performing a sequence of nested optimizations on progressively finer meshes. First, we optimize the current distribution on a very coarse mesh, capturing the dominant, large-scale features. We then interpolate those coarse-mesh currents as the initial guess for the next level of refinement, run a second optimization there to resolve mid-scale details, and finally carry that solution forward into the full–pixel-level mesh for a final, high-resolution optimization. At each stage the solver focuses only on the new degrees of freedom introduced by refinement, so the large-scale behavior remains intact while small-scale adjustments are efficiently learned. This staged approach dramatically accelerates convergence and improves robustness compared to optimizing directly on the finest mesh.

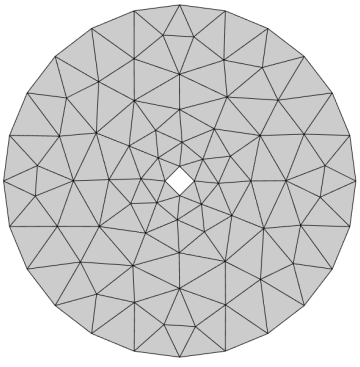

Figure 5.4. Coarse Mesh

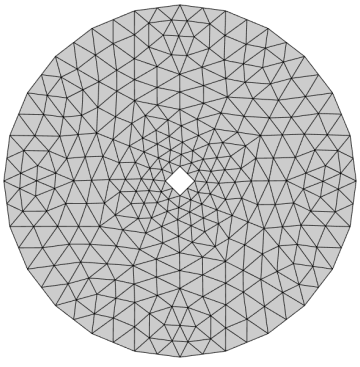

Figure 5.5. Mid-Level Mesh

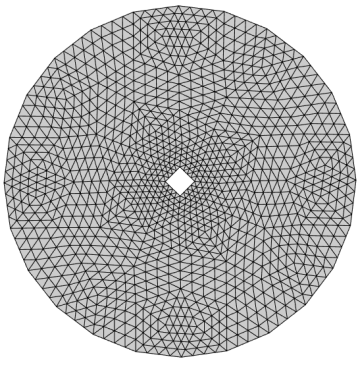

Figure 5.6. Fine Mesh

The mesh information for each level is summarized in Table 5.4.

Despite the theoretical advantages and a sparse basis-change matrix that offers computational benefits, practical results in the context of inverse design of metasurface antennas, showed limited improvements in convergence. We attribute this to a fundamental mismatch between the multiresolution structure of the current representation and the fine-scale enforcement of physical constraints. Specifically, although the surface current is expanded hierarchically, the key constraints—such as local passivity, losslessness, and impedance bounds—are evaluated on the finest mesh. This effectively nullifies the smoothing and regularization benefits of coarser resolution levels, forcing the optimizer to respect high-resolution constraints even during early iterations on the coarser mesh. As a result, the optimizer cannot fully exploit the coarse-to-fine convergence acceleration that multiresolution methods typically enable in unconstrained problems.

One could further improve this approach by imposing distinct constraint sets at each resolution level, which would necessitate reformulating the underlying optimization problem in multiresolution coordinates. Because our present focus is on accelerating convergence within the existing formulation, we leave that extension to future work.

Table 5.4. Geometric parameters of mesh levels

Parameter	**Coarse Mesh**	**Mid-Level Mesh**	**Fine Mesh**
Number of mesh nodes	84	308	1176
Number of mesh cells (triangles)	140	560	2240
Number of RWG basis functions	196	812	3304

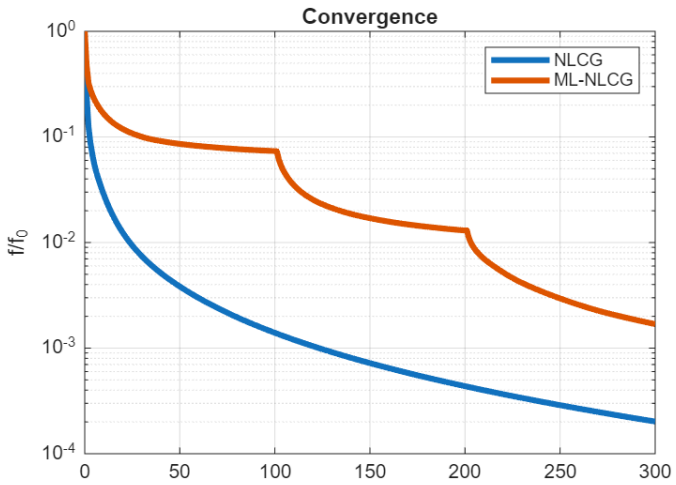

Figure 5.7. Convergence Comparison : NLCG, ML-NLCG

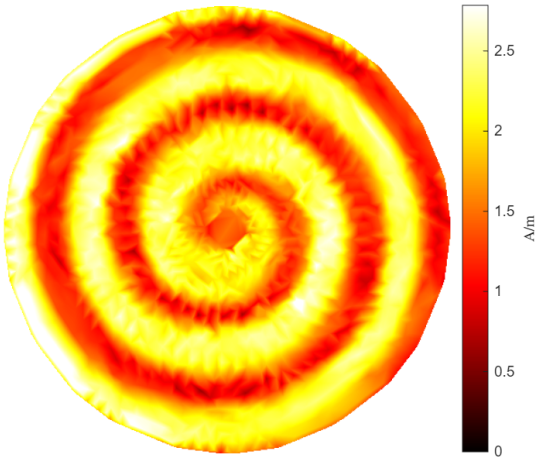

Figure 5.8. Current Optimized: NLCG

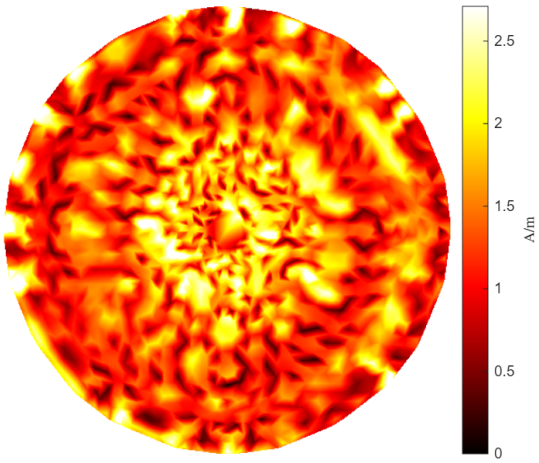

Figure 5.9. Current Optimized: ML-NLCG

5.2.2 Effect of Unit Edge-Flux Preconditioning on Convergence

This method is tested on different structure and globally results in a better and smoother convergence. In this section we presented and validated our unit-edge-flux preconditioning strategy, which normalizes each RWG coefficient by its supporting edge length so that every basis function contributes exactly one unit of total current flux across its edge.

NLCG runs the classic non-linear conjugate gradient optimizer while DP-NLCG runs the optimization on preconditioned space. DP refers to as Diagonl Preconditioning. We can see that as indicated in theory theory convergence is better in diagonal preconditioning satisfying Unite Edge Flux condition.

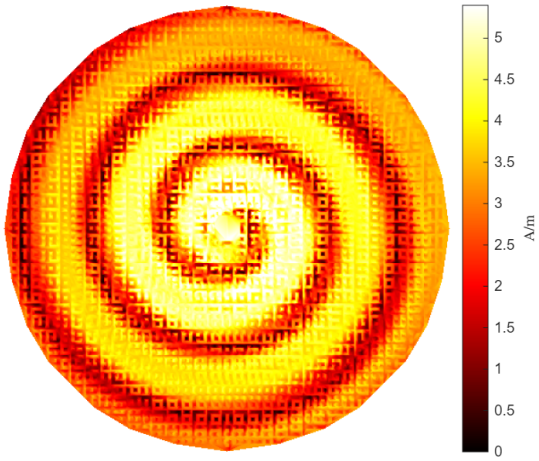

Figure 5.11. Current Optimized: NLCG

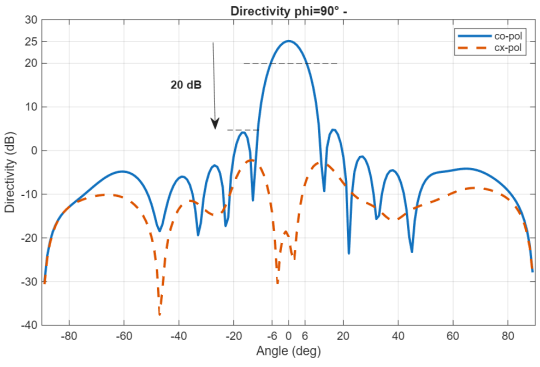

Figure 5.12. Directivity: NLCG

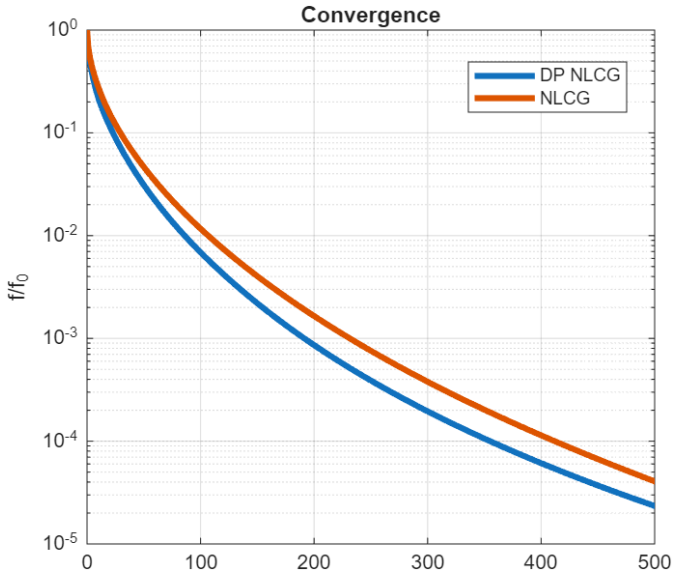

Figure 5.10. Convergance Comparison : NLCG, DP-NLCG

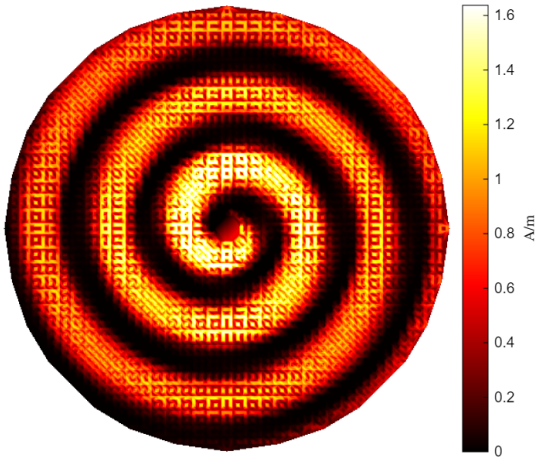

Figure 5.13. Current Optimized: DP-NLCG

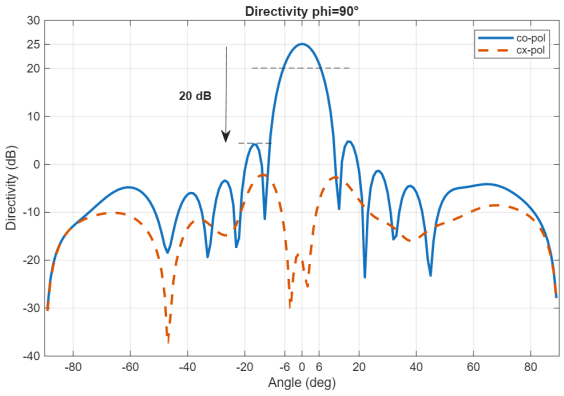

Figure 5.14. Directivity: DP-NLCG

The test results shows improvement in both realized optimization of the current and the convergence.

Under a common convergence threshold of 10^{-4}, DP–NLCG reaches the stopping criterion in approximately fifty fewer iterations than standard NLCG. In large-scale metasurface optimizations, this reduction equates to a significant savings in total CPU-time and computational cost.

5.2.3 MR-RWG Basis Change

In this test, the optimizer operates directly in the MR–RWG basis, which is expected to yield faster convergence.

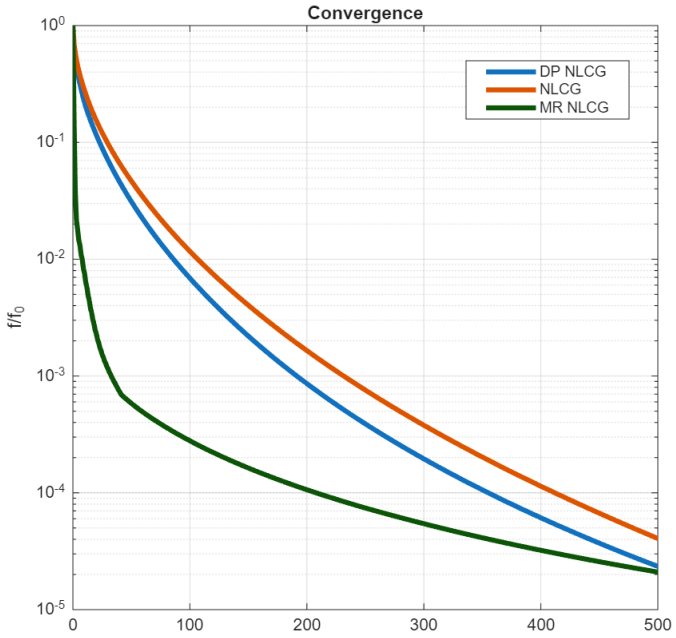

Figure 5.15. Convergence Comparison

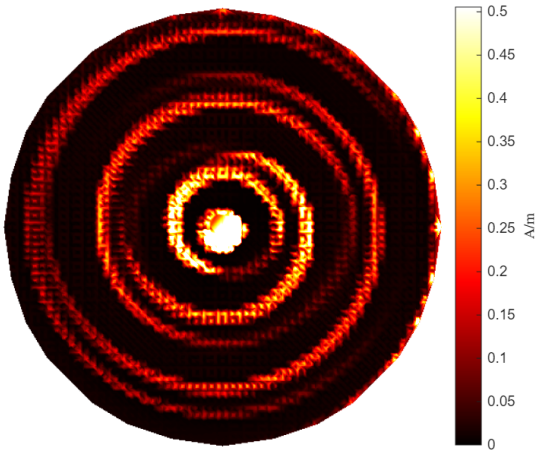

Figure 5.16. Current Optimized: MR-NLCG

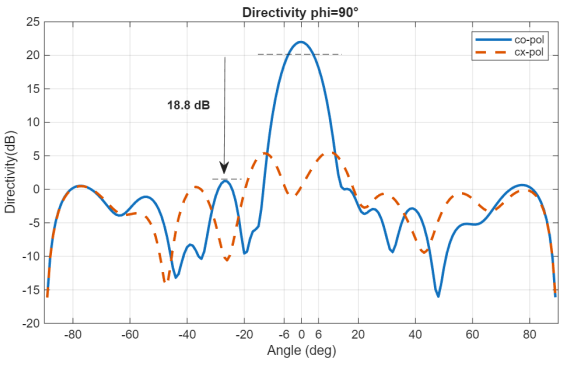

Figure 5.17. Directivity: MR-NLCG

Although the MR-RWG basis accelerates convergence in the early iterations, the resulting optimized current exhibits rapid coefficient fluctuations, making it impractical for

fabrication.

The key lesson is that convergence remains highly sensitive to the choice of initial guess, even though transforming into the MR-RWG space typically produces a smoother optimization trajectory. This dependency arises from the highly non-convex landscape intrinsic to metasurface inverse-design problems, where multiple local minima can trap the optimizer if not properly initialized. In practice, the effectiveness of both unit edge-flux preconditioning and MR–RWG basis transforms varies with the underlying geometry and mesh topology of a given metasurface.

5.3 Second Order Information

By injecting approximate second-order information into each update, the optimizer adapts its step sizes precisely to the local curvature, yielding two key benefits for our metasurface design problems. First, "stiff" directions, coordinates where the cost landscape curves sharply are automatically damped, preventing the optimizer from taking overly aggressive steps that would otherwise force a conservative global learning rate. Second, "flat" directions, coordinates with very gentle curvature are amplified, ensuring that even weakly sensitive parameters continue to make meaningful progress rather than stagnating. In practice, these curvature-scaled updates translate into considerably fewer line-search backtracks, more consistent per iteration cost reductions, and a markedly smoother convergence profile. Our Hutchinson-based diagonal Hessian estimates thus act as a low overhead yet powerful preconditioner.

5.3.1 Hutchinson's Based Diagonal Estimation

In large-scale problems such as inverse design of metasurface antennas, exploiting full second-order information is often infeasible due to the required dense matrix–matrix products. The Method-of-Moments impedance matrix can reach dimensions of tens of thousands, making explicit Hessian assembly and factorization prohibitively expensive. By contrast, Hutchinson's stochastic diagonal estimator replaces these operations with a small number of Hessian–vector products, incurring a per-iteration cost that grows only linearly with the number of probe vectors while still capturing essential curvature information.

For optimization, it suffices to capture the overall curvature trends rather than the exact Hessian diagonal. Consequently, we can use fewer probe vectors just enough to approximate these global variations effectively.

Figure 5.18 shows the accuracy of the diagonal estimation, and the Figure 5.19 represents how estimating the diagonal with different number of prob vectors can affect the convergence.

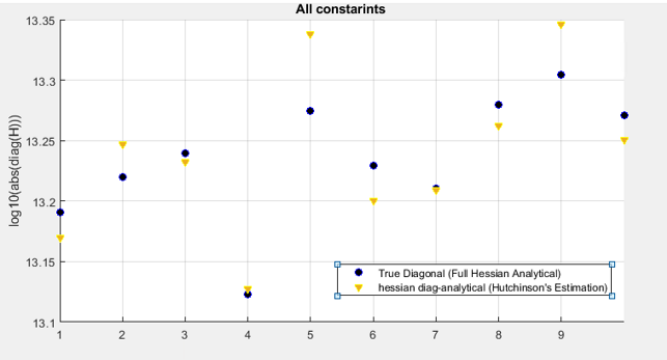

Figure 5.18. Diagonal Estimation

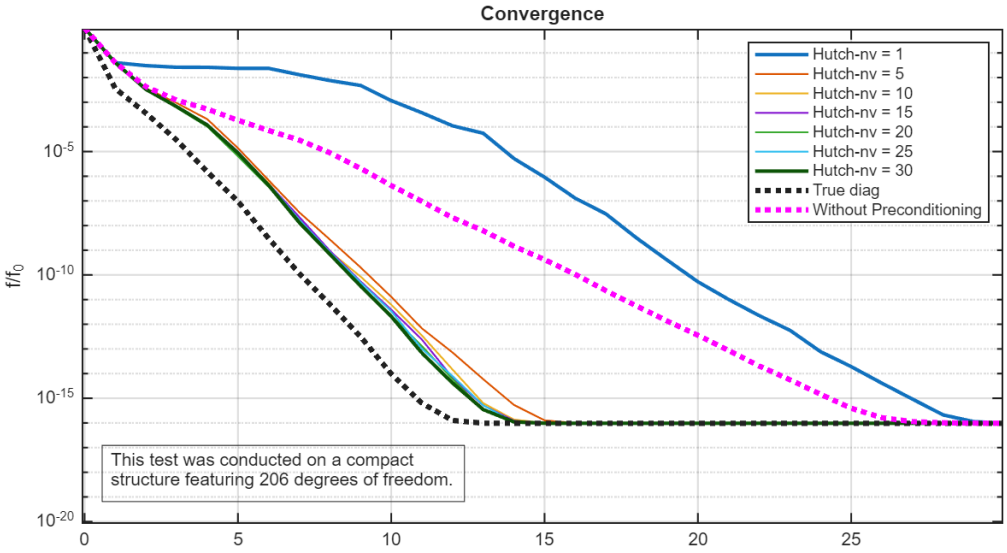

Figure 5.19. Convergence Comparison : NLCG vs Hutch-NLCG

5.4 New Optimizers

In this chapter, we also explore the potential of modern adaptive optimizers to accelerate and stabilize our metasurface-antenna inverse-design workflow. These methods dynamically adjust each parameter's step size based on local gradient and curvature information, offering a natural remedy for the highly nonconvex, scale-varying sensitivity landscape that hinders plain gradient descent. We implement two state-of-the-art algorithms within our MoM-based framework and benchmark them against our established schemes. By evaluating not only the final antenna performance but also metrics such as convergence rate, numerical stability, and robustness across multiple test geometries, we aim to identify

when—and to what extent—off-the-shelf adaptive optimizers can yield practical improvements for complex electromagnetic design problems.

5.4.1 Adam Optimizer

The test structure is Figure 5.3.

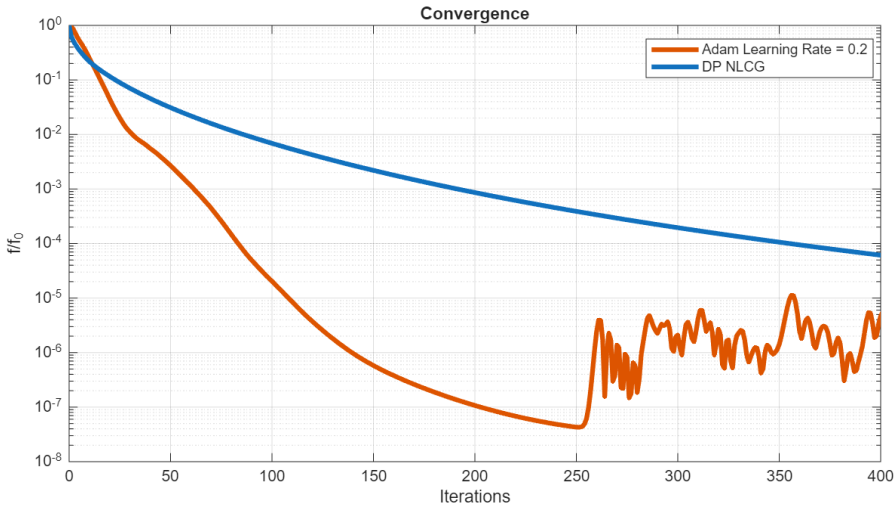

Figure 5.20. Convergence Comparison : NLCG vs Adam

Adam's adaptive, momentum-driven updates give it a remarkable ability to explore the optimization landscape broadly, automatically scaling each parameter's step size to negotiate both shallow and steep regions. This "global" search capability makes Adam far less sensitive to a poor initialization—it can accelerate through wide plateaus, skirt around shallow traps, and gradually home in on promising basins. By contrast, NLCG relies strictly on local gradient directions and line-searches, so if its starting point lies near a suboptimal valley its purely deterministic updates will often circle and stall there.

In practice, one can combine their strengths—first running a few Adam epochs to find a robust "warm" initialization, then switching to NLCG (or DP-NLCG) for rapid local convergence. Adam frees the optimizer from bad starting guesses, while NLCG then exploits high-quality curvature information for efficient final polishing.

5.4.2 AdaHessian

The green trace shows AdaHessian with a fixed learning rate of 0.2. In the very early iterations, AdaHessian descends more rapidly than the DP-NLCG baseline (blue), benefitting from its per-parameter curvature scaling. Unlike plain Adam (orange), which plunges quickly but then begins to oscillate wildly once it reaches very small residuals, AdaHessian remains stable throughout—no sudden spikes or stalls—even as it continues to make steady progress down toward 10^{-5}. In other words, AdaHessian combines more

aggressive early-stage descent than a first-order preconditioned CG with the long-run robustness of a curvature-aware method, avoiding both the sluggish tail of DP-NLCG and the late-stage jitter of Adam.

That said, its ultimate rate of decay here is slower than Adam's when Adam is well-tuned, suggesting that the choice of learning rate (and possibly the curvature exponent k) still has a significant impact. More in-depth investigations—especially into per-parameter step-size schedules and richer Hessian-diagonal estimation strategies—will be essential to unlock AdaHessian's full potential in our metasurface-antenna optimizations.

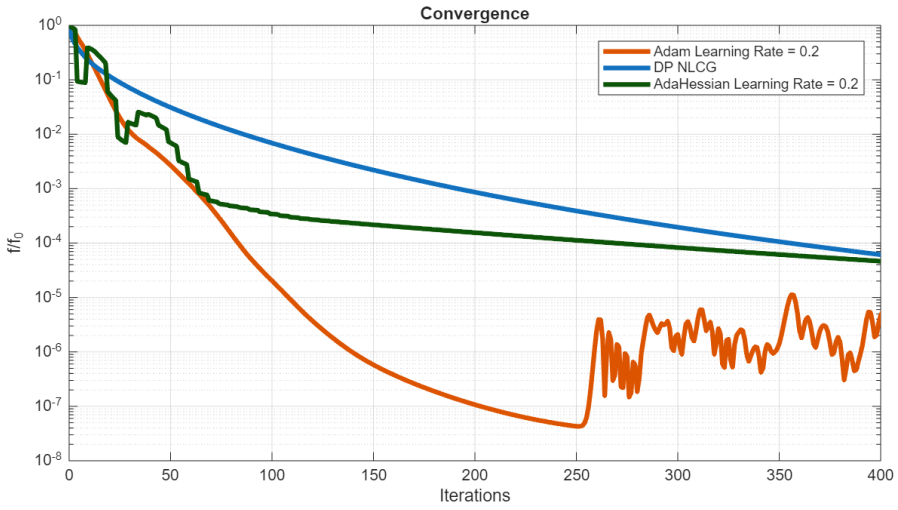

Figure 5.21. Convergence Comparison : NLCG vs Adam

5.5 Results of the Impedance Realization for DP-NLCG

The realized impedance confirms that unit-flux preconditioning is a reliable and effective enhancement for the optimization process.

Separated Square	iter	f	f/f0	gnorm	step	Total run time
NLCG	1000	2.4666e+04	1.2675e-06	3.79e+07	2.79e-06	19449 s
DP-NLCG	1000	1.6905e+04	1.827e-07	3.21e+04	3.30e-03	19062 s

Table 5.5. Comparison of optimization results NLCG, DP-NLCG

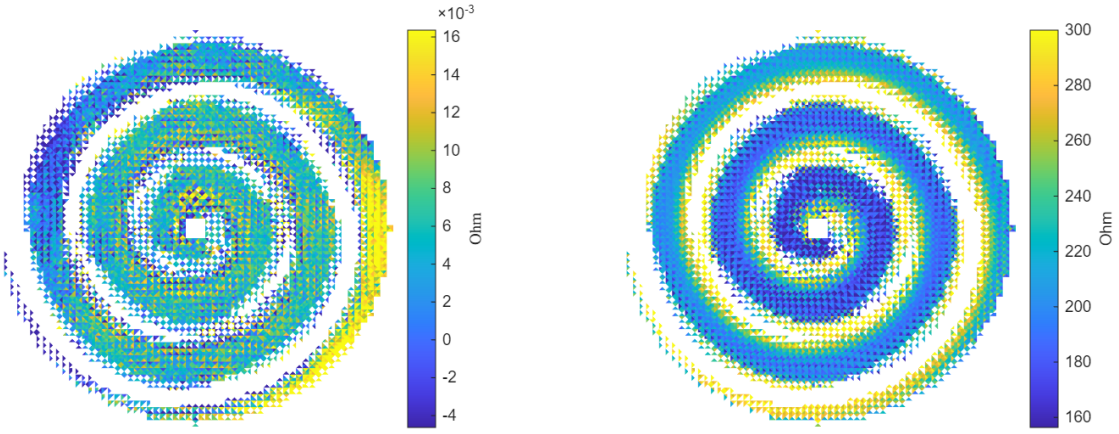

Figure 5.22. Realization of Impedance for DP-NLCG

Chapter 6

Conclusion and Future Works

This work focused on improving the speed and robustness of a fully automated, current-based inverse-design framework for 3D metasurface antennas. By modeling the unknowns as surface currents within a Method-of-Moments (MoM) formulation, the framework ensures passive, lossless, and fabrication-ready solutions while meeting far-field constraints.

Three numerical strategies were explored. First, unit-flux preconditioning reduced gradient sensitivity and significantly improved convergence. Second, a multiresolution RWG basis improved numerical behavior but showed limited practical benefit due to constraint enforcement at full resolution, and the resulting currents were not ideal for fabrication. Third, Hessian-diagonal estimation helped adjust optimization step sizes more effectively, though integrating it fully into the solver remains a challenge.

Adaptive optimizers (Adam and AdaHessian) were also evaluated. Adam offered fast early progress, while AdaHessian provided better stability at tighter tolerances. A hybrid approach combining both was found to be promising.

Looking ahead, future work should explore hybrid solvers, automated hyperparameter tuning, and multilevel constraint enforcement to better utilize MR–RWG structures. Overall, the results show that combining unit-flux scaling and second-order information can substantially accelerate metasurface inverse design, and further integration with adaptive, resolution-aware methods could bring the framework closer to practical, fabrication-ready implementations.

Appendix

I Antenna Generalities

I.1 Field Polarization

In most applications, electromagnetic field sources have a sinusoidal waveform. Therefore, in a linear, time-invariant medium the resulting fields are time-harmonic:

$$\mathbf{E}(\mathbf{r},t) = \Re\{\mathbf{E}(\mathbf{r})\, e^{j\omega t}\}, \tag{1}$$

$$\mathbf{H}(\mathbf{r},t) = \Re\{\mathbf{H}(\mathbf{r})\, e^{j\omega t}\}, \tag{2}$$

where $\mathbf{r}$ is position, ω the angular frequency, and $\mathbf{E}(\mathbf{r}), \mathbf{H}(\mathbf{r})$ are complex phasor fields. Focusing on $\mathbf{E}$, write

$$\mathbf{E} = E_x\, \hat{\mathbf{x}} + E_y\, \hat{\mathbf{y}} + E_z\, \hat{\mathbf{z}},$$

with $E_x, E_y, E_z \in \mathbb{C}$. Equivalently one may decompose

$$\mathbf{E} = \mathbf{E}' + j\, \mathbf{E}'',$$

where $\mathbf{E}', \mathbf{E}''$ are real "in-phase" and "quadrature" vectors. In general the tip of $\mathbf{E}(t)$ traces an ellipse, whose axes and sense (right- or left-handed) are determined by $\mathbf{E}'$ and $\mathbf{E}''$.

A polarization is

- *linear* if $\mathbf{E}' \times \mathbf{E}'' = 0$;
- *circular* if $\|\mathbf{E}'\| = \|\mathbf{E}''\|$ and $\mathbf{E}' \cdot \mathbf{E}'' = 0$.

We define the unit–polarization vector

$$\hat{\mathbf{p}} = \frac{\mathbf{E}}{\|\mathbf{E}\|} = \frac{\mathbf{E}' + j\, \mathbf{E}''}{\sqrt{\|\mathbf{E}'\|^2 + \|\mathbf{E}''\|^2}} = \hat{\mathbf{p}}' \; + \; j\, \hat{\mathbf{p}}''. \tag{A.1}$$

Most simple wire antennas (e.g. dipoles) radiate a linearly polarized field. Circular polarization is obtained by feeding two orthogonal elements with a 90° phase shift (e.g. turnstile).

The *co–* and *cross–* polarization components of an arbitrary field $\mathbf{E}$ (relative to nominal polarization $\hat{\mathbf{p}}$) are

$$E_{\text{co}} = \mathbf{E} \cdot \hat{\mathbf{p}}^*, \qquad E_{\text{cx}} = \mathbf{E} \cdot \hat{\mathbf{q}}^*, \tag{A.2}$$

where $\hat{\mathbf{q}}$ is a unit vector orthogonal to $\hat{\mathbf{p}}$.

I.2 Directivity, Gain, and Efficiency

The time-average Poynting vector in the far field is

$$\mathbf{S}(\mathbf{r},\theta,\phi) = \tfrac{1}{2}\Re\{\mathbf{E}\times\mathbf{H}^*\} = \frac{1}{2\eta_0}\,|\mathbf{E}|^2\,\hat{\mathbf{r}}, \tag{A.3}$$

since $\mathbf{H} = (1/\eta_0)\,\hat{\mathbf{r}}\times\mathbf{E}$ and η_0 is the free-space impedance.

The *directivity* pattern is

$$d(\theta,\phi) = \frac{S(r,\theta,\phi)}{S_{\mathrm{av}}(r)}, \quad S_{\mathrm{av}}(r) = \frac{1}{4\pi r^2}\int_{\mathrm{sphere}} S\,d\Sigma = \frac{P_{\mathrm{rad}}}{4\pi r^2}, \tag{A.4–A.5}$$

so

$$d(\theta,\phi) = \frac{S(r,\theta,\phi)}{P_{\mathrm{rad}}/(4\pi r^2)}. \tag{A.6}$$

The *maximum directivity* is

$$D = \max_{\theta,\phi} d(\theta,\phi). \tag{A.7}$$

Define the ohmic (radiation) efficiency

$$\eta = \frac{P_{\mathrm{rad}}}{P_{\mathrm{in}}} < 1, \tag{A.8}$$

then the *gain* pattern is

$$g(\theta,\phi) = \frac{S(r,\theta,\phi)}{P_{\mathrm{in}}/(4\pi r^2)} = \eta\,d(\theta,\phi), \tag{A.9}$$

and the *maximum gain*

$$G = \max_{\theta,\phi} g(\theta,\phi). \tag{A.10}$$

In receive mode, the *effective area* A_{eff} is defined by

$$A_{\mathrm{eff}} = \frac{P_{\mathrm{av}}}{S_{\mathrm{inc}}}, \tag{A.11}$$

assuming perfect polarization match, $|\hat{\mathbf{p}}_{\mathrm{tx}}\cdot\hat{\mathbf{p}}_{\mathrm{rx}}|^2 = 1$. The *aperture efficiency* for an aperture antenna of physical area A_{geom} is

$$\eta_{\mathrm{ap}} = \frac{A_{\mathrm{eff}}}{A_{\mathrm{geom}}}. \tag{A.12}$$

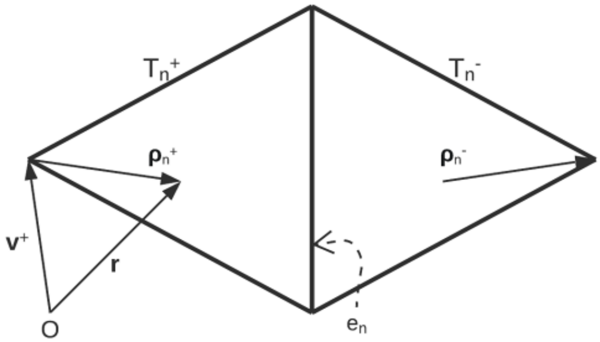

Figure 1. Graphical representation of the RWG basis function.

II RWG Basis Definition

When dealing with scattering or radiation problems in three-dimensional space, it is a very common practice to discretize the surface by dividing it into triangular cells. In this way, it is possible to model surfaces of arbitrary shape and size. In this scenario, one of the most common choices for addressing the problem is to use RWG basis functions, which are local basis functions defined on pairs of triangles that share an edge. Considering two adjacent triangles T_n^+ and T_n^- that share the common edge e_n of length ℓ_n, the RWG basis function $\mathbf{f}_n(\mathbf{r})$ associated with e_n is defined as

$$\mathbf{f}_n(\mathbf{r}) = \begin{cases} \dfrac{\ell_n}{2A_n^+}\,\rho_n^+(\mathbf{r}), & \mathbf{r} \in T_n^+, \\ -\dfrac{\ell_n}{2A_n^-}\,\rho_n^-(\mathbf{r}), & \mathbf{r} \in T_n^-, \\ 0, & \text{otherwise}, \end{cases}$$

where

- $\ell_n = |e_n|$ is the length of the common edge,
- $A_n^\pm$ are the areas of triangles $T_n^\pm$,
- $\rho_n^\pm(\mathbf{r}) = \mathbf{r} - \mathbf{r}_n^\pm$ is the vector from the free vertex $\mathbf{r}_n^\pm$ of triangle $T_n^\pm$ to the field point $\mathbf{r}$.

By construction, $\mathbf{f}_n(\mathbf{r})$ is divergence-conforming and ensures continuity of the normal component of surface current across the shared edge e_n.

The RWG basis functions are widely used in three-dimensional electromagnetic problems due to the following properties:

- The surface current has no component normal to boundary edges (edges not shared by two triangles).

- The normal component of the current across any internal edge is continuous.
- The tangential component of the current across any internal edge is discontinuous.
- The surface divergence of f_n is given by

$$\nabla_s \cdot f_n(\mathbf{r}) = \begin{cases} \dfrac{l_n}{A_n^+}, & \mathbf{r} \in T_n^+, \\ -\dfrac{l_n}{A_n^-}, & \mathbf{r} \in T_n^-, \\ 0, & \text{otherwise}, \end{cases}$$

  which makes the charge density constant on each triangle and ensures zero net charge on the pair of adjacent triangles T_n^+ and T_n^-.

III Surface Equivalence Theorem

The uniqueness theorem in electromagnetics states that the solution to a field problem, generated by sources within a region, is uniquely determined once the tangential component of the electric or magnetic field is specified on the boundary of that region. From this result follows the *surface equivalence theorem*: consider a homogeneous medium and a closed surface S enclosing all true sources $\mathbf{J}, \mathbf{M}$. Denote the interior of S by V_1 and the exterior by V_2. The actual sources in V_1 radiate fields $(\mathbf{E}_1, \mathbf{H}_1)$ everywhere.

The surface equivalence theorem asserts that one may replace the true sources by equivalent surface currents on S, provided the fields satisfy the boundary conditions across S. For arbitrary fields $(\mathbf{E}, \mathbf{H})$ in V_1 and the radiated fields $(\mathbf{E}_1, \mathbf{H}_1)$ in V_2, continuity of the tangential fields enforces

$$\begin{aligned} \hat{\mathbf{n}} \times (\mathbf{H}_1 - \mathbf{H}) &= \mathbf{J}_{\text{eq}}\,, \\ (\mathbf{E}_1 - \mathbf{E}) \times \hat{\mathbf{n}} &= \mathbf{M}_{\text{eq}}\,, \end{aligned}$$

where $\hat{\mathbf{n}}$ is the outward normal on S. These equivalent currents $\mathbf{J}_{\text{eq}}, \mathbf{M}_{\text{eq}}$ then radiate into the unbounded homogeneous region V_2, and the resulting fields can be computed by the usual radiation integrals.

A particularly convenient choice is to set the interior fields to zero, $\mathbf{E} = \mathbf{H} = \mathbf{0}$ in V_1. Equations (**??**)–(**??**) then simplify to

$$\begin{aligned} \hat{\mathbf{n}} \times \mathbf{H}_1 &= \mathbf{J}_{\text{eq}}\,, \\ \mathbf{E}_1 \times \hat{\mathbf{n}} &= \mathbf{M}_{\text{eq}}\,. \end{aligned}$$

Substituting these into the radiation integrals provides a complete solution for the fields in V_2, without regard to the interior region.

The equivalence theorem also applies when S is an actual closed surface separating two media. Let V_1 and V_2 be volumes filled with material parameters ε_1, μ_1 and ε_2, μ_2, respectively. In the unbounded region V_2, an electromagnetic field exists—either due to sources

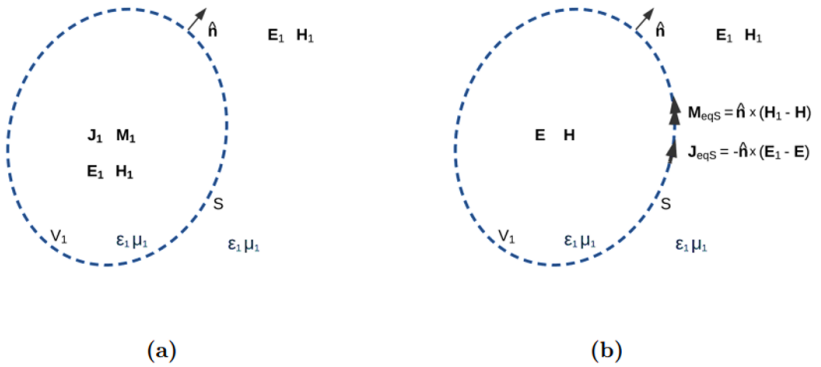

Figure 2. Surface equivalence theorem illustration: (a) actual and (b) equivalent problem.

in V_2 or an incident wave (e.g. a plane wave) in the absence of the obstacle in V_1. The total fields in V_2 are then

$$\begin{aligned}\mathbf{E}_{\text{tot}} &= \mathbf{E}_{\text{inc}} + \mathbf{E}_{\text{s}},\\ \mathbf{H}_{\text{tot}} &= \mathbf{H}_{\text{inc}} + \mathbf{H}_{\text{s}}.\end{aligned}$$

Applying the surface equivalence theorem, we may set the fields inside V_1 to zero and replace the actual medium in V_2 by the background medium. We then enforce equivalent surface currents on S so that the fields in V_2 remain unchanged:

$$\begin{aligned}\hat{\mathbf{n}} \times \mathbf{H}_{\text{tot}} &= \mathbf{J}_{\text{eq}},\\ \mathbf{E}_{\text{tot}} \times \hat{\mathbf{n}} &= \mathbf{M}_{\text{eq}}.\end{aligned}$$

These currents radiate into an unbounded homogeneous space, producing the scattered fields $\mathbf{E}_{\text{s}}(\mathbf{J}_{\text{eq}}, \mathbf{M}_{\text{eq}})$ and $\mathbf{H}_{\text{s}}(\mathbf{J}_{\text{eq}}, \mathbf{M}_{\text{eq}})$, computable via the free-space Green's functions.

If we choose the equivalent magnetic current $\mathbf{M}_{\text{eq}} = 0$ (continuity of the tangential $\mathbf{E}$-field across S), then only $\mathbf{J}_{\text{eq}}$ remains. Denoting by B the boundary-condition operator and by L the integral operator that produces the scattered field, we have on S:

$$\begin{cases}\mathbf{E}_{\text{tot}} = B(\mathbf{J}_{\text{eq}}),\\ \mathbf{E}_{\text{s}} = L(\mathbf{J}_{\text{eq}}),\\ \mathbf{E}_{\text{tot}} = \mathbf{E}_{\text{inc}} + \mathbf{E}_{\text{s}}.\end{cases}$$

Combining these yields the electric field integral equation (EFIE)

$$B(\mathbf{J}_{\text{eq}}) \;=\; L(\mathbf{J}_{\text{eq}}) \;+\; \mathbf{E}_{\text{inc}}\big|_{S}.$$

In the special case of an infinitely extended PEC plane, $B(\mathbf{J}_{\text{eq}}) = 0$ on S. For a metasurface, an impedance boundary condition replaces the PEC constraint, as detailed in Chapter. Since $\mathbf{E}_{\text{inc}}$ is known, (III) can be solved numerically for $\mathbf{J}_{\text{eq}}$ alone.

IV Definition of the Complex Gradient

The **complex gradient operator** is introduced to generalize gradient-based optimization methods to complex-valued functions without splitting them into real and imaginary parts separately. This is particularly important because dealing with separate real and imaginary components **complicates analytical derivations**.

The **complex derivative** of a function $f(z)$ is defined as:

$$\frac{\widetilde{\partial} f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}\right),$$

where $z = x + jy$ is a complex number with $x, y \in \mathbb{R}$.

For vector functions $f(z)$ where $z \in \mathbb{C}^N$, the **complex gradient operator** is defined as:

$$\widetilde{\nabla} f(z) = \frac{1}{2}\left(\nabla_x f(x,y) + j\nabla_y f(x,y)\right),$$

where ∇_x and ∇_y are the real gradient operators acting on the real and imaginary parts separately.

- Instead of handling gradients separately for real and imaginary parts (which introduces unnecessary complexity), this definition provides a **compact, algebraically consistent way** to define derivatives in the complex space.
- It simplifies optimization in applications involving **complex variables**, such as current-based metasurface antenna optimization.
- It enables seamless integration with **gradient-based optimizers** such as **AdaHessian**, which rely on efficient computation of derivatives.

For a vector-valued function $f : \mathbb{C}^N \to \mathbb{C}^M$, the **complex Jacobian** is defined as:

$$\widetilde{\nabla} f(z) = \begin{bmatrix} \widetilde{\nabla} f_1(z) & \widetilde{\nabla} f_2(z) & \dots & \widetilde{\nabla} f_M(z) \end{bmatrix},$$

where each column contains the gradient of the corresponding component.

- This is particularly relevant for **multi-objective optimization**, where the function to be minimized consists of multiple objectives (e.g., minimizing reflection losses while maximizing gain).
- The Jacobian provides a **systematic way to compute the gradient across all objectives**, ensuring proper optimization of multi-parameter problems.
- It helps in applications where **different performance metrics (e.g., impedance matching, radiation efficiency, or signal gain) must be optimized simultaneously**.

The document provides several key properties of the complex gradient that are useful in optimization. Some important ones include:

Linearity

$$\widetilde{\nabla}(f+g) = \widetilde{\nabla}f + \widetilde{\nabla}g$$
$$\widetilde{\nabla}(cf) = c\widetilde{\nabla}f, \quad \text{for } c \in \mathbb{C}$$

Product Rule

$$\widetilde{\nabla}(fg) = f\widetilde{\nabla}g + g\widetilde{\nabla}f$$

Norm and Inner Product Rules

$$\widetilde{\nabla}|f(z)|^2 = f^*(z)\widetilde{\nabla}f(z) + f(z)\widetilde{\nabla}f^*(z)$$
$$\widetilde{\nabla}\mathrm{Re}(f(z)) = \frac{1}{2}(\widetilde{\nabla}f(z) + \widetilde{\nabla}f^*(z))$$
$$\widetilde{\nabla}\mathrm{Im}(f(z)) = \frac{1}{2j}(\widetilde{\nabla}f(z) - \widetilde{\nabla}f^*(z))$$

These properties make computing gradients much easier when dealing with loss functions that involve power norms, inner products, or squared magnitudes.

The gradient of a Hermitian quadratic form $z^H M z$ is simply:

$$\widetilde{\nabla}(z^H M z) = Mz$$

This is crucial for implementing Hessian-based optimizers like AdaHessian, where second-order curvature information is required.

V Power Objective, Gradient, and Hessian

The coefficient of the electric field in the RWG basis is given by

$$V \;=\; G^{-1}\,(V_{\mathrm{inc}} + L\,I)$$

where the Gram matrix G has entries

$$(G)_{mn} \;=\; \iint_{S_{\mathrm{IBC}}} \Lambda_m(\mathbf{r}) \;\cdot\; \Lambda_n(\mathbf{r})\,\mathrm{d}S.$$

The power-related quantities used in the penalty functions can be written in two equivalent ways. If we average over each cell S_i of area A_i, then

$$P_i = \Re\Big\{\tfrac{1}{A_i}\iint_{S_i} \mathbf{E}\cdot\mathbf{J}^*\,\mathrm{d}S\Big\} = \Re\{\,I^H\,\Gamma_i\,I\},$$

$$Q_i = \Im\Big\{\tfrac{1}{A_i}\iint_{S_i} \mathbf{E}\cdot\mathbf{J}^*\,\mathrm{d}S\Big\} = \Im\{\,I^H\,\Gamma_i\,I\},$$

$$J_i = \tfrac{1}{A_i}\iint_{S_i} |\mathbf{J}|^2\,\mathrm{d}S = I^H\,\Gamma_i\,I,$$

$$E_i = \tfrac{1}{A_i}\iint_{S_i} |\mathbf{E}|^2\,\mathrm{d}S = V^H\,\Gamma_i\,V,$$

where the cell-averaging matrix Γ_i has entries

$$(\Gamma_i)_{mn} \;=\; \tfrac{1}{A_i} \iint_{S_i} \Lambda_m(\mathbf{r}) \,\cdot\, \Lambda_n(\mathbf{r})\,\mathrm{d}S.$$

Here, A_i is the area of the i-th cell S_i. If the relevant quantities are sampled at the centroid r_i^c of each cell, then they may be defined as

$$\begin{aligned} P_i = \Re\big(\mathbf{E}(r_i^c)\cdot\mathbf{J}^*(r_i^c)\big) \;&=\; \Re\{\, I^H\,\Gamma_i\,V\},\\ Q_i = \Im\big(\mathbf{E}(r_i^c)\cdot\mathbf{J}^*(r_i^c)\big) \;&=\; \Im\{\, I^H\,\Gamma_i\,V\},\\ J_i = |\mathbf{J}(r_i^c)|^2 \;&=\; I^H\,\Gamma_i\,I,\\ E_i = |\mathbf{E}(r_i^c)|^2 \;&=\; V^H\,\Gamma_i\,V. \end{aligned}$$

where

$$(\Gamma_i)_{mn} \;=\; \Lambda_m(r_i^c)\,\Lambda_n(r_i^c).$$

Here, r_i^c is the centroid of the i-th cell S_i, defined as

$$r_i^c \;=\; \frac{r_i^1 + r_i^2 + r_i^3}{3},$$

where r_i^1, r_i^2, r_i^3 are the position vectors of the three vertices of S_i.
The gradients of the involved quantities are

$$\tilde{\nabla}_I = 0, \qquad \tilde{\nabla}_{I^*} = 1,$$

$$\tilde{\nabla}V = \tilde{\nabla}\big(G^{-1}V_{\mathrm{inc}} + G^{-1}L\,I\big) = 0, \qquad \tilde{\nabla}V^* = \tilde{\nabla}\big(G^{-1}V_{\mathrm{inc}}^* + G^{-1}L^*\,I^*\big) = L^H\,G^{-1}.$$

$$\nabla_I = 1, \qquad \nabla_{I^*} = 0,$$

$$\nabla V = \nabla\big(G^{-1}V_{\mathrm{inc}} + G^{-1}L\,I\big) = (G^{-1}L)^T, \qquad \nabla V = \nabla\big(G^{-1}V_{\mathrm{inc}}^* + G^{-1}L^*\,I^*\big) = 0.$$

By defining

$$S_i = I^H\Gamma_i\,V = P_i + j\,Q_i,$$

we can find the gradient and mixed block hessian of the objective functions.

$$\begin{aligned} f_{rlz} \;=\; &\sum_i w_i^{\mathrm{act}}\,\rho_i^{\mathrm{act}} \;+\; \sum_i w_{L,i}^{\mathrm{imp}}\,\rho_{L,i}^{\mathrm{imp}} \;+\; \sum_i w_{U,i}^{\mathrm{imp}}\,\rho_{U,i}^{\mathrm{imp}} \;+\; \sum_i w_i^{\mathrm{scal}}\,\rho_i^{\mathrm{scal}} \;+\; \sum_i w_i^{\mathrm{pec}}\,\rho_i^{\mathrm{pec}}\\ &+\; \sum_i w_i^{s/o}\,\rho_i^{s/o} \;+\; \sum_i w_i^{\mathrm{cur}}\,\rho_i^{\mathrm{cur}}. \end{aligned}$$

where

$$
\begin{aligned}
\rho_i^{\text{act}} &= P_i^2, \\
\rho_{L,i}^{\text{imp}} &= r^2(\Psi_{L,i}^{\text{imp}}), \quad \Psi_{L,i}^{\text{imp}} = X_{L,i}\, J_i - Q_i, \\
\rho_{U,i}^{\text{imp}} &= r^2(\Psi_{U,i}^{\text{imp}}), \quad \Psi_{U,i}^{\text{imp}} = Q_i - X_{U,i}\, J_i, \\
\rho_i^{\text{scal}} &= \mathcal{E}_i\, J_i - (P_i^2 + Q_i^2), \\
\rho_i^{\text{pec}} &= \mathcal{E}_i^2, \\
\rho_i^{s/o} &= \mathcal{E}_i\, J_i, \\
\rho_i^{\text{cur}} &= r^2(\Psi_i^{\text{cur}}), \quad \Psi_i^{\text{cur}} = J_i - J_U.
\end{aligned}
$$

The gradient of f is

$$
\begin{aligned}
\widetilde{\nabla} f_{rlz} \;=\; & \sum_i w_i^{\text{act}}\, \widetilde{\nabla}\rho_i^{\text{act}} \;+\; \sum_i w_{L,i}^{\text{imp}}\, \widetilde{\nabla}\rho_{L,i}^{\text{imp}} \;+\; \sum_i w_{U,i}^{\text{imp}}\, \widetilde{\nabla}\rho_{U,i}^{\text{imp}} \\
& +\; \sum_i w_i^{\text{scal}}\, \widetilde{\nabla}\rho_i^{\text{scal}} \;+\; \sum_i w_i^{\text{pec}}\, \widetilde{\nabla}\rho_i^{\text{pec}} \;+\; \sum_i w_i^{s/o}\, \widetilde{\nabla}\rho_i^{s/o} \;+\; \sum_i w_i^{\text{cur}}\, \widetilde{\nabla}\rho_i^{\text{cur}}.
\end{aligned}
$$

$$
\begin{aligned}
\widetilde{\nabla}\rho_i^{\text{act}} &= 2P_i\, \widetilde{\nabla} P_i = P_i(\nu_i + K^H \ell_i), \\
\widetilde{\nabla}\rho_{L,i}^{\text{imp}} &= 2\, r(\Psi_{L,i}^{\text{imp}})(X_{L,i}\, \widetilde{\nabla} J_i - \widetilde{\nabla} Q_i) = r(\Psi_{L,i}^{\text{imp}})(2\, X_{L,i}\, \ell_i - \mathrm{i} K^H \ell_i + \mathrm{j}\, \nu_i), \\
\widetilde{\nabla}\rho_{U,i}^{\text{imp}} &= 2\, r(\Psi_{U,i}^{\text{imp}})(\widetilde{\nabla} Q_i - X_{U,i}\, \widetilde{\nabla} J_i) = r(\Psi_{U,i}^{\text{imp}})(\mathrm{j} K^H - 2X_{U,i})\ell_i - \mathrm{j}\, \nu_i), \\
\widetilde{\nabla}\rho_i^{\text{scal}} &= J_i\, \widetilde{\nabla}\mathcal{E}_i + \mathcal{E}_i\, \widetilde{\nabla} J_i - 2P_i\, \widetilde{\nabla} P_i - 2Q_i\, \widetilde{\nabla} Q_i \\
&\quad = J_i(K^H \nu_i) + \mathcal{E}_i\, \ell_i - P_i(\nu_i + K^H \ell_i) + \mathrm{j}\, Q_i(\nu_i - K^H \ell_i), \\
\widetilde{\nabla}\rho_i^{\text{pec}} &= 2\, \mathcal{E}_i\, \widetilde{\nabla}\mathcal{E}_i = 2\, \mathcal{E}_i(K^H \nu_i), \\
\widetilde{\nabla}\rho_i^{s/o} &= J_i\, \widetilde{\nabla}\mathcal{E}_i + \mathcal{E}_i\, \widetilde{\nabla} J_i = J_i(K^H \nu_i) + \mathcal{E}_i\, \ell_i, \\
\widetilde{\nabla}\rho_i^{\text{cur}} &= 2\, r(\Psi_i^{\text{cur}})\, \widetilde{\nabla} J_i = 2\, r(\Psi_i^{\text{cur}})\, \ell_i.
\end{aligned}
$$

The mixed term hessian of f is

$$
\begin{aligned}
\nabla(\widetilde{\nabla} f_{rlz}) \;=\; & \sum_i w_i^{\text{act}}\, \nabla(\widetilde{\nabla}\rho_i^{\text{act}}) \;+\; \sum_i w_{L,i}^{\text{imp}}\, \nabla(\widetilde{\nabla}\rho_{L,i}^{\text{imp}}) \;+\; \sum_i w_{U,i}^{\text{imp}}\, \nabla(\widetilde{\nabla}\rho_{U,i}^{\text{imp}}) \\
& +\; \sum_i w_i^{\text{scal}}\, \nabla(\widetilde{\nabla}\rho_i^{\text{scal}}) \;+\; \sum_i w_i^{\text{pec}}\, \nabla(\widetilde{\nabla}\rho_i^{\text{pec}}) \;+\; \sum_i w_i^{s/o}\, \nabla(\widetilde{\nabla}\rho_i^{s/o}) \;+\; \sum_i w_i^{\text{cur}}\, \nabla(\widetilde{\nabla}\rho_i^{\text{cur}}).
\end{aligned}
$$

$$
\nabla(\nabla\widetilde{\rho}_i^{\text{act}}) \;=\; \frac{1}{2}\left[\nu_i\, \nu_i^H \;+\; 2\,\Re\{(K_i^H \nu_i)\, \nu_i^H\} \;+\; K_i\, \ell_i^H\, k\right]^T \;+\; \rho_i\, 2\,\Re\{K^H \Gamma_i\}\,.
$$

$$\nabla(\widetilde{\nabla}\rho_{L,i}^{\mathrm{imp}}) = \frac{1}{2}\,u(\Psi_{L,i}^{\mathrm{imp}})\,\Big[4\,\mathrm{j}\,X_{L,i}\,\Re\{\,K^H\ell_i\,\ell_i^H\} - 4\,\mathrm{j}\,X_{L,i}\,\Re\{\,\ell_i\,\gamma_i^H\} + 2\,\Re\{\,K^H\gamma_i^H\}$$
$$-\;K^H\,\ell_i^H\,K + 4\,X_{L,i}\,\ell_i\,\ell_i^H - \gamma_i\,\gamma_i^H\Big]^T + 2\,r(\Psi_{L,i}^{\mathrm{imp}})\,\Big[X_{L,i}\,\Gamma_i\;+\;\Im\{\,K^H\,\Gamma_i\}\Big].$$

$$\nabla(\nabla\widetilde{\rho}_i^{\mathrm{act}}) = \frac{1}{2}\,\Big[\nu_i\,\nu_i^H + 2\,\Re\{(K_i^H\nu_i)\,\nu_i^H\} + K_i\,\ell_i^H\,K\Big]^T + \rho_i\,2\,\Re\{K^H\Gamma_i\},$$

$$\nabla(\nabla\widetilde{\rho}_{L,i}^{\mathrm{imp}}) = \frac{1}{2}\,u(\Psi_{L,i}^{\mathrm{imp}})\,\Big[4\,\mathrm{j}\,X_{L,i}\,\Re\{K^H\ell_i\,\ell_i^H\} - 4\,\mathrm{j}\,X_{L,i}\,\Re\{\ell_i\,\gamma_i^H\} + 2\,\Re\{K^H\gamma_i^H\}$$
$$-\;K^H\,\ell_i^H\,K + 4\,X_{L,i}\,\ell_i\,\ell_i^H - \gamma_i\,\gamma_i^H\Big]^T + 2\,r(\Psi_{L,i}^{\mathrm{imp}})\,[X_{L,i}\,\Gamma_i + \Im\{K^H\Gamma_i\}],$$

$$\nabla(\nabla\widetilde{\rho}_{U,i}^{\mathrm{imp}}) = \frac{1}{2}\,u(\Psi_{U,i}^{\mathrm{imp}})\,\Big[\gamma_i\,\gamma_i^H + 4\,X_{U,i}\,\ell_i\,\ell_i^H + 4\,X_{U,i}\,\Im\{\ell_i\,\gamma_i^H\} + 4\,X_{U,i}\,\Im\{K^H\,\ell_i\,\ell_i^H\}$$
$$+\,2\,\Re\{K^H\,\gamma_i^H\} + K^H\,\ell_i^H\,K\Big] - 2\,r(\Psi_{U,i}^{\mathrm{imp}})\,[\Im\{K^H\Gamma_i\} + X_{U,i}\,\Gamma_i],$$

$$\nabla(\nabla\widetilde{\rho}_i^{\mathrm{scal}}) = \mathcal{E}_i\,\Gamma_i + \mathcal{J}_i\,(K^H\,\Gamma_i\,K) - 2\,P_i\,\Re\{K^H\,\Gamma_i\} - 2\,Q_i\,\Im\{K^H\,\Gamma_i\},$$

$$\nabla(\nabla\widetilde{\rho}_i^{\mathrm{pec}}) = (K^H\,\nu_i\,\nu_i^H\,K)^T + 2\,\mathcal{E}_i\,(K^H\,\Gamma_i\,K),$$

$$\nabla(\widetilde{\nabla}\rho_i^{\mathrm{s/o}}) = 2\,\Re\big((K^H\,\nu_i\,\ell_i^H)^T\big) + \mathcal{E}_i\,\Gamma_i - J_i\,(K^H\,\Gamma_i\,K)$$

VI Far-Field Objective, Gradient, and Hessian

The far-field objective $f_{rad}(I)$ measures compliance with radiation masks via squared ramp penalties on the co- and cross-polarized far-field power. In explicit form:

$$\widetilde{\nabla} f_{\mathrm{rad}}(I) \;=\; w_{\mathrm{ref}}\,\widetilde{\nabla}\rho_{\mathrm{ref}} + \sum_j \Big[w_{\mathrm{co,L},j}\,\widetilde{\nabla}\rho_{\mathrm{co,L},j} + w_{\mathrm{co,U},j}\,\widetilde{\nabla}\rho_{\mathrm{co,U},j} + w_{\mathrm{cx},j}\,\widetilde{\nabla}\rho_{\mathrm{cx},j} + w_{\mathrm{tot},j}\,\widetilde{\nabla}\rho_{\mathrm{tot},j}\Big].$$

$$\rho_{\mathrm{ref}}(I) = \left[r\,(M_0 - F_{\mathrm{ref}}(I))\right]^2.$$

$$F_{\mathrm{ref}}(I) = F_0 + \sum_j \zeta_{\mathrm{ref},j}\,F_{\mathrm{co},j}(I),$$

$$\rho_{\mathrm{co,L},j}(I) = \left[r\,(M_{\mathrm{co,L},j} - F_{\mathrm{co},j}(I))\right]^2.$$

$$\rho_{\mathrm{co,U},j}(I) = \left[r\,(F_{\mathrm{co},j}(I) - M_{\mathrm{co,U},j})\right]^2.$$

$$\rho_{\mathrm{cx},j}(I) = \left[r\,(F_{\mathrm{cx},j}(I) - M_{\mathrm{cx},j})\right]^2.$$

$$\rho_{\mathrm{tot},j}(I) = \left[r\left(F_{\mathrm{tot},j}(I) - M_{\mathrm{tot},j}\right)\right]^2, \quad \text{with} \quad F_{\mathrm{tot},j} = F_{\mathrm{co},j} + F_{\mathrm{cx},j}.$$

$$M_{\mathrm{co,U},j} = \sigma_{\mathrm{co,U},j} F_{\mathrm{ref}}, \quad M_{\mathrm{co,L},j} = \sigma_{\mathrm{co,L},j} F_{\mathrm{ref}}, \quad M_{\mathrm{cx},j} = \sigma_{\mathrm{cx},j} F_{\mathrm{ref}}, \quad M_{\mathrm{tot},j} = \sigma_{\mathrm{tot},j} F_{\mathrm{ref}} \quad (2)$$

$$E_{\mathrm{co}}(\hat{r}, I) = (E_\theta, E_\phi) \cdot \hat{p}^*(\hat{r}), \quad F_{\mathrm{co}}(\hat{r}, I) = |E_{\mathrm{co}}|^2.$$

$$F_{\mathrm{co},j}(I) = \left|(R_\theta I)_j\, \hat{p}_j^{\theta *} + (R_\phi I)_j\, \hat{p}_j^{\phi *}\right|^2,$$

$$E_{\mathrm{cx}}(\hat{r}, I) = (E_\theta, E_\phi) \cdot \hat{q}^*(\hat{r}), \quad F_{\mathrm{cx}}(\hat{r}, I) = |E_{\mathrm{cx}}|^2.$$

In matrix form:

$$F_{\mathrm{cx},j}(I) = \left|(R_\theta I)_j\, \hat{q}_j^{\theta *} + (R_\phi I)_j\, \hat{q}_j^{\phi *}\right|^2,$$

The gradient of each objective function is:

$$\widetilde{\nabla} f_{\mathrm{rad}}(I) \;=\; w_{\mathrm{ref}}\, \widetilde{\nabla}\rho_{\mathrm{ref}} + \sum_j \Big[w_{\mathrm{co,L},j}\, \widetilde{\nabla}\rho_{\mathrm{co,L},j} + w_{\mathrm{co,U},j}\, \widetilde{\nabla}\rho_{\mathrm{co,U},j} + w_{\mathrm{cx},j}\, \widetilde{\nabla}\rho_{\mathrm{cx},j} + w_{\mathrm{tot},j}\, \widetilde{\nabla}\rho_{\mathrm{tot},j} \Big].$$

$$\widetilde{\nabla}\rho_{\mathrm{ref}} = -2\, r(\Psi_{\mathrm{ref}}) \widetilde{\nabla} F_{\mathrm{ref}}, \quad \text{where} \quad \Psi_{\mathrm{ref}} \triangleq M_0 - F_{\mathrm{ref}}(I)$$

$$\widetilde{\nabla}\rho_{\mathrm{co,L},j} = 2\, r(\Psi_{\mathrm{co,L},j}) \left(\sigma_{\mathrm{co,L},j} \widetilde{\nabla} F_{\mathrm{ref}} - \widetilde{\nabla} F_{\mathrm{co},j}\right)$$

$$\widetilde{\nabla}\rho_{\mathrm{co,U},j} = 2\, r(\Psi_{\mathrm{co,U},j}) \left(\widetilde{\nabla} F_{\mathrm{co},j} - \sigma_{\mathrm{co,U},j} \widetilde{\nabla} F_{\mathrm{ref}}\right)$$

$$\widetilde{\nabla}\rho_{\mathrm{cx},j} = 2\, r(\Psi_{\mathrm{cx},j}) \left(\widetilde{\nabla} F_{\mathrm{cx},j} - \sigma_{\mathrm{cx},j} \widetilde{\nabla} F_{\mathrm{ref}}\right)$$

$$\widetilde{\nabla}\rho_{\mathrm{tot},j} = 2\, r(\Psi_{\mathrm{tot},j}) \left(\widetilde{\nabla} F_{\mathrm{tot},j} - \sigma_{\mathrm{tot},j} \widetilde{\nabla} F_{\mathrm{ref}}\right)$$

The mixed-term hessian of each objective function is:

$$\begin{aligned} \nabla(\widetilde{\nabla} f_{\mathrm{rad}}(I)) = w_{\mathrm{ref}}\, \nabla(\widetilde{\nabla}\rho_{\mathrm{ref}}) \;+\; & \sum_j w_{\mathrm{co,L},j}\, \nabla(\widetilde{\nabla}\rho_{\mathrm{co,L},j}) \\ & + \sum_j w_{\mathrm{co,U},j}\, \nabla(\widetilde{\nabla}\rho_{\mathrm{co,U},j}) \;+\; \sum_j w_{\mathrm{cx},j}\, \nabla(\widetilde{\nabla}\rho_{\mathrm{cx},j}) \\ & + \sum_j w_{\mathrm{tot},j}\, \nabla(\widetilde{\nabla}\rho_{\mathrm{tot},j}) \end{aligned}$$

$$\begin{aligned}
\nabla(\widetilde{\nabla}\rho_{\mathrm{co,L},j}) &= 2\,r'(\Psi_{\mathrm{co,L},j})\big(\sigma_{\mathrm{co,L},j}\,\nabla F_{\mathrm{ref}} - \nabla F_{\mathrm{co},j}\big)\big(\sigma_{\mathrm{co,L},j}\,\widetilde{\nabla} F_{\mathrm{ref}} - \widetilde{\nabla} F_{\mathrm{co},j}\big)^H \\
&\quad + 2\,r(\Psi_{\mathrm{co,L},j})\big(\sigma_{\mathrm{co,L},j}\,\nabla(\widetilde{\nabla} F_{\mathrm{ref}}) - \nabla(\widetilde{\nabla} F_{\mathrm{co},j})\big), \\
\nabla(\widetilde{\nabla}\rho_{\mathrm{ref}}) &= 2\,r'(\Psi_{\mathrm{ref}})\,\nabla F_{\mathrm{ref}}\;\widetilde{\nabla} F_{\mathrm{ref}}^H \;-\; 2\,r(\Psi_{\mathrm{ref}})\,\nabla(\widetilde{\nabla} F_{\mathrm{ref}}), \\
\nabla(\widetilde{\nabla}\rho_{\mathrm{co,U},j}) &= 2\,r'(\Psi_{\mathrm{co,U},j})\big(\nabla F_{\mathrm{co},j} - \sigma_{\mathrm{co,U},j}\,\nabla F_{\mathrm{ref}}\big)\big(\widetilde{\nabla} F_{\mathrm{co},j} - \sigma_{\mathrm{co,U},j}\,\widetilde{\nabla} F_{\mathrm{ref}}\big)^H \\
&\quad + 2\,r(\Psi_{\mathrm{co,U},j})\big(\nabla(\widetilde{\nabla} F_{\mathrm{co},j}) - \sigma_{\mathrm{co,U},j}\,\nabla(\widetilde{\nabla} F_{\mathrm{ref}})\big), \\
\nabla(\widetilde{\nabla}\rho_{\mathrm{cx},j}) &= 2\,r'(\Psi_{\mathrm{cx},j})\big(\nabla F_{\mathrm{cx},j} - \sigma_{\mathrm{cx},j}\,\nabla F_{\mathrm{ref}}\big)\big(\widetilde{\nabla} F_{\mathrm{cx},j} - \sigma_{\mathrm{cx},j}\,\widetilde{\nabla} F_{\mathrm{ref}}\big)^H \\
&\quad + 2\,r(\Psi_{\mathrm{cx},j})\big(\nabla(\widetilde{\nabla} F_{\mathrm{cx},j}) - \sigma_{\mathrm{cx},j}\,\nabla(\widetilde{\nabla} F_{\mathrm{ref}})\big), \\
\nabla(\widetilde{\nabla}\rho_{\mathrm{tot},j}) &= 2\,r'(\Psi_{\mathrm{tot},j})\big(\nabla F_{\mathrm{co},j} + \nabla F_{\mathrm{cx},j} - \sigma_{\mathrm{tot},j}\,\nabla F_{\mathrm{ref}}\big)\big(\widetilde{\nabla} F_{\mathrm{co},j} + \widetilde{\nabla} F_{\mathrm{cx},j} - \sigma_{\mathrm{tot},j}\,\widetilde{\nabla} F_{\mathrm{ref}}\big)^H \\
&\quad + 2\,r(\Psi_{\mathrm{tot},j})\big(\nabla(\widetilde{\nabla} F_{\mathrm{co},j}) + \nabla(\widetilde{\nabla} F_{\mathrm{cx},j}) - \sigma_{\mathrm{tot},j}\,\nabla(\widetilde{\nabla} F_{\mathrm{ref}})\big).
\end{aligned}$$

For a given j, the Hessians of the power quantities can be compactly expressed:

$$\nabla(\widetilde{\nabla} F_{\mathrm{co},j}) = W_{\mathrm{co},j}^H W_{\mathrm{co},j} \quad \text{where} \quad W_{\mathrm{co},j} \triangleq \hat{p}_j^{\theta}[R_\theta]_{j,:} + \hat{p}_j^{\phi}[R_\phi]_{j,:}$$

$$\nabla(\widetilde{\nabla} F_{\mathrm{co},j}) = R_{\mathrm{co}}^H \Omega_j R_{\mathrm{co}}, \quad \nabla(\widetilde{\nabla} F_{\mathrm{cx},j}) = R_{\mathrm{cx}}^H \Omega_j R_{\mathrm{cx}}$$

$$\nabla(\widetilde{\nabla} F_{\mathrm{ref}}) = \sum_m \zeta_{\mathrm{ref},m} W_{\mathrm{co},m}^H W_{\mathrm{co},m} = R_{\mathrm{co}}^H \,\mathrm{diag}(\zeta_{\mathrm{ref}}) R_{\mathrm{co}}$$

where:

- $\Omega_j = e_j e_j^T$ selects the jth direction,
- $R_{\mathrm{co}} \triangleq \mathrm{diag}(\hat{p}^{\theta}) R_\theta + \mathrm{diag}(\hat{p}^{\phi}) R_\phi$
- $R_{\mathrm{cx}} \triangleq \mathrm{diag}(\hat{q}^{\theta}) R_\theta + \mathrm{diag}(\hat{q}^{\phi}) R_\phi$

Combining all contributions (and including the weight factors w from Eq. (1)), we obtain an expression for the mixed Hessian matrix of f_{rad}. It can be written as a sum over directions of outer-product terms and Hessian-of-F terms as in Eqs. (4)–(7) above. For

compactness, we present the result grouped by each category of cost term:

$$
\begin{aligned}
\nabla_I(\nabla_{I^*} f_{\mathrm{rad}}(I)) = \; & 2\, w_{\mathrm{ref}}\Big[u(\Psi_{\mathrm{ref}})\, \nabla_I F_{\mathrm{ref}}\, (\nabla_{I^*} F_{\mathrm{ref}})^T \; - \; r(\Psi_{\mathrm{ref}})\, \nabla_I(\nabla_{I^*} F_{\mathrm{ref}})\Big] \\
& + 2\sum_j w_{\mathrm{co},L,j}\Big[u(\Psi_{\mathrm{co},L,j})\; (\sigma_{\mathrm{co},L,j}\, \nabla_I F_{\mathrm{ref}} - \nabla_I F_{\mathrm{co},j})\, (\sigma_{\mathrm{co},L,j}\, \nabla_{I^*} F_{\mathrm{ref}} - \nabla_{I^*} F_{\mathrm{co},j})^T \\
& \qquad + \; r(\Psi_{\mathrm{co},L,j})\; (\sigma_{\mathrm{co},L,j}\, \nabla_I(\nabla_{I^*} F_{\mathrm{ref}}) - \nabla_I(\nabla_{I^*} F_{\mathrm{co},j}))\Big] \\
& + 2\sum_j w_{\mathrm{co},U,j}\Big[u(\Psi_{\mathrm{co},U,j})\; (\nabla_I F_{\mathrm{co},j} - \sigma_{\mathrm{co},U,j}\, \nabla_I F_{\mathrm{ref}})\, (\nabla_{I^*} F_{\mathrm{co},j} - \sigma_{\mathrm{co},U,j}\, \nabla_{I^*} F_{\mathrm{ref}})^T \\
& \qquad + \; r(\Psi_{\mathrm{co},U,j})\; (\nabla_I(\nabla_{I^*} F_{\mathrm{co},j}) - \sigma_{\mathrm{co},U,j}\, \nabla_I(\nabla_{I^*} F_{\mathrm{ref}}))\Big] \\
& + 2\sum_j w_{\mathrm{cx},j}\Big[u(\Psi_{\mathrm{cx},j})\; (\nabla_I F_{\mathrm{cx},j} - \sigma_{\mathrm{cx},j}\, \nabla_I F_{\mathrm{ref}})\, (\nabla_{I^*} F_{\mathrm{cx},j} - \sigma_{\mathrm{cx},j}\, \nabla_{I^*} F_{\mathrm{ref}})^T \\
& \qquad + \; r(\Psi_{\mathrm{cx},j})\; (\nabla_I(\nabla_{I^*} F_{\mathrm{cx},j}) - \sigma_{\mathrm{cx},j}\, \nabla_I(\nabla_{I^*} F_{\mathrm{ref}}))\Big] \\
& + 2\sum_j w_{\mathrm{tot},j}\Big[u(\Psi_{\mathrm{tot},j})\; (\nabla_I F_{\mathrm{co},j} + \nabla_I F_{\mathrm{cx},j} - \sigma_{\mathrm{tot},j}\, \nabla_I F_{\mathrm{ref}}) \\
& \qquad \times (\nabla_{I^*} F_{\mathrm{co},j} + \nabla_{I^*} F_{\mathrm{cx},j} - \sigma_{\mathrm{tot},j}\, \nabla_{I^*} F_{\mathrm{ref}})^T \\
& \qquad + \; r(\Psi_{\mathrm{tot},j})\; (\nabla_I(\nabla_{I^*} F_{\mathrm{co},j}) + \nabla_I(\nabla_{I^*}(F_{\mathrm{cx},j}) - \sigma_{\mathrm{tot},j}\, \nabla_I(\nabla_{I^*} F_{\mathrm{ref}}))\Big].
\end{aligned}
$$

List of the unknowns:

Symbol	Description	Type
$\nabla_I F_{\mathrm{ref}}$	Gradient of reference intensity	$\mathbb{C}^N$
$\nabla_{I^*} F_{\mathrm{ref}}$	Conjugate gradient of reference intensity	$\mathbb{C}^N$
$\nabla_I F_{\mathrm{co},j}$	Gradient of co-pol power (direction j)	$\mathbb{C}^N$
$\nabla_{I^*} F_{\mathrm{co},j}$	Conjugate gradient of co-pol power	$\mathbb{C}^N$
$\nabla_I F_{\mathrm{cx},j}$	Gradient of cross-pol power (direction j)	$\mathbb{C}^N$
$\nabla_{I^*} F_{\mathrm{cx},j}$	Conjugate gradient of cross-pol power	$\mathbb{C}^N$
$\nabla_I\,(\nabla_{I^*} F_{\mathrm{ref}})$	Mixed Hessian of reference intensity	$\mathbb{C}^{N\times N}$
$\nabla_I\,(\nabla_{I^*} F_{\mathrm{co},j})$	Mixed Hessian of co-pol power (direction j)	$\mathbb{C}^{N\times N}$
$\nabla_I\,(\nabla_{I^*} F_{\mathrm{cx},j})$	Mixed Hessian of cross-pol power (direction j)	$\mathbb{C}^{N\times N}$

Table 1. List of unique gradient and mixed Hessian terms appearing in $H_{I,I^*}[f_{\mathrm{rad}}]$.

$$
\nabla_I\,(\nabla_{I^*} F_{\mathrm{ref}}) = \left[R_{\mathrm{co}}^H \left(\sum_j \zeta_j^{\mathrm{ref}}\, \Omega_j\right) R_{\mathrm{co}}\right]^* .
$$

$$
[\nabla_I F_{\mathrm{ref}}]\,[\nabla_{I^*} F_{\mathrm{ref}}]^\top = \left(R_{\mathrm{co}}^H \left(\zeta^{\mathrm{ref}} \odot E^{\mathrm{co}}\right)\right)^* \left(R_{\mathrm{co}}^H \left(\zeta^{\mathrm{ref}} \odot E^{\mathrm{co}}\right)\right)^\top
$$

$$[\nabla_I F_{\text{ref}}]\,[\nabla_{I^*} F_{\text{co},j}]^\top = \left(R_{\text{co}}^H \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right)\right)^* \left(R_{\text{co}}^H \Omega_j E^{\text{co}}\right)^\top$$

$$[\nabla_I F_{\text{co},j}]\,[\nabla_{I^*} F_{\text{ref}}]^\top = \left(R_{\text{co}}^H\, \Omega_j\, E^{\text{co}}\right)^* \left(R_{\text{co}}^H \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right)\right)^\top$$

$$[\nabla_I F_{\text{co},j}]\,[\nabla_{I^*} F_{\text{co},j}]^\top = \left(R_{\text{co}}^H\, \Omega_j\, E^{\text{co}}\right)^* \left(R_{\text{co}}^H\, \Omega_j\, E^{\text{co}}\right)^\top$$

$$\nabla_I \left(\nabla_{I^*} F_{\text{co},j}\right) = \left(R_{\text{co}}^H \Omega_j R_{\text{co}}\right)^*$$

$$[\nabla_I F_{\text{cx},j}]\,[\nabla_{I^*} F_{\text{cx},j}]^\top = \left(R_{\text{cx}}^H\, \Omega_j\, E^{\text{cx}}\right)^* \left(R_{\text{cx}}^H\, \Omega_j\, E^{\text{cx}}\right)^\top$$

$$[\nabla_I F_{\text{ref}}]\,[\nabla_{I^*} F_{\text{cx},j}]^\top = \left(R_{\text{co}}^H \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right)\right)^* \left(R_{\text{cx}}^H \Omega_j E^{\text{cx}}\right)^\top$$

$$[\nabla_I F_{\text{cx},j}]\,[\nabla_{I^*} F_{\text{ref}}]^\top = \left(R_{\text{cx}}^H\, \Omega_j\, E^{\text{cx}}\right)^* \left(R_{\text{co}}^H \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right)\right)^\top$$

$$\nabla_I \left(\nabla_{I^*} F_{\text{cx},j}\right) = \left(R_{\text{cx}}^H\, \Omega_j\, R_{\text{cx}}\right)^*$$

We define the co-polarized and cross-polarized radiation operators as follows:

$$R_{\text{co}} = \text{diag}(\hat{p}_\theta^*)\, R_\theta \;+\; \text{diag}(\hat{p}_\phi^*)\, R_\phi,$$
$$R_{\text{cx}} = \text{diag}(\hat{q}_\theta^*)\, R_\theta \;+\; \text{diag}(\hat{q}_\phi^*)\, R_\phi.$$

Here:

- $\hat{p}_\theta, \hat{p}_\phi$ are the co-polarization unit vectors (complex).
- $\hat{q}_\theta, \hat{q}_\phi$ are the cross-polarization unit vectors.
- R_θ, R_ϕ are the radiation operators in the θ and ϕ components.

The directional projector onto sample j is given by:

$$\Omega_j = e_j e_j^\top,$$

where e_j is the canonical unit vector selecting the jth far-field direction.

Calculating the full statement substituting the unknowns.

$$\nabla_I(\nabla_{I^*} f_{\text{rad}}(I)) =$$

$$\begin{aligned}
&R_{\text{co}}^H \Big[\left(\zeta^{\text{ref}} \odot E^{\text{co}}\right) \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right)^H w_A - \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right) (W_B \odot E^{\text{co}})^H \\
&\qquad + (W_C \odot E^{\text{co}}) \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right)^H + \text{Diag}(W_D \odot (E^{\text{co}} \odot E^{\text{co}})) + \text{Diag}(W_E) + \text{Diag}(W_F) \Big] R_{\text{co}} \\
&-R_{\text{co}}^H \left[\left(\zeta^{\text{ref}} \odot E^{\text{co}}\right) (W_I \odot E^{\text{cx}})^H \right] R_{\text{cx}} \\
&-R_{\text{cx}}^H \left[(W_H \odot E^{\text{cx}}) \left(\zeta^{\text{ref}} \odot E^{\text{co}}\right)^H \right] R_{\text{co}} \\
&+R_{\text{cx}}^H \left[\text{Diag}\left(W_G \odot (E^{\text{cx}} \odot E^{\text{cx}})\right) + \text{Diag}(W_J) \right] R_{\text{cx}}
\end{aligned}$$

where

$$w_A = \sum_i 2w_{\text{co},L,i}\, u(\Psi_{\text{co},L,i}) \left(\delta_{\text{co},L,i}\right)^2$$

$$W_{B,j} = 2w_{\text{co},L,j}\, u\big(\Psi_{\text{co},L,j}\big)\, \delta_{\text{co},L,j} + 2w_{\text{co},U,j}\, u\big(\Psi_{\text{co},U,j}\big)\, \delta_{\text{co},U,j} + 2w_{\text{tot},j}\, u\big(\Psi_{\text{tot},j}\big)\, \delta_{\text{tot},j}$$

$$W_{C,j} = W_{B,j}$$

$$W_{D,j} = 2w_{\text{co},L,j}\, u\left(\Psi_{\text{co},L,j}\right) + 2w_{\text{co},U,j}\, u\left(\Psi_{\text{co},U,j}\right) + 2w_{\text{tot},j}\, u\left(\Psi_{\text{tot},j}\right)$$

$$W_{E,j} = 2w_{\text{co},L,j}\, r\left(\Psi_{\text{co},L,j}\right)\, \delta_{\text{co},L,j} - 2w_{\text{co},U,j}\, r\left(\Psi_{\text{co},U,j}\right)\, \delta_{\text{co},U,j} - 2w_{\text{cx},j}\, r\left(\Psi_{\text{cx},j}\right)\, \delta_{\text{cx},j} - 2w_{\text{tot},j}\, r\left(\Psi_{\text{tot},j}\right)\, \delta_{\text{tot},j}$$

$$W_{F,j} = -2w_{\text{co},L,j}\, r\left(\Psi_{\text{co},L,j}\right) + 2w_{\text{co},U,j}\, r\left(\Psi_{\text{co},U,j}\right) + 2w_{\text{tot},j}\, r\left(\Psi_{\text{tot},j}\right)$$

$$W_{G,j} = 2w_{\text{cx},j}\, u\left(\Psi_{\text{cx},j}\right) + 2w_{\text{tot},j}\, u\left(\Psi_{\text{tot},j}\right)$$

$$W_{H,j} = 2w_{\text{cx},j}\, u\left(\Psi_{\text{cx},j}\right) \delta_{\text{cx},j} + 2w_{\text{tot},j}\, u\left(\Psi_{\text{tot},j}\right) \delta_{\text{tot},j}$$

$$W_{I,j} = 2w_{\text{cx},j}\, u\left(\Psi_{\text{cx},j}\right) \delta_{\text{cx},j} + 2w_{\text{tot},j}\, u\left(\Psi_{\text{tot},j}\right) \delta_{\text{tot},j}$$

$$W_{J,j} = 2w_{\text{cx},j}\, r\left(\Psi_{\text{cx},j}\right)\, \delta_{\text{cx},j} + 2w_{\text{tot},j}\, r\left(\Psi_{\text{tot},j}\right)\, \delta_{\text{tot},j}$$

Finally we can write:

$$\nabla_I(\nabla_{I^*} f_{\text{rad}}(I)) = R_{\text{co}}^H M_1 R_{\text{co}} - R_{\text{co}}^H M_2 R_{\text{cx}} - R_{\text{cx}}^H M_2 R_{\text{co}} + R_{\text{cx}}^H M_3 R_{\text{cx}}$$

You can collect all 16 of the little "θ/ϕ" terms into the single block-matrix form

$$\boxed{H \;=\; \begin{bmatrix} R_\theta^H & R_\phi^H \end{bmatrix} \begin{bmatrix} A_{\theta\theta} & A_{\theta\phi} \\ A_{\phi\theta} & A_{\phi\phi} \end{bmatrix} \begin{bmatrix} R_\theta \\ R_\phi \end{bmatrix}}$$

where the four $N_f \times N_f$ diagonal blocks are

$$
\begin{aligned}
A_{\theta\theta} &= \operatorname{diag}(p_\theta)\, M_1 \operatorname{diag}(p_\theta^*) - \operatorname{diag}(p_\theta)\, M_2 \operatorname{diag}(q_\theta^*) - \operatorname{diag}(q_\theta)\, M_2 \operatorname{diag}(p_\theta^*) + \operatorname{diag}(q_\theta)\, M_3 \operatorname{diag}(q_\theta^*),\\
A_{\theta\phi} &= \operatorname{diag}(p_\theta)\, M_1 \operatorname{diag}(p_\phi^*) - \operatorname{diag}(p_\theta)\, M_2 \operatorname{diag}(q_\phi^*) - \operatorname{diag}(q_\theta)\, M_2 \operatorname{diag}(p_\phi^*) + \operatorname{diag}(q_\theta)\, M_3 \operatorname{diag}(q_\phi^*),\\
A_{\phi\theta} &= \operatorname{diag}(p_\phi)\, M_1 \operatorname{diag}(p_\theta^*) - \operatorname{diag}(p_\phi)\, M_2 \operatorname{diag}(q_\theta^*) - \operatorname{diag}(q_\phi)\, M_2 \operatorname{diag}(p_\theta^*) + \operatorname{diag}(q_\phi)\, M_3 \operatorname{diag}(q_\theta^*),\\
A_{\phi\phi} &= \operatorname{diag}(p_\phi)\, M_1 \operatorname{diag}(p_\phi^*) - \operatorname{diag}(p_\phi)\, M_2 \operatorname{diag}(q_\phi^*) - \operatorname{diag}(q_\phi)\, M_2 \operatorname{diag}(p_\phi^*) + \operatorname{diag}(q_\phi)\, M_3 \operatorname{diag}(q_\phi^*).
\end{aligned}
$$

where

$$
\begin{aligned}
M_1 = {} & \left(\zeta^{\mathrm{ref}} \odot E^{\mathrm{co}}\right)\left(\zeta^{\mathrm{ref}} \odot E^{\mathrm{co}}\right)^H w_A - \left(\zeta^{\mathrm{ref}} \odot E^{\mathrm{co}}\right)(W_B \odot E^{\mathrm{co}})^H \\
& + (W_C \odot E^{\mathrm{co}})\left(\zeta^{\mathrm{ref}} \odot E^{\mathrm{co}}\right)^H + \operatorname{Diag}\left(W_D \odot (E^{\mathrm{co}} \odot E^{\mathrm{co}})\right) + \operatorname{Diag}(W_E) + \operatorname{Diag}(W_F)\\
M_2 = {} & \left(\zeta^{\mathrm{ref}} \odot E^{\mathrm{co}}\right)(W_I \odot E^{\mathrm{cx}})^H\\
M_3 = {} & \left[\operatorname{Diag}\left(W_G \odot (E^{\mathrm{cx}} \odot E^{\mathrm{cx}})\right) + \operatorname{Diag}(W_J)\right]
\end{aligned}
$$

Equivalently,

$$H = R_\theta^H A_{\theta\theta} R_\theta + R_\theta^H A_{\theta\phi} R_\phi + R_\phi^H A_{\phi\theta} R_\theta + R_\phi^H A_{\phi\phi} R_\phi.$$

That is the final Hessian block, written only in terms of R_θ, R_ϕ and their Hermitians.

$$
\begin{aligned}
\left(w^{\mathrm{ref}}\, \nabla\left(\tilde{\nabla}\rho^{\mathrm{ref}}\right)\right)^* = {} & R_\theta^H\Big[2\, w_{\mathrm{ref}}\, u(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot E_\theta^{\mathrm{co}}\right)\left(\zeta^{\mathrm{ref}} \odot E_\theta^{\mathrm{co}}\right)^H - 2\, w_{\mathrm{ref}}\, r(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot |\hat{p}_\theta|^2\right)\Big] R_\theta \\
& + R_\theta^H\Big[2\, w_{\mathrm{ref}}\, u(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot E_\theta^{\mathrm{co}}\right)\left(\zeta^{\mathrm{ref}} \odot E_\phi^{\mathrm{co}}\right)^H - 2\, w_{\mathrm{ref}}\, r(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot (\hat{p}_\theta \odot \hat{p}_\phi^*)\right)\Big] R_\phi \\
& + R_\phi^H\Big[2\, w_{\mathrm{ref}}\, u(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot E_\phi^{\mathrm{co}}\right)\left(\zeta^{\mathrm{ref}} \odot E_\theta^{\mathrm{co}}\right)^H - 2\, w_{\mathrm{ref}}\, r(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot (\hat{p}_\phi \odot \hat{p}_\theta^*)\right)\Big] R_\theta \\
& + R_\phi^H\Big[2\, w_{\mathrm{ref}}\, u(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot E_\phi^{\mathrm{co}}\right)\left(\zeta^{\mathrm{ref}} \odot E_\phi^{\mathrm{co}}\right)^H - 2\, w_{\mathrm{ref}}\, r(\Psi_{\mathrm{ref}})\left(\zeta^{\mathrm{ref}} \odot |\hat{p}_\phi|^2\right)\Big] R_\phi\,.
\end{aligned}
$$

Bibliography

[1] S. Adrian, A. Dely, D. Consoli, A. Merlini, and F. Andriulli. Electromagnetic integral equations: Insights in conditioning and preconditioning. *IEEE Open Journal of Antennas and Propagation*, 2:1143–1174, 2021.

[2] F. P. Andriulli, A. Tabacco, and G. Vecchi. A multiresolution approach to the electric field integral equation in antenna problems. *SIAM Journal on Scientific Computing*, 28(1), 2006.

[3] M. Beccaria, A. Niccolai, R. E. Zich, and P. Pirinoli. Shaped-beam reflectarray design by means of social network optimization (sno). *Electronics*, 10(6):744, 2021.

[4] J. Budhu, L. Szymanski, and A. Grbic. Design of planar and conformal, passive, lossless metasurfaces that beamform. *IEEE Transactions on Antennas and Propagation*, 69(3):1651–1664, Mar. 2021.

[5] O. Chapelle and D. Erhan. Gradient descent optimization algorithms. In *NIPS 2011 Deep Learning and Unsupervised Feature Learning Workshop*, 2011. https://deeplearningworkshopnips2011.files.wordpress.com/2011/10/optimization.pdf.

[6] ElProCus Editorial Team. Reflector antenna – working and its applications. https://www.elprocus.com/reflector-antenna/, n.d. Accessed: 2025-05-12.

[7] A. Epstein and G. V. Eleftheriades. Passive lossless huygens metasurfaces for conversion of arbitrary source field to directive radiation. *IEEE Transactions on Antennas and Propagation*, 62(11):5680–5695, Nov. 2014.

[8] M. Faenzi, G. Minatti, D. González-Ovejero, F. Caminita, E. Martini, C. D. Giovampaola, and S. Maci. Metasurface antennas: New models, applications and realizations. *Scientific Reports*, 9:10178, 2019.

[9] R. Grosse and J. Ba. Csc421/2516 lectures 7–8: Optimization. Lecture slides, University of Toronto, 2018. Accessed: 2025-06-15.

[10] A. Guida. Optimization strategies for metasurface antenna design. Master's degree thesis, Politecnico di Torino, Turin, Italy, 2023. Advisor: Prof. Giuseppe Vecchi; Co-advisors: Dr. Marcello Zucchi, Dr. Giorgio Giordanengo.

[11] C. T. Howard, W. D. Hunt, and K. W. Allen. Equivalent circuit modeling and design of a reconfigurable loaded dogbone metasurface element. *IEEE Transactions on Antennas and Propagation*, 68(7):5632–5644, 2020.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] K. Kreutz-Delgado. The complex gradient operator and the cr-calculus. Technical report, Electrical and Computer Engineering, Jacobs School of Engineering, University of California, San Diego, 2009. Version UCSD-ECE275CG-S2009v1.0.

[14] J. Li. *Integral Equations in Computational Electromagnetics: Formulations, Properties and Isogeometric Analysis*. Ph.D. dissertation, Michigan State University, East Lansing, MI, USA, 2018. Advisor: Balasubramaniam Shanker.

[15] S. Maci, G. Minatti, M. Casaletti, and M. Bosiljevac. Metasurfing: Addressing waves on impenetrable metasurfaces. *IEEE Antennas and Wireless Propagation Letters*, 10:1499–1502, 2011.

[16] P. Mavrikakis and O. J. F. Martin. Hybrid meshes for the modeling of optical nanostructures with the surface integral equation method. *Journal of the Optical Society of America A*, 42(3):315–326, 2025.

[17] Microwaves101 Editors. Phased array antennas. https://www.microwaves101.com/encyclopedias/phased-array-antennas, n.d. Accessed: 2025-05-12.

[18] C. Narendra, T. Brown, and P. Mojabi. Gradient-based electromagnetic inversion for metasurface design using circuit models. *IEEE Transactions on Antennas and Propagation*, 70(3):2046–2058, Mar. 2022.

[19] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 2 edition, 2006.

[20] A. Osipov, J. Hannula, P. Naccachian, et al. Physical limitations of phased array antennas. *IEEE Transactions on Antennas and Propagation*, 2021. In Press.

[21] D. Rozban, A. Barom, G. Kedar, A. Etinger, T. Rabinovitz, and A. Abramovich. Enhancement of phase dynamic range in design of reconfigurable metasurface reflect array antenna using two types of unit cells for e band communication. *Electronics*, 13(9):1779, 2024.

[22] A. A. Sakr, E. A. Soliman, and A. K. Abdelmageed. A surface integral equation formulation for electromagnetic scattering from a conducting cylinder coated with multilayers of homogeneous materials. *Journal of Applied Physics*, 116(5):054902, 2014.

[23] S. Tariq, W. T. Sethi, A. A. Rahim, F. Faisal, and T. Djerafi. A metasurface assisted pin loaded antenna for high gain millimeter wave systems. *Scientific Reports*, 15(1):12056, 2025.

[24] F. Vipiana, G. Vecchi, and P. Pirinoli. A multiresolution system of rao–wilton–glisson functions. *IEEE Transactions on Antennas and Propagation*, 57(7), July 2009.

[25] G. Xu, V. G. Ataloglou, S. V. Hum, and G. V. Eleftheriades. Extreme beam-forming with impedance metasurfaces featuring embedded sources and auxiliary surface wave optimization. *IEEE Access*, 10:31577–31591, 2022.

[26] M. Zucchi. *Numerical Techniques for the Automated Design of Metasurface Antennas*. Doctoral dissertation, Politecnico di Torino, Turin, Italy, 2022. Doctoral Program in Electrical, Electronics and Communications Engineering (34^{th} cycle).

[27] M. Zucchi, F. Vernì, M. Righero, and G. Vecchi. Current based automated design of realizable metasurface antennas with arbitrary pattern constraints. *IEEE Transactions on Antennas and Propagation*, 71(6):4888–4902, June 2023.