


DEVELOPMENT OF NUMERICAL OPTIMIZATION TECHNIQUES FOR
OPTIMAL DESIGN OF NANOPHOTONIC AND NANOPLASMONIC SYSTEMS

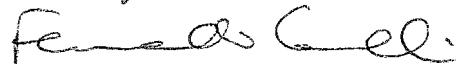
by

Carmen A. Caiseda
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Sciences and Informatics

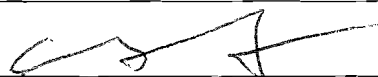
Committee:



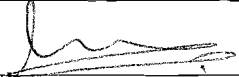
Dr. Igor Griva, Dissertation Director



Dr. Fernando Camelli, Committee Member



Dr. Chi Yang, Committee Member



Dr. Padmanabhan Seshaiyer,
Committee Member



Dr. Michael Summers, Director of School of
Physics, Astronomy, and Computational
Sciences



Dr. Timothy L. Born, Associate Dean for
Student and Academic Affairs, College of
Science



Dr. Vikas Chandhoke, Dean, College of
Science

Date:



Summer Semester 2012
George Mason University
Fairfax, VA

Development of Numerical Optimization Techniques for Optimal Design of Nanophotonic
and Nanoplasmonic Systems

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor
of Philosophy at George Mason University

By

Carmen A. Caiceda
Master of Science
University of Puerto Rico, 1991
Bachelor of Science
University of Puerto Rico, 1987

Director: Igor Griva, Associate Professor
Department of Computational and Data Sciences

Summer Semester 2012
George Mason University
Fairfax, VA

Copyright © 2012 by Carmen A. Caiseda
All Rights Reserved

Dedication

In loving memory of my father Dr. David Caiseda García, who departed on June 23, 2011, but in his weakness managed to be a joyful source of strength, encouragement, and blessings with wise simple words: “Dios te bendiga hasta que sobreabunde”, “pa’ lante”. He is my earthly reminder of the day by day progress and needed comfort I have received from the Heavenly Father, that heals through faith and trust.

I dedicate this dissertation to Adaljisa, Natanael y José Esteban who have become my family, source of perspective and support, as we weathered together the storms of threefold cancer, transplant, work and school work in the safe vessel of love.

Acknowledgments

I would like to thank my mentor and director Dr. Igor Griva, for his efficient, bright and common sense guidance, and the time he invested to teach and challenge me to do this work. To Dr. Vladimir Aksyuk, Center for Nanoscale Science and Technology, NIST for a fascinating research problem. The members of my dissertation committee: Dr. Fernando Camelli, Dr. Chi Yang and Dr. Padmanabhan Seshaiyer for their teachings, support, encouragement, and availability even in challenging times.

I am grateful the generous financial support from Google and John I. Wilson NEA Foundation through the Hispanic College Fund, and the Computational and Data Sciences department, directed by Dr. Papaconstantopoulos, and the most excellent graduate coordinator, Dr. Estela Blastein-Baroja a defender of what is right.

To my host Adaljisa Justiniano for her continuous financial, emotional and spiritual support throughout these five years.

The Inter American University of Puerto Rico has been a champion of professional development trusting in the investment in my education, for which I am happily in debt and eager to pay back and forward.

My family in PR are my constant source of inspiration and love. My mom Abu is my life pillar, my sons Ernesto and Gabriel are forever gifts from God, my daughters in law Millie and Yadixa added new love, and my precious grandchildren Gabriel, Beatriz and Andres are my joy.

I am amazed with the scientific knowledge of our times, that light shines and is reflected by bulk silver but is also absorbed by silver nanoparticles. This I feel is a valuable lesson learned from the Father of Lights who gives all good things so that His light over us like silver may be reflected toward others, but also absorbed within for transformation. I thank Him as the ultimate source of light and wisdom.

Table of Contents

| | Page |
|---|------|
| List of Tables | vii |
| List of Figures | viii |
| Abstract | x |
| 1 Introduction | 1 |
| 1.1 Previous Work | 2 |
| 1.1.1 Derivative Free Optimization | 4 |
| 1.1.2 Gradient Based Optimization | 5 |
| 1.2 Overview of Dissertation | 10 |
| 2 Theoreticall Background | 12 |
| 2.1 Maxwell's Equations | 12 |
| 2.2 Finite Element Method | 14 |
| 2.3 Nonlinear Programming | 19 |
| 2.4 Adjoint Method | 22 |
| 3 Problem Formulation | 24 |
| 3.1 State Equations: PDE | 25 |
| 3.2 FEM Discretization | 28 |
| 3.3 Complex Variables Decomposition | 30 |
| 3.4 Gradients of Functionals | 31 |
| 3.4.1 Numerical Gradients | 33 |
| 3.4.2 Gradient formulation | 34 |
| 3.4.3 Theoretical Results | 36 |
| 4 FEM solver | 38 |
| 4.1 Computational Domain | 39 |
| 4.2 The Uniform Grid | 40 |
| 4.3 Computational Implementation | 41 |
| 4.4 FEM Computational Results | 45 |
| 5 Maximum Light Absorption | 50 |
| 5.1 Introduction | 50 |
| 5.2 Optimization Criteria | 51 |

| | | |
|-------|---|-----|
| 5.3 | AMPL Model | 54 |
| 5.3.1 | Cubic Spline Permittivity | 56 |
| 5.3.2 | Results for 1D AMPL model | 58 |
| 5.3.3 | 2D AMPL Model | 59 |
| 5.4 | Gradient of $J(\lambda)$ | 66 |
| 5.4.1 | Computational Results | 67 |
| 5.5 | Gradient of $J(\epsilon)$ | 70 |
| 5.5.1 | Derivatives | 72 |
| 5.5.2 | Model 1 | 74 |
| 5.5.3 | Model 2 | 76 |
| 5.5.4 | Results | 77 |
| 6 | Optimal Design of Plasmon Coupler | 82 |
| 6.1 | Introduction | 82 |
| 6.2 | Formulation of the Problem | 83 |
| 6.2.1 | Designing the Geometry | 84 |
| 6.2.2 | The Objective Function | 89 |
| 6.3 | Plane Wave | 91 |
| 6.3.1 | Objective Function and Gradients | 91 |
| 6.3.2 | Adjoint Method | 97 |
| 6.4 | The Gaussian Beam | 100 |
| 6.4.1 | Objective Function and Gradients | 102 |
| 6.5 | Development of Computational Tools | 109 |
| 6.5.1 | Computer Implementation | 110 |
| 6.5.2 | Numerical Gradient | 120 |
| 6.6 | Optimal Design Results | 121 |
| 6.6.1 | Deformed Mesh Approach | 122 |
| 6.6.2 | Fixed Mesh Approach | 123 |
| 7 | Conclusions and Future Work | 131 |
| 7.1 | Derivation of formulas | 132 |
| 7.2 | Development of the 1D and 2D AMPL models | 133 |
| 7.3 | Numerical optimization of maximum light absorption functional | 134 |
| 7.4 | Numerical optimization of the Grating Coupler | 136 |
| 7.5 | Future Work | 139 |
| A | Appendix A | 141 |
| B | Appendix B | 154 |
| | Bibliography | 156 |

List of Tables

| Table | Page |
|--|------|
| 3.1 Electromagnetism parameters | 26 |
| 4.1 Grid size and J | 41 |
| 4.2 Time statistics | 43 |
| 5.1 AMPL Assembly Parameters | 59 |
| 5.2 AMPL 1D Model Performance | 60 |
| 5.3 Optimization Results | 64 |
| 5.4 Optimization Solver Statistics | 64 |
| 5.5 Delta J for lambda | 69 |
| 5.6 Absorption and norm(E) at center | 81 |
| 6.1 Initial Design Parameters | 85 |
| 6.2 Moving Mesh optimization | 122 |
| 6.3 Fixed Mesh optimization | 126 |
| B.1 In Plane TM mode properties | 154 |
| B.2 In Plane TM waves Constants | 155 |

List of Figures

| Figure | Page |
|---|------|
| 4.1 2D PDE domain | 40 |
| 4.2 Uniform Triangle Grid | 41 |
| 4.3 Refined Corners | 42 |
| 4.4 Comsol's Ex: Refined Unstructured Mesh | 46 |
| 4.5 Comsol's Ez: Refined Unstructured Mesh | 47 |
| 4.6 Ex in Coarse Grid | 47 |
| 4.7 Ex in Refined Grid | 48 |
| 4.8 Ex with average | 48 |
| 4.9 Ez in Coarse Grid | 49 |
| 4.10 Ez in Refined Grid | 49 |
| 5.1 Cubic Spline for $\Delta\lambda=1,2,5,10,20\text{nm}$ | 56 |
| 5.2 First Derivative of Cubic Spline: $\Delta\lambda=1,2,5,10,20\text{nm}$ | 57 |
| 5.3 Second Derivative of Cubic Spline: $\Delta\lambda=1,2,5,10,20\text{nm}$ | 57 |
| 5.4 Full 21 Node Uniform Mesh | 61 |
| 5.5 Half Uniform Mesh Symmetry | 61 |
| 5.6 Solution in full domain | 62 |
| 5.7 Solution in half domain | 62 |
| 5.8 Light Absorption | 63 |
| 5.9 Refined Grid J_c in $[300,400]$ nm | 64 |
| 5.10 Ex comparison for 900 sq nm rectangles I | 79 |
| 5.11 Ex comparison for 900 sq nm rectangles II | 80 |
| 5.12 Ey comparison for 900 sq nm rectangles I | 80 |
| 5.13 Ey comparison for 900 sq nm rectangles II | 81 |
| 6.1 Coarse vs Refined Mesh Visualization | 83 |
| 6.2 Initial Computational Domain | 85 |
| 6.3 Final groove design | 86 |
| 6.4 Design of Grooves | 87 |
| 6.5 Three Parameter Displacements | 88 |
| 6.6 Plane Wave Parameters | 91 |

| | | |
|------|---|-----|
| 6.7 | Gaussian Beam axis and radial propagation | 101 |
| 6.8 | Gaussian Beam Parameters | 102 |
| 6.9 | 27-cases for 3 parameters: Right, Left, Bottom | 114 |
| 6.10 | Multiphysics solution of moving mesh and magnetic field | 115 |
| 6.11 | parametrized Geometry boundary and mesh definitions | 116 |
| 6.12 | Coarse vs Refined Mesh Visualization | 117 |
| 6.13 | 139K and 557K elements meshes | 117 |
| 6.14 | No-grooves with plasmonic Mode | 118 |
| 6.15 | 4-grooves with Plasmonic Mode | 118 |
| 6.16 | 20-grooves with Plasmonic Mode | 119 |
| 6.17 | Final Design Solution for Plane Wave | 124 |
| 6.18 | Grating Final Design for Plane Wave | 125 |
| 6.19 | Optimization for Gaussian Beam Parameters Only | 126 |
| 6.20 | Design 1: Magnetic Field | 128 |
| 6.21 | Design 1: Grating | 128 |
| 6.22 | Design2 : Magnetic Field | 129 |
| 6.23 | Design 2: Grating | 129 |
| 6.24 | Best Design 4: Magnetic Field | 130 |
| 6.25 | Design 4: Grating | 130 |

Abstract

DEVELOPMENT OF NUMERICAL OPTIMIZATION TECHNIQUES FOR OPTIMAL DESIGN OF NANOPHOTONIC AND NANOPLASMONIC SYSTEMS

Carmen A. Caiceda, PhD

George Mason University, 2012

Dissertation Director: Dr. Igor Griva

There is a steadily growing interest in building new photonic and plasmonic nanosystems capable of tailoring the electromagnetic properties of light. An optimal design of these nanosystems is critical for their efficiency. One essential component of optimal design of nanodevices is numerical simulation and optimization that provide the optimal structure to be tested experimentally, and eventually implemented as actual device. This dissertation focuses on the development of numerical optimization techniques to analyze and design efficient nanoplasmonic and nanophotonic systems. In this work the electromagnetic field is modeled through the numerical solution of Maxwell's equations in the frequency domain, and numerical techniques that address optimization problems with these PDE constraints are developed. Application of the techniques to problems of i) maximization of light absorption by metal nanoparticle and ii) efficient surface plasmon generation demonstrate considerable practical value of the developed methodology

No preferred strategy has yet emerged from the nanophotonic research community to solve optimization problems with partial differential equation constraints, despite continuous theoretical developments in topology and shape optimization, large-scale nonlinear optimization and sensitivity analysis. This dissertation considers two approaches to the problem.

The first approach is to discretize and incorporate the PDE into a constrained optimization problem to solve with an appropriate nonlinear programming algorithm. The second optimization approach is to formulate and compute the gradient and modify the parameters accordingly, using the current data and PDE solution obtained from the solver. The first approach has been implemented in AMPL modeling language for problem i). The second numerical optimization approach is the main strategy implemented for both problems i) and ii) using COMSOL Multiphysics and MATLAB.

Although this effort to solve a design optimization problem is specific to nanophotonic/nanoplasmonic systems, the result of this work afford computational tools with broader applications to advance the wider problem of optimization with PDE constraints.

Chapter 1: Introduction

Nanophotonics and Nanoplasmonics study the behavior and properties of light, and its interaction with matter on the nanoscale. The design and study of nanophotonic and nanoplasmonic (nph/npl) systems have stirred the research community since its recent origins because of its potential of pushing the limits of technology to new levels. The foundational research in photonic crystals (PC) was done by E. Yablonovitch [1] at Bell Communications Research and S. John [2] at Princeton University during 1987. Their work has produced periodic structures that are able to inhibit the propagation of certain frequencies of light to a desired behavior, a successful result of optimal design. Today researchers use these mechanisms to control the dispersion, deceleration, and storage of light. Also in nanoplasmonics there is much space for invention. Nanoplasmonics is a branch of nanophotonics that studies surface plasmons, a collective oscillations of electron coupled to an electromagnetic field in a metal or semiconductor. Surface Plasmon (SP) polaritons are quasi-particles, hybrid of the photon and electron plasma. SP modes are the electromagnetic waves that propagate along the interface of metal and dielectric. SP's interaction with light can enhance absorption, scattering and near-fields by modifying the surface of the conductor. This interaction allows manipulation of light in new ways, with startling applications.

Nph/npl phenomena is not only of interest to electromagnetism (EM), an interdisciplinary research community, but also to the general population because of its potential for applications hitherto only in the purview of science fiction. The creation of new materials and photonic devices is common to optics, magneto-optic data storage, microscopy, solar cells and biology (where it is used to detect targeted molecules). Two popular applications that have captured the public imagination are the cloak of invisibility, and the possibility of creating a superlens. By designing a cloaking device with a particular configuration of permeability (μ) and permittivity (ϵ) values, a covered object can be invisible to an EM wave.

Materials with negative refractive index, $\nu = \pm\sqrt{\epsilon\mu}$, permit the creation of superlenses with spatial resolution below the optical wavelength. A metamaterial-based superlens can render images with better resolution than the diffraction limit by locating the objective lens hundreds of nanometers (near field) from the object.

The contribution of the computational sciences to the development and study of optimal design tools is fundamental. The Committee on Nanophotonics Accessibility and Applicability of the Division on Engineering and Physical Sciences, National Research Council of the National Academies in its citation from the 2008 document *Nanophotonics Accessibility and Applicability* [3] states that:

Quantitative and computationally intensive EM modeling has assumed a dominant and increasingly important role in this field, employing fully numerical methods such as finite element, boundary element, and finite difference-time domain approaches; analytical or semi-analytical methods such as plasmon hybridization and discrete dipole approximation; and lumped circuit concepts for the design of complex plasmonic systems.

To better develop these kind of applications, efficient use of simulation and optimization algorithms is needed. The effort to develop tools for optimal design of electromagnetic devices not only enables the creation of more efficient nanodevices, but also new conceptual discoveries.

1.1 Previous Work

Optimal design in EM is a PDE constraint optimization problem. A survey of the PDE or field solvers and optimizers used in recently published work is profitable to assess the state of the art in this field. Numerical techniques used to solve the EM field or PDE solvers include:

1. Finite Difference (FD): FD in the optimization of aperiodic structures [4], and finite difference time domain (FDTD) in Iterative Optimization [5]

2. Finite Element Method (FEM): FEM codes include COMSOL and Omega3P in optimization of photonic nanostructures in [6], [7],[8],[9] and
3. Fourier-Bessel series: aperiodic nph design in [10].

Numerical optimization techniques for nanoplasmonic and nanophotonic systems that satisfy partial differential equations constraints include:

1. Genetic Algorithms: for photonic crystals [11]
2. Robust Optimization: using gradient information on a npl grating design [12]
3. Method of Moving Asymptotes: with topology and shape optimization in [6], [7]
4. Iterative Optimization: for plasmon resonant nanostructure in [5]
5. Guided Random Walk: in aperiodic nph design [10] and
6. Sequential Quadratic Programming (SQP): deterministic optimization in shape optimization of low-loss cavity for the international linear collider in Physics in [9].

Recently, optimization with the Adjoint Method (AM) is present in [6], [7], [8], [4], [12], [9], and has become popular to find the gradient with respect to design variables. AM is preferred for large-scale optimization over automatic differentiation, numerical differentiation or direct sensitivity analysis as seen in [9]. The following review of previous work in EM show a variety of numerical techniques implemented in optimal design. The choice of numerical PDE-field solvers and numerical optimization algorithms vary in this list, showing that there is no universally preferred tool for EM optimal design problems. The following review of previous work classify nph/npl optimal design according to use of gradient-based NLP algorithms. The numerical optimization technique or optimizer implemented is described, as well as methods used to model the field or PDE solvers, and to find and evaluate the objective function and its gradient.

1.1.1 Derivative Free Optimization

The Derivative Free optimization approach iteratively runs the PDE solver over changed parameters to find the best design without using gradient or derivative based optimization in [11], [10]. A very common gradient free optimization algorithm is the Genetic Algorithm. The technique starts with a candidate solution that evolves by choosing parameters that improve the at each iteration regarding an objective criteria.

Jiang L. et al. in [11] use a Genetic Algorithm, a stochastic search algorithm from evolutionary computation, to optimize the design of a photonic crystal waveguide with grating-like surface in order to maximize highly-efficient directional emission. The FDTD method is used to calculate the efficiency of the directional emission as a fitness value for each chromosome consisting of a binary list of parameters that generate the grating-like surface. The list contains from four to eight parameters. Selection creates a new generation with more positions in the new population for chromosome with good fitness values, and eliminating the ones with poor values. The determined fitness level of each chromosome, by FDTD, is used to find the probability of a chromosome to be selected. On the other hand crossover produces two children that share the chromosomes of both parents' genetic material using a single point crossover at a position determined as a random integer. Mutation is used to guarantee that a particular good characteristic is not lost through selection and crossover, and non-zero probability of searching for a given string. The algorithm is repeated until a predefined number of generations is obtained and decoded as an optimum design for the grating-like surface.

Gheorma, I.L. et al. in [10] study the photon scattering properties of aperiodic nanoscale lossless dielectric structures in two configurations resembling photonic crystals (PC) structures: lossless dielectric rods in air and circular holes in a dielectric. The use of adaptive algorithms for device design of photonic crystal structures seeks to match desired EM transmission and scattering properties. The 2D EM field solver uses the analytical solution of Helmholtz equation by separation of variables in polar coordinates and expresses it and the Gaussian input beam as a Fourier-Bessel series. The optimization strategy is an iterative

solution of inverse problem Guided Random Walk. The position of individual cylinders is randomly changed by a small amount. The scattered fields are modified, recalculated, and accepted if the result is closer to the target function and within the error tolerance, and rejected otherwise. The target function involves redirecting and reshaping an EM beam using optimized aperiodic nanostructures. The parameter considered in the sensitivity of the design is the change in position of the cylinders. This adaptive algorithm approximates the distribution of the target function dependent of the number of cylinders, and therefore is limited by increased computational effort as the number increases.

1.1.2 Gradient Based Optimization

Gradient based optimization require more computationally intensive information on the objective function in exchange for a better convergence to the optimal solution. NLP algorithms that use the Hessian are not found in this review of published work on nph/npl design. The NLP algorithms used include the GM and SQP. The method of moving asymptotes (MMA), developed in 1987 by Svanberg [13] for structural optimization problems, is used in Frei et al. [6] and [7].

In [5] Pavaskar et al. use finite difference time domain (FDTD) in 2D cylinder configuration to determine the configuration of nanoparticles that give a maximum electric field intensity in the center of the cluster. The algorithm is summarized as follows:

1. Determine resonance frequency of initial configuration with a FDTD simulation with broad pulse and initialize parameters.
2. Calculate electric field intensity gradients in the x, y directions.
3. Move one nanoparticle in the direction of the electric field intensity gradient provided there is no overlap with other nanoparticle.

4. Determine resonant wavelength of new configuration and run FDTD simulation of the electric field for the entire grid.
5. Compare electric field intensity at the center of the cluster and determine if the new value is the new maximum or discard.

The algorithm converges after five iterations using three nanoparticles. The maximum intensity can be achieved by placing the third nanoparticle either at the top or at the bottom of the center of the cluster. The electric field intensity increased monotonically and the resonant wavelength changes slightly by a maximum of 2 %. This approach permits the authors to add one nanoparticle at a time until they reach a 20 nanoparticle configuration. When comparing the optimal configuration with a linear or chain positioning, the field intensity increases by a factor of 2500. In practice the following proved helpful: a list of bad nanoparticle configurations to avoid visiting twice, a modified step size to achieve convergence, and allowing downhill movement in the optimization to avoid local optima. Because this is implemented in 2D the algorithm requires roughly five minutes of computation time to move one nanoparticle.

W. R. Frei et al. in [6], [7] use Topology Optimization to improve the field in a photonic crystal structure with a photonic band gap (PBG) waveguide. This numerical technique uses the adjoint method (AM) to solve a nonlinear programming problem and design sensitivity analysis by the FEM to improve photonic crystal structures. A waveguide is created by defining an area with a dielectric rod lattice structure. The area with no rods is the waveguide while the rods prohibit light from propagating in any other direction. The performance of a compact waveguide termination in a square lattice rods-in-air photonic crystal is increased five-fold in power incident upon the target area over the termination through the use of topology optimization. This numerical optimization method is a three step procedure.

1. Solve the differential equation and define the objective function.

2. The sensitivity of the objective function to the solution is assessed at the computed distribution of the material within the domain.
3. The material distribution is adjusted based on the sensitivity analysis.

This process is repeated until the objective function reaches the target function. The objective is to maximize the amount of power crossing the boundary at a given distance from the waveguide termination. This indicates greater directional emissivity at the optimized waveguide termination area compared to the strong diffraction in the non-optimal design. The PDE is a Helmholtz equation with second order absorbing boundary condition solved with the frequency domain FEM. Sensitivity of the objective function, i.e. the derivative with respect to the set of design parameters, is obtained through the AM. The design parameters or controls are the dielectric constant as a function of position $\epsilon_r^m(\vec{x})$, within the m elements of the FEM model. After obtaining the objective function and its derivative with respect to the dielectric constants in the elements, ϵ_r^m are modified to improve the objective function. A non-linear constraint optimization problem is formulated from the obtained objective function and its derivative with bounded dielectric constants. A numerical optimization package is used to implement the Method of Moving Asymptotes (MMA) optimizer to obtain the modified dielectric constants. The MMA is a deterministic optimization algorithm that proceeds modifying the design α variable bounded by shrinking asymptotes, $L_k \leq \alpha_k \leq U_k$, to maximize the solution until a suitable termination criteria is met. The authors report that the computational implementation of the topology optimization runs in a desktop computer for less than an hour, using linear quadrilateral element in FE and 100 iterations of the optimization algorithm. Second order quadrilateral elements increase run-time by more than a factor of four. Because the deterministic optimizer search depends on an initial topology, many other local optima are expected. Interpretation of the design is an open problem, [14].

The Akcelik et al. team in [9] works in the development of the International Linear Collider (ILC) for high-energy physics research design to create high-energy particle collisions

between electrons, positrons, and related antimatter. This shape optimization problem governed by a large-scale EM eigenvalue problem uses SQP optimization, AM gradients and FEM solver. The shape of the accelerator cavity is optimized to damp high-order dipole modes (HOMs). Because of the high accuracy requirements of the frequency constraint, gradient-based methods are used to solve the optimization problem. The design variables to find the derivative of the objective function and constraints are hundreds of computer aided design (CAD) parameters for the accelerator cavity shape. The best approach to compute gradients for this problem is the continuous AM. This method scales only with active constraints, much fewer in number than the design variables. It also avoids differentiation through the mesher that might be non-differentiable from CAD design. The design problem is to maximize stored energy of the trapped modes in an end cell, subject to the constraint of the cavity shape satisfying a desired accelerating frequency. Here the state eigenvalue problem gives the relation between the design variables and eigenpairs. In order to use highly-efficient SQP, the gradient of the objective function and the accelerating frequency constraint with respect to design variables are obtained to solve the optimization problem. Since the trapped modes frequency are expected to be well-separated, the adjoint eigenvalue problem is non-singular and the gradient is continuous. The adjoint eigenpair is a linear self-adjoint system. In the computational implementation, four variables are chosen to parametrize the shape corresponding to the outer layer of elements. The state eigenvalue problem is solved by Omega3P finite element electromagnetism code automatically, obtaining the trapped mode and accelerating mode. The adjoint eigenvalue problem is solved by an adjoint extension to the Omega3P using the same mesh, elements, and numerical components in the state eigenproblem. Extensions to Omega3P are also used to compute the objective, accelerating frequency constraint, objective gradient, and constraint gradient expressions involving boundary and domain integrals. The function, gradient information, and bounds on design variables are passed to the SQP code *fSQP* to solve a quadratic subproblem. These generate a search direction, perform a line search, and return new design

vector. The updated design variables generate the deformed mesh and the process is repeated until convergence. The dominant computational cost is the solution of the Maxwell’s (state) eigenvalue problem solution for frequencies and mode shapes, over solving the linear adjoint eigenvalue problem and remeshing. This processes depend on volume computations. The algorithm converges in five iterations where the objective value of the trapped energy in the end cell is improved by 58%, even when the initial design is not feasible violating the accelerating frequency constraint by 17%.

Bertsimas et al. in [12] use Robust Optimization to solve the EM scattering problem and optimization of aperiodic dielectric structures on lowest mode TE_{10} . Robust optimization is applied to a non-convex objective function to mimic a power distribution of a target surface. The objective of the robust optimization is to “minimize the worst case cost under implementation error from manufacturing or experiments.” It minimizes the robust or worst case cost over a geometric 2D position $p \in P$, the admissible set. That is minimize $\max_{\Delta p} J(p + \Delta p)$, where Δp is the implementation error in uncertainty set $U = \{\Delta p \mid \|\Delta p\| \leq \Gamma = 550\mu m\}$. U is the set of implementations errors to avoid in the design. The advantage of robust optimization is that it only needs to evaluate the objective function. Commercial optimization solvers are used to find the solution of the second order cone problem that describes the search of a good direction d . The direction d is one that minimizes the angle between the robust point and previously visited points in the neighborhood of p_k with maximum cost values. The EM field is modeled by the Helmholtz equation for lossless dielectric scatterers with Dirichlet condition on metallic horn and perfectly matched layers (PML) boundary conditions, and the solution is found with finite-difference (FD). Power on surface is computed with interpolation for nearest four mesh points and standard Gaussian weight. Linear Algebra Packages are used to solve the system of equations via LU decomposition with UMFPACK, and Goto-BLAS is used for vector and matrix operations. The computational implementation takes 0.7 seconds on ab Intel Xeon 3.4 GHz for 70K x 70K matrix. The optimization procedure consists of two steps: nominal and robust optimization. The

nominal optimization problem finds a good configuration as initial solution for step two. A random search does not produce fast improvement, therefore, two alternative algorithms are tested: a gradient free stochastic algorithm and a modified gradient descent using the AM to compute the gradient. The advantage of the adjoint equation is the efficient use of the same structure and linear operators of the linear system for FD PDE operator producing a gradient evaluation optimization at very little additional computational cost. The modified gradient descent converges faster. The modifications consist of the following: if the configuration is infeasible, the step size is halved until it meets a threshold value. Thus the modified gradient will apply a gradient step only to feasible configurations. The robust optimization step iteratively updates the configuration with one more robust (minimum worst case) until a termination condition is met. Since the robust objective function $\max J(p + \Delta p)$ does not have a closed form, efficient local searches are used within a neighborhood of the initial configuration p .

1.2 Overview of Dissertation

This dissertation is organized in the following way. In chapter 2 the theoretical background related to the problem of optimal design of nph/npl systems is introduced. In Chapter 3 the problem formulation is developed from basic principles, including a theoretical exposition of the ways to obtain the gradient in the context of the dissertation problem, and discusses equivalence of diverse formulations. The following chapters present the development and computational application of the two main problems in this dissertation. Chapter 4 presents the computer implementation of the FEM solver and its results. In Chapter 5 the maximum light absorption problem is developed from formulations of objective function and its gradients, to the computer implemented in AMPL and MATLAB. The main problem of the optimal design of the plasmon coupler is presented in Chapter 6. The development of the model, formulations, numerical optimization strategies and implementation in COMSOL-MATLAB code, is developed for two different electromagnetic waves. The plane wave and the Gaussian beam devices are studied and results are presented. The final Chapter 7 is a

summary of results, lessons learned, and options for further development of this work.

Chapter 2: Theoretical Background

This section summarizes the theoretical background that is relevant to this work. In this dissertation the EM field is modeled by partial differential equations (PDE) obtained from Maxwell's Equations subject to some modeling assumptions and constitutive relations. In the optimization problem, these equations become the constraints that the state variables must satisfy. To find the numerical solution of the PDE obtained from Maxwell's Equations, this dissertation uses the Finite Element Method (FEM) algorithm. In addition, the project will use numerical methods to find the gradient information in order to perform optimization. A summary of non linear programming (NLP) algorithms used in large-scale optimization problems is included. Finally the Adjoint Method (AM) is discussed as a viable way to find the gradient of the objective or cost function.

2.1 Maxwell's Equations

Maxwell's Equations govern EM phenomena. The PDE used to model EM fields is obtained from Maxwell's equations under simplifying assumptions that define the model of the phenomena. Maxwell's equations in differential form are:

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \text{ Ampere-Maxwell's Law} \quad (2.1)$$

$$-\nabla \times \mathbf{E} = \frac{\partial \mathbf{B}}{\partial t} \text{ Faraday's Law} \quad (2.2)$$

$$\nabla \cdot \mathbf{D} = \rho \text{ Gauss Law-Magnetism} \quad (2.3)$$

$$\nabla \cdot \mathbf{B} = 0 \text{ Gauss Law.} \quad (2.4)$$

Here:

\mathbf{E} = electric field intensity (volts/meter)

\mathbf{D} = electric displacement field or flux density (coulombs/meter²)

\mathbf{H} = magnetic field intensity (amperes/meter)

\mathbf{B} = magnetic flux density (webers/meter²)

\mathbf{J} = current density (amperes/meter²)

ρ = electric charge density (coulombs/meter³).

For the light propagation in a linear dispersion-less time-invariant material the following assumptions are made along with the constitutive relations:

- $\mathbf{J} = 0$, there is no current.
- $\rho = 0$, No charges introduced to the field.
- $\mathbf{D} = \epsilon \mathbf{E}$, linearity.
- $\mathbf{B} = \mu \mathbf{H}$, linearity of Ampere-Maxwell's equation .
- For harmonic light propagation, $\mathbf{E}(x, y, z, t) = \mathbf{E}(x, y, z)e^{j\omega t}$ and $\mathbf{H}(x, y, z, t) = \mathbf{H}(x, y, z)e^{j\omega t}$.
- $\mu_r = 1$ for high frequency light wave.

The parameters of the dielectric material are ϵ : permittivity(farads/meter) and μ : permeability (henrys/meter). The general Maxwell's equations are then simplified to the following:

$$\nabla \times \mathbf{H} = \frac{\epsilon \partial \mathbf{E}}{\partial t} \quad (2.5)$$

$$-\nabla \times \mathbf{E} = \frac{\partial \mathbf{B}}{\partial t} = \mu \frac{\partial \mathbf{H}}{\partial t} \quad (2.6)$$

$$\nabla \cdot \mathbf{D} = \nabla \cdot \epsilon \mathbf{E} = 0 \quad (2.7)$$

$$\nabla \cdot \mathbf{B} = 0. \quad (2.8)$$

By assuming time harmonic light propagation, the derivative with respect to t of \mathbf{E} and \mathbf{H} is obtained in equations (2.5) and (2.6) that become:

$$\nabla \times \mathbf{H}(r)e^{j\omega t} = j\epsilon\omega\mathbf{E}(r)e^{j\omega t}, \quad \nabla \times \mathbf{E}(r) = -j\mu\omega\mathbf{H}(r),$$

taking $r = (x, y, z)$. By substituting out \mathbf{H} in equation (2.5) a partial differential Helmholtz equation in space for \mathbf{E} is obtained:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{E}(r) \right) = \epsilon\omega^2\mathbf{E}(r).$$

One way to model an artificial boundary that simulates an open domain is to use absorbing boundary conditions [15]. These are defined as

$$\vec{n} \times (\nabla \times \mathbf{E}) - jk\vec{n} \times (\mathbf{E} \times \vec{n}) = -\vec{n} \times (\mathbf{E}_0 \times j(k\vec{n} - \vec{k}))e^{-j\vec{k} \cdot \vec{r}},$$

where \vec{n} is the normal vector at the boundary, \mathbf{E}_0 the source, and the wave number $k = \left(\frac{2\pi}{\lambda}\right)$. Analytical solutions for this PDE's with complex boundary conditions are rarely available. Consequently, numerical solutions need to be obtained. Efficient numerical PDE solvers can be developed using FEM, and will be discussed in the next section.

2.2 Finite Element Method

The FEM is an numerical tool used extensively in computational EM and simulation research. Popular commercial PDE solvers, like COMSOL, are FEM solvers. Currently, FEM and computational processing combine to tackle more complex problems and better model phenomena described by PDEs.

The solid mathematical foundation behind FEM support the existence of the solution and stability of the method for numerical simulations. The mathematical framework is well

developed using calculus of variations and weak formulation of PDE's in functional analysis. The Lax-Milgram Theorem establishes the existence and uniqueness of a solution in an infinite dimensional Hilbert space. The requirements for stability are automatically met by FEM based in energy estimates, unlike other numerical PDE solvers like Finite Differences (FD). Although a thorough mathematical treatment of Maxwell's equations has been developed in the study $H(Curl, \Omega)$ Sobolev Spaces, this proposal focuses in the development of numerical methods. The solution is a discrete approximation to the continuous solution by using a finite dimensional ansatz basis for a finite dimensional space $V \subset H(Curl, \Omega)$. The numerical solution belongs to the finite dimensional ansatz space.

The variational or weak formulation of the PDE in V an infinite Hilbert space is used to derive the discretized ansatz space problem. The variable $E \in V$ is approximated by $E_h \in V_h \subset V$ an n -dimensional finite subspace with chosen basis functions $\psi_k(z)$, $k \in 1, \dots, n$. Assuming for example that $n = 2$, then $\psi_1(z), \psi_2(z)$ and $E_h(z) = E_1\psi_1(z) + E_2\psi_2(z)$. After applying the essential boundary conditions (Dirichlet), and integration by parts (that accounts for the natural-Neumann boundary conditions), a linear system is obtained for each element. As an example a four step derivation is considered, for a general 1D elliptic problem in $u : (0, 1) \rightarrow \mathbb{R}$, with given functions $a, f, r : (0, 1) \rightarrow \mathbb{R}$, given by

$$(-au')' + ru = f, \quad u \text{ in } (0, 1)$$

$$u(x) = 0, \quad \text{on } x = 0, x = 1 \tag{2.9}$$

1. Variational formulation: Define $V = \{v : (0, 1) \rightarrow \mathbb{R} : \int_0^1 (v^2(x) + v'(x)^2) dx < \infty, v(0) = v(1) = 0\}$. Multiply equation (2.9) by a test (weight) function $v \in V$ and integrating by parts to obtain

$$\begin{aligned}
\int_0^1 f v \, dx &= \int_0^1 (-au')' v + ruv \, dx = \\
&= [-au'v]_0^1 + \int_0^1 au'v' + ruv \, dx \\
&= \int_0^1 au'v' + ruv \, dx
\end{aligned} \tag{2.10}$$

The variational formulation for (2.9) is

$$A(u, v) = L(v), \quad \forall v \in V \tag{2.11}$$

with

$$\begin{aligned}
A(u, v) &= \int_0^1 au'v' + ruv \, dx \\
L(v) &= \int_0^1 f v \, dx.
\end{aligned} \tag{2.12}$$

To obtain from the variational formulation a solution to (2.9) some assumptions on the smoothness of a, r, f must be made so that $-(au')' + ru - f$ is continuous. Then working from (2.11) the result is (2.9). Using integration by parts

$$0 = \int_0^1 au'v' + ruv - f v dx = \int_0^1 (-au')' + ru - f) v \, dx,$$

implies $\forall v \in V$ the desired solution $-(au')' + ruv - f = 0$.

2. Finite Dimensional ansatz space: V_h can be obtained by dividing $(0, 1)$ into N elements

with nodes $0 = x_0 < x_1 < \dots < x_N = 1$. The discrete space is defined as

$$V_h = \{v \in V : v(x)|_{(x_i, x_{i+1})} = c_i x + d_i, i = 1, \dots, N, v \in C(0, 1)\},$$

continuous piecewise linear functions. The problem is to find $u_h \in V_h$ such that

$$A(u_h, v) = L(v), \tag{2.13}$$

the finite dimensional approximation of the solution u of (2.9).

3. Choose a basis for V_h : Basis functions $\psi_i \in V_h$ for $i = 0, \dots, N$ satisfy:

$$\psi_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$

Therefore $\forall w \in V_h$

$$w(x) = \sum_{i=0}^N w_i(x) \psi_i(x),$$

a linear combination of the basis functions ψ_i .

4. Solve the discrete problem: By choosing the basis function $\psi_j = v$ in (2.13), and

$$u_h(x) = \sum_{i=0}^N c_i \psi_i(x)$$

then

$$L(\psi_j) = A(u_h, \psi_j) = A\left(\sum_i c_i \psi_i, \psi_j\right) = \sum_{i,j} A(\psi_i, \psi_j).$$

This is the linear system to be solved

$$\hat{A}\hat{c} = \hat{L}, \quad (2.14)$$

with

$$\hat{A}_{j,i} = A(\psi_i, \psi_j),$$

$$\hat{L}_j = L(\psi_j),$$

$$\hat{c} = (c_0, \dots, c_N).$$

The obtained $N \times N$ matrix \hat{A} is the stiffness matrix, and the vector \hat{L} the load vector that characterize the discretized problem.

The computer implementation of these methods give rise to various degrees of accuracy and performance. A higher degree polynomial used as basis functions gives better accuracy. The refinement of the mesh on areas of interest provide more information on the distribution and properties of the computed field. Symmetries of the domain can also enhance FEM performance on a particular problem by reducing the computational domain to a fraction of the original problem, saving time and space.

In electromagnetism the finite elements in the frequency domain has been used to produce efficient models of the EM field. In the optimal design of nph/npl devices, the FEM solution of the Helmholtz PDE determine the state variables of the electromagnetic field or state problem. An optimization problem is defined using the FEM computed field as constraints of a large-scale nonlinear programming (NLP) optimization. The following section discusses NLP algorithms.

2.3 Nonlinear Programming

Large-scale simulations that model reality more accurately, and PDE solvers that are faster and more efficient stir the development of efficient optimizers capable of finding the optimal solution of simulation-based problems. Nonlinear Programming (NLP) tools are efficient algorithms with the potential for solving optimal design problems. The theory and application of NLP to optimization with PDE constraints is a new area of study with many challenges to explore. In this section some of the most efficient NLP algorithms are described that are needed to be able to solve large-scale optimization problems. However the discussion refers to finding the optimal $x^* \in R^n$ a finite metric space, rather than $u \in U$ an infinite Banach space. The existence and regularity theory to generalize the optimization theory to Banach spaces with semi-linear elliptic PDE is developed in the book by Hinze et al. *Optimization with PDE Constraints* [16].

NLP is the study of optimization involving nonlinear functions:

$$\begin{aligned} &\text{minimize} && J(x) \\ &\text{subject to} && g_i(x) = 0 \\ & && c_j(x) \geq 0 \\ & && i = 1, \dots, r, \quad j = 1, \dots, m \end{aligned}$$

Nonlinear optimization methods can be broadly classified into two categories: methods that use the gradient of the objective functions and methods that do not use the gradient. Methods that use the gradient are the focus of interest in this section. The Gradient Method (GM) is the most basic method in this classification. GM has a very poor convergence behavior comparable to a zigzag path toward the minimum. Methods that use the Hessian, like Newton-based methods converge rapidly near the solution and are preferred. Nonetheless, the Optimal Gradient Method has brought the attention of the simulation research community back to GM. This responds to the fact that finding the Hessian for large-scale simulation problems might be problematic.

The general unconstrained minimization problem is the starting point of the study of constrained gradient-based optimization. The unconstrained problem is to minimize $f(x)$, and the maximum can be obtained by minimizing the opposite of the objective function. The algorithm seeks the minimizer of a nonlinear function $f(x)$ in a local neighborhood. Note then that a numerical optimization algorithm searches for a local versus global minimum because it considers a finite number of points. Global minimization requires information of the function at all points.

Numerical optimization algorithms for unconstrained minimization can be classified by the use of gradient or Hessian in the computation of the descent direction. In its most general form, numerical optimization algorithms search for a direction p_s such that in the least number of iterations s , the update $x_{s+1} = x_s + tp_s$ will reach the minimum of $f(x)$ within a given tolerance. The strategies used for the algorithm to convergence, to find the minimizing direction p , and the step size t define each method. If $f(x) \in C^1$ is assumed, the gradient can be computed to apply Gradient method (GM). In the GM the direction of descent is taken as the anti-gradient $p = -\nabla f(x)$. GM is defined by the iteration: $x_{s+1} = x_s - t_s \nabla f(x_s)$, $s = 1, 2, \dots$. If f is strongly convex the convergence rate of the algorithm is linear, that is $\|x_{s+1} - x^*\| \leq \gamma \|x_s - x^*\|$, $0 < \gamma < 1$. Furthermore under C^2 smoothness assumption, the first-order necessary condition and second-order sufficient conditions apply for x^* to be a local minimizer. These are derived from Taylor series $f(x^* + p) = f(x^*) + \nabla f(x^*)^T p + \frac{1}{2} p^T \nabla^2 f(\xi) p$. If x^* is a local minimum, then it must be a stationary point and thus $\nabla f(x^*) = 0$. Note that this first-order condition is not sufficient since a stationary point can also be a saddle-point or a local maximum. The second-order sufficient condition states that x^* is a strict local minimizer if $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$, positive definite matrix. For a positive semi-definite matrix x^* will be a candidate local minimizer only. Also in computer implementation the condition $\nabla f(x) = 0$ must be adapted to a given tolerance.

The treatment of constrained optimization follows by introducing the Lagrangian and Lagrange multipliers and performing unconstrained minimization of the Lagrangian. The

constrained minimization problem with equality constraints is

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && g_i(x) = 0 \\ & && i = 1, \dots, r. \end{aligned}$$

The Lagrangian is defined : $L(x, y) = f(x) - \sum_{i=1}^r y_i g_i(x)$, where y_i are the Lagrange multipliers or dual variables. The Lagrange multipliers are important not only because of its mathematical significance in the dual functional space but also because of the information on the sensitivity of the optimal objective value. Lagrange multipliers express the gradient at the optimum as a linear combination of the rows of the constraint matrix A . These also indicate the sensitivity of the optimal objective value to changes on the problem data or perturbation. The optimality conditions for constrained optimization are obtained from the gradient of the Lagrangian with respect to both x, y as follows

$$\begin{cases} \nabla_x L(x, y) = \nabla f(x) - \nabla g^T(x) y = 0 \\ \nabla_y L(x, y) = g(x) = 0, \end{cases}$$

where ∇g is the Jacobian of the constraints.

Sequential Quadratic Programming (SQP) as implemented in solvers like SNOPT, is other type of constrained nonlinear programming method that has been used in the large-scale optimization with PDE. The purpose is to obtain the descent direction by solving a problem with quadratic objective function and linear constraints. Consider the Lagrangian $L(x, y) = f(x) - \sum_{i=1}^r y_i g_i(x)$, with Lagrange multipliers y_i and equality constraints $g_i(x)$. The quadratic programming is obtained from the optimality conditions and Newton's method. The quadratic problem formulation is:

$$\begin{aligned} & \text{minimize}_{\Delta x} && 0.5 \Delta x^T \nabla_{xx}^2 L(x, y) \Delta x + \nabla_x L(x, y)^T \Delta x \\ & \text{subject to} && \nabla g(x) \Delta x + g(x) = 0 \end{aligned}$$

to obtain Δx , the primal variable update: $\hat{x} = x + \Delta x$.

NLP algorithms provide the tools to solve constrained optimization problems. The gradient and Hessian are important tools for an efficient NLP algorithm to find the optimizer. Therefore, in the next section one of the forms to compute the gradient using available information from the PDE simulation is discussed.

2.4 Adjoint Method

The previous section discussed the importance of computing the gradient in order to apply the most efficient NLP algorithms. Recent published work show that the Adjoint Method (AM) has been favored in the nph/npl optimal design because of the efficient use of simulation data.

AM is a very efficient way of finding the gradient when using a PDE solver like FEM because the gradient and PDE field variables can be simultaneously obtained. The general formulation of the optimization with PDE constraints is

$$\begin{aligned} & \text{minimize} && J(\alpha, u) \\ & \text{subject to} && A(\alpha, u(\alpha)) = 0, \end{aligned}$$

where $\alpha \in \mathbb{R}^d$ are design variables, $u \in \mathbb{R}^n$ the state variables, and $A(\alpha, u(\alpha)) = 0$ the PDE. AM calculates the gradient with respect to the d design variables by starting with the first variation or directional derivative of the objective function as follows:

$$\delta J = \lim_{h \rightarrow 0} \frac{J(\alpha + h\beta) - J(\alpha)}{h} = D_u J(\alpha, u) \delta u + D_\alpha J(\alpha, u) \beta.$$

Here $D_u J$ is the gradient. The variation of the PDE constraint is: $D_u A(\alpha, u) \delta u + D_\alpha A(\alpha, u) \beta = 0$. Including the constraint variation times the Lagrange multiplier p in δJ :

$$\delta J = [D_u J(\alpha, u) - p \cdot D_u A(\alpha, u)] \delta u + [D_\alpha J(\alpha, u) - p^T \cdot D_\alpha A(\alpha, u)] \beta.$$

If p is chosen such that it satisfies the adjoint equation $D_u J(\alpha, u) - p^T \cdot D_u A(\alpha, u) = 0$ the first variation reduces to the *Gâteaux* derivative

$$\delta J = \lim_{h \rightarrow 0} \frac{J(\alpha + h\beta) - J(\alpha)}{h} = [D_\alpha J(\alpha, u) - p^T \cdot D_\alpha A(\alpha, u)]\beta,$$

for all direction β . The gradient thus defined:

$$\nabla J = D_\alpha J(\alpha, u) - p^T \cdot D_\alpha A(\alpha, u).$$

By FEM the PDE is discretized as a linear system $A(\alpha)u(\alpha) = f(\alpha)$, with $A : N \times N$ matrix, and p, u : vectors of dimension N . Using the same matrix data, the adjoint is computed from the similar system $A(\alpha)p(\alpha) = D_u J(\alpha, u)$. Therefore the gradient is

$$\nabla J = D_\alpha J(\alpha, u) + p^T \cdot (D_\alpha f(\alpha) - D_\alpha A(\alpha)u(\alpha)).$$

Here

- $D_\alpha J \in \mathbb{C}^d$
- $p \in \mathbb{C}^N$
- $D_\alpha f$ the $N \times d$ Jacobian
- $D_\alpha A(\alpha)u(\alpha) = \sum_{i=1}^N (D_\alpha a_{i,j})u_i(\alpha)$, for $j = 1..N$.

After considering the theoretical background in this line of work that has generated so much interest, the problem formulation of this dissertation is discussed in the next chapter.

Chapter 3: Problem Formulation

In this dissertation work numerical optimization techniques are developed for two main problems: the maximum light absorption and the optimal grating coupler problem. These share a common structure as well as the same PDE constraint. The optimal design problems studied have the general form:

$$\begin{aligned} & \max_v && J(u, v) \\ & \text{subject to} && Ku = b, \\ & && g_d(u) \leq 0 \quad d=1, \dots, m \end{aligned}$$

where J is the objective (performance) function, v the design parameters, $Ku = b$ is the state equation obtained from the FEM discretization of the weak form of the 2D Helmholtz PDE, and $g(u)$ the m inequality constraints.

In this dissertation there are two main objective functions to measure the performance of two different nanoplasmonic systems. The first function denoted as J is the maximum light absorption of a silver nanoparticle. It is used to find the maximizer wavelength λ and the optimal size of a silver nanoparticle. The complete development and results are included in chapter 5.

The second objective function, denoted as $P(J)$ is the wave coupler functional used to measure the performance of a grating coupler. This important problem is developed and implemented obtaining very improved designs. The details are discussed in chapter ??.

The PDE describes the behavior of an electromagnetic field in the presence of a metal, thus a nanoplasmonic system. The derivation of the PDE from Maxwell's equations in the 2D frequency domain is discussed in the next section.

3.1 State Equations: PDE

The problem at hand consists in maximizing the value of an objective function J that depends on an electromagnetic field quantities. The electromagnetic field becomes the constraint or state equations that describes the behavior of light in the presence of a nanoscale silver particle. The behavior of light and electromagnetic phenomena is described by Maxwell's equations: (2.1) - (2.4). From these equations a partial differential equation (PDE) in the frequency domain is obtained, assuming an homogeneous medium of propagation with no sources or excitation. The dissertation is in the two dimensional space , $r = zx$ -plane, for an in-plane electromagnetic wave (light) with direction of propagation in the z -axis. This implies that by assuming that light propagates in a time-harmonic fashion, $E(r, t) = E(r)e^{j\omega t}$, the problem reduces to finding $E(r)$, in the space domain, i.e. Maxwell's equations in the frequency domain. Because the dissertation studies the behavior of the electric field in the presence of a silver nanoparticle, the equations of the transverse magnetic (TM) waves in the y axis (H_y) are relevant, for the transverse electric (TE) waves have no electric field component in the direction of propagation. All parameters and assumptions are summarized in table 3.1.

The PDE obtained from Maxwell's equations is a Helmholtz equation. The general 3D formulation is given by:

$$\nabla \times \nabla \times \mathbf{E}(\vec{r}) = \left(\frac{2\pi}{\lambda}\right)^2 \epsilon_r \mathbf{E}(\vec{r}).$$

The simplification in $r = \langle x, 0, z \rangle$ renders a system of partial differential equations:

$$\begin{aligned} \frac{\partial^2 E_z}{\partial z \partial x} - \frac{\partial^2 E_x}{\partial z^2} &= \left(\frac{2\pi}{\lambda}\right)^2 \epsilon_r E_x \\ \frac{\partial^2 E_x}{\partial x \partial z} - \frac{\partial^2 E_z}{\partial x^2} &= \left(\frac{2\pi}{\lambda}\right)^2 \epsilon_r E_z. \end{aligned} \tag{3.1}$$

Table 3.1: Electromagnetism parameters

| parameter | definition |
|---|---|
| \vec{r} | $\langle x, y, z \rangle$ |
| $\mathbf{E}(\vec{r})$ | $\langle E_x(\vec{r}), 0, E_z(\vec{r}) \rangle$ |
| $\mathbf{H}(\vec{r})$ | $\langle 0, H_y(\vec{r}), 0 \rangle$ |
| \vec{n} normal vector at left boundary | $\langle 0, 0, -1 \rangle$ |
| \vec{n} normal vector at right boundary | $\langle 0, 0, 1 \rangle$ |
| k | $2 * \pi / \lambda$ |
| \vec{k} | $\langle 0, 0, k \rangle$ |
| $J = 0$ | electric current density amperes/ m^2 |
| $\rho = 0$ | electric charge density coulombs/ m^3 |
| j | $\sqrt{-1}$ |
| ϵ_r | relative permittivity |
| c_0 | speed of light 299,792,458 m/s |
| ω | <i>angular frequency</i> : kc_0 |
| $\epsilon\mu$ | ϵ_r/c_0^2 |

The TM wave equation is a scalar PDE function in H_y for a plane wave obtained from (3.1) by substitution of the following identity obtained also from (2.1)-(2.4):

$$j\epsilon\omega\mathbf{E}(r) = \nabla \times \mathbf{H}(r). \quad (3.2)$$

Therefore the scalar TM Helmholtz equation in H_y is:

$$-\frac{\partial}{\partial x} \left(\frac{1}{\epsilon_r} \frac{\partial H_y}{\partial x} \right) - \frac{\partial}{\partial z} \left(\frac{1}{\epsilon_r} \frac{\partial H_y}{\partial z} \right) = \left(\frac{2\pi}{\lambda} \right)^2 \mu_r H_y((x, z)), \quad (3.3)$$

where $\mu_r \approx 1$ for air and silver.

The absorbing boundary and perfect electric conductor boundary conditions complete the formulation of the PDE. The absorbing boundary conditions (ABC) discussed in [15], also known as scattering, is a Robin or third type boundary conditions. ABC artificially truncate the air domain surrounding a silver square particle with minimal non-physical reflection.

Absorbing boundary conditions are also easy to implement in a FEM code. The derivations for the xz plane from the general formulation follow.

The 3D formulation at boundaries of an enclosing air square ($1/\epsilon_r = 1$) is:

$$\vec{n} \times (\nabla \times \mathbf{E}) - jk\vec{n} \times (\mathbf{E} \times \vec{n}) = -\vec{n} \times (\mathbf{E}_0 \times j < k\vec{n} - \vec{k} >) e^{-j\vec{k} \cdot \vec{r}}. \quad (3.4)$$

Using the x -axis as the propagation in z , the light source is at $z = 0$, the left vertical boundary of the air square, and the rightmost vertical boundary corresponds to $z = L$, for an $L \times L$ enclosing air square. At $z = 0, \vec{n} = < 0, 0, -1 > \mathbf{E}_0 = < 1, 0, 0 >, \vec{k} \cdot \vec{r} = < 0, 0, k > \cdot < x, 0, z > = kz$. By substitution in (3.4):

$$\begin{aligned} < 0, 0, -1 > \times (\nabla \times \mathbf{E}) - < 0, 0, -jk > \times (\mathbf{E} \times < 0, 0, -1 >) \\ = < 0, 0, 1 > \times (< 1, 0, 0 > \times < 0, 0, -2jk >) e^{-jkz}, \end{aligned}$$

simplifies to $\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} + jkE_x = 2jk$.

At the rightmost boundary $z = L$ also from (3.4), with $\vec{n} = < 0, 0, 1 >, \mathbf{E}_0 = < 0, 0, 0 >$

$$\begin{aligned} < 0, 0, 1 > \times (\nabla \times \mathbf{E}) - < 0, 0, jk > \times (\mathbf{E} \times < 0, 0, 1 >) \\ = < 0, 0, -1 > \times (< 0, 0, 0 > \times j < 0, 0, 0 >) e^{-jkL}, \end{aligned}$$

obtaining $\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - jkE_x = 0$. The boundary conditions in terms of \mathbf{H} are obtained via the substitution (3.2). Because $\nabla \times \mathbf{H} = < -\frac{\partial H_y}{\partial z}, 0, \frac{\partial H_y}{\partial x} >$ then:

$$\begin{aligned} z = 0 : & -\frac{\partial^2 H_y}{\partial z^2} - \frac{\partial^2 H_y}{\partial x^2} + jk \frac{\partial H_y}{\partial z} = 2k\epsilon\omega \\ z = L : & \frac{\partial^2 H_y}{\partial z^2} + \frac{\partial^2 H_y}{\partial x^2} + jk \frac{\partial H_y}{\partial z} = 0. \end{aligned} \quad (3.5)$$

Consider now that at the boundaries the transverse H_y field can be expressed as a plane wave with equation: $H_y = H_y(x, y, z, t) = H_y^0 e^{j(\omega t - kz)}$, therefore

$$\frac{\partial H_y}{\partial z} = (-jk)H_y^0 e^{j(\omega t - kz)} = -jkH_y^0$$

$$\frac{\partial^2 H_y}{\partial z^2} = \frac{\partial}{\partial z} (-jkH_y) = (-jk) \frac{\partial H_y}{\partial z}$$

$$\frac{\partial^2 H_y}{\partial x^2} = 0.$$

The derived ABC boundary conditions for \mathbf{H} , after substitution in (3.5) and multiplication by j/k , are thus

$$z = 0 : -\frac{\partial H_y}{\partial z} + jkH_y = 2jkH_y^0 \quad (3.6)$$

$$z = L : -\frac{\partial H_y}{\partial z} - jkH_y = 0, \quad (3.7)$$

where $H_y^0 := \epsilon\omega$. The derivation of the FEM linear system is discussed next.

3.2 FEM Discretization

The solution of the PDE is obtained numerically by solving a finite dimensional weak form of the PDE in the ansatz space. A k -dimensional approximation to the solution: $u \approx \sum_i u_i w_i$, is computed. The test function w in the weak form of the integral of the Helmholtz PDE is replaced by a finite dimensional set of basis functions w_i , also used for the approximation of u . In this form the nodal 2D finite element method (FEM) or Ritz-FEM is obtained and implemented in this dissertation.

The discretization of the PDE by the Finite Element Method (FEM) and boundary conditions give the linear system $Ku = b$, where u is the numerical approximation vector to the magnetic field at each node, K the stiffness matrix, and b the load vector that models the

light incident to the left boundary of the enclosing air square. The FEM matrix for linear triangular element is derived from the nodes, and three linear interpolation shape functions computed at each node. The m th element matrix is:

$$[K^m] = \iint_T \frac{1}{\epsilon_r^m} B^T B - k^2 N^T N \, dT,$$

T a right triangle. $N^T N, B^T B$ are 3×3 matrices obtained from the 3 linear shape (test) functions in 2D coordinates $N_i(x, y)$, its respective gradients $\nabla N_i(x, y) = \left\langle \frac{\partial N_i}{\partial x}, \frac{\partial N_i}{\partial y} \right\rangle, i = 1, 2, 3$, and ϵ_r^m the relative permittivity assumed to be constant at element m . The shape functions in 2D (x, z) , where z is visualized in the x -axis, is defined at each element $N_i(x, z) = \frac{1}{2A}(a_i + b_i z + c_i x)$, $i = 1, 2, 3$. Here $a_i = z_{i+1}x_{i+2} - x_{i+1}z_{i+2}$, $b_i = x_{i+1} - x_{i+2}$, $c_i = z_{i+2} - z_{i+1}$, and area of the element $A = 0.5(b_i c_{i+1} - b_{i+1} c_i)$, following a natural permutation on $(1, 2, 3)$. For details on the derived formulas the reader is referred to [15],[17].

The stiffness matrix K is obtained from the integrals in the following form: $[K^m] =$

$$\left(\frac{p_r - j p_j}{(p_r^2 + p_j^2)} \right) \iint_T B^T B - k^2 N^T N \, dT = \left(\frac{p_r - j p_j}{(p_r^2 + p_j^2)} \right) [K_1] - k^2 [K_2]. \text{ Here}$$

$$[K_1] = \iint_T B^T B \, dT, \quad B^T B = \begin{bmatrix} \nabla N_1 \cdot \nabla N_1 & \nabla N_1 \cdot \nabla N_2 & \nabla N_1 \cdot \nabla N_3 \\ \nabla N_2 \cdot \nabla N_1 & \nabla N_2 \cdot \nabla N_2 & \nabla N_2 \cdot \nabla N_3 \\ \nabla N_3 \cdot \nabla N_1 & \nabla N_3 \cdot \nabla N_2 & \nabla N_3 \cdot \nabla N_3 \end{bmatrix} \quad (3.8)$$

$$[K_2] = \iint_T N^T N \, dT, \quad N^T N = \begin{bmatrix} N_1 N_1 & N_1 N_2 & N_1 N_3 \\ N_2 N_1 & N_2 N_2 & N_2 N_3 \\ N_3 N_1 & N_3 N_2 & N_3 N_3 \end{bmatrix}. \quad (3.9)$$

Note that the global stiffness matrix $[K]$ is a sparse symmetric matrix as only elements that share nodes have non-zero entries. This added to the fact that element and global matrices are positive definite, are attractive characteristics of the FEM discretization system.

These matrices and the numerical solution are complex valued. The decomposition of the equations into real and imaginary parts is described in the next section.

3.3 Complex Variables Decomposition

In this dissertation complex variables arising in electromagnetism applications are decomposed into real and imaginary components. The state equation obtained from transverse magnetic (TM) Maxwell's Equations in an isotropic media, (3.3), simplify by the variational formulation (weak form) to:

$$(\epsilon_r)^{-1} \iint_{\Omega} \frac{\partial H_y}{\partial x} \frac{\partial w}{\partial x} + \frac{\partial H_y}{\partial z} \frac{\partial w}{\partial z} - k^2 H_y w d\Omega - \int_{\Gamma} \frac{\partial H_y}{\partial n} w d\Gamma = 0.$$

Here w is the test (shape, trial) function, Ω the domain, Γ the boundary, and negative constants have been simplified via integration by parts, also known as Green's theorem. After discretization of the domain into n nodes and m elements, the numerical solution is reduced to a linear system via Finite Element Method.

The FEM approximation of the magnetic field is given by the solution of a complex valued system of equations $Ku = b$. This system is decomposed into real and imaginary components of the form: $K = K^r + j * K^j$, $j = \sqrt{-1}$, $K^r = \text{real}(K)$, $K^j = \text{imaginary}(K)$.

$$\begin{bmatrix} K^r & -jK^c \\ jK^c & K^r \end{bmatrix} \begin{Bmatrix} u^r \\ ju^c \end{Bmatrix} = \begin{Bmatrix} b^r \\ jb^c \end{Bmatrix}. \quad (3.10)$$

Theoretical results for the formulations of gradients in \mathbb{C} are presented in next section as the dissertation evaluates the most efficient approach to compute gradients of functionals.

3.4 Gradients of Functionals

The formulation of design optimization problems in computational electromagnetism is characterized by the extensive use of complex vector analysis. The differential equations of magnetic (H) and electric (E) fields, boundary conditions, wave propagation and permittivity variables, are complex valued expression. Unfortunately in this interdisciplinary community the use of notation and concepts of complex vector analysis in electromagnetism is not mathematically rigorous as can be seen in *Historial Study of Vector Analysis* [18]. Here the author in 1995 writes:

We have examined a history covering a period of over a century. It represents a very interesting period in the development of the mathematical foundations of electromagnetic theory. However in view of the long-entrenched and widespread mis-use of the gradient operator as a component of the divergence and curl operators, the obligation of sharing the insight presented here with many of our colleagues in this field has been a labor fraught with frustration.

Albert Einstein’s comment to the tensor treatment of the mathematician Levi-Civita is revealing : “I admire the elegance of your method of computation; it must be nice to ride through these fields upon the horse of true mathematics while the like of us have to make our way laboriously on foot.” [19]. The controversy between the appropriate use of operators like ∇ is very much alive in 2011 as the reader might verify by searching “Matrix Calculus” in Wikipedia, where the factual accuracy is disputed in three sections, and the discussion in the talk pages shows unsolved disagreement.

Further issues arise during differentiation of functionals of complex valued fields and conjugates that are functions of complex valued variables. The use of operators like the partial derivative of a functional with respect to a complex vector field are common. For example

the following derivation is seen in [20]:

$$\begin{aligned}
J = w(v) &= \iint_{\Omega} Nv \cdot Nv^* d\Omega = \\
\frac{\partial w}{\partial v}(v) &= \iint_{\Omega} N^T Nv^* d\Omega,
\end{aligned} \tag{3.11}$$

where z^* denotes the complex conjugate of a complex valued vector z . It is common in computational electromagnetism papers,[7], [21], [22], [23] to define the derivative $\frac{\partial J}{\partial E}$ of a functional $J(E, E^*) = \iint_{\Omega} E \cdot E^* d\Omega$ with the following the argument:

$$\begin{aligned}
\frac{\partial J}{\partial E} &= \iint \frac{\partial E}{\partial E} E^* + \frac{\partial E^*}{\partial E} E d\Omega \\
&= 2\text{Real} \left(\frac{\partial E}{\partial E} E^* \right) = 2\text{Real}(E^*).
\end{aligned} \tag{3.12}$$

Here the argument is based on the observation $\text{real}(z) = \frac{z+z^*}{2}, \text{imag}(z) = \frac{z-z^*}{2j}$, where $z = x + jy, j = \sqrt{-1}$.

In [24] an extension of the complex derivative when the analytical derivate does not exist is defined. The formal definition of the *real-derivative* given as $\frac{\partial f}{\partial z}|_{z^*=k}, k$ constant, is complex-analytic (holomorphic). Similarly a *conjugate-real derivative* is defined as $\frac{\partial f}{\partial z^*}|_{z=k}$. The rigorous definition of the real and conjugate-real derivative of nonanalytic complex functions that agrees with the complex-derivative when the function is analytic (holomorphic) is:

$$\begin{aligned}
\frac{\partial f}{\partial z} &= \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right), \\
\frac{\partial f}{\partial z^*} &= \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right)
\end{aligned} \tag{3.13}$$

Although this non rigorous treatment have been useful in published works, a complex valued function must satisfy Cauchy-Riemann conditions in order for the derivative to exist and

be continuous. The Cauchy-Riemann conditions for $f(z) = u(x, y) + jv(x, y), z = x + jy$ are:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}.$$

Then because $f(z) = zz^* = (x^2 + y^2)$, with $u(x, y) = x^2 + y^2, v(x, y) = 0$, then f is not holomorphic. (The interested reader is referred to *Wirtinger Calculus*).

Formulations in this dissertation will follow a standard mathematical treatment, departing from the usage depicted in (3.11). Operators like the differentiation of a functional with respect to a complex valued electric field and its complex conjugate will be treated separating the real and imaginary components. This approach allows the identification of the stationary points needed for optimization. In the following section the dissertation analyzes ways to obtain the numerical gradient that will be implemented in the computational chapters.

3.4.1 Numerical Gradients

The discrete application of the gradients of functionals is derived from the generalization of the directional derivative to functionals in Banach Spaces X , or *Gâteaux* derivative. Formally the change in $v \in V \subset X$, in the direction of $\phi \in X$ is defined as: $dJ(v; \phi) = \lim_{t \rightarrow 0} \frac{J(v+t\phi) - J(v)}{t}$, where $J : U \rightarrow \mathbb{C}$, and the limit is taken over the topology of \mathbb{C} . In the context of this dissertation, the functional $J(v)$, is the integral that measures a physical quality or performance of a design, with respect to a change in a parameter v . Therefore the finite dimensional space V is the space of design parameters, J is a real valued integral over v , and its derivative gives the variation δJ needed for optimization.

The current dissertation uses the gradient to determine the direction of increase of the functional J . In optimization with PDE constraints, J depends on the numerical approximation of the PDE, here a complex valued field $u(v)$. Gradients δJ require the computation of analytical or numerical derivatives of J with respect to u , and of u with respect to v . A first approach to find a numerical gradient is the finite difference of the objective function:

$\frac{J(v+\Delta v)-J(v)}{\Delta v}$. This amounts to solving the large scale state equations (PDE) at least twice in order to evaluate J for each parameter and its perturbation. This the most straight forward option, but also more costly by requiring multiple solver runs in order to find the value of the gradient. The adjoint and direct differentiation gradients are more efficient, and therefore developed and implemented in this project.

In this chapter two numerical ways to obtain the gradient are discussed, along with some equivalent formulations encountered when developing the formulas for this work. Also the opportunity to implement this forms is assessed in the context of the projects in this dissertation.

3.4.2 Gradient formulation

Two gradient formulations are studied: the direct differentiation gradient obtained from linearization by Taylor's theorem (DD), and the adjoint method (AM) as in [7],[9]. First note that the variation in a parameter v of a linear system of equations $A(v)x(v) = b(v)$ is given by Taylor's Theorem linear approximation of the derivative, where:

$$\begin{aligned}\frac{A(v+\Delta v)-A(v)}{\Delta v} &= D_v A + O(\Delta v^2) \approx D_v A, \\ \frac{x(v+\Delta v)-x(v)}{\Delta v} &\approx D_v x, \\ \frac{b(v+\Delta v)-b(v)}{\Delta v} &\approx D_v b.\end{aligned}\tag{3.14}$$

Therefore, by product rule differentiation on $Ax = b$, and cancelling Δv :

$$\begin{aligned}D_v Ax + AD_v x &= D_v b \\ AD_v x &= D_v b - D_v Ax.\end{aligned}\tag{3.15}$$

In DD, the variation of the functional $J(x, v)$ is given by chain rule differentiation: $\delta J(x(v)) = (D_x J)(D_v x)$. The gradient $D_v x$ is obtained from (3.15), and the computation of $D_v A, D_v b$

can either be numerical or analytical, if the expressions are differentiable in the parameter v . The numerical $D_v A$ can be obtained by finite difference subtracting the matrix or vector from its perturbed counterpart, as in (3.14). Moreover if the formulas for $A(v), b(v)$ are known, D_v can be obtained by exploiting the linear vector space structure of the system: $A(v + \Delta v) - A(v) = A(\Delta v)$, and $b(v + \Delta v) - b(v) = b(\Delta v)$. The stiffness matrix A is used here two times: to obtain the PDE solution and its gradient.

The AM starts with a different formulation of the objective function $\hat{J}(x, v)$ obtained by introducing the adjoint variables p and the PDE constraint into the objective function expression:

$$\hat{J} = J + p^T(b - Ax).$$

This expression is equivalent to J when x is feasible, i.e. $b - Ax = 0$. The adjoint variables p^T are a generalization of Lagrange multipliers in NLP. The total differential of \hat{J} is:

$$\begin{aligned} \delta \hat{J} &= D_v J + D_x J D_v x + p^T(D_v b - [D_v A]x - A D_v x) \\ &= D_v J + (D_x J - p^T A) D_v x + p^T(D_v b - [D_v A]x). \end{aligned}$$

Then if p satisfies the adjoint equation: $p^T A = D_x J$, the adjoint gradient simplifies to:

$$\delta \hat{J} = D_v J + p^T(D_v b - [D_v A]x). \quad (3.16)$$

Note that the same matrix A is efficiently used twice: to solve the state and the adjoint equations $p^T A = D_x J$.

The main advantage of the AM over DD occurs when the number of design variables, v is a large number. In this case the computation of $D_v x$, x an n -dimensional field, is very costly. Then the investment in solving the adjoint equation is well justified. Nonetheless for the objective function of the electric field obtained from the gradient of the magnetic field, AM is not the best option. The FEM discretization $Ku = b$, gives the magnetic field u , while the objective function depends on the electric field. Therefore the matrix $K \neq A$ in the

adjoint equation (3.16).

3.4.3 Theoretical Results

This dissertation hopes to contribute to the interdisciplinary community many formulations to find gradients of expressions, a contemporary discussion considering the unresolved issues in the Matrix Calculus topics mentioned in the introduction. An equivalent formulation simplifies computations, saves time and effort, and renders formulas that can be verified. First in this section an equivalent formulation for the gradient of the linear system matrix A^{-1} with respect to a design parameter v is proposed.

Proposition 1. $[D_v A^{-1}] = -A^{-1}[D_v A]A^{-1}$.

Proof. By (3.15):

$$\begin{aligned} D_v x &= A^{-1}(-[D_v A]x + D_v b) = -A^{-1}[D_v A]x + A^{-1}D_v b, \\ &= -A^{-1}[D_v A]A^{-1}b + A^{-1}D_v b \end{aligned} \quad (3.17)$$

Also since $x = A^{-1}b$ then

$$D_v x = [D_v A^{-1}]b + A^{-1}D_v b. \quad (3.18)$$

The proposition follows from equality of (3.17) and (3.18). \square

In order to verify the proposition $[D_v A^{-1}]$ was computed using a finite difference of inverse matrices and compared to the obtained formula. For $\Delta x = 0.5, \rightarrow \Delta x = 0.1$, the maximum difference between elements of both matrices also vanished.

Whenever possible the analytical derivative is preferred over the numerical for parameters that are differentiable as is the case with the complex valued permittivity ϵ . From FEM

weak form of the Helmholtz PDE the integral operator expression is given by:

$$A = \frac{1}{\epsilon} \iint_{\Omega} B^T B - k^2 N^T N \, d\Omega.$$

Because N, B do not depend on ϵ but on the geometry of the triangular elements, the analytical derivative is given by

$$D_{\epsilon} A = -\frac{1}{\epsilon^2} \iint_{\Omega} B^T B - k^2 N^T N \, d\Omega.$$

A finite difference approximation of A must be computed by separating the real and imaginary parts of ϵ . Otherwise a difference in terms of $\delta = (\epsilon + \delta) - \epsilon$ is misleading, because a direction of increase, that is an order in (r, i) , is not defined in the complex plane. Separating the real (or imaginary) difference only, e.g. $\delta = (\text{real}(\epsilon) + \delta) - \text{real}(\epsilon)$, and error for the finite difference of A is given in the proposition.

Proposition 2. The error of the finite difference δ approximation of real or complex part of the parameter $\epsilon = (\epsilon^r, \epsilon^i) \neq \vec{0}$ is $\left| \frac{\delta}{\epsilon^2(\epsilon + \delta)} \right| < \tau$, where $\delta = (\delta^r, \delta^i)$ is the difference in $\text{real}(\epsilon)$ and $\text{imaginary}(\epsilon)$ respectively, and τ a desired tolerance.

Proof. This is obtained by observing that the finite difference simplifies to: $-1/(\epsilon^2 + \epsilon\delta)$, while the analytical derivative is $-1/\epsilon^2$. \square

The following are computational chapters where the dissertation develops and presents results obtained from the methods and theory following the discussed approach in this chapter. Nonetheless there are new formulations derived and presented in chapters 5 and 6 to find gradients with respect to a real valued λ , complex valued ϵ and geometric parameters v . These formulations will add to the theoretical results in this dissertation.

Chapter 4: FEM solver

PDE's obtained from Maxwell's Equations model the magnetic and electric field or state equations that nanophotonic/nanoplasmonic systems satisfy, i.e. the constraints of the optimization problem. Furthermore in the optimal design context the objective function J and its gradient are functions of the solution of these state equations. Therefore the initial step in the development of numerical optimization tools for the solution of PDE constrained problems is the study of the PDE solvers.

In this dissertation the numerical solution to the state equations modeled by the 2D Helmholtz equation in the scalar magnetic field is obtained from the FEM linear system. An efficient FEM solver provides also important data to be used in the gradient computations for the PDE constrained optimization problem. It is also the most computationally expensive part of the optimization with PDE constraints because of the large number of variables of the order of the mesh.

In this chapter the MATLAB implementation of a 2D FEM complex variable solver is presented. This solver is a main tool in the numerical optimization of the maximum light absorption problem in chapter 5. Two main solvers have been developed, and classified according to the way the FEM matrix is assembled: by node sets or global assembly. Both FEM solvers use extensively the connectivity of the uniform grid and therefore there is no need to read a file with the mesh connectivity and node coordinates. In this applications only the size of the grid and of the silver particle are needed as input.

The description of the computational domain, the uniform grid, and the computational results are included starting with the development of the computational domain with the corresponding boundary conditions.

4.1 Computational Domain

The FEM solver obtains the numerical approximation of the scalar magnetic that satisfies the 2D Helmholtz equation in (3.3) in the (x, z) domain, with z in the horizontal axis. The magnetic wave is incident to the leftmost boundary of a 200 nm air square. It is enhanced by the interaction with a 30 nm silver square. The complete formulation requires the application of boundary conditions.

The boundary conditions include absorbing boundary conditions (ABC) and perfect electric conductors (PEC). The formulation of the ABC boundary in (3.7) is discussed in detail in chapter 3, and reduces to:

$$\begin{aligned} z = 0 : -\frac{\partial H_y}{\partial z} + jkH_y &= 2j\epsilon\omega \\ z = 200 \text{ nm} : -\frac{\partial H_y}{\partial z} - jkH_y &= 0. \end{aligned}$$

These are easily incorporated in the FEM formulation and include the incident magnetic field H_0 in the left boundary.

For top and bottom boundary conditions are obtained by assuming the medium outside the computational domain is a perfect electric conductor: $\nabla \times E = \vec{0}$, i.e. the curl of the electric field vanishes at the boundary. This models a perfect electric conducting boundary. Then at the top and bottom air interface, $x = 200, x = 0$, the boundary conditions simplify to the homogeneous natural boundary condition. This follows because the normal component of the electric field at the top and bottom interface with the perfect conducting medium: $\vec{n} \times E = \langle 0, \pm E_z, 0 \rangle = \vec{0}$. Then by equation (3.2) $\Rightarrow \frac{\partial H_y}{\partial x} = 0$. Figure 4.1 summarizes the boundary conditions in the geometry on the 2D PDE computational domain.

The grid used to model the geometry of this simple computational domain is discussed next.

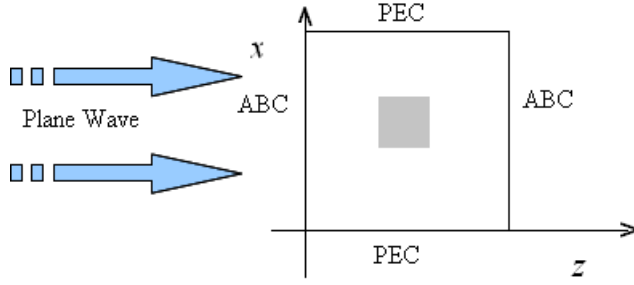


Figure 4.1: 2D PDE domain

4.2 The Uniform Grid

A most practical decision when studying a large-scale optimization problem is to use the simplest mesh that can compute an acceptable numerical approximation to the state equation, reducing the computational cost. Remeshing remains an option if needed in further steps. A uniform grid provides a simple geometry, known connectivity for bookkeeping, and a finite difference domain with no need of interpolation. In this simple geometry of squares the grid is also an accurate match. Therefore in this project a uniform grid of size h is used and will remain constant to study the gradient of the objective function with respect to a design variable. This guarantees that there is no mesh related variation introduced.

In order to determine the best value h when $\lambda = 350$ nm, an experiment is conducted modifying the size of the grid to observe the change in the light absorption objective function J at the interior nodes of the silver particle. Table 4.1 summarizes the sensitivity of the objective function when refining the grid. It shows results when increasing the number of nodes in the $L = 150$ nm air square, $\lambda = 350$ nm and $s = 30$ nm silver square centered at point $(x, z) = (75, 75)$ in nanometers.

Note that the increase in the value of the absorption of light J in this experiment is caused in part by the fact that only the interior nodes of the silver square are used in this computation of J . This measure is adopted here in order to avoid the discontinuities in the air/silver interface and the sharp corners of the silver square. Therefore the area of the

Table 4.1: Grid size and J

| h | number of silver nodes | total nodes | J |
|--------|------------------------|-------------|--------------|
| 7.5 nm | 9 | 441 | 6.316272e+05 |
| 6 nm | 16 | 676 | 1.124639e+06 |
| 5 nm | 25 | 961 | 1.755978e+06 |
| 3 nm | 81 | 2601 | 5.582489e+06 |

interior of the silver square increases when the grid is refined, and the fix grid approach is adopted for gradient studies. The $h = 5$ nm with $31 \times 31 = 961$ nodes uniform grid will be used for the $J(\epsilon)$ experiments. A coarser mesh with $h = 7.5$ nm and $21 \times 21 = 441$ is used for AMPL based experiments. The triangular uniform grid and refined corner grid are included in figures: 4.2 and 4.3 respectively.

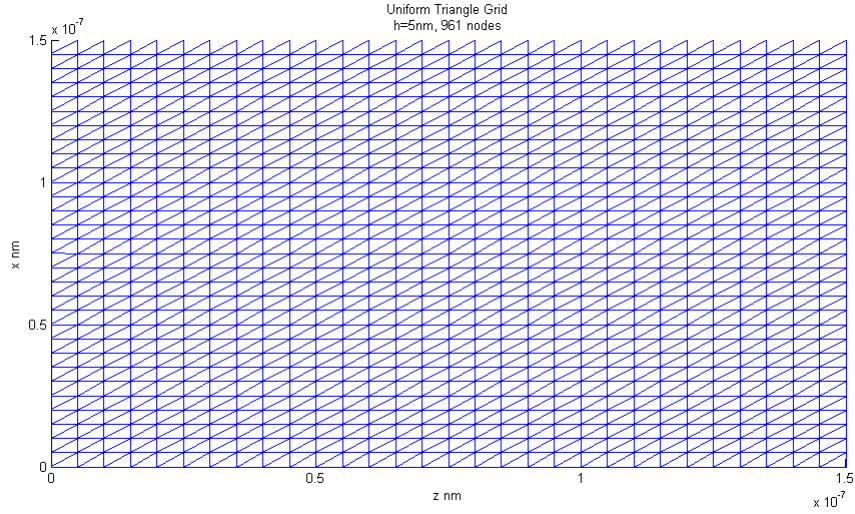


Figure 4.2: Uniform Triangle Grid

4.3 Computational Implementation

The FEM has been implemented in MATLAB using different approaches. The algorithms are classified in two categories: set building or assembling algorithms. Both work on a uniform triangular grid or a refined grid over the corners of the silver square. The use of

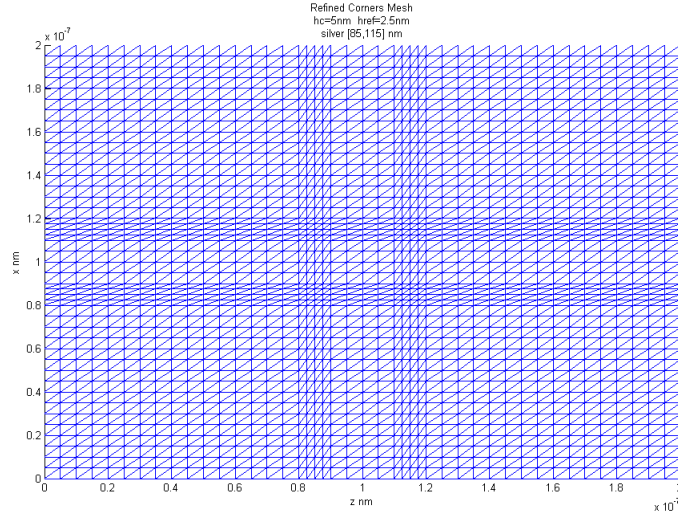


Figure 4.3: Refined Corners

sets is the strategy used to built AMPL models, but is much slower because the stiffness matrix is defined by searching nodes in the lists: boundary, corner, silver or air node sets. The projects accomplish several tasks:

1. Assembling Algorithm: Given the size of a uniform triangle mesh and silver square, assemble the element matrices into the global matrix.
2. Set building Algorithm: Generate sets of nodes according to material properties and geometry, and define the global matrices.
3. Generate the FEM linear system decomposed into real and imaginary parts.
4. Incorporate scattering boundary conditions from element connectivity or set information.
5. Solve the discretized system by iterative solver.
6. Compute the Electric field by finite difference with half-step treatment along the air/silver interface.

Table 4.2: Time statistics

| nodes | Assembly Alg. | Assembly Alg + File | Set Alg. |
|-------|---------------|---------------------|----------|
| 441 | 0.28125s | 0.43745s | 2.45313s |
| 961 | 0.22657s | 0.69531s | 8.92187s |
| 2601 | 1.15625s | 1.64063s | 54s |

Some speedup is obtained by using the connectivity of a uniform triangular grid, and then the connectivity matrix can be generated in the algorithm vs. reading from a mesh file. The 3×3 element matrices K^e are the same for all air and silver elements. Therefore by identifying the node as air or silver, the global matrix K can be built by equations. Note that in a triangular grid all interior nodes belong to 6 elements and the corresponding edges: 2 horizontal, 2 vertical and 2 diagonal edges. The global matrix K is obtained by adding contributions of all elements. Therefore, equations are obtained from the element matrices K^e for interior nodes, corner and boundary nodes. The corresponding equations obtained from the element matrices K^e for interior nodes of air or silver follow, n the number horizontal nodes.

1. Nodes: $K(i, i) = 2K^e(1, 1) + 2K^e(2, 2) + 2K^e(3, 3)$.
2. Horizontal edges: $K(i, i + 1) = K(i, i - 1) = 2 * K^e(1, 2) = K(i + 1, i) = K(i - 1, i)$,
by symmetry.
3. Vertical edges: $K(i, i + n) = K(i, i - n) = 2 * K^e(2, 3) = K(i + n, i) = K(i - n, i)$.
4. Diagonal edges: $K(i, i + n + 1) = K(i, i - n - 1) = 2K^e(1, 3) = K(i + n + 1, i) = K(i - n - 1, i)$.

Time statistics of a typical run are included in table 4.2.

The finite difference method (FDM) has been implemented in order to find the electric field from the magnetic field: $E_x = -\frac{1}{j\omega\epsilon} \frac{\partial H_y}{\partial z}$, and $E_z = \frac{1}{j\omega\epsilon} \frac{\partial H_y}{\partial x}$. For all interior air or silver nodes the central finite difference in space is used as it is second order accurate. The solution is obtained in an n points list u , where nodes are numbered for constant x , by

increasing z coordinates: i.e. (x, z) coordinates according are numbered $1 = (0, 0), 2 = (0, h), 3 = (0, 2h) \dots n = ((tnodes - 1) * h, (tnodes - 1) * h)$, where $tnodes$ is the number of nodes per axis. At the i th node $r_i = (x_i, 0, z_i)$, the central difference (CFD) in the $u[1 : npoints]$ node list is obtained as:

$$\frac{\partial H_y(r_i)}{\partial z} \approx \frac{u[i + 1] - u[i - 1]}{2h}, \quad \frac{\partial H_y(r_i)}{\partial x} \approx \frac{u[i + tnodes] - u[i - tnodes]}{2h}.$$

At an interface nodes a half space is substituted by averaging the value of both edge-nodes to obtain the midpoint value forward:

$$fmid_x(u_i) = 0.5(u[i + tnodes] + u[i]), fmid_z(u_i) = 0.5(u[i + 1] + u[i]),$$

or backward:

$$bmid_x(u_i) = 0.5(u[i - tnodes] + u[i]), bmid_z(u_i) = 0.5(u[i - 1] + u[i]).$$

When $u[i + 1]$ is in the silver/air interface the CFD is modified using the forward midpoint:

$$\frac{\partial H_y(x_i, z_i)}{\partial z} \approx \frac{fmid_z(u_i) - u[i - 1]}{1.5h}, \quad \frac{\partial H_y(x_i, z_i)}{\partial x} \approx \frac{fmid_x(u_i) - u[i - tnodes]}{1.5h}. \quad (4.1)$$

This happens at the left and bottom boundary nodes of the silver square. At the top and right silver square boundary nodes, the backward $bmid_x, bmid_z$ substitute $fmid_x, fmid_z$ in (4.1).

The treatment of the FDM at boundary nodes was explored graphically and compared with COMSOL's solution. The best match is to use backward difference at the right and top silver square boundary nodes. This amounts to defining all silver/air interface nodes as silver.

4.4 FEM Computational Results

The complex valued FEM has been implemented in MATLAB for various computational grids to obtain the electromagnetic field H_y , and then compute the vector valued electric field E_x, E_y from H_y by finite differences. The results are presented here as visualizations of the real part of the electromagnetic fields obtained with MATLAB, and then compared to visualizations obtained by the commercial package COMSOL-Multiphysics 3.5. All are computed for $\lambda = 300$ nm, for a 30 nm silver square centered in a 200 nm air square.

COMSOL uses a continuity condition around the interface of silver and air. This condition is equivalent to introducing a double boundary at the interface: one in silver, and other in air. The electric field obtained in this project from H_y , obtained from FEM discretization, is computed via finite difference in space. Therefore the project uses a half-step around the interface such that nodes in silver or air would not use interface nodes that are contained in both materials.

In figure 4.5 obtained from COMSOL, a refined unstructured grid is used to compute the electric field in the direction of propagation, in this case the x -axis. The comparable graphs in figures 4.6-4.8 are obtained from the finite difference of the H_y solution obtained via MATLAB implementation of the solver, and visualization. The project's implementation of FEM used structured grid, refined corners and finally averaged values around the silver particle interface.

For the E_x , the averaged solution is the best match to COMSOL's refined solution. This is accomplished by taking the mean of the FEM solution u over all adjacent nodes of the interface nodes. The finite difference is then used to find the numerical approximation to $\frac{\partial H_y}{\partial z}$, using a half step at the discontinuous interface between air and silver.

Regarding the results for the E_z component, a reduction toward zero is observed in the refined grid, vs the coarse grid in figures 4.9 vs. 4.10. It is noted that the better solution makes the non-zero brighter spots around the corner smaller in the visualizations. This is the sharp corners that are problematic for numerical models.

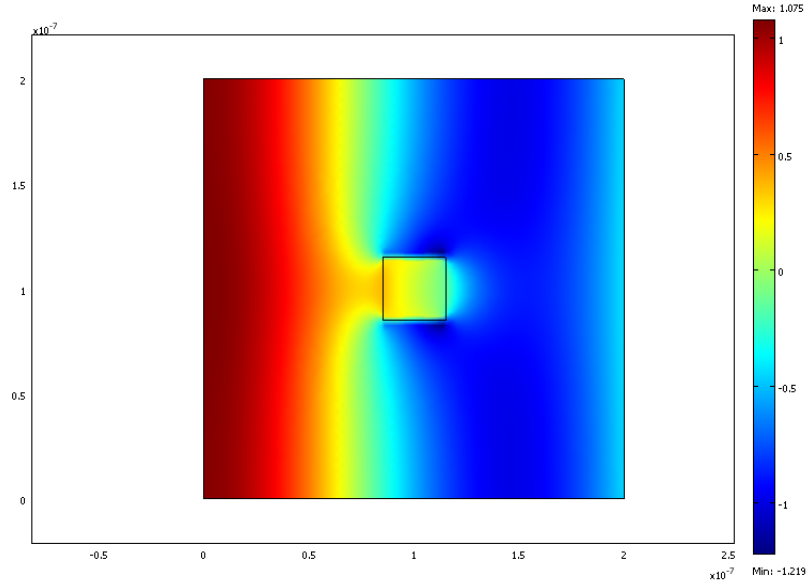


Figure 4.4: Comsol's Ex: Refined Unstructured Mesh

Visualizations show the expected near field enhancements when the electromagnetic field interacts with a silver nanoparticle produced by the localized plasmon (LPP). A very significant difference is seen when the mesh is refined. The most notable area for improvement is around the sharp corners of the square. This is well known to be discontinuous because there is no unique normal direction, and a source of error in numerical electromagnetism related to the lost of tangential continuity. An option for sharp corner treatment is to introduce edge elements on triangles containing the corners into the FEM discretization, as suggested in [25]. The project did not pursue further solver improvement in order to continue with the gradient based computations.

The next chapter explores numerical optimization techniques that include and develops further this FEM solver tool.

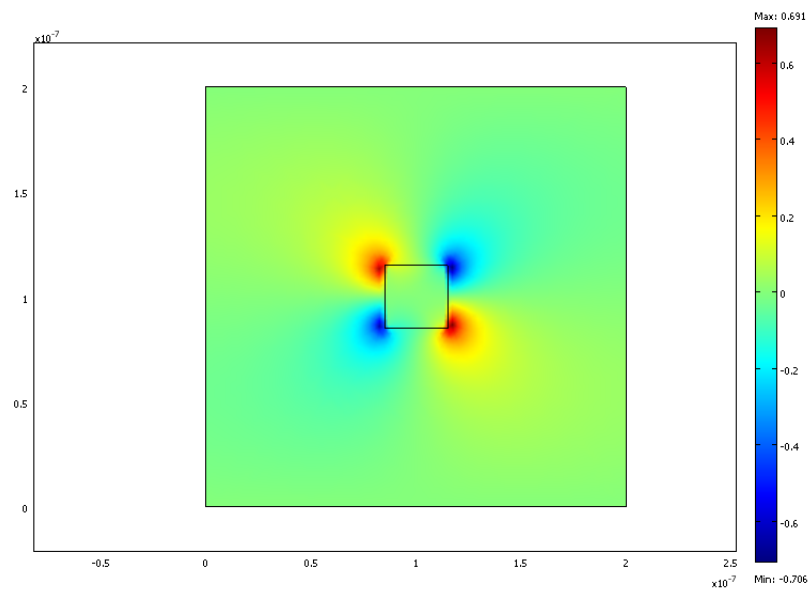


Figure 4.5: Comsol's Ez: Refined Unstructured Mesh

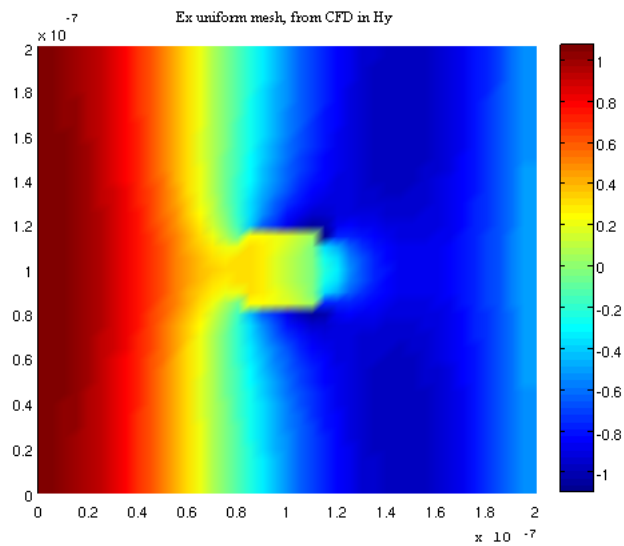


Figure 4.6: Ex in Coarse Grid

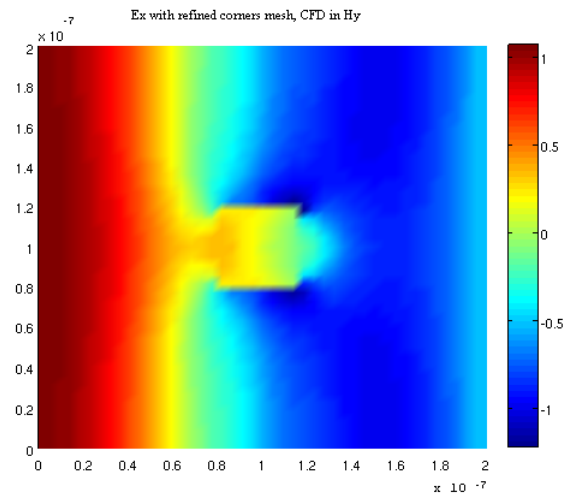


Figure 4.7: Ex in Refined Grid

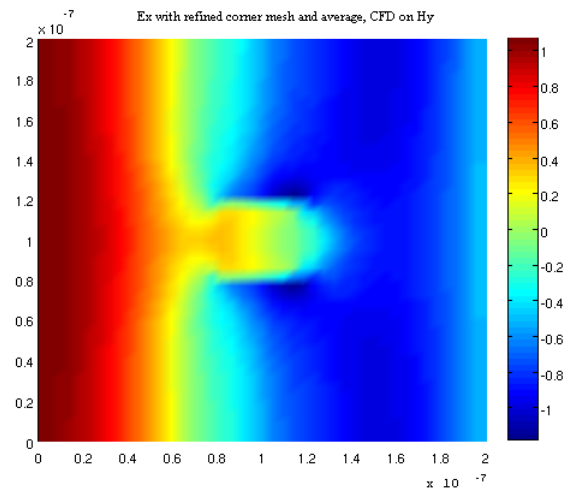


Figure 4.8: Ex with average

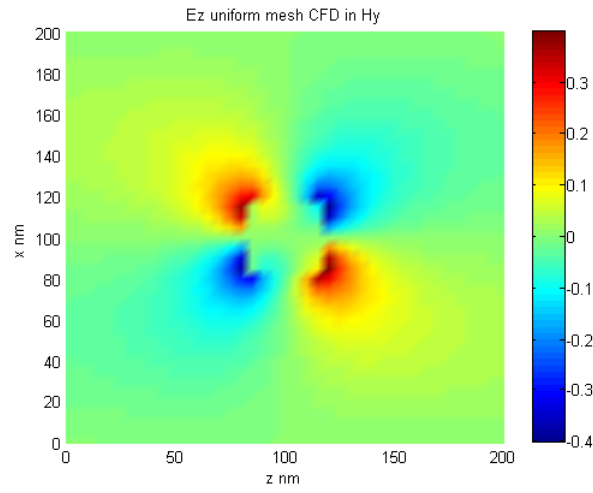


Figure 4.9: Ez in Coarse Grid

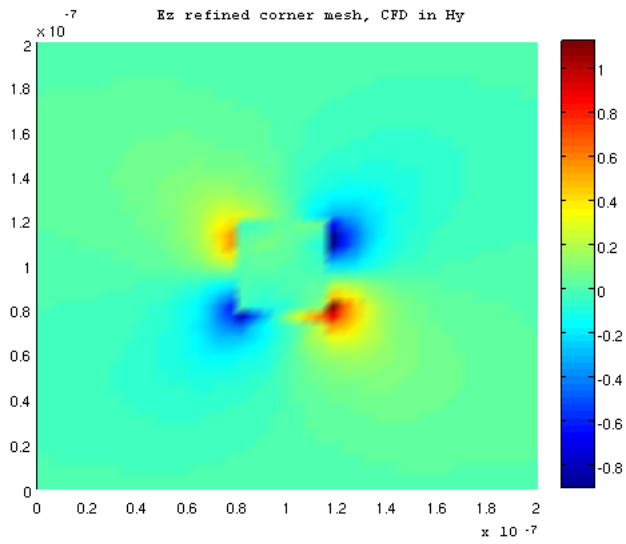


Figure 4.10: Ez in Refined Grid

Chapter 5: Maximum Light Absorption

5.1 Introduction

The light absorption of silver particles of sub wavelength size is an amazing property in contrast with the reflection of light occurring in bulk silver. The investigation of light absorption enhancement by nanoparticles is an important application in our times, with promises of future development of new efficient sensors and powerful greener devices [26]. Therefore the first optimal design study in this dissertation is to find optimal parameters to maximize the light absorption J of a silver nanoparticle.

The maximum light absorption is studied using two different optimization approaches. The first approach is to discretize then optimize. This numerical optimization approach is implemented in using the AMPL mathematical modeling language. The second approach is the parameter optimization using the gradient of J with respect to its parameters λ and ϵ . For this approach the formulations are derived and then implemented in MATLAB.

There are two contexts for the light absorption functional that are defined by the parameter of interest: $J(\lambda)$ and $J(\epsilon)$. In the $J(\lambda)$ context, the maximizer λ is searched for a 30 nm silver square in air. In the $J(\epsilon)$, the optimal size of a silver nanoparticle is the variable of study. Each problem require the development of a particular gradient formulation and strategy. Both use the FEM solver developed in chapter 4 plus new developed tools to obtain the objective function and its gradients.

The formulation of the objective function is introduced first in this chapter.

5.2 Optimization Criteria

A simplified equation in [26] of the absorption of silver is obtained from the divergence of the Poynting Vector. The objective function formula is given by:

$$J = -0.5\omega\epsilon_0\text{imag}(\epsilon_r) \iint_S E\bar{E}dS, \quad (5.1)$$

where $E\bar{E}$ is the norm (or modulus) squared of the complex valued electric field, S is a silver square of sub wavelength scale, ϵ_0 permittivity of vacuum, $\text{imag}(\epsilon_r)$ the imaginary part of the permittivity of silver for a given wavelength, and $\omega = \frac{2\pi c_0}{\lambda}$ the angular frequency. Because this objective function is in the electric field and the Helmholtz equation is in the magnetic field a formulation of $J(H)$ is needed and derived in the first section. The objective function J is subject to the field equations derived from Maxwell's Equations in the 2D (x, z) plane. J is modeled by the equation in the time harmonic electric phasor $Ee^{j\omega\epsilon t}$ and its complex conjugate \bar{E} :

$$J(\lambda, \epsilon) = -0.5\epsilon_0\omega(\text{imag}(\epsilon_r)) \iint_{SS} E \cdot \bar{E} dz dx. \quad (5.2)$$

By Maxwell's equations the electric field under the appropriate assumptions can be obtained from the electromagnetic field H through the equation $E = \frac{1}{j\omega\epsilon}\nabla \times H$, where $\epsilon = \epsilon_0\epsilon_r$, ϵ_0 the permittivity of air. Denote $per = \epsilon_r$ the relative permittivity of silver obtained numerically from the database $getper()$, $p_i = \text{imag}(per)$, $p_r = \text{real}(per)$, and $\epsilon_r = p_r + jp_i$. The gradient of the objective function is denoted $\delta J = D_\lambda J$, then $J(\omega(\lambda), per(\lambda), E(\lambda))$. The electric field E is obtained from the numerical magnetic field H that is the numerical solution to the differential equation computed via Finite Elements.

For a transverse magnetic field $H_y = H$ the electric field simplifies to

$$E(\lambda) = \langle E_x, 0, E_z \rangle = \frac{1}{\omega(\lambda)(p_r(\lambda)j - p_i(\lambda))\epsilon_0} \langle -\partial H/\partial z, 0, \partial H/\partial x \rangle.$$

Therefore the numerical approximation of the electric field is computed using the central finite difference gradient of the magnetic field in the silver square.

$$E_x = \frac{-\partial H}{\partial z} \cdot \frac{1}{j\omega\epsilon} = \frac{-\partial U}{\partial z} \cdot \frac{1}{j\omega\epsilon} \approx - \left(\frac{U_z^{r,n+1} - U_z^{r,n-1}}{2h} + j \frac{U_z^{i,n+1} - U_z^{i,n-1}}{2h} \right) \frac{1}{j\omega\epsilon}$$

$$E_z = \frac{\partial H}{\partial x} \cdot \frac{1}{j\omega\epsilon} = \frac{\partial U}{\partial x} \cdot \frac{1}{j\omega\epsilon} \approx \left(\frac{U_x^{r,n+1} - U_x^{r,n-1}}{2h} + j \frac{U_x^{i,n+1} - U_x^{i,n-1}}{2h} \right) \frac{1}{j\omega\epsilon}.$$

Here the real and imaginary parts are separated and the following notation is used: $2h = z_{n+1} - z_{n-1}$, $U_z^{r,n+1} = \text{real}(H(z_{n+1}))$, $U_z^{i,n-1} = \text{imag}(H(z_{n-1}))$, $U_x^{r,n+1} = \text{real}(H(x_{n+1}))$, $U_x^{i,n+1} = \text{imag}(H(x_{n+1}))$. The numerical integration over the silver square is performed by the 2D trapezoidal rule. The 2D trapezoidal rule is defined over the $h \times w$ rectangle by denoting the corner vertices set $\{A, B, C, D\}$, sb = boundary nodes set (sides of the rectangle), sil =interior nodes set, and $z + \Delta h \in H, x + \Delta w \in W$:

$$\begin{aligned} \int \int_{SS} E \cdot \bar{E} \, dz \, dx &= \int \int_{SS} f(x, z) \, dz \, dx \approx \\ 1/4hw \cdot (f(A) + f(B) + f(C) + f(D) + 2f(sb) + 4f(sil)). \end{aligned} \quad (5.3)$$

Note $f(sb) = \sum_{m \in H} \sum_{n \in W} f(x_n, z_m)$ for $(x_n, z_m) \in sb$, and similarly

$f(sil) = \sum_{m \in H} \sum_{n \in W} f(x_n, z_m)$ for $(x_n, z_m) \in sil$.

The following comprises the derivation of the objective function by decomposing into real

and imaginary. Denote

$$\delta u_z^r = \frac{U_z^{r,n+1} - U_z^{r,n-1}}{2h} \approx \text{real}\left(\frac{\partial U}{\partial z}\right)$$

$$\delta u_z^i = \frac{U_z^{i,n+1} - U_z^{i,n-1}}{2h} \approx \text{imag}\left(\frac{\partial U}{\partial z}\right)$$

and the finite difference notation corresponding to $\frac{\partial U}{\partial x} : \delta u_x^r, \delta u_x^i$. Then the following is obtained:

$$E_x \approx -\delta u_z^r \left(\frac{-jp_r - p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} \right) - \delta u_z^i \cdot \left(\frac{p_r - jp_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} \right)$$

Therefore the real and imaginary parts of E_x are:

$$\text{real}(E_x) = \frac{\delta u_z^r p_i - \delta u_z^i p_r}{\omega \epsilon_0 (p_r^2 + p_i^2)} = a \quad (5.4)$$

$$\text{imag}(E_x) = \frac{\delta u_z^r p_r + \delta u_z^i p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} = b. \quad (5.5)$$

Similarly:

$$E_z \approx \delta u_x^r \left(\frac{-jp_r - p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} \right) + \delta u_x^i \left(\frac{p_r - jp_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} \right)$$

and

$$\text{real}(E_z) = \frac{\delta u_x^i p_r - \delta u_x^r p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} = c \quad (5.6)$$

$$\text{imag}(E_z) = \frac{-\delta u_x^r p_r - \delta u_x^i p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} = d. \quad (5.7)$$

Finally the integrand is: $f(x, z) = E \cdot \bar{E} = \text{real}(E_x)^2 + \text{imag}(E_x)^2 + \text{real}(E_z)^2 + \text{imag}(E_z)^2 = a^2 + b^2 + c^2 + d^2$.

Denote

$$Q = \frac{1}{\epsilon_0} (\omega(p_r^2 + p_i^2))^{-1} \text{ and}$$

$$a^2 = (\delta u_z^r p_i - \delta u_z^i p_r)^2 Q^2,$$

$$b^2 = (\delta u_z^r p_r + \delta u_z^i p_i)^2 Q^2,$$

$$c^2 = (\delta u_x^i p_r - \delta u_x^r p_i)^2 Q^2,$$

$$d^2 = (- (\delta u_x^r p_r + \delta u_x^i p_i))^2 Q^2.$$

The derived formula is:

$$E \cdot \bar{E} = a^2 + b^2 + c^2 + d^2 = Q^2(p_r^2 + p_i^2) [(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2] \quad (5.8)$$

The objective function is thus

$$\begin{aligned} J(\lambda, \epsilon) &= -0.5\epsilon_0\omega p_i \iint_{\Omega} \frac{1}{\omega^2\epsilon_0^2(p_r^2 + p_i^2)} [(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2] d\Omega \\ &= -\frac{0.5}{\epsilon_0} \iint_{\Omega} \frac{p_i}{\omega(p_r^2 + p_i^2)} [(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2] d\Omega. \end{aligned} \quad (5.9)$$

The following project is the first approach to the optimization with PDE constraints. The AMPL mathematical modeling language allows to discretize the PDE and define as variables in the constraint and objective function definition of a large scale comprehensive optimization problem. The details of the AMPL models and results are presented next.

5.3 AMPL Model

AMPL is an algebraic modeling language for linear and nonlinear optimization problems. It has access to many optimization solvers that have been developed considering very diverse optimization algorithms including among others:

1. SNOPT: constrained optimization for linear/nonlinear functions subject to based on the sequential quadratic programming (SQL) algorithm sparse solver
2. LOQO: smooth constrained optimization solver based on infeasible primal-dual interior-point method applied to a sequence of quadratic approximations to the problem.

Through an AMPL mathematical modeling the user defines the optimization problem using a model that AMPL set up to provide gradients for a chosen optimization solver. A one-dimensional model for the light absorption problem was successfully solved by the 2000 version of SNOPT, but was not solved by LOQO or any other optimization solver. The project now developed a two dimensional PDE constrained optimization problem using the assembled linear system obtained from the FEM discretization as the linear constraint.

The AMPL optimization model built accomplishes the following tasks:

1. Define a uniform mesh of right triangles of size h in a 150 nm square of air with a centered 30 nm silver square centered.
2. Build sets of nodes with similar connectivity and material properties: corner, boundary and interior nodes in air or silver.
3. For the nodes of each set, define the symmetric real and imaginary stiffness matrices A_r, A_i according to the material properties, absorbing boundary conditions (Robin or third type) on boundary nodes, using the known connectivity of the uniform mesh.
4. Uses cubic spline data to find the permittivity of the variable wavelength λ .
5. Define the system of equations as a set of 18 constraints on the real and imaginary parts: $[A_r]U_r - [A_c]U_i = 0, [A_c]U_r + [A_r]U_i = b_i$.
6. Define the light absorption objective function on the center or interior nodes of the silver square.

The uniform mesh provides the most simple structure to start the large-scale optimization. It is a key component in the AMPL modeling of the FEM linear system from sets of nodes. The global matrix is defined by using formulas obtained from the connectivity of the uniform mesh.

Because the permittivity of silver depends on the wavelength, a cubic spline interpolation of experimental data is used to serve as the permittivity function of λ as explained next.

5.3.1 Cubic Spline Permittivity

In this project experimental data of the relative permittivity of silver at various wavelengths is available in a supra database. Permittivity of silver $\epsilon_r = p_r + jp_i$ and its derivative are part of the definition of the objective function and its gradient. Therefore a cubic spline interpolation has been obtained to determine permittivity at various wavelengths in the AMPL model. For MATLAB, permittivity is obtained from the database, and the gradient is obtained via finite difference approximation: $D_\lambda \epsilon \approx \frac{\epsilon(\lambda+\Delta\lambda) - \epsilon(\lambda)}{\Delta\lambda}$.

The cubic spline interpolation in figure 5.1 shows a smooth function of the permittivity data with different step numbers in the 200 to 400 nm wavelength interval: $\Delta\lambda = 1, 2, 5, 10, \text{ and } 20$ nm. Cubic splines are continuous at connecting nodes for function values, first and second derivatives. Therefore the system of equations for each data value x_m : $s_m(x_m) = y_m, s_{m+1}(x_m) = y_m, m = 1, \dots, n - 1$, plus first and second derivatives $s'_m(x_m) = s'_{m+1}(x_m), s''_m(x_m) = s''_{m+1}(x_{m+1})$. The first and second derivatives of the cubic spline are computed and visualized. The results are included in figures 5.2 - 5.3.

In AMPL mathematical modeling, the gradient-based NLP optimization solvers require

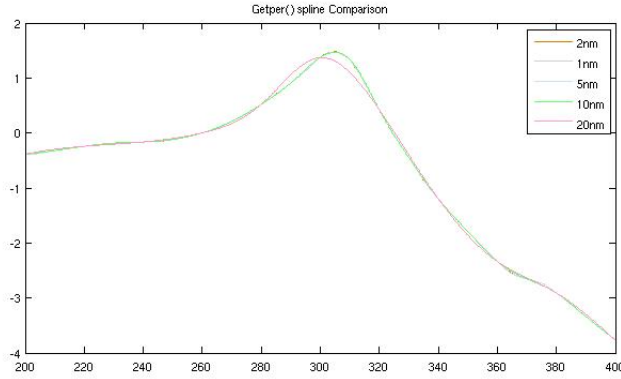


Figure 5.1: Cubic Spline for $\Delta\lambda=1,2,5,10,20\text{nm}$

the derivative of the permittivity function to be smooth. From the graphs, the 10 and 20 nm step show derivatives with less abrupt changes. The 10 nm cubic spline interpolation function is preferred over 20 nm, for it represents more accurate values from the experimental

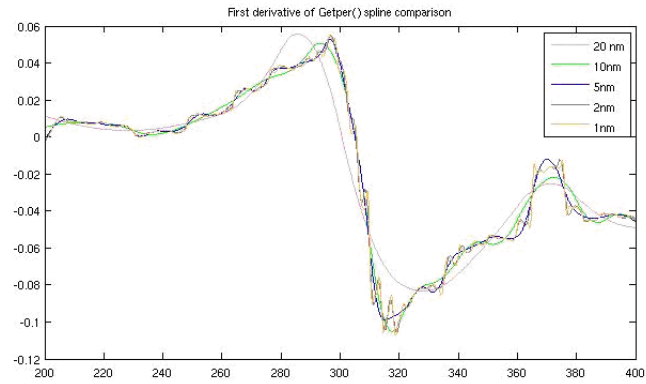


Figure 5.2: First Derivative of Cubic Spline: $\Delta\lambda=1,2,5,10,20\text{nm}$

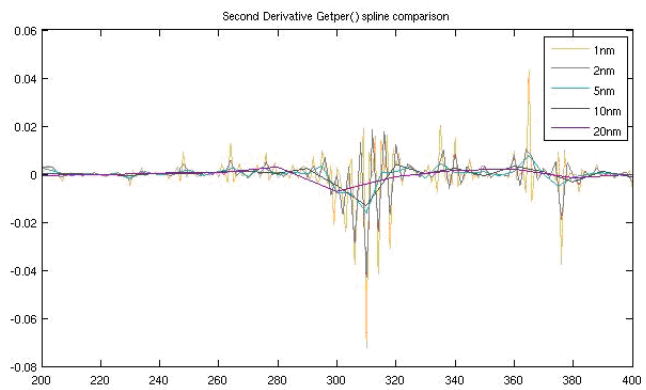


Figure 5.3: Second Derivative of Cubic Spline: $\Delta\lambda=1,2,5,10,20\text{nm}$

permittivity data. The AMPL model uses this cubic spline to compute the permittivity of silver. The coefficients of the cubic polynomials are provided in the model via nested “if-then-else” statements, producing some limitations in the model as will be seen in the results. A one dimensional AMPL model is built, and provided good information on the performance of the optimization solvers. This is discussed next.

5.3.2 Results of 1D AMPL model

A preliminary AMPL model for the 1-D Maxwell’s equation was tested. The finite element matrix $Ku = b$ was decomposed into real components K_r, u_r, b_r and imaginary ones K_i, u_i, b_i obtaining a $2n \times 2n$ system of equations

$$K = \begin{pmatrix} K_r & -K_i \\ K_i & K_r \end{pmatrix} \begin{pmatrix} u_r \\ u_i \end{pmatrix} = \begin{pmatrix} b_r \\ b_i \end{pmatrix}.$$

The one dimensional problem using a length of $L = 200$ nm assumes that a silver element with variable permittivity $\epsilon_s(\lambda)$ is located in the interval $z = [100, 150]$ nm. To illustrate the process, for $n = 4$ elements we obtained the following submatrices:

$$K_r = \frac{1}{6h} * \begin{pmatrix} a_1 & a_2 & 0 & 0 & 0 \\ a_2 & a_1 + a_1 & a_2 & 0 & 0 \\ 0 & a_2 & a_1 + a_{1\epsilon} & a_{2\epsilon} & 0 \\ 0 & 0 & a_{2\epsilon} & a_{1\epsilon} & a_2 \\ 0 & 0 & 0 & a_2 & a_1 \end{pmatrix}$$

$$K_i = \frac{imag(\epsilon_s)hc}{6} \begin{pmatrix} k & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & k \end{pmatrix} \text{ and}$$

$$b_r = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, b_i = \begin{pmatrix} 2k \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{The parameters are given in the table.}$$

Table 5.1: AMPL Assembly Parameters

| parameter | value |
|-----------------------|---|
| a_1 | $6 - 2h^2k^2$ |
| a_2 | $-6 - 2h^2k^2$ |
| $a_1 + a_1$ | $12 - 4h^2k^2$ |
| $a_1 + a_{1\epsilon}$ | $12 - 2(1 + \text{real}(\epsilon_s))h^2k^2$ |
| $a_{2\epsilon}$ | $-6 - \text{real}(\epsilon_s)h^2k^2$ |
| k | $2 * \pi / \lambda$ |
| b | $-\frac{\text{imag}(\epsilon_s)hk^2}{6}$ |
| L | length = 200nm |
| $nele$ | number of elements = 4 |
| h | $L/nele$ |

The problem was reduced to consider wavelengths $200 \leq \lambda \leq 400$ because the cubic spline obtained for silver permittivity in $200 \leq \lambda \leq 2000$ exceeds AMPL's capabilities to program nested "if-then-else" statements. The optimization ran with the solvers: LOQO 6.01, SNOPT 7.2-8 and MINOS 5.5. Only SNOPT 5.3-5 was able to produce the maximal wavelength solution: $\lambda = 200$. Results for AMPL's performance are shown in table 5.2.

The table shows that for 180 1D elements SNOPT could not produce a solution. This shows that the size of the discretization affects the performance of the optimization solver. Another important observation is the size limitation on the problem that will affect greatly a 2D model.

5.3.3 2D AMPL Model

The greatest advantage of AMPL optimization is the possibility of finding a simultaneous solution of the PDE constraint and the optimization of the objective function. Nonetheless the size of the mesh, and some limitations in the definition of the permittivity using cubic

| Table 5.2: AMPL 1D Model Performance | |
|--------------------------------------|----------------------------|
| N-elements | AMPL-SNOPT time statistics |
| 20 | 0.03s |
| 40 | 0.06s |
| 80 | 0.12s |
| 100 | 0.22s |
| 120 | 0.38s |
| 180 | Cannot improve |
| 200 | 1.56s |
| 400 | 10.09s |
| 500 | 14.62s |
| 600 | Not enough Memory |

splines were challenging. The initial mesh size was too big for the SNOPT solver to obtain the solution to the FEM constraint. For a reduced mesh on a smaller air square ($s = 150nm$) and $21^2 = 441$, nodes corresponding to $h = 7.5nm$ SNOPT's solutions was unattainable (error message: *Not enough storage for the basis factors*). Interior point method solver LOQO correctly solved the sparse linear system but was unable to solve the optimization problem. A model *test11.mod* included in the Appendix A, overcame the size limitation by taking into account the symmetry of the problem. For a grid of uniform size 7.5 nm this strategy reduces the variables from 441 to 231. Graphs 5.4-5.5 show the original domain and the reduced domain due to $y = 75nm$ line symmetries on the geometry, material and boundary conditions presented. A comparison between the values of the scalar magnetic field H_y for full and half computational domain is illustrated in the graphs 5.6 and 5.7 that show very good visual correspondence of the real the magnetic field. The objective function used in the AMPL model was simplified to the computation of light absorption at interior nodes of silver. This simplification avoids the introduction of constants from the numerical trapezoidal rule since all nodes are equally weighted in the summation. Using only interior nodes also avoid the numerical errors associated with the silver square interface with air, and sharp corners. The objective function that models absorption at the interior nodes of

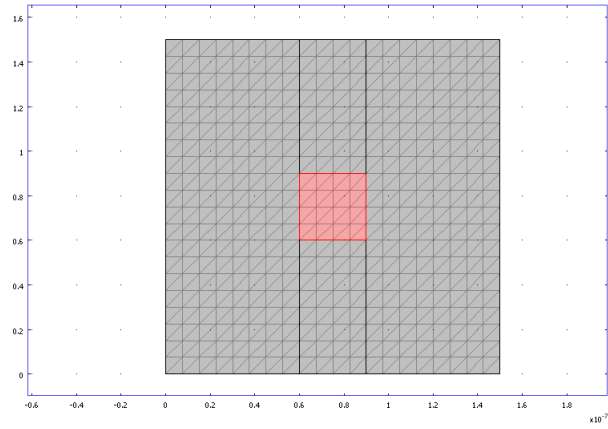


Figure 5.4: Full 21 Node Uniform Mesh

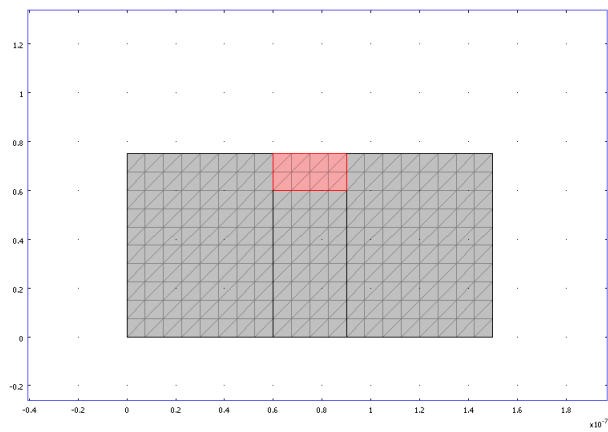


Figure 5.5: Half Uniform Mesh Symmetry

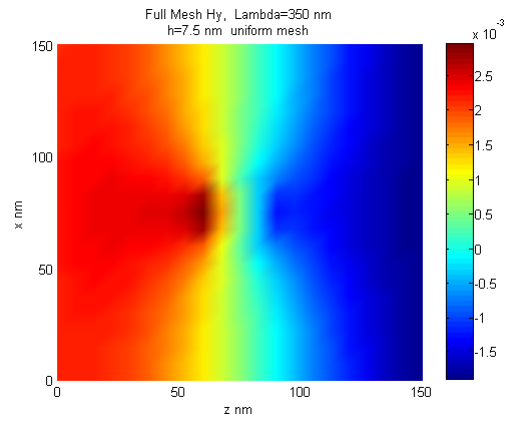


Figure 5.6: Solution in full domain

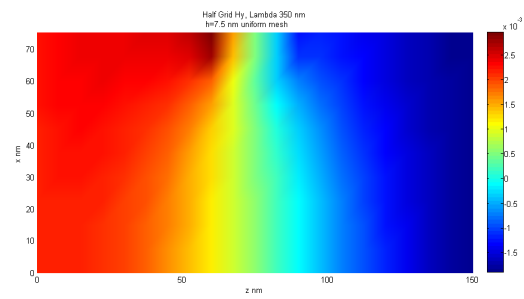


Figure 5.7: Solution in half domain

a 30 nm silver square is defined

$$J_{int}(\lambda) = \text{imag}(\epsilon(\lambda)) \sum_n \text{real}(Ex)_n^2 + \text{imag}(Ex)_n^2 + \text{real}(Ez)_n^2 + \text{imag}(Ez)_n^2,$$

where n are the interior nodes of the silver square. The graph of J_{int} is included in figure 5.8.

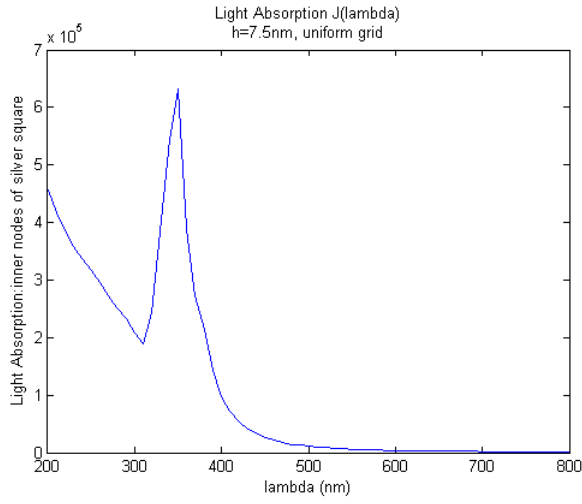


Figure 5.8: Light Absorption

An equivalent formulations J_c for J_{int} was introduced to submit the model for optimization via NEOS-Server [27], [28],[29]. J_c measures absorption at interior nodes in a horizontal line. Reciprocal variables were redefined to reduce round-off errors. The constant parameter $0.25h^2$ has been omitted as it will not change the maximizer nor the shape of J_c . Initialization of variables included $\lambda = 310$ nm and its corresponding permittivity values for silver. The expected local maximizer $\lambda^* = 350$ nm was obtained by the NEOS solver IPOPT [30] in $\lambda \in [300, 400]$ nm at an initial value of $\lambda_0 = 310$ nm. A closer look at the J_c graph in Figure 5.9 around the maximizer $\lambda^* = 350$ nm reveals the jagged shape of this challenging function. Table 5.3 summarizes the optimization results for J_c obtained by IPOPT close to

$\lambda = 350$ nm, taking half of the increasingly refined grids with 21^2 , 31^2 and 51^2 nodes.

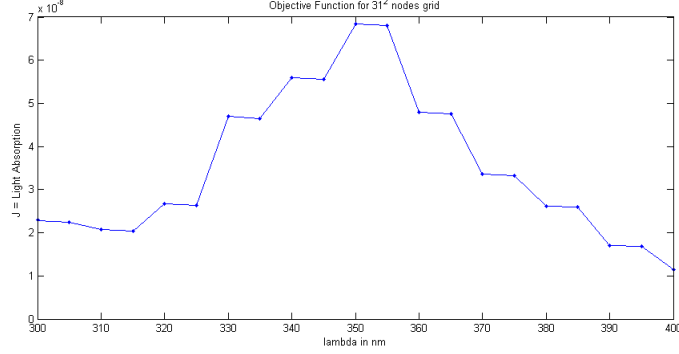


Figure 5.9: Refined Grid J_c in $[300,400]$ nm

Table 5.3: Optimization Results

| n | λ^* | J |
|------|-------------|-------------|
| 231 | 349.887 nm | 3.80358e-08 |
| 496 | 349.92 nm | 2.66856e-8 |
| 1326 | 349.94nm | 1.66275e-8 |

The error to satisfy the PDE constraint and solution time on NEOS server are summarized in Table 5.4. The error norm shown is taken from the real part of the magnetic field. The imaginary part gives similar results. The NEOS-solver IPOPT 3.10.1 solved the problem

Table 5.4: Optimization Solver Statistics

| n | iterations | l_2 norm (Real(H)) | CPU time |
|------|------------|----------------------|--------------|
| 231 | 5 | 5.99e-11 | 0.003s(NEOS) |
| 496 | 5 | 2.29e-11 | 0.008s(NEOS) |
| 1326 | 6 | 8.18e-12 | 0.048s(NEOS) |

consistently in half domain grids of 21^2 , 31^2 , 51^2 nodes. The fact that for a refined grid of a thousand nodes the solver took .048 seconds is very promising for this large scale PDE constraint optimization. Numerical optimization using modeling languages and optimization solvers is currently a strategy of interest for optimization based in simulation as seen in the benchmark effort lead by H.D. Mittelmann in [31].

Regarding the PDE constraint, when fixing $\lambda^* = 350$ nm, a solution obtained from IPOPT

matched the corresponding FEM MATLAB solution for the magnetic field, using symmetries. The electromagnetic field matched MATLAB's values for H_y , and the electric field variable was defined by central difference as $Ex = -\frac{1}{j\omega\epsilon} \frac{\partial H}{\partial z}$, and $Ez = \frac{1}{j\omega\epsilon} \frac{\partial H}{\partial x}$. The visualization in graphs 5.6 and 5.7 show a match between the magnetic field on the full mesh and the more efficient symmetrical computation on the bottom half of the computational domain. A comparison of values shows differences between the half and full mesh with absolute error of order $O(10^{-5})$, relative error $O(10^{-3})$.

Despite the great advantage of solving a PDE constraint optimization as an AMPL model, there are various limitations. Results confirm the extreme sensitivity of J to small changes in λ . The wavelength, λ is an pervasive variable in all the components of the problem. Permittivity of silver at different wavelengths, the wave constant k , and consequently the FEM stiffness matrix, the electromagnetic fields H, E are all variables dependent on the wavelength in the AMPL model. Dependence on lambda of the gradient is then very nested as seen in the derivation of the gradient obtained in (5.11). The dependence on this one variable λ single-handedly makes this a large-scale problem, with dimension of the order of the mesh.

Another limitation is on the AMPL modeling language capabilities. The permittivity in this problem is modeled by a cubic spline approximation from an available database. In AMPL this amounts to a nested *if-then-else* structure that when bigger than 31 statements, will crash the program. Therefore $\lambda \in [200, 500]$ was a restriction imposed by the language, and not the NLP optimization solvers. Reducing the permittivity to a simpler interpolation equation like least squares approximation, can help the computations but hurt the accuracy of computations. An improvement can be attained by exploring the use of the *Brendel-Bormann* method to compute the permittivity of silver for visible light wavelengths.

A new approach is now developed using the MATLAB solver tools and developing new tools to evaluate the objective function J and its gradients. The gradient of $J(\lambda)$ is obtained first to be used in this next numerical optimization strategy.

5.4 Gradient of $J(\lambda)$

The first optimization problem under consideration is to find the wavelength λ that maximizes light absorption J of a nanoscale silver square SS in air. This is given in [26] and obtained in (5.9):

$$J(\lambda, \epsilon) = -\frac{0.5}{\epsilon_0} \iint_{\Omega} \frac{p_i}{\omega(p_r^2 + p_i^2)} [(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2] d\Omega.$$

Because λ is an ubiquitous parameter in this formulation, the numerical gradient is a combination of analytical as well as various finite difference approximations for permittivity $\epsilon = (p_r, p_i)$ and the electric field phasors obtained as finite difference of the magnetic field $(\delta u_z^r, \delta u_z^i)$. To obtain the numerical approximation of the derivative of J with respect to λ the following derivatives are needed:

$$\frac{\Delta p_i}{\Delta \lambda} = p'_i$$

$$\frac{d(\omega^{-1})}{d\lambda} = \frac{1}{2\pi c_0}$$

$$\frac{d(p_r^2 + p_i^2)^{-1}}{d\lambda} = -2(p_r^2 + p_i^2)^{-2}(p_r p'_r + p_i p'_i).$$

Permittivity is obtained numerically by interpolation. Therefore numerical derivative with respect to lambda, is obtained by finite differences and denoted p'_r, p'_i . By definition $\omega = \frac{2\pi c_0}{\lambda}$ and its derivative with respect to lambda $\omega' = -2\pi c_0 \lambda^{-2}$. Note also that in the objective function $\frac{1}{\omega} = \frac{\lambda}{2\pi c_0}$, and therefore $(\frac{1}{\omega})' = \frac{1}{2\pi c_0}$.

To obtain second derivatives, first assume $\delta u_{r,\lambda} = \delta u_{\lambda,r}$, for space variables $\vec{r} = \{x, z\}$. The second derivatives are obtained by computing $\frac{\partial u}{\partial \lambda}$, and then differentiating numerically by finite differences in space. Thus $\frac{\partial^2 u}{\partial \lambda \partial \vec{r}} \approx \delta^2 u_{\lambda,r} = \delta^2 u_{\lambda,r}^r + j\delta^2 u_{\lambda,r}^i$. To compute $\delta u_{\lambda} = \delta u_{\lambda}^r + j\delta u_{\lambda}^i$, the system of equations obtained by FEM discretization of the PDE is

differentiated at the element level, where K is the element's stiffness matrix.

$$Ku = b \Rightarrow K \frac{\partial u}{\partial \lambda} = \frac{\partial b}{\partial \lambda} - \frac{\partial [K]}{\partial \lambda} u.$$

The solution of the system δu_λ approximates $\frac{\partial u}{\partial \lambda}$.

To compute $\frac{\partial [K]}{\partial \lambda}$ note that by the finite element discretization $K = \epsilon(\lambda)^{-1} \iint_T B^T B - k^2(\lambda) N^T N dT$, where the matrices B, N are the linear interpolation functions, also called shape or test functions. These do not depend on λ . The permittivity gradient is obtained numerically $D_\lambda(\epsilon(\lambda))^{-1} = -\epsilon(\lambda)^{-2} * (p'_r(\lambda) + jp'_j(\lambda))$. The main result of this section is

$$\begin{aligned} \frac{\partial J}{\partial \lambda} = & -\frac{0.5}{\epsilon_0} \int \int_{SS} \frac{p'_i}{\omega(p_r^2 + p_i^2)} \left[(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2 \right] \\ & + \frac{p_i}{2\pi c_0(p_r^2 + p_i^2)} \left[(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2 \right] \quad (5.10) \\ & - \frac{2p_i(p_r^2 + p_i^2)^{-2}(p_r p'_r + p_i p'_i)}{\omega} \left[(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2 \right] \\ & + \frac{2p_i}{\omega(p_r^2 + p_i^2)} \left[\delta u_x^r \cdot \delta^2 u_{x,\lambda}^r + \delta u_x^i \cdot \delta^2 u_{x,\lambda}^i + \delta u_z^r \cdot \delta^2 u_{z,\lambda}^r + \delta u_z^i \cdot \delta^2 u_{z,\lambda}^i \right] dx dz. \end{aligned}$$

Naturally the next step is to implement the derived formulation, compare the accuracy of the gradient and find the maximizer for the light absorption objective function J .

5.4.1 Computational Results

The above formulation for δJ , the direct differentiation (DD) gradient, has been implemented in the MATLAB FEM solver tool as a new project. The project consists of the following tasks:

1. Create a uniform mesh of right triangles of size h in a 150 nm square of air with a 30 nm silver square centered at 75 nm.

2. Implement Finite Element Method to find the state variables that model the Magnetic phasor in 2D (assuming time harmonic).
3. Compute the Electric Field from the Central Finite Difference (special treatment of the interface silver-air includes using half step $h/2$).
4. Compute the Light Absorption integral on the silver square using only interior nodes.
5. Compute the gradient δJ according to the equation (5.11).
6. To compare results, compute the gradient of J using the finite difference approximation (FD) given as $\frac{J(\lambda+\Delta\lambda)-J(\lambda)}{\Delta\lambda}$, $\lambda \in [200, 400]$ nm.

The uniform grid with fixed size h simplifies assembly because the structure is known. Only the size of the silver square s and the mesh step h is needed, eliminating the need to read files to obtain coordinates and connectivity of each element. The grid also facilitates the computation of the electric field through central differences without interpolation. The implementation uses sets to identify the structure of the mesh in the same way the AMPL model was coded. The sets include: boundary nodes, corner nodes, and interior nodes of air and silver. Table 5.5 contains the numerical results comparing the DD based gradient with the FD gradient of J for a mesh with size $h = 7.5$ nm. Results are similar although a discrepancy for the local minimum $\lambda = 310$ nm is obtained in the gradient formulation. The graph for $J(\lambda)$ was obtained from a developed MATLAB code FEM solutions and is included in 5.8. It shows a cusp around $\lambda = 310$ nm that may account for the difficulty of the gradient to obtain the expected increasing J value. The gradient on depends on the direct differentiation of the magnetic field and the sensitivity of the electric field. The 2nd degree accurate central finite difference approximation of the sensitivity of the electric field efficiently computes the gradient from the available direct differentiation results. Nonetheless there are some errors inherent to finite difference schemes regarding the appropriate step size. The DD based gradient is also very computational. Numerical errors are present in the approximation of the derivatives of permittivity and matrix sensitivity, as well as the

numerical solution of the electromagnetic phasors.

Table 5.5: Delta J for lambda

| λ nm | Finite Difference Gradient 1.0e+13* | Obtained Gradient 1.0e+13* | J 1.0e+06 |
|--------------|--|-------------------------------|--------------|
| 200 | -1.19980 | -1.2538 | 1.30640 |
| 210 | -0.91770 | -1.0777 | 1.18640 |
| 220 | -0.71800 | -0.8129 | 1.09460 |
| 230 | -0.62180 | -0.6376 | 1.02280 |
| 240 | -0.60170 | -0.6326 | 0.96060 |
| 250 | -0.60540 | -0.6148 | 0.90050 |
| 260 | -0.61460 | -0.6217 | 0.83990 |
| 270 | -0.61330 | -0.6242 | 0.77850 |
| 280 | -0.56410 | -0.5875 | 0.71710 |
| 290 | -0.70140 | -0.5969 | 0.66070 |
| 300 | -0.58120 | -0.7424 | 0.59060 |
| 310 | 1.46950 | -0.0863 | 0.53250 |
| 320 | 4.98590 | 3.9088 | 0.67940 |
| 330 | 2.53520 | 4.7352 | 1.17800 |
| 340 | 3.24440 | 1.9880 | 1.43150 |
| 350 | -5.14470 | -0.1160 | 1.75600 |
| 360 | -3.70430 | -6.5292 | 1.24150 |
| 370 | -1.92250 | -1.5566 | 0.87110 |
| 380 | -2.33330 | -2.6004 | 0.67880 |
| 390 | -1.45370 | -1.8222 | 0.44550 |
| 400 | -0.82760 | -1.1014 | 0.30010 |

The results in table 5.5 and figure 5.8 show a surprisingly successful adjoint gradient. The maximizer of the light absorption function at interior nodes J_{int} of a 30 nm silver square in the center of a 150 nm square air domain occurs at the wavelength $\lambda = 350$ nm. The result is consistent with those published in [26] stating that for a 10 nm silver sphere embedded in glass the resonance wavelength is approximately 400 nm. The DD gradient is very close to the FD gradient, yet more computationally efficient. The biggest computational advantage of the adjoint gradient strives in the fact that the gradient is computed from the obtained FE solution, using the available stiffness matrix and solution vector as described in the derived gradient equation (5.11). The finite difference gradient, on the other hand, is in this case particularly costly because it requires at least two FE runs. In the complex valued model that the project uses, each linear system amounts to double the nodes sparse system.

This is significant consider that node-variables are of the order of 10^3 in most commonly used grids.

Moreover table 5.5 show the sensitivity of the DD based gradient to small steps. The cusp observed from figure 5.8 is a red flag. For $\lambda = 310$ nm DD-gradient does not match the information of the FD gradient for a step $\Delta\lambda = 10$ nm. After examining $J(310.05e - 9) = 5.322e + 05 < 5.3250e + 05$, it is confirmed that the DD-gradient gives the correct information for a small step showing that a refined mesh also would improve accuracy.

The optimization of light absorption as a function of wavelength, obtained from the inner nodes of the silver particle, is proven to be challenging for the gradient based optimization algorithms used in this project. The value of local extrema, both minimum and maximum changes naturally with the precision and size of the step. Nonetheless the computational results show that the computed gradients could correctly describe the behavior of the numerical objective function in most points for a given step size. The DD-gradient developed and implemented here gives an efficient and sensitive information of the gradient of this function. The computational results show that for a 30 nm silver particle embedded in a 150 nm air-square, the objective function modeling the light absorption at interior silver nodes has a local minimizer in the vicinity of $\lambda_* = 310$ nm, and a maximizer in the vicinity of $\lambda^* = 350$ nm.

5.5 Gradient of $J(\epsilon)$

A new proposed design optimization problem under study is to find the size of a nanoscale silver rectangle S in a 150 nm air square that maximizes light absorption J at a fixed wavelength $\lambda = 350$ nm. Because it is assumed that the bigger square will have more absorption two variations of the objective function are considered: J_c the absorption at the center node or center of mass, and $J_A = \frac{J}{A}$ the absorption per unit area.

In this work the dissertation explores the dependency of the objective function on the size of

silver rectangle through a change in the distribution of ϵ in the computational domain similar to topology optimization approach. The dependency of J on $\epsilon \in \mathbb{C}$ leads to modifying the value of ϵ according to the sign of the gradient. In maximizing J it is assumed that a positive gradient leads to increase in real or imaginary component of $\epsilon = \text{real}(\epsilon) + j \text{imag}(\epsilon) = p_r + jp_i$, and decrease otherwise. In order to increase the real part or imaginary part of ϵ the dissertation computes: $\frac{\partial J}{\partial p_r}$ and $\frac{\partial J}{\partial p_i}$. It is not clear, though, how to interpret the value of ϵ according to the gradient as p_r, p_i are independent variables, but in this application ϵ is either silver (ϵ_2) or air (ϵ_1). If the gradient of the real part and imaginary part do not agree toward any of one permittivity, the information will not be useful, because it leads toward a different maybe unknown material.

In this work there are two ways to define the gradient of a complex valued variable. First a complex gradient is considered obtained from the real and imaginary permittivity derivatives of objective function $J(\text{real}(\epsilon), \text{imag}(\epsilon))$. The "extension" of the complex gradient is defined in [24] as: $\frac{\partial J}{\partial \epsilon} = 0.5 \left(\frac{\partial J}{\partial p_r} - j \frac{\partial J}{\partial p_i} \right)$. Second, defining permittivity via the parameter t , $\epsilon(t) = \epsilon_1 + t(\epsilon_2 - \epsilon_1)$, $0 \leq t \leq 1$ gives a real-valued permittivity gradient of $J(\epsilon(t))$. A change of t gives either ϵ_1, ϵ_2 , or an intermediate value. The real $\epsilon(t)$ equation is naturally: $\epsilon^r(t) = \epsilon_1^r + t(\epsilon_2^r - \epsilon_1^r)$, and similarly for the imaginary $\epsilon^i(t)$. Visualization of the results guide shape modifications to maximize the function. Various models will be developed and implemented. This results summarized in a visualization serves to assess the value of the information given by the gradients.

This work defines three models to compute the complex gradient δJ of m design permittivity variables over the N elements of the design computational domain.

1. $m = N$, all elements in the design domain are design elements. This is a topology optimization approach where the design variable is allowed to change independently in all elements of the design computational domain.
2. $m = 2$ design variables (permittivity of air or silver). This requires assembly of all air and silver elements in the design domain. Preserving a square, the model assembles

an L structure: the left/bottom corner (L shape) boundary around the silver square for exterior (air) and interior (silver) elements.

3. $m < N$, define m rectangular strips surrounding the silver square. Each structure is composed of w elements, w =width of the silver square. The design variables are defined only around the silver/air interface, which is the significant area.

In this project the first and second models are studied for the light absorption $J_c(\text{real}(\epsilon), \text{imag}(\epsilon))$ objective function defined over the square and at the center node. The third model is developed and implemented on the $J(\epsilon(t))$ objective function.

5.5.1 Derivatives

The derivatives for the solution vectors with respect to the permittivity are obtained from the numerical solution of the PDE, the state equations, by the finite element method (FEM). The discretized FEM linear system $Ku = b$ gives u the numerical approximation vector to the magnetic field at each node, K the stiffness matrix, and b the load vector that models the light incident to the left boundary of the enclosing air square. Denote the relative permittivity $\epsilon_r^m = \epsilon^m$ and assume no design variable lies on the leftmost boundary where light is incident, then $\frac{\partial b}{\partial \epsilon^m} = 0$. The derivative of $Ku = b$ with respect to the design variables ϵ^m is given by

$$\frac{\partial K}{\partial \epsilon^m} u + K \frac{\partial u}{\partial \epsilon^m} = 0. \quad (5.11)$$

Then

$$K \frac{\partial u}{\partial \epsilon^m} = -\frac{\partial K}{\partial \epsilon^m} u. \quad (5.12)$$

Assuming that ϵ^m is only defined at element m , then because $f(z) = \frac{1}{\epsilon^m}$ is holomorphic the non-zero computation reduces to the corresponding element matrices

$$\frac{\partial K}{\partial \epsilon^m} = \frac{\partial [K^m]}{\partial \epsilon^m} = -(\epsilon^m)^{-2} \int_T B^T B \, dT,$$

on element T . The derivative of variable u with respect to the design permittivity at element m is thus obtained from

$$K \frac{\partial u}{\partial \epsilon^m} = \left((\epsilon^m)^{-2} \int_T B^T B \, dT \right) u. \quad (5.13)$$

In particular second derivatives can be obtained from 5.13 and finite difference in space $\vec{r} = (x, z)$ assuming $\delta^2 u_{\epsilon^m, \vec{r}} = \delta^2 u_{\vec{r}, \epsilon^m}$. The second derivatives with respect to $\epsilon^m = p_r + jp_i$, $\frac{\partial^2 u^r}{\partial p_r \partial \vec{r}} \approx \delta^2 u_{p_r, \vec{r}}^r$, $\frac{\partial^2 u^r}{\partial p_i \partial \vec{r}} \approx \delta^2 u_{p_i, \vec{r}}^r$, are obtained by 5.13 and finite differences on space \vec{r} . Let

$$M_{p_r} = \frac{\partial[K^m]}{\partial p_r} = M_{p_r}^r + jM_{p_r}^i.$$

Following the principle of separating equations into real and imaginary components with $\vec{r} = (x, z)$, then $K \delta u_{pr} = M_{p_r} u$ become the following system of equations,

$$\begin{bmatrix} K^r & -j K^i \\ j K^i & K^r \end{bmatrix} \begin{Bmatrix} \delta u_{pr}^r \\ j \delta u_{pr}^i \end{Bmatrix} = \begin{bmatrix} M_{pr}^r & -j M_{pr}^i \\ j M_{pr}^i & M_{pr}^r \end{bmatrix} \begin{Bmatrix} u^r \\ j u^i \end{Bmatrix}. \quad (5.14)$$

$M_{p_r} = \frac{\partial[K^m]}{\partial p_r} = \left(\frac{(p_i^2 - p_r^2) + 2jp_r p_i}{(p_r^2 + p_i^2)^2} \right) \int_T B^T B \, dT = M_{p_r}^r + jM_{p_r}^i$. Now for the uniform mesh, the integrals of the derivative of the shape functions $\int_T \nabla N_i \nabla N_j \, dT$, $j = 1, 2, 3$ in (3.9) at each element reduces to the following matrices where h = size of the isosceles right

triangle, $A = 0.5h^2$ area of triangular elements, and $0.5 = h^2/4A$,

$$M_{p_r}^r = \frac{p_i^2 - p_r^2}{(p_r^2 + p_i^2)^2} \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix} \quad (5.15)$$

$$jM_{p_r}^i = j \frac{2p_r p_i}{(p_r^2 + p_i^2)^2} \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}. \quad (5.16)$$

A corresponding formulation is derived for $M_{p_i} = \frac{\partial[K^m]}{\partial p_i} = \left(\frac{j(p_i^2 - p_r^2) - 2p_r p_i}{(p_r^2 + p_i^2)^2} \right) \int_T B^T B \, dT = M_{p_i}^r + jM_{p_i}^i$ by observing that $M_{p_i} = jM_{p_r}$. Thus the corresponding real and imaginary components of M_{p_i}

$$M_{p_i}^r = -M_{p_r}^i, \quad M_{p_i}^i = M_{p_r}^r$$

and

$$\begin{bmatrix} K^r & -j K^i \\ j K^i & K^r \end{bmatrix} \begin{Bmatrix} \delta u_{p_i}^r \\ j \delta u_{p_i}^i \end{Bmatrix} = \begin{bmatrix} -M_{p_r}^i & -j M_{p_r}^r \\ j M_{p_r}^r & -M_{p_r}^i \end{bmatrix} \begin{Bmatrix} u^r \\ j u^i \end{Bmatrix}. \quad (5.17)$$

5.5.2 Model 1

An objective function that is independent of the area of the silver square, J_c is defined as the light absorption at the central node of the silver square, as a measure of the strength at a center of mass of the silver square. Here the project considers a topology optimization approach by defining a permittivity variable ϵ_m at each element m , independently obtaining then as many variables as design elements in the uniform mesh. The objective function reduces to $J_c(\epsilon) = -0.5\epsilon_0\omega * \text{imag}(\epsilon) * E_c(\epsilon) \cdot \bar{E}_c(\epsilon)$, the light absorption at the center node with electric field E_c . Note also that no integration is needed in this simplified approach.

In a $L = 150 \text{ nm}$ air square, the center node of the silver square at $(x, z) = (75, 75) \text{ nm}$ corresponds to node number 481 on the $h = 5 \text{ nm}$ grid. The silver square is given by 7×7 uniformly spaced nodes corresponding to an area over the silver square $SS = (6 \text{ nm}) * h = 30 \text{ nm}^2$.

The objective function is then:

$$J_c(p_r, p_i, \delta u_z^r, \delta u_z^i, \delta u_x^r, \delta u_x^i) = -\frac{0.5\epsilon_i}{\omega\epsilon_0(\epsilon_r^2 + \epsilon_i^2)} [(\delta u_z^r)^2 + (\delta u_z^i)^2 + (\delta u_x^r)^2 + (\delta u_x^i)^2], \quad (5.18)$$

where $(\delta u_x^i)_c$, denotes the finite difference approximation of $\text{imag}\left(\frac{\partial u}{\partial x}\right)$, at center node u_c .

Assuming that the surrounding air elements are the design variable then

$$\begin{aligned} \frac{\partial J_c}{\partial p_r} = & \frac{-\epsilon_i}{\omega\epsilon_0(\epsilon_r^2 + \epsilon_i^2)^2} [(\delta u_x^r)(\delta^2 u_{x,p_r}^r) \\ & + (\delta u_x^i)(\delta^2 u_{x,p_r}^i) + (\delta u_z^r)(\delta^2 u_{z,p_r}^r) + (\delta u_z^i)(\delta^2 u_{z,p_r}^i)], \end{aligned} \quad (5.19)$$

$$\begin{aligned} \frac{\partial J_c}{\partial p_i} = & \frac{-\epsilon_i}{\omega\epsilon_0(\epsilon_r^2 + \epsilon_i^2)} (\delta u_x^r)(\delta^2 u_{x,p_i}^r) + (\delta u_x^i)(\delta^2 u_{x,p_i}^i) + (\delta u_z^r)(\delta^2 u_{z,p_i}^r) + (\delta u_z^i)(\delta^2 u_{z,p_i}^i). \end{aligned} \quad (5.20)$$

The second derivatives $\delta^2 u_{r,p_i}^i$ above are obtained from 5.13-5.17. Otherwise if the ϵ_r, ϵ_i are design variables, then

$$\begin{aligned} \frac{\partial J_c}{\partial p_r} = & \frac{p_i p_r}{\omega\epsilon_0(p_r^2 + p_i^2)^2} [(\delta u_x^r)^2 + (\delta u_x^i)^2 + (\delta u_z^r)^2 + (\delta u_z^i)^2] \\ & - \frac{p_i}{\epsilon_0\omega(p_r^2 + p_i^2)} [(\delta u_x^r)(\delta^2 u_{x,p_r}^r) + (\delta u_x^i)(\delta^2 u_{x,p_r}^i) + (\delta u_z^r)(\delta^2 u_{z,p_r}^r) + (\delta u_z^i)(\delta^2 u_{z,p_r}^i)], \end{aligned} \quad (5.21)$$

$$\begin{aligned} \frac{\partial J_c}{\partial p_i} &= \frac{p_i^2 - 0.5(p_r^2 + p_i^2)}{\omega \epsilon_0 (p_r^2 + p_i^2)^2} [(\delta u_z^r)^2 + (\delta u_z^i)^2 + (\delta u_x^r)^2 + (\delta u_x^i)^2] \\ &- \frac{p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} [(\delta u_x^r)(\delta^2 u_{x,p_i}^r) + (\delta u_x^i)(\delta^2 u_{x,p_i}^i) + (\delta u_z^r)(\delta^2 u_{z,p_i}^r) + (\delta u_z^i)(\delta^2 u_{z,p_i}^i)]. \end{aligned} \quad (5.22)$$

When the function is real-differentiable, that is $f(z, \bar{z}) : \mathbb{R}^2 \rightarrow \mathbb{R}$, the complex gradient of the objective function with respect to real and imaginary components for the complex variables [24],[32],[33], in this case ϵ_m is

$$\delta J(p_r, p_i, \delta u_z^r, \delta u_z^i, \delta u_x^r, \delta u_x^i) = \frac{1}{2} \left(\frac{\partial J}{\partial p_r} - j \frac{\partial J}{\partial p_i} \right). \quad (5.23)$$

This numerical extension imposes much more structure to the problem than needed. The next model would find a new formulation of the problem that does not require a vector-valued complex gradient.

5.5.3 Model 2

The objective function as developed in (5.9) depends on the area of the square, as it is expected greater absorption in a bigger square. A more interesting alternative objective function is obtained by dividing J by the Area A of the silver square: $J_A(\epsilon) = J(\epsilon)/A(\epsilon)$. This new problem seeks to maximize the absorption per unit area of the silver nanoparticle. The permittivity of the element ϵ_r^m is assumed constant within each element T or structure of elements S . Permittivity of air (ϵ_1) or silver (ϵ_2) for a fixed wavelength λ is defined in terms of a parameter $t = 0, 1 : \epsilon(t) = \epsilon_1 + t(\epsilon_2 - \epsilon_1)$. Therefore t increases from air $t = 0$ to silver $t = 1$. The derivative with respect to the parameter t is obtained by the quotient rule: $\delta J_A = \frac{\delta J * A - J * \delta A}{A^2}$. At a given distribution of ϵ_r , this gradient will help determine if a change in the size of the square will increase the new objective function J_A , subject to the field equations derived from Maxwell's Equations in the 2D $\vec{r} = (x, z)$ plane.

The J in J_A (5.9) is obtained on the partial derivatives of the magnetic field u with respect to

the space variables $\vec{r} = (x, z)$, *real* $(\frac{\partial u}{\partial \vec{r}})$, *imag* $(\frac{\partial u}{\partial \vec{r}})$. These are approximated by their corresponding finite difference forms $\delta u_x^r, \delta u_z^i, \delta u_x^r, \delta u_z^i$. The complex variable $\epsilon(t)$ is decomposed in real and imaginary components as: $\epsilon^r(t) = \epsilon_1^r + t(\epsilon_2^r - \epsilon_1^r) = p_r$, $\epsilon^i(t) = \epsilon_1^i + t(\epsilon_2^i - \epsilon_1^i) = p_i$. The gradient is denoted $\delta J = \frac{\partial J}{\partial f} \frac{\partial f}{\partial p_r} \frac{dp_r}{dt} + \frac{\partial J}{\partial f} \frac{\partial f}{\partial p_i} \frac{dp_i}{dt}$, where $f = \{\delta u_z^r, \delta u_z^i, \delta u_x^r, \delta u_x^i\}$. The main derived formulas used to obtain $\delta J = \frac{\partial J}{\partial p_r} \frac{dp_r}{dt} + \frac{\partial J}{\partial p_i} \frac{dp_i}{dt}$ are:

$$\begin{aligned} \frac{\partial J}{\partial p_r} = \iint & -\frac{p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} [(\delta u_z^r)(\delta^2 u_{p_r, z}^r) + (\delta u_z^i)(\delta^2 u_{p_r, z}^i) + (\delta u_x^r)(\delta^2 u_{p_r, x}^r) + (\delta u_x^i)(\delta^2 u_{p_r, x}^i)] \\ & + \frac{p_r p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)^2} [(\delta u_z^r)^2 + (\delta u_z^i)^2 + (\delta u_x^r)^2 + (\delta u_x^i)^2] d\Omega \end{aligned} \quad (5.24)$$

$$\begin{aligned} \frac{\partial J}{\partial p_i} = \iint & -\frac{p_i}{\omega \epsilon_0 (p_r^2 + p_i^2)} [(\delta u_z^r)(\delta^2 u_{p_i, z}^r) + (\delta u_z^i)(\delta^2 u_{p_i, z}^i) + (\delta u_x^r)(\delta^2 u_{p_i, x}^r) + (\delta u_x^i)(\delta^2 u_{p_i, x}^i)] \\ & + \frac{0.5(p_i^2 - p_r^2)}{\omega \epsilon_0 (p_r^2 + p_i^2)^2} [(\delta u_z^r)^2 + (\delta u_z^i)^2 + (\delta u_x^r)^2 + (\delta u_x^i)^2] d\Omega. \end{aligned} \quad (5.25)$$

Some interesting results are presented in the next section.

5.5.4 Results

The MATLAB tools developed to compute the FEM solution and DD-gradients of the silver absorption function are the first contribution of this work. These tools accomplish the following tasks:

1. FEM 2D Helmholtz equation solver with absorbing boundary conditions that uses faster assembled matrices given the geometrical information.
2. Assemble matrices by knowing position and size of the silver nanoparticle and the size h of a uniform grid.
3. Compute the light absorption of the silver nanoparticle from the magnetic and FD

obtained electric phasor by 2D trapezoid integration over the entire design domain.

4. Compute the gradient of real and imaginary components of the objective function J , with respect to t given $\epsilon(t) = \epsilon_1 + t(\epsilon_2 - \epsilon_1)$ using the derived formulation.
5. Compute the gradient of a structure of elements strips surrounding the initial silver nanoparticle to assess movement of the silver/air boundary and therefore a change of size.

The study of the absorption of a silver nanoparticle using a topology optimization approach has developed many formulations in the area of complex valued functionals and its gradients. The gradients obtained have pointed to an increase in the size of the silver rectangle increases the absorption per area, Model 2. Therefore the maximum absorption per area is maximized when the silver particle is at its maximum allowed size in the domain. This result is not obvious because the absorption over the nanoparticle is divided by an area that is increasing with size.

The objective function computed by the tools is shown to be dependent on the quality of the mesh. The initial uniform mesh failed to give correct values for objective function. The MATLAB code objective function, in agreement with the DD-based gradient, gave decreasing values as the silver rectangle increased vertically. This was verified and discarded as the mesh was refined in the sharp corners using a unstructured mesh in COMSOL-Multiphysics 3.5.

The following graphs were obtained by COMSOL. The graphs show that for symmetric rectangular shapes, circle and triangle with the same area, the square maximizes absorption per area. The graphs generated with constant electric field range, show that the square has greater extreme values around the boundaries and corners than the other rectangular shapes with sharp corners. The numerical data for the absorption statistic for each geometric figure of same area is summarized in 5.6. These values are obtained from COMSOL with refined unstructured mesh with greater than $2e+004$ elements. A proposed problem for shape optimization, is then to prove or disprove that the square is the shape that optimizes light

absorption per area.

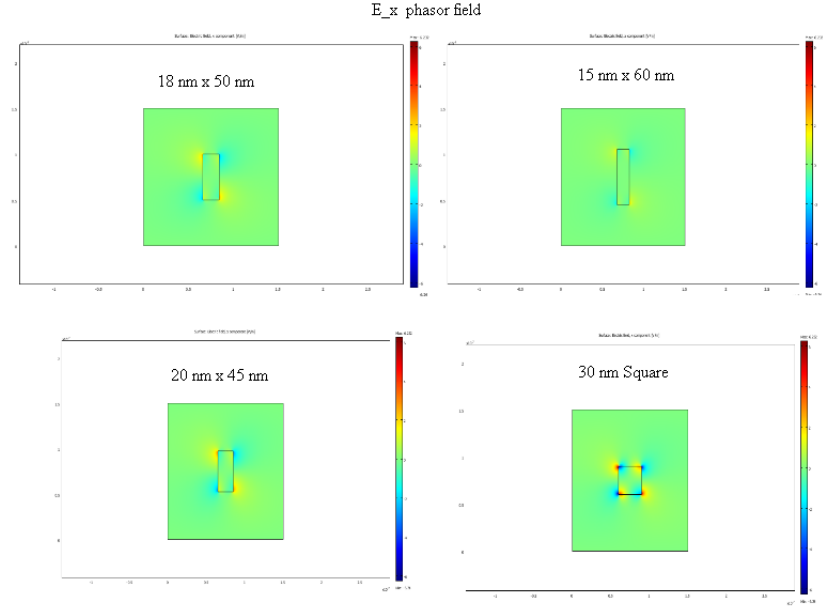


Figure 5.10: E_x comparison for 900 sq nm rectangles I

The next chapter tackles the largest optimal design project in this dissertation work by using the same consistent strategies to obtain the gradient and add some new developments.

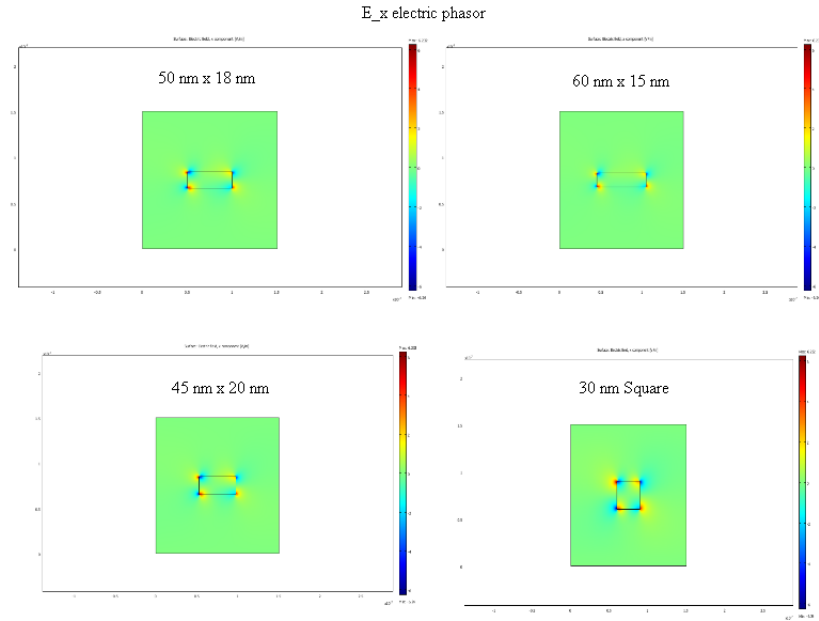


Figure 5.11: E_x comparison for 900 sq nm rectangles II

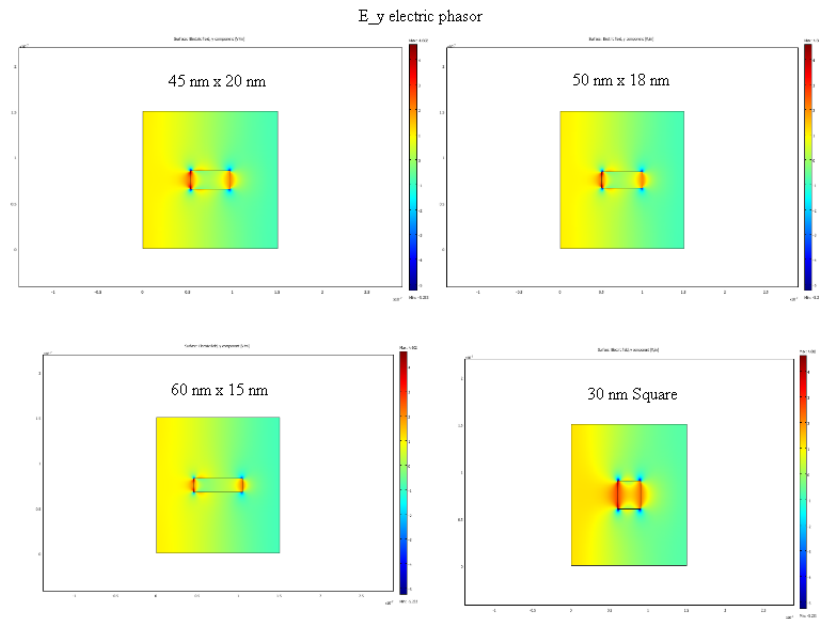


Figure 5.12: E_y comparison for 900 sq nm rectangles I

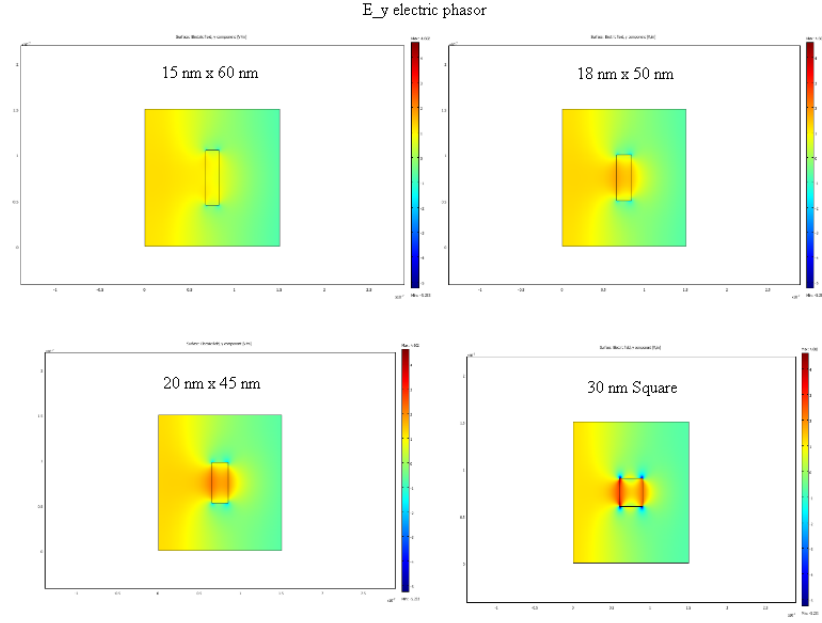


Figure 5.13: E_y comparison for 900 sq nm rectangles II

Table 5.6: Absorption and norm(E) at center

| figure | dimensions (w x h) nm | (J, J_c) |
|-----------|-----------------------|------------------|
| square | 30 x 30 | (7.11e-11, 1.47) |
| rectangle | 45 x 20 | (3.87e-11, 0.83) |
| rectangle | 20 x 45 | (5.95e-11, 2.19) |
| rectangle | 50 x 18 | (3.32e-11, 0.76) |
| rectangle | 18 x 50 | (4.72e-11, 2.13) |
| circle | $r = 1.69257e-008$ | (4.62e-11, 1.85) |
| rectangle | 60 x 15 | (2.54e-11, 0.66) |
| rectangle | 15 x 60 | (3.26e-11, 1.96) |
| triangle | b=30 nm, h=60 nm | (4.67e-11, 2.09) |
| triangle | b=60 nm, h= 30 nm | (4.15e-11, 1.33) |

Chapter 6: Optimal Design of Plamon Coupler

6.1 Introduction

In quest for the design and development of smaller and more powerful devices, the ability to efficiently use energy is of central importance. Therefore the optimal design of the Plasmon Coupler is of great interdisciplinary interest for its value in enhancing/scattering electromagnetic energy. The grating coupler was introduced two decades ago as an efficient form of generating plasmons, becoming a very productive area of research. For example, a search in the Web of Science database for the grating coupler produced 1,883 papers since 1992. An efficient plasmon generator is thus an important contribution for practitioners to be able to succeed building better and useful devices.

Excitation of surface plasmons via photons of a light beam directed to a grating-like interface between a metal and dielectric, produces an enhanced electromagnetic field in a small vicinity of the grating grooves. An efficient resonance coupling drives the collective oscillation of the beam of light and electrons on the grating surface of the metal from an input wave to a plasmonic mode with less energy loss. The plasmonic mode excited in this way or SPP has shown great sensitivity to changes in geometry, and characteristics of the incident wave. A good design will promote that a greater percentage of the incident wave will be converted into the plasmonic mode. An illustration is shown in figure 6.1.

In this work the change in geometry and characteristics of the incident wave are optimized using gradient information. An optimal design with 14 variables is tackled using efficiently the available numerical data and derived formulas. A gradient obtained from numerical as well as analytical methods is computed and applied to the design of the grating coupler using a combination of direct differentiation and finite difference gradients.

In this chapter original work on the design of a plasmon coupler is presented. It consists

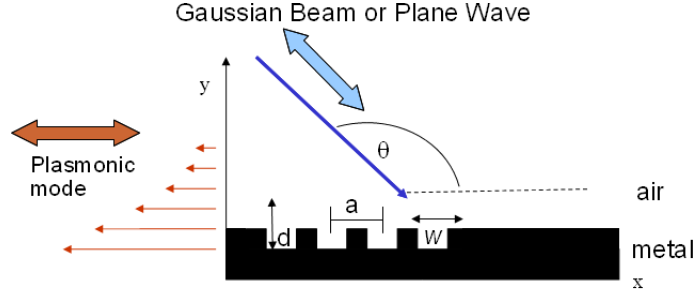


Figure 6.1: Coarse vs Refined Mesh Visualization

of two main experiments: the conversion of the plane wave, and the Gaussian beam. The problem formulation, including formulas that model the electromagnetic waves, the objective function and its gradient are derived. Some mathematical analysis for the application of the Adjoint Method is also presented. The computational tool developed using COMSOL Multiphysics and MATLAB tools, the geometry and modeling decisions are discussed. Very improved final designs are obtained, and analyzed, along with some suggestions for future work.

6.2 Formulation of the Problem

The problem development from governing equations, modeling assumptions and obtained formulas are presented in this section. Some important theoretical results from this section are the formulas derived for the objective functions and gradients of both electromagnetic waves: plane wave and Gaussian beam. The intention of sharing the steps in the development of a working model is to give interested readers insights to apply in this as well as other optimal design problems.

The conversion of energy between an electromagnetic wave and the plasmonic mode can be modeled in two equivalent directions by choosing which one will be incident at the boundary of the system. This defines an output or input coupler [23]. In this dissertation the output coupler is studied, where the plasmonic beam is incident to the boundary, and the energy transformed into an output electromagnetic wave is measured. This choice proves to

be very convenient because changes in the parameters of the output wave will modify the objective function, not the solution (state variables). The latter is more computationally expensive data. The quantity of energy converted from the plasmonic mode to a target output wave will measure the efficiency of the coupler. The conversion efficiency is measured as the percentage of the energy transferred from the source modes such as a plane wave or Gaussian beam to the plasmonic mode. The energy that the plasmon has lost as it travels the distance between the boundary and the grating is also taken into account.

6.2.1 Designing the Geometry

A simplified grating coupler consists of a rectangular domain with an interface between silver and air. The silver slab contains a grating-like surface that consists of a number of grooves. In the output coupler the plasmonic mode is incident to the leftmost boundary of the domain, and its coupling with an electromagnetic wave is measured at an horizontal boundary over the parallel silver surface. The size of the grating depends on the number of grooves.

As a guide for the initial design, the following measurements were obtained from practitioners in the NIST. The total computational domain is a rectangle of dimensions $9.0\text{e-}5$ m wide and $1.0\text{e-}5$ m high when the grating consists of an initial geometry of n grooves in the center. This is the design or variable part of the computational domain. Each groove has initial dimensions of 100 nm wide and 50 nm deep. The center of adjacent grooves is separated by 600 nm from a neighboring groove. This grating has initial dimensions $600 \times 50 = 3.0 \text{ e-}5$ m wide and $2.0\text{e-}7$ m high. An additional left and right domain at both sides of the design domain are used for boundary conditions, giving the total length of $9\text{e-}5$ m plus perfectly matched layers (PML) domains. The objective function boundary is at the top of the rectangle 8,000 nm away from the parallel silver surface. The figure 6.2.1 shows the geometry of the first computational domain. Table 6.1 described the geometric parameters of the initial design.

Table 6.1: Initial Design Parameters

| parameter | description | initial value |
|-----------|--|---------------|
| \bar{x} | fixed position of first groove | 3.01e-5 m |
| d | displacement between adjacent grooves | 600 nm |
| n | number of grooves | 50, 20, 4 |
| w | initial width of grooves | 100 nm |
| h | initial depth of a groove | 50 nm |
| Lm (Rm) | left (right) margin of computational domain | 30,000 nm |
| PML | perfectly matched layer domains top/right/left | 1,000 nm |
| hsil | depth of silver slab | 200 nm |
| B | objective function boundary | 8,000 nm |

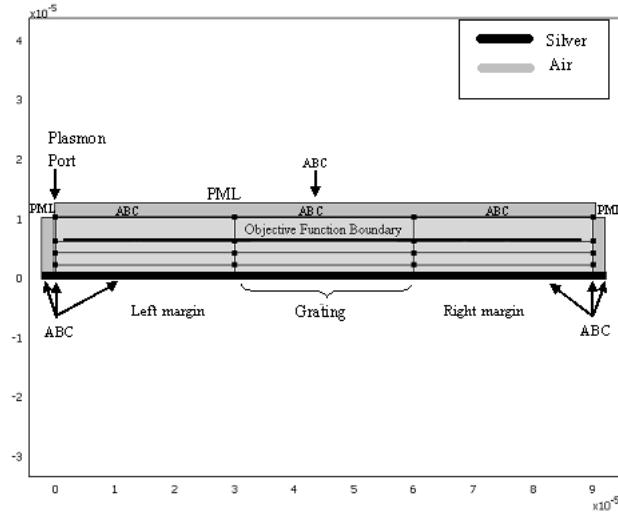


Figure 6.2: Initial Computational Domain

Usually problems in computational electromagnetism require specialized boundary conditions capable of modeling an incident electromagnetic mode while allowing other modes to exit the system with minimal interaction. These specialized boundary conditions are modeled by using a combination of perfectly matched layers (PML) domains, absorbing boundary conditions (ABC) and ports at the surrounding boundaries of the a computational domain. These are discussed in [15], and included in COMSOL's RF-module. The computational domain with the boundary conditions is illustrated in figure 6.2.1.

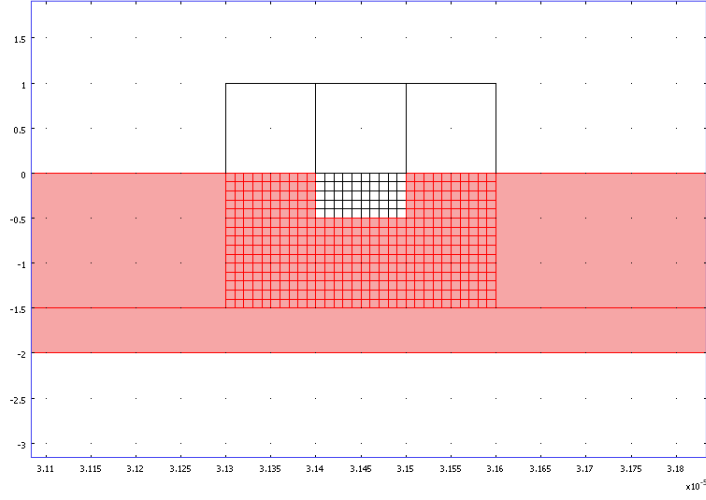


Figure 6.3: Final groove design

An important development of the work happened during the design of the geometry of the grooves. Originally the grating contained 50 grooves. The size of the grating was to be rescaled as a single composite structure consisting of only one design variable. The grating would be compressed or stretched according to a transformation factor. A second approach would allow modification of the grooves only. For a set of 20 grooves the right boundary of each groove would be displaced simultaneously using one displacement parameter. A design node on each groove determined the position of the groove. The geometry changes according to the new position of the design nodes. Unfortunately this new geometry naturally required a new mesh, or a moving mesh algorithm. The problem with this approach is that it was found that modifying the mesh alone without modifying the actual geometry may significantly affect the value of the objective function. Therefore a new design is proposed that provides the desired control of the geometry of the grooves without the unwanted consequences of redrawing/remeshing.

The final geometry of the groove is shown in figure 6.2.1. This geometry allows discrete changes of all three boundaries for each rectangular groove: the right, left and bottom boundaries according to the size of the introduced grid. The top boundary is not modified

Design Modifications of Grooves

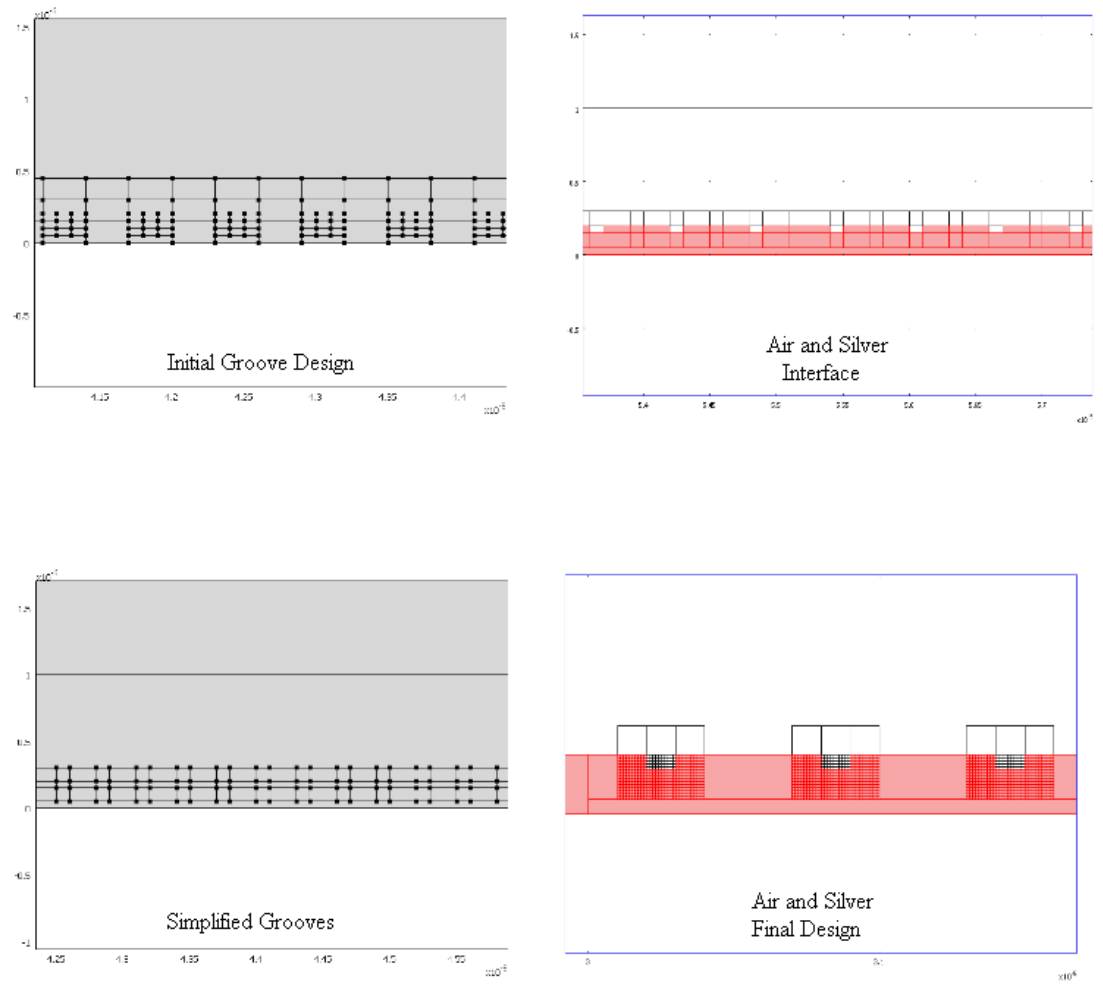


Figure 6.4: Design of Grooves

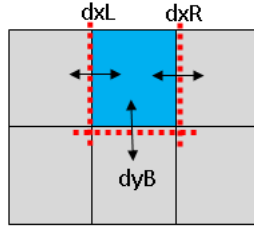


Figure 6.5: Three Parameter Displacements

as it is the fixed interface between air and the silver slab. The design is updated via three independent parameters for each groove. In this way each groove would change width, depth and separation from adjacent grooves independently. The geometry design provided for a way to modify the rectangular grooves without moving the mesh. The groove's geometry consists of 10 nm squares in a 300 by 150 grid. Each small square could be identified as air or silver, thus modifying the width and depth of any groove in discrete steps of 10 nm. A very important computational gain obtained with this design is that it does not require remeshing as the design changes. From the optimization viewpoint this provides a desirable effect because there is no need to account for how the mesh change affects the optimization criteria. After defining an initial good quality mesh, it remains fixed during optimization and therefore would not introduce variation to the objective function. Changes in the value of the objective function can be better understood as a consequence of changes in the material distribution, that is, groove size and position. The evolution of the grooves design is shown in figure 6.4.

The displacement design variable of the left (dxL), right (dxR) and bottom (dyB) interfaces of air and silver can either be increased, decreased or not moved at all. This gives $3^3 = 27$ geometric design possibilities for each groove. An illustration of all three displacements for a groove is given in figure 6.5. The design can then be modified geometrically in $27n$ ways. Because the first groove's left boundary is to remain fixed, in the final implementation work with 4 grooves there are $4(3) - 1 = 11$ geometric parameters to conduct the geometrical

optimization of the grating.

Once the geometry has been designed, the objective function measures the performance of a particular grating design. The objective function P , its normalization and decay coefficient computation are presented next.

6.2.2 The Objective Function

The choice of the objective function in this work is different from other optimal design work on the grating coupler [22],[34],[23]. Published work in the design of this device consider the Poynting vector as a measure of performance [22],[34],[23]. This far-field approach measures power in general, and therefore results are the same no matter what electromagnetic wave is used. In this work the objective function is tailored to a particular output wave so that results differ greatly between a plane wave and a Gaussian beam. The practitioner has more information on the expected conversion of the design when choosing a wave to irradiate the silver grating. This near-field information is necessary in order to build a device that will perform best for a particular source of electromagnetic radiation.

The equation of a wave coupler is given by the line integral over a determined boundary Γ in the computational domain.

$$J = \frac{1}{4} \int_{\Gamma} ((\bar{E}_B \times H) + (E \times \bar{H}_B)) \cdot < 0, 1, 0 > ds, \quad (6.1)$$

where $\{E_B, H_B\}$ are the electric and magnetic field of the source, $\{E, H\}$ the FEM approximation of the field in the computational domain induced by the plasmonic mode. The objective function is taken to be the square of the magnitude of the integral in (6.1):

$$P(J) = \text{real}(J)^2 + \text{imag}(J)^2 = |J|^2. \quad (6.2)$$

This is then normalized by dividing by the coupler of a wave with itself P_N , so that finally the objective function is defined as $P = P(J)/P_N$.

When the air/silver interface occurs at $y = 0$, the magnetic field of the plasmonic mode is modeled analytically by the expression in [35]:

$$H_z(y) = \begin{cases} A_1 e^{-k_2 y} & y \geq 0 \\ A_2 e^{k_1 y} & y < 0 \end{cases} \quad (6.3)$$

$$k_2 = \sqrt{\beta^2 - k_0^2},$$

$$\beta = k_0 \sqrt{\frac{\epsilon_2}{1 + \epsilon_2}}$$

$$k_0 = 2\pi/\lambda, \quad k_1 = -k_2 \epsilon_2,$$

where ϵ_2 is the permittivity of silver, the permittivity of air is $\epsilon_1 = 1$, and k_0 the wave propagation constant. The energy decay or evanescence factor is a function of β , the propagation constant, the magnetic field of the plasmonic mode in (6.3), and x_0 , the distance from the incidence boundary. The coefficient is obtained as: $c^2 = e^{real(-2j\beta x_0)}$. Therefore since $P = c^2 P_0$, the initial power at the boundary $P_0 = P/c^2$. In this work the decay factor $1/c^2 \approx 1.6159$, with $real(\beta) = 7.998326e3$, $x_0 = 3.0e-5$ meters.

The objective function and its gradient depend on the particular equation of the mode to be coupled with the EM field. The problem is modeled in a two dimensional xy plane, and the time-harmonic scalar Helmholtz equation in H_z models the electromagnetic field. The corresponding electric phasor vector is now $\langle E_x, E_y, 0 \rangle$. A combination of numerical and analytic strategies are explained next to exploit the available information from the solution in obtaining the gradient in the following. These derivations obtained for the plane wave and the Gaussian Beam as he modes coupled with the EM field, starting with the plane wave.

6.3 Plane Wave

The plane wave is a suitable approximation of most light waves. It is characterized by plane wavefronts, and describes a perfectly monochromatic parallel beam of light. Because the laser beam can be modeled as a plane wave in a localized region, the plane wave is used by practitioners to study absorption and scattering of light [36].

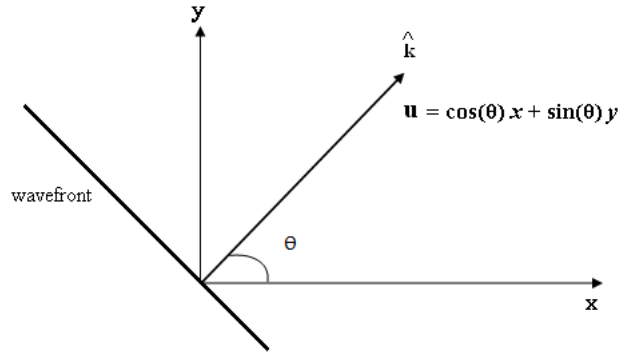


Figure 6.6: Plane Wave Parameters

The plane wave is characterized by the propagation vector \hat{k} in the direction perpendicular planar wavefront. Because in this experiment light propagation is assumed time-harmonic and the free space wavelength is fixed, the only parameter that modifies the equation of the plane wave is the angle between the vector \hat{k} and the positive direction of the x axis, see figure 6.6. Because the plane wave can be generated with a laser and a simple mathematical formalism, the PW was selected as the first option for the plasmon generation. This starting experiment is described next.

6.3.1 Objective Function and Gradients

To obtain the equation for the plane wave in 2D the angle of incidence is measured counter-clockwise from the positive direction of the x axis in the xy plane, such that $0 \leq \theta \leq \pi$, with

wave propagation vector $\vec{k} = \langle k \cos(\theta), k \sin(\theta), 0 \rangle$, as seen in figure 6.6. The formulation then starts with the generalized equation of the plane wave in 2D. The derivations are obtained by the time harmonic convention $H(r, t) = H(r)e^{j\omega t}$, $r = \langle \cos(\theta)x, \sin(\theta)y, 0 \rangle$, and the identity from Maxwell's equation $j\omega\epsilon\mathbf{E} = \nabla \times \mathbf{H} = \left\langle \frac{\partial H_z}{\partial y}, -\frac{\partial H_z}{\partial x}, 0 \right\rangle$, simplified by the fact $\mathbf{H} = H_z$ a scalar. Furthermore $\mathbf{E} = \nabla \times \mathbf{H} = -\frac{j}{\omega\epsilon} \left\langle \frac{\partial H_z}{\partial y}, -\frac{\partial H_z}{\partial x}, 0 \right\rangle$. Assuming air's relative permittivity to be $\epsilon_r = 1$, magnetic field constant $H_0 = (c_0\mu_0)^{-1} \approx 0.0026544$, and $\frac{k}{\omega} = \frac{1}{c_0}$, the 2D plane wave is then modeled by:

$$H_z(x, y, \theta, t) = H_0 e^A,$$

$$\mathbf{E} = \langle E_x, E_y, 0 \rangle = \frac{H_0}{c_0\epsilon_0} \langle -\sin(\theta)e^A, \cos(\theta)e^A, 0 \rangle, \quad (6.4)$$

where $A = j(\omega t - k \cos(\theta)x - k \sin(\theta)y)$. The objective function $P(J)$ is obtained from the square of the magnitude of the functional in (6.1) taking as the coupled propagation modes the plane wave (E_B, H_B) , and the electromagnetic field produced by an incident plasmonic mode (E, H) over the boundary Γ .

To measure the percentage of conversion the results are normalized by obtaining the coupling coefficient of the plane wave coupled with itself. A general cross product of 3D vectors gives the following:

$$\mathbf{E} \times \mathbf{H} = \langle e_y h_z - e_z h_y, e_z h_x - e_x h_z, e_x h_y - e_y h_x \rangle$$

$$= \langle e_y h_z, -e_x h_z, 0 \rangle, \quad (6.5)$$

after simplifying by the definition of phasors $H_z, \langle E_x, E_y \rangle$. Then (6.1) reduces to:

$$\begin{aligned} J &= \frac{1}{4} \int_{\Gamma} (\langle \bar{e}_y h_z, -\bar{e}_x h_z, 0 \rangle + \langle e_y \bar{h}_z, -e_x \bar{h}_z, 0 \rangle) \cdot \langle 0, 1, 0 \rangle ds \\ &= \frac{1}{4} \int_{\Gamma} -\bar{e}_x h_z - e_x \bar{h}_z ds. \end{aligned} \quad (6.6)$$

Substituting (6.4) in (6.6), $H_0 \in \mathbb{R}, (e^{\bar{A}}) = e^{-A}$ the integral becomes a real number:

$$\begin{aligned} J_N &= \frac{\bar{H}_0 H_0}{4\epsilon_0 c_0} \int_{\Gamma} e^A e^{-A} (2 \sin(\theta)) ds \\ &= \frac{H_0^2}{2\epsilon_0 c_0} \sin(\theta) \int_{\Gamma} ds \\ &= \frac{H_0^2}{2\epsilon_0 c_0} \sin(\theta) s, \end{aligned} \quad (6.7)$$

(6.8)

s the length of the boundary Γ . For normalization $P_N = \text{sqrt}(J_N^2) = J_N$.

The integral equation for the coupled electromagnetic field excited by the plasmonic beam, and the output plane wave in (6.1) is obtained similarly. Substitution of (6.4) in (6.1) gives:

$$J = \frac{H_0}{4} \int_{\Gamma} e^{-A} \left(\frac{\sin(\theta)}{c_0 \epsilon_0} H_z + \frac{j}{\omega \epsilon_0} \frac{\partial H_z}{\partial y} \right) ds \quad (6.9)$$

$$-A = jk(\cos(\theta)x + \sin(\theta)y) - j\omega t. \quad (6.10)$$

From now on the time harmonic term $e^{-j\omega t}$ is omitted from the derivations in space.

To find gradients of a complex valued functional the real and imaginary parts are first

obtained with $e^{-A} = e^{ja} = \cos(a) + j \sin(a)$:

$$\begin{aligned} J^r &= \frac{H_0}{4\epsilon_0} \int_{\Gamma} \cos(a) \left(\frac{\sin(\theta)}{c_0} H^r - \frac{1}{\omega} \frac{\partial H^i}{\partial y} \right) - \sin(a) \left(\frac{\sin(\theta)}{c_0} H^i + \frac{1}{\omega} \frac{\partial H^r}{\partial y} \right) ds \\ J^i &= \frac{H_0}{4\epsilon_0} \int_{\Gamma} \cos(a) \left(\frac{\sin(\theta)}{c_0} H^i + \frac{1}{\omega} \frac{\partial H^r}{\partial y} \right) + \sin(a) \left(\frac{\sin(\theta)}{c_0} H^r - \frac{1}{\omega} \frac{\partial H^i}{\partial y} \right) ds, \end{aligned} \quad (6.11)$$

where $f^r = \text{real}(f)$, $f^i = \text{imag}(f)$ are real quantities, and $a = k(\cos(\theta)x + \sin(\theta)y)$.

The gradient is obtained in two parts: plane wave θ and geometric parameters \mathbf{v} . This is natural because the plane wave, in this project $\{E_B, H_B\} = H_0 e^{A} \left\{ \frac{-\sin(\theta)}{c_0 \epsilon_0}, 1 \right\}$, is independent of changes in the geometric parameters \mathbf{v} that affect the EM field. Likewise the field induced by the plasmonic mode is independent of the changes in the angle of the output plane wave. First the simpler gradient with respect to the Plane Wave parameter, θ is obtained.

To simplify the derivations note that equations (6.11) are of the form:

$$\begin{aligned} J^r &= \frac{H_0}{4\epsilon_0} \int_{\Gamma} \cos(a) F_1 - \sin(a) F_2 ds \\ J^i &= \frac{H_0}{4\epsilon_0} \int_{\Gamma} \cos(a) F_2 + \sin(a) F_1 ds, \end{aligned} \quad (6.12)$$

where $F_1 = \left(\frac{\sin(\theta)}{c_0} H^r - \frac{1}{\omega} \frac{\partial H^i}{\partial y} \right)$, and $F_2 = \left(\frac{\sin(\theta)}{c_0} H^i + \frac{1}{\omega} \frac{\partial H^r}{\partial y} \right)$. Then

$$\begin{aligned} D_{\theta} F_1 &= \left(\frac{\cos(\theta)}{c_0} \right) H^r \\ D_{\theta} F_2 &= \left(\frac{\cos(\theta)}{c_0} \right) H^i. \end{aligned}$$

The general form of the gradient is $D_\theta P = 2J^r(D_\theta J^r) + 2J^i(D_\theta J^i)$. The first part of the gradient is then,

$$D_\theta(J^r) = \frac{0.25H_0}{\epsilon_0} \int_{\Gamma} -\sin(a)(k_0(\cos(\theta)y - \sin(\theta)x))F_1 + \cos(a)D_\theta F_1 \\ - \cos(a)(k_0(\cos(\theta)y - \sin(\theta)x))F_2 - \sin(a)D_\theta F_2 \, ds \quad (6.13)$$

$$D_\theta(J^i) = \frac{0.25H_0}{\epsilon_0} \int_{\Gamma} \cos(a)(k_0(\cos(\theta)y - \sin(\theta)x))F_1 + \sin(a)D_\theta F_1 \\ - \sin(a)(k_0(\cos(\theta)y - \sin(\theta)x))F_2 + \cos(a)D_\theta F_2 \, ds \quad (6.14)$$

After obtaining the derivative of the plane wave parameter θ , the final part of the derivation of the gradient $D_{\mathbf{v}}P$ is discussed. The geometric design parameters $\mathbf{v} = d_1, \dots, d_n =$ displacement of the right, left and bottom boundaries of $n/3$ grating grooves. As with the plane wave θ parameter, the general gradient of $P(J) = (J^r)^2 + (J^i)^2$ is:

$$D_{\mathbf{v}}P = 2J^r(D_{\mathbf{v}}J^r) + 2J^i(D_{\mathbf{v}}J^i) = 2J^r \left(\frac{\partial J^r}{\partial \mathbf{v}} \right) + 2J^i \left(\frac{\partial J^i}{\partial \mathbf{v}} \right). \quad (6.15)$$

The most significant part of the derivation is to obtain the formulas for the gradients $D_{\mathbf{v}}J^r, D_{\mathbf{v}}J^i$.

$$D_{\mathbf{v}}(J^r) = \frac{0.25H_0}{\epsilon_0} \int_{\Gamma} \cos(a) \left(c_1 \frac{\partial H^r}{\partial \mathbf{v}} - c_2 \frac{\partial^2 H^i}{\partial y \partial \mathbf{v}} \right) \\ - \sin(a) \left(c_1 \frac{\partial H^i}{\partial \mathbf{v}} + c_2 \frac{\partial^2 H^r}{\partial y \partial \mathbf{v}} \right) \, ds \quad (6.16)$$

$$D_{\mathbf{v}}(J^i) = \frac{0.25H_0}{\epsilon_0} \int_{\Gamma} \cos(a) \left(c_1 \frac{\partial H^i}{\partial \mathbf{v}} + c_2 \frac{\partial^2 H^r}{\partial y \partial \mathbf{v}} \right) \\ + \sin(a) \left(c_1 \frac{\partial H^r}{\partial \mathbf{v}} - c_2 \frac{\partial^2 H^i}{\partial y \partial \mathbf{v}} \right) \, ds. \quad (6.17)$$

Here

- $c_1 = \frac{\sin(\theta)}{c_0}$
- $c_2 = \omega^{-1}$
- $a = k(\cos(\theta)x + \sin(\theta)y)$

The numerical approximation of the second derivative $\frac{\partial^2 H}{\partial y \partial \mathbf{v}}$ is obtained from the 8th order accurate central finite difference defined as:

$$\frac{\partial f}{\partial y}|_{y=y_0} \approx \frac{1}{h} \sum_{k=1}^4 c_k (f(y_k) - f(y_{-k}))$$

$$y_{-k} = y_0 - kh,$$

$$y_k = y_0 + kh, \tag{6.18}$$

$$y_0 = 8000 \text{ nm}$$

$$c_k = \{4/5, -1/5, 4/105, -1/280\}, k = 1, \dots, 4$$

where $f(y_k) = \frac{\partial H(x, y_k)}{\partial \mathbf{v}}$ for a fixed x , and $h = dy$. This is better than the 2nd degree central finite difference used in previous chapters to compute the λ and ϵ gradients of the light absorption function. It uses 8 sensitivity values to approximate the y -space derivative corresponding to the sensitivity of the electric field to the design parameters v . Nonetheless results show that the choice of the step h is critical as the gradient values vary significantly. This is further discussed in the numerical gradients section.

The finite difference approximation of the geometric gradient is the weak link in obtaining the gradient. Therefore the next section presents some theoretical results on the implementation of an Adjoint Method to find the gradient of this objective function.

6.3.2 Adjoint Method

The Adjoint Method is a very efficient way to obtain the gradient of an objective function with respect to the geometric design variables. This direct computation saves the intermediate step of computing the sensitivity of the variables with respect to the state variables, a number of the order of the mesh. In this dissertation then it is natural to analyze the possibility of using the adjoint method to find gradients of the coupler objective function. Because (6.11) contains terms $\frac{\partial H}{\partial y} = D_y H$, the adjoint method would require the solution of two systems of equations: $KH = b$, and, by linearization with respect the space derivative y , $K \frac{\partial H}{\partial y} = \frac{\partial b}{\partial y} - [D_y K]H$. When considering the derivatives with respect to a design parameter v this last equation would produce undesirable double derivative on the stiffness matrix $D_v(D_y K)$. A simpler formulation of J in terms of the x component of the electric field: E_x would be more useful, as follows:

$$J = \frac{H_0}{4} \int_{\Gamma} (e^{\bar{A}}) \left(\frac{\sin(\theta)}{c_0 \epsilon_0} H_z - E_x \right) ds \quad (6.19)$$

$$(e^{\bar{A}}) = \cos(a) + j \sin(a), \quad a = k \cos(\theta)x + k \sin(\theta)y - \omega t. \quad (6.20)$$

Again the real and imaginary components become:

$$J^r = \frac{H_0}{4\epsilon_0} \int_{\Gamma} \cos(a) \left(\frac{\sin(\theta)}{c_0 \epsilon_0} H^r - E_x^r \right) - \sin(a) \left(\frac{\sin(\theta)}{c_0 \epsilon_0} H^i - E_x^i \right) ds \quad (6.21)$$

$$J^i = \frac{H_0}{4\epsilon_0} \int_{\Gamma} \cos(a) \left(\frac{\sin(\theta)}{c_0 \epsilon_0} H^i - E_x^i \right) + \sin(a) \left(\frac{\sin(\theta)}{c_0 \epsilon_0} H^r - E_x^r \right) ds. \quad (6.22)$$

Using the Adjoint Method the objective function to be maximized after introducing the adjoint variables p, q is:

$$P = (J^r)^2 + (J^i)^2 + p^T(b_1 - K_1 H) + q^T(b_2 - K_2 E_x) \quad (6.23)$$

where the discretization of the PDE for magnetic and electric field correspond to $K_1 H = b_1$, $K_2 E_x = b_2$. The gradient of $D_{\mathbf{v}} P$ and its derivative with respect to a list of parameters $vec v$ is the main derivation in the following proposition.

Proposition 3. Let P be the objective function that is not directly dependent on the geometric parameter v . No design geometric parameter is in the boundary of the incident wave, and the adjoint variables p are also constant in v . Then the gradient of the objective function is given by

$$D_{\mathbf{v}} P = -p^T [D_{\mathbf{v}}(K_1)] H - q^T [D_{\mathbf{v}}(K_2)] E_x, \quad (6.24)$$

where the complex valued quantities $p = p^r + jp^i, q = q^r + jq^i, H = H^r + jH^i, E_x =$

$$E_x^r + jE_x^i, \text{ and complex valued matrices with } [M] = M^r + jM^i = \begin{bmatrix} M^r & -M^i \\ M^i & M^r \end{bmatrix} \text{ satisfy the}$$

following auxiliary adjoint equations:

$$p^T K_1 = \left[2(J) \frac{\partial J}{\partial H} \right] \quad (6.25)$$

$$q^T K_2 = \left[2(J) \frac{\partial J}{\partial E_x} \right]. \quad (6.26)$$

Proof. By a generalized chain rule, and the assumption that P does not depend directly on v :

$$D_{\mathbf{v}} P = D_{\mathbf{v}}((J^r)^2) + D_{\mathbf{v}}((J^i)^2) + p^T [-D_{\mathbf{v}}(K_1)H - K_1 D_{\mathbf{v}}(H)] + q^T [-D_{\mathbf{v}}(K_2)E_x - K_2 D_{\mathbf{v}}(E_x)].$$

Decomposing into real and imaginary parts:

$$D_{\mathbf{v}}((J^r)^2) = 2(J^r) \left[\frac{\partial J^r}{\partial H^r} \frac{\partial H^r}{\partial \mathbf{v}} + \frac{\partial J^r}{\partial E_x^r} \frac{\partial E_x^r}{\partial \mathbf{v}} + \frac{\partial J^r}{\partial H^i} \frac{\partial H^i}{\partial \mathbf{v}} + \frac{\partial J^r}{\partial E_x^i} \frac{\partial E_x^i}{\partial \mathbf{v}} \right].$$

Similarly

$$D_{\mathbf{v}}((J^i)^2) = 2(J^i) \left[\frac{\partial J^i}{\partial H^r} \frac{\partial H^r}{\partial \mathbf{v}} + \frac{\partial J^i}{\partial E_x^r} \frac{\partial E_x^r}{\partial \mathbf{v}} + \frac{\partial J^i}{\partial H^i} \frac{\partial H^i}{\partial \mathbf{v}} + \frac{\partial J^i}{\partial E_x^i} \frac{\partial E_x^i}{\partial \mathbf{v}} \right].$$

Then $p^T[-D_{\mathbf{v}}(K_1)H - K_1 D_{\mathbf{v}}(H)] =$

$$-[p^r, p^i] \begin{bmatrix} D_{\mathbf{v}}(K_1^r) & -D_{\mathbf{v}}(K_1^i) \\ D_{\mathbf{v}}(K_1^i) & D_{\mathbf{v}}(K_1^r) \end{bmatrix} \begin{bmatrix} H^r \\ H^i \end{bmatrix} - [p^r, p^i] \begin{bmatrix} K_1^r & -K_1^i \\ K_1^i & K_1^r \end{bmatrix} \begin{bmatrix} \frac{\partial H^r}{\partial \mathbf{v}} \\ \frac{\partial H^i}{\partial \mathbf{v}} \end{bmatrix}. \quad (6.27)$$

The same holds for q , taking K_2, E_x instead of K_1, H . Regrouping these scalar quantities:

$$\begin{aligned} D_{\mathbf{v}}P &= \left(2(J^r) \frac{\partial J^r}{\partial H^r} + 2(J^i) \frac{\partial J^i}{\partial H^r} - [p^r, p^i] \begin{bmatrix} K_1^r \\ K_1^i \end{bmatrix} \right) \frac{\partial H^r}{\partial \mathbf{v}} \\ &+ \left(2(J^r) \frac{\partial J^r}{\partial E_x^r} + 2(J^i) \frac{\partial J^i}{\partial E_x^r} - [q^r, q^i] \begin{bmatrix} K_2^r \\ K_2^i \end{bmatrix} \right) \frac{\partial E_x^r}{\partial \mathbf{v}} \\ &+ \left(2(J^r) \frac{\partial J^r}{\partial H^i} + 2(J^i) \frac{\partial J^i}{\partial H^i} - [p^r, p^i] \begin{bmatrix} -K_1^i \\ K_1^r \end{bmatrix} \right) \frac{\partial H^i}{\partial \mathbf{v}} \\ &+ \left(2(J^r) \frac{\partial J^r}{\partial E_x^i} + 2(J^i) \frac{\partial J^i}{\partial E_x^i} - [p^r, p^i] \begin{bmatrix} -K_2^i \\ K_2^r \end{bmatrix} \right) \frac{\partial E_x^i}{\partial \mathbf{v}} \\ &- [p^r, p^i] \begin{bmatrix} D_{\mathbf{v}}(K_1^r) & -D_{\mathbf{v}}(K_1^i) \\ D_{\mathbf{v}}(K_1^i) & D_{\mathbf{v}}(K_1^r) \end{bmatrix} \begin{bmatrix} H^r \\ H^i \end{bmatrix} - [q^r, q^i] \begin{bmatrix} D_{\mathbf{v}}(K_2^r) & -D_{\mathbf{v}}(K_2^i) \\ D_{\mathbf{v}}(K_2^i) & D_{\mathbf{v}}(K_2^r) \end{bmatrix} \begin{bmatrix} E_x^r \\ E_x^i \end{bmatrix}. \end{aligned} \quad (6.28)$$

To cancel the $\frac{\partial H}{\partial \mathbf{v}}, \frac{\partial E_x}{\partial \mathbf{v}}$ terms take:

$$\begin{aligned} \begin{bmatrix} 2(J^r) \frac{\partial J^r}{\partial H^r} + 2(J^i) \frac{\partial J^i}{\partial H^r} \\ 2(J^r) \frac{\partial J^r}{\partial H^i} + 2(J^i) \frac{\partial J^i}{\partial H^i} \end{bmatrix} &= [p_r, p_i] \begin{bmatrix} K_1^r & -K_1^i \\ K_1^i & K_1^r \end{bmatrix}, \\ \begin{bmatrix} 2(J^r) \frac{\partial J^r}{\partial E_x^r} + 2(J^i) \frac{\partial J^i}{\partial E_x^r} \\ 2(J^r) \frac{\partial J^r}{\partial E_x^i} + 2(J^i) \frac{\partial J^i}{\partial E_x^i} \end{bmatrix} &= [q_r, q_i] \begin{bmatrix} K_2^r & -K_2^i \\ K_2^i & K_2^r \end{bmatrix}, \end{aligned} \tag{6.29}$$

equivalent to the auxiliary equations in (6.26). \square

The proposition shows that a form of the Adjoint Method requires two solutions. The solutions for the auxiliary equations in p, q : magnetic and electric fields adjoint variables are required in the gradient. Although further theoretical development might be a worthy endeavor, at this point the proposition is not favorable for computational implementation. The approach developed in previous chapters of this dissertation uses one solution, the scalar magnetic field, and a numerical approximation of the the electric field. This is faster computationally and a natural choice for implementation, pending further theoretical development. Therefore the computer implementation applies gradients obtained using this hybrid numerical and analytical approach. The forward sensitivity or direct differentiation is used to obtain the gradient of the magnetic field, and an 8th degree accurate numerical central finite difference is used for the electric field gradient.

In the next section the formulation of the objective function and its gradients are obtained for the Gaussian beam.

6.4 The Gaussian Beam

The natural next step to improve the performance of the design is to substitute the plane wave by a Gaussian beam. The Gaussian beam has curved wavefronts in contrast with the

plane wave. Visualization of the magnetic field of the initial design show a curved magnetic field that suggests that a Gaussian beam can produce better conversion. The parameters governing the Gaussian beam allow more control on the shape and position of the beam parameters that can help tailor the wave to conform to the characteristics of the generated plasmonic beam. Moreover, the Gaussian beam can be generated by an optical fiber, which make it readily available from the practical point of view.

The Gaussian beam propagates according to a propagation axis u , as the plane wave, and radially v away from the propagation axis, see figure 6.7¹. At $u=0$ the beam has a center (x_0, y_0) , where the beam has a maximum intensity or amplitude of the electric field. As the distance from the center increases the intensity or amplitude of the field drops in a the form of a Gaussian curve form. The radial propagation function $w(u)$ has its smallest value at $w(0) = w_0$ the waist. Therefore the Gaussian beam can be characterized by these parameters: center, waist and angle of propagation axis.

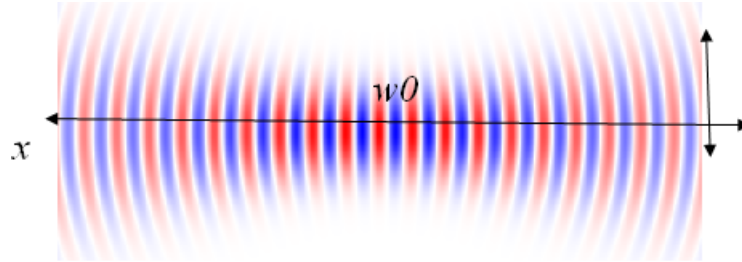


Figure 6.7: Gaussian Beam axis and radial propagation

practitioners use the Gaussian beam to model light from optical fiber, and therefore some restrictions are physically imposed on the dissertation model. A typical width of the optical fiber is around $w = 130\text{e-}6$ m. Therefore the center of the Gaussian beam cannot be placed closer to the grooves that some minimal possible distance, unless the optical fiber is placed perpendicular to the surface with $\theta = \pi/2$ (see Figure 6.8). Therefore the parameter *dist* is

¹http://www.rp-photonics.com/Gaussian_beams.html

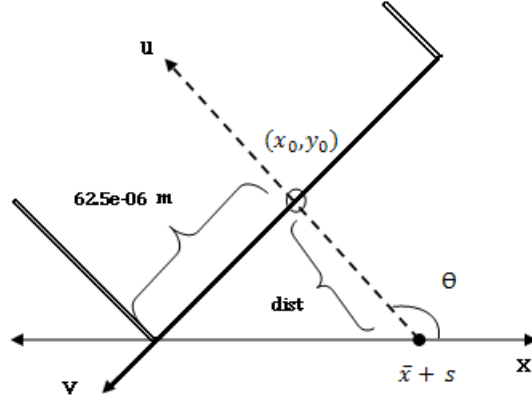


Figure 6.8: Gaussian Beam Parameters

constrained with the inequality: $dist \geq 0.5 * w * \cot(\theta)$, where w is the width of the optical fiber. The point where the optical fiber touches the silver slab can also change allowing horizontal movement of the optical fiber. The distance from the boundary to the grating is denoted as \bar{x} . Then, the distance from the point where the optical fiber touches the silver slab to \bar{x} is the parameter s . Finally the design allows change in the angle of the optical fiber and therefore the Gaussian beam. In figure 6.8 the parameters of the Gaussian beam are illustrated.

6.4.1 Objective Function and Gradients

The Gaussian Beam formulation in the 2D space (x, y) is given by the formula in the scalar magnetic field and transformed axis (u, v) :

$$H_G(x, y) = H_0 \sqrt{\frac{w_0}{w}} \exp\left(-\frac{v^2}{w^2}\right) \exp\left(-jk\left(u + \frac{v^2}{2R}\right) + \frac{j}{2} \arctan\left(\frac{u}{x_r}\right)\right), \quad (6.30)$$

with

- axis of propagation $u = (x - x_0) \cos(\theta) + (y - y_0) \sin(\theta)$
- radial distance from propagation axis $v = -(x - x_0) \sin(\theta) + (y - y_0) \cos(\theta)$

- beam center (x_0, y_0)
- beam radius along the u propagation axis $w = w(u) = w_0 \sqrt{1 + \left(\frac{u}{x_r}\right)^2}$
- size of beam waist $w_0 = w(0)$
- radius of curvature of the beam's wavefront $R = R(u) = u \left(1 + \left(\frac{x_r}{u}\right)^2\right)$
- Rayleigh length $x_r = \frac{\pi w_0^2}{\lambda}$, propagation length of beam without significant divergence
- wave number $k = \frac{2\pi}{\lambda}$
- magnitude of incident magnetic field H_0

In this dissertation the center is also a variable dependent on some natural design parameters: the Gaussian Beam rotation angle θ , the displacement s in the x -axis from the position of the first groove \bar{x} , and the distance $dist$ from the x -axis to the center along the Gaussian Beam's u -axis. The center coordinates are now defined by:

$$\begin{aligned} x_0 &= (\bar{x} + s) + dist * \cos(\theta) \\ y_0 &= dist * \sin(\theta) \end{aligned} \tag{6.31}$$

The x , component of the electric field $E_G = -\frac{j}{\omega\epsilon} \frac{\partial H_G}{\partial y}$, is obtained to derive the objective functional $J = -\frac{1}{4} \int_{\Gamma} \bar{E}_G H + E_x \bar{H}_G d\sigma$, with \bar{z} the complex conjugate, E_x, H the computed x -component of the electric and magnetic field respectively. In order to simplify the necessary derivations note that (6.30) is of the form:

$$H_G = H_0 f \exp(g) \exp(jh) = H_0 f \exp(g) (\cos(h) + j \sin(h)),$$

$$\bar{H}_G = H_0 f \exp(g) (\cos(h) - j \sin(h)), \text{ and}$$

- $f = f(w(u)) = \sqrt{w_0 w(u)^{-1}}$
- $g = g(v, w(u)) = -v^2 w(u)^{-2}$

- $h = h(u, v, R(u)) = -k(u + 0.5v^2R(u)^{-1}) + 0.5 \arctan\left(\frac{u}{x_r}\right).$

Using this notation the real and imaginary components are simply:

$$real(H_G) = H_G^r = H_0 f \exp(g) \cos(h),$$

$$imag(H_G) = H_G^i = H_0 f \exp(g) \sin(h).$$

A general expression for the electric field of the Gaussian Beam is given in the following main result:

$$E_G = \frac{-jH_0}{\omega\epsilon} [\exp(g)(f' + fg')(\cos(h) + j \sin(h)) + \exp(g)(fh')(-\sin(h) + j \cos(h))] \quad (6.32)$$

$$real(E_G) = \frac{H_0}{\omega\epsilon} \exp(g) [\sin(h)(f' + fg') + \cos(h)fh'] \quad (6.33)$$

$$imag(E_G) = \frac{H_0}{\omega\epsilon} \exp(g) [\sin(h)(fh') - \cos(h)(f' + fg')], \quad (6.34)$$

with derivatives with respect to y :

$$f' = \frac{\partial f}{\partial y} = -0.5w' \sqrt{w_0 w^{-3}}$$

$$w' = \frac{\partial w}{\partial y} = w_0(1 + (u/x_r)^2)^{-1/2}(uu'/x_r^2)$$

$$g' = \frac{\partial g}{\partial y} = -2vv'w^{-2} + 2v^2w^{-3}w'$$

$$h' = \frac{\partial h}{\partial y} = -k(u' + vv'R^{-1} - 0.5v^2R^{-2}R') + 0.5 \frac{u'x_r}{x_r^2 + u^2}$$

$$R' = \frac{\partial R}{\partial y} = u' (1 - u^{-2}x_r^2)$$

$$u' = \frac{\partial u}{\partial y} = \sin(\theta)$$

$$v' = \frac{\partial v}{\partial y} = \cos(\theta).$$

The normalization P_N is obtained easily by noticing that the coupling of the Gaussian beam with itself in the equation (6.1) for complex $E_G = (a + bj)$, $H_G = (c + dj)$ has the form:

$$\begin{aligned} P_N &= -0.25 \int_{\Gamma} (a - bj)(c + dj) + (a + bj)(c - dj) \, ds \\ &= -0.25 \int_{\Gamma} 2(ac + bd) + (ad - bc)j + (bc - ad)j \, ds \\ &= -0.5 \int_{\Gamma} (ac + bd) \, ds \\ &= -0.5 \int_{\Gamma} (E_G^r)(H_G^r) + (E_G^i)(H_G^i) \, ds. \end{aligned} \tag{6.35}$$

To obtain the gradient of the objective function for the geometric design parameters \mathbf{v} , the real and imaginary components of J are defined first. Thus

$$\begin{aligned} \text{real}(J) = J^r = & -0.25 \int_{\Gamma} \text{real}(E_G) \text{real}(H) + \text{imag}(E_G) \text{imag}(H) \\ & + \text{real}(E_x) \text{real}(H_G) + \text{imag}(E_x) \text{imag}(H_G) \, ds \end{aligned} \quad (6.36)$$

$$\begin{aligned} \text{imag}(J) = J^i = & -0.25 \int_{\Gamma} \text{real}(E_G) \text{imag}(H) - \text{imag}(E_G) \text{real}(H) \\ & - \text{real}(E_x) \text{imag}(H_G) + \text{imag}(E_x) \text{real}(H_G) \, ds. \end{aligned} \quad (6.37)$$

Because the Gaussian Beam is not dependent on the geometric design parameters $\mathbf{v} = d_1, \dots, d_n$ the displacement of the left, right, bottom boundaries of the grating grooves:

$$\begin{aligned} \frac{\partial J^r}{\partial \mathbf{v}} = & -0.25 \int_{\Gamma} E_G^r \frac{\partial H^r}{\partial \mathbf{v}} + E_G^i \frac{\partial H^i}{\partial \mathbf{v}} \\ & + \frac{\partial E_x^r}{\partial \mathbf{v}} H_G^r + \frac{\partial E_x^i}{\partial \mathbf{v}} H_G^i \, ds \end{aligned} \quad (6.38)$$

$$\begin{aligned} \frac{\partial J^i}{\partial \mathbf{v}} = & -0.25 \int_{\Gamma} E_G^r \frac{\partial H^i}{\partial \mathbf{v}} - E_G^i \frac{\partial H^r}{\partial \mathbf{v}} \\ & - \frac{\partial E_x^r}{\partial \mathbf{v}} H_G^i + \frac{\partial E_x^i}{\partial \mathbf{v}} H_G^r \, ds. \end{aligned} \quad (6.39)$$

The sensitivity of the state variables $\frac{\partial H^r}{\partial \mathbf{v}}, \frac{\partial H^i}{\partial \mathbf{v}}$ to the geometric parameters \mathbf{v} is obtained from COMSOL Multiphysics Sensitivity Analysis applications. Nonetheless the sensitivity to the electric field is computed using the 8th degree accurate central finite difference approximation in (6.18): $\frac{\partial E_x}{\partial \mathbf{v}} = - \left(\frac{j}{\omega \epsilon} \right) \frac{\partial^2 H}{\partial y \partial \mathbf{v}} = \delta_y \left(\frac{\partial H}{\partial \mathbf{v}} \right)$.

Next the gradient of J^r, J^i with respect to the Gaussian Beam parameters: $\alpha = \{\theta, \text{dist}, s\}$

is derived aided by the fact that the state variables are independent of α :

$$\begin{aligned} \frac{\partial J^r}{\partial \alpha} &= -0.25 \int_{\Gamma} \frac{\partial E_G^r}{\partial \alpha} H^r + \frac{\partial E_G^i}{\partial \alpha} H^i \\ &\quad + E_x^r \frac{\partial H_G^r}{\partial \alpha} + E_x^i \frac{\partial H_G^i}{\partial \alpha} ds \end{aligned} \quad (6.40)$$

$$\begin{aligned} \frac{\partial J^i}{\partial \alpha} &= -0.25 \int_{\Gamma} \frac{\partial E_G^r}{\partial \alpha} H^i - \frac{\partial E_G^i}{\partial \alpha} H^r \\ &\quad - E_x^r \frac{\partial H_G^i}{\partial \alpha} + E_x^i \frac{\partial H_G^r}{\partial \alpha} ds. \end{aligned} \quad (6.41)$$

Therefore a general gradient form for Gaussian Beam design parameters is:

$$\frac{\partial H_G^r}{\partial \alpha} = H_0 \exp(g)(\cos(h)[f_{\alpha} + fg_{\alpha}] - \sin(h)[fh_{\alpha}]) \quad (6.42)$$

$$\frac{\partial H_G^i}{\partial \alpha} = H_0 \exp(g)(\sin(h)[f_{\alpha} + fg_{\alpha}] + \cos(h)[fh_{\alpha}]) \quad (6.43)$$

$$\begin{aligned} \frac{\partial E_G^r}{\partial \alpha} &= \frac{H_0}{\omega \epsilon} \exp(g)(\sin(h)[g_{\alpha}(f' + fg') + f'_{\alpha} + f_{\alpha}g' + fg'_{\alpha} - h_{\alpha}fh'] \\ &\quad + \cos(h)[g_{\alpha}fh' + h_{\alpha}(f' + fg') + f_{\alpha}h' + fh'_{\alpha}]) \end{aligned} \quad (6.44)$$

$$\begin{aligned} \frac{\partial E_G^i}{\partial \alpha} &= \frac{H_0}{\omega \epsilon} \exp(g)(\sin(h)[g_{\alpha}fh' + h_{\alpha}(f' + fg') + f_{\alpha}h' + fh'_{\alpha}] \\ &\quad + \cos(h)[h_{\alpha}fh' - g_{\alpha}(f' + fg') - f'_{\alpha} - f_{\alpha}g' - fg'_{\alpha}]) \end{aligned} \quad (6.45)$$

Important derivations to compute $\frac{\partial E_G}{\partial \alpha}$, $\frac{\partial H_G}{\partial \alpha}$ in (6.42)-(6.45), with $\alpha = \theta$ follow.

- $\frac{\partial u}{\partial \theta} = u_{\theta} = -(x - x_0) \sin(\theta) + (y - y_0) \cos(\theta)$
- $\frac{\partial v}{\partial \theta} = v_{\theta} = -(x - x_0) \cos(\theta) - (y - y_0) \sin(\theta) - dist$
- $\frac{\partial u}{\partial dist} = u_{dist} = -\cos^2(\theta) - \sin^2(\theta) = -1$
- $\frac{\partial v}{\partial dist} = v_{dist} = \sin(\theta) \cos(\theta) - \cos(\theta) \sin(\theta) = 0$

- $\frac{\partial u}{\partial s} = u_s = -\cos(\theta)$
- $\frac{\partial v}{\partial s} = v_s = \sin(\theta)$
- $\frac{\partial^2 u}{\partial y \partial \theta} = u'_\theta = \cos(\theta)$
- $\frac{\partial^2 v}{\partial y \partial \theta} = v'_\theta = -\sin(\theta)$
- $\frac{\partial^2 u}{\partial y \partial dist} = u'_{dist} = 0$
- $\frac{\partial^2 v}{\partial y \partial dist} = v'_{dist} = 0$
- $\frac{\partial w}{\partial \alpha} = w_\alpha = \frac{w_0 u u_\alpha}{x_r^2 \sqrt{1+(u/x_r)^2}}$
- $\frac{\partial f}{\partial \alpha} = f_\alpha = -0.5 w_\alpha \sqrt{w_0 w^{-3}}$
- $\frac{\partial g}{\partial \alpha} = g_\alpha = 2 v w^{-2} (v w_\alpha / w - v_\alpha)$
- $\frac{\partial R}{\partial \alpha} = R_\alpha = u_\alpha (1 - (x_r^2 / u^2))$
- $\frac{\partial h}{\partial \alpha} = h_\alpha = -k(u_\alpha + v v_\alpha R^{-1} - 0.5 v^2 R^{-2} R_\alpha) + 0.5 \frac{u_\alpha x_r}{x_r^2 + u^2}$
- $\frac{\partial^2 f}{\partial y \partial \alpha} = f'_\alpha = -0.5 w'_\alpha \sqrt{w_0 w^{-3}} + 0.75 w' w_\alpha \sqrt{w_0 w^{-5}}$
- $\frac{\partial^2 w}{\partial y \partial \alpha} = w'_\alpha = -\frac{w_0 u^2 u' u_\alpha}{x_r^4 (1+(u/x_r)^2)^{3/2}} + \frac{w_0 (u'_\alpha u + u' u_\alpha)}{x_r^2 (1+(u/x_r)^2)^{1/2}}$
- $\frac{\partial^2 g}{\partial y \partial \alpha} = g'_\alpha = 2 w^{-2} [v_\alpha (2 v w' / w - v') + w_\alpha (2 v v' / w - 3 v^2 w' / w^2) + v (v w'_\alpha / w - v'_\alpha)]$
- $\frac{\partial^2 h}{\partial y \partial \alpha} = h'_\alpha = -k[u'_\alpha + v R^{-2} (-v_\alpha R' + R_\alpha (v R' / R - v') - 0.5 v R'_\alpha) + (v' v_\alpha + v v'_\alpha) / R]$
 $+ \frac{x_r}{x_r^2 + u^2} [0.5 u'_\alpha - \frac{u u' u_\alpha}{x_r^2 + u^2}]$
- $\frac{\partial^2 R}{\partial y \partial \alpha} = R'_\alpha = u'_\alpha (1 - x_r^2 / u^2) + 2 x_r^2 u' u_\alpha / u^3.$

In the spirit of sharing the acquired knowledge with other interested researchers the next section includes the development of the computational model from the initial models to the final simplified form. The challenges encountered guided the development of a useful models that together with the derived formulas of gradients and objective functions, guided the discovery of very improved designs. The computational implementation results and tools developed using COMSOL-multiphysics and MATLAB are discussed next.

6.5 Development of Computational Tools

The dissertation work has produced useful computational tools for the optimal design of devices that tackle a 14 dimensional optimization problem. A gradient based update of 11 geometrical and 3 analytical parameters produced much improved designs for the plane wave and Gaussian beam nanoplasmonic systems. These tools have produced new and greatly improved designs of the grating coupler, and therefore have proven useful as a first or beta version. And, there is much more designs yet to be discovered and tested as the tools can be used for a larger number of grooves.

The developed tools are useful to conduct numerical optimization in different ways. The developed numerical optimization techniques tools for the 4 groove grating include:

1. Numerical Optimization of 11 geometric parameters with fixed beam parameters.
2. Optimization of grating design of 3 beam parameters with fixed geometry.
3. Simultaneous optimization of geometric and beam parameters.
4. Simultaneous optimization of geometric and beam parameters with quality of gradients verification.

These programs use efficiently available data and solvers. The final Gaussian beam design was obtained in approximately one hour in a Dell Precision T7400 desktop computer (Intel (R) Xeon(R) CPU, X5482@3.2 GHz, 3.19GHz, 32.0GB RAM) by using a combination of these tools to find a better initial design and best Gaussian beam parameters.

The grating coupler is a complex problem to model and optimize. It requires specialized boundary conditions developed for use in computational electromagnetism, geometry and refined mesh generators, as well as multiphysics capabilities. The complex form of the objective function and gradient also require a combination of sensitivity analysis and numerical techniques to update the designs. Therefore to develop the numerical optimization techniques for the grating coupler, the experiments are conducted using the commercial engineering, simulation and FE analysis software environment COMSOL Multiphysics, version 3.5 and MATLAB. COMSOL provides a MATLAB based script language since its origins as FEMLAB, a distributed MATLAB package.

This hybrid computer tools presented in this dissertation accomplish several tasks to optimize the grating coupler design. Through COMSOL's tools the geometrical domains are defined as silver and air elements according to the corresponding permittivity. The parametrized geometry of the Arbitrary Lagrangian Eulerian (ALE) application is used in combination with the forward sensitivity analysis. These together with the electromagnetic application makes use of COMSOL's multiphysics capabilities. The geometry, boundary conditions, visualization, solvers, refined unstructured mesh, and interpolation tools of COMSOL are exploited. The MATLAB code conducts post processing of COMSOL's solutions to compute the numerical gradient using the formulas obtained in the previous sections, control the optimization algorithm, and do necessary updates to COMSOL's domain and boundary lists to obtain the next design update. The computational results and details of the implementation of the MATLAB and COMSOL are presented next.

6.5.1 Computer Implementation

One of the main products of this project are the codes developed by combining COMSOL and MATLAB tools. The computational implementation and modeling decisions presented here are also of interest for its potential use in other optimal design problems. The main algorithms, strategies for updating the geometry, and some COMSOL Multiphysics environment application, solvers, drawing and mesh generation are discussed. Visualization of

the results are also included.

The computer implementation of the numerical optimization technique is a composite algorithm that does the following tasks:

1. Initializations.
 - (a) Build the geometry from COMSOL's tools.
 - (b) Build the refined unstructured mesh using COMSOL's mesher.
 - (c) Define in COMSOL the three multiphysics application modes.
 - i. Electromagnetism transverse magnetic (TM)-mode.
 - ii. parametrized geometric (pg)-mode.
 - iii. Sensitivity analysis (sa)-mode.
 - (d) Solve using COMSOL's Linear static solver for symmetric matrices UMFPACK.
2. Find gradient with MATLAB and COMSOL tools.
 - (a) Interpolate the sensitivity values for each of the 11 parameters in the objective function grid.
 - (b) Compute 8th accurate central finite difference of the sensitivity in the grid.
 - (c) Compute the gradient using the derived formulas for geometry parameter derivatives.
3. For each groove update the domains and boundaries of the new geometry according to the gradient information.
4. Obtain current values of the derivatives of the output beam parameters using the derived formulas (6.42)-(6.45) for Gaussian beam, and (6.13) for the plane wave.
5. Update the values of the beam parameters into the FEM structure: commands `<multiphysics>` and `<fem.sol = fem0.sol>`.
6. Solve and repeat until objective function cannot improve.

The update of the domains and boundaries of the design requires and modifying two COMSOL lists: domain list in the TM application mode and the boundary list in the pg application mode. In order to update each groove's size there are 27 cases on the movement of the boundaries and domains that the algorithm needs to be able to update. The 27 different ways to modify a grating are illustrated in figure 6.9. The following algorithm works for all 27 cases.

1. For right boundary displacement (r).
 - (a) Shift to right or none ($dxR=\{0,1\}$).
 - i. Update with the final right domain position (R).
 - ii. Change silver domains to air from R to level 0 (top of silver slab).
 - iii. Update boundaries to the right of (R) domain.
 - (b) Shift to left ($dxR= -1$).
 - i. Update domains from the vertically shifted right domain (Rv).
 - ii. Change air domains to silver from Rv to level 0.
 - iii. Update boundaries to the left of Rv .
2. For left boundary displacement (l).
 - (a) Shift to right ($dxL= 1$).
 - i. Update domains from the vertically shifted left domain (Lv).
 - ii. Change air domains to silver from Lv to level 0.
 - iii. Update boundaries to the right of Lv .
 - (b) Shift to left or none ($dxL = \{-1,0\}$).
 - i. Update domains with the final left domain position (L).
 - ii. Change silver domains to air from L to level 0.
 - iii. Update boundaries to the left of the domain L .
3. For bottom boundary displacements (dyB).

- (a) Shift down ($dyB = -1$).
 - i. Change silver domains to air from L to R.
 - ii. Update bottom boundaries from L to R.
 - (b) Shift up or none ($dyB = \{0,1\}$).
 - i. Change air domains to silver from horizontal shifts of L and R (Lh, Rh).
 - ii. Update upper boundaries from Lh to Rh .
4. Special treatment of corner domains.
- (a) $dxR = -1, dxL = \{-1,0\}$: change previous right domain r to silver.
 - (b) $dxL = 1$: change previous left domain l to silver.
 - (c) $dxR = -1, dxL = 1$: change previous domains l, r to silver.

The first step to generate a COMSOL model is to define the geometry. After using the GUI to draw, the commands are saved in a MATLAB/COMSOL file in order to build the code and generalize the geometry commands. Caution must be exercised when selecting the objects type. The initial geometry had to be redrawn using solid objects in order to make a composite geometry consisting of gratings, silver, PML, objective function boundary and computational domain. Curves cannot become part of a composite object that contains solids and should be avoided. For this reason the objective function boundary is the lower part of a rectangle at $y = 8000$ nm, that is 2000 nm from the top PML. This remedy provided a way to identify the expressions for the objective function as a <boundary expression>. That is, the gradient formulas and objective function variables are defined as <inline> expressions at this boundary. The computational domain contains a silver rectangle at $y = [-2.0e-7, 0]$ nm, so that the plasmon equation is defined as in (6.3). Apart from this, the objects are defined by coordinates in a straight forward fashion. The real challenge in order to have a general drawing code is in the definition of the geometrical variables and numbering both domains and boundaries for COMSOL application modes. This can be done in a text handling utility.

(dxR, dxL, dyB)

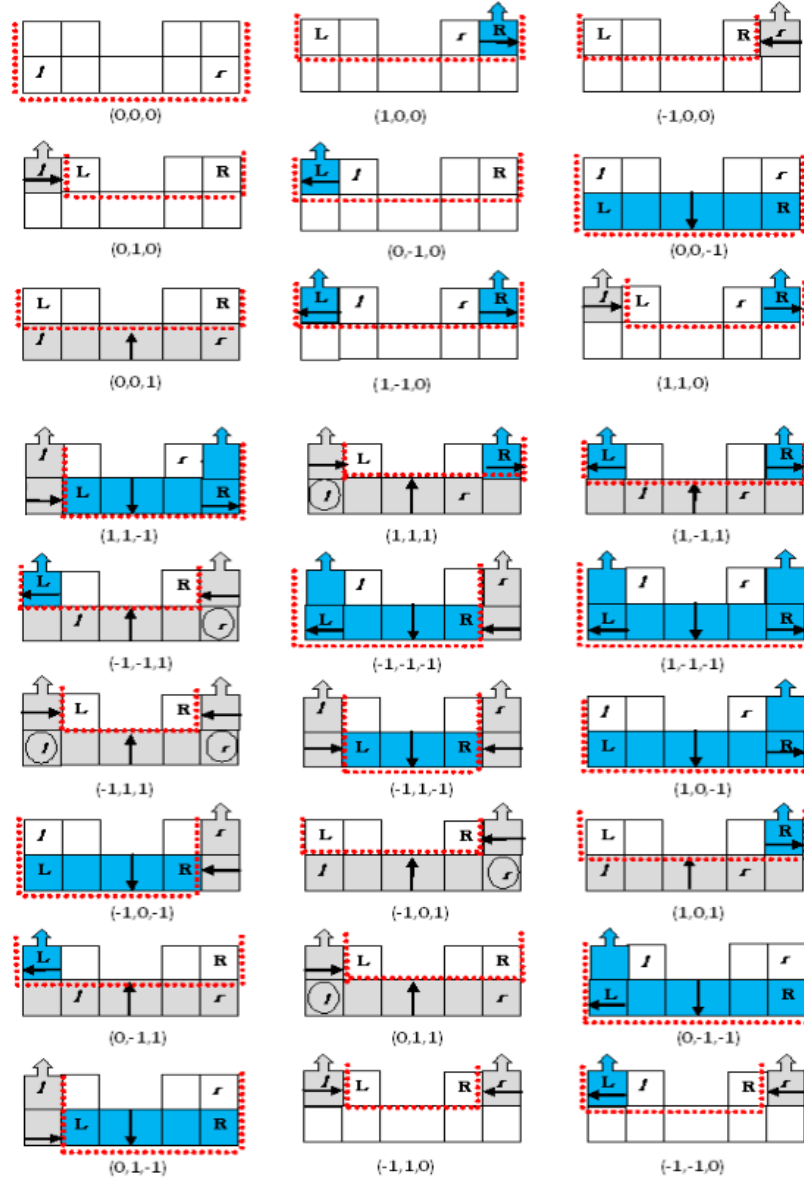


Figure 6.9: 27-cases for 3 parameters: Right, Left, Bottom

There are three COMSOL Multiphysics application modes used in the experiments: parametrized geometry (pg), sensitivity analysis (sa) and transverse magnetic mode (tm). The parametrized geometry (pg) mode provided a moving mesh to update the design according to the gradient of the geometrical parameters. By fixing the interface between air and silver of each groove the mesh deformation modeled the change in the size of the grating. This application uses only the space coordinate system. The key difference between the application of parametrized geometry and a moving mesh (ALE) is that in the latter the physics guides the movement of the mesh, but in the former the change in the mesh affects the physics through the objective function values.

Appropriate boundary conditions and domain definitions for the parametrized geometry

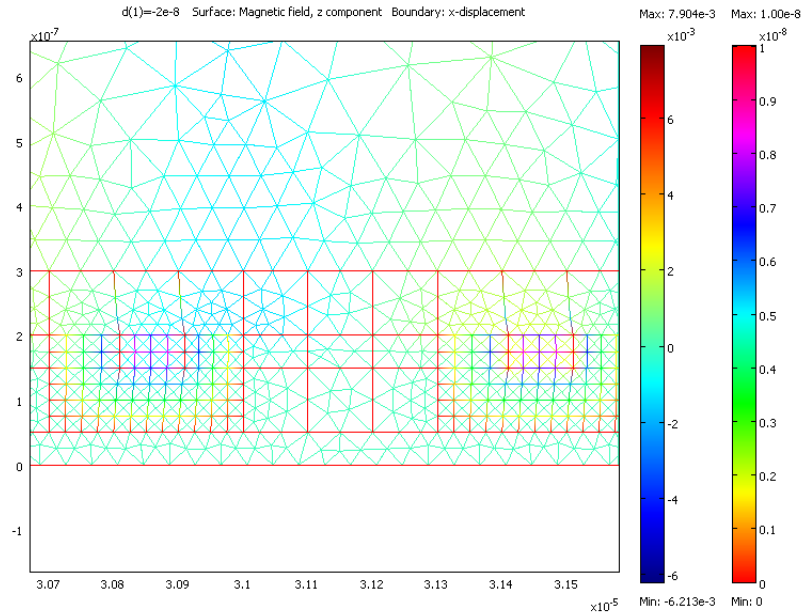


Figure 6.10: Multiphysics solution of moving mesh and magnetic field

would provide a smooth transformation of the mesh inside the domains, as a solution of the Laplacian PDE in the 2-dimensional displacement in space dx, dy , the velocity of the mesh. The boundary conditions include Dirichlet condition when a particular movement of the boundary, including no movement at all, is prescribed. Continuity boundary conditions are

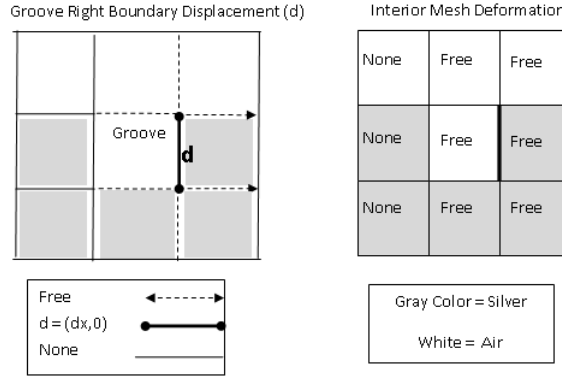


Figure 6.11: parametrized Geometry boundary and mesh definitions

assumed when the boundary can deform freely. The figure 6.10 illustrates the deformation of the mesh of two adjacent grooves using the parametrized geometry and parametric solver. In this case the multiphysics solution for both the magnetic field and the displacement is shown as the grooves increase in size toward the right boundary. The definition of the boundary conditions used in this deformation including prescribed displacement, shown in figure 6.11, aimed to make the deformation match the rectangular shape of a groove while changing the size.

The forward sensitivity analysis (SA) provided the sensitivity of solution to a parameter change. This is a key component of the gradient of the objective function that has been derived for the plane wave and Gaussian beam in (6.42)-(6.45) and (6.13). The sensitivity analysis mode is independent of the physics and has no boundary conditions or equations to be defined. The solver was defined to find: optimization> sensitivity analysis>forward sensitivity or in COMSOL script mode:

```
fem.sol =femstatic(fem, 'solcomp',
{'d9','d8','d5','d4','d7','d6','d1','d11','d2','Hz','d3','d10','y','x'}, ...
'outcomp',{'d9','d8','d5','d4','d7','d6','d1','d11','d2','Hz','d3','d10','y','x'}, ...
'blocksize','auto', ...
'sensmethod','forward', ...
```


'senscomp',{ 'd1','d2','d3','d4','d5','d6','d7','d8','d9','d10','d11'}).

The use of the Laplacian PDE of the parametrized geometry mode coupled to the TM electromagnetic equations and sensitivity analysis defined a form to study modification of the geometry of the grooves and how this affect the objective function. Nonetheless, the smooth deformation of the mesh also introduces changes to the objective function. This strategy, although developed to some extent, is not explored further in this dissertation work because of this. For more details on the management of COMSOL-Multiphysics application modes see Appendix A.

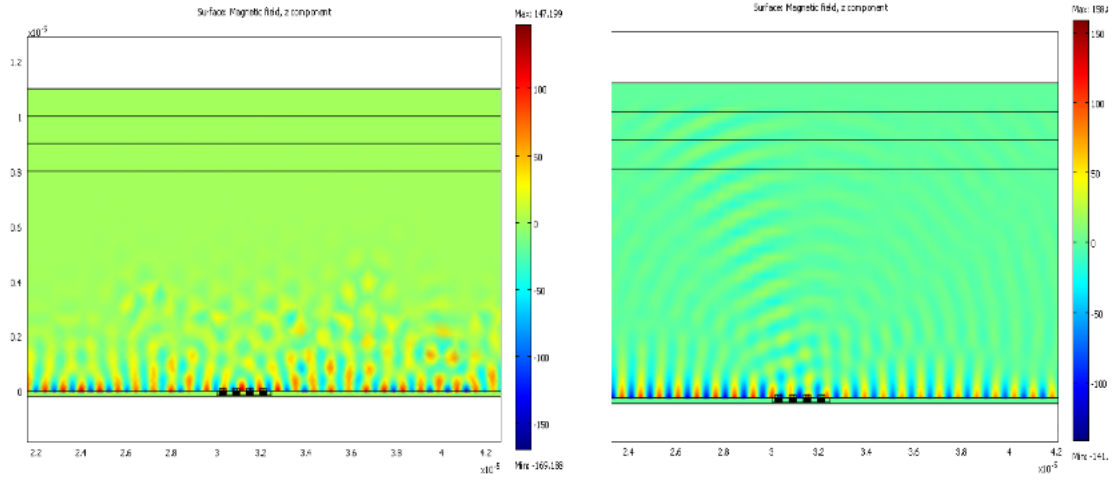


Figure 6.12: Coarse vs Refined Mesh Visualization

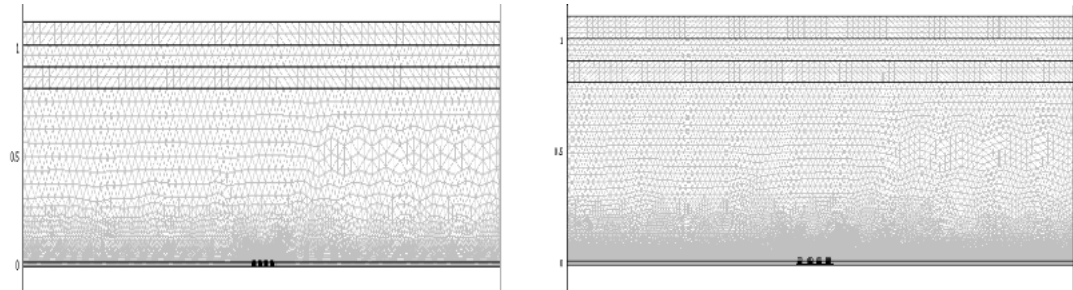


Figure 6.13: 139K and 557K elements meshes

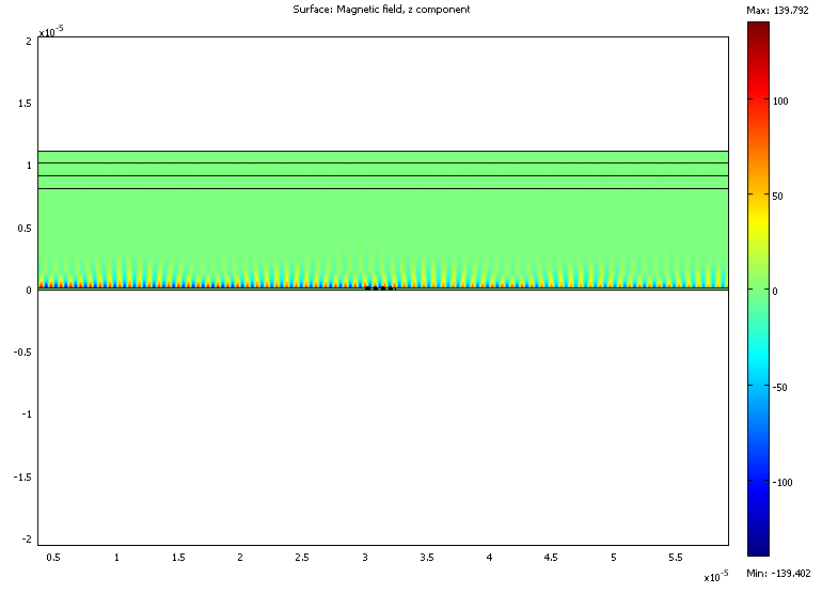


Figure 6.14: No-grooves with plasmonic Mode

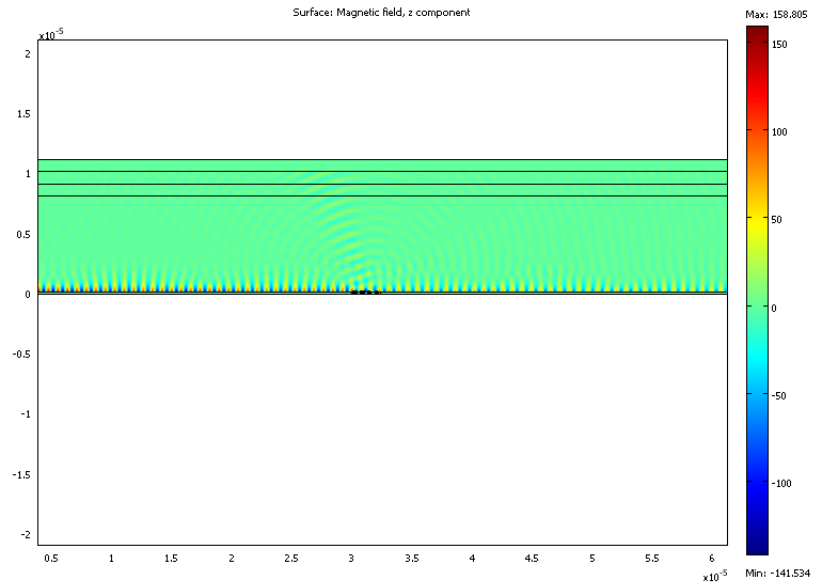


Figure 6.15: 4-grooves with Plasmonic Mode

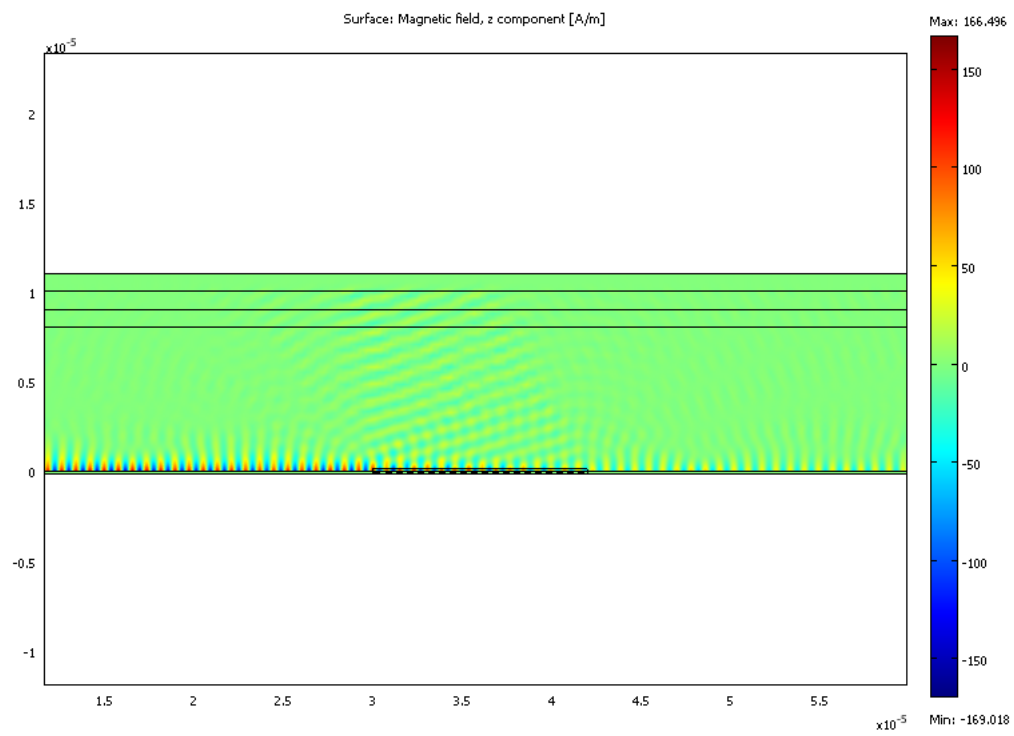


Figure 6.16: 20-grooves with Plasmonic Mode

The quality of the unstructured mesh chosen greatly affects the outcome visually and numerically. The computational domain is very big compared to the design domain and therefore the mesh has irregular refinement. The grating required a very refined mesh. The rest of the domain had no restriction. Some problems with quality of the mesh were encountered. The visualization of the plasmon helped identify the need of refinement of the mesh. In a refined mesh the quality of the waves showed an angle of propagation of roughly 120 degrees from the grating in direction to the P boundary. This could not be seen in a coarser mesh. Figure 6.12 shows the comparison between the visualization of the initial design using a refined and a coarser mesh. Also a significant numerical problem was detected in the objective function value. The no-groove design for both Gaussian Beam and Plane Wave objective function is assumed to be less than the 4-groove case. With a coarser mesh of 139,252 elements the value of the objective function with no groove is $P_0 = 2.06\text{e-}14$, and the 4-groove $P_4 = 7.78\text{e-}15$. After refining the mesh to 557,008 elements: $P_0 = 7.92\text{e-}13$, and $P_4 = 1.62\text{e-}12$. In figure 6.13 these meshes can be observed. Visualization using the refined mesh are included in figures 6.14-6.16 for no-groove, 4 and 20 grooves grating. These visualizations show an increase in emission from no-grooves, to the 20 grooves grating. The numerical gradient obtained in this dissertation work is discussed and analyzed.

6.5.2 Numerical Gradient

The numerical gradient of the 11 geometric design variables introduced in this work is obtained by using efficiently the available information of the solver but at the cost of accuracy. Because COMSOL uses the factorization of the FEM matrix to find the forward sensitivity for each parameter \mathbf{v} , the sensitivity is obtained with no extra cost from solver. The post processing of the sensitivity in order to obtain $\frac{\partial^2 H}{\partial \mathbf{v} \partial y}$ via the finite difference of $D_{\mathbf{v}}H$ is also a very inexpensive step. The sensitivity of the field to the 11 design parameters can be obtained from one solver step at the cost of a matrix multiplication for each parameter. The alternative of developing a new FEM model to obtain the electric field and its sensitivity to find a gradient will make each optimization step doubly expensive. Nonetheless by choosing

to post process the gradient via a finite difference approximation the algorithm inherits the errors associated this numerical scheme.

The sensitivity of the magnetic field is computed by forward sensitivity using COMSOL's static symmetric matrix solvers. Then it is also used to obtain the electric field sensitivity at a grid around the objective function boundary Γ . This is obtained numerically by a finite difference in space of the computed magnetic field sensitivities during post processing. The gradient obtained from this hybrid method is then used to update the design. Therefore the process to obtain the geometrical parameter's gradient is the most costly computationally, as opposed to the beam parameter optimization obtained by the derived formulas in (6.42)-(6.45), and (6.13).

The finite difference approximation of a gradient is dependent on the step size $h = dy$. In this dissertation work $dy = \{0.5, 0.75, 1.1\} \text{ e-7 m}$ are used. Some strategies to find a good step size include computing the gradient as an average of the value from different step sizes. In this work the 2nd degree central difference was used in the mesh deformation and plane wave optimization. For the Gaussian beam an 8th degree accuracy central difference formula is implemented to improve the gradient and was used to obtain the best designs with $dy = 0.5\text{e-7}$.

A tool to study errors of the numerical gradient was developed. It showed that there is no indication that any particular groove parameter is more error prone due to position. These tool can be used to further improve solutions by using only trustworthy parameter derivatives to continue the optimization algorithm, as identified by the tool.

In the next section the results of the use of the implemented computational tools to obtain the optimal design of the grating coupler are presented.

6.6 Optimal Design Results

In this dissertation work there are two approaches explored in the optimal design of the grating coupler: the parametrized geometry (PG) deformed mesh to update geometry and

the fixed mesh approach. The first one controlled only 4 geometrical parameters, the independent displacement of the right boundary of each grooves. The second approach was developed further to solve 14 dimensional optimization problem. These are eleven discrete geometric parameters and 3 Gaussian beam parameters. The most important results are obtained with the latter approach. Results for both are presented next.

6.6.1 Deformed Mesh Approach

The PG deformed mesh approach with the plane wave coupler objective function is the first numerical optimization technique implemented. Here the gradient has been obtained only for the 4 geometric parameters that control the movement of the right boundary of each groove. The advantage of this approach is that any step of movement can be chosen. The disadvantage is that as the mesh deforms the objective function value is also affected not only by a change in size but by mesh deformations and element quality. The deformed mesh value of the objective function P for a design where grooves $\{1,2\}$ are 150 nm wide, groove 3 is 120, and groove 4 is 125 nm, $(g1, g2, g3, g4) = (150, 150, 120, 125)$ nm, is $P = 2.910564\text{e-}24$. When verifying the value by creating a matching geometry and similar mesh quality gives $P = 3.49799\text{e-}24$. This is a relative difference of almost 17%.

The results of an initial numerical optimization run of the deformed mesh algorithm is summarized in the table 6.2. The numerical gradient is obtained by direct differentiation and 2nd degree finite central difference algorithm. A 314,272 element unstructured mesh was used, and an initial geometry was $(g1, g2, g3, g4) = (120, 120, 120, 125)\text{nm}$, where $g1...g4$ refer to grooves 1-4, and 120, 125 refer to the width of each corresponding groove. Only the right boundary is moved and there are then 4 parameters. The optimization step is initially $\alpha = 10$ nm, and is reduced as needed as in the gradient method.

Table 6.2: Moving Mesh optimization

| Design | Gradients | P value |
|-------------------|--|----------|
| (120,120,120,125) | DvP=(3.5e-13,-5.7-13,-6.3e-14,1.6e-13) | 3.05e-19 |
| (130,120,120,135) | DvP=(2.2e-13,-5.6e-13,-6.6e-14,1.6e-13) | 3.07e-19 |
| (140,120,120,135) | DvP=(9e-14, -5.5e-13, -6.5e-14, 1.2e-13) | 3.1e-19 |

Although this option gives better flexibility to choose the size of the grooves, the design could not be verified because of the difference in the objective function values when the geometry was redrawn and meshed. Also the introduction of more parameters to control the depth and left boundary movement of a groove will introduce bigger deformations to the mesh. Therefore a discrete design approach with 11 geometric parameters and a fixed mesh was chosen as the next numerical optimization technique.

6.6.2 Fixed Mesh Approach

Using the geometry design illustrated in 6.2.1 and a very refined mesh the composite 14 dimensional gradient of geometric and beam parameters was computed from analytical and numerical techniques. The Gaussian beam parameters are optimized analytically by the derivative equations in (6.42)-(6.45) for Gaussian beam, and (6.13) for the plane wave objective function. The current solution values of the electric and magnetic field are used and therefore this is a very fast computation. The 11 geometric parameters gradient is a composite procedure that uses the forward sensitivity and post processing. The forward sensitivity is obtained from the FEM solution at the cost of a substitution when the matrix has been factored by LU decomposition. The 8th degree central finite difference is used to obtain the electric field sensitivity in lieu of finding the solution for the electric field to compute the sensitivity. This is a fast solution that comes with a cost.

An algorithm to record the faulty numerical gradients obtained the following information. This can be used in the future to continue the optimization using the derivative that are trustworthy to update the corresponding design parameter. The bad derivative, those giving incorrect information, will remain at the current value, and will not update the design. Faulty sensitivity information in the gradient are expected as the design reaches close to optimal values. This is due in part to the smaller values of the derivative, and in part to the numerical errors introduced by the finite difference component of the sensitivity.

The plane wave coupler as well as the Gaussian beam coupler optimization produced changes

that incremented the directional emissivity of the device. Furthermore the results as measured by the wave coupler objective function measure the coupling and conversion rate of each device according to the targeted wave. This results provide more specific information on the performance of the simulated device for the practitioner when using different waves. The dissertation presents visualizations of the change in emissivity, value of the objective function and its conversion rate. The best conversion rate is produced by the Gaussian beam coupler as expected from the number of parameters that tailor the shape of the beam. The best results are presented and discussed.

The initial geometrical design (Design 0) consisted of four grooves of dimension 100×50

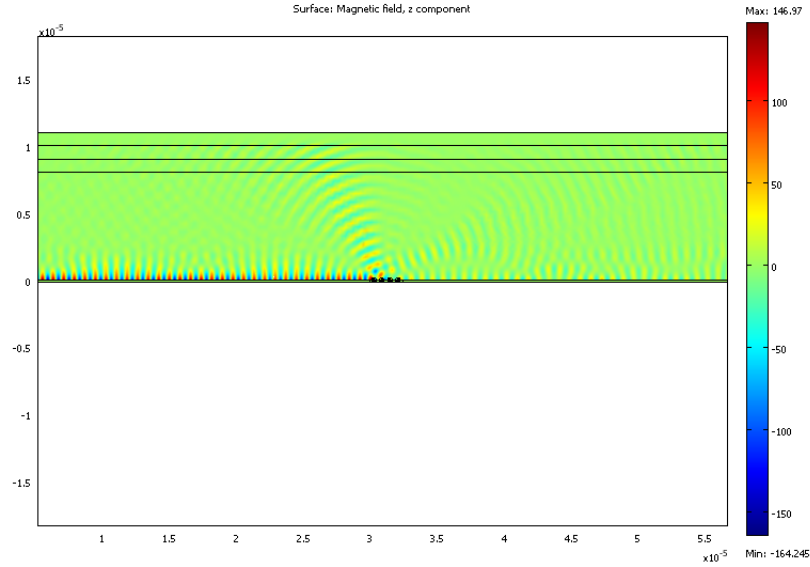


Figure 6.17: Final Design Solution for Plane Wave

$\text{nm} = \text{width} \times \text{depth}$. The separation between adjacent center of groove is 600 nm. The design was greatly improved by a change in geometry and angle guided by the 12 parameter gradient: 11 geometrical parameters and 1 for the plane wave. The initial design converts 0.008%, while the best design converts 1.92%, an improvement of 240 times the initial design conversion. Figure 6.17 shows the new magnetic field solution with the new groove

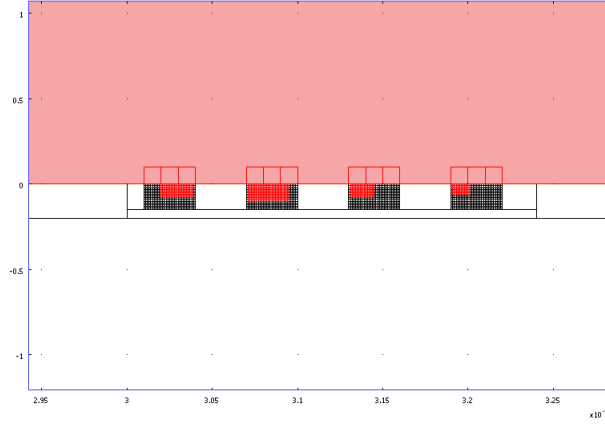


Figure 6.18: Grating Final Design for Plane Wave

design change: $(190 \times 80, 240 \times 100, 140 \times 80, 100 \times 60)$ showing a change in strength. Figure 6.18 shows the final design for the plane wave. Even though the new design shows great improvement with the plane wave, the small conversion rate is of little interest for practitioners. For 20 grooves the initial design could only convert 0.2%. Although this is 25 times better than the 0.008% of the 4 grooves initial design, the small percentage indicated that a design with a Gaussian beam is a better option to obtain a design that gives a significant conversion percentage.

The optimal design of the Gaussian beam grating coupler is the best grating coupler device in this dissertation work. The results of the best designs are summarized in the table 6.3 for $\lambda=800$ nm. These results use the restrictions of the design given by practitioners in NIST: $\text{dist} \geq 62.5 \text{ e-}6 \cdot \cot(\theta)$, $w_0 \in [3.3\lambda, 3.5\lambda]$. All the designs shown in the table have greater conversion percentage with $w_0 = 3.3\lambda$, because the normalization constant is smaller: $P_N = 4.386294\text{e-}9$. An improved design was obtained by conducting simultaneous optimization of the geometry and Gaussian beam parameters, from the initial geometrical design obtained from practitioners, Design 0. When the gradient could not improve the objective function, only parameters of the Gaussian beam are optimized. This produced for Design 1 a strictly increasing objective function from $P=6.755307\text{e-}10$ to $6.780889\text{e-}10$. The

Table 6.3: Fixed Mesh optimization

| Design | grooves | GB parameters | P value | |
|--------|-----------------------------------|---|----------|---------|
| 0 | (100x50,100x50, 100x50,100x50) | $s=5.49052\text{e-}7$, $\theta=1.58\text{e-}5$, dist=1.81 | 1.71e-10 | 6.3% |
| 1 | (130x80,120x80, 100x80,100x80) | $s=3.257\text{e-}7$, $\theta=1.813527$, dist=1.551379 | 6.76e-10 | 24.89% |
| 2 | (150x60,130x80, 170x80,140x80) | $s=3.4\text{e-}7$, $\theta=1.81$, dist=1.525e-5 | 6.81e-10 | 25.088% |
| 3 | (160x70,150x80, 160x80,120x90) | $s=4.34163\text{e-}7$, $\theta=1.831687$, dist=1.994596e-5 | 7.03e-10 | 25.898% |
| 4 | (160x70,150x80, 160x80,120x90) | $s=3.4\text{e-}7$, $\theta=1.81$, dist=1.525e-5 | 7.3e-10 | 26.89% |

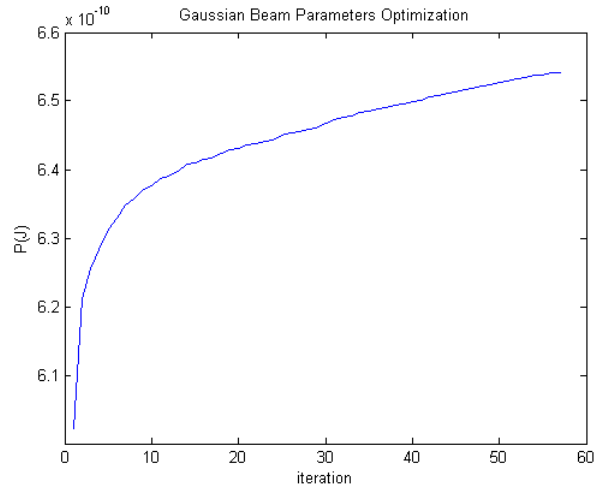


Figure 6.19: Optimization for Gaussian Beam Parameters Only

results of the Gaussian beam parameter optimization are summarized in the graph 6.19. The total run, geometric and Gaussian beam parameters, took 39.38 minutes in the Dell Precision T7400 desktop computer.

The best design, Design 4 is obtained by the use of various computational tools. The initial design, Design 0 was used to obtain Design 1 using the simultaneous optimization tool. This design was used as the new initial design to obtain Design 3. Finally the results of the optimization of the optimal Gaussian beam parameters used with the optimal geometry produced Design 4 the final and best result.

The simplicity of the silver/air design could not convert more than 65% because it is known that 35% of the energy is lost in the substrate. For this reason more complex designs use silicon on insulators couplers with bottom reflectors as in [37]. Also it is estimated in previous work [34] that the minimum number of grooves for a correct wave vector matching is approximately nine. The best conversion rate obtained with a Gaussian beam in a four groove grating is 26.9%. Nonetheless some percentage is lost to the substrate. In some experiments it is reported that 65% is lost, giving more than a 43% relative conversion.

The figures 6.20-6.25 show the best geometrical designs obtained. In this way the maximum conversion of the device is restricted by the simplicity of the design. Nonetheless the initial design obtained from practitioners at NIST has been improved greatly, more than 100 times, by the use of gradient-based numerical optimization. The initial Gaussian beam design was improved from 6.3%, a result obtained using good Gaussian beam parameters, to 27%. This is more than four times the efficiency of the initial design, within the constraints of the optical fiber, and using only a silver/air design.

The next and final chapter presents a summary of the dissertation work, analysis of the obtained results, and future work.

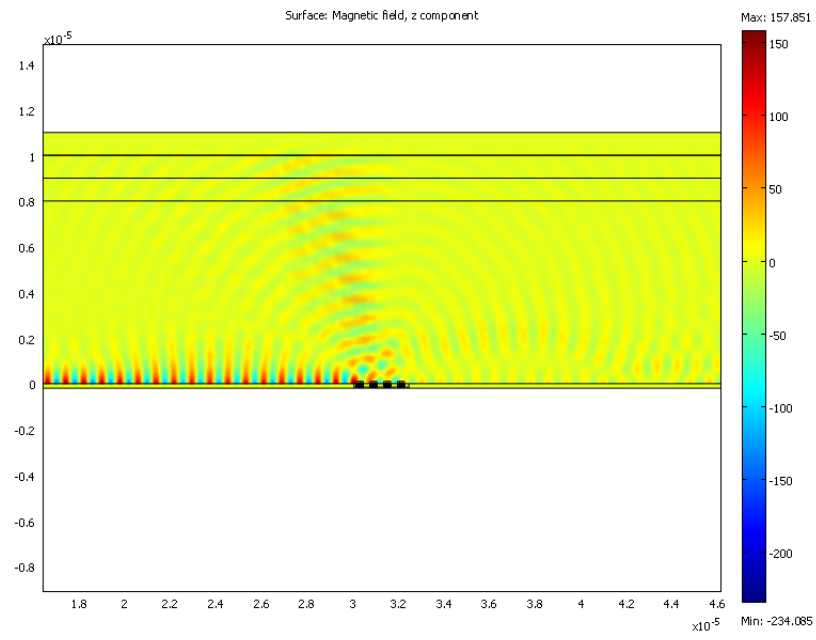


Figure 6.20: Design 1: Magnetic Field

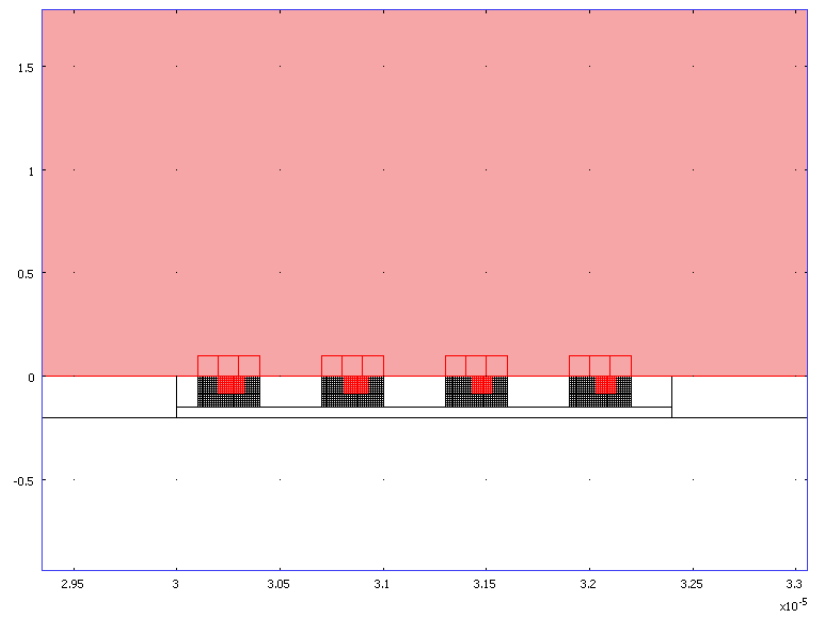


Figure 6.21: Design 1: Grating

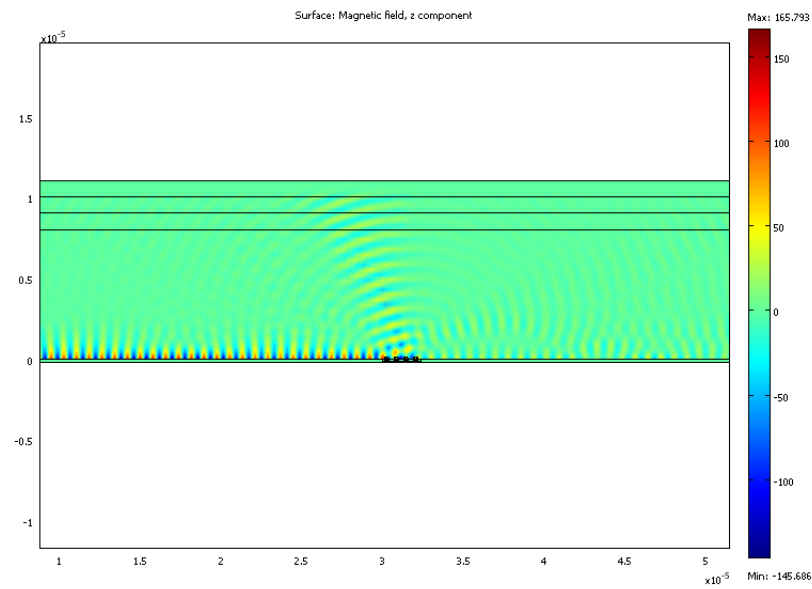


Figure 6.22: Design2 : Magnetic Field

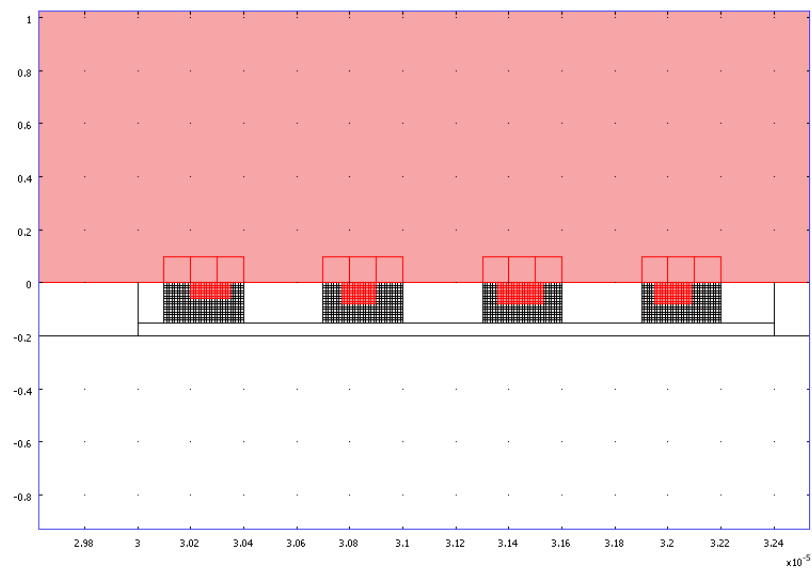


Figure 6.23: Design 2: Grating

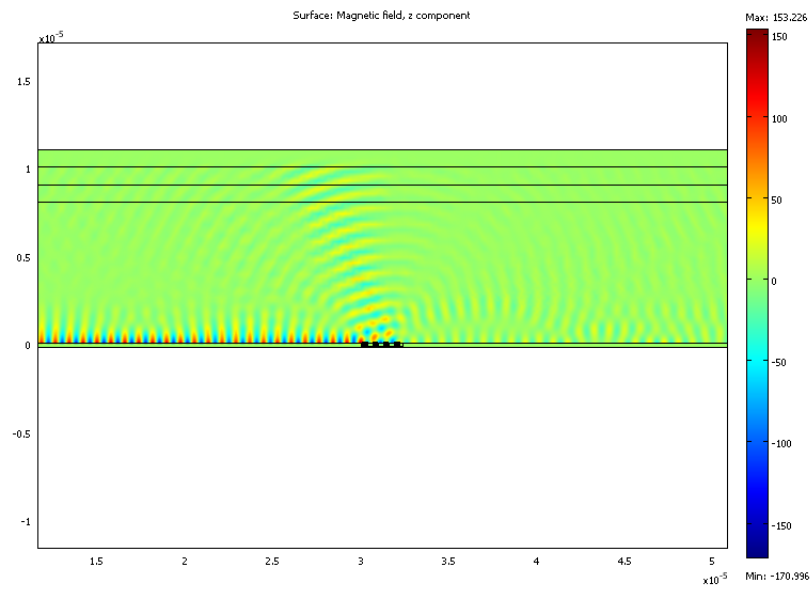


Figure 6.24: Best Design 4: Magnetic Field

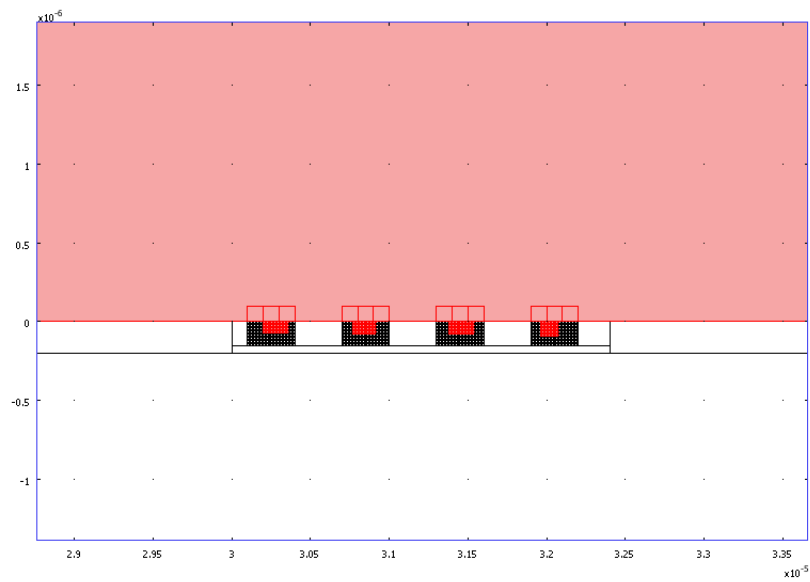


Figure 6.25: Design 4: Grating

Chapter 7: Conclusions and Future Work

The research work in this dissertation has been built like a path in new terrain: by walking over the same area in a consistent direction, and a little further ahead. Following the natural direction from the simplest to the more involved, this project is paved with answers to the challenges encountered. From one to two dimensions, from real to complex parameters, from sensitivity to parameters of equations to those of a geometry, from the optimal design of a silver square to that of a grating, from energy conversion of a plane wave to that of a Gaussian beam, from MATLAB to composite COMSOL Multiphysics-MATLAB tools, the dissertation has developed more than one useful products. Similar to a new built path, the formulations, methodology and computational tools contributed in this dissertation are for others to explore similar problems in multidirectional areas of research.

The contributions in this dissertation are summarized in the following list.

1. Derivations of formulas for the following:
 - (a) the gradients of the objective functions with respect to the design parameters for two problems: the maximum light absorption and grating coupler optimal design problems.
 - (b) the Adjoint Method for the grating coupler problem.
 - (c) the error bound for finite difference approximation of the matrix sensitivity.
 - (d) proof of an equivalent formulation for the inverse matrix sensitivity.
2. Development of the 1D and 2D AMPL models for maximum light absorption of a silver nanoparticle in air.
3. Development of the methodology and computational tools for the numerical optimization of the maximum light absorption of a silver nanoparticle.

- (a) Developing a MATLAB based Finite element solver for 2D silver rectangle in air and visualization tools.
 - (b) Finding the optimal wavelength for the 30 nm silver square in air.
 - (c) Computing gradient for permittivity of silver to determine the optimal size of the silver rectangular nanoparticle.
 - (d) Finding the optimal size for the maximum light absorption per area of the silver rectangle nanoparticle.
4. Development of the methodology and computational tools for the numerical optimization of the plane wave and grating coupler.
- (a) Finding the optimal design for the plane wave grating coupler.
 - (b) Finding the optimal design for the Gaussian beam grating coupler.

Some details of these contributions are further discussed next.

7.1 Derivation of formulas

In this dissertation several numerical optimization techniques have been developed while working on two main design problems. For these problems, the maximum light absorption and the grating coupler, a methodology to conduct gradient based optimization have been developed. Therefore a first contribution is to obtain the formulations of the objective functionals and the derivation of gradients from analytical as well as numerical methods.

The formulas for the gradients using a combination of finite difference and direct differentiation of the FEM linear system have been derived from governing equations in this work. A proposition stating the mathematical structure of the Adjoint Method gradient for the optimal design of the grating has been proved. The added value of these derivations is obtained from the complexity of these objective function. These objective functions and their gradients are not commonly used in literature of optimal design problems. The optimal design of a silver nanoparticle that maximizes absorption per area has not been found in

the literature review for this dissertation to this day. However, addressing this problem is an important step for the design of efficient sensors. The optimal design of the grating coupler is a very important optimal design problem based on literature review, but the objective function is usually simplified to measure power in a far-field approach. In this dissertation a wave coupler function tailored for specific waves yield a conversion rate value. This information is more useful for practitioners to appreciate a design that can be built. The maximum light absorption functional was formulated in terms of the magnetic field variable (H) in order to use the solution obtained from FEM solution of the Helmholtz PDE in H . The gradient of the maximum light absorption objective function was obtained with respect to two parameters: the real valued wavelength (λ) and the complex valued permittivity (ϵ). This gradient is a combination of direct differentiation and finite difference approximation of the sensitivity with respect to the electric field. The sensitivity of the FEM matrix is computed from direct differentiation, and the numerical approximation error of a finite difference approximation is obtained. More details are provided in chapter 5.

The wave coupler functional P was derived using two different source beams: the plane wave and the Gaussian beam. The formulation of the gradient of P for the geometric parameters \mathbf{v} was obtained first, followed by the gradient of P with respect to the beam parameters $\theta, dist, s$. The Adjoint method formulation was also obtained for the plane wave design. More details are found in chapter 6.

7.2 Development of the 1D and 2D AMPL models

The AMPL models developed and implemented in this dissertation project contribute to the optimization with PDE constraints. The 1D and 2D models show a way to conduct optimization with PDE constraints as a comprehensive mathematical modeling problem. The Finite Element matrix defined through sets of nodes is a key component to define the PDE constraints. Nonetheless, even in 1D, the model can successfully solve the PDE. It is observed that SNOPT 5.3 found a solution by reaching the wavelength's lower limit, the

maximizer. The 2D model was also successfully developed, even though the complex valued matrix that characterizes the electromagnetism equations makes the problem twice the size of the matrix in just real domain. Each node in the grid is a variable in constraints and functions of this large scale optimization problem. Therefore using symmetry of the computational domain, the number of variables was reduced to a half domain. The half domain model built the matrix from equations using node sets and obtained the connectivity from the known structure of a uniform grid. The 2D model found the magnetic field solution for the PDE. Again, most solvers, including the state of the art NEOS-server solvers failed to converge to a maximizing wavelength. Nonetheless a solution was found by IPOPT 3.10.1 in coarse and refined grids. The limitations inherent to the cubic spline definition of the permittivity from experimental data added complexity to the problem, and met the modeling language limits. There is room for further development of these tools.

7.3 Numerical optimization of the maximum light absorption functional

The development of the methodology and computational tools for the numerical optimization of the maximum light absorption of a silver nanoparticle in air was studied as two separate problems: the optimal wavelength for a 30 nm silver square, and the optimal size of the silver nanoparticle. The methodology, and computational tools developed in MATLAB are discussed in this section.

In the first problem the gradient of the light absorption functional J with respect to the real valued parameter λ (free space wavelength) has been obtained and implemented. The first challenge encountered in the derivations is the fact that λ is ubiquitous in the variables of the objective function and PDE. The next challenge is the fact that J is defined in terms of the electric field, while the PDE are in the magnetic field. Thus the gradient of J with

respect to λ required the derivative of the sensitivity of the solution to λ . The dissertation used direct differentiation (DD) of the FEM linear system to obtain the magnetic field sensitivities, and central finite difference in a uniform grid to obtain the electric field sensitivities. The obtained gradient was implemented in MATLAB and compared to a finite difference approximation of J as a function of λ . The newly obtained gradient compared favorably with the finite difference approximation because of its efficiency. Unlike the finite difference gradient that requires at least two solver iterations per parameter, the obtained gradient uses only the current matrix data and solution for all parameters in the problem. This numerical technique to find the sensitivity with respect to the electric phasor is to be continuously used throughout the dissertation work in optimal design.

The second problem is to find the optimal size of a silver rectangular nanoparticle that maximizes J/A , i.e. light absorption per area. This was studied using a topology optimization approach with the added value of considering a structures of elements as the design variable. The gradient of J/A with respect to a strip of horizontal or vertical elements surrounding the silver particle would point to an increase or decrease of the size of the silver square. The permittivity of either air or silver for a structure of homogeneous elements was parametrized as $\epsilon(t) = \epsilon_1 + (t)(\epsilon_2 - \epsilon_1)$, $t \in [0, 1]$ in the design domain. The gradient of J/A with respect to $\epsilon(t)$ was derived and implemented in MATLAB. The grid had to be refined in order to obtain consistent results for gradient and the functional J/A . The gradient showed that an increase in the size of the silver square increases the absorption per area. The square was generalized to a rectangle of the same area. Visualization and values for J/A obtained for various rectangular and symmetric geometrical shapes of same area, suggest that the square is the optimal shape to maximize light absorption per area. Using a fixed range for all graphs, visualizations show a stronger electric field surrounding the boundaries and corners of the square. A proof or disproof that the square is the optimal shape for J/A is an interesting future problem to study in shape optimization.

The optimal size of the silver nanoparticle was more challenging. The gradient of gradient of J with respect to a complex valued permittivity parameter ϵ is a vector gradient when

ϵ is decomposed in real and imaginary components. There are some proposals in literature to extend the complex gradient to non analytic functions such as J , and it is a commonly accepted that $\frac{\partial J}{\partial \text{conj}(E)} = 0$. In the dissertation work the gradient was obtained using the standard mathematical procedure of decomposing the expressions in real and complex parts. This became a standard practice that continued throughout the project and proved to be very useful in the optimal design work.

The dissertation work has produced much needed computational tools for the optimal design of devices. The simplest tool is a 1D and 2D FEM solver and visualization tools in MATLAB. This FEM solver can use either sets or an implicit structured mesh to assemble complex and real valued matrices. The visualizations are also conducted in MATLAB and include a time-harmonic animation. To the solver tool the computation of gradients of the light absorption problem with respect to the complex valued permittivity of silver and air is added. The capability of finding the gradient with respect to a strip of air or silver elements is built to use in a topology optimization approach. These tools have proven useful in the optimization of light absorption per area problem.

7.4 Numerical optimization of the Grating Coupler

The optimal design of the grating coupler was challenging in the design of geometry, boundary conditions and gradient formulation. The dissertation took advantage of the simplifications introduced by the output coupler formulation in order to conduct optimization. By defining the plasmon beam as incident on the left boundary and thus part of the PDE, the optimization changes of the output beams' parameters will not require solving a modified PDE, but only updating values according to the derived formulations of P and its gradients. Some new challenges included more specialized boundary conditions. The requirements included absorbing boundary conditions equipped with perfectly matched layers to indicate

the direction of absorption on each of three domains surrounding the air part of the computational domain. The grating coupler also required a port in order to define the plasmon mode as irradiating at the boundary.

A final challenge in the development of the gradients is finding the gradient with respect to shape or geometric parameters of the 2D grating coupler design. The complexity of the defined objective function $P(J) = P$ required both magnetic and electric field sensitivities. The same strategy used to obtain sensitivities in the previous problems applied, with the added complexity that geometrical changes affect the shape interpolation functions used to build the FEM matrix. The forward sensitivity of the magnetic field was obtained from COMSOL-Multiphysics. The sensitivity of the electric field was obtained with an improved 8th accurate central finite difference. But this is only the geometrical parameters of a more involved gradient. The complete gradient for the optimal grating problem included parameters that govern the output waves: plane wave and Gaussian beam. Because the beams could be transformed by changing the angle of incidence and distance from the silver slab and grating, another contribution is obtained in the derivation of the form of the objective function P from the transformed coordinates of a Gaussian beam and plane wave. Then the dissertation work developed formulations for the derived P gradients of the Gaussian beam and plane wave parameters: $\theta, s, dist$ that modify the beam's position in the 2D plane.

The hard work of modeling and deriving the formulations reaped good rewards. Many designs were obtained for the plane wave and Gaussian beam grating. Nonetheless the best design in energy conversion is the Gaussian beam coupler. Considering the available energy in the system to be 65% [37], the Design #4 attains better than a 40% conversion for the Gaussian beam coupler. The 27% conversion of the available energy is a significant accomplishment for such a simple design. It has been reported that 35% of the energy is lost in the substrate and therefore more elaborate designs use silicon on insulators couplers with bottom reflectors as in [37]. A grating with 4 grooves is also less than the minimum of 9 grooves reported in literature for correct wave vector matching [34]. The design incorporated also constraints on the available equipment that affects the waist and distance of

the Gaussian beam. The use of the computational tools developed in this dissertation open the possibilities to discover many more new designs for efficient couplers.

There are multiple computational tools developed. The optimal design of the grating coupler tools tackle a 14 dimensional optimization problem using combined COMSOL-MATLAB tools. The geometrical model for the gratings allow the use of a fixed unstructured and refined mesh. A gradient based update of 11 geometrical and 3 analytical parameters produced much improved designs for the plane wave and Gaussian beam nanoplasmonic systems. Algorithms to compute geometry only optimization, beam only parameter optimization, and a combined run of geometry and beam are implemented. There are many ways to conduct numerical optimization using these tools even when the design update is not improving the objective function. For example, after geometry gradients are no longer giving improvement, the algorithm can continue optimization of beam parameters for the best attained geometry.

These flexible numerical optimization tools are important for electromagnetism and a bigger interdisciplinary area. In the geometrical parameter optimization, for example, the use of rectangular domains serves as a first step to designing other arbitrary shapes. The tools developed in this project applied to any design can help to determine the size or scale of a shape that will produce a better design according to a optimal bounding rectangle, providing an initial draft for components of more involved designs. The hardware requirements are not high, because program runs in a Dell Precision T7400 desktop computer are less than an hour long.

The fact that the gradient of the wave coupler function uses a finite difference approximation for electric field sensitivity is efficient but comes with the defects associated with this numerical scheme. Therefore a tool to identify faulty gradients has been implemented providing the capability of continuing geometry optimization using only the reliable components of the gradient. Nonetheless this is a perfect place to continue development.

7.5 Future Work

This dissertation has many open problems and therefore there are many paths to continue exploring in the near future. The 2D AMPL model can continue development by solving other simulation based models. The silver light absorption problem and tools can continue development to find the optimal shape per area. Nonetheless, the most successful and important tools for the optimal design of the grating coupler are presently undergoing much development.

The grating coupler tools have proven useful in this first beta version, and many more designs can be discovered and tested. Nonetheless there is much space for improvement and development. The tools for the development of the grating couple have found the optimal design that the accuracy of the discrete size updates yield. The accuracy can be further improved. The model presented in this work that uses a moving mesh is a promising strategy. This would have the capability of updating close to optimal design a small arbitrary distance. This is a significant improvement on the discrete step updates that can help the tools to converge to an optimal design.

It is very interesting to implement a grating model with more an arbitrary number n of grooves and compare the conversion percentage. Some improvements to the tools are planned for this purpose. A general geometry tool that can automatically produce the design for n grooves requires a text utility to identify the variables and lists for COMSOL application modes. This is a very useful addition because the definition of 1827 silver/air domains and 3854 boundaries takes a lot of time and is not fun.

To improve the gradient information there are various options planned. The finite difference gradient can be substituted by a differential quadrature or other numerical gradient method. The study of the objective function can also produce simplifications that do not require the use of the sensitivity to the electric phasor and thus eliminate the need to use an approximation to the gradient. The Adjoint Method gradient is a promising option because it is independent of the magnetic and electric phasor solutions. Some derivations have been already obtained in this work that remains a continuously developing path.

Appendix A:

```
AMPL model developed on July 2012.

\* AMPL Model for PDE optimization in 1/2 domain

\* Need to change pr,pc in order to achieve feasibility
    during initialization according to lamda!

\* Some values of lambda are hard converge: lam=400, but the PDE
is satisfied.

reset;

model;

option substout 1;


param nodes := 51; \*number of elements per grid +1
param nodes1 := 52; \*nodes per grid +1
param nhalf := round(nodes2/2);
param npoin := nodes*nhalf; \*number of half grid nodes

\* Spline Parameters (if changed need to recompute data with Matlab)
param nl := 200E-9; \*min lambda in spline from data
param nr := 400E-9; \*max lambda in spline from data
param nperm := 180; \*lambda step is 10 in data
param pi := 3.141592653589793;
param eps0 := 8.8541878176E-12;


\* Variables

\* x is the wavelength

var x := 310e-9, >= 300e-9, <=400e-9;
```

```

var k = 2*pi/x;
\*>=2*pi/400e-9, <=2*pi/200e-9;
var Ur{1..npoin}; \* real part of solution
var Ui{1..npoin}; \*complex part of solution

#parameters
var pr :=1.33827; \* requires good initial value
permittivity for lambda = 300nm
var pc := -1.73245; \* for lambda = 300nm
#var pc := - 0.757719496436341; #permittivity complex for lambda = 350
#var pr := -8.479293893595084; \* permittivity real
param silc {1..nperm,1..4}; \* Silver cubic Spline complex
param silr {1..nperm,1..4}; \* Silver cubic Spline real
param c0 := 299792458;
param L := 150E-9; \*space value z= [0,L]
param h := L/(nodes-1); \*uniform mesh and space steps
param sl := round(60E-9/h + 1); \*left silver node: 85/h + 1
param sr := round(90E-9/h+1); \*silver node right: 115/h +1

\* Buildind sets
\*silver corner and boundary nodes
param sn1 := (sl-1)*nodes+sl; \*177
param sn2 := (sl-1)*nodes+sr; \*181
param sn3 := (nhalf-1)*nodes+sr; \*265
param sn4 := (nhalf-1)*nodes+sl; \*261
set sil:=: {1144..1152}; \* interior Silver nodes;

set sb1 := sn1+1..sn2-1;

```

```

set sb2 := sn2+nodes..sn3-nodes by nodes;
set sb3 := sn4+1..sn3-1;
set sb4 := sn1+nodes..sn4-nodes by nodes;
set sc1 := {sn1};
set sc2 := {sn2};
set sc3 := {sn3};
set sc4 := {sn4};

\*vacuum corners and boundary nodes
param vn1 := 1;
param vn2 := nodes;
param vn3 := nodes*nhalf;
param vn4 := nodes*(nhalf-1)+1; #vacuum corners
set vb1 := vn1+1..vn2-1;
set vb2 := vn2+nodes..vn3-nodes by nodes;
set vb3 := {vn4+1..sn4-1} union {sn3+1..vn3-1};
set vb4 := vn1+nodes..vn4-nodes by nodes;
set vc1 := {vn1};
set vc2 := {vn2};
set vc3 := {vn3};
set vc4 := {vn4};

set allbd := vc1 union vb1 union vc2 union vb2 union vb4
union vc4 union vb3 union sc4 union sb3 union sc3 union vc3;

\* Define boundary segments
\*param b1 = nodes-1;
\*param Bs1{1..b1,1..2};
\*param Bs2{1..b1,1..2};

```

```

\* Real 3 x 3 matrix elements for uniform elements: h*w=h^2
var K11 = 1/2 - k^2*h^2/12;
var K12 = -1/2 - k^2*h^2/24;
var K13 = -k^2*h^2/24;
var K22 = 1 - k^2*h^2/12;
var KM11 = pr/(2*(pr^2+pc^2)) - k^2*h^2/12;
var KM12 = -pr/(2*(pr^2+pc^2)) - k^2*h^2/24;
var KM22 = pr/(pr^2+pc^2) - k^2*h^2/12;
var Ar{1..npoin,1..npoin};
var Ac{1..npoin,1..npoin};

\* Imaginary (Silver) 3x3 matrix elements
var KMi11 = -pc/(2*(pr^2+pc^2));
var KMi22 = -pc/(pr^2+pc^2); \*center silver complex element
var KMi12 = pc/(2*(pr^2+pc^2));
\*\*\*\* Boundary conditions at z=0 and z=L
var ksi11 = k*h/3; \*for left and right boundary Condition nodes
var ksi12 = k*h/6;
\*\*\*\* B vector at z=0
var bs11 = k*h/(c0*pi*4E-7); \*=2*k*h*H0/2 only left boundary

\*\*\*\* Build Ar Real part of Matrix
s.t. m1 {i in 1..npoin, j in 1..npoin}:
Ar[i,j] = if (i==j) then (if i in sil then 4*KM11+2*KM22 else
                        if i in {sc1} then 2*K11 + 2*K22+2*KM11 else
                        if i in {sc3} then 2*KM11 + K22 else
                        if i in {sc2} then 4*K11 + K22+ KM22 else

```

```

        if i in {sc4} then KM22 + 2*K11 else
        if i in {sb1 union sb2 union sb4}
then 2*K11+K22+2*KM11+KM22 else
        if i in {sb3} then 2*KM11+KM22 else
        if i in {vc1 union vc3} then 2*K11 else
        if i in {vc2 union vc4} then K22 else
        if i in {vb4 union vb2 union vb1 union vb3}
then 2*K11 + K22
else 4*K11 + 2*K22)
    else if (i== j-1 or i == j+1) then
    (if (i in {vb2 union vc2 union vc3} and j>i) then 0
    else if (i in {vb4 union vc1 union vc4} and j<i)
then 0 else if i in {vb1 union vb3 union vc1
    union vc2 union vc3 union vc4}
then K12 else if ((i==sn4 and j<i) or (i==sn3 and i<j))
then K12
    else if ((i in sb1) or (i == sn1 and j>i) or (i == sn2 and i>j))
then K12 + KM12 else if ((i in sb3) or (i == sn4 and j>i)
    or (i == sn3 and i>j))
then KM12 else if ((i in sil)
    or (i in sb4 and j>i) or (i in sb2 and i>j))
then 2*KM12
    else 2*K12)
        else if (i==j-nodes or i==j+nodes)
then (if i in {vb2 union vc2 union vc3 union vb4
    union vc1 union vc4} then K12 else
    if ((i in {sb2 union sb4}) or (i in {sc3 union sc4} and j<i)
    or (i in {sc2 union sc1} and i<j)) then KM12 + K12 else

```

```

if ((i in sil) or ((i in sb1) and (j>i)) or ((i in sb3) and (i>j)))
  then 2*KMi12 else 2*Ki12)
      else if (i==j+nodes+1 or i==j-nodes-1)
  then (if i in {sil union sb1 union sb3} then 2*Ki13
else  if (i in {vb2 union vc2 union vc3} and j>i) then 0
else if (i in {vb4 union vc1 union vc4} and j< i) then 0 else 2*Ki13)
      else 0;

```

**** Build Ac Complex part of Matrix

```

s.t. m2 {i in 1..npoin, j in 1..npoin}: Ac[i,j]= if (i==j) then
(if i in sil then 4*KMi11+2*KMi22 else
      if i in {sc1 union sc3} then 2*KMi11 else
      if i in {sc2 union sc4} then Ki22  else
      if i in {sb1 union sb2 union sb3 union sb4}
      then 2*KMi11+Ki22 else
      if i in {vc1 union vc3 union vc2 union vc4}
then ksi11
else  if i in {vb4 union vb2} then 2*ksi11
      else 0)
      else if (i== j-1 or i == j+1) then
(if ((i in {sb1 union sb3}) or (i in {sc1 union sc4} and j>i)
or (i in {sc2 union sc3} and i>j)) then Ki12 else
      if ((i in sil) or (i in sb4 and j>i) or (i in sb2  and i>j))
then 2*KMi12 else 0)
      else if (i==j-nodes or i==j+nodes)
then (if i in {vb2 union vc2 union vc3 union vb4 union vc1 union vc4}
then ksi12 else
      if ((i in {sb2 union sb4})

```

```

or (i in {sc3 union sc4} and j<i) or (i in {sc2 union sc1} and i<j))
then KMi12 else
    if ((i in sil) or (i in sb1 and j>i) or (i in sb3 and i>j))
    then 2*KMi12 else 0) else 0;

\* Right hand side
var bc {i in 1..npoin} = if (i in {vc1 union vc4})
then bsil else if i in vb4
then 2*bsil else 0; \*Complex part of right hand side
var aEx {i in sil} = ((Ur[i+1]-Ur[i-1])*pc
+(Ui[i-1]-Ui[i+1])*pr)*x/(4*h*pi*c0*eps0*(pr^2+pc^2));
var bEx {i in sil} = ((Ur[i+1]-Ur[i-1])*pr
+(Ui[i+1]-Ui[i-1])*pc)*x/(4*h*pi*c0*eps0*(pr^2+pc^2));
var aEz {i in sil} = ((Ui[i+nodes]-Ur[i-nodes])*pr
+(Ui[i-nodes]-Ui[i+nodes])*pc)*x/(4*h*pi*c0*eps0*(pr^2+pc^2));
var bEz {i in sil} = ((Ur[i-nodes]-Ur[i+nodes])*pr
+(Ui[i-nodes]-Ui[i+nodes])*pc)*x/(4*h*pi*c0*eps0*(pr^2+pc^2));

\*PDE constraints
subject to Pde1 {i in vc1}: Ar[i,i]*Ur[i]+ Ar[i,i+1]*Ur[i+1]
+Ar[i,i+nodes]*Ur[i+nodes]+Ar[i,i+nodes1]*Ur[i+nodes1]
-Ac[i,i]*Ui[i]-Ac[i,i+1]*Ui[i+1]-Ac[i,i+nodes]*Ui[i+nodes]
-Ac[i,i+nodes1]*Ui[i+nodes1]=0;

subject to Pde2 {i in vc3}: Ar[i,i]*Ur[i]+ Ar[i,i-1]*Ur[i-1]
+Ar[i,i-nodes]*Ur[i-nodes]+Ar[i,i-nodes-1]*Ur[i-nodes-1]
-Ac[i,i]*Ui[i] - Ac[i,i-1]*Ui[i-1] - Ac[i,i-nodes]*Ui[i-nodes]
- Ac[i,i-nodes-1]*Ui[i-nodes-1]= 0;

```

```

subject to Pde3 {i in vc2}: Ar[i,i]*Ur[i]+ Ar[i,i-1]*Ur[i-1]
+Ar[i,i+nodes]*Ur[i+nodes]-Ac[i,i]*Ui[i] - Ac[i,i-1]*Ui[i-1]
- Ac[i,i+nodes]*Ui[i+nodes]=0;
subject to Pde4 {i in vc4}: Ar[i,i]*Ur[i]+ Ar[i,i+1]*Ur[i+1]
+Ar[i,i-nodes]*Ur[i-nodes]-Ac[i,i]*Ui[i] - Ac[i,i+1]*Ui[i+1]
- Ac[i,i-nodes]*Ui[i-nodes]=0;
subject to Pde5 {i in vb1}: Ar[i,i]*Ur[i]
+ Ar[i,i+1]*Ur[i+1]+Ar[i,i-1]*Ur[i-1]
+Ar[i,i+nodes]*Ur[i+nodes]+Ar[i,i+nodes+1]*Ur[i+nodes+1]
-Ac[i,i]*Ui[i] - Ac[i,i+1]*Ui[i+1]-Ac[i,i-1]*Ui[i-1]
- Ac[i,i+nodes]*Ui[i+nodes] - Ac[i,i+nodes+1]*Ui[i+nodes+1]=0;
subject to Pde6 {i in vb3 union sb3 union sc3 union sc4}:
Ar[i,i]*Ur[i]+ Ar[i,i+1]*Ur[i+1]+Ar[i,i-1]*Ur[i-1]
+Ar[i,i-nodes]*Ur[i-nodes]+Ar[i,i-nodes-1]*Ur[i-nodes-1]
-Ac[i,i]*Ui[i] - Ac[i,i+1]*Ui[i+1]-Ac[i,i-1]*Ui[i-1]
- Ac[i,i-nodes]*Ui[i-nodes] - Ac[i,i-nodes-1]*Ui[i-nodes-1]=0;
subject to Pde7 {i in vb2}: Ar[i,i]*Ur[i]
+ Ar[i,i+nodes]*Ur[i+nodes]
+Ar[i,i-nodes]*Ur[i-nodes]+Ar[i,i-1]*Ur[i-1]
+ Ar[i,i-nodes-1]*Ur[i-nodes-1]-Ac[i,i]*Ui[i]
-Ac[i,i+nodes]*Ui[i+nodes]-Ac[i,i-nodes]*Ui[i-nodes]
- Ac[i,i-1]*Ui[i-1] - Ac[i,i-nodes-1]*Ui[i-nodes-1]=0;
subject to Pde8 {i in vb4}: Ar[i,i]*Ur[i]+ Ar[i,i+nodes]*Ur[i+nodes]
+Ar[i,i-nodes]*Ur[i-nodes]+Ar[i,i+1]*Ur[i+1]
+ Ar[i,i+nodes+1]*Ur[i+nodes+1]-Ac[i,i]*Ui[i]
-Ac[i,i+nodes]*Ui[i+nodes]-Ac[i,i-nodes]*Ui[i-nodes]
- Ac[i,i+1]*Ui[i+1] - Ac[i,i+nodes+1]*Ui[i+nodes+1]=0;
subject to Pde9 {i in {1..npoin} diff {allbd}}:

```



```

Ar[i,i]*Ur[i]+ Ar[i,i+nodes]*Ur[i+nodes]
+Ar[i,i-nodes]*Ur[i-nodes]+Ar[i,i-1]*Ur[i-1]+Ar[i,i+1]*Ur[i+1]
+ Ar[i,i+nodes+1]*Ur[i+nodes+1]+Ar[i,i-nodes-1]*Ur[i-nodes-1]
-Ac[i,i]*Ui[i]-Ac[i,i+nodes]*Ui[i+nodes]-Ac[i,i-nodes]*Ui[i-nodes]
- Ac[i,i-1]*Ui[i-1]-Ac[i,i+1]*Ui[i+1] - Ac[i,i+nodes+1]*Ui[i+nodes+1]
- Ac[i,i-nodes-1]*Ui[i-nodes-1]=0;    #matrix multiplication
subject to Pde10 {i in vc1}: Ac[i,i]*Ur[i]
+ Ac[i,i+1]*Ur[i+1] +Ac[i,i+nodes]*Ur[i+nodes]
+Ac[i,i+nodes+1]*Ur[i+nodes+1] +Ar[i,i]*Ui[i] + Ar[i,i+1]*Ui[i+1]
+ Ar[i,i+nodes]*Ui[i+nodes]+ Ar[i,i+nodes+1]*Ui[i+nodes+1]=bc[i];
subject to Pde11 {i in vc3}: Ac[i,i]*Ur[i]+ Ac[i,i-1]*Ur[i-1]
+Ac[i,i-nodes]*Ur[i-nodes]+Ac[i,i-nodes-1]*Ur[i-nodes-1] +Ar[i,i]*Ui[i]
+ Ar[i,i-1]*Ui[i-1] + Ar[i,i-nodes]*Ui[i-nodes]
+ Ar[i,i-nodes-1]*Ui[i-nodes-1]=bc[i];
subject to Pde12 {i in vc2}: Ac[i,i]*Ur[i]+ Ac[i,i-1]*Ur[i-1]
+Ac[i,i+nodes]*Ur[i+nodes]+Ar[i,i]*Ui[i] + Ar[i,i-1]*Ui[i-1]
+ Ar[i,i+nodes]*Ui[i+nodes]=bc[i];
subject to Pde13 {i in vc4}: Ac[i,i]*Ur[i]+ Ac[i,i+1]*Ur[i+1]
+Ac[i,i-nodes]*Ur[i-nodes]+Ar[i,i]*Ui[i] + Ar[i,i+1]*Ui[i+1]
+ Ar[i,i-nodes]*Ui[i-nodes]=bc[i];
subject to Pde14 {i in vb1}: Ac[i,i]*Ur[i]
+ Ac[i,i+1]*Ur[i+1]+Ac[i,i-1]*Ur[i-1]+Ac[i,i+nodes]*Ur[i+nodes]
+Ac[i,i+nodes+1]*Ur[i+nodes+1] +Ar[i,i]*Ui[i] +Ar[i,i+1]*Ui[i+1]
+Ar[i,i-1]*Ui[i-1] + Ar[i,i+nodes]*Ui[i+nodes]
+Ar[i,i+nodes+1]*Ui[i+nodes+1]=bc[i];
subject to Pde15 {i in vb3 union sb3 union sc3 union sc4}:
Ac[i,i]*Ur[i]+ Ac[i,i+1]*Ur[i+1]+Ac[i,i-1]*Ur[i-1]
+Ac[i,i-nodes]*Ur[i-nodes]+Ac[i,i-nodes-1]*Ur[i-nodes-1]

```

```

+Ar[i,i]*Ui[i] +Ar[i,i+1]*Ui[i+1]+Ar[i,i-1]*Ui[i-1]
+Ar[i,i-nodes]*Ui[i-nodes] +Ar[i,i-nodes-1]*Ui[i-nodes-1]=bc[i];
subject to Pde16 {i in vb2}: Ac[i,i]*Ur[i]+ Ac[i,i+nodes]*Ur[i+nodes]
+Ac[i,i-nodes]*Ur[i-nodes]+Ac[i,i-1]*Ur[i-1]
+ Ac[i,i-nodes-1]*Ur[i-nodes-1]+Ar[i,i]*Ui[i]+Ar[i,i+nodes]*Ui[i+nodes]
+Ar[i,i-nodes]*Ui[i-nodes] + Ar[i,i-1]*Ui[i-1]
+ Ar[i,i-nodes-1]*Ui[i-nodes-1]=bc[i];
subject to Pde17 {i in vb4}: Ac[i,i]*Ur[i]+ Ac[i,i+nodes]*Ur[i+nodes]
+Ac[i,i-nodes]*Ur[i-nodes]+Ac[i,i+1]*Ur[i+1]
+Ac[i,i+nodes+1]*Ur[i+nodes+1]+Ar[i,i]*Ui[i]
+Ar[i,i+nodes]*Ui[i+nodes]+Ar[i,i-nodes]*Ui[i-nodes] + Ar[i,i+1]*Ui[i+1]
+ Ar[i,i+nodes+1]*Ui[i+nodes+1]=bc[i];
subject to Pde18 {i in {1..npoin} diff {allbd}}: Ac[i,i]*Ur[i]
+ Ac[i,i+nodes]*Ur[i+nodes]+Ac[i,i-nodes]*Ur[i-nodes]
+Ac[i,i-1]*Ur[i-1]+Ac[i,i+1]*Ur[i+1]
+ Ac[i,i+nodes+1]*Ur[i+nodes+1]+Ac[i,i-nodes-1]*Ur[i-nodes-1]
+Ar[i,i]*Ui[i]+Ar[i,i+nodes]*Ui[i+nodes]+Ar[i,i-nodes]*Ui[i-nodes]
+ Ar[i,i-1]*Ui[i-1]+Ar[i,i+1]*Ui[i+1] + Ar[i,i+nodes+1]*Ui[i+nodes+1]
+ Ar[i,i-nodes-1]*Ui[i-nodes-1]=bc[i]; \ #matrix multiplication

\* THE CUBIC SPLINE define the permittivity for the wavelength variable
s.t. permr: pr = if x >= 200E-9 and x < 210E-9
then sum{i in 1..4} silr[1,i]*x^(4-i) else
if x >= 210E-9 and x < 220E-9 then sum{i in 1..4} silr[2,i]*x^(4-i) else
if x >= 220E-9 and x < 230E-9 then sum{i in 1..4} silr[3,i]*x^(4-i) else
if x >= 230E-9 and x < 240E-9 then sum{i in 1..4} silr[4,i]*x^(4-i) else
if x >= 240E-9 and x < 250E-9 then sum{i in 1..4} silr[5,i]*x^(4-i) else
if x >= 250E-9 and x < 260E-9 then sum{i in 1..4} silr[6,i]*x^(4-i) else

```

```

if x >= 260E-9 and x < 270E-9 then sum{i in 1..4} silr[7,i]*x^(4-i) else
if x >= 270E-9 and x < 280E-9 then sum{i in 1..4} silr[8,i]*x^(4-i) else
if x >= 280E-9 and x < 290E-9 then sum{i in 1..4} silr[9,i]*x^(4-i) else
if x >= 290E-9 and x < 300E-9 then sum{i in 1..4} silr[10,i]*x^(4-i) else
if x >= 300E-9 and x < 310E-9 then sum{i in 1..4} silr[11,i]*x^(4-i) else
if x >= 310E-9 and x < 320E-9 then sum{i in 1..4} silr[12,i]*x^(4-i) else
if x >= 320E-9 and x < 330E-9 then sum{i in 1..4} silr[13,i]*x^(4-i) else
if x >= 330E-9 and x < 340E-9 then sum{i in 1..4} silr[14,i]*x^(4-i) else
if x >= 340E-9 and x < 350E-9 then sum{i in 1..4} silr[15,i]*x^(4-i) else
if x >= 350E-9 and x < 360E-9 then sum{i in 1..4} silr[16,i]*x^(4-i) else
if x >= 360E-9 and x < 370E-9 then sum{i in 1..4} silr[17,i]*x^(4-i) else
if x >= 370E-9 and x < 380E-9 then sum{i in 1..4} silr[18,i]*x^(4-i) else
if x >= 380E-9 and x < 390E-9 then sum{i in 1..4} silr[19,i]*x^(4-i) else
if x >= 390E-9 and x < 400E-9 then sum{i in 1..4} silr[20,i]*x^(4-i) else
sum{i in 1..4} silr[21,i]*x^(4-i);

```

```

s.t. permc: pc = if x >= 200E-9 and x < 210E-9
then sum{i in 1..4} silc[1,i]*x^(4-i) else
if x >= 210E-9 and x < 220E-9 then sum{i in 1..4} silc[2,i]*x^(4-i) else
if x >= 220E-9 and x < 230E-9 then sum{i in 1..4} silc[3,i]*x^(4-i) else
if x >= 230E-9 and x < 240E-9 then sum{i in 1..4} silc[4,i]*x^(4-i) else
if x >= 240E-9 and x < 250E-9 then sum{i in 1..4} silc[5,i]*x^(4-i) else
if x >= 250E-9 and x < 260E-9 then sum{i in 1..4} silc[6,i]*x^(4-i) else
if x >= 260E-9 and x < 270E-9 then sum{i in 1..4} silc[7,i]*x^(4-i) else
if x >= 270E-9 and x < 280E-9 then sum{i in 1..4} silc[8,i]*x^(4-i) else
if x >= 280E-9 and x < 290E-9 then sum{i in 1..4} silc[9,i]*x^(4-i) else
if x >= 290E-9 and x < 300E-9 then sum{i in 1..4} silc[10,i]*x^(4-i) else
if x >= 300E-9 and x < 310E-9 then sum{i in 1..4} silc[11,i]*x^(4-i) else

```

```

if x >= 310E-9 and x < 320E-9 then sum{i in 1..4} silc[12,i]*x^(4-i) else
if x >= 320E-9 and x < 330E-9 then sum{i in 1..4} silc[13,i]*x^(4-i) else
if x >= 330E-9 and x < 340E-9 then sum{i in 1..4} silc[14,i]*x^(4-i) else
if x >= 340E-9 and x < 350E-9 then sum{i in 1..4} silc[15,i]*x^(4-i) else
if x >= 350E-9 and x < 360E-9 then sum{i in 1..4} silc[16,i]*x^(4-i) else
if x >= 360E-9 and x < 370E-9 then sum{i in 1..4} silc[17,i]*x^(4-i) else
if x >= 370E-9 and x < 380E-9 then sum{i in 1..4} silc[18,i]*x^(4-i) else
if x >= 380E-9 and x < 390E-9 then sum{i in 1..4} silc[19,i]*x^(4-i) else
if x >= 390E-9 and x < 400E-9 then sum{i in 1..4} silc[20,i]*x^(4-i) else
sum{i in 1..4} silc[21,i]*x^(4-i);

```

```

maximize obj: sum {i1 in sil}\\
(aEx[i1] + bEx[i1]+aEz[i1]+bEz[i1])*-pc*scl*14767.1957;

```

```

\\* Cubic spline data has been truncated

```

```

data;

```

```

param silr: 1 2 3 4 :=

```

```

1 -4.195872e-006 1.797891e-004 5.344454e-003 -3.877652e-001
2 -4.195872e-006 5.391292e-005 7.681474e-003 -3.205376e-001
3 -8.453746e-006 -7.196324e-005 7.500971e-003 -2.425274e-001
4 1.669650e-005 -3.255756e-004 3.525582e-003 -1.831678e-001
5 7.493353e-006 1.753193e-004 2.023019e-003 -1.637730e-001
6 1.031412e-006 4.001199e-004 7.777411e-003 -1.185175e-001
7 4.976405e-006 4.310622e-004 1.608923e-002 2.999579e-004
8 -1.357134e-005 5.803544e-004 2.620340e-002 2.092749e-001
9 3.257923e-005 1.732143e-004 3.373908e-002 5.157730e-001

```

```

10 -1.179722e-004 1.150591e-003 4.697714e-002 9.030645e-001
;

param silc: 1 2 3 4 :=
1 -5.336101e-006 6.467304e-004 -3.397986e-002 -2.654169e+000
2 -5.336101e-006 4.866474e-004 -2.264608e-002 -2.934631e+000
3 -1.636969e-005 3.265643e-004 -1.451396e-002 -3.117763e+000
4 5.766864e-006 -1.645265e-004 -1.289359e-002 -3.246616e+000
5 2.930984e-006 8.479475e-006 -1.445406e-002 -3.386238e+000
6 2.326190e-006 9.640899e-005 -1.340517e-002 -3.526999e+000
7 3.152028e-006 1.661947e-004 -1.077913e-002 -3.649084e+000
8 3.166349e-005 2.607555e-004 -6.509631e-003 -3.737104e+000
9 1.607032e-005 1.210660e-003 8.204526e-003 -3.744461e+000
10 8.439983e-005 1.692770e-003 3.723883e-002 -3.525279e+000

solve;

```

Appendix B:

Some details on COMSOL Multiphysics application modes used in this dissertation work follow in the spirit of helping others to build or modify optimal design codes using tools from MATLAB and COMSOL. In this work three application modes were used in the multiphysics capabilities: the electromagnetism application or TM, the parametrized geometry or PG and the sensitivity analysis (SA) using forward sensitivity or FW. The RF-module transverse magnetic (TM) application was defined as in B.1: In order to compute the gra-

Table B.1: In Plane TM mode properties

| property | value |
|-----------------------|-----------------------|
| Default element type: | Lagrange-Linear |
| Analysis Type | harmonic propagation |
| Field type | TM waves |
| specify wave using | free space wavelength |
| Frame | pg |
| Weak Constraint | OFF |
| Constraint type | ideal |

dient and define the plane wave the following constants are defined as seen in the table B.2. The functionals J_r, J_i are defined as $\langle \text{integration coupling variables} \rangle \langle \text{boundary variables} \rangle$ defined by the derived formulas for the Gaussian beam and plane wave experiments. The $\langle \text{boundary expression} \rangle$ become part of the $\langle \text{boundary integration expressions} \rangle$. COMSOL then integrates to find the values. The formulas for the objective function P and normalized PN are defined as a $\langle \text{Global Expression} \rangle$.

Table B.2: In Plane TM waves Constants

| Plane Wave parameter | value | description |
|----------------------|--|-------------------------------------|
| λ | 800 nm | wavelength |
| k_0 | $2\pi/\lambda$ | propagation constant |
| ϵ_{s2} | -27.913031524989577 - 1.507030911560426*j | permittivity of silver at 800 nm |
| HH | 2.654418729438e-3 [A/m] | magnetic field amplitude |
| ϵ_0 | 8.854187817e-12 [F/m] | vacuum permittivity |
| θ | $\pi/4$ | propagation angle |
| c_0 | 2.99792458e8 [m/sec] | speed of light |
| ω | $2 * \pi * c_0/800e-9$ | frequency angle |
| a_0 | HH*0.25/ ϵ_0 | gradient constant |

Bibliography

Bibliography

- [1] E. Yablonovitch, "Inhibited spontaneous emission in solid-state physics and electronics," *Physical Review Letters*, vol. 58, no. 20, pp. 2059–2063, 1987.
- [2] S. John, "Strong localization of photons in certain disordered dielectric superlattices," *Physical Review Letters*, vol. 58, no. 23, pp. 2486–2490, 1987.
- [3] N. R. C. C. on Nanophotonics Accessibility and Applicability, *Nanophotonics: Accessibility and Applicability*. The National Academies Press, 2008.
- [4] P. Seliger, M. Mahvash, C. Wang, and A. Levi, "Optimization of aperiodic dielectric structures," *Journal of Applied Physics*, vol. 100, p. 034310, 2006.
- [5] P. Pavaskar and S. Cronin, "Iterative optimization of plasmon resonant nanostructures," *Applied Physics Letters*, vol. 94, p. 253102, 2009.
- [6] W. R. Frei, D. A. Tortorelli, and H. T. Johnson, "Topology optimization of a photonic crystal waveguide termination to maximize directional emission," *Applied Physics Letters*, vol. 86, no. 11, 2005.
- [7] W. R. Frei, H. T. Johnson, and D. A. Tortorelli, "Optimization of photonic nanostructures," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 41-42, pp. 3410–3416, 2008.
- [8] W. R. Frei, H. T. Johnson, and K. D. Choquette, "Optimization of a single defect photonic crystal laser cavity," *Journal of Applied Physics*, vol. 103, no. 3, 2008.
- [9] V. Akcelik, G. Biros, O. Ghattas, D. Keyes, K. Ko, L.-Q. Lee, and E. G. Ng, "Adjoint methods for electromagnetic shape optimization of the low-loss cavity for the international linear collider," *Journal of Physics: Conference Series*, vol. 16, pp. 435–445, 2005. [Online]. Available: <http://iopscience.iop.org/1742-6596/16/1/059>
- [10] I. L. Gheorma, S. Haas, and A. F. J. Levi, "Aperiodic nanophotonic design," *Journal of Applied Physics*, vol. 95, no. 3, pp. 1420–1426, 2004.
- [11] L. Jiang, H. Li, W. Jia, X. Li, and Z. Shen, "Genetic optimization of photonic crystal waveguide termination for both on-axis and off-axis highly efficient directional emission," *Optics Express*, vol. 17, no. 12, pp. 10 126–10 135, 2009.
- [12] D. Bertsimas, O. Nohadani, and K. Teo, "Robust optimization in electromagnetic scattering problems," *Journal of Applied Physics*, vol. 101, p. 074507, 2007.

- [13] K. Svanberg, "Method of moving asymptotes-a new method for structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 359–373, 1987. [Online]. Available: <http://www.math.kth.se/krille/originalmma.pdf>
- [14] W. R. Frei, D. A. Tortorelli, and H. T. Johnson, "Geometry projection method for optimizing photonic nanostructures," *Optics Letters*, vol. 32, no. 1, pp. 77–79, 2007.
- [15] J. Jin, *The finite element method in electromagnetics*. John Wiley and Sons, Inc., 2002.
- [16] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*. Springer Science, 2009.
- [17] J. Reddy, *An introduction to the finite element method*. McGraw-Hill, 2005.
- [18] C. Tai, "Historical study of vector analysis," The University of Michigan, Ann Arbor, MI, Tech. Rep. RL 915, 1995.
- [19] Wikipedia. (2011, sep) Tullio levi-civita. [Online]. Available: http://en.wikipedia.org/wiki/Tullio_Levi-Civita
- [20] W. R. Frei, "Optimization of photonic crystal structures," University of Illinois at Urbana-Champaign, Urbana, IL, Tech. Rep., 2007.
- [21] J. S. Jensen and O. Sigmund, "Topology optimization of photonic crystal structures: a high-bandwidth low-loss t-junction waveguide," *J. Opt. Soc. Am. B*, vol. 22, pp. 1191–1198, 2005.
- [22] Y. Huang, "Optimal design for the grating coupler of surface plasmons," University of California at Los Angeles, Los Angeles, CA, Tech. Rep., 2008.
- [23] J. Andkjær, S. Nishiwaki, T. Nomura, and O. Sigmund, "Topology optimization of grating couplers for the efficient excitation of surface plasmons," *J. Opt. Soc. Am. B*, vol. 27, no. 9, pp. 1828–1832, 2010.
- [24] K. Kreutz-Delgado. (2011, sep) The complex gradient operator and the cr-calculus. [Online]. Available: <http://dsp.ucsd.edu/kreutz/PEI-05%20Support%20Files/complexderivatives.pdf>
- [25] I. Bardi, O. Biro, R. Dyczij-Edlinger, K. Preis, and K. Richter, "On the treatment of sharp corners in the fem analysis of high frequency problems," *IEEE Transactions on magnetics*, vol. 30, no. 5, pp. 3108–3111, 1994.
- [26] F. Hallermann, C. Rockstuhl, S. Fahr, G. Seifert, S. Wackerow, H. Graener, G. V. Plessen, and F. Lederer, "On the use of localized plasmon polaritons in solar cells," *Physica Status Solidi*, vol. 205, no. 12, pp. 2844–2861, 2008.
- [27] J. Czyzyk, M. Mesnier, and J. Moré, "The neos server," *Science and Engineering*, pp. 68–75, 1998.
- [28] W. Gropp and J. Moré, "Optimization environments and the neos server," *Theory and Optimization*, pp. 167–182, 1997.

- [29] E. Dolan, “The neos server 4.0 administrative guide,” Argonne National Laboratory, Tech. Rep. ANL/MCS-TM-250.
- [30] A. Wächter and L. Biegler, “On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [31] H. D. Mittelmann, “Decision tree for optimization software,” 2012. [Online]. Available: <http://plato.asu.edu/guide.html>
- [32] Z. Nehari, *Introduction to Complex Analysis*. Allyn and Bacon, Inc., 1961.
- [33] R. Remmert, *Theory of Complex Functions*. Springer-Verlag, 1991.
- [34] J. Lu, C. Petre, and E. Yablonovitch, “Numerical optimization of a grating coupler for the efficient excitation of surface plasmons at an ag-sio2 interface,” *J. Opt. Soc. Am. B*, vol. 24, no. 9, pp. 2268–2271, 2007.
- [35] S. Maier, *Plasmonics: Fundamentals and Applications*. Springer, 2007.
- [36] M. Mischenko, L. Travis, and A. Lacis, *Scattering, Absorption, and Emission of Light by Small Particles*. Cambridge University Press, 2002.
- [37] D. Taillaert, P. Bienstman, and R. Baets, “Compact efficient broadband grating coupler for silicon-on-insulator waveguides,” *Optics Letter*, vol. 29, no. 23, pp. 2749–2751, 2004.

Curriculum Vitae

Carmen Caiseda was born to Puerto Rican parents on 1961 in Detroit, Michigan. Being among the first graduates of the recently established School of Tropical Medicine of the University of Puerto Rico, her parents were selected to complete their internship in Michigan. After the year of internship they returned to PR and raised a family of four children while working for the public health department.

Carmen completed High School at Colegio San Antonio, Rio Piedras, PR in 1979. Never sure of what to study but being an enthusiastic student, she started undergraduate studies on Social Sciences at the University of Puerto Rico, transferred to Georgetown University, Washington D.C. as a Philosophy major, Computer Science minor, and returned to the University of Puerto Rico (UPR) to study Mathematics. After the tragic loss of her youngest brother in 1986, Carmen found in the Christian faith the source of hope, guidance and courage needed to meet the daily challenges. Now married and with two sons, Ernesto (5 y/o) and Gabriel (3 y/o), she completed a B.S. in applied mathematics at UPR in 1987. Hoping to provide to the best of her ability for her children she went on to complete an M.S. in applied mathematics at UPR in 1991. She became a mathematics assistant professor at the Inter American University of Puerto Rico, Bayamón (IAUPR), and dedicated 15 years to teaching and mentoring students.

The M.S. thesis on *Characterization, enumeration and generation of pseudo-Boolean functions for 5 and 6-cubes* under Dr. Reza Emamy was important in her development as a computational scientist. Programming in FORTRAN77 FX/Alliant super computer with remote access with a bulky small screen monitor was then the latest technology, after using punched card readers for most of her undergraduate homework. At the IAUPR she continued work in the visualization of cut-complexes of the hypercube using MATHEMATICA. Here she contributed mostly to educational projects that implemented strategies to increase retention of students with remedial mathematics needs, through tutoring, mathematics laboratory activities, and development of interdisciplinary activities.

In 2007 she was called to transfer to Fairfax VA because of the need of a friend whose 6 y/o boy would go on to fight cancer. The friend herself faced cancer before her young son relapsed in 2011 and underwent a bone marrow transplant. Miraculously son and mother are cancer free today. Committed to continue her professional development, Carmen received this as a challenge and an opportunity to help and be helped, as she was able to pursue a PhD in computational sciences from George Mason University.