# ABSTRACT

VOGT, RYAN HUBERT. Interface Problems and Binary Electromagnetic Cloaking Designs in Computational Electromagnetics. (Under the direction of Zhilin Li.)

This thesis studies two type of problems in the area of computational electromagnetics. The first topic is numerically solving Maxwell's equations with discontinuous permittivity and permeability across a material interface. We develop convergent finite-difference methods using the immersed interface method (IIM) on Cartesian meshes. Our method takes into the account of the jump conditions from the equations so that the scheme maintains second-order accuracy in time and space. We design methods for Maxwell's equations in one and two dimensional space. For the one dimensional problem we introduce two separate methods: a direct approach that uses the jump conditions directly, and an alternating direction implicit method that takes advantage of augmented variables. We numerically show the consistency, stability, and convergence for our methods. The second topic is constructing effective binary electromagnetic cloaking designs in three different instances. All of these problems can be written as a mixed-integer partial-differential equation constrained optimization (MIPDECO) problem. The goal of solving the optimization problem is to find a cloaking design that minimizes the scattering in a predefined region of space. We define a subregion of the domain which we divide into a control mesh, made up of subdomains. On each of these subdomains, we define a binary variable. The value of this binary variable determines if we place material at a given subdomain location or not. The constraint on the optimization problem is that the scattering wave must satisfy Maxwell's equations with a Robin boundary condition. Because our binary cloak design is not time dependent, we convert the time domain Maxwell equations into their frequency counterpart. The resulting constraint equation then becomes a Helmholtz equation. To solve this optimization problem we derive an optimize-then-discretize approach. We first derive the necessary conditions of optimality, which gives us a gradient, then we apply a trust-region method to solve the problem. We present numerical results that show that our method produces effective electromagnetic cloaks. We takes these steps for each instance of the problem. The first instance we study is to make 2D binary cloaks with a single material to cloak a region in 2D space. In the second case, we generalize the results from the first case, allowing for a binary cloak to be made out of several different materials. In the third and final case we construct binary designs in 3D to cloak a region in space.

# Interface Problems and Binary Electromagnetic Cloaking Designs in Computational Electromagnetics

by
Ryan Hubert Vogt

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2020

APPROVED BY:

Alen Alexanderian

David Aspnes

Pierre Gremaud

Sven Leyffer
External Member

Zhilin Li
Chair of Advisory Committee

# DEDICATION

I would like to first and foremost dedicate this thesis to my parents Hubert and Anne Vogt; their support and love is the reason why I have been successful in academia and in life.

I would like to also dedicate this thesis to my entire family. I had a great environment to grow up in as a child and that is because I have such a great family. My grandmothers, Rita Hills and Dorthy Vogt, have made a lasting impact on my life. My capacity to be kind and compassionate is due to them. I would also like to thank my Aunt Brenda Vogt, who has been one of my biggest supporters. Being reminded through graduate school with an e-mail that she was thinking about me always ment a lot. I would also like to thank my cousin, Matt West, for calling now and again - I really enjoyed our conversations.

I would like to thank all of my teachers at Wayland-Cohocton Central School, in Wayland, New York, that made me the person I am today (including elementary, middle and high school). While in high-school I participate in the boys varsity swimming, indoor-track, cross-country, and outdoor-track. These programs taught me several hard lessons (in a good way). I can say without reservation that I wouldn't be the person I am today without the investments of Coach Matt Bondgren, Coach Jeff Giordano, Coach Robert Hughes, Coach Jay Kaczmarek, Coach Dan Moore, Coach Rich Saxton, and Coach David Stewart. In particular, the influence of Coach Kaczmarek and Coach Moore were important to my development, which I would like to highlight. Coach Kaczmarek taught me (and many others around my age) the importance of personal responsibility, and failure (which we famously referred to as the F word). He always had our best interests at heart. I know many of us would have been lost without his guidance. I didn't enjoy school very much, but I would always enjoy being in his class. We all know being a public school teacher can be, well on most days, a thankless job. I wanted to particularly mention him because the extra effort he put into me, and others, has made a major impact on my life and the lives of others. We had no idea how lucky we were to have a teacher that seriously cared five-hundred times more than he should have about us. He didn't get paid anymore for doing this, but he still did it anyway. It is unfortunate it took me until adulthood to realize this. I can say without reservation that I would not have been able to obtain a Ph.D in Mathematics if it weren't for the lessons about failure (there were many I promise) that Coach Kaczmarek gave in AP Government, and in general interactions. Even though those lessons were many years ago, I still reflect on them today. Coach Moore indirectly started me on this almost decade long journey. Coach Moore was my running coach in high school and taught me to love running. Running taught me many life lessons that I believe I would have not learned anywhere else. These same life lessons gave me the skills to become strong in mathematics. Even though I was the slowest member on the cross country team (**PROUDLY** the first place loser always!) he always spent time running with me (one-on-one) for well over 500 miles. He taught me many lessons. First, no child is ever lost. Second, the most important and selfless investment one can make is helping children figure out who they want to be, and sharing your expertise with them. Finally, the importance of kindness toward others and the importance of selflessness. This journey I have participated in for the last decade in academia is entirely due to the selflessness and investments of Coach Kaczmarek and Coach Moore. They have had a life long lasting impact on me. I hope someday when I "grow up" I can make the same impact on kids lives that Coach Kaczmarek and Coach Moore have made on my life. Every great oak tree started beneath a mother oak (well in this case two).

I am grateful to my previous professors at Finger Lakes Community College and Rochester Institute of Technology for developing my interests in Computer Science and Mathematics; the time they spent with me was crucial to my intellectual development. I would first like to thank Professor Timothy Biehler, Professor Charles Hoffman and Professor Ronald Metzger of Finger Lakes Community College for fostering my original interest in mathematics. I would also like to thank my Cross Country coach at FLCC, Coach Rob Wink, for making running fun! In particular both Professor Biehler and Professor Hoffman invested a lot of time talking to me about mathematics. I am forever indebted for this small investment, which led to large many consequences on my life. Professor Metzger taught me differential equations, taking differential equations was the first experience I had in math that made me extremely interested in the subject. It comes as no surprise that this original interest propelled me towards studying partial-differential equations at RIT and NCSU. I sometimes wonder if I didn't have a positive experience taking this course with him if I where I am today. I do not believe so.

At Rochester Institute of Technology, I would like to thank Dr. Anthony Harkin, Dr. David Ross, Dr. Chris Wahle, and Dr. Tamas Wiandt for sharing their expertise in mathematics with me over several years; their mentoring caused me to appreciate mathematics a great deal. I took my first course on partial-differential equations with Dr. Harkin as the professor. Because I had such a positive experience at FLCC doing ordinary differential equations it motivated me to take this course. This course was also an extremely positive experience, and a large driving force towards my interest of pursing PDE as a research topic. Dr. Wahle also taught me, in independent study, partial differential equations, and in another separate independent study perturbation theory. I learned a great deal of about applied mathematics, and its importance, while listening to him. His easy going attitude and humor when it came to mathematics was another driving force that made me want to study the subject more. Dr. Ross, like Dr. Harkin and Dr. Wahle, also spent a great amount of time mentoring me in the subject of partial-differential equations, as well as other subjects like numerical methods for PDE and dynamical systems. His door was always open to me, and he helped me many times on the white board in his office on math problems. He taught me the importance of rigor and theory in mathematics, in addition to the importance of taking the time to mentor young minds in the subject. Dr. Wiandt was the only professor in this list to not discuss partial-differential equations with me, but that does not mean he was any less important to my development. After taking partial-differential equations I was considering picking up a math major, because I thought maybe I could be successful at mathematics, but I was still on fence. In Spring 2016, I took complex variables with Dr. Wiandt. This course solidified my interest in mathematics, and I haven't looked back since. I also learned differential geometry (which turned out to be extremely useful in graduate school) from him. Like Dr. Wahle, Dr. Wiandts easy going attitude and humor made me really interested in mathematics.

I would like to lastly dedicate this thesis to Snoop Dogg for giving the world these wise words about mathematics and its importance today: "If you stop at general math, you're only going to make general math money".

# BIOGRAPHY

Ryan Hubert Vogt graduated high school from Wayland-Cohocton Central School in January 2011. He started studying computer science and engineering science at Finger Lakes Community College where he graduated with associate of science degrees in computer science and engineering science in May 2013. Next he attended Rochester Institute of Technology, where he was awarded Bachelor of Science degrees in Applied Mathematics and Computer Science in May 2015.

Ryan earned master degrees in Applied Mathematics and Operations Research in December 2019. During his graduate career he was a Given's Associate at Argonne National Laboratory during the summers of 2018 and 2019. In addition he has been a research assistant to Dr. Sven Leyffer and Dr. Todd Munson since August 2018. At Argonne National Laboratory he has worked on Mixed-Integer Partial Differential Equation Optimization Theory and Optimal Control Theory.

# ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Dr. Zhilin Li, for his kindness, advice and friendship during the construction of my thesis. I really enjoyed conducting research under his guidance. Dr. Li brought me to the forefront of understanding numerical methods for interface problems, among other topics in numerical analysis and partial-differential equations. I am forever indebted to him for the time and energy he has invested in the development of my thesis. I look forward to continuing collaboration with my adviser on future projects related to the contents of this thesis.

In addition, I would like to take the opportunity to thank each member of my graduate committee at NCSU: Dr. Alen Alexanderian, Dr. David Aspnes and Dr. Pierre Gremaud. Dr. Alexanderian was important for developing my interest in partial-differential equation constrained optimization theory and inverse problems. The experience and expertise he gave to me led to me conducting research in this area at Argonne National Laboratory. He taught me how theory is important to solve problems of practical importance. Dr. David Aspnes has been my mentor in Electrodynamics for several years. His mentorship has made me appreciate the differences between the scientist and mathematician, and how I should strive to be both at the same time. I cannot thank him enough for his constant encouragement of my questions and thoughts when it came to electrodynamics. He taught me the importance of interdisciplinary communication and how this is key to doing not only good math, but also good science. Every time I solve a problem, I always start with “the basics” and this is because of him. Lastly, but surely not least, Dr. Gremaud has been a critical part of my graduate school experience and growth. He has always had his door open and made time to answer my questions, and help me get a little bit better at math every day. His influence has made me mature in many ways over the five years of my Ph.D studies. He has made me appreciate the fields of numerical analysis and partial differential equations a great deal. He taught me the importance of helping others who want to learn.

Moreover, I would like to thank all the facility, students, and staff in the math department for their kindness towards me during my time in the program. I would like to thank my office mate, Dr. Joey Hart (currently a mathematician at Sandia National Laboratory), for many helpful and interesting discussions about mathematics. I would of not of pursed a position at Argonne National Laboratory without his encouragement. A great deal of my success over the past few years is due to him taking the time to explain the national lab system and why it are a great place to be. I learned a lot by listening to him. I would like to thank all my professors at NCSU for their time and patience: Dr. Lorena Bociu, Dr. Stephen Campbell, Dr. Shu-Cherng Fang, Dr. Yahya Fathi, Dr. Michael Kay, Dr. C.T Kelley, Dr. Yunan Liu, Dr. Robert Martin, Dr. Maria Mayorga, Dr. Negash Medhin, Dr. Carl Meyer, Dr. Mette Olufsen, Dr. Tom Reiland, Dr. Jesus Rodriguez, and Dr. Ralph Smith. I learned a lot about mathematics and myself from listening to them.

I would like to lastly thank Dr. Sven Leyffer and Dr. Todd Munson of Argonne National Laboratory (ANL) for their mentorship during my Ph.D studies. I was a Given's associate and research assistant over a two year period due to the generosity of Dr. Leyffer and Dr. Munson. Dr. Leyffer was kind enough to be an external committee member for my thesis. Dr. Leyffer reviewed my thesis thoroughly and gave me important feedback. He gave me key feedback that helped me improve the quality of my thesis substantially. My scientific writing skills and love for optimization theory grew greatly under his guidance. Dr. Munson, even though not a committee member, has had a major impact on my thesis. Dr. Munson also thoroughly read through my thesis and gave me critical feedback on my writing and ideas. His encouragement through the past two years has had a massive positive impact on the math I have produced in this thesis. Chapters 4, 5 and 6 of this thesis is a direct result of my work at ANL. I am grateful to both Dr. Leyffer and Dr. Munson for giving me experience at a national lab. The time I spent at Argonne exposed me to many forefront problems in optimization theory and partial-differential equation constrained optimization. The experience also taught me about a range of important problems in STEM that are being solved in the Department of Energy, and its role in propelling STEM discoveries. I learned a great deal from listening to them, and I hope I can make major contributions to optimization theory and numerical optimization as they have someday. I will never forget their kindness towards me. I also would like to note that this work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program through the FASTMath Institute under Contract DE-AC02-06CH11357 at Argonne National Laboratory. This work was also supported by the U.S. Department of Energy through grant DE-FG02-05ER25694.

## TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

CHAPTER

# 1

# INTRODUCTION

In this thesis we discuss two topics in the field of electromagnetics. The first topic that we study is to solve Maxwell's equations numerically with discontinuous permittivity and permeability across a material interface. We develop second order, stable, finite difference schemes using the immersed interface method (IIM) [64]. We numerically show that our schemes are consistent, stable and convergent in both one and two dimensions. The second topic that we study is the development of binary cloak designs. We use the Haslinger model [41] as the foundation to formulate a mixed-integer partial-differential equation constrained optimization (MIPDECO) problem for the electromagnetic cloaking design. We introduce a trust-region method for solving the MIPDECO.

An overview of the thesis is as follows: we first give an overview of the different subjects that appear in this thesis; these include, classical electromagnetics, interface problems in electromagnetics, electromagnetic cloaking design, partial differential equation constrained optimization, mixed-integer nonlinear programming, and mixed-integer partial differential equation optimization. Next we give a brief summary of the contributions of this thesis. In Chapter 2, we present several finite-difference schemes for solving Maxwell's equations in one dimension when a material interfaces are present. We conclude the study of each scheme presented with a test problem to illustrate the success of our methods. In Chapter 3 we introduce an augmented IIM method [64] to solve Maxwell's equations in one and two dimensional space; like Chapter 2 we study the stability, consistency, and convergence of our method. We introduce numerical results to show that our method works for 2D curved interfaces while also maintaing second order accuracy in space and time. In chapter 4, we introduce a two dimensional model for designing electromagnetic cloaks made out of a single non-magnetic material with the intent of making a predefined area of space optically invisible; the resulting problem is a MIPDECO. We then introduce a trust-region steepest-descent method, using an adjoint-based gradient, to solve the electromagnetic cloak problem numerically. Next, in chapter 5 and chapter 6 we introduce two variants of the problem. The first variant that we study is the construction of an electromagnetic cloak with several different materials under a mass constraint. We study the impact that mass and materials have on the impact of the electromagnetic cloaking capability. We introduce continuation methods to improve cloaking capability of the multiple material cloaks in an iterative way. The second variant is generalizing the single material model to three dimensional space and solving the resulting problem. In chapter 7 we describe several different directions the work shown in this thesis can be extended.

## 1.1 Classical Electromagnetics

We begin with reviewing classical electromagnetics, which is the basis for all work in this thesis.

### 1.1.1 Maxwell's Equations

The fundamental equations of electromagnetics, in the time domain. were first presented in their entirety by James Clark Maxwell [74]:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon} \qquad \text{(Gauss's Law)} \qquad (1.1)$$

$$\nabla \cdot \mathbf{B} = 0 \qquad \text{(Gauss's Law For Magnetism)} \qquad (1.2)$$

$$\nabla \times \mathbf{E} = \frac{\partial \mathbf{B}}{\partial t} \qquad \text{(Maxwell Faraday Equation)} \qquad (1.3)$$

$$\nabla \times \mathrm{B} = \mu\left(\mathbf{J} + \epsilon \frac{\partial \mathbf{E}}{\partial t}\right) \qquad \text{(Amp\`{e}re's Law).} \qquad (1.4)$$

These four equations establish the relationship between the electric field, **E** and the magnetic field **B**. We will denote $\mathbf{E} = (E^x, E^y, E^z)$ and $\mathbf{B} = (B^x, B^y, B^z)$. Each component of the field can be dependent on all spatial variables **x** and the scalar time variable t. In the equation above, the quantity $\rho := \rho(\mathbf{x}, t)$ represent the electromagnetic charge density that exists at the vector position **x** at time $t$, the quantity $\mathbf{J} := \mathbf{J}(\mathbf{x}, t)$ denote the electromagnetic current density that exists at the vector position **x** at time $t$. The material properties $\epsilon := \epsilon(\mathbf{x}, t)$ and $\mu := \mu(\mathbf{x}, t)$ denote the electric permittivity and magnetic permeability, respectively. Both measure how a material stores electric and magnetic energy respectively. A given materials permittivity and permeability are summarized by the material response at a given frequency $\omega$,

$$\epsilon := \epsilon(\omega) = \epsilon_r(\omega)\epsilon_o \tag{1.5}$$

$$\mu := \mu(\omega) = \mu_r(\omega)\mu_o. \tag{1.6}$$

The quantity $\epsilon_o$ represents the permittivity of free space, with $\epsilon_o \approx 8.8 \times 10^{-12}$ Farad per meter and $\mu_o$ represents the permeability of free space,with $\mu_o \approx 1.3 \times 10^{-12}$ Farad per meter. $\epsilon_r$ and $\mu_r$ denote a materials relative permittivity and permeability respectively.

We consider only linear media for the purpose of this thesis. In addition we will assume that materials of interest cannot be polarized and or magnetized.

We also introduce two additional fields: the displacement field **D**, and the magnetizing field **H**. We define these fields as

$$\mathbf{D} = \epsilon \mathbf{E}$$

$$\mathbf{H} = \frac{1}{\mu}\mathbf{B},$$

and in this work we assume that there is neither a polarization or magnetization field present in the physical system.

Each of Maxwell's Equations describe important physical phenomena found in electromagnetics [7]. When numerically solving Maxwell's equations only (1.3) and (1.4) need to be considered. Maxwell's first two equations can easily be derived by using charge conservation arguments.

We briefly show this, first deriving Gauss's Law of Magnetism from the Maxwell Faraday equation. We first take the divergence of both sides of (1.3),

$$\nabla \cdot \nabla \times \mathbf{E} = -\frac{\partial \left(\nabla \cdot \mathbf{B}\right)}{\partial t}. \tag{1.7}$$

Using the vector identity that the divergence of the curl of a vector field with sufficient regularity yields

$$-\frac{\partial \left(\nabla \cdot \mathbf{B}\right)}{\partial t} = 0. \tag{1.8}$$

However this implies that

$$\nabla \cdot \mathbf{B} = f(\mathbf{x}), \tag{1.9}$$

where $f$ is some arbitrary function with sufficient regularity in time. Two arguments can be made to enforce that $f = 0$. The first is a classical argument, which states that there has been no experimental observation of a magnetic monopole. This directly implies the divergence of the magnetic field is zero [50]. Another argument, which is consistent with the classical argument, comes from quantum mechanics [16, 57]. This argument predicts the existence of magnetic monopoles due to the big-bang. However their density has rapidly decreased due to the rapid expansion of space, becoming negligible. This theoretical prediction enforces that $f = 0$.

Next we discuss the derivation of Gauss's Law from Amp$\grave{e}$re's Law. We first take the divergence of both sides of equation (1.4),

$$\nabla \cdot \nabla \times \mathrm{B} = \nabla \cdot \left( \mu \left( \mathbf{J} + \epsilon \frac{\partial \mathbf{E}}{\partial t} \right) \right). \tag{1.10}$$

Using the divergence applied to a curl identity we obtain

$$\nabla \cdot \mathbf{J} + \frac{\partial \left( \epsilon \nabla \cdot \mathbf{E} \right) )}{\partial t} = 0. \tag{1.11}$$

We now assume that conservation of charge holds on the charge density function $\rho$, which means

$$\nabla \cdot \mathbf{J} + \frac{\partial \rho}{\partial t} = 0. \tag{1.12}$$

When comparing equations (1.11) and (1.12), we observe for both equations to be true simultaneously that

$$\epsilon \nabla \cdot \mathbf{E} = \rho + C, \tag{1.13}$$

where $C$ is some arbitrary constant. The equation is equivalent to

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon} + \frac{C}{\epsilon}. \tag{1.14}$$

Because we can always define the charge density relative to our definition of ground, we can enforce $\frac{\rho}{\epsilon} := \frac{\rho}{\epsilon} + \frac{C}{\epsilon}$, which gives us Gauss's Law.

When solving (1.3) and (1.4) with a numerical method we cannot guarantee we satisfy (1.1) and (1.2). We therefore will favor methods that solve (1.3) and (1.4) accurately but also enforce conservation of electric and magnetic as written in (1.1) and (1.2). In chapter 2, we will solve (1.3) and (1.4) under the assumption that material permittivity, $\epsilon$, and the material permeability, $\mu$, are discontinuous functions. This is to say that we are interested in observing the behavior of electromagnetic waves as they move through several homogeneous materials in a physical domain.

### 1.1.2 The Electromagnetic Wave Equation

Another important version of Maxwell's equations is when the electric and magnetic field are written in the time domain wave equation form. We begin by showing from (1.3) and (1.4) that we can derive wave equations for the magnetic and electric field. We first take the curl of (1.4) and (1.3):

$$\nabla \times \left(\nabla \times \mathbf{E}\right) = \frac{\partial\left(\nabla \times \mathbf{B}\right)}{\partial t} \tag{1.15}$$

$$\nabla \times \left(\nabla \times \mathrm{B}\right) = \mu\left(\nabla \times \mathbf{J} + \epsilon \frac{\partial\left(\nabla \times \mathbf{E}\right)}{\partial t}\right). \tag{1.16}$$

We will use the vector identity, for a general differential vector field $\mathbf{F} = \left(F_1, F_2, F_3\right)$,

$$\nabla \times \left(\nabla \times \mathbf{F}\right) = \nabla\left(\nabla \cdot \mathbf{F}\right) - \Delta \mathbf{F}, \tag{1.17}$$

where $\Delta$ is the element wise Laplacian operator,

$$\Delta \mathbf{F} = \begin{bmatrix} \Delta F_1 \\ \Delta F_2 \\ \Delta F_3 \end{bmatrix} = \begin{bmatrix} \nabla \cdot \left(\nabla F_1\right) \\ \nabla \cdot \left(\nabla F_2\right) \\ \nabla \cdot \left(\nabla F_3\right) \end{bmatrix}. \tag{1.18}$$

We then write the final form of Maxwell's equations in their electromagnetic wave equation form,

$$\Delta \mathbf{E} = \frac{1}{c^2}\frac{\partial^2 \mathbf{E}}{\partial t^2} + \mu \frac{\partial \mathbf{J}}{\partial t} + \nabla\left(\frac{\rho}{\epsilon}\right) \tag{1.19}$$

$$\Delta \mathbf{B} = \frac{1}{c^2}\frac{\partial^2 \mathbf{B}}{\partial t^2} - \mu\left(\nabla \times \mathbf{J}\right), \tag{1.20}$$

where $c = \frac{1}{\sqrt{\epsilon\mu}}$ is the speed of light [50].

One can solve either the classical form of Maxwell's Equations or solve the electromagnetic wave equations to find the electric and magnetic fields.

### 1.1.3 Helmholtz Equation

In subsections 1.1.2, we have written the equations in the time domain. Another important frame of reference of these equations are in the frequency domain. This frame of reference is important both from a practical and mathematical perspective [17, 26, 94, 101, 111] . In this thesis, our motivation of using the frequency domain version of Maxwell's equations is to determine how, for a fixed non-magnetic material and electromagnetic frequency, we construct a cloak that is successful at optical cloaking.

We show how one derives the Helmholtz equation, which is just the wave equation in the frequency domain. If we solve (1.19) and (1.20) under the assumption of zero current density, $\mathbf{J} = \mathbf{0}$, and zero charge density, $\rho = 0$, then any equation in the system can be written as

$$\Delta v = \frac{1}{c^2}\frac{\partial^2 v}{\partial t^2}, \tag{1.21}$$

where $v$ is a scalar function.

To transfer the homogeneous wave equation to the frequency domain, we apply the Fourier transformation to both sides:

$$\int_{-\infty}^{\infty} \Delta v e^{i\omega t} dt = \int_{-\infty}^{\infty} \frac{1}{c^2}\frac{\partial^2 v}{\partial t^2} e^{i\omega t} dt. \tag{1.22}$$

where $i = \sqrt{-1}$ is the unit imaginary number and $\omega$ is a given frequency. We will define the Fourier Transformation of $v(\mathbf{x}, t)$ in the time domain to its frequency domain counter part $u(\mathbf{x}, \omega)$ as

$$u(\mathbf{x}, \omega) = \int_{-\infty}^{\infty} v(\mathbf{x}, t) e^{i\omega t} dt. \tag{1.23}$$

Conducting the contour integral in (1.22) on the complex plane [71] yields

$$\Delta u + \omega^2 u = 0. \tag{1.24}$$

The equation in (1.24) is known as the Helmholtz equation. A modified version of this equation will appear in chapter 4, as a constraint to a special instance of a partial differential equation optimization problem.

## 1.2 Interface Problems in Electromagnetics

In Chapter 2, we focus on the development of efficient numerical methods and analysis for the Maxwell's equations, in particular, we focus on electromagnetic scattering in heterogeneous media.

The first topic is to solve and analyze Maxwell's equations that models electromagnetic waves in heterogeneous media. We start with the one dimensional Maxwell's equations that can be decomposed into two independent sets of equations, the transverse magnetic (TM) mode and the transverse electric (TE) mode [14, 47, 50, 62]. The focus is to solve the one dimensional problem and then ultimately solve the two dimensional problem. The one dimensional TE and TM equations are

$$\begin{aligned} \frac{\partial H^y}{\partial t} &= \frac{1}{\mu}\frac{\partial E^z}{\partial x} \\ \frac{\partial E^z}{\partial t} &= \frac{1}{\epsilon}\frac{\partial H^y}{\partial x} \end{aligned} \tag{1.25}$$

$$\begin{aligned} \frac{\partial E^y}{\partial t} &= -\frac{1}{\epsilon}\frac{\partial H^z}{\partial x} \\ \frac{\partial H^z}{\partial t} &= -\frac{1}{\mu}\frac{\partial E^y}{\partial x}, \end{aligned} \tag{1.26}$$

We are interested in the case when the electromagnetic waves pass through different materials as illustrated in the diagram for a two dimensional problem, as ween in Fig. 1.1 which were shown in [82, 87]. For design and cloaking purposes, we assume that the interface between two media is sharp in contrast to phase field models. At an interface between two media, say medium 1 and medium 2, the interface conditions can be expressed as

$$\mathbf{n}\times(\mathbf{E}_1-\mathbf{E}_2)=0, \quad \mathbf{n}\times(\epsilon_1\mathbf{E}_1-\epsilon_2\mathbf{E}_2)=0, \tag{1.27}$$

$$\mathbf{n}\times(\mathbf{H}_1-\mathbf{H}_2)=0, \quad \mathbf{n}\times\left(\mu_1\mathbf{H}_1-\mu_2\mathbf{H}_2\right)=0, \tag{1.28}$$

denoted as

$$[\mathbf{n}\times\mathbf{E}]=0, \quad [\mathbf{n}\times\epsilon\mathbf{E}]=0, \tag{1.29}$$

$$[\mathbf{n}\times\mathbf{H}]=0, \quad \left[\mathbf{n}\times\mu\mathbf{H}\right]=0, \tag{1.30}$$

where **n** is the outward unit vector perpendicular to the interface. The representations to the right are called *jump* conditions. We will use standard approaches for dealing with outer boundary conditions such as perfect electric conductors (PEC), or absorbing boundary conditions (BCs), or perfectly matched layer (PML) techniques, but focus on our effort on the treatment of interfaces. When electromagnetic (EM) waves reach an interface characterized by an abrupt change in physical properties, portions of the signal undergo reflection, refraction and transmission which makes it possible to design the wave patterns for cloaking purposes.

For the purposes of the 2D problem, we will focus on solving the TM equation using an augmented immersed interface method [64]:

$$\frac{\partial H^x}{\partial t} + \frac{1}{\mu}\frac{\partial E^z}{\partial y} = 0, \tag{1.31}$$

$$\frac{\partial H^y}{\partial t} - \frac{1}{\mu}\frac{\partial E^z}{\partial x} = 0, \tag{1.32}$$

$$\frac{\partial E^z}{\partial t} + \frac{1}{\epsilon}\left(\frac{\partial H^x}{\partial y} - \frac{\partial H^y}{\partial x}\right) = 0. \tag{1.33}$$

Our method is such that there are minor differences between solving the 2D TM and TE equation, therefore we will focus on the TM equations for this work.

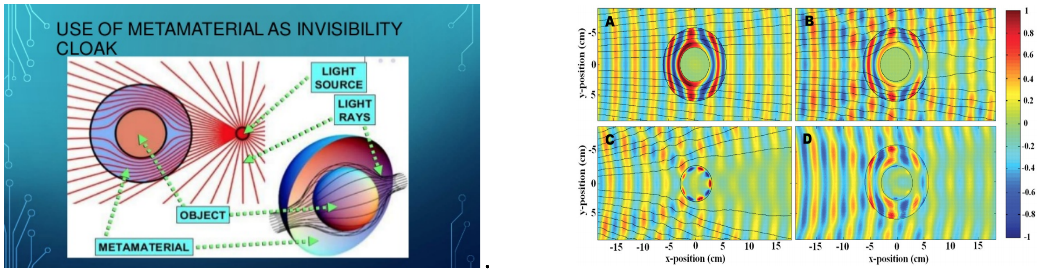


**Figure 1.1** A diagram of cloaking technique and electromagnetic wave scattering in different media. The images can be seen in [82, 87] respectively.

**Yee's Scheme.** We plan to develop efficient dimensional jump splitting methods for solving Maxwell's equations. Our approach will be based on Yee's algorithm [105] and the direct and augmented strategy [64]. Yee's algorithm uses a staggered Cartesian mesh and has a number of significant properties: (1) It is second order accurate both in space and time without using second order partial derivatives as in the Lax-Wendroff scheme (2): the CFL condition is favorable $\Delta t \sim Ch$ (3) Yee's scheme is a time-domain technique, and when a broadband pulse (such as a Gaussian pulse) is used as the source, then the response of the system over a wide range of frequencies can be obtained with a single simulation. This is useful in applications where resonant frequencies are not exactly known, or anytime that a broadband result is desired (4) it allows us to specify the material at all points within the computational domain. A wide variety of linear and nonlinear dielectric and magnetic materials can be naturally and easily modeled (5) Yee's scheme allows the effects of apertures to be determined directly. Thus shielding effects can be found, and the fields both inside and outside a structure can be found directly or indirectly (6) Yee's scheme solves the electric and magnetic fields directly so it is convenient that no conversions must be made after the simulation has run to get these values, like the wave equations approach. We also propose to develop a dimensional jump splitting approach for Helmholtz equations that can pin down particular wave numbers for particular material designs. One of the challenges naturally is how to maintain the same order of accuracy and stability when waves pass through physical interfaces between different materials.

**Why structured (such as Cartesian or adaptive Cartesian) meshes?** There are a variety of methods for solving Maxwell's equations. Finite-element methods have been developed particularly for frequency domains using Helmholtz equations, which are elliptic PDEs for which FEM and theory are mature. A body-fitted mesh coupled with a finite element formulation is a natural choice for elliptic PDEs and Helmholtz equations with interfaces and has been extensively studied in the literature. In addition the body-fitted approach is easy to analyze and often leads to optimal convergence. However, for arbitrary interfaces with complicated geometries, generating a satisfactory body fitting mesh is often time consuming. Such a difficulty becomes even more severe for moving interface or free boundary problems in which the mesh has to be generated again and again in the time stepping procedure. Note that the primitive Maxwell's equations are a first order hyperbolic system that can be solved using a finite-difference time-domain (FDTD) method, such as the simple Yee's scheme rather easily. Even in the frequency domain using the Helmholtz equations, a fast Helmholtz solver, e.g., the one from Fishpack et al [3] can be utilized, but only on structured meshes.

For moderately complex geometries, it is unnecessary to apply unstructured meshes and domain decomposition techniques. For complicated geometries, methods have been developed for adaptive Cartesian meshes for various problems, see [67, 68]. There are limited sharp interface methods using Cartesian meshes for hyperbolic interface problems, see, for example, [107], but even fewer for Maxwell's equations with discontinuous permittivity and permeability until recently [98, 106] for straight interfaces, the derivative matching method in [109, 110], and the immersed interface method [23]. In these methods, the finite difference stencil are usually enlarged and the coefficients are determined from a small to medium size of linear system of equations (36 grid points in [23] for 2D problems). As a result, smaller Courant Friedrichs Lewy (CFL) numbers [60] are needed. These methods are not practical in general for optimal controls since forward solvers are too expensive.

**Why a dimensional jump splitting approach in 2D?** Note that discretizations in one spatial dimension are much easier and more efficient not only in terms of implementations, but also in convergence analysis. This is one of the motivations to introduce a dimension by dimension approach for 2D & 3D. In addition, parallel computing can be done more easily in each coordinate dimension. One challenge is how to deal with the discontinuities at the interface that are not in the coordinate directions. We introduce the use of the dimensional jump splitting strategy by introducing one or several jump variables (in the coordinate directions) whose dimensions are at least one order lower than that of the primitive variables. Note that while augmented strategies have been applied to various problems [24, 46, 49, 51, 69], the dimensional jump splitting approach is new. Applications, including the scattering problems using Maxwell's equations and the related optimal control, are new. The dimensional jump variables are from the solution and/or its gradient and will not affect the wellposedness of the problem. Advantages of our proposed approach to Maxwell's equations with discontinuities include: (1): Yee's compact finite difference remains the same with modified right hand sides. Thus we maintain the simplicity and the structure of the FDTD method; (2): The CFL condition remains at least asymptotically the same; (3): For fixed interface and time step size, the coefficient matrix for the augmented jump variables is a constant matrix thus it needs to be computed just once. For moving interfaces or adaptive time steps, the GMRES or rank one or two updating methods can be utilized since the change is relatively small; (4): It is relatively easier to carry out the convergence analysis since the finite difference stencil and coefficients do not change.

## 1.3 Mixed-Integer Partial-Differential Equation Constrained Optimization and Electromagnetic Cloaking Problems

In chapter 4, 5 and 6 we solve a mixed-integer partial-differential equation constrained optimization problem (MIPDECO) to build effective electromagnetic cloak designs. We first present an overview of electromagnetic cloaking on the experimental and theoretical front. Afterwards we present an overview of PDE constrained optimization, mixed-integer programming and MIPDECO.

**Electromagnetic Cloaking:** In our application, electromagnetic cloaking, we focus on manipulating the light around an object to make the object nearly optically invisible. Since the study of optics began, the ability to control the properties of light with materials found in nature has been limited. The introduction of meta materials, which cannot be found in nature, has made the creation of electromagnetic cloaks feasible and has brought a strong interest to this field of study [88]. The goal of an electromagnetic cloak on an object is for the light to flow around the object, rather than being scattered by the object, causing the object to optically disappear. Experiments have established the first practical implementation of an electromagnetic cloak over a small frequency band [88]. While this cloak was unsuccessful in obtaining perfect cloaking of a copper cylinder, the experiment showed the ability to decrease the shadow of an object, getting closer to resembling empty space; thus reinforcing the theory of electromagnetic cloaking and the practicality of an implementation. In [89] experimental observations established the construction of electromagnetic cloaks which are not dependent on prior knowledge of the incident wave, while [20] outlines an in-depth summary of electromagnetic cloaking by means of meta materials. Highlights include several key experiments, including [88], but also a discussion of future directions for electromagnetic cloaking, especially in the area of cloaking for a large frequency band, which has not yet been observed. There has been interest also in using material science to address the challenge of constructing a metamaterial that will be effective at cloaking. Numerical simulations and experiments are leading advancements in the construction of electromagnetic cloaks [15, 21].

**Partial-Differential Equation Constrained Optimization:** In this portion of the thesis we define the class of partial-differential equation constrained optimization (PDECO) problems. We first discuss the methods for solving these problems. Next, we give some examples of application problems which have fueled the interest and study of PDECO; then we cite several books and references that discuss both the theory and numerical methods available for solving PDECO problems.

A general formulation for the class of PDECO problem can be written as,

$$\begin{aligned} \underset{u,v}{\text{minimize}} \quad & J(u,v) \\ \text{subject to} \quad & P(u,v)=0 \quad \text{in } D, \end{aligned} \tag{1.34}$$

where $P$, the constraint vector, can contain scalar, algebraic, and differential constraints with respect to the state vector $u$ and the control vector $v$. We assume that at least one of the constraints present in the problem is a partial differential equation (PDE) in the state $u$ over domain $D$. $P$ also contains relevant information such a boundary conditions, initial conditions (for time dependent problems) and can also contain further constraints on the state and control vector. The goal of the optimization problem is to find an optimal state and control pair, $(u^*, v^*)$, which minimizes the cost functional $J(u,v)$. The formulation in (1.34) is referred as a full space formulation because we are optimizing over the state and control vector simultaneously.

While discussing PDECO in this thesis, we use the reduced space formulation approach [97]. In this approach, given a control vector $v$, we can produce a unique state vector $u(v)$, thus eliminating the state constraint from the optimization problem and only need to optimize over $v$:

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & J(u(v),v) \\ \text{subject to} \quad & C(u(v),v)=0 \quad , \end{aligned} \tag{1.35}$$

where $C$ is the remaining constraint(s) present in the problem when the PDE constraint is eliminated.

There are two school of thoughts when it comes to solving PDECO problems, the first is the discretize-then-optimize perspective (DTO). We do not use this approach in this thesis, but instead summarize the approach. In this approach, we first discretize the infinite-dimensonal problem. We would first discretize the objective function (which we assume in our work to be an integral whose integrand contains the state and or the control) by means of a quadrature rule. We also discretize the state equation by means of a numerical method that is appropriate and discretize all other constraints to be consistent with the state discretization. The resulting problem will be a large scale (finite dimensional) nonlinear optimization problem. We note that the size of this problem is determined by how refined the computational domain is. Once the problem is a finite-dimensional nonlinear programming problem, the wide range of tools and theory that are available can be used to solve such a problem numerically. Particularly first order methods that only require gradient information, $g$, and second order methods that require both $g$ and the Hessian matrix, $H$ [80].

The second approach is the optimize-then-discretize (OTD) approach. In this approach we first pose the Lagrangian associated with the PDE optimization problem,

$$\mathscr{L}(u,v,\lambda) = J + \int_D \lambda^T C dD, \tag{1.36}$$

where $\lambda$ is refereed to as the adjoint (multiplier) vector; we note that each component of this vector contains a scalar adjoint variable that is associated with each constraint and is a function.

In this thesis, we derive the the weak form of the state and adjoint equation to pose the optimality conditions for the reduced space formulation. Before we discuss the optimality conditions, we discuss the concept of a weak and strong solution to a partial differential equation. In classical PDE theory it is sufficient to having solutions exist in the function space $C^N(D)$; this function space contains all continuous function on $D$, whose derivatives up to order $N$ are also continuous functions [86]. We call these solutions strong (classical) solutions because the derivatives of the solution satisfy the limit definition of a derivative. The classical definition of a partial derivative of a function requires regularity assumptions on the function. For example, in the classical sense, if a function is not continuous then it cannot be differentiable. The study of weak solutions to a partial differential equation were introduced and studied by Sergei Sobolev, a Russian mathematician, in the 1930s. The motivation of introducing the concept of a weak solution was to understand the solutions of fluid equations [28]. These equations could be proven to not admit solutions in $C^N(D)$. Soblev was able to construct function spaces, which are called Soblev spaces [86] and denoted $W^{k,p}(D)$, which define the function space of all functions that are $L^p(D)$ and weak derivatives up to order $k$ are also $L^p(D)$ in the sense of distributions. We briefly now discuss the concept of a weak derivative on $D$ in the sense of distributions as first described in his work. We first define a test function space on $D$

$$\Phi = \{\phi | \phi \in C^\infty(D),\ \bar{D}^\infty \phi|_{\partial D} = 0\}, \tag{1.37}$$

where $\partial D$ denotes the boundary of the set $D$ and $\bar{D}$ is a multi-index differential operator on the elements of $\Phi$. We can summarize the test function space in words by saying that $\Phi$ is the space of all continuous functions on $D$ that are compactly supported. Next we have a function, say $u$ and we are interested in calculating $\bar{D}^n(u)$ in the weak sense, which is any partial derivative up to order $n$. Next we choose an arbitrary test function with this differential operator applied to it and "test" it against u :

$$\int_D u D^n(\phi) dD. \tag{1.38}$$

we then conduct integration by parts several times, also using the compactness of the test functions, until we have removed the differential operator on the test function:

$$\int_D u D^n(\phi) dD = (-1)^n \int_D f \phi dD, \quad \forall \phi \in \Phi. \tag{1.39}$$

We say that $f$ is the corresponding weak derivative u, in the sense of distributions, for the differential operator $\bar{D}^n$. We note that if u is differential in the standard $C^n(D)$ sense, then $f = \bar{D}^n(u)$ is the strong derivative of u. We note that the weak derivative and the strong derivative of a function is the same if the function is differentiable in the traditional sense. This is an important observation when looking at partial differential equations, because, if we apply this idea to a classical PDE, then we recover classical solutions with our weak formulation, however for more complicated equations, which do not admit classical solutions, we can seek the appropriate Soblev space for which our equation is satisfied in a weak sense. The framework that Soblev established to examine equations that do not admit classical solutions has defined the study of PDE for almost the last 100 years. A complete discussion of the theory of Soblev spaces and the use of them to successfully solve PDE can be found in [86].

With the idea of weak derivatives in mind, we then compute the integral in (1.36) by means of integration by parts, treating the adjoint variable like a test function, to weaken the regularity assumptions on the state vector $u$. Next, we introduce the first-order optimality conditions [97], which are based on the variational (Gateaux) derivatives of the Lagrangian:

$$\begin{aligned}
\mathscr{L}_\lambda[\tilde{\lambda}] &= \frac{d}{d\epsilon}\Big[\mathscr{L}(u,v,\lambda+\epsilon\tilde{\lambda})\Big]\Bigg|_{\epsilon=0} = 0 \quad \forall \tilde{\lambda} \in \tilde{F}_1 \quad \text{(Weak State Equation)}\\
\mathscr{L}_u[\tilde{u}] &= \frac{d}{d\epsilon}\Big[\mathscr{L}(u+\epsilon\tilde{u},v,\lambda)\Big]\Bigg|_{\epsilon=0} = 0 \quad \forall \tilde{u} \in \tilde{F}_2 \quad \text{(Weak Adjoint Equation)}\\
\mathscr{L}_v[\tilde{v}] &= \frac{d}{d\epsilon}\Big[\mathscr{L}(u,v+\epsilon\tilde{v},\lambda)\Big]\Bigg|_{\epsilon=0} = 0 \quad \forall \tilde{v} \in \tilde{F}_3 \quad \text{(Weak Gradient Equation),}
\end{aligned} \tag{1.40}$$

where $\tilde{F}_1$, $\tilde{F}_2$ and $F_3$ are the appropriate function spaces for the state and adjoint equation. These spaces will be problem dependent. The weak adjoint equation, like the weak state equation, is also a partial differential equation.

The optimality conditions (1.40) are defined in infinite-dimension. These conditions define the first order conditions that a local minimizing solution $v$, must satisfy.

When using either the DTO or OTD approach a finite dimensional problem must be solved eventually. In the DTO case, we derive discretize the infinite dimensional problem which results in a large scale optimization problem. We then write down the KKT system which defines optimality criteria. In the OTD, we find the gradient by solving the state and adjoint equations with the incumbent control. We note that we can pose second order conditions of optimality using a Lagrange multiplier approach.

The main numerical techniques for solving finite-dimensional continuous optimization problems are line-search and trust-region methods [53]. In the line-search approach, given an initial guess $v^0$, we define an iterative scheme

$$v^{k+1} = v^k + \alpha^k p^k, \tag{1.41}$$

where $\alpha^k$ is defined to be the step length and $p_k$ is the descent direction on the $k$-th iteration. A few popular methods for picking $\alpha_k$ include the Armijo (backtracking) and Wolfe line search [53]. Newton method [54] and Quasi-Newton methods [80] have been extremely popular as line search techniques. Newton's method for large scale problems is less attractive because second-order information must be known explicitly. This is the motivation of the quasi-Newton methods which approximates the Hessian. A popular quasi-Newton is LBFGS (limited-memory BFGS) [112].

In the trust-region method, we solve a sequence of optimization problems, $P^k$, where we take the original optimization problem and add the trust-region constraint $||v - v^k|| \leq \Delta_k$:

$$\begin{aligned} &\underset{v}{\text{minimize}} \quad \hat{J}^k \\ &\text{subject to} \\ &\quad ||v - v^k|| \leq \Delta_k \\ &\quad v \in \tilde{V}, \end{aligned} \tag{1.42}$$

where $\hat{J}^k$ is the Taylor series about an incumbent solution $v^k$:

$$\hat{J}^k = J^k + g^T(v - v^k) + \frac{1}{2}(v - v^k)^T H(v - v^k) \tag{1.43}$$

and $\Delta_k$ is the trust-region radius on the $k$-th iteration. The spirit of the trust-region method is to determine a second order (Taylor) approximation to the objective function that is to be minimized, subject to the previous constraints, in a ball of set size [80]. If such a $v$ can be found on a given iteration, we move to the solution which produces a lower objective function and either increase the trust-region radius or keep it the same. If we cannot, we decrease the radius and resolve the problem. We continue this until the trust-region radius is zero which indicates that we have found a local minimum.

Many physical phenomenon from fluid flow [19], electrodynamics [50], quantum mechanics [85], etc can be modeled by partial-differential equations (PDEs). A natural question to ask is can a physical system (described by a PDE) be driven (controlled) in such a way that a goal is reached. In this thesis, our goal is to find an electromagnetic cloak that can minimize scattering in a region to induce cloaking, which is governed by a Helmholtz equation. However, this is not the first work that addresses PDECO problems with applications in mind. An example of PDECO problems with application in mind is where the PDE constraint comes from fluid mechanics (Navier-Stokes, Euler gas laws, conservation laws). Two notable examples that researchers are interested in are boundary control to influence fluid flow and heat transfer in fluids [36]. Another example comes from electrodynamics, where the PDE constraint is Maxwell's equations (Helmholtz equation in the frequency domain). The goal of the optimization problem is to construct a topology to induce electromagnetic cloaking in a region [41]. Finally in quantum mechanics, where the governing PDE is Schrodinger equation, optimal control problems have been posed to drive quantum systems to a desired states [93]. This is not an exhaustive list of applications of PDE optimization, but it does demonstrate that researchers from different scientific domains are interested in these class of problems.

There are several references that may be an interest of readers that focus on the main components of PDE optimization: theory and numerical methods. In [43] a survey of both the full space and reduced space is explored. This includes the "one-shot" approach for solving the KKT system with both Newton and inexact Newton. It also discusses the adjoint approach for obtaining the gradient, using the Lagrangian method. The book by F. Troltzsch [97] focuses on the reduced space formulation of PDECO problems for elliptic and parabolic PDE constraints. The book rigorously proves both the first and second order optimality conditions in infinite dimension. The book by K. Ito [48] also gives the theoretical overview of the Lagrangian approach to PDECO and includes several application problems. An overview of the numerical techniques for solving PDECO (which boils down to Newton and quasi-Newton methods) can be found in [12, 42, 83]. We suggest the book of J. De los Reyes [84] as an excellent short reference for both the numerical and theoretical components of PDECO. The book by M. Gunzburger [36] provides a more in depth discussion of both the theoretical and numerical challenges but is mainly focused on fluid mechanic application problems.

**Mixed-Integer Programming:** A general mixed-integer nonlinear program (MINLP) can be written as:

$$\begin{aligned}
&\underset{\bar{x},\bar{y}}{\text{minimize}} \quad f(\bar{x},\bar{y})\\
&\text{subject to}\\
&\quad h_p(\bar{x},\bar{y})=0 \quad p=1...m,\\
&\quad g_q(\bar{x},\bar{y})\leq 0 \quad q=1...n,\\
&\quad \bar{x}\in X, \quad \bar{y}\in Y,
\end{aligned} \tag{1.44}$$

where $h_p$ for $p = 1....m$ and $g_q$ for $q = 1....m$ are real valued functions which we will refer to the equality and inequality constraints respectively. We define $\bar{y}$ to be the continuous (real-valued) variables that we are optimizing over and they exist in a set $Y \subset \mathbb{R}^k$. We denote $\bar{x}$ to be the integer variables present in the problem, which exist in a set $X \subset \mathbb{Z}^k$. We remind the reader that $\mathbb{Z}^k$ is the set of all integer points in $\mathbb{R}^k$. In this work we focus on binary integer variables, that is to say that the integer variables must exist in the set $\{0,1\}$, we call this problem a mixed-binary integer program (MBIP).

The theory for MINLP begins with convex MINLP, which is to say that $f$, $h_p$, and $g_q$ are convex functions. A naive approach for solving MBIP would first sample one of the binary outcomes (

there are $2^N$ binary combinations, where $N$ is the number of binary variables in the problem), then solving the nonlinear program. For small $N$ this is approachable, and known as the total enumeration (brute-force) approach [78]. If we look at a slightly larger number, for example $N = 20$ and $N = 40$, we would have to solve approximately $10^6$ and $10^{12}$ nonlinear programs respectively to obtain the global optimal solution. This exponential growth of nonlinear programs that are needed to be solved is not computationally tractable. Many applications of MIBP have hundreds if not thousands of variables present in a problem. The traveling salesman problem is an example of MIBP that introduces a large amount of binary integer variables. The goal of solving the problem is to find the path from one location to another, while minimizing the distance traveled between the two points. The problem is represented as a graph [103], where the nodes of the graph represent the various locations and the edge between nodes indicates a path from one location to another. One can interpret the traveling salesman problem as a figure out the edges to choose and not choose to take to get to the desired location. In some cases, this problem could have hundreds of thousands to millions of variables. The brute force approach would not be appropriate.

The Branch and Bound (B&B) method [58] is a heuristic based method to address the computational intractability that MBIP introduces. The goal of B&B is to attempt to enumerate through the binary space in such a way that total enumeration can be avoided. The algorithm can be summarized in two steps :

- branching : choosing how to explore integer space (this is heuristic based but motivated by the specific problem) [13, 18, 77].
- bounding : Introduce a relaxation of the problem, which is a nonlinear program. We say a program for the MBIP is a relaxation if the objective value for this program provides a lower bound for the original MNIP problem. Two popular relaxations for binary problems are continuous and Lagrangian relaxation [61]. In this thesis we will focus on the continuous relaxation. The original MBIP has the binary variable $x \in \{0,1\}$, while the relaxed problem will have this constraint replaced with $0 \leq x \leq 1$. After we solved the relaxed problem we introduce a rounding heuristic to round the relaxation back to $0,1$. Conducting this rounding produces an integer solution that produces an upper bound for the MBIP. Calculate the integraility gap [61], if the integraility gap is zero/near zero, then the global/sub optimal solution has been found.

We note that in practice most practitioners never solve a problem to global optimality, especially for large amount of integer variables, because of the computational effort required.

Another popular class of methods for solving convex MBIP are cutting plane methods [55]. These methods introduce a mirror problem with additional linear constraints, called cuts, which are defined so that the optimal solution to the original program does not change. It is natural to combine cutting plane technique with B&B, which is called the branch and cut method [44]. These additional linear constraints will make a given relaxation harder to solve, but the goal is to find tighter integraility gaps which will cause a natural pruning of the B&B tree. This can help the computational tractability of a problem.

In the case of nonconvex MBIP, methods have been developed to solve convex problems such as Convexification methods [45, 90, 95] convert a nonconvex problem into by doing approximations, by convex underestimates, of objective and constraints by convex functions. This new convex problem can then be solved using B&B or branch and cut.

A beginning reference that gives an introduction to linear integer programming is the book by L. Wolsey [104]. This book introduces the B&B method, cutting plane methods, model formulation, etc. For an advanced treatment of MINLP, which includes nonconvex MINLP, nonlinear MINLP, convexification, etc, the book by D. Li [61] and the book by S. Leyffer and J. Lee [59] are excellent references.

**Mixed-Integer Partial-Differential Equation Constrained Optimization** In chapters 4,5 and 6 of this thesis we study mixed-integer partial-differential equation constrained (MIPDECO) problems. These problem combine mixed-integer programming and PDECO. In this thesis, we assume there is one control present in the problem and this control can only take integer values. We focus on a subset of these problems; the particular case we focus on is when the control present in the optimization problem takes binary values $\{0,1\}$, but make note that all of the approaches and ideas introduced in this thesis can be easily extended if the control can take general integer values. The general reduced space MIPDECO we consider in this thesis, which we refer to as a binary PDECO problem, is:

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & J(u(v),v) \\ \text{subject to} \quad & C(u(v),v)=0 \\ & v \in \{0,1\}. \end{aligned} \tag{1.45}$$

We note that the only difference between PDECO and binary MIPDECO is that we restrict the control to take only binary values, whereas in the classical reduced space PDECO formulation the control is assumed to take real values.

**The Computational Bottleneck for MIPDECO** Very little work has been done in the area of MIPDECO due to the computational challenges that arise. However in the face of these many challenges there also are many opportunities to make improvements on the theory, methods, and applications of MIPDECO. The standard approach for solving MIPDECO is first to discretize the infinite dimensional optimization problem [38–40]. This results in a large scale MINLP that can be solved with B&B if the problem is convex, and convexification methods when the problem is not convex. While executing the branch and bound method the solution of the continuous relaxation is required. The continuous relaxation problem is defined to be when we relax the integer variables from being in the set $\{0, 1\}$ to being in the continuous interval $[0, 1]$, which makes the large scale MINLP into a large scale NLP. We may need to solve a relaxation hundreds, if not thousands of times to satisfy the MINLP optimality conditions, which may or not be computationally tractable. The size of this large scale nonlinear program is directly related to the mesh refinement. This is a problem because as we refine the mesh to obtain a refined state and control (if the control is also mesh dependent), we also increase the size of the problem drastically (exponentially if the control is mesh dependent). Even though several papers have been written in the subject recently [38–40], there are still many challenges when solving MIPDECO from the DTO approach.

In this thesis we will choose a OTD approach for several reasons. The first reason is that by finding the optimality conditions in infinite-dimensions and then discretizing these conditions we are maintaining physics of the underlying system. The second reason is that our PDE optimization contains integer variables. The DTO approach has traditionally been applied the MIPDECO which requires a large-scale MINLP to be solved. This problem is solved using a branch and bound method [8]. Solving large scale MINLP problems in an efficient manner is still an open question. This had led to the computational intractability of these problems and has been a major bottleneck for progress in solving MIPDECO. We instead introduce a trust-region method, which uses an adjoint based gradient, to solve MIPDECO. Our trust-region method uses a linear approximation of the objective function instead of the standard quadratic approximation. The reason we do not introduce the Hessian (quadratic) term to the trust-region subproblem is to ensure the computational tractability of the problem. Because we only use a linear model, the trust-region problem reduces to a knapsack problem that can be solved in $\mathscr{O}(N\log(N))$, where $N$ is the number of binary integers present in the problem. Our method produces quality minimizing solutions in a reasonable amount of trust-region iterations. This is important to note because even though the search space if $2^N$, we are able to find a solution to a given instance of our MIPDECO within one-hundred iterations.

## 1.4 Thesis Contributions

**The Immersed Interface Method for the One and Two Dimensional Maxwell's Equation:**
In chapter 2 we introduce a direct immersed interface method that uses a modified Yee's Scheme to solve the 1D Maxwell's equations with discontinuous permittivity and permeability. This work has provided the following contributions:

- An explicit finite difference scheme to solve Maxwell's equations in one dimensional space with interfaces using the interface conditions directly. Even with discontinuous permittivity and permeability, our finite difference scheme solves the equations to second order accuracy in time and space.

- Numerical experiments to demonstrate the consistency, stability and convergence of the direct method.

In chapter 3 we introduce an ADI Yee's scheme method to solve Maxwell's equations in 1D and 2D with discontinuous permittivity and permeability. In 2D we can solve the interface problem even with a curved interface. This work has provided the following contributions:

- An ADI-Yee's scheme in 1D and 2D that takes advantage of augmented variables, to solve the problem dimension by dimension [64].
- A proof for the stability of the ADI-Yee's Scheme for 1D Maxwell's equations.
- Numerical experiments to establish the consistency, stability and convergence of the methods.

**Mixed-Integer Partial Differential Equation Optimization Theory and Binary Electromagnetic Cloaks:**
In Chapter 4 we have made the following contributions:

- A benchmark (application) problem for MIPDECO: 2D single material electromagnetic binary cloaking, which can be modeled as a MIPDECO.
- A steepest-descent trust-region method for MIPDECO. The method is effective at producing good local solutions and is efficient even with a large amount of integer variables present in the problem. Our framework for MIPDECO relies on solving a sequential set of PDE and trust-region sub problems (knapsack problems) in order to arrive at a local solution in a reasonable amount of iterations. We showed through numerous computational experiments that the knapsack problems take negligible computational time when compared with the computational effort required to first solve the state and adjoint PDEs, then build the gradient for the sub problem. This is encouraging because the computational bottleneck of MIPDECO has been traditionally encountered due to the presence of a large amount of integer variables. With our perspective, we have reduced MIPDECO back to the difficulties associated with PDECO which is to first efficiently solving the state and adjoint equation and then build the gradient using the state and adjoint variables.
- We introduced a systematic manner to solve MIPDECO with our trust-region method: first we solve the continuous relaxation, next we round the continuous relaxation using a rounding heuristic that produces a feasible starting guess for the MIPDECO, then we use the rounded relaxation as a starting guess for our trust-region method. We have compared this approach versus starting the trust-region with a random guess, and have found that even though significant computational effort is required to solve the continuous relaxation, we take less trust-region iterations to find a local solution. Moreover this solution produces an objective function value that is lower than random sampling. Still sampling could still be attractive. If a

user is just interested in a local minimum solution and does not care about the quality of the minimum, then starting the trust-region method from a random sample is more computationally efficient. Moreover, since many MIPDECO of interest are nonconvex, one could take several samples and make an attempt to "globalize" the algorithm (also known as multi-start methods [73]).

In chapter 5 we extend our work from chapter 4, assuming that a 2D binary electromagnetic cloak can be built using several materials, leading to the following contributions:

- A benchmark (application) problem for MIPDECO: 2D multi-material electromagnetic binary cloaking. This model introduces non-trivial constraints to the MIPDECO that are studied.
- A modified trust-region method that is based on our original trust-region algorithm for MIPDECO to deal with constraints. We show through numerical experiments that even with the introduction of constraints, the numerical computation of this instance of a MIPDECO is still computationally tractable.
- Two separate continuation techniques in order to find local minimum to this instance of a MIPDECO problem that produces minimizers with superior quality when compared to solving the problem from scratch.

In chapter 6 we extend our work from chapter 4 to a 3D generalized model from the 2D model. The contributions of this work include:

- A benchmark (application) problem for MIPDECO: 3D electromagnetic binary cloaking using a single material.
- The structure of this problem introduces a large number of integer variables naturally. We solve an instance of the problem with 8000 cloaking blocks that make up the 3D binary electromagnetic cloak. Even with a large number of integer variables present in the problem our trust-region method is still computationally tractable and finds good local solutions.

CHAPTER

# 2

# A DIRECT APPROACH FOR SOLVING MAXWELL'S EQUATIONS IN ONE DIMENSION WITH THE IMMERSED INTERFACE METHOD

## 2.1 Chapter Outline

In this chapter we will introduce the philosophy of the immersed interface method (IIM)[64], for finite difference methods, then apply the method to Maxwell's equations in one dimension. We start this chapter by describing how the IIM works on a simple ordinary differential equation (ODE). We then discuss consistency, stability and convergence of the scheme. In the remainder of the chapter, we discuss our seconder order IIM schemes for Maxwell's equations. We present numerical results to establish the consistency, stability and convergence.

## 2.2 The IIM Applied to a Simple ODE

The purpose of the IIM is to solve either ordinary or partial differential equations (PDEs) with discontinuities present in the problem. These discontinuities can occur either in terms of the equation, the solution to the equation, or the solutions derivatives. We will demonstrate the IIM method applied to the simplest initial value problem as an example to demonstrate how the method works:

$$\frac{dy}{dx}(x) = c\,y(x) \quad \text{in} \quad \Omega = [0,1] \tag{2.1}$$

$$y(0) = 1, \tag{2.2}$$

where

$$c(x) = \begin{cases} c_1 & x < \alpha \\ c_2 & x \geq \alpha \end{cases} \tag{2.3}$$

and $c_1$, $c_2$, $\alpha$ are real numbers. The quantity $\alpha$ is a point where $c$ could be discontinuous. We refer to the point $x = \alpha$ as the interface point.

### 2.2.1 The Failure of the Euler's Method

We begin to show how the traditional approach to solve this ODE with a discontinuous coefficient fails. To numerically solve (2.1), with a finite difference approach, we would first discretize $\Omega$ into $n+1$ discrete points $x_0..x_n$ where

$$x_i = ih \quad i = 0..n \tag{2.4}$$

$$h = 1/n. \tag{2.5}$$

Next we approximate the derivative on this grid at point $x_i$ with the forward difference formula

$$\frac{dy}{dx}(x_i) = \frac{y(x_{i+1}) - y(x_i)}{h} + \mathcal{O}(h). \tag{2.6}$$

Now we write an explicit formula to approximate $y^{i+1} := y(x_{i+1})$, which is known as the forward Euler method,

$$y^{i+1} = y^i + hc_i y^i = (1 + hc_i)y^i, \tag{2.7}$$

with $y_0 = y(0) = 1$.

We conduct a numerical study to show the failure nodes of Euler's method applied to a differential equation with discontinuous coefficients. We show this pictorially and theoretically. We will take

$n = 2^k$ for $k = 5, 6, 7, ..., 11$. We also calculate the convergence order

$$\gamma \approx \frac{\log\left(\frac{E_n}{E_{2n}}\right)}{\log\left(\frac{h}{\frac{h}{2}}\right)}. \tag{2.8}$$

Because the error on a grid broken into $n$ pieces $\mathscr{O}(h)$, then a grid broken into $2n$ pieces would be $O\left(\frac{h}{2}\right)$. Ignoring the constant in the $\mathscr{O}$ term, we find theoretically that

$$\gamma \approx \frac{\log(2)}{\log(2)} = 1. \tag{2.9}$$

So as we continue to refine the mesh, we should see numerically that the convergence order is 1, which is the same as saying that the error is reduced by $\frac{1}{2}$ every time we refine the mesh.

We solve the ODE (2.1) in two cases; the first where $c_1 = c_2 = 0.5$, thus c is continuous; the second case $c_1 = 0.5, c_2 = 1$, thus c is a discontinuous function. We set $\alpha = 0.5$. The true (exact) solution to the ODE, which we define as $y_{\text{exact}}$, is

$$y_{\text{exact}}(x) = \begin{cases} e^{c_1 t} & x \leq \frac{1}{2} \\ e^{c_2 t} & x \geq \frac{1}{2}. \end{cases} \tag{2.10}$$

We measure the error for a fixed $n$ in the infinity norm, by taking $y_{\text{approx}}(x_i) := y(x_i) = y^i$ then defining the corresponding error $E_n$ as

$$E_n = ||y_{\text{exact}} - y_{\text{approx}}||_\infty = \max_i |y_{\text{exact}}(x_i) - y_{\text{approx}}(x_i)|. \tag{2.11}$$

We first report the error and convergence order found numerically for the two cases in Table 2.1 and Table 2.2 respectively. We observe in Table 2.1 that for the continuous $c$ we recover first order convergence, which is expected. However in Table 2.2 we observe that our convergence order is not converging to the value one, instead converging to zero. This indicates that when $c$ is discontinuous we make negligible progress towards reducing the error as we refine the mesh.

We now plot the exact solution versus the numerical solution for $n = 2048$ (a very fine mesh) for both the continuous and discontinuous case in Figure 2.1. This illustrates why the standard finite difference method fails. We see that for the continuous coefficient, our numerical and exact solution match. However for the discontinuous coefficient we see that the exact and approximate

**Table 2.1** Error and convergence for continuous coefficient.

| n | Error | Convergence Order |
|---|---|---|
| 32 | 6.2513e-03 | - |
| 64 | 3.1740e-03 | 0.9778 |
| 128 | 1.5987e-03 | 0.9894 |
| 256 | 8.0221e-04 | 0.9948 |
| 512 | 4.0181e-04 | 0.9974 |
| 1024 | 2.0108e-04 | 0.9987 |
| 2048 | 1.0059e-04 | 0.9993 |

**Table 2.2** Error and convergence for discontinuous coefficient.

| n | Error | Convergence Order |
|---|---|---|
| 32 | 6.1726e-01 | - |
| 64 | 6.0988e-01 | 0.0173 |
| 128 | 6.0574e-01 | 0.0098 |
| 256 | 6.0355e-01 | 0.0052 |
| 512 | 6.0242e-01 | 0.0026 |
| 1024 | 6.0186e-01 | 0.0013 |
| 2048 | 6.0157e-01 | 0.0006 |

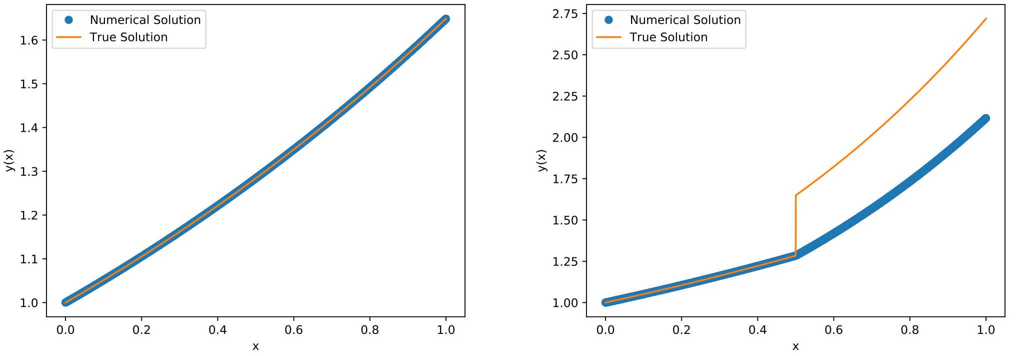
solution agree until we reach the point of discontinuity. After this point, the approximate and exact solution diverge. This is why the convergence order is nearly zero for the discontinuous problem; because as we refine the mesh, we cannot resolve the jump that was caused in the solution due to the discontinuous coefficient. In the next subsection we address this challenge with the IIM.

### 2.2.2 A Modification to the Euler's Method

In the previous Subsection 2.2.1 we observed that Euler's method fails when applied to discontinuous coefficients. In this subsection we address this challenge with the IIM.

We first briefly discuss why Euler method fails. When deriving finite difference approximations, some assumptions on regularity are required. In the case of our forward difference approximation, this approximation is only valid if $y \in C^2(\Omega)$, where $C^p$ is the space of all functions that are continuous and have continuous derivatives up to order $p$. When we look at the true solution (2.10), we see that the solution is not continuous when $c_1 \neq c_2$, which already imply the function does not have classical derivatives. Because the solution does not admit a classical derivative, the finite-difference approximation is not valid.

We do see that on the pieces of $\Omega$, $\Omega^- := \Omega^-_\alpha = \{x : x < \alpha\}$ and $\Omega^+ := \Omega^+_\alpha\{x : x \geq \alpha\}$, the solution is $C^2(\Omega^-_\alpha)$ and $C^2(\Omega^+_\alpha)$. This observation is what motivates the finite-difference approach for



**(a)** Continuous Coefficient $c_1 = c_2 = 0.5$. **(b)** Discontinuous Coefficient $c_1 = 0.5, c_2 = 1.$.

**Figure 2.1** Comparison of solutions for continuous and discontinuous coefficient.

the immersed interface method. Because we have the regularity required away from the point of discontinuity, the typical forward difference approximation is valid. The difficulty is when we are approximating near the point where the discontinuity occurs. For the remainder of this chapter, we focus only on the case when $c_1 \neq c_2$ ($c$ is discontinuous). We now discuss the implementation of the IIM applied to (2.1). An outline of the steps are

- Determine the jump conditions for the PDE.
- Break the interval $[0, 1]$ into a uniform mesh (Cartesian Mesh).
- Determine which points in the discretization are regular and irregular.
- Choose a finite difference stencil to approximate the derivative.
- Use the standard finite difference approximation at regular grid points and a corrected approximation at irregular grid points.

We first discuss the jump conditions for our ODE 2.1. Our notation for the jump conditions will be as followed: we use $[y] = (y^+ - y^-)\Big|_{x=\alpha}$, to denote the jump of the state over the interface point $\alpha$, which we refer as the zero order jump condition. Similarly we use $[\frac{dy}{dx}] = (\frac{dy}{dx}^+ - \frac{dy}{dx}^-)\Big|_{x=\alpha}$ to denote the first order jump condition in the first derivative. We will use a direct approach, which we know $[u], [u_x]$. We will use our exact solution for the interface problem to derive these exactly to illustrate how the IIM works. The jump conditions are then

$$\begin{aligned}\left[u\right] &= e^{c_2\alpha} - e^{c_1\alpha}\\ \left[\frac{dy}{dx}\right] &= c_2 e^{c_2\alpha} - c_1 e^{c_1\alpha}.\end{aligned}$$

We now generate a Cartesian mesh $x_0....x_n$. In our approach however we classify these grid points into two classes - regular and irregular grid points.

**Definition 2.2.1.** Irregular Grid Point: A grid point $x_i$ is said to be irregular for our finite difference approximation of the first derivative, if $x_i \leq \alpha \leq x_{i+1}$.

*Remark.* As noted in the definition, the grid points that are considered irregular for an approximation is dependent on the grid points uses in the approximation of a given derivative. In general if an approximation of a derivative of any order is made at $x_i$, this grid point is irregular if $x = \alpha$ is contained in the span of the finite difference stencil.

**Definition 2.2.2.** Regular Grid Point: A grid point $x_i$ is said to be regular if it is not irregular.

We now discuss creating the scheme for the interface problem. At the regular grid points, $x_i$, we take advantage of the standard forward difference approximation. At irregular grid points, we do a Taylor series expansion to approximate the derivative at $x = \alpha$ on the left side of the interface,

$$\frac{dy}{dx}^{-}(\alpha) = a y^{i+1} + b y^{i}, \tag{2.12}$$

where $a, b$ are determined in such a way that the approximation is valid. Because we are at an irregular grid point $x_i \leq \alpha \leq x_{i+1}$, then $x_{i+1} \in \Omega^+$ and $x_i \in \Omega^-$. We then expand based on which side of the interface we are on, which in this case is the left of the interface point, which results in

$$\begin{aligned}\frac{dy}{dx}^{-}(\alpha) = a\Big(y^+(\alpha) + \frac{dy}{dx}^{+}(\alpha)\big(x_{i+1}-\alpha\big) + \mathcal{O}(\big(x_{i+1}-\alpha\big)^2)\Big)\\ +b\Big(y^-(\alpha) + \frac{dy}{dx}^{-}(\alpha)\big(x_i-\alpha\big) + \mathcal{O}\big(((x_i-\alpha\big)^2)\big).\end{aligned} \tag{2.13}$$

We see that our expansion has terms which are evaluating the state and its derivative on the two sides of the equation. To compare the two sides of the equation, we replace the positive side terms with negative side terms using the jump conditions (2.12) in our expansion by solving plus terms,

$$\begin{aligned}u^+ &= u^- + e^{c_2 t} - e^{c_1 t}\\ \frac{dy}{dx}^{+} &= \frac{dy}{dx}^{-} + c_2 e^{c_2 t} - c_1 e^{c_1 t}.\end{aligned} \tag{2.14}$$

When we substitute (2.14) into our expansion and then equate the sides, we have a linear system,

$$\begin{bmatrix} 1 & 1 \\ x_{i+1}-\alpha & x_i-\alpha \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}=\begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.15}$$

The analytic solution to the linear equation is

$$a=\frac{1}{h}, \qquad b=-\frac{1}{h}.$$

When we substitute these values back into the Taylor approximation we find that

$$\frac{dy^-}{dx}(\alpha)=\frac{y^{i+1}-y^i}{h}-C_i+\mathcal{O}(h) \tag{2.16}$$

where we denote $C_i$ to be the correction term at irregular node $x_i$ and

$$C_i=\frac{1}{h}\left(e^{c_2\alpha}-e^{c_1\alpha}+(c_2e^{c_2\alpha}-c_1e^{c_1\alpha})(x_{i+1}-\alpha)\right). \tag{2.17}$$

We use the standard forward Euler method when updating regular grid points, when we are updating irregular grid points we edit the scheme with a correction to guarantee an approximation that takes into account the jump conditions, therefore creating a method that will successfully solve the equation (2.1) when $c$ is discontinuous. Our scheme is summarized as follows: if the grid point $x_i$ is regular grid point then

$$y^{i+1}=(1+c_ih)y^i \quad \text{(Standard Forward Euler)}.$$

If the grid point $x_i$ is an irregular grid point then the corrected scheme is

$$y^{i+1}=(1+c_ih)y^i+hC_i \quad \text{(Corrected Forward Euler)}.$$

### 2.2.3 Stability, Consistency, Convergence of the Modified Euler's Scheme

In the context of the IIM method we have two classes of grid points: regular and irregular. At regular grid points we use a classical scheme, where stability and consistency results have already been established. We will only focus on linear equations in this thesis, therefore establishing stability and consistency of the scheme at both irregular and regular grid points is sufficient to establish convergence. This is due to the Lax Equivalence Theorem [64, 66]. In our corrected forward Euler method, the same stability and consistency arguments hold at irregular grid points as the regular grid points, since the stencil weights are the same. Because we have the same finite-difference weights in the irregular scheme, we maintain the first order error at both regular or irregular grid points. The correction term does not alter the stability of the problem and can be considered a source term which eliminates the extra terms of the Taylor series to guarantee our derivative is first order accurate (as it is in the standard forward Euler method).

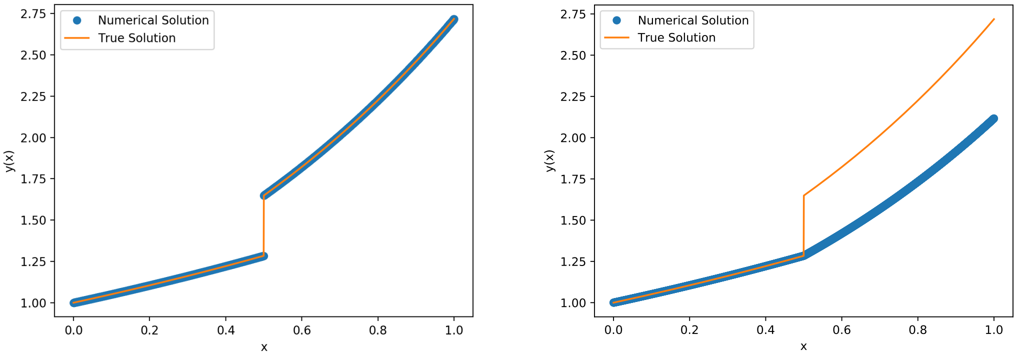
### 2.2.4 Numerical Results of the Modified Forward Euler Method

We now redo the previous simulation with our corrected forward Euler scheme with $c_1 = .5$, $c_2 = 1$. We first show the convergence results in Table 2.3. We see that our corrected scheme has a convergence order of one, that is to say that our corrected scheme maintains first order accuracy as we refine the mesh.

**Table 2.3** Error and Convergence for Discontinuous Coefficient with Corrected Euler Method.

| n | Error | Convergence Order |
|---|---|---|
| 32 | 2.3126e-02 | - |
| 64 | 1.2126e-02 | 0.9314 |
| 128 | 6.2031e-03 | 0.9670 |
| 256 | 3.1365e-03 | 0.9838 |
| 512 | 1.5770e-03 | 0.9919 |
| 1024 | 7.9067e-04 | 0.9960 |
| 2048 | 3.9588e-04 | 0.9980 |

We plot the solution for the regular Euler scheme and the corrected Euler scheme on a $n = 2048$ point mesh in Figure 2.2. We observe that our corrected scheme produces a result that is consistent with the true solution. We have captured the jump that is caused by the discontinuity with our corrected scheme, which was the failure mode for the standard Euler's method. We have shown with this simple example that the IIM can successfully address the challenge of solving differential equations with a discontinuous coefficient.



**(a)** Modified Euler Scheme $c_1 = 0.5, c_2 = 1$.

**(b)** Standard Euler Scheme $c_1 = 0.5, c_2 = 1$.

**Figure 2.2** Comparison Solutions for Continuous and Discontinuous Coefficient.

## 2.3 The IIM Applied to the 1D Maxwell's Equation

We begin with defining Maxwell Equations in 1D:

$$\begin{aligned} &\frac{\partial H^y}{\partial t} = \frac{1}{\mu}\frac{\partial E^z}{\partial x} \\ &\frac{\partial E^z}{\partial t} = \frac{1}{\epsilon}\frac{\partial H^y}{\partial x} \\ &\frac{\partial E^y}{\partial t} = -\frac{1}{\epsilon}\frac{\partial H^z}{\partial x} \\ &\frac{\partial H^z}{\partial t} = -\frac{1}{\mu}\frac{\partial E^y}{\partial x} \\ &a \leq x \leq b \\ &0 \leq t \leq T_{\text{final}}, \end{aligned} \tag{2.18}$$

where $a, b$ are real numbers and $T_{\text{final}}$ is the maximum time we observe the behavior of the electromagnetic wave. We will work with (and derive numerical methods) for the equations (2.18). We choose to work with the electric field **E** and the magnetizing field **H**. We note that one can go between the field the electric field **E** and displacement field **D** by the constitutive relation,

$$\mathbf{D} = \epsilon \mathbf{E}. \tag{2.19}$$

Similarly we can go between the magnetic field **B** and the magnetizing field **H** by the constitutive relation

$$\mathbf{B} = \mu \mathbf{H}. \tag{2.20}$$

Historically, (2.18) is known as the TM (Transverse Magnetic) equation, we will also solve the TE (Transverse Electric) equation as well [50]. Both the transverse electric and magnetic equations can be expressed as

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{1}{\beta}\frac{\partial v}{\partial x} &= f \\ \frac{\partial v}{\partial t} + K\frac{\partial u}{\partial x} &= g. \end{aligned} \tag{2.21}$$

When $u = -H^y$, $v = -E^z$, $\beta = \mu$, $K = \frac{1}{\epsilon}$, and $f = 0$, (2.21) is the TM equations. When $u = E^y$, $v = H^z$, $\beta = \epsilon$, $K = \frac{1}{\mu}$, and $g = 0$, (2.21) is the TE equations. We use this system for the remainder of this chapter. We note that we add the source terms $f, g$ in our derivation in order to easily validate theoretical findings in a later sections. Additionally, we use this system to derive jump conditions for both the TM and TE equations simultaneously in subsection 2.3.2.

Next we briefly discuss Maxwell's equations numerically. The popular methods include the finite volume approach, the finite-difference time domain (FDTD) approach, commonly referred as Yee's Scheme [105], and the finite element time domain (FETD) approach. A general summary of these methods, including their strengths and weaknesses in practice is discussed in [47]. In this thesis we focus only on the FDTD approach. As discussed in 2.2, we base our IIM approach on using a standard finite difference stencil. The FDTD stencil is a natural choice due to being well known in the numerical electromagnetic community as a stable second order method for solving Maxwell's Equations with continuous permittivity and permeability. We extend the FDTD method to also solve the equation with discontinuous permittivity and permeability, for,

$$\epsilon = \begin{cases} \epsilon^- & x < \alpha \\ \epsilon^+ & x \geq \alpha, \end{cases} \tag{2.22}$$

$$\mu = \begin{cases} \mu^- & x < \alpha \\ \mu^+ & x \geq \alpha. \end{cases} \tag{2.23}$$

The outline for the remainder of this section is as follows: we first introduce the FDTD (Yee Scheme) method. We then follow the steps that were outlined in Section 2.2, but apply these steps to the 1D Maxwell equations with discontinuous permittivity and permeability. We first derive the jump conditions for (2.21), then produce numerical schemes to solve both the TE and TM equations with discontinuous coefficients. We analyze and establish our methods consistency, stability, and convergence. We end the section with the presentation of numerical simulations that validate our theoretical findings. The derivation and analysis of the TM and TE schemes are very similar, therefore we discuss the main ideas in the TM section. We omit some details in the TE section that do not add intellectual content to the thesis due to repetitive information.

### 2.3.1 Introduction

In this section we discuss a direct IIM method for the 1D Maxwell Equations. We call the method "direct" because we will derive the jump conditions for the interface problem explicitly [64]. An outline of the section is as follows: we first derive the jump conditions for our system (2.21), which will hold for both the TM and TE equations. Next we introduce the Yee Scheme [105] both for the TM and TE equations. Afterwards we derive a IIM method for first the TM equation, and then the TE equation. In each case we will study the consistency and stability of respective method, then establish convergence results. We conclude the discussion of each method by presenting numerical results which support our theoretical findings.

### 2.3.2 Jump Conditions

We begin with deriving the jump conditions which are required to produce our IIM method. Because we are interested in producing a second-order scheme in time and space we only need to derive the zero; first; and second-order jump conditions. We first write the jump conditions in full, then we discuss the derivation of the conditions. The jump conditions for (2.21), for a fixed one-dimensional interface $x=\alpha$ are

$$[u]=0 \qquad\qquad [v]=0 \tag{2.24}$$

$$[u_t]=0 \qquad\qquad [v_t]=0 \tag{2.25}$$

$$[u_{tt}]=0 \qquad\qquad [v_{tt}]=0 \tag{2.26}$$

$$\left[K u_x\right]=[g] \qquad\qquad \left[\frac{1}{\beta} v_x\right]=[f] \tag{2.27}$$

$$\left[\frac{K}{\beta} u_{xx}\right]=\left[\frac{g_x}{\beta}-f_t\right] \qquad\qquad \left[\frac{K}{\beta} v_{xx}\right]=\left[K f_x-g_t\right]. \tag{2.28}$$

The jump conditions (2.24)-(2.26) are assumed to be true. A reference to the validity of (2.24) can be found in [50]. The remaining assumptions are based on the empirical observation concerning conservation of charge (which has never been violated experimentally) [50]. If these jump conditions were to be violated, this would imply that charge had entered or left a given system instantaneously. This violates conversation of charge, which states that charge can neither be created nor destroyed.

The jump conditions (2.27) are obtained using our assumption in the jump condition (2.25). We first apply to argument to obtain the jump condition for the spatial derivative of $u$:

$$\left[\frac{\partial v}{\partial t}+K\frac{\partial u}{\partial x}\right]=[g] \tag{2.29}$$

$$\Longrightarrow\left[\frac{\partial v}{\partial t}\right]+\left[K\frac{\partial u}{\partial x}\right]=[g], \tag{2.30}$$

but $[v_t]=0$, therefore

$$\left[K\frac{\partial u}{\partial x}\right]=[g]. \tag{2.31}$$

Without loss of generality, we can apply the same argument to the remaining equation, instead using $[u_t]=0$, which yields

$$\left[\frac{1}{\beta}v_x\right]=[f]. \tag{2.32}$$

The jump conditions (2.28) are derived by manipulating (2.21) and applying the assumed jump conditions (2.26). We first derive the second order jump condition for $u$. We notice from (2.21) the following system is also true:

$$\begin{aligned} u_{tt}+\frac{1}{\beta}v_{xt}&=f \\ \frac{1}{\beta}v_{tx}+\frac{K}{\beta}u_{xx}&=\frac{g_x}{\beta}. \end{aligned} \tag{2.33}$$

Because we assume enough regularity, the mixed partials have the property $v_{tx}=v_{xt}$, therefore when we subtract the second equation from the first equation we obtain

$$\frac{K}{\beta}u_{xx}-u_{tt}=\frac{g_x}{\beta}-f_t. \tag{2.34}$$

We enforce our assumption that $[u_{tt}]=0$ to obtain the second order jump condition for u:

$$\left[\frac{K}{\beta}u_{xx}\right]=\left[\frac{g_x}{\beta}-f_t\right]. \tag{2.35}$$

We again use a similar argument to derive the second order jump condition of v, noticing

$$Ku_{tx}+\frac{K}{\rho}v_{xx}=Kf_x \tag{2.36}$$

$$v_{tt}+Ku_{xt}=g_t. \tag{2.37}$$

Again, enforcing that the mix partials are equal, $[v_{tt}]$=0, and subtracting the second equation from the first equation yields

$$\left[\frac{K}{\beta}v_{xx}\right]=\left[Kf_x-g_t\right]. \tag{2.38}$$

We will use these jump conditions to correct our FDTD scheme. As we previously noted, we will use these jump conditions directly to correct the FDTD scheme. When we correct the scheme, it will be dependent if the interface point $x=\alpha$ is to the left or right of where we expand our Taylor approximation. We write the jump conditions explicitly (omitting the jump in time derivatives), which we will reference when deriving our IIM method.

$$u^+=u^- \qquad u^-=u^+ \tag{2.39}$$

$$v^+=v^- \qquad v^-=v^+ \tag{2.40}$$

$$u_x^+=\frac{K^-}{K^+}u_x^-+\frac{[g]}{K^+} \qquad u_x^-=\frac{K^+}{K^-}u_x^--\frac{[g]}{K^-} \tag{2.41}$$

$$v_x^+=\frac{\beta^+}{\beta^-}v_x^-+\beta^+[f] \qquad v_x^-=\frac{\beta^-}{\beta^+}v_x^--\beta^-[f] \tag{2.42}$$

$$u_{xx}^+=\frac{\beta^+K^-}{\beta^-K^+}u_{xx}^-+\frac{\beta^+}{K^+}\left[\frac{g_x}{\beta}-f_t\right] \qquad u_{xx}^-=\frac{\beta^-K^+}{\beta^+K^-}u_{xx}^+-\frac{\beta^-}{K^-}\left[\frac{g_x}{\beta}-f_t\right] \tag{2.43}$$

$$v_{xx}^+=\frac{\beta^+K^-}{\beta^-K^+}v_{xx}^-+\frac{\beta^+}{K^+}\left[Kf_x-g_t\right] \qquad v_{xx}^-=\frac{\beta^-K^+}{\beta^+K^-}v_{xx}^+-\frac{\beta^-}{K^-}\left[Kf_x-g_t\right] \tag{2.44}$$

### 2.3.3 The 1D TM Equations and Yee's Scheme

In this subsection we discuss the grids that will be used for the FDTD method, then we introduce the classical FDTD method for the TM equations. We introduce the IIM scheme that uses the FDTD method at regular grid points and a corrected FDTD scheme at irregular grid points. We study the consistency, stability and convergence of our scheme. We end the subsection by conducting a numerical experiment to illustrate the success of our method in practice. Before discussing the scheme, we first discuss the spatial and time mesh for our problem. We will define two spatial meshes and two time meshes. We first define two spatial meshes. The first mesh is a discretization of $[a,b]$ into $n_x+1$ points. We define the first spatial mesh as

$$X_{\text{full}}=\left\{x_i : x_i=a+i\Delta x,\ i=0\ldots n_x,\ \Delta x=\frac{b-a}{n_x}\right\}, \tag{2.45}$$

and the second spatial mesh as

$$X_{\text{half}} = \left\{ x_{i+\frac{1}{2}} : x_{i+\frac{1}{2}} = a + \frac{\Delta x}{2} + i\Delta x,\ i = 0 \ldots n_x,\ \Delta x = \frac{b-a}{n_x} \right\}. \tag{2.46}$$

We refer to $X_{\text{full}}$ as the main grid, and $X_{\text{half}}$ as the staggered grid in space. We now define the same grid structure for the time variable, this time taking $n_t + 1$ time steps:

$$T_{\text{full}} = \left\{ t_i : t_i = i\Delta t,\ i = 0 \ldots n_t,\ \Delta t = \frac{T_{\max}}{n_t} \right\}, \tag{2.47}$$

and the second spatial mesh as

$$T_{\text{half}} = \left\{ t_{i+\frac{1}{2}} : t_{i+\frac{1}{2}} = \frac{\Delta t}{2} + i\Delta t,\ i = 0 \ldots n_t,\ \Delta t = \frac{T_{\max}}{n_t} \right\}. \tag{2.48}$$

The FDTD scheme takes advantage of these grids approximating the derivatives in the TM equations as follows:

$$\frac{\partial u}{\partial t}(x_{i+\frac{1}{2}}, t_n) = \frac{u_{i+\frac{1}{2}}^{n+\frac{1}{2}} - u_{i+\frac{1}{2}}^{n-\frac{1}{2}}}{\Delta t} + \mathcal{O}(\Delta t^2), \tag{2.49}$$

$$\frac{\partial v}{\partial t}(x_i, t_{n+\frac{1}{2}}) = \frac{v_i^{n+1} - v_i^n}{\Delta t} + \mathcal{O}(\Delta t^2), \tag{2.50}$$

$$\frac{\partial u}{\partial x}(x_i, t_{n+\frac{1}{2}}) = \frac{u_{i+\frac{1}{2}}^{n+\frac{1}{2}} - u_{i-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} + \mathcal{O}(\Delta x^2), \tag{2.51}$$

$$\frac{\partial v}{\partial x}(x_{i+\frac{1}{2}}, t_n) = \frac{v_{i+1}^n - v_i^n}{\Delta x} + \mathcal{O}(\Delta x^2). \tag{2.52}$$

We now define the FDTD (Yee) scheme for the TM equation:

$$\begin{aligned} u_{i+\frac{1}{2}}^{n+\frac{1}{2}} &= u_{i+\frac{1}{2}}^{n-\frac{1}{2}} - \frac{\Delta t}{\beta_{i+\frac{1}{2}}\Delta x}\left( v_{i+1}^n - v_i^n \right) + \Delta t f_{i+\frac{1}{2}}^n \\ v_i^{n+1} &= v_i^n - K_i \frac{\Delta t}{\Delta x}\left( u_{i+\frac{1}{2}}^{n+\frac{1}{2}} - u_{i-\frac{1}{2}}^{n+\frac{1}{2}} \right) + \Delta t g_i^n \end{aligned} \tag{2.53}$$

We use the short hand notation $u_{i+\frac{1}{2}}^{n+\frac{1}{2}} = u(x_{i+\frac{1}{2}}, t_{n+\frac{1}{2}})$, $v_i^{n+1} = v(x_i, t_{n+1})$, etc. This scheme is $\mathcal{O}(\Delta x^2 + \Delta t^2)$ (second order accurate in time and space) for continuous permittivity and permeability. Moreover, the method is stable, in the sense of Von-Neumann stability analysis [60], for continuous $K, \beta$, under the assumption that Courant Friedrichs Lewy (CFL) condition is satisfied [47]:

$$\max_{\Omega, t}\left( \sqrt{\frac{K}{\beta}} \right)\frac{\Delta t}{\Delta x} \leq 1. \tag{2.54}$$

### 2.3.4 The Corrected Yee's Scheme for the 1D TM Equation

We can use the standard scheme (2.53) at regular grid points. At irregular grid points we need to correct the scheme. Because the first and second time derivatives are continuous across the interface we only need to correct the scheme for the spatial derivative terms.

We first discuss the correction required for $\frac{\partial v}{\partial x}$. The standard Yee scheme uses the spatial stencil points $(x_i, x_{i+\frac{1}{2}}, x_{i+1})$ to update the $u$ at $x_{i+\frac{1}{2}}$, for the next time step $t_{n+\frac{1}{2}}$. So there are two cases to consider. In the first case, the interface point $x = \alpha$ satisfies

$$x_i \le \alpha \le x_{i+\frac{1}{2}} \le x_{i+1}, \tag{2.55}$$

and in the second case,

$$x_i \le x_{i+\frac{1}{2}} \le \alpha \le x_{i+1}. \tag{2.56}$$

We begin the by addressing the approximation in the first case. Because the classical Yee approximation of the derivative is at $x_{i+\frac{1}{2}}$ and $\alpha \le x_{i+\frac{1}{2}}$, then $x_{i+\frac{1}{2}} \in \Omega^+$. We use the method of undetermined coefficients to approximate the derivative about $x = \alpha$ by

$$\frac{\partial v^+}{\partial x}(\alpha, t_n) = a_v^+ v_{i+1}^n + b_v^+ v_i^n. \tag{2.57}$$

The goal is to find the coefficients $(a_v^+, b_v^+)$ that make the approximation valid, while also obtaining a desirable order of accuracy. In our application, we are interested in making an approximation that yields an error $\mathcal{O}(\Delta x^2)$. We expand the right side of (2.63), noting that $x_i \in \Omega^-$ and $x_{i+1} \in \Omega^+$ to obtain

$$\begin{aligned}\frac{\partial v^+}{\partial x}(\alpha, t) = & a_v^+\left( v^+(\alpha,t) + \frac{\partial v^+}{\partial x}(x_{i+1}-\alpha) + \frac{\partial^2 v^+}{\partial x^2}\frac{(x_{i+1}-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right) + \\ & b_v^+\left( v^-(\alpha,t) + \frac{\partial v^-}{\partial x}(x_i-\alpha) + \frac{\partial^2 v^-}{\partial x^2}\frac{(x_i-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right).\end{aligned} \tag{2.58}$$

We further expand the right hand side to convert the "-" terms to "+" terms, using the jump conditions (2.39),

$$\begin{aligned}\frac{\partial v^+}{\partial t}(\alpha, t) = & a_v^+\left( v^+(\alpha,t) + \frac{\partial v^+}{\partial x}(x_{i+1}-\alpha) + \frac{\partial^2 v^+}{\partial x^2}\frac{(x_{i+1}-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right) + \\ & b_v^+\left( v^+(\alpha,t) + \left(\frac{\beta^-}{\beta^+}\frac{\partial v^+}{\partial x} - \beta^-[f]\right)(x_i-\alpha)\right. \\ & \left. + \left(\frac{\beta^- K^+}{\beta^+ K^-}\frac{\partial^2 v^+}{\partial x^2} - \frac{\beta^-\left[K f_x - g_t\right]}{K^-}\right)\frac{(x_i-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right).\end{aligned} \tag{2.59}$$

We do this so that now we can compare the two sides. Comparing the two sides yields a system of linear equations for the unknown stencil weights for the corrected derivative,$(a_v^+, b_v^+)$ ,

$$\begin{bmatrix} 1 & 1 \\ x_{i+1}-\alpha & \frac{\beta^-}{\beta^+}(x_i-\alpha) \\ \frac{(x_{i+1}-\alpha)^2}{2} & \frac{\beta^- K^+(x_i-\alpha)^2}{2\beta^+ K^-} \end{bmatrix} \begin{bmatrix} a_v^+ \\ b_v^+ \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{2.60}$$

This linear equation can be solved by means of the least squares method. Once we have solved the system for the coefficients, we introduce a correction, $C_i^{v^+}$,

$$\frac{\partial v^+}{\partial x}(\alpha, t_n) = a_v^+ v_{i+1}^n + b_v^+ v_i^n - C_i^{v^+} + \mathcal{O}((a_v^+ + b_v^+)\Delta x^3), \tag{2.61}$$

where

$$C_i^{v^+} = b_v^+\left( -\beta^-[f](x_i-\alpha) - \frac{\beta^-\left[K f_x - g_t\right]}{K^-}\frac{(x_i-\alpha)^2}{2}\right) \tag{2.62}$$

to correct the terms generated by the jump conditions across the interface, resulting in an approximation of order $\mathcal{O}((a_v^+ + b_v^+)\Delta x^3)$.

We now examine the second case of the finite-difference approximation for $\frac{\partial v}{\partial x}$ when $x_i \le x_{i+\frac{1}{2}} \le \alpha \le x_{i+1}$. In this case, $x_{i+\frac{1}{2}} \in \Omega^-$. Therefore we approximate the derivative on the left side of the interface by finding the undermined coefficients $(a_v^-, b_v^-)$ that yields a valid approximation:

$$\frac{\partial v^-}{\partial x}(\alpha, t_n) = a_v^- v_{i+1}^n + b_v^- v_i^n. \tag{2.63}$$

Next we expand the right side:

$$\begin{aligned} \frac{\partial v^-}{\partial x}(\alpha, t) =& a_v^-\left( v^+(\alpha, t) + \frac{\partial v^+}{\partial x}(x_{i+1}-\alpha) + \frac{\partial^2 v^+}{\partial x^2}\frac{(x_{i+1}-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right) + \\ & b_v^-\left( v^-(\alpha, t) + \frac{\partial v^-}{\partial x}(x_i-\alpha) + \frac{\partial^2 v^-}{\partial x^2}\frac{(x_i-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right). \end{aligned} \tag{2.64}$$

We further expand the right hand side to convert the "+" terms to "-" terms, using the jump conditions (2.39),

$$\begin{aligned} \frac{\partial v^+}{\partial t}(\alpha, t) =& a_v^-\left( v^-(\alpha, t) + \left(\frac{\beta^+}{\beta^-}\frac{\partial v^-}{\partial x} + \beta^+[f]\right)(x_{i+1}-\alpha) + \right. \\ & \left. \left(\frac{\beta^+ K^-}{\beta^- K^+}\frac{\partial^2 v^+}{\partial x^2} + \frac{\beta^+\left[K f_x - g_t\right]}{K^+}\right)\frac{(x_{i+1}-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right) \\ & + b_v^-\left( v^-(\alpha, t) + \frac{\partial v^-}{\partial x}(x_i-\alpha) + \frac{\partial^2 v^-}{\partial x^2}\frac{(x_i-\alpha)^2}{2} + \mathcal{O}(\Delta x^3)\right). \end{aligned} \tag{2.65}$$

We do this so that now we can compare the two sides. Comparing the two sides yields a system of linear equations for the unknown stencil weights for the corrected derivative,$(a_v^+, b_v^+)$ ,

$$\begin{bmatrix} 1 & 1 \\ \frac{\beta^+}{\beta^-}(x_{i+1}-\alpha) & (x_i-\alpha) \\ \frac{\beta^+ K^-(x_{i+1}-\alpha)^2}{2\beta^- K^+} & \frac{(x_i-\alpha)^2}{2} \end{bmatrix} \begin{bmatrix} a_v^- \\ b_v^- \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{2.66}$$

This linear equation can be solved by means of the least squares method. Once we have solved the system for the coefficients, we introduce a correction, $C_i^{v^-}$,

$$\frac{\partial v^-}{\partial x}(\alpha, t_n) = a_v^- v_{i+1}^n + b_v^- v_i^n - C_i^{v^+} + \mathcal{O}((a_v^- + b_v^-)\Delta x^3), \tag{2.67}$$

where

$$C_i^{v^-} = a_v^- \left( \beta^+[f](x_{i+1}-\alpha) + \frac{\beta^+\Big[K f_x - g_t\Big]}{2K^+} \frac{(x_{i+1}-\alpha)^2}{2} \right) \tag{2.68}$$

to correct the terms generated by the jump conditions across the interface, resulting in an approximation of order $\mathcal{O}((a_v^+ + b_v^+)\Delta x^3)$.

We now repeat the process of deriving the correction term for $\frac{\partial u}{\partial x}$. The standard Yee scheme uses the spatial stencil points $(x_{i-\frac{1}{2}}, x_i, x_{i+\frac{1}{2}})$ to update the $v$ at $x_i$, for the next time step $t_{n+1}$. Just like in the case for $\frac{\partial v}{\partial x}$, there are two cases to consider for a valid approximation of $\frac{\partial u}{\partial x}$. In the first case, the interface point $x = \alpha$ satisfies

$$x_{i-\frac{1}{2}} \leq \alpha \leq x_i \leq x_{i+\frac{1}{2}}, \tag{2.69}$$

and in the second case,

$$x_{i-\frac{1}{2}} \leq x_i \leq \alpha \leq x_{i+\frac{1}{2}}. \tag{2.70}$$

In the first case our approximation is on the "+" side of the interface, thus we wish to construct a valid second order approximation

$$\frac{\partial u^+}{\partial x} = a_u^+ u_{i+\frac{1}{2}}^{n+\frac{1}{2}} + b_u^+ u_{i-\frac{1}{2}}^{n+\frac{1}{2}}. \tag{2.71}$$

In the second case, because the interface is on the "-" side of the interface, our approximation will take the form

$$\frac{\partial u^-}{\partial x} = a_u^- u_{i+\frac{1}{2}}^{n+\frac{1}{2}} + b_u^- u_{i-\frac{1}{2}}^{n+\frac{1}{2}}. \tag{2.72}$$

Based on whether $x_i$ is on the left or right side of the interface we find the coefficients $(a_u^+, b_u^+)$ or $(a_u^-, b_u^-)$

We omit the details regarding expanding the Taylor series, which was previously shown for $\frac{\partial v}{\partial x}$, we do however write the the linear systems to find the stencil weights for both approximations, and the corresponding correction terms respectively. First we write down the linear system and correction term corresponding with the approximation. The linear system to find the stencil weights $(a_u^+, b_u^+)$ is

$$\begin{bmatrix} 1 & 1 \\ (x_{i+\frac{1}{2}} - \alpha) & \frac{K^+}{K^-}(x_{i-\frac{1}{2}} - \alpha) \\ \frac{(x_{i+\frac{1}{2}} - \alpha)^2}{2} & \frac{\beta^- K^+ (x_{i-\frac{1}{2}} - \alpha)^2}{2\beta^+ K^-} \end{bmatrix} \begin{bmatrix} a_u^+ \\ b_u^+ \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{2.73}$$

The corresponding correction term is

$$C_i^{u^+} = b_u^+ \left( \frac{-[g]}{K^-}(x_{i-\frac{1}{2}} - \alpha) - \frac{\beta^-}{K^-}\left[\frac{g_x}{\beta} - f_t\right]\frac{(x_{i-\frac{1}{2}} - \alpha)^2}{2} \right). \tag{2.74}$$

Next we write down the linear system and correction term associated with the coefficients $(a_u^-, b_u^-)$:

$$\begin{bmatrix} 1 & 1 \\ \frac{K^-}{K^+}(x_{i+\frac{1}{2}} - \alpha) & (x_{i-\frac{1}{2}} - \alpha) \\ \frac{\beta^+ K^- (x_{i+\frac{1}{2}} - \alpha)^2}{2\beta^- K^+} & \frac{(x_{i-\frac{1}{2}} - \alpha)^2}{2} \end{bmatrix} \begin{bmatrix} a_u^- \\ b_u^- \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{2.75}$$

with the corresponding correction term

$$C_i^{u^-} = a_u^- \left( \frac{[g]}{K^+}(x_{i+\frac{1}{2}} - \alpha) + \frac{\beta^+}{K^+}\left[\frac{g_x}{\beta} - f_t\right]\frac{(x_{i+\frac{1}{2}} - \alpha)^2}{2} \right). \tag{2.76}$$

Then based on if the irregular “central grid point” of the spatial derivative approximation is to the left (on the “+” side) or right (on the “-” side), we time step using the corrected scheme,

$$\begin{aligned} u_{i+\frac{1}{2}}^{n+\frac{1}{2}} &= u_{i+\frac{1}{2}}^{n-\frac{1}{2}} - \frac{\Delta t}{\beta_{i+\frac{1}{2}}}\left( a_v^\pm v_{i+1}^n + b_v^\pm v_i^n \right) + \Delta t f_{i+\frac{1}{2}}^n + \frac{\Delta t}{\beta_{i+\frac{1}{2}}} C_i^{v^\pm} \\ v_i^{n+1} &= v_i^n - \Delta t K_i \left( a_u^\pm u_{i+\frac{1}{2}}^{n+\frac{1}{2}} + b_u^\pm u_{i-\frac{1}{2}}^{n+\frac{1}{2}} \right) + \Delta t g_i^n + \Delta t K_i C_i^{u^\pm}. \end{aligned} \tag{2.77}$$

We note that at regular grid points the corrected scheme (2.88) simplifies to the classical Yee scheme (2.53). That is to say that the correction term at regular grid points is zero, and the stencil weights are the same as the Yee scheme.

We can recover the original TM fields by undoing the change of variables:

$$H^y = -u, \quad E^z = -v. \tag{2.78}$$

### 2.3.5 A Numerical Experiment

In this section we conduct a numerical experiment to demonstrate that our method solves the TM equations with discontinuous permittivity and permeability to second order accuracy. We will test our method by using the method of manufactured solution. We propose a solution to the TE equations

$$u(x,t) = v(x,t) = \begin{cases} \sin(x - \frac{1}{3}) + t^3 & x < \frac{1}{3} \\ \sin((x - \frac{1}{3})^2) + t^3 & x \geq \frac{1}{3}, \end{cases} \tag{2.79}$$

with permittivity function,

$$\epsilon = \begin{cases} 7 & x < \frac{1}{3} \\ 8 & x \geq \frac{1}{3}, \end{cases} \tag{2.80}$$

and permeability function

$$\mu = \begin{cases} 4 & x < \frac{1}{3} \\ 5 & x \geq \frac{1}{3}. \end{cases} \tag{2.81}$$

We begin with numerically demonstrating the consistency of our method by calculating the convergence order. For all simulations, we take $\Delta x = \frac{1}{n_x}$, with spatial mesh $\Omega = [0,1]$ and $T_{\text{final}} = 1$. We also take $\Delta t = \frac{\Delta x}{2\sqrt{\frac{K}{\beta}}}$ In Tables 2.4 and 2.5 we calculate the convergence order, using the $||\cdot||_\infty$ to measure the error. We see that the convergence order of our method is oscillating about two. This has been observed in numerous interface problems that are solved with IIM. In [64], it has been rigorously proven that a second order method will show the behavior seen in the tables. The reason for this is because the constant in error term in the Taylor series is dependent on where the interface is with respect to the interface. The average convergence value determines the convergence order of the method. When we average these convergence orders we find that the average convergence value for the TM scheme is 2.01, and the average convergence order has a value of 2.04. Because the average convergence order is close to two, we can conclude that our method is second order accurate (this also shows the method is stable, but we verify this separately).

Next we show that the method is stable. We introduce the following Definition and subsequent

**Table 2.4** Error and Convergence for Solution u.

| $n_x$ | Error | Convergence Order |
|---|---|---|
| 16 | 3.56e-4 | - |
| 32 | 1.10e-4 | 1.69 |
| 64 | 2.06e-5 | 2.41 |
| 128 | 6.64e-6 | 1.63 |
| 256 | 1.26e-6 | 2.39 |
| 512 | 4.11e-7 | 1.61 |
| 1024 | 7.86e-8 | 2.38 |

**Table 2.5** Error and Convergence for Solution v.

| $n_x$ | Error | Convergence Order |
|---|---|---|
| 16 | 2.96e-4 | - |
| 32 | 9.90e-5 | 1.58 |
| 64 | 1.60e-5 | 2.62 |
| 128 | 6.00e-6 | 1.41 |
| 256 | 9.66e-7 | 2.63 |
| 512 | 3.76e-7 | 1.35 |
| 1024 | 5.59e-8 | 2.65 |

Lemma, taken from [96], to discuss the stability of our method:

**Definition 2.3.1.** Suppose a finite difference scheme of the form $\mathbf{Z}^{n+1} = S_k\mathbf{Z}^n + \mathbf{F}^n$, where $\mathbf{F}$ is bounded and the subscript $k$ indicates that the time-stepping matrix is dependent on the time step. The time-stepping matrix $S_k$ is said to be stable if for some $C > 0$,

$$||S_k^n|| \leq C$$

for all $n$ and $k$ such that $0 \leq nk \leq T_{\text{final}}$.

*Lemma* 2.3.1. If all the eigenvalues of $S_k$ satisfy $|\lambda_i(S_k)| \leq 1 + \mathcal{O}(\Delta t)$, and $||\mathbf{F}^n|| \leq C_2$ for some constant independent of $\Delta x$ and $\Delta t$, then the explicit scheme $\mathbf{Z}^{n+1} = S_k\mathbf{Z}^n + \mathbf{F}^n$ is stable where $\mathbf{F}$ is the source vector.

We note in our instance that $\mathbf{F}^n$ is always bounded, therefore we only need to show that the eigenvalues satisfy the condition stability criterion. In Table 2.6 we show the result of numerically calculating the largest eigenvalue for the matrix $S_k$ in magnitude (the spectral radius of $S_k$), which we denote as $\rho[S_k]$ as we continually refine the mesh. The matrix $S_k$ is the discretized differential operator that is obtain by applying our finite-difference method to the infinite dimensional problem.

**Table 2.6** Eigenvalues of time step matrix $S_k$ for different meshes such that $\Delta t = \frac{\Delta x}{2\sqrt{\frac{K}{\beta}}}$, and $\Delta x = \frac{1}{n_x}$.

| $n_x$ | $\rho[S_k]$ |
|---|---|
| 16 | 1.000 |
| 32 | 1.000 |
| 64 | 1.000 |
| 128 | 1.000 |
| 256 | 1.000 |
| 512 | 1.000 |
| 1024 | 1.000 |

We observe that in all cases that the maximum eigenvalue is exactly one. This means that our method is stable. This shows that as long as we satisfy the CFL condition, we obtain stability of the IIM scheme,

Our numerical experiment demonstrates the consistency and the stability of our scheme. Because the state equation is a linear PDE, we invoke the Lax-Equivalence Theorem [64, 66] to establish convergence. Since we have shown numerically that the scheme is consistent and stable, then we can guarantee the convergence numerically. In Figure 2.3 we plot the true and numerical solution for $u, v$ at $t = 1$ on the finest mesh with $n_x = 1024$. We see that indeed the numerical and true solution agree (which is supported by our average convergence order).

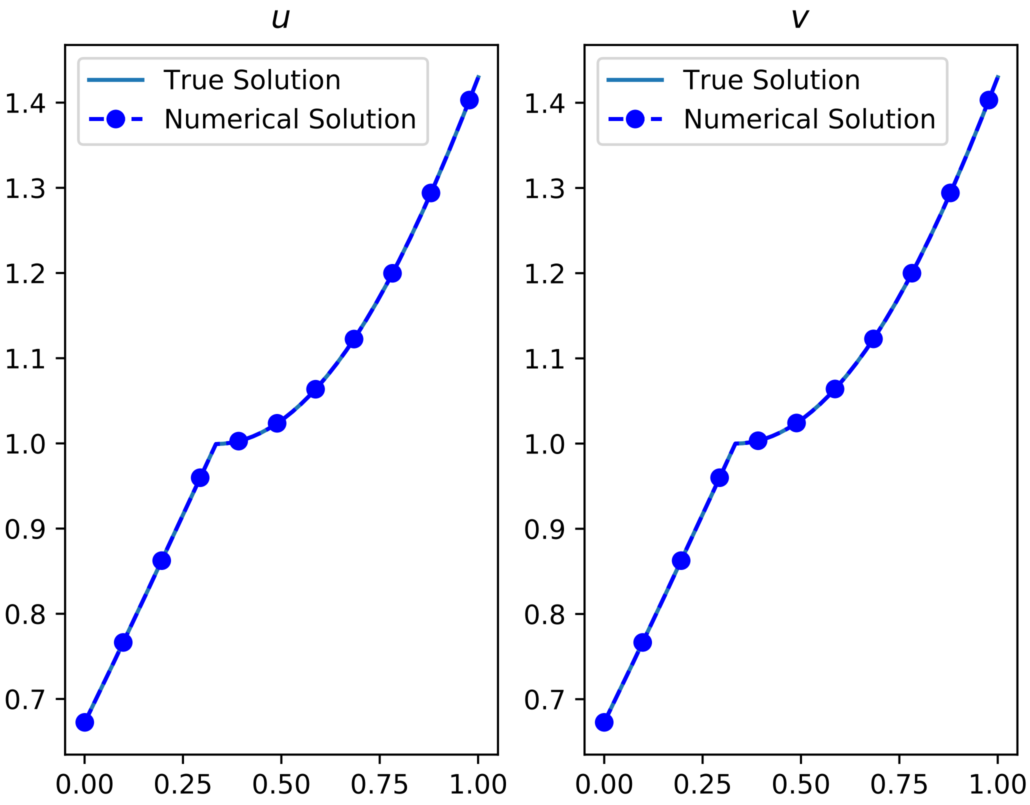


**Figure 2.3** Solution of the TM equation at t=1.

## 2.4 The 1D TE Equations and Yee's scheme

In this Section we introduce the FDTD scheme for the TE equations. This subsection will follow a similar structure to the previous subsection 2.3.3. We direct the reader to Section 2.3.3 for a description of the mesh used in the FDTD method and the notation convention used to describe the scheme. We note that the TE scheme uses the exact same mesh as the TM scheme.

In the TE equations, we approximate the derivatives as follows:

$$\frac{\partial u}{\partial t}(x_i, t_{n+\frac{1}{2}}) = \frac{u_i^{n+1} - u_i^n}{\Delta t} + \mathcal{O}(\Delta t^2), \tag{2.82}$$

$$\frac{\partial v}{\partial t}(x_{i+\frac{1}{2}}, t_n) = \frac{v_{i+\frac{1}{2}}^{n+\frac{1}{2}} - v_{i+\frac{1}{2}}^{n-\frac{1}{2}}}{\Delta t} + \mathcal{O}(\Delta t^2), \tag{2.83}$$

$$\frac{\partial u}{\partial x}(x_{i+\frac{1}{2}}, t_n) = \frac{u_{i+1}^n - u_i^n}{\Delta x} + \mathcal{O}(\Delta x^2), \tag{2.84}$$

$$\frac{\partial v}{\partial x}(x_i, t_{n+\frac{1}{2}}) = \frac{v_{i+\frac{1}{2}}^{n+\frac{1}{2}} - v_{i-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} + \mathcal{O}(\Delta x^2). \tag{2.85}$$

We now define the FDTD (Yee's) scheme for the TE equation:

$$\begin{aligned} u_i^{n+1} &= u_i^n - \frac{\Delta t}{\beta_i \Delta x}\left(v_{i+\frac{1}{2}}^{n+\frac{1}{2}} - v_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right) + \Delta t f_i^{n+\frac{1}{2}} \\ v_{i+\frac{1}{2}}^{n+\frac{1}{2}} &= v_{i+\frac{1}{2}}^{n-\frac{1}{2}} - K_{i+\frac{1}{2}} \frac{\Delta t}{\Delta x}\left(u_{i+1}^n - u_i^n\right) + \Delta t g_{i+\frac{1}{2}}^n. \end{aligned} \tag{2.86}$$

As was the case for the TM scheme, the TE scheme is $\mathcal{O}(\Delta x^2 + \Delta t^2)$ (second order accurate in time and space) for continuous permittivity and permeability. We must also satisfy the CFL condition for stability of the scheme:

$$\max_{\Omega, t}\left(\sqrt{\frac{K}{\beta}}\right)\frac{\Delta t}{\Delta x} \leq 1. \tag{2.87}$$

### 2.4.1 The Modified Yee's Scheme for the 1D TE Equation

We omit the details of deriving the linear systems and correction term for the spatial derivatives in the TE equations because the derivation is exactly the same as was done for the TM equations in subsection 2.3.3. We provide the summary of the scheme as follows: if a spatial gridpoint is irregular then

$$\begin{aligned} u_i^{n+1} &= u_i^n - \frac{\Delta t}{\beta_i}\left(a_v^{\pm} v_{i+\frac{1}{2}}^{n+\frac{1}{2}} + b_v^{\pm} v_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right) + \Delta t f_i^{n+\frac{1}{2}} + \frac{\Delta t}{\beta_i} C_i^{v^{\pm}} \\ v_{i+\frac{1}{2}}^{n+\frac{1}{2}} &= v_{i+\frac{1}{2}}^{n-\frac{1}{2}} - \Delta t K_{i+\frac{1}{2}}\left(a_u^{\pm} u_{i+1}^n + b_u^{\pm} u_i^n\right) + \Delta t g_{i+\frac{1}{2}}^n + \Delta t K_{i+\frac{1}{2}} C_i^{u^{\pm}}. \end{aligned} \tag{2.88}$$

If regular then we use the standard FDTD scheme. The weights and correction term based on the position of the irregular grid point is

$$\begin{bmatrix} 1 & 1 \\ x_{i+\frac{1}{2}}-\alpha & \frac{\beta^-}{\beta^+}(x_{i-\frac{1}{2}}-\alpha) \\ \frac{(x_{i+\frac{1}{2}}-\alpha)^2}{2} & \frac{\beta^-K^+(x_{i-\frac{1}{2}}-\alpha)^2}{\beta^+K-2} \end{bmatrix} \begin{bmatrix} a_v^+ \\ b_v^+ \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 \\ \frac{\beta^+}{\beta^-}(x_{i+\frac{1}{2}}-\alpha) & (x_{i-\frac{1}{2}}-\alpha) \\ \frac{\beta^+K^-(x_{i+\frac{1}{2}}-\alpha)^2}{2\beta^-K^+} & \frac{(x_{i-\frac{1}{2}}-\alpha)^2}{2} \end{bmatrix} \begin{bmatrix} a_v^- \\ b_v^- \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{2.89}$$

where

$$C_i^{v^+} = b_v^+\left(-\beta^-[f](x_{i-\frac{1}{2}}-\alpha) - \frac{\beta^-\left[Kf_x-g_t\right]}{K^-}\frac{(x_{i-\frac{1}{2}}-\alpha)^2}{2}\right), \tag{2.90}$$

$$C_i^{v^-} = a_v^-\left(\beta^+[f](x_{i+\frac{1}{2}}-\alpha) + \frac{\beta^+\left[Kf_x-g_t\right]}{2K^+}\frac{(x_{i+\frac{1}{2}}-\alpha)^2}{2}\right), \tag{2.91}$$

and

$$\begin{bmatrix} 1 & 1 \\ (x_{i+1}-\alpha) & \frac{K^+}{K^-}(x_i-\alpha) \\ \frac{(x_{i+1}-\alpha)^2}{2} & \frac{\beta^-K^+(x_i-\alpha)^2}{2\beta^+K^-} \end{bmatrix} \begin{bmatrix} a_u^+ \\ b_u^+ \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 \\ \frac{K^-}{K^+}(x_{i+1}-\alpha) & (x_i-\alpha) \\ \frac{\beta^+K^-(x_{i+1}-\alpha)^2}{2\beta^-K^+} & \frac{(x_i-\alpha)^2}{2} \end{bmatrix} \begin{bmatrix} a_u^- \\ b_u^- \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{2.92}$$

where

$$C_i^{u^+} = b_u^+\left(\frac{-[g]}{K^-}(x_i-\alpha) - \frac{\beta^-}{K^-}\left[\frac{g_x}{\beta}-f_t\right]\frac{(x_i-\alpha)^2}{2}\right), \tag{2.93}$$

$$C_i^{u^-} = a_u^-\left(\frac{[g]}{K^+}(x_{i+1}-\alpha) + \frac{\beta^+}{K^+}\left[\frac{g_x}{\beta}-f_t\right]\frac{(x_{i+1}-\alpha)^2}{2}\right). \tag{2.94}$$

We note that we can recover the original TE fields by undoing the change of variables:

$$E^y = u \quad H^z = v. \tag{2.95}$$

### 2.4.2 A Numerical Experiment

In this section we conduct a numerical experiment to demonstrate that our method solves the TM equations with discontinuous permittivity and permeability to second order accuracy. We will test our method by using the method of manufactured solution. We propose a solution to the TM equations

$$u(x,t) = v(x,t) = \begin{cases} \sin(x-\frac{1}{3})+t^3 & x<\frac{1}{3} \\ \sin((x-\frac{1}{3})^2)+t^3 & x\geq\frac{1}{3}, \end{cases} \tag{2.96}$$

with permittivity function,

$$\epsilon = \begin{cases} 8 & x<\frac{1}{3} \\ 4 & x\geq\frac{1}{3}, \end{cases} \tag{2.97}$$

and permeability function

$$\mu = \begin{cases} 3 & x<\frac{1}{3} \\ 6 & x\geq\frac{1}{3}. \end{cases} \tag{2.98}$$

We repeat the steps discussed in Section 2.3.5. Using the same spatial and time mesh as we used for the TM numerical experiment. We first report the convergence order for the different mesh sizes in Tables 2.7 and 2.8. The average convergence orders are 2.04 and 2.04 respectively, which means that our method is second order accurate [64].

**Table 2.7** Error and Convergence for Solution u.

| n | Error | Convergence Order |
|---|---|---|
| 16 | 2.90e-4 | - |
| 32 | 1.03e-4 | 1.49 |
| 64 | 1.59e-5 | 2.69 |
| 128 | 6.16e-6 | 1.37 |
| 256 | 9.65e-7 | 2.67 |
| 512 | 3.86e-7 | 1.32 |
| 1024 | 5.99e-8 | 2.68 |

**Table 2.8** Error and Convergence for Solution v.

| n | Error | Convergence Order |
|---|---|---|
| 16 | 2.88e-4 | - |
| 32 | 1.10e-4 | 1.38 |
| 64 | 1.56e-5 | 2.82 |
| 128 | 6.70e-6 | 1.22 |
| 256 | 9.48e-7 | 2.82 |
| 512 | 4.14e-7 | 1.19 |
| 1024 | 5.87e-8 | 2.81 |

We omit the details of the time step matrix, which was shown in Section 2.3.5, because in this case the matrix is the transpose of the matrix that was presented. A square matrix and its transpose matrix share the same eigenvalues [75], however, we still calculate and present the eigenvalues for different mesh sizes in Table 2.9.

**Table 2.9** Eigenvalues of the transpose of the time step matrix $S_k$ for different meshes such that $\Delta t = \frac{\Delta x}{2\sqrt{\frac{K}{\beta}}}$, and $\Delta x = \frac{1}{n_x}$.

| $n_x$ | $\rho[S_k^T]$ |
|---|---|
| 16 | 1.000 |
| 32 | 1.000 |
| 64 | 1.000 |
| 128 | 1.000 |
| 256 | 1.000 |
| 512 | 1.000 |
| 1024 | 1.000 |

Since we have numerically established the consistency and stability of our method, we again can invoke the Lax-Equivalence Theorem [64, 66] to establish the numerical convergence of the numerical solution to the true solution of the problem. In Figure 2.4 we plot the true and numerical solution for $u, v$ at $t = 1$ on the finest mesh with $n_x = 1024$. We see that indeed the numerical and true solution agree (which is supported by our average convergence order).

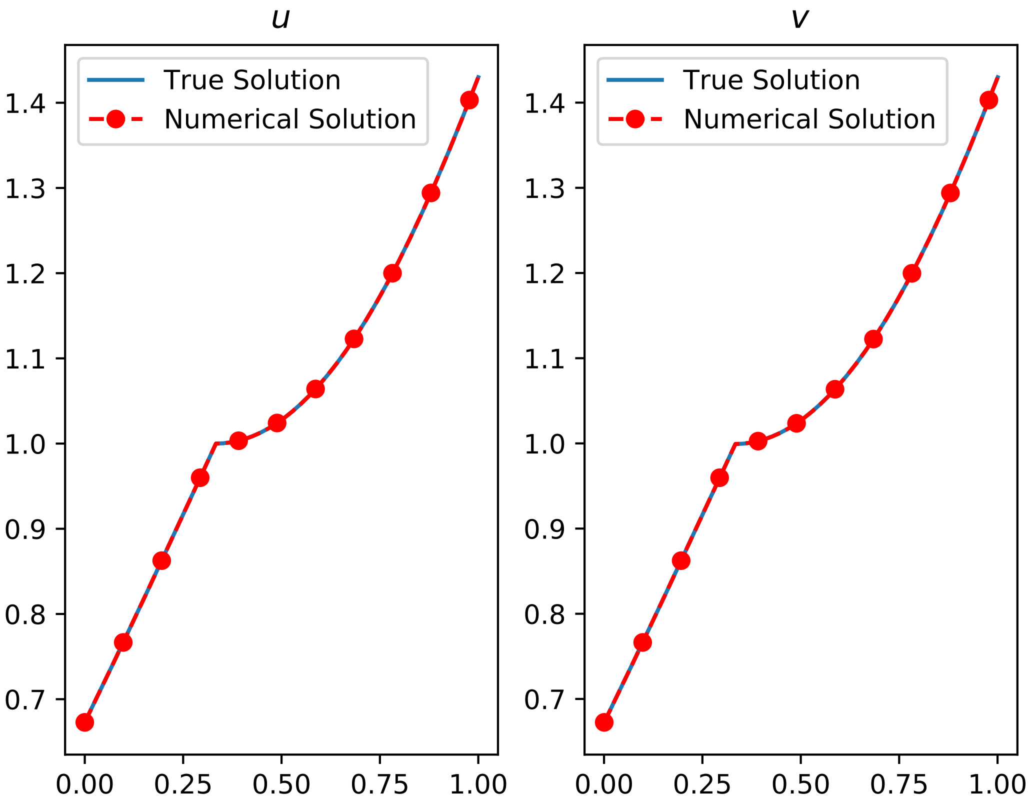


**Figure 2.4** Solution of the TE equation at t=1.

CHAPTER

# 3

# AN AUGMENTED IMMERSED INTERFACE METHOD TO SOLVE MAXWELL'S EQUATIONS IN ONE AND TWO DIMENSIONAL SPACE

## 3.1 Chapter Outline

In this chapter we introduce an ADI-Yee's method combined with the IIM[64]. In chapter 2, we introduced a direct IIM that took advantage of Yee's scheme. We cannot apply the same idea for 2D Maxwell's equations because the jump conditions cannot be isolated for i.e we may know the jump, $[u_{xx}+u_{yy}]$, but wouldn't be able to establish what the direct jumps conditions were from the previous jump condition i.e $[u_{xx}], [u_{yy}]$. Instead we introduce an ADI-Yee's scheme that takes advantage of the augmented IIM [64]. In this perspective, we introduce augmented variables to take the role of the value of unknown jump conditions we cannot explicitly derive from the PDE. We first present the idea for the 1D Maxwell's equations, then generalize this framework to solve 2D Maxwell's equations where the interface can be curved. As in chapter 2, our IIM method for solving Maxwell's equations is second order in time and space. In 1D we prove that our method is stable and in 2D we show numerical results which establish the stability, without proof. In all situations we observe as long as the CFL condition is satisfied this is sufficient to ensure the numerical stability of our method. We provide two numerical experiments to demonstrate that the convergence order of our method is two. The convergence order being two indicates the method admits a second order truncation order in time and space, and is stable.

## 3.2 The ADI-Yee's Scheme for One-Dimensional Maxwell's equation

We first use the one-dimensional example to illustrate the idea. Note that if we let

$$u = -H^y, \qquad p = E^z, \qquad \rho = \mu, \qquad K = \frac{1}{\epsilon}, \tag{3.1}$$

then the one-dimensional Maxwell's equations (TM) can be written in the form of the one-dimensional acoustic wave equations

$$u_t + \frac{1}{\rho} p_x = 0 \quad (\text{or} \quad f(x,t)), \qquad 0 < t, \quad a < x < b, \tag{3.2}$$

$$p_t + K u_x = 0 \quad (\text{or} \quad g(x,t)), \qquad 0 < t, \quad a < x < b, \tag{3.3}$$

with initial conditions $u(x,0)$ and $p(x,0)$ and appropriate boundary conditions, where $u_t = \frac{\partial u}{\partial t}$, $u_x = \frac{\partial u}{\partial x}$, and so on. We add source terms $f$ and $g$ to the right-hand side so that we can construct analytic solutions more easily.

### 3.2.1 Yee's scheme for a regular one-dimensional problems

Given a mesh $x_i = a + ih$, $i = 0, 1, \cdots, N$, $h = (b-a)/N$, and a time step size $\Delta t$, $t^k = k\Delta t$, Yee's scheme for the one-dimensional problem has the following form:

$$u_{i+1/2}^{k+1/2} = u_{i+1/2}^{k-1/2} - \frac{1}{\rho}\frac{\Delta t}{h}\left(p_{i+1}^k - p_i^k\right) + \Delta t\, f_{i+1/2}^k \tag{3.4}$$

$$p_i^{k+1} = p_i^k - \frac{K\Delta t}{h}\left(u_{i+1/2}^{k+1/2} - u_{i-1/2}^{k+1/2}\right) + \Delta t\, g_i^{k+1/2}, \tag{3.5}$$

where $u_{i+1/2}^{k+1/2} \approx u(x_i + h/2, t^k + \Delta t/2)$ is an approximation to the solution and so on. The discretization is explicit and second order accurate $O((\Delta t)^2 + h^2)$. Note also that the discretization is a second order time splitting scheme at $(x_{i+1/2}, t^k)$ for the first equation, and at $(x_i, t^{k+1/2})$ for the second equation. The CFL condition is

$$\max_{\Omega,t}\left\{\sqrt{\frac{K}{\rho}}\frac{\Delta t}{h}\right\} \le 1, \quad \text{or} \quad \frac{\Delta t}{h} \le \min_{\Omega,t}\sqrt{\frac{\rho}{K}}. \tag{3.6}$$

The finite difference scheme can be written as

$$\mathbf{Z}^{k+1} = S_k \mathbf{Z}^k + \mathbf{F}^k, \tag{3.7}$$

where $\mathbf{Z}^k = \left[ u_{i+1/2}^{k-1/2},\ p_i^k \right]_{i=0}^{N}$ is the solution vector at time $t^k$ and so on. When the CFL condition is satisfied, the spectral radius of $S$ satisfies $\rho(S) \le 1 + C\Delta t$ for the convergence, where $C$ is a constant independent of $h$ and $\Delta t$.

### 3.2.2 Jump Conditions and the Modified Finite-Difference Approximations

Now we assume that $\rho$ and $K$ are piecewise constants and have finite jump discontinuities across a fixed interface $x = \alpha$ in the solution domain. We refer to the jump conditions in 2.3.2, which we again use here. The jumps are defined as the differences of the related limiting values from each side of the interface.

Numerically there are several approaches in discretizing the partial differential equations accurately. One approach is to modify the finite difference stencil and coefficients as in [23, 110], which often leads to large finite difference stencils and compromises the stability analysis. Another approach is to keep the finite difference stencil unchanged but introduce augmented variable(s) and equations as in [65]. For the one-dimensional case, the differences between different approaches are not significant but will be in two dimensions. We introduce an ADI-Yee's method for one-dimensional problems and carry out the consistency and stability analysis. The fundamental discretization formulas are also used for two-dimensional problems.

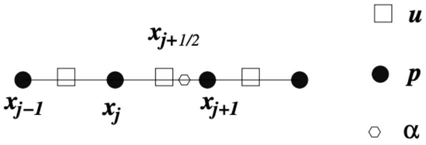


**Figure 3.1** A diagram of one-dimensional stencil and an interface $\alpha$.

If we know the jump condition $[u_x] = q_1$ and $[u_{xx}] = q_2$, which we are unable to derive explicitly from Maxwell's equations, then we can get a second order accurate discretization from the following lemma.

*Lemma* 3.2.1. Let $u(x)$ be a piecewise $C^3$ function in $(x_j, \alpha)$ and $(\alpha, x_{j+1})$ with finite jumps $[u]$, $[u_x]$, and $[u_{xx}]$ at $\alpha$, where $x_j \le \alpha < x_{j+1}$, $x_{j+1} - x_j = h$. Then we have

$$\frac{u(x_{j+1}) - u(x_j)}{h} = \begin{cases} u'\left(x_{j+\frac{1}{2}}\right) + C_j + O(h^2) & \text{if } x_j + \frac{h}{2} \le \alpha, \\ u'\left(x_{j+\frac{1}{2}}\right) - C_j + O(h^2) & \text{if } \alpha < x_j + \frac{h}{2}, \end{cases} \tag{3.8}$$

where $C_j$ is given by

$$C_j = \frac{[u] + (x_{j+1} - \alpha)[u_x] + \frac{1}{2}(x_{j+1} - \alpha)^2 [u_{xx}]}{h}, \tag{3.9}$$

as seen in [64].

*Proof.* We prove the first case in which $x_j + h/2 \le \alpha$. Define the extension of $u(x)$ from the left side of $\alpha$ to the right as

$$u^e(x) = u^- + (x - \alpha) u_x^- + \frac{1}{2}(x - \alpha)^2 u_{xx}^- + \frac{1}{6}(x - \alpha)^3 u_{xxx}^-, \qquad x \ge \alpha. \tag{3.10}$$

Then we have

$$\frac{u(x_{j+1}) - u(x_j)}{h} = \frac{u^+ + (x_{j+1} - \alpha) u_x^+ + \frac{1}{2}(x_{j+1} - \alpha)^2 u_{xx}^+ + O(h^3) - u(x_j)}{h} \tag{3.11}$$

$$= \frac{u^- + (x_{j+1} - \alpha) u_x^- + \frac{1}{2}(x_{j+1} - \alpha)^2 u_{xx}^- + \frac{1}{6}(x_{j+1} - \alpha)^3 u_{xxx}^- + O(h^3) - u(x_j)}{h} \tag{3.12}$$

$$+ \frac{[u] + (x_{j+1} - \alpha)[u_x] + \frac{1}{2}(x_{j+1} - \alpha)^2 [u_{xx}]}{h} \tag{3.13}$$

$$= \frac{u^e(x_{j+1}) - u(x_j)}{h} + C_j + O(h^2) = u'\left(x_{j+\frac{1}{2}}\right) + C_j + O(h^2) \tag{3.14}$$

□

The proof for the case $x_j \le \alpha < x_j + h/2$ is similar.

### 3.2.3 Interpolation for the Interface Conditions

We can approximate $[u_x]$ and $[p_x]$ explicitly as it was done in [23, 110], which will destroy the structure of Yee's scheme and will lead to a small CFL number. In our approach, we solve $[u_x]$ and $[p_x]$ implicitly and we will show that the CFL condition is nearly unchanged with a possible order $O(h^2)$ perturbation. The augmented variables should be chosen such that the jump condition $[K u_x] = [g]$ is satisfied at $t^{k+\frac{1}{2}}$ while $\left[\frac{1}{\rho} p_x\right] = [f]$ is satisfied at $t^{k+1}$.

In our algorithm, $[u_x]$ and $[p_x]$ are used as intermediate variables so that the interface conditions can be satisfied, which is done through some interpolation formulas from function values. A second order accurate interpolation formula is stated in the following lemma.

*Lemma* 3.2.2. Let $u(x)$ be a piecewise $C^3$ function in $(x_{j-1},\alpha)$ and $(\alpha,x_{j+1})$ with finite jumps $[u]$, $[u_x]$, and $[u_{xx}]$ at $\alpha$, where $x_j \le \alpha < x_{j+1}$, $x_j - x_{j-1} = x_{j+1} - x_j = h$. Then we have

$$u_x^-(\alpha) = \gamma_1 u(x_{j-1}) - (\gamma_1+\gamma_2)u(x_j) + \gamma_2 u(x_{j+1}) + \bar{C}_j + O(h^2), \tag{3.15}$$

where the coefficients and the correction term are given by

$$\gamma_1 = \frac{\alpha - x_j}{h^2} - \frac{1}{2h}, \qquad \gamma_2 = \frac{\alpha - x_j}{h^2} + \frac{1}{2h}; \tag{3.16}$$

$$\bar{C}_j = -\gamma_2\left([u] + (x_{j+1}-\alpha)[u_x] + \frac{1}{2}(x_{j+1}-\alpha)^2[u_{xx}]\right). \tag{3.17}$$

*Proof.* Similar to the proof in Lemma 3.2.1, we use Taylor expansion of $u(x_{j-1})$, $u(x_j)$, and $u(x_{j+1})$ at $\alpha$ from each side of the interface to have

$$\gamma_1 u(x_{j-1}) - (\gamma_1+\gamma_2)u(x_j) + \gamma_2 u(x_{j+1}) + \bar{C}_j \tag{3.18}$$

$$= u_x^-(\alpha) + \gamma_2\left([u] + (x_{j+1}-\alpha)[u_x] + \frac{1}{2}(x_{j+1}-\alpha)^2[u_{xx}]\right) + \bar{C}_j + O(h^2), \tag{3.19}$$

which leads to the lemma immediately. Note that $\bar{C}_j = -\gamma_2 C_j h$ where $C_j$ is defined in 3.9.

The interface condition $[K u_x] = [g]$ is discretized using the interpolation scheme above as follows. We use the following approximation

$$K^+ u_x^+ - K^- u_x^- = K^+\left(u_x^- + [u_x]\right) - K^- u_x^- = [K]u_x^- + K^+[u_x] = [g]_\alpha, \tag{3.20}$$

$$\Longrightarrow [u_x] = \frac{[g]_\alpha - [K]u_x^-}{K^+} \tag{3.21}$$

$$= \frac{[g]_\alpha}{K^+} - \frac{[K]}{K^+}\left(\gamma_1 u(x_{j-1}) - (\gamma_1+\gamma_2)u(x_j) + \gamma_2 u(x_{j+1}) + \bar{C}_j\right) + O(h^2). \tag{3.22}$$

□

From the expressions above, we have the following lemma.

*Lemma* 3.2.3. Let $u(x)$ be a piecewise $C^3$ function in $(x_{j-1},\alpha)$ and $(\alpha,x_{j+1})$ with finite jumps $[u]$, $[u_x]$, and $[u_{xx}]$ at $\alpha$, where $x_j \le \alpha < x_{j+1}$, $x_{j+1} - x_j = x_j - x_{j-1} = h$. Then we have

$$[u_x] = \frac{1}{1 - \frac{[K](x_{j+1}-\alpha)\gamma_2}{K^+}}\left(\frac{[g]_\alpha}{K^+} - \frac{[K]}{K^+}\left(\gamma_1 u(x_{j-1}) - (\gamma_1+\gamma_2)u(x_j)\right.\right. \tag{3.23}$$

$$\left.\left. + \gamma_2 u(x_{j+1}) + \tilde{C}_j\right)\right) + O(h^2) \tag{3.24}$$

$$= \beta_1 u(x_{j-1}) - (\beta_1+\beta_2)u(x_j) + \beta_2 u(x_{j+1}) + \beta_3[g]_\alpha + \beta_4\tilde{C}_j + O(h^2), \tag{3.25}$$

for some constants $\beta_j$ $(j = 1,2,3,4)$ that are determined from above expression, where

$$\tilde{C}_j = -\gamma_2 \left( [u] + \frac{1}{2}(x_{j+1} - \alpha)^2 [u_{xx}] \right). \tag{3.26}$$

### 3.2.4 The ADI-Yee's Scheme for the One-Dimensional Maxwell's Equations with an Interface

To deal with the discontinuities in the coefficients and maintain the structure of Yee's scheme, we need to add correction terms when the interface falls in the finite difference stencil. Take the diagram in Figure 3.1 in which $x_{j+\frac{1}{2}} \le \alpha < x_{j+1}$, for example, only the finite difference formulas for $u_{j+\frac{1}{2}}^{k+\frac{1}{2}}$ and $p_{j+1}^{k+1}$ need to be modified as

$$u_{j+1/2}^{k+1/2} = u_{j+1/2}^{k-1/2} - \frac{1}{\rho}\frac{\Delta t}{h}\left(p_{j+1}^k - p_j^k - C_{j+1/2}^p\right) + \Delta t\, f_{j+1/2}^k, \tag{3.27}$$

$$p_{j+1}^{k+1} = p_{j+1}^k - \frac{K\Delta t}{h}\left(u_{j+3/2}^{k+1/2} - u_{j+1/2}^{k+1/2} + C_{j+1}^u\right) + \Delta t\, g_{j+1}^{k+1/2}, \tag{3.28}$$

where (note that $[u] = 0$ and $[p] = 0$)

$$C_{j+1/2}^p = (x_{j+1} - \alpha)\left[p_x^k\right] + \frac{1}{2}(x_{j+1} - \alpha)^2\left[p_{xx}^k\right], \tag{3.29}$$

$$C_{j+1}^u = (x_{j+1/2} - \alpha)\left[u_x^{k+1/2}\right] + \frac{1}{2}(x_{j+1/2} - \alpha)^2\left[u_{xx}^{k+1/2}\right]. \tag{3.30}$$

For practical reasons, we approximate the second order jumps explicitly. The modified finite difference scheme can be written as

$$\mathbf{Z}^{k+1} = S_k \mathbf{Z}^k + B\mathbf{Q}^{k+1} + \mathbf{F}^k, \tag{3.31}$$

where $\mathbf{Z}^k = \left[u_{j+1/2}^{k-1/2}, p_j^k\right]_{j=0}^N$ is the $k$-th finite difference solution vector, $\mathbf{F}^k$ is the modified source term due to contributions from the second order jump corrections, and $\mathbf{Q}^{k+1}$ has only two non-zero entries corresponding to $\left[p_x^k\right]$ and $\left[u_x^{k+1/2}\right]$. The so-called augmented variable $\mathbf{Q}^{k+1}$ should be chosen such that the interface conditions in 2.3.2: at time $t^{k+1}$ for the first equation, and $t^{k+1/2}$ for the second equation, are satisfied. In discretization, $\mathbf{Q}^{k+1}$'s two components have the expressions like (3.23). Thus the discretization of the interface conditions can be written as

$$R\mathbf{Z}^{k+1} + G\mathbf{Q}^{k+1} - \mathbf{F}_2^k = 0. \tag{3.32}$$

If we put the two equations 3.31 and 3.32 together, we get

$$\begin{bmatrix} I & -B \\ R & G \end{bmatrix}\begin{bmatrix} \mathbf{Z}^{k+1} \\ \mathbf{Q}^{k+1} \end{bmatrix} = \begin{bmatrix} S_k\mathbf{Z}^k + \mathbf{F}^k \\ \mathbf{F}_2^k \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1^k \\ \mathbf{F}_2^k \end{bmatrix}. \tag{3.33}$$

Eliminating $\mathbf{Z}^{k+1}$ in (3.33) gives the Schur complement equation for $\mathbf{Q}$

$$(G + RB)\mathbf{Q}^{k+1} = \mathbf{F}_2^k - R\mathbf{F}_1^k. \tag{3.34}$$

For fixed interface and time step size $\Delta t$, the matrix $(G + RB)$ is also fixed. Thus, we can use one step $P(G + RB) = LU$ decomposition and find subsequent solutions $\mathbf{Q}^{k+1}$ rather easily, where $P$ is a permutation matrix.

### 3.2.5 The Stability Analysis for the One-Dimensional Algorithm

If there is no interface, then the CFL condition for Yee's scheme is $\Delta t/h \le \sqrt{\rho/K}$, see [25, 47, 76]. Since we treat the correction terms implicitly, we do not expect that the CFL condition will change as verified in our numerical tests. Note that, for the first order linear system of partial differential equations with interfaces, second order derivative corrections are not so important since they are $O(h^2)$ terms. In this work, to further simplify the new algorithms, we treat the jumps in the second order derivatives explicitly. Thus, if we treat $[u_{xx}]$ and $[p_{xx}]$ as known source terms, then for one-dimensional problems, we can find an explicit expression of $\mathbf{Q}^{k+1}$ using (3.23) as

$$[u_x]^{k+\frac{1}{2}} = \beta_1 u_{j-1/2}^{k+\frac{1}{2}} - (\beta_1 + \beta_2) u_{j+1/2}^{k+\frac{1}{2}} + \beta_2 u_{j+3/2}^{k+\frac{1}{2}} + \bar{C}^{k+\frac{1}{2}}, \tag{3.35}$$

where the correction term $\bar{C}^{k+\frac{1}{2}}$ depends on jumps $[g]^{k+\frac{1}{2}}$ and $[u_{xx}]^{k+\frac{1}{2}}$ etc. There is a similar relation for $[p_x]$. Our stability analysis is based on Lemma 2.3.1. Note that the stability condition in Lemma 2.3.1 is also sufficient for the convergence for any initial data and $\mathbf{F}^0$. For Yee's scheme without an interface, the condition is satisfied if the CFL condition is met. Now we consider the ADI-Yee's scheme when there is an interface $\alpha$. In this case, we can find the entries of $B$ in (3.31) explicitly for the one-dimensional problem. For instance, assume that $x_{j+\frac{1}{2}} \le \alpha < x_{j+1}$; then the entries of $B$ are zero except for a 6-by-6 block corresponding to
$[u_{j-1/2}^{k+1/2},\ u_{j+1/2}^{k+1/2},\ u_{j+3/2}^{k+1/2},\ p_j^{k+1},\ p_{j+1}^{k+1},\ p_{j+2}^{k+1}]$, see Figure 3.1 for an illustration. In fact, from Lem-

mas 3.2.2 and 3.2.3 we know that

$$\alpha_1 = C_1\left(\frac{\alpha - x_j}{h^2} - \frac{1}{2h}\right)(x_{j+1} - \alpha), \tag{3.36}$$

$$\alpha_2 = C_1\left(\frac{\alpha - x_j}{h^2} + \frac{1}{2h}\right)(x_{j+1} - \alpha), \tag{3.37}$$

$$\beta_1 = C_2\left(\frac{\alpha - x_{j+\frac{1}{2}}}{h^2} - \frac{1}{2h}\right)\left(x_{j+1} + \frac{h}{2} - \alpha\right), \tag{3.38}$$

$$\beta_2 = C_2\left(\frac{\alpha - x_{j+\frac{1}{2}}}{h^2} + \frac{1}{2h}\right)\left(x_{j+1} + \frac{h}{2} - \alpha\right), \tag{3.39}$$

where $C_1 = \frac{1}{\rho}\frac{\Delta t}{h} \leq 1$ and $C_2 = \frac{K\Delta t}{h} \leq 1$ from the usual CFL constraints of Yee's scheme. Using the relation (3.23) we can write the ADI-Yee's Scheme as

$$(I - E)\mathbf{Z}^{k+1} = S_k\mathbf{Z}^k + \mathbf{F}^k. \tag{3.40}$$

We note that $E$ is a matrix of all zeros, except in certain diagonal blocks. These special blocks correspond to 6-by-6 matrices, which we denote as $E_j$, for each irregular grid point with index $j$. The block matrix $E_j$ takes the form

$$E_j = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_1 & -(\alpha_1+\alpha_2) & \alpha_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \beta_1 & -(\beta_1+\beta_2) & \beta_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.41}$$

The eigenvalues of $I_6 - E_j$ are $\lambda_i < 1$, $i = 3, \cdots, 6$, due to the stencil weights associated with the Yee's scheme [47], except for two listed below that we find using Maple:

$$\lambda_1 = \frac{\left(-h^2 + \sqrt{C_1C_2\left(-2x_j - 3h + 2\alpha\right)(2\alpha - 2x_j - h)(\alpha - x_j)^2}\right)h^2}{4C_1C_2\left(\alpha - x_j - h/2\right)\left(\alpha - 3h/2 - x_j\right)\left(\alpha - x_j\right)^2 - h^4}, \tag{3.42}$$

$$\lambda_2 = -\frac{\left(-h^2 + \sqrt{C_1C_2\left(-2x_j - 3h + 2\alpha\right)(2\alpha - 2x_j - h)(\alpha - x_j)^2}\right)h^2}{4C_1C_2\left(\alpha - x_j - h/2\right)\left(\alpha - 3h/2 - x_j\right)\left(\alpha - x_j\right)^2 - h^4}. \tag{3.43}$$

The stability of the ADI-Yee's scheme is stated in the following theorem.

*Theorem* 3.2.4. If the conditions of Lemma 2.3.1 are satisfied, and $[u] = 0$, $[p] = 0$; and $[u_{xx}]$ and $[p_{xx}]$ are known (exact) quantities, then the ADI-Yee's scheme for one-dimensional Maxwell interface problems is stable, and thus is also convergent.

*Proof.* From the relation (3.23) we can write

$$(I-E)\mathbf{Z}^{k+1} = S_k\mathbf{Z}^k + \mathbf{F}^k. \tag{3.44}$$

Using Maple shows that $I_6 - E_j$ is invertible and its eigenvalues are bounded by $(1+C\Delta t)$ assuming $\Delta t \sim h$.

It is easy to find the extreme values of $|\lambda_1|$ and $|\lambda_2|$ above for $x_{j+\frac{1}{2}} \le \alpha < x_{j+1}$ as a single function of $\alpha$. We use Maple again to obtain that $|\lambda_i| = 1$, $i = 1,2$. Note that when $\alpha = x_{j+\frac{1}{2}}$, we have exactly $\lambda_1 = 1$.

For The ADI-Yee's scheme, $S_k$ can be written $S_k = I + \bar{E}_k$ and $\|\bar{E}_k\| \le C\Delta t$ for some constant $C$, and the structure of $\bar{E}_k$ is determined by the Yee stencil weights. Using eigenvalue perturbation theory, we know that the eigenvalues of $(I-E)^{-1} S_k$ are those eigenvalues of $(I-E)^{-1}$ with an order of $\Delta t$ perturbations, that is, $\lambda_i((I-E)^{-1} S_k) = \lambda_i((I-E)^{-1}) + O(\Delta t)$. Thus, there is a matrix norm such that $\|(I-E)^{-1} S_k\| \le 1 + C\Delta t$, which guarantees the stability of the ADI-Yee's scheme. □

We use an explicit one-sided finite difference to approximate second order derivative jumps of $[u_{xx}]$ and $[p_{xx}]$. Since the second order derivative jumps are high order terms $(O(h^2))$, they have little effect on the stability. In one space dimension, whether we treat second order jump conditions explicitly or implicitly does not make much difference, but it will for two-dimensional problems.

## 3.3 The ADI-Yee's scheme for the two-dimensional Maxwell's equation with an Interface

In two space dimensions, Maxwell's equations for the TM waves are

$$\frac{\partial H^x}{\partial t} + \frac{1}{\mu}\frac{\partial E^z}{\partial y} = 0, \tag{3.45}$$

$$\frac{\partial H^y}{\partial t} - \frac{1}{\mu}\frac{\partial E^z}{\partial x} = 0, \tag{3.46}$$

$$\frac{\partial E^z}{\partial t} + \frac{1}{\epsilon}\left(\frac{\partial H^x}{\partial y} - \frac{\partial H^y}{\partial x}\right) = 0. \tag{3.47}$$

The jump conditions are

$$[\boldsymbol{\tau}\cdot\mathbf{H}] = 0, \qquad \left[\mathbf{n}\cdot\mu\mathbf{H}\right] = 0, \qquad [E^z] = 0, \tag{3.48}$$

where $\mathbf{n}$ is the normal vector to the interface and $\boldsymbol{\tau}$ is the tangent vector to the interface curve. Again, we assume that all the coefficients are piecewise constants that can have a finite jump across the interface Γ. Note that if we let

$$u = -H^y, \qquad v = H^x, \qquad p = E^z, \qquad \rho = \mu, \qquad K = \frac{1}{\epsilon}, \tag{3.49}$$

then the two-dimensional Maxwell's equations (TM) can be written in the form of the two-dimensional acoustic wave equations

$$\frac{\partial v}{\partial t} + \frac{1}{\rho}\frac{\partial p}{\partial y} = 0, \tag{3.50}$$

$$\frac{\partial u}{\partial t} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0, \tag{3.51}$$

$$\frac{\partial p}{\partial t} + K\left(\frac{\partial v}{\partial y} + \frac{\partial u}{\partial x}\right) = 0, \tag{3.52}$$

with given initial and boundary conditions.

The jump conditions now become

$$\left[\boldsymbol{\tau}\cdot\begin{pmatrix} v \\ -u \end{pmatrix}\right] = 0, \quad \left[\mathbf{n}\cdot\rho\begin{pmatrix} v \\ -u \end{pmatrix}\right] = 0, \quad \left[K\left(v_y + u_x\right)\right] = 0, \quad \left[\frac{1}{\rho}p_n\right] = 0, \tag{3.53}$$

along with $\left[p\right] = 0$. More discussion regarding the jump relations can be found in [23, 108].

### 3.3.1 Augmented Variables and the Enlarged System

To use the ADI dimensional splitting approach for Maxwell's equations with discontinuities across a material interface $\Gamma = (X(s), Y(s))$, where $s$ is a parameter on the interface, say the arc-length, we introduce several intermediate variables, also called augmented variables. We set

$$q_1(s,t) = \left[\mathbf{n}\cdot\begin{pmatrix} v \\ -u \end{pmatrix}\right], \quad q_2(s,t) = \left[\frac{\partial u}{\partial n}\right], \quad q_3(s,t) = \left[\frac{\partial v}{\partial n}\right], \quad q_4(s,t) = \left[\frac{\partial p}{\partial n}\right]. \tag{3.54}$$

If these variables are known, then we can discretize Maxwell's equations dimension by dimension as in the one-dimensional case. That is why the new method is called the ADI-Yee's scheme.

Note that since we know $[\boldsymbol{\tau}\cdot[u, -v]^T]$, we can solve for $[u]$ and $[v]$ in terms of the known normal and tangential directions. Note also that there are different ways in introducing augmented variables. The main motivation of our choice is that we can still use Yee's scheme near and at the interface with modified right-hand sides.

Given $q_1, q_2, q_3$, and $q_4$ with required regularity, for example, $q_1 \in C(\Gamma^2)$, $q_2, q_3, q_4 \in C(\Gamma)$, we can solve Maxwell's equations (3.45) or (3.50) to get the solution $(u, v, p)$ that depends on $q_1$, $q_2$, $q_3$, and $q_4$. The solution in general does not satisfy the original jump conditions (3.53). Thus we have an enlarged system: the original Maxwell's equations (3.45) or (3.50), the original jump conditions (3.53), and the newly introduced interface jump conditions eq3.54.

### 3.3.2 Equivalence of the Original and Enlarged Systems

The wellposedness of the Maxwell initial and boundary value problems is well established with consistent initial and boundary conditions [47]. Assume that the original initial and boundary value problem of Maxwell's equations (3.50) and the jump conditions (3.53) has a unique solution [86, 108]. Then the solution is also the solution of the enlarged system since $q_1, \cdots, q_4$ are all from the solution and their derivatives. The existence of the solution of the enlarged system can be established easily. On the other hand, if there is another set of solution $(u, v, p)$, then it will satisfy Maxwell's equations (3.50) and the jump conditions (3.53) in addition to the augmented relations (3.54). Thus from the uniqueness of the original initial and boundary value problem of Maxwell's equations (3.50) and the jump conditions (3.53), the solution must be the solution to the original problem.

### 3.3.3 The ADI-Yee's scheme in two dimensions

We assume that the domain $\Omega$ is rectangular, say $[a, b] \times [c, d]$. We use a uniform $M \times N$ grid with $x_i = a + ih$, $y_j = c + jh$, $h = (b-a)/M = (d-c)/N$, for simplicity of the presentation.

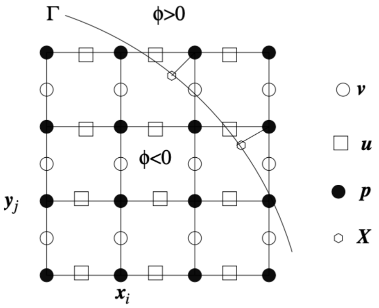


**Figure 3.2** A diagram of a two-dimensional stencil, an interface Γ, and orthogonal projections where the augmented variables are defined.

We use the transverse magnetic (TM) case to illustrate our algorithm. According to Yee's scheme, we use $p_{ij}^k \approx p(x_i, y_j, t^k)$ as the finite difference approximation at a grid point $(x_i, y_j)$ and the time level $t^k$, and similarly for $u_{i+1/2,j}^{k+1/2}$ and $v_{i,j+1/2}^{k+1/2}$. Yee's scheme for non-interface problems is the following

$$v_{i,j+1/2}^{k+1/2}=v_{i,j+1/2}^{k-1/2}-\frac{1}{\rho}\frac{\Delta t}{h_y}\left(p_{i,j+1}^{k}-p_{ij}^{k}\right), \tag{3.55}$$

$$u_{i+1/2,j}^{k+1/2}=u_{i+1/2,j}^{k-1/2}-\frac{1}{\rho}\frac{\Delta t}{h_x}\left(p_{i+1,j}^{k}-p_{ij}^{k}\right), \tag{3.56}$$

$$p_{ij}^{k+1}=p_{ij}^{k}-K\Delta t\left(\frac{v_{i,j+1/2}^{k+1/2}-v_{i,j-1/2}^{k+1/2}}{h_y}+\frac{u_{i+1/2,j}^{k+1/2}-u_{i-1/2,j}^{k+1/2}}{h_x}\right). \tag{3.57}$$

We use a level set representation for an interface $\Gamma$ in the domain. That is, the interface is the zero level of a Lipschitz continuous function $\varphi(x,y)$. For example, for a unit circle interface $x^2+y^2=1$, one of (and the ideal) level set functions is $\varphi(x,y)=\sqrt{x^2+y^2}-1$. In general, the signed distance function is recommended for numerical computations. In discretization, the level set function is determined explicitly by its values at the grid points $\varphi_{ij}=\varphi(x_i,y_j)$, for example, for the unit circle interface, $\varphi_{ij}=\sqrt{x_i^2+y_j^2}-1$.

We classify a grid point $(x_i,y_j)$ to be *regular* or *irregular* using the level set function. At a grid point $(x_i,y_j)$, define

$$\varphi^{max}=\max\left\{\varphi_{i-1,j},\varphi_{i,j},\varphi_{i+1,j},\varphi_{i,j-1},\varphi_{i,j+1}\right\}, \tag{3.58}$$

$$\varphi^{min}=\min\left\{\varphi_{i-1,j},\varphi_{i,j},\varphi_{i+1,j},\varphi_{i,j-1},\varphi_{i,j+1}\right\}. \tag{3.59}$$

If $\varphi^{max}\varphi^{min}>0$, the grid point $(x_i,y_j)$ is called *regular*; otherwise it is *irregular*, meaning the interface cuts through the standard five point finite difference stencil. Note that, with the level set function representation, the interface can cut the grid line only between two neighboring grid points, for example, $(x_{i-1},y_j)$ and $(x_i,y_j)$ once.

At an irregular grid point where the interface cuts through either the $x$-axis or $y$-axis, we simply add the correction terms to Yee's scheme as in the one-dimensional case:

$$v_{i,j+1/2}^{k+1/2}=v_{i,j+1/2}^{k-1/2}-\frac{1}{\rho}\frac{\Delta t}{h_y}\left(p_{i,j+1}^{k}-p_{ij}^{k}+C_{ij}^{p,y}\right), \tag{3.60}$$

$$u_{i+1/2,j}^{k+1/2}=u_{i+1/2,j}^{k-1/2}-\frac{1}{\rho}\frac{\Delta t}{h_x}\left(p_{i+1,j}^{k}-p_{ij}^{k}+C_{ij}^{p,x}\right), \tag{3.61}$$

$$p_{ij}^{k+1}=p_{ij}^{k}-K\Delta t\left(\frac{v_{i,j+1/2}^{k+1/2}-v_{i,j-1/2}^{k+1/2}+C_{ij}^{v,y}}{h_y}+\frac{u_{i+1/2,j}^{k+1/2}-u_{i-1/2,j}^{k+1/2}+C_{ij}^{u,x}}{h_x}\right), \tag{3.62}$$

where $C_{ij}^{p,y}$, $C_{ij}^{p,x}$, $C_{ij}^{v,y}$, and $C_{ij}^{u,x}$ correspond to correction terms, see Lemma 3.2.1 and 3.8-3.9, if the interface cuts the staggered finite difference stencil. Since we use the dimension by dimension discretization and the specially selected augmented variables, the formula in Lemma 3.2.1 is used for the correction terms. Again, we approximate the second order jumps explicitly. The modified finite difference scheme can be written as

$$\mathbf{Z}^{k+1} = S_k \mathbf{Z}^k + B\mathbf{Q}^{k+1} + \mathbf{F}^k, \tag{3.63}$$

where $\mathbf{Q}^{k+1}$ is the vector of the discrete augmented variables $q_1, q_2, q_3$, and $q_4$ defined in 3.54.

### 3.3.4 Discretization of the Interface Conditions

In order to discretize the interface conditions, we need to choose a set of points $(X_l, Y_l)$, $l = 1, 2, \cdots, N_b$, called control points, on the interface. With a level set function representation and assuming that the level set function is a good approximation of the signed distance function, see for example [81, 91], we use the orthogonal projections of irregular grid points $(x_i, y_j)$ on the interface from one particular side, often $\Omega^+$, see the little hexagons in Figure 3.2 for an illustration. If we know values of a given function at those points, we can find the function value, its first and second order surface derivatives from a local reconstruction, see [64] for the details regarding how this idea has been used for other application problems.

The discretization of (3.53) is done using the second order least squares interpolation scheme [64]. In the least squares interpolation scheme, we select more grid points, often $9 \sim 12$ in two dimensions, than that of needed for second order accuracy. Advantages of the least squares interpolation include balanced coefficients and more accurate derivative approximations.

We take the last jump condition $\left[\frac{1}{\rho} p_n\right] = 0$ in (3.53) at $(X_l, Y_l)$ at time level $k+1$ as an example. The interpolation using $p_{ij}^{k+1}$, $[p] = 0$, and $q_4 = [p_n]$ is an approximation to $\frac{1}{\rho^+} p_n^+ - \frac{1}{\rho^-} p_n^- = 0$ at $(X_l, Y_l)$ and the time level $k+1$. The interface condition is equivalent to $\left[\frac{1}{\rho}\right] p_n^- + \frac{q_4}{\rho^+} = 0$. The least squares interpolation scheme is

$$\left[\frac{1}{\rho}\right] \sum_{z=0}^{z_I - 1} \gamma_z^l \, p_{i^*+z,\, j^*+z}^{k+1} - C_l^{k+1} + \frac{q_{4,l}^{k+1}}{\rho^+} = 0, \tag{3.64}$$

where $(x_{i^*}, y_{j^*})$ is the closest grid point to $(X_l, Y_l)$, $z_I$ is the number of grid points involved in the interpolation scheme $(9 \sim 12)$, and $C_l^{k+1}$ is a correction term.

It is more convenient to use the local coordinates in the normal and tangential directions,

$$\xi = (x - X_l)\cos\theta + (y - Y_l)\sin\theta, \tag{3.65}$$

$$\eta = -(x - X_l)\sin\theta + (y - Y_l)\cos\theta, \tag{3.66}$$

where $\theta$ is the angle between the $x$-axis and the normal direction $\mathbf{n} = (\cos\theta, \sin\theta)$ at $(X_l, Y_l)$.

For simplification of notations, we will drop the time level dependance $t^{k+1}$ and $k+1$. The coefficients $\{\gamma_z^l\}$ are determined by minimizing the interpolation error of (3.64) when $p_{i^*+z,\, j^*+z}$ is substituted by the exact solution $p(x_{i^*+z}, y_{j^*+z}, :)$. Using the local coordinate system centered at the point $(X_l, Y_l)$, and denoting the local coordinates of $(x_{i^*+z}, y_{j^*+z})$ as $(\xi_z, \eta_z)$, we have the following from the Taylor expansion at $\mathbf{X}_l = (X_l, Y_l)$, or $(0,0)$ in the local coordinates:

$$p(x_{i^*+z}, y_{j^*+z}) = p(\xi_z, \eta_z) = p^{\pm} + \xi_z p_{\xi}^{\pm} + \eta_z p_{\eta}^{\pm} \tag{3.67}$$

$$+\frac{1}{2}\xi_z^2 p_{\xi\xi}^{\pm} + \xi_z \eta_z p_{\xi\eta}^{\pm} + \frac{1}{2}\eta_z^2 p_{\eta\eta}^{\pm} + O(h^3), \tag{3.68}$$

where the '+' or '−' sign is chosen depending on whether $(\xi_z, \eta_z)$ lies on the '+' or '−' side of $\Gamma$, $p^{\pm}$, $p_{\xi}^{\pm}, \cdots, p_{\eta\eta}^{\pm}$ are evaluated at the local coordinates $(0,0)$, or $\mathbf{X}_l = (X_l, Y_l)$ in the original coordinate system. The coefficients $\{\gamma_k\}$'s satisfy the following linear system of equations:

$$\sum_{z=0}^{z_l-1} \gamma_z^l = 0, \qquad \sum_{z=0}^{z_l-1} \gamma_z^l \xi_z = 1, \tag{3.69}$$

$$\sum_{z=0}^{z_l-1} \gamma_z^l \eta_z = 0, \qquad \frac{1}{2}\sum_{k=0}^{z_l-1} \gamma_z^l \xi_z^2 = 0, \tag{3.70}$$

$$\sum_{z=0}^{z_l-1} \gamma_z^l \xi_z \eta_z = 0, \qquad \frac{1}{2}\sum_{z=0}^{z_l-1} \gamma_z^l \eta_z^2 = 0. \tag{3.71}$$

Once $\{\gamma_z^l\}$'s are computed, the correction term $C_l^{k+1}$ is determined from the following:

$$C_l^{k+1} = a_4\, q + a_6 [p_\eta] + a_8 [p_{\xi\xi}] + a_{10} [p_{\xi\eta}] + a_{12} [p_{\eta\eta}], \tag{3.72}$$

where $a_2 = \sum_{z\in K^+} \gamma_z^l$, $a_4 = \sum_{z\in K^+} \xi_z \gamma_z^l$, $a_6 = \sum_{z\in K^+} \eta_z \gamma_z^l$, and so on; $z \in K^+$ means that the grid point involved in the interpolation $(x_{i^*+z}, y_{j^*+z}) \in \Omega^+$.

If we carry out such a local least squares interpolation for all the interface conditions 3.53, the discretization can be written as

$$R\mathbf{Z}^{k+1} + G\mathbf{Q}^{k+1} - \mathbf{F}_2^k = 0, \tag{3.73}$$

where the vector $\mathbf{Q}^{k+1}$ contains the discrete values of $q_1$, $q_2$, $q_3$, and $q_4$, and the non-zero entries of $R$ and $G$ are the coefficients of the local interpolations scheme. Thus, the size of $\mathbf{Q}^{k+1}$ is $O(4N_b) \sim O(4N)$. The procedure is the same as described early except that the matrices $R$, $B$, and $G$ are different, and unlike the one-dimensional case, we can not solve $\mathbf{Q}^{k+1}$ explicitly. Once again, eliminating $\mathbf{Z}^{k+1}$ in (3.63) gives the Schur complement equation for $\mathbf{Q}$

$$(G + RB)\mathbf{Q}^{k+1} = \mathbf{F}_2^k - R\mathbf{F}_1^k. \tag{3.74}$$

For fixed interface and time step size $\Delta t$, the matrix $(G + RB)$ is also fixed. Thus, we use one step $P(G + RB) = LU$ decomposition and find subsequent solutions $\mathbf{Q}^{k+1}$ rather easily, where $P$ is a permutation matrix.

**The Convergence Discussion:** The ADI-Yee's scheme is consistent with $O(h^2)$. This is a result of using the Yee stencil from Yee's scheme. For a line interface that is parallel to one of the axes, the Maxwell interface problem is essentially a one-dimensional interface problem, and the ADI-Yee's scheme is stable with the usual Yee's CFL constraint. For general interfaces, the study of the stability is complicated and it is still a challenging open problem. Nevertheless, all our numerical tests show that the usual Yee's CFL constraint is enough for the stability. There are two intuitive explanations of the stability. The augmented variables are determined implicitly, which often improve the stability. Secondly, the modifications are needed at grid points near or on the interfaces. The number of those grid points has co-dimension one compared with the total number of grid points.

## 3.4 Numerical Examples

We have implemented our new method for one and two-dimensional problems with different interfaces and discontinuities and compared the numerical results with those from the literature. All numerical experiments show second order convergence under usual CFL conditions for Yee's scheme.

The first example is a two-dimensional problem from [106] . We show grid refinement analysis using the two-dimensional example from [106] in which a dielectric of relative permittivity $\epsilon_2$, occupying the spatial region $[0,1/2]\times[0,1]$, is inserted in a PEC-bounded $[0,5/4]\times[0,1]$ domain. An exact solution in this case is $(0 \le y \le 1)$:

$$E^z = p = \begin{cases} \sin(a_1 x)\sin(by)\sin(\omega t), \\ \cos(a_2 x)\sin(by)\sin(\omega t), \end{cases} \tag{3.75}$$

$$H^y = -u = \begin{cases} \frac{a_1}{\omega}\cos(a_1 x)\sin(by)\cos(\omega t), & \text{if } 0 \le x \le \frac{1}{2}, \\ \frac{a_2}{\omega}\sin(a_2 x)\sin(by)\cos(\omega t), & \text{if } \frac{1}{2} \le x \le \frac{5}{4}, \end{cases} \tag{3.76}$$

$$H^x = v = \begin{cases} \frac{b}{\omega}\sin(a_1 x)\cos(by)\cos(\omega t), & \text{if } 0 \le x \le \frac{1}{2}, \\ \frac{b}{\omega}\cos(a_2 x)\cos(by)\cos(\omega t), & \text{if } \frac{1}{2} \le x \le \frac{5}{4}. \end{cases} \tag{3.77}$$

We use the same parameters as in [106, 110], $\epsilon_1 = 1$, $\epsilon_2 = 2$, $a_1 = 2\pi$, $b = \pi$, and $\omega = \sqrt{5}\pi$, and present grid refinement results in Table 3.1, which clearly confirm second order accuracy for all involved quantities.

In this constructed general example, we allow the solutions $(p, u, v)$ and their derivatives, the coefficients of the partial differential equations, and source terms, to be discontinuous. The jump conditions are non-homogeneous with more general interfaces.

**Table 3.1** Grid refinement analysis for Maxwell's equations with a line interface $x = 1/2$ using the dimensional jump splitting method. Second order convergence is confirmed. The final time is $T = 5$.

| $N$ | $E(u)$ | $r$ | $E(v)$ | $r$ | $E(p)$ | $r$ |
|---|---|---|---|---|---|---|
| 16 | 1.5859e-02 | | 6.0688e-02 | | 6.0688e-02 | |
| 32 | 4.0064e-03 | 1.9850 | 1.5691e-02 | 1.9514 | 1.5691e-02 | 2.0138 |
| 64 | 1.0535e-03 | 1.9271 | 4.0553e-03 | 1.9521 | 4.0553e-03 | 2.0537 |
| 128 | 2.7546e-04 | 1.9353 | 1.0455e-03 | 1.9556 | 1.0455e-03 | 2.0027 |
| 256 | 6.9981e-05 | 1.9768 | 2.6496e-04 | 1.9804 | 2.6496e-04 | 1.9820 |
| 512 | 1.7399e-05 | 2.0080 | 6.6046e-05 | 2.0042 | 6.6046e-05 | 2.0063 |

The true solution is designed as below

$$u_0(x,y) = \begin{cases} -a_1 x + b_1 \sin(x+y) + a_3 h(t) & \text{if } (x,y) \in \Omega^+, \\ -a_2 x + b_2 \sin(x+y) + a_3 h(t) & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{3.78}$$

$$v_0(x,y) = \begin{cases} b_1 \sin(x+y) + a_3 h(t) & \text{if } (x,y) \in \Omega^+, \\ b_2 \sin(x+y) + a_3 h(t) & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{3.79}$$

$$p_0(x,y,t) = \begin{cases} t + a_3 \sin x \sin y & \text{if } (x,y) \in \Omega^+, \\ t + a_3(x^2+y^2) & \text{if } (x,y) \in \Omega^-. \end{cases} \tag{3.80}$$

The interface is an ellipse that is represented by the zero level set of

$$\varphi(x,y) = \frac{x^2}{r_a^2} + \frac{y^2}{r_b^2} - 1. \tag{3.81}$$

The source term is derived from the solution,

$$f_u(x,y,t) = \begin{cases} \frac{a_3 \cos x \cos y}{\rho_1} + a_3 h'(t) & \text{if } (x,y) \in \Omega^+, \\ \frac{2a_3 x}{\rho_2} + a_3 h'(t) & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{3.82}$$

$$f_v(x,y,t) = \begin{cases} \frac{a_3 \sin x \sin y}{\rho_1} + a_3 h'(t) & \text{if } (x,y) \in \Omega^+, \\ \frac{2a_3 y}{\rho_2} + a_3 h'(t) & \text{if } (x,y) \in \Omega^-, \end{cases} \tag{3.83}$$

$$f_p(x,y,t) = 0. \tag{3.84}$$

Table 3.2 shows results of grid refinement analysis with $h(t) = \sin t$ with the usual Yee's CFL condition. The interface is a circle: $x^2 + y^2 = 0.8^2$. The parameters are $a_1 = 10$, $a_2 = 0.7$, $b_1 = 1.5$, $b_2 = 3$, $\rho_1 = 10$, $\rho_2 = 2$, and $a_3 = 1$. Average second order accuracy is confirmed. Note that for Cartesian mesh methods that do not fit the interface, the error constants depend on the interface because the relative position of the interface and the underlying grid is not fixed, see for example

**Table 3.2** Grid refinement analysis for Maxwell's equations with a circular interface $x^2 + y^2 = 0.8^2$ using the ADI-Yee's scheme with the usual Yee's CFL constraint. Second order convergence is confirmed. The final time is $T = 1.2$.

| $N$ | $E(u)$ | $r$ | $E(v)$ | $r$ | $E(p)$ | $r$ |
|---|---|---|---|---|---|---|
| 16 | 2.8221e-02 | | 2.1101e-02 | | 1.7693e-02 | |
| 32 | 4.6064e-03 | 2.6151 | 3.0213e-03 | 2.8041 | 2.4169e-03 | 2.8719 |
| 64 | 5.3087e-04 | 3.1172 | 5.7567e-04 | 2.3918 | 2.5706e-04 | 3.2330 |
| 128 | 1.4724e-04 | 1.8502 | 2.0385e-04 | 1.4977 | 8.0732e-05 | 1.6709 |
| 256 | 3.4879e-05 | 2.0777 | 4.8737e-05 | 2.0644 | 1.9963e-05 | 2.0158 |
| Average | | 2.4151 | | 2.1895 | | 2.4479 |

[63]. Similar results are observed with different $h(t) \in C^1(t > 0)$.

In Table 3.3, we show the results of grid refinement analysis with large jump ratios in the coefficients and an elliptic interface: $\frac{x^2}{1.1^2} + \frac{y^2}{0.8^2} = 1$. The parameters are $a_1 = 1.2$, $a_2 = 50$, $b_1 = 5$, $b_2 = 3$, $\rho_1 = 1.5$, $\rho_2 = 100$, and $a_3 = 1$ with $h(t) = \sin t$. In this case, the coarse grid with $N = 16$ would not generate a meaningful result due to the unresolved (complicated) geometry.

In the bottom Table 3.3, we switch everything in the parameters used in the top table, that is, the interface is $\frac{x^2}{0.8^2} + \frac{y^2}{1.1^2} = 1$, and $a_1 = 50$, $a_2 = 1.2$, $b_1 = 3$, $b_2 = 5$, $\rho_1 = 100$, and $\rho_2 = 1.5$.

**Table 3.3** Grid refinement analysis for Maxwell's equations at a final time $T = 1.2$ with an elliptic interface $\frac{x^2}{1.1^2} + \frac{y^2}{0.8^2} = 1$ and large jumps in the coefficients using the ADI-Yee's scheme. Second order convergence is observed again. The bottom table shows the results of switching the parameter values inside and outside with the interface being $\frac{x^2}{0.8^2} + \frac{y^2}{1.1^2} = 1$.

| $N$ | $E(u)$ | $r$ | $E(v)$ | $r$ | $E(p)$ | $r$ |
|---|---|---|---|---|---|---|
| 32 | 5.2014e-01 | | 2.1529e-01 | | 2.6368e-01 | |
| 64 | 3.1325e-03 | 7.3755 | 4.6313e-03 | 5.5387 | 1.9557e-03 | 7.0750 |
| 128 | 8.1156e-04 | 1.9485 | 8.3639e-04 | 2.4692 | 3.4856e-04 | 2.4882 |
| 256 | 1.4849e-04 | 2.4503 | 1.1429e-04 | 2.8715 | 3.7206e-05 | 3.2278 |
| Average | | 3.4630 | | 3.2360 | | 3.7017 |
| 32 | 3.5252e-01 | | 2.9110e-01 | | 1.7246e-01 | |
| 64 | 8.8514e-02 | 1.9937 | 5.3917e-02 | 2.4327 | 1.8491e-02 | 3.2214 |
| 128 | 1.8516e-02 | 2.2572 | 1.0871e-02 | 2.3102 | 2.0227e-03 | 3.1925 |
| 256 | 1.0516e-03 | 4.1380 | 2.4153e-03 | 2.1702 | 6.0501e-04 | 1.7412 |
| Average | | 2.6167 | | 2.2444 | | 2.5427 |

## 3.5 Conclusions

In this chapter, we proposed a new method for Maxwell's equations with material interfaces based on Yee's scheme and the augmented strategy. With some intermediate (augmented) variables, we can discretize Maxwell's equations dimension by dimension with some added correction terms to treat the discontinuities. The new method maintains the structure and accuracy of Yee's scheme even with the presence of discontinuities.

CHAPTER

# 4

# A MIXED-INTEGER PDE-CONSTRAINED OPTIMIZATION FORMULATION FOR THE CONSTRUCTION OF ELECTROMAGNETIC CLOAKS WITH A SINGLE MATERIAL IN 2D

## 4.1 Chapter Outline

In this work, we study the design of electromagnetic cloaks by formulating and solving a mixed-integer partial-differential equation constrained optimization problem. The objective is to minimize the integral of the response in the region we wish to cloak. The constraints include a 2D Helmholtz equation with absorbing boundary conditions for each component of the complex valued scattering wave, that are parameterized by the design of the electromagnetic cloak. The design is obtained using binary variables that determine whether a cell in the system is filled with material or not. The cloak design is dependent on the angle of the incidence wave, the cloaking material, and the wavenumber.

We also consider the design of a cloak that works well for several different angles of the incident wave leading to a stochastic programming formulation where we minimize the expected value of the objective function with respect to a discrete set of angles. Rather than a single 2D Helmholtz equation, this uncertain version has a 2D Helmholtz equation for each angle.

In both the single and uncertain incident wave formulation we solve the combinatorial optimization problem in two steps. The first step is to relax the binary variables, solve the relaxation, and apply a rounding heuristic to obtain an initial design. In the second step we use a new trust-region method to improve upon the initial design to obtain a final design. Our approach replaces the large-scale MINLP by a sequence of easier linear knapsack problems.

**Chapter Outline:** In Section 4.2.1 we define the infinite-dimensional electromagnetic cloaking problem using a single material for a nominal wave angle. We then extend this formulation to an optimization problem under uncertainty that can account for uncertainty in the angle of the incident wave, and we compute sensitivities with respect to the design variables that are needed in our optimization approach. In Section 4.2.4 we derive a finite-dimensional optimal design problem by discretizing the PDE using a finite-element approach, resulting in a large-scale MINLP. Next, we present an approach to solve the relaxed MINLP. In Section 4.2.5 we motivate a simple rounding heuristic and describe our trust-region steepest-descent algorithm. We provide numerical experiments in Section 4.2.6 to demonstrate the success of our trust-region method, and we comment on the computational effort to solve our problems. We define success as our ability to bring the integrand of the objective function to a small value. We conclude with a brief summary in Section 4.3. Detailed numerical results are presented in the appendix section .1.

## 4.2 Electromagnetic Cloaking Design for a Single Material in 2D

In this section we review the infinite-dimensional cloaking design problem for a single material. We extend the nominal design and propose a stochastic optimization formulation for the design under uncertainty. We also show how adjoints of the problems are derived. We choose the adjoint method because we have a large amount of controls present in the problem. The adjoint method requires only one state solve and one adjoint solve to produce the gradient (independent of the number of controls). An approach, such as calculating sensitivity equations, would require us to solve the state equation as many times as there are controls present in the problem. These adjoints are needed for the solution of the continuous relaxation and for our proposed trust-region method.

### 4.2.1 Formulation of the Nominal Wave Electromagnetic Cloaking Problem

We consider the design of an electromagnetic cloak from [41], where we aim to determine the optimal topology that induces electromagnetic cloaking. We let $D \subset \mathbb{R}^2$ be the computational domain and

$D_0 \subset D$ be the part of the domain that we wish to cloak. We let $\widehat{D} \subset D - D_0$ be the subregion in which we can design the cloak and $\hat{\Omega} \subset \widehat{D}$ be the topology of the cloak. The optimal topology is determined by solving the following topology optimization problem that aims to minimize the response due to the incidence wave in $D_0$ subject to Helmholtz equation in $D$:

$$\begin{aligned} \underset{u,\Omega\subset\widehat{D}}{\text{minimize}} \quad & \frac{1}{2}\|u+\bar{u}_\theta\|_{2,D_0}^2 \\ \text{subject to} \quad & -\Delta u - k_0^2\big(1+q\mathbb{1}_\Omega\big)u = k_0^2\mathbb{1}_\Omega\bar{u}_\theta, \quad \text{in } D, \\ & \frac{\partial u}{\partial n} - i k_0 u = 0, \quad \text{on } \partial D, \end{aligned} \tag{4.1}$$

where $i=\sqrt{-1}$, $\partial D$ is the boundary of the computational domain and

$$\bar{u}_\theta(x,y) = \exp(i k_0(x\cos(\theta) + y\sin(\theta))) \tag{4.2}$$

is the incident wave in direction $\mathbf{d} = (\cos(\theta), \sin(\theta))$, with wave number $k_0$. The angle $\theta$ at which the incident wave approaches the cloak is measured from the x-axis, and we refer to $\theta$ as the nominal angle of attack. The state variable $u$ represents the electromagnetic field intensity, and the parameter $q$ is the electromagnetic permittivity difference between $\Omega$, the part of the cloak filled with material, and $\widehat{D} - \hat{\Omega}$, the part of the cloak without material.

The interpretation of this MIPDECO is as follows: the objective function represents the scattering that occurs in the region $D_0$. Our goal is to minimize this quantity. If the scattering in the region $D_0$ is small, then this means $D_0$ is optically invisible. The constraint, which is Maxwell's wave equation in the frequency domain, describes the scattering behavior that will occur as the electromagnetic wave that travels through the domain interacts with the cloak. The optimization problem is to determine the cloak that will cause scattering to be minimized in $D_0$; by minimizing the scattering we are inducing electromagnetic cloaking.

When manufacturing cloaks, there is typically a lower bound on the size of the features in the cloak, $\hat{\Omega}$. We partition the cloak $\hat{\Omega}$ into a finite number of equally sized squares, $\hat{\Omega}_n$, for $n = 1, \ldots, N$. We refer to this partition as the *control mesh* and assume that the control mesh defines a partition of the computational domain, i.e, that

$$\hat{\Omega} = \bigcup_{n=1}^{N} \hat{\Omega}_n \quad \text{and} \quad \big(\hat{\Omega}_n \setminus \partial\hat{\Omega}_n\big) \cap \big(\hat{\Omega}_p \setminus \partial\hat{\Omega}_p\big) = \emptyset \text{ for } n \neq p,$$

see Figure 4.1 for an illustration.

We represent the decision of whether or not to fill a square $\hat{\Omega}_n \subset \hat{\Omega}$ by a binary (control) variable

$v_n \in \{0,1\}$ for $n = 1, \ldots, N$, and we define the indicator function $\mathbb{1}_{\hat{\Omega}_n}$ for $\hat{\Omega}_n$ as

$$\mathbb{1}_{\hat{\Omega}_n}(x, y) = \begin{cases} 1 & \text{if}\,(x, y) \in \hat{\Omega}_n \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

for $n = 1 \ldots N$. The resulting reformulation of (4.1) can be rewritten as the MIPDECO

$$\begin{aligned} \underset{u,v,w}{\text{minimize}} \quad & \frac{1}{2}\|u + \bar{u}_\theta\|^2_{2,D_0} \\ \text{subject to} \quad & -\Delta u - k_0^2(1 + q\,w)u = k_0^2 q\,w\,\bar{u}_\theta && \text{in } D, \\ & \frac{\partial u}{\partial n} - i\,k_0\,u = 0 && \text{on } \partial D, \\ & w = \sum_{n=1}^{N} v_n \mathbb{1}_{\hat{\Omega}_n \setminus \partial \hat{\Omega}_n}, \\ & v_n \in \{0,1\} && \forall n = 1, \ldots, N, \end{aligned} \tag{4.4}$$

where the defined control, $w$, now takes the place of the topological variable, $\hat{\Omega}$. In this formulation, we have a fixed finite number of binary variables $v_n$ that represent the design. Our formulation (4.4) is a full-space problem. We could equally derive a reduced-space formulation in which $u = u(v)$ is implicitly defined as the solution of the PDE.

Because $u, \bar{u}_\theta : \mathbb{R}^2 \to \mathbb{C}$ are complex valued, we split $u$ and $\bar{u}_\theta$ into its real and the imaginary parts. We then formulate two separate PDEs for each part. In particular, we set $u = u_{\text{Re}} + i\,u_{\text{Im}}$ and $\bar{u}_\theta = \bar{u}_{\text{Re}} + i\,\bar{u}_{\text{Im}}$. We then rewrite the PDE-constraints in (4.4) as

$$\begin{aligned} -\Delta u_{\text{Re}} - k_0^2(1 + q\,w)u_{\text{Re}} &= k_0^2 q\,w\,\bar{u}_{\text{Re}} \\ -\Delta u_{\text{Im}} - k_0^2(1 + q\,w)u_{\text{Im}} &= k_0^2 q\,w\,\bar{u}_{\text{Im}} \\ \frac{\partial u_{\text{Re}}}{\partial n} = -k_0\,u_{\text{Im}}, \qquad \frac{\partial u_{\text{Im}}}{\partial n} &= k_0\,u_{\text{Re}}. \end{aligned} \tag{4.5}$$

In general, we have an $m \times m$ control mesh made up of $N = m^2$ squares. The optimization problem is to decide if we place material ($v_n = 1$) or do not place material ($v_n = 0$) in any $\hat{\Omega}_n \subset D$ to best cloak the region $D_0$. We illustrate a sample design in Figure 4.1, where red indicates material and blue indicates no material. It is straightforward to add constraints on $v$ to model other design restrictions, such as an upper bound on the amount of cloaking material which can be modeled by $e^T v \leq M$, where $M \leq N$ is the total number of patches that can be filled.

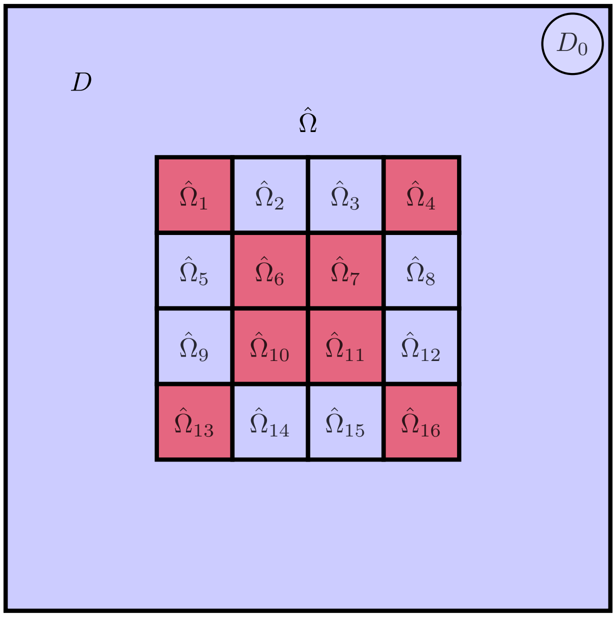


**Figure 4.1** Illustration of a 4×4 grid of control elements in which the goal is to cloak the region $D_0$, in this case a circle in the top right corner of the domain.

### 4.2.2 Optimal Design for Electromagnetic Cloaking Under Uncertainty

Here, we extend the optimal design problem to account for uncertainty in the incidence angle or angle of attack, $\theta$. We formulate an optimization problem to create a design that can effectively cloak a given region even if the incidence angle of the wave changes.

To account for uncertainty in the design, we generalize problem (4.4) to allow a range of incidence angles $\theta$ and we regard the incidence angle as a random variable. We can then formulate an optimization problem that minimizes, for example, the expected value of the effectiveness of the cloak, which is a risk-neutral formulation [4, 29, 35, 56] . We let $(\Theta, \mathscr{F}, P)$ be a triple of sample space, $\sigma$-algebra, and probability measure, and we formulate the following stochastic optimization problem:

$$\begin{aligned} \underset{u,v,w}{\text{minimize}} \quad & \mathscr{J}(u,v,w) = \mathbb{E}_\theta\left[\frac{1}{2}\|u_\theta + \bar{u}_\theta\|^2_{2,D_0}\right] & \\ \text{subject to} \quad & -\Delta u_\theta - k_0^2(1+qw)u_\theta = k_0^2 q w \bar{u}_\theta & \text{in } D,\ \forall \theta \in \Theta \\ & \frac{\partial u_\theta}{\partial n} - i k_0 u_\theta = 0 & \text{on } \partial D,\ \forall \theta \in \Theta \\ & w = \sum_{n=1}^{N} v_n \mathbb{1}_{\hat{\Omega}_n \backslash \partial \hat{\Omega}_n} & \\ & v_n = \{0,1\} & \forall n = 1,\dots,N, \end{aligned} \tag{4.6}$$

where $\mathbb{E}_\theta[\cdot]$ is the expectation with respect to the random variable $\theta$, which is distributed uniformly, and $u_\theta = u_\theta(v)$ is the solution of the PDE for a given control, $v$, and incidence angle $\theta$. Problem (4.6) has a structure similar to that of a two-stage stochastic program [92] if we interpret the controls $w$ and $v$ as the first-stage decision variables, and the states $u_\theta$ as the second-stage variables. However, there is no optimization in the second stage because $w$, $v$, and $\theta$ uniquely determine $u_\theta$. To approximate (4.6), we select a finite set of $s$ angles denoted by the set $\Theta(s) \subset \Theta$, for example,

$$\Theta(s) = \left\{ \theta_j \middle| \, \theta_j = \frac{\pi j}{2s}, j = 0, \ldots, s \right\},$$

and we replace the expectation by the sample average approximation (SSA). We assume that each $\theta_j$ has a probability $P_j$ of occurring, and we assume that all angles of attack are equally likely, $P_j = \frac{1}{|\Theta(s)|}$ (though this assumption is readily relaxed). We can interpret $\Theta(s)$ as a discretization of a uniform distribution over the interval of interest. We now rewrite our cloaking optimization problem for the finite set $\Theta(s)$ as

$$\begin{aligned}
\underset{u^{(j)}, v, w}{\text{minimize}} \quad & \mathscr{J}(u^{(j)}, v, w) = \frac{1}{2|\Theta(s)|} \sum_{j=0}^{s} \| u^{(j)} + \bar{u}^{(j)} \|_{2, D_0}^2 & & \\
\text{subject to} \quad & -\Delta u^{(j)} - k_0^2 (1 + q w) u^{(j)} = k_0^2 q w \bar{u}^{(j)} & & \text{in } D, \ \forall j = 0, \ldots, s \\
& \frac{\partial u^{(j)}}{\partial n} - i k_0 u^{(j)} = 0 & & \text{on } \partial D, \ \forall j = 0, \ldots, s \qquad (4.7) \\
& w = \sum_{n=1}^{N} v_n \mathbb{1}_{\hat{\Omega}_n \setminus \partial \hat{\Omega}_n} & & \\
& v_n = \{0, 1\} & & \forall n = 1, \ldots, N.
\end{aligned}$$

where $u^{(j)}$ is the state corresponding to the $j^{\text{th}}$ scenario, $\theta_j$, and the incidence wave associated with $\theta_j$ is defined as

$$\bar{u}^{(j)}(x, y) = e^{\left( i k_0 (x \cos(\theta_j) + y \sin(\theta_j)) \right)}, \ \forall j = 0, \ldots, s. \tag{4.8}$$

This stochastic program has $2|\Theta(s)|$ state variables and $2|\Theta(s)|$ adjoint variables (two for each angle of attack $\theta_j$). The evaluation of the objective function and gradient of (4.6) requires the solution of $4|\Theta(s)|$ PDEs for each fixed $v$.

### 4.2.3 Adjoint-Based Gradient Computation

Our method for solving (4.7) is based on solving the continuous relaxation followed by a rounding step and an incremental trust-region improvement algorithm. Both steps require computing gradients with respect to the control variables $v$. Here, we present the derivation of the weak state and weak adjoint equation for the uncertain problem (the derivation for the easier nominal problem follows in a similar way). In addition we provide the strong gradient derivation for (4.6) using the adjoint approach for calculating the gradient for the continuous relaxation ($0 \leq v \leq 1$). We begin by formally defining the strong Lagrangian $\mathscr{L}$ of (4.7) as

$$\begin{aligned}
&\mathscr{L}(u_{\text{Re}}^{(j)}, u_{\text{Im}}^{(j)}, v; \lambda_{\text{Re}}^{(j)}, \lambda_{\text{Im}}^{(j)}\theta) \\
&= \frac{1}{2|\Theta(s)|}\left(\sum_{j=0}^{s}\Big(\int_{D_0} \big(u_{\text{Re}}^{(j)} + \bar{u}_{\text{Re}}^{(j)}\big)^2 + \big(u_{\text{Im}}^{(j)} + \bar{u}_{\text{Im}}^{(j)}\big)^2 \, dD_0\right) \\
&+ \sum_{j=0}^{s}\left\langle \lambda_{\text{Re}}^{(j)}, \left(-\Delta u_{\text{Re}}^{(j)} - k_0^2(1+qw)u_{\text{Re}}^{(j)} - k_0^2 qw\bar{u}_{\text{Re}}^{(j)}\right)\right\rangle_{L^2(D)} \\
&+ \sum_{j=0}^{s}\left\langle \lambda_{\text{Im}}^{(j)}, \left(-\Delta u_{\text{Im}}^{(j)} - k_0^2(1+qw)u_{\text{Im}}^{(j)} - k_0^2 qw\bar{u}_{\text{Im}}^{(j)}\right)\right\rangle_{L^2(D)},
\end{aligned} \tag{4.9}$$

where $\langle\cdot,\cdot\rangle_{L^2(D)}$ is the $L^2$ inner product over $D$ and $\lambda_{\text{Re}}^{(j)}, \lambda_{\text{Im}}^{(j)}$ are the adjoint variables for the $j$th real and imaginary state equation respectively. (4.9) defines the strong Lagrangian, because we assume that the pairs $u_{\text{Re}}^{(j)}, u_{\text{Im}}^{(j)} \in C^2(D)$ are twice continuously differentiable. We weaken this regularity assumption because we intend to search for weak solutions with the finite-element method. Next, we derive the weak form of the Lagrangian by applying Green's theorem to the two constraint terms:

$$\begin{aligned}
&\sum_{j=0}^{s}\left\langle \lambda_{\text{Re}}^{(j)}, \left(-\Delta u_{\text{Re}}^{(j)} - k_0^2(1+qw)u_{\text{Re}}^{(j)} - k_0^2 qw\bar{u}_{\text{Re}}^{(j)}\right)\right\rangle_{L^2(D)} \\
&= \sum_{j=0}^{s}\left[\int_D \lambda_{\text{Re}}^{(j)}\left(-\Delta u_{\text{Re}}^{(j)} - k_0^2(1+qw)u_{\text{Re}}^{(j)} - k_0^2 qw\bar{u}_{\text{Re}}^{(j)}\right) dD\right] \\
&= \sum_{j=0}^{s}\left[\int_D \nabla u_{\text{Re}}^{(j)} \cdot \nabla\lambda_{\text{Re}}^{(j)} dD + \int_{\partial D} k_0 u_{\text{Im}}^{(j)}\lambda_{\text{Re}}^{(j)} d\partial D\right. \\
&\left. - \int_D \lambda_{\text{Re}}^{(j)}\left(k_0^2(1+qw)u_{\text{Re}}^{(j)} + k_0^2 qw\bar{u}_{\text{Re}}^{(j)}\right) dD\right]
\end{aligned} \tag{4.10}$$

and

$$\begin{aligned}
&\sum_{j=0}^{s}\Big[\Big\langle \lambda_{\text{Im}}^{(j)}, \Big(-\Delta u_{\text{Im}}^{(j)} - k_0^2(1+qw)u_{\text{Im}}^{(j)} - k_0^2 q w \bar{u}_{\text{Im}}^{(j)}\Big)\Big\rangle_{L^2(D)}\Big] \\
&=\sum_{j=0}^{s}\Big[\int_D \lambda_{\text{Im}}^{(j)}\big(-\Delta u_{\text{Im}}^{(j)} - k_0^2(1+qw)u_{\text{Im}}^{(j)} - k_0^2 q w \bar{u}_{\text{Im}}^{(j)} dD\big)\Big] \\
&=\sum_{j=0}^{s}\Big[\int_D \nabla u_{\text{Im}}^{(j)}\cdot\nabla\lambda_{\text{Im}}^{(j)} dD - \int_{\partial D} k_0 u_{\text{Re}}^{(j)}\lambda_{\text{Im}}^{(j)} d\partial D \\
&-\int_D \lambda_{\text{Im}}^{(j)}\Big(k_0^2(1+qw)u_{\text{Im}}^{(j)} + k_0^2 q w \bar{u}_{\text{Im}}^{(j)}\Big) dD\Big].
\end{aligned}\tag{4.11}$$

We substitute (4.10) and (4.11) into (4.9) and obtain the weak Lagrangian:

$$\begin{aligned}
&\mathscr{L}(u_{\text{Re}}^{(j)}, u_{\text{Im}}^{(j)}, v, \lambda_{\text{Re}}^{(j)}, \lambda_{\text{Im}}^{(j)}, \theta) \\
&=\frac{1}{2|\Theta(s)|}\Big(\sum_{j=0}^{s}\big(\int_{D_0}\big(u_{\text{Re}}^{(j)}+\bar{u}_{\text{Re}}^{(j)}\big)^2 + \big(u_{\text{Im}}^{(j)}+\bar{u}_{\text{Im}}^{(j)}\big)^2 dD_0\big)\Big) \\
&+\sum_{j=0}^{s}\Big[\int_D \nabla u_{\text{Re}}^{(j)}\cdot\nabla\lambda_{\text{Re}}^{(j)} dD + \int_{\partial D} k_0 u_{\text{Im}}^{(j)}\lambda_{\text{Re}}^{(j)} d\partial D \\
&\qquad -\int_D \big[\lambda_{\text{Re}}^{(j)}\big[k_0^2(1+qw)u_{\text{Re}}^{(j)} + k_0^2 q w \bar{u}_{\text{Re}}^{(j)}\big]\big] dD\Big] \\
&+\sum_{j=0}^{s}\Big[\int_D \nabla u_{\text{Im}}^{(j)}\cdot\nabla\lambda_{\text{Im}}^{(j)} dD - \int_{\partial D} k_0 u_{\text{Re}}^{(j)}\lambda_{\text{Im}}^{(j)} d\partial D \\
&\qquad -\int_D \big[\lambda_{\text{Im}}^{(j)}\big[k_0^2(1+qw)u_{\text{Im}}^{(j)} + k_0^2 q w \bar{u}_{\text{Im}}^{(j)}\big]\big] dD\Big].
\end{aligned}\tag{4.12}$$

In the final form (4.12), we assume that $u_{\text{Re}}^{(j)}, u_{\text{Im}}^{(j)} \in H^1(D)$, where $H^1(D) := W^{1,2}(D)$ is the Sobolev space of all functions that are $L^2(D)$ (square integrable) and whose weak partial derivatives, in the sense of distributions, is also in $L^2(D)$ [86]. To find the sensitivities of the objective function, $\mathscr{J}$, with respect to $v$, we exploit the fact that there exists a unique $w$ for any choice of $v$ and that (under suitable conditions on $k_0$) there exists a unique $u$ for any $v$. We start by deriving the weak state equations, which are obtained by taking variations of the Lagrangian with respect to $\tilde{\lambda}_{\text{Re}}^{(j)}, \tilde{\lambda}_{\text{Im}}^{(j)}$ and setting the resulting expression to zero:

$$\begin{aligned}\mathscr{L}_{\lambda_{\mathrm{Re}}^{(j)}}[\tilde{\lambda}_{\mathrm{Re}}^{(j)}] &= \int_D \nabla u_{\mathrm{Re}}^{(j)} \cdot \nabla \tilde{\lambda}_{\mathrm{Re}}^{(j)} dD + \int_{\partial D} k_0 u_{\mathrm{Im}}^{(j)} \tilde{\lambda}_{\mathrm{Re}}^{(j)} d\partial D \\ &\quad - \int_D \tilde{\lambda}_{\mathrm{Re}}^{(j)} \big[k_0^2(1+qw)u_{\mathrm{Re}}^{(j)} + k_0^2 q w \bar{u}_{\mathrm{Re}}^{(j)}\big] dD \\ &= 0 \quad \forall \tilde{\lambda}_{\mathrm{Re}}^{(j)} \in H^1(D), \forall j = 0, \dots, s \end{aligned} \tag{4.13}$$

$$\begin{aligned}\mathscr{L}_{\lambda_{\mathrm{Im}}^{(j)}}[\tilde{\lambda}_{\mathrm{Im}}^{(j)}] &= \int_D \nabla u_{\mathrm{Im}}^{(j)} \cdot \nabla \tilde{\lambda}_{\mathrm{Im}}^{(j)} dD - \int_{\partial D} k_0 u_{\mathrm{Re}}^{(j)} \tilde{\lambda}_{\mathrm{Im}}^{(j)} d\partial D \\ &\quad - \int_D \tilde{\lambda}_{\mathrm{Im}}^{(j)} \big[k_0^2(1+qw)u_{\mathrm{Im}}^{(j)} + k_0^2 q w \bar{u}_{\mathrm{Im}}^{(j)}\big] dD \\ &= 0 \quad \forall \tilde{\lambda}_{\mathrm{Im}}^{(j)} \in H^1(D), \forall j = 0, \dots, s. \end{aligned} \tag{4.14}$$

We note that the weak state equation pair has a unique solution in $H^1(D)$ [41]. Once we have solved this state equation pair, we can then calculate the adjoint pairs, $(\lambda_{\mathrm{Re}}^j, \lambda_{\mathrm{Im}}^j)$, by solving the weak adjoint equations obtained by taking variations of the Lagrangian with respect to $\tilde{u}_{\mathrm{Re}}^{(j)}, \tilde{u}_{\mathrm{Im}}^{(j)}$ and setting the resulting expression to zero:

$$\begin{aligned}\mathscr{L}_{u_{\mathrm{Re}}^{(j)}}[\tilde{u}_{\mathrm{Re}}^{(j)}] &= \frac{1}{|\Theta(s)|} \int_{D_0} \big(u_{\mathrm{Re}}^{(j)} + \bar{u}_{\mathrm{Re}}^{(j)}\big) \tilde{u}_{\mathrm{Re}}^{(j)} dD_0 \\ &\quad + \int_D \nabla \tilde{u}_{\mathrm{Re}}^{(j)} \cdot \nabla \lambda_{\mathrm{Re}}^{(j)} dD - \int_D \lambda_{\mathrm{Re}}^{(j)} k_0^2 (1+qw) \tilde{u}_{\mathrm{Re}}^{(j)} dD \\ &\quad - \int_{\partial D} k_0 \tilde{u}_{\mathrm{Re}}^{(j)} \lambda_{\mathrm{Im}}^{(j)} d\partial D \\ &= 0 \quad \forall \tilde{u}_{\mathrm{Re}}^{(j)} \in H^1(D), \forall j = 0, \dots, s \end{aligned} \tag{4.15}$$

$$\begin{aligned}\mathscr{L}_{u_{\mathrm{Im}}^{(j)}}[\tilde{u}_{\mathrm{Im}}^{(j)}] &= \frac{1}{|\Theta(s)|} \int_{D_0} \big(u_{\mathrm{Im}}^{(j)} + \bar{u}_{\mathrm{Im}}^{(j)}\big) \tilde{u}_{\mathrm{Im}}^{(j)} dD_0 \\ &\quad + \int_D \nabla \tilde{u}_{\mathrm{Im}}^{(j)} \cdot \nabla \lambda_{\mathrm{Im}}^{(j)} dD - \int_D \lambda_{\mathrm{Im}}^{(j)} k_0^2 (1+qw) \tilde{u}_{\mathrm{Im}}^{(j)} dD \\ &\quad + \int_{\partial D} k_0 \tilde{u}_{\mathrm{Im}}^{(j)} \lambda_{\mathrm{Re}}^{(j)} d\partial D \\ &= 0 \quad \forall \tilde{u}_{\mathrm{Im}}^{(j)} \in H^1(D), \forall j = 0, \dots, s. \end{aligned} \tag{4.16}$$

The proof for the existence and uniqueness of the solution to general elliptic equations with Robin boundary conditions in $H^1(D)$ can be found in chapter 4 of [86] and extends to the adjoint equation pair, which differs from the state equation pair only in its source term. An argument similar to [41] can then be applied to show the existence and uniqueness of the solution to the adjoint equation.

After we have calculated the state and adjoint variables for all scenarios, we can assemble the $n$th component of the gradient of the Lagrangian with respect to $v_n$:

$$\left(\nabla_v \mathscr{L}\right)_n = -\sum_{j=0}^{s} \int_{\hat{\Omega}_n} \left( k_0^2 q(u_{\text{Re}}^{(j)} \lambda_{\text{Re}}^{(j)} + u_{\text{Im}}^{(j)} \lambda_{\text{Im}}^{(j)}) + k_0^2 q(\bar{u}_{\text{Re}}^{(j)} \lambda_{\text{Re}}^{(j)} + \bar{u}_{\text{Im}}^{(j)} \lambda_{\text{Im}}^{(j)}) \right) d\hat{\Omega}_n. \tag{4.17}$$

One can be show, as illustrated in [97],

$$\mathscr{L}_v[\tilde{v}] = \langle \nabla_v \mathscr{L}, \tilde{v} \rangle_{\mathbb{R}^N} = \langle \hat{\mathscr{J}}'(v), \tilde{v} \rangle_{\mathbb{R}^N} \quad \forall \tilde{v} \in \mathbb{R}^N. \tag{4.18}$$

Following Lemma 2.21 [97],

$$\left( \hat{\mathscr{J}}'(v^*) \right)^T (v - v^*) \geq 0 \quad \forall v \in V = \{v : 0 \leq v \leq 1, v \in \mathbb{R}^N\} \tag{4.19}$$

are the necessary optimality conditions of the reduced space formulation,

$$\underset{0 \leq v \leq 1}{\text{minimize}} \quad \hat{\mathscr{J}}(v) = \frac{1}{2|\Theta(s)|} \sum_{j=0}^{s} \|S_{\theta_j}(v) + \bar{u}^{(j)}\|_{2,D_0}^2, \tag{4.20}$$

with $S_{\theta_j}$ being the solution operator of the state equation $S_j : v \to u_j$ for an angle of attack $\theta_j$.

### 4.2.4 Solving the Continuous Optimal Design Problem

In the previous section we have described the infinite-dimension optimal design problem, and derived optimality conditions in function space. In this section we describe our discretization and then show how we solve the continuous design problem using a reduced-space method.

#### 4.2.4.1 Discretization of State and Adjoint Equations

Here we show how to solve discretizations of the state and adjoint PDEs and compute gradients (discretized) that can be used in a reduced-space approach. To solve the state and adjoint PDEs we use a continuous Galerkin FEM [66]. We decompose our physical domain $D$ into a computational domain made up of uniform triangle elements, $E_i$, with length and height $h$ as illustrated in Figure 4.2, such that

$$D = \bigcup_i E_i \quad \text{and} \quad \left(E_i \setminus \partial E_i\right) \cap \left(E_j \setminus \partial E_j\right) = \emptyset, \ \forall i \neq j. \tag{4.21}$$

We also assume that each control element $\Omega_n$ covers a fixed number of finite elements $E_i$, as shown in Figure 4.2.

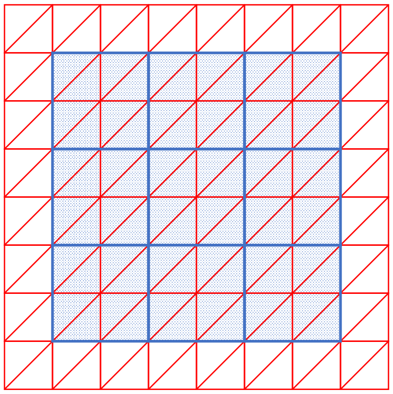


**Figure 4.2** An illustration of the state and adjoint FEM discretization. Each control element (blue) is composed of eight triangular finite elements. In the FEM approach we represent each control element as a constant with value $v_n$. The state and adjoint variables are defined on the finite-element mesh.

We construct an approximate solution to the state and adjoint variables using piecewise linear test functions on our finite elements, giving linear test functions that have compact support on their element. We then approximate the solution by a linear combination of the test functions for the $j$th scenario:

$$u_{\text{Re}}^{h,j} = \sum_{i=1}^{R(h)} \mathbf{u}_{\text{Re}}^{i,j} \phi_i \qquad u_{\text{Im}}^{h,j} = \sum_{i=1}^{R(h)} \mathbf{u}_{\text{Im}}^{i,j} \phi_i \tag{4.22}$$

$$\lambda_{\text{Re}}^{h,j} = \sum_{i=1}^{R(h)} \boldsymbol{\lambda}_{\text{Re}}^{i,j} \phi_i \qquad \lambda_{\text{Im}}^{h,j} = \sum_{i=1}^{R(h)} \boldsymbol{\lambda}_{\text{Im}}^{i,j} \phi_i, \tag{4.23}$$

where $\mathbf{u}_{\text{Re}}^{j}, \mathbf{u}_{\text{Im}}^{j}, \boldsymbol{\lambda}_{\text{Re}}^{j}$, and $\boldsymbol{\lambda}_{\text{Im}}^{j}$ are the coefficient vectors of our expansion that are determined by the FEM (we use boldface to indicate finite-dimensional vectors in the remainder). Here, $R(h)$ is the number of elements in the discretization for a fixed step size $h$.

We choose a piecewise linear FEM basis. Recall that solving the Helmholtz equation with triangular elements gives an error order of $p+1$; this result can be found in chapter two of [66], where p is the order of polynomial used in the approximation over the triangular elements. Thus, with piecewise basis function, we obtain second order error, $\mathcal{O}(h^2)$.

Given a $v$ (and hence, $w$), we substitute (4.22) and (4.23) into (5.11), (5.12), (5.13), and (5.14). This approach allows us to find the coefficients by solving a linear system of equations that defines our approximate solution to the state and adjoint equations over the domain $D$. The resulting linear systems of equations (for fixed $v$) is obtained by substituting our trial solution into the real and imaginary state equations

$$\begin{bmatrix} \mathbf{A}(v) & \mathbf{B}(v) \\ -\mathbf{B}(v) & \mathbf{A}(v) \end{bmatrix} \begin{bmatrix} \mathbf{u}^j_{\text{Re}} \\ \mathbf{u}^j_{\text{Im}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^j_{\text{Re}}(v) \\ \mathbf{f}^j_{\mathbf{Im}}(v) \end{bmatrix} \tag{4.24}$$

for the $j$th angle of attack, where

$$\begin{aligned} \Big[\mathbf{A}(v)\Big]_{i,l} &= \int_D \nabla\phi_i \cdot \nabla\phi_l \, dD + \int_{\partial D} k_0 \phi_i \phi_l \, d\partial D \\ &\quad - \int_D k_0^2 \left(1 + q\left(\sum_{n=1}^N v_n \mathbb{1}_{\Omega_n \setminus \partial\Omega_n}\right)\right) \phi_i \phi_l \, dD, \end{aligned}$$

$$\Big[\mathbf{B}(v)\Big]_{i,l} = \int_{\partial D} k_0 \phi_i \phi_l \, d\partial D,$$

$$\Big[\mathbf{f}^j_{\text{Re}}(v)\Big]_i = \int_D k_0^2 q\left(\sum_{n=1}^N v_n \mathbb{1}_{\Omega_n \setminus \partial\Omega_n}\right) \bar{u}^{(j)}_{\text{Re}} \phi_i \, dD,$$

and

$$\Big[\mathbf{f}^j_{\text{Im}}(v)\Big]_i = \int_D k_0^2 q\left(\sum_{n=1}^N v_n \mathbb{1}_{\Omega_n \setminus \partial\Omega_n}\right) \bar{u}^{(j)}_{\text{Im}} \phi_i \, dD.$$

Next we write down the systems for the adjoint variables associated with the real and imaginary adjoint equations. We note that the weak adjoint equations depend on the solution to the state equations. In the discretized adjoint equations we use the discretized solution found by solving the state equations as an approximation to the state. The system for the adjoint variables is

$$\begin{bmatrix} \mathbf{A}(v) & -\mathbf{B}(v) \\ \mathbf{B}(v) & \mathbf{A}(v) \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^j_{\text{Re}} \\ \boldsymbol{\lambda}^j_{\text{Im}} \end{bmatrix} = \begin{bmatrix} \mathbf{g}^j_{\text{Re}}(v) \\ \mathbf{g}^j_{\mathbf{Im}}(v) \end{bmatrix} \tag{4.25}$$

for the $j$th angle of attack, where

$$\Big[\mathbf{g}^j_{\text{Re}}(u)\Big]_i = -\frac{1}{|\Theta(s)|} \int_{D_0} \left(u^{h,j}_{\text{Re}} + \bar{u}^{(j)}_{\text{Re}}\right) \phi_i \, dD_0$$

and

$$\left[\mathbf{g}^{j}_{\mathrm{Im}}(u)\right]_i = -\frac{1}{|\Theta(s)|}\int_{D_0}\left(u^{h,j}_{\mathrm{Im}} + \bar{u}^{(j)}_{\mathrm{Im}}\right)\phi_i \, dD_0$$

is an affine function in the states, $u$.

Once we have found the solution to the state and adjoint equations, we construct an approximation of the $n$th component of the reduced gradient, $\left(\mathscr{J}'(v)\right)^h_n$, by using a tensorized two-point Gauss-Legendre quadrature rule:

$$\left(\nabla_v \hat{J}^h(v)\right)_n = -\sum_{j=0}^{s}\int_{\hat{\Omega}_n}\left(k_0^2 q(u^{h,j}_{\mathrm{Re}}\lambda^{h,j}_{\mathrm{Re}} + u^{h,j}_{\mathrm{Im}}\lambda^{h,j}_{\mathrm{Im}}) + k_0^2 q(\bar{u}^{h,j}_{\mathrm{Re}}\lambda^{h,j}_{\mathrm{Re}} + \bar{u}^{h,j}_{\mathrm{Im}}\lambda^{h,j}_{\mathrm{Im}})\right) d\hat{\Omega}_n. \tag{4.26}$$

When we approximate both the components of the reduced gradient and the objective function we obtain an error of the order $\mathscr{O}(h^2)$, which is consistent with the discretization error of the state and adjoint equations.

#### 4.2.4.2 Reduced-Space Method for Continuous Design

The derivation in the proceeding section shows that we can write the finite-dimensional approximation of (4.4) as

$$\underset{v\in[0,1]^N}{\text{minimize}} \quad \hat{J}^h(v) = \frac{1}{2|\Theta(s)|}\sum_{j=0}^{s}\|S^h_{\theta_j}(v) + \bar{u}^{(j)}_h\|^2_{2,D_0}, \tag{4.27}$$

where $S^h_{\theta_j}(v)$ is the solution operator for the FEM mesh of size $h$ given by system (4.24) with angle $\theta_j$. The derivation of the reduced gradient, $\nabla_v J^h(v)$, is given in (4.26), and we apply a reduced-space optimization approach using a quasi-Newton approximation of the Hessian.

We note that the structure of the discretized state and adjoint equations (4.24) and (4.25), respectively imply that this optimization problem is nonconvex, because the state equations are bilinear in $v$ and $u$. We investigate the effect of this nonconvexity further in Section 4.2.6.

### 4.2.5 Solving the Mixed-Integer Design Problem

The finite-dimensional approximation of the nominal (single angle) and uncertain (multiple angle) MIPDECO (4.4) and (4.6), respectively, become finite-dimensional pure integer nonlinear optimization problems:

$$\underset{v\in\{0,1\}^N}{\text{minimize}} \quad \hat{J}^h(v) = \frac{1}{2}\|S^h_{\theta}(v) + \bar{u}_h\|^2_{2,D_0} \tag{4.28}$$

and

$$\underset{v\in\{0,1\}^N}{\text{minimize}} \quad \hat{J}^h(v) = \frac{1}{2|\Theta(s)|}\sum_{j=0}^{s}\|S^h_{\theta_j}(v) + \bar{u}^{(j)}_h\|^2_{2,D_0}, \tag{4.29}$$

respectively, where $S_\theta^h(v)$ is the solution operator defined in (4.24).

In this work, we are not concerned with the existence of the optimal solution as we refine the control mesh. Instead, we keep the control mesh fixed, so that we always minimize over the same integer space $\{0,1\}^N$. This restriction ensures that the solution to the weak forward and adjoint system exist, and can be consistently approximated using the FEM discretization.

The structure of (4.28) and (4.29) involving the solution operators $S_\theta^h(v)$ makes it difficult to apply the standard MINLP solvers (e.g., [2, 10, 72]) that require analytic expressions for all the functions. One approach might be to include the discretized PDEs (4.24) and (4.25) directly as constraints. However, we found on other MIPDECOs that such an approach does not provide a MINLP that can be solved by modern MINLP solver in a reasonable time. Hence we consider a heuristic approach, described next.

Our approach for solving the MIPDECOs, (4.28) and (4.29), is a two-phase method. We first solve a relaxation, (4.20) and then round its solution. We apply a discrete steepest-descent trust-region approach to improve this rounded solution. Our approach builds on the method in [37] and can be interpreted as a rounding heuristic followed by repeated application of local branching [32, 79]. Unlike local branching, however, we use only first-order information to solve a sequence of approximations enforcing descent from one iteration to the next. As a result, each trust-region subproblem requires the solution of the discretized forward and adjoint PDEs, (4.24) and (4.25), respectively, making it computationally efficient for MIPDECOs. Moreover, the trust-region problem itself can be formulated and solved efficiently as a knapsack problem.

#### 4.2.5.1 Rounding Heuristic

After solving the relaxation, we use a rounding heuristic with rounding threshold $\tau$,

$$R_\tau(v_n) = \begin{cases} 1 & v_n \geq \tau \\ 0 & v_n < \tau. \end{cases} \tag{4.30}$$

A popular rounding heuristic is $\tau = 0.5$ [6, 70]. However, we chose $\tau = 0.8$. We have observed in our numerical computation that using this value of $\tau$ yields an objective value that is at least 10 percent lower than using $\tau = 0.5$.

#### 4.2.5.2 Steepest-Descent Trust-Region for MIPDECO

Here we describe our steepest-descent trust-region approach to improve the rounded solution. Our approach is related to the local branching heuristic [32, 79], but it neither backtracks nor uses the full nonlinear relaxation. Instead, we only require the solution of the state and adjoint equations to make progress and solve a knapsack problem at each iteration. The approach is motivated by

[37]. The trust-region algorithm can use any feasible point $v \in \{0,1\}^N$ as a starting guess. However, we create our starting guess from using our rounding scheme from 4.2.5.1 on the solution found by solving continuous relaxation,(4.20). The algorithm on the $k$th iteration then tries to produce a better design on each iteration by varying the trust-region radius, $\Delta_k$, at most $\Delta_k$ elements of $v^{(k)}$ to improve the cloaking capability of our design. The algorithm terminates once $\Delta_k < 1$, which means that no local improvement could be found and reducing $\Delta_k$ further would not yield a better point. The algorithm is shown in Algorithm 1.

```
Given initial trust-region radius $\Delta_0 = \bar{\Delta} \geq 1$ and initial guess $v^{(0)} \in \{0,1\}^N$
Select an acceptance step parameter $\bar{\rho}$, and set $k \leftarrow 0$
Evaluate the objective function $\hat{J}_h^{(k)} = \hat{J}^h(v^{(k)})$ and the gradient $g_h^{(k)} = \nabla_v \hat{J}^h(v^{(k)})$;
while $\Delta_k \geq 1$ do
    Solve the trust-region (knapsack) subproblem for $\widehat{v}$:
        $$\begin{aligned} \widehat{v} = \underset{v}{\operatorname{argmin}} \quad & g_h^{(k)^T}\left(v - v^{(k)}\right) + \hat{J}_h^{(k)} \\ \text{subject to} \quad & \|v - v^{(k)}\|_1 \leq \Delta_k \\ & v \in \{0,1\}^N \end{aligned}$$
     ;
    Evaluate the objective $\hat{J}_h(\widehat{v}, u(\widehat{v}))$ by solving state equations with $\widehat{v}$;
    Compute the ratio of actual over predicted reduction: $\rho_k = \frac{\hat{J}^{(k)} - \hat{J}(\widehat{v}, u(\widehat{v}))}{-\left(g^{(k)}\right)^T\left(\widehat{v} - v^k\right)}$;
    if $\rho_k > \bar{\rho}$ then
        Accept the step: $v^{(k+1)} = \widehat{v}$, and evaluate the gradient $g^{(k+1)} = \hat{J}'(v^{(k+1)})$
        if $\|v^{(k+1)} - v^{(k)}\|_1 = \Delta_k$, then  increase the trust-region radius $\Delta_{k+1} = 2\Delta_k$ ;
    else if $\rho_k > 0$ then
        Accept the step $v^{(k+1)} = \widehat{v}$, and evaluate the gradient $g^{(k+1)} = \hat{J}'(v^{(k+1)})$
        Keep trust-region radius unchanged $\Delta_{k+1} = \Delta_k$
    else
        Reject the step, set $v^{(k+1)} = v^{(k)}$, and copy the gradient $g^{(k+1)} = g^{(k)}$
        Reduce the trust-region radius $\Delta_{k+1} = \text{floor}\left(\frac{\Delta_k}{2}\right)$;
    Set $k \leftarrow k + 1$;
```

**Algorithm 1:** Steepest-Descent Trust-Region Algorithm.

We interpret Algorithm 1 as a trust-region method for solving MIPDECOs. We choose the $l_1$-norm trust-region because it is equivalent to the hamming distance between $v$ and $v^{(k)}$, and hence $\text{floor}\left(\Delta_k\right)$ corresponds to the maximum number of components of $v$ that can change from their current value $v^{(k)}$. One can see easily that the trust-region constraint $\|v - v^{(k)}\|_1 \leq \Delta_k$ is equivalent to the following affine constraint

$$\sum_{\substack{i=1 \\ v_i^{(k)}=0}}^{N} v_i + \sum_{\substack{i=1 \\ v_i^{(k)}=1}}^{N} (1 - v_i) \leq \Delta_k, \tag{4.31}$$

resulting in a knapsack constraint. The step acceptance and step rejection criteria are taken directly from the nonlinear trust-region method; see, for example [80] in chapter four.

Our approach is motivated by [38] in which the authors present a convergence analysis for refining a sequence of control meshes. We empirically evaluate the performance of our algorithm in the next section.

### 4.2.6 Numerical Experiments

In this section we describe our experience in solving the nominal cloak design (4.28) and the cloak design under uncertainty (4.29). We start by stating the common features of both experiments. We examine the convexity properties of our design problem and then finally present the results for the nominal and uncertain design in turn.

#### 4.2.6.1 Experimental Setup

We consider the cloaking of three different domains $D_0^1$, $D_0^2$, and $D_0^3$ given by

$$\begin{aligned} D_0^1 &= \{x, y \mid -0.6 \leq x \leq 0.6\, 0.7 \leq y \leq 1\} && \text{(Rectangle)} \\ D_0^2 &= \{x, y \mid 0.7 \leq x \leq 1\, 0.7 \leq y \leq 1\} && \text{(Square)} \\ D_0^3 &= \{x, y \mid (x - .85)^2 + (y - .85)^2 \leq (.1)^2\} && \text{(Circle).} \end{aligned} \tag{4.32}$$

We run all our tests using a MacOS desktop with a 4-core Intel processor working at 3.3 GHz. We use the FEniCS package in Python [5] version 2017.2.0 to solve the state and adjoint PDEs using piecewise linear basis functions. In addition we use the FEniCS default two-point Gauss quadrature method to build the components of the gradient. To solve the relaxation, we use the Toolkit for Advanced Optimization (TAO) [22] with the “blmvm" solver, which is a limited-memory line-search quasi-Newton method for solving bound-constrained problems. We use CPLEX 12.8.0.0 to solve our trust-region knapsack problem. In our trust-region method, we use $\bar{\rho} = .75$, as recommended by [80]), and an initial trust-region radius of $\Delta_0 = 256$. All numerical experiments use the following set of common parameters: wave number $k_0 = 6\pi$, material constant $q = 0.75$, and cloak location $\widehat{D} \subset D$, where

$$D = \left[-1, 1\right] \times \left[-1, 1\right] \quad \text{and} \quad \widehat{D} = \left[-\frac{5}{8}, \frac{5}{8}\right] \times \left[-\frac{5}{8}, \frac{5}{8}\right]. \tag{4.33}$$

Preliminary numerical simulations indicate that a 128×128 finite-element mesh provides a good balance between an efficient solution of the state and adjoint equations and a reliable gradient evaluation. We create six test problems each for the nominal and uncertain design by taking the three cloaking regions from (6.3) and two angles of attack for each region, namely, $\theta = \frac{\pi}{4}$ and $\theta = \frac{\pi}{2}$ in the nominal design. In the case of the design under uncertainty, we consider the same two nominal angles with an uncertainty set of $\pm\frac{\pi}{4}$, giving $\theta \in \left[0, \frac{\pi}{2}\right]$ and $\theta \in \left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$, respectively.

#### 4.2.6.2 Effect of the Nonconvexity

We investigate the nonconvexity of our optimal design problem in the following experiment. We first define a parametric function $w : [0,1] \to L^2(\hat{\Omega})$,

$$w(t)(x,y) = t\,w_1 + (1-t)w_2, \tag{4.34}$$

where

$$w_1 = \sum_{n=1}^{N} v_n^1 \mathbb{1}_{\Omega_n \backslash \partial\Omega_n}, \quad w_2 = \sum_{n=1}^{N} v_n^2 \mathbb{1}_{\Omega_n \backslash \partial\Omega_n}, \tag{4.35}$$

for the particular choice

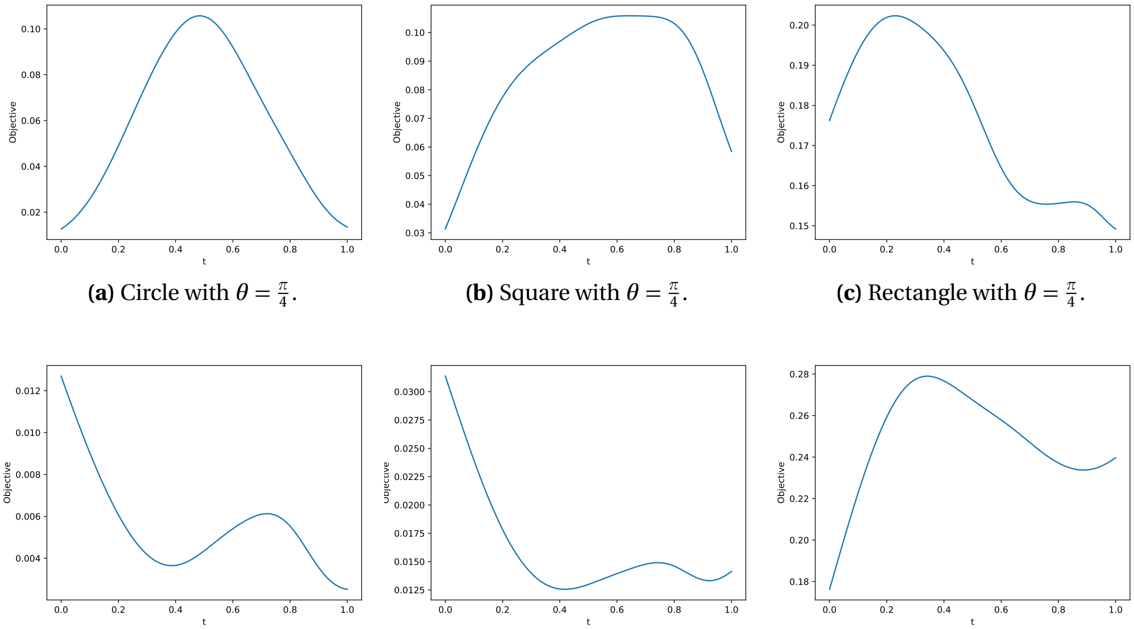
$$v_n^1 = \begin{cases} 1 & n \quad \text{even} \\ 0 & n \quad \text{odd} \end{cases} \quad \text{and} \quad v_n^2 = \begin{cases} 1 & n \quad \text{odd} \\ 0 & n \quad \text{even.} \end{cases} \tag{4.36}$$

It follows that $0 \le w(t) \le 1$, $\forall t \in [0,1]$, and thus $w(t)$ is a feasible point for the relaxation. We then discretize the interval $[0,1]$ into 100 discrete points, $t_1, \ldots, t_{100}$, calculate $w(t_i)$ for $i = 1, .., 100$, and numerically solve the PDE constraint in (4.4) using $w := w(t_i)$. Next, we evaluate the objective $J^h(w(t))$ on a $128 \times 128$ mesh and plot $J^h(w(t))$ versus $t$ for the circle, square and rectangle geometries (we define these in Section 4.2.6) for $\theta = \frac{\pi}{4}, \frac{\pi}{2}$, for a $20 \times 20$ cloak to demonstrate that the objective function is nonconvex; see Figure 4.3.

In all cases we see that the objective function is a nonconvex function of $t$. Because of this nonconvexity, our quasi-Newton yields only a local minimum. Consequently, we cannot guarantee that the objective associated with the solution of the relaxation found is a valid lower bound on the integer solution. In fact, we observe that even the rounding step can improve/reduce the objective value.

#### 4.2.6.3 Results for the Nominal Problem

Next, we evaluate how well the trust-region method performs from random starting points for all cloaking domains on a 20×20 mesh and a 40×40 control mesh. For each instance, we create 25



**(a)** Circle with $\theta = \frac{\pi}{4}$. **(b)** Square with $\theta = \frac{\pi}{4}$. **(c)** Rectangle with $\theta = \frac{\pi}{4}$.

**(d)** Circle with $\theta = \frac{\pi}{2}$. **(e)** Square with $\theta = \frac{\pi}{2}$. **(f)** Rectangle with $\theta = \frac{\pi}{2}$.

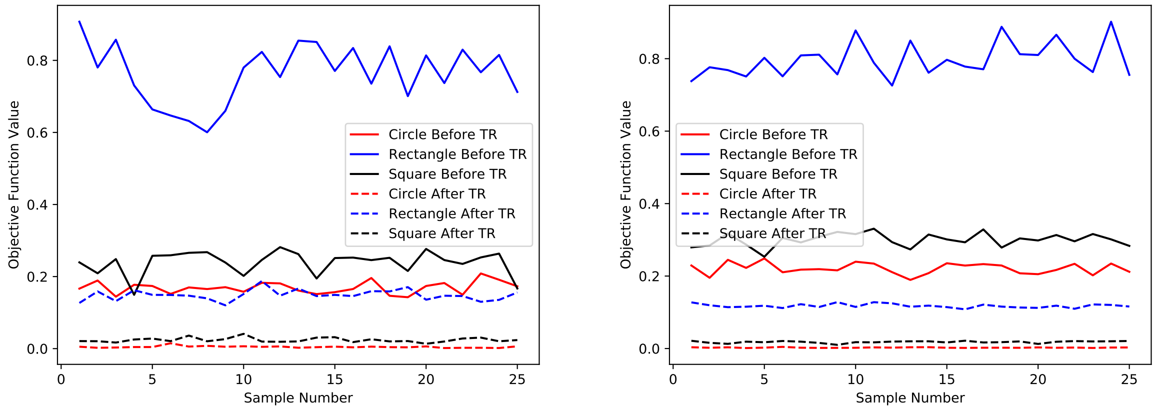
**Figure 4.3** Nonconvexity of the objective function for a single angle of attack.

random samples of $v \in \{0,1\}^N$, solve for the state, and plot the corresponding objective value. We then use each of these samples $v$ as a starting guess for our trust-region method, with $\theta = \frac{\pi}{4}$, and plot the objective at the end. The results are shown in Figure 4.4.

We observe that not only has the trust-region method reduced the objective function value in all cases but also that the variance of the final objective value, is significantly lower. We include a summary of the comparison in Table 4.1. We conclude that the trust-region method helps design competitive cloaking devices. We have observed similar behavior also in our designs under uncertainty.

Next, we investigate solving the relaxation of (4.28), with TAO for the rectangular domain with $\theta = \frac{\pi}{2}$ for the 20 × 20 and the 40 × 40 control mesh (Figures 4.5a and 4.5b respectively). We note that both the objective function value (Figures 4.5c and 4.5d) and the norm of the projected gradient (Figures 4.5e and 4.5f) are reduced; in both cases, TAO reduces the projected gradient below $10^{-3}$. We also note that the optimal solution for both mesh sizes is similar.

Next, we apply our rounding strategy (4.30) to obtain a rounded solution to use as a feasible starting point for our trust-region method. Our numerical results in Table 4.1 demonstrate that starting the trust-region method with an initial starting point of the rounded relaxation yields a lower objective value when compared to starting the trust-region method from a randomly generated



**(a)** 20×20 nominal. **(b)** 40×40 nominal.

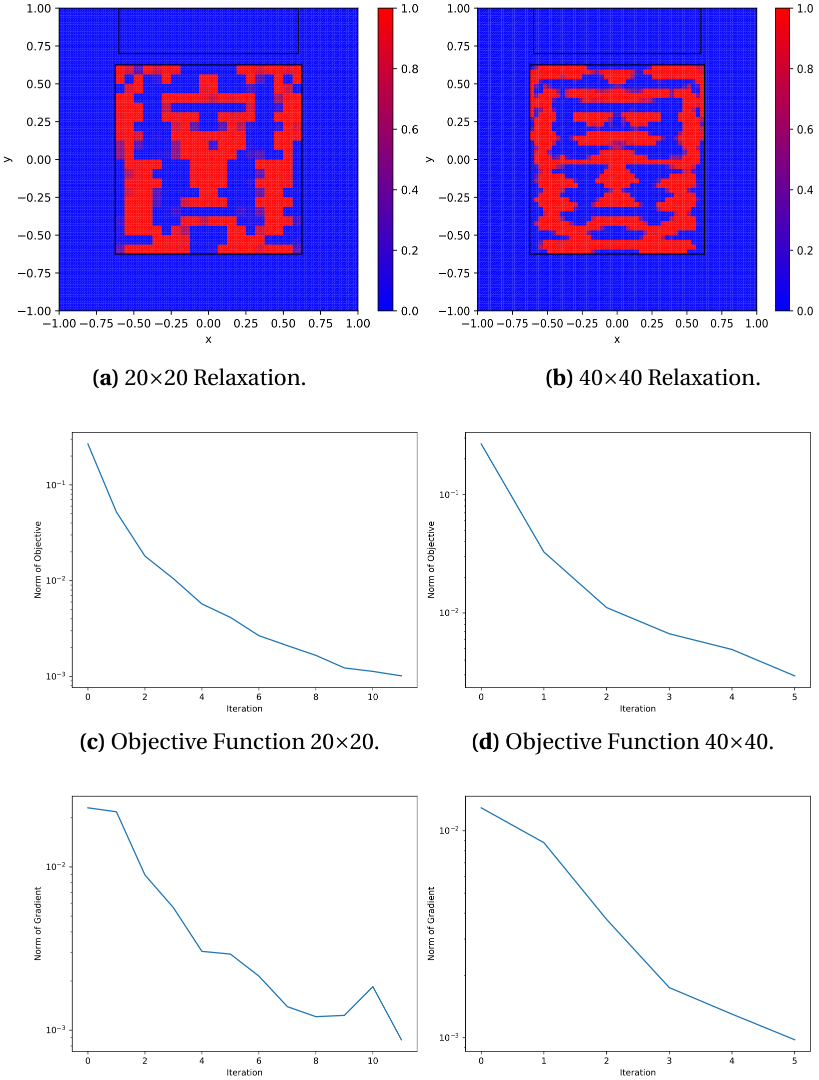
**Figure 4.4** Trust-region method applied to 25 randomly generated samples for $\theta = \frac{\pi}{4}$ and three cloaking domains. The solid lines show the objective function value at initial guess and the dashed lines of the corresponding color show the objective value after the trust-region method terminated.

design.

**Table 4.1** Comparison of the trust-region method final objective when starting from a random initial binary cloak versus starting from a cloak yield by rounding the continuous relaxation. We report the sample objective that was the lowest amoung the 25 samples

| Problem Instance | | | |
|---|---|---|---|
| Domain | Control | Trust-Region Sample | Trust-Region Rounded |
| $D_0^1$ | $20 \times 20$ | 0.1201 | 0.0168 |
| $D_0^2$ | $20 \times 20$ | 0.0134 | 0.0052 |
| $D_0^3$ | $20 \times 20$ | 0.0015 | 0.0011 |
| $D_0^1$ | $40 \times 40$ | 0.1081 | 0.0163 |
| $D_0^2$ | $40 \times 40$ | 0.0101 | 0.0032 |
| $D_0^3$ | $40 \times 40$ | 0.0013 | 0.0010 |

Next we investigate the value of solving the relaxation. Our numerical experiments demonstrate that starting the trust-region method with an initial starting point of the rounded relaxation yields a lower objective value when compared to starting the trust-region method from a randomly generated design. We demonstrate this for $\theta = \frac{\pi}{4}$ in Table 4.1 for the both $20 \times 20$ and $40 \times 40$ cloak on all geometries. We take the smallest objective from the result of our trust-region method after initializing the method with 25 different samples, as illustrated in Figure 4.4. We also solve the MIPDECO



**(a)** 20×20 Relaxation.

**(b)** 40×40 Relaxation.

**(c)** Objective Function 20×20.

**(d)** Objective Function 40×40.

**(e)** Norm of Projected Gradient 20× 20.

**(f)** Norm of Projected Gradient 40× 40.

**Figure 4.5** TAO relaxation results on the rectangular region with $\theta = \frac{\pi}{2}$ for $20 \times 20$ control mesh (left column) and $40 \times 40$ control mesh (right column).

relaxation, round it, then apply our trust-region algorithm to determine the objective value. We observe that in all cases the objective function value obtained by starting with the rounded relaxation is always lower than the lowest objective function value obtained by starting from 25 random samples. In the case of the circle geometry, we see only slight gain in using the rounded relaxation when compared to the sampling approach, however, in the case of the square and rectangle geometries we see that the objective function value is an order of magnitude smaller using the rounded relaxation as a starting iterate versus sampling.

Even though the relaxation takes a considerable amount of time to solve and sampling might appear attractive, we note that the size of the design space for a 20×20 and 40×40 control mesh is $2^{400}$ and $2^{1600}$, respectively, making it unlikely that sampling can beat the rounded relaxation. The success of our algorithm when started from the rounded relaxation solution is an indication that our trust-region approach may be able to find good local solutions in this application.

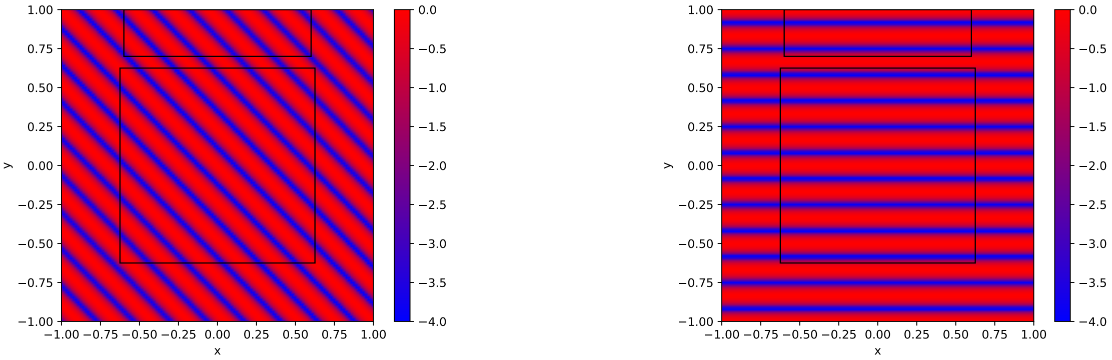
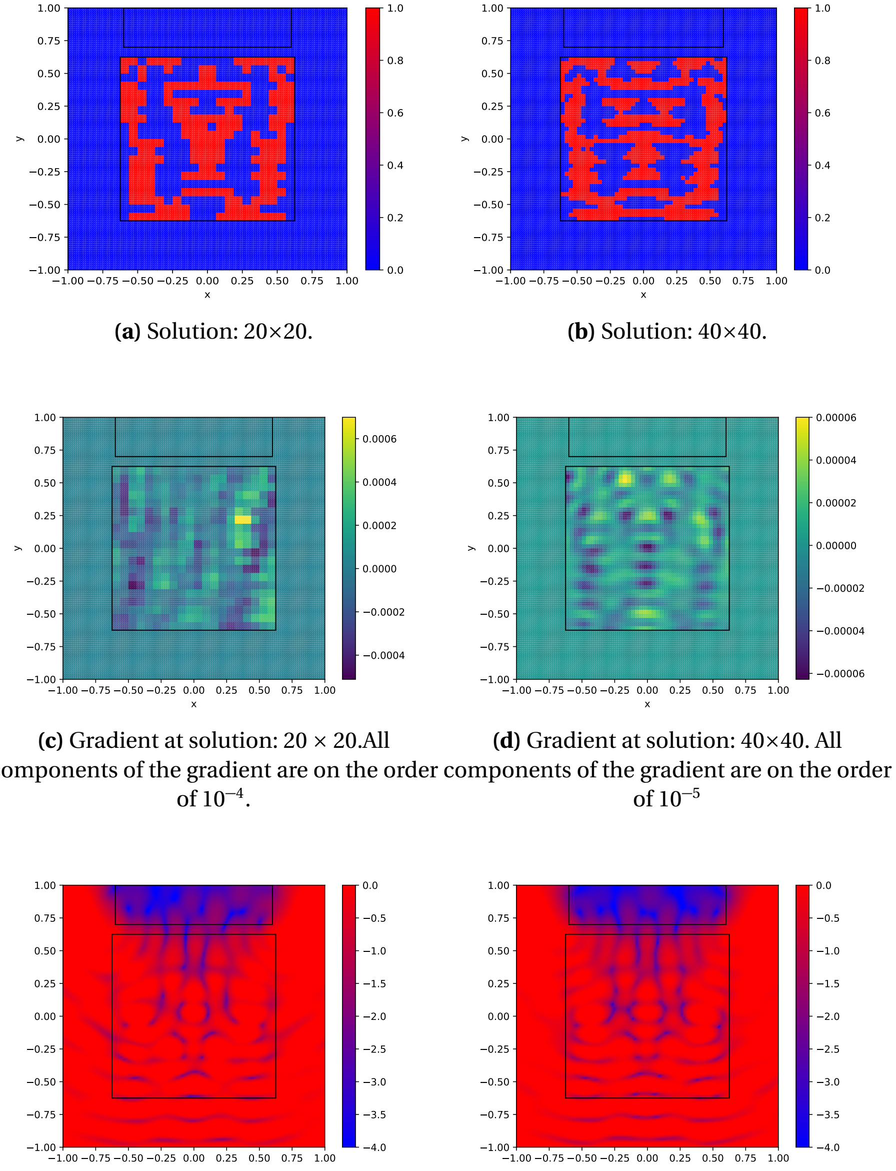


**Figure 4.6** $\log_{10}$ of $||\bar{u}_\theta||_{L^2(D_0)}$ (the unscattered field) for $\theta = \frac{\pi}{4}$ and $\theta = \frac{\pi}{2}$.

In Figures 4.7e and 4.7f we show a $\log_{10}$ plot of the scattering field, $\log_{10}(|u + \bar{u}_\theta|)$, and show that we reduce the integrand of the objective value in the region of interest to a small value. In addition, the gradient over the cloaking region, as shown in Figures 4.7c and 4.7d, is small, indicating that we have found a local optimum to (4.28). The unscattered field in the $\log_{10}$ scale is shown in Figure 4.6, when there is no cloak present ($v = 0$). When we compare the scattering field produced by our cloaks in Figure 4.7e and 4.7f with the unscattered field in Figure 4.6 we observe that our cloak achieves pointwise scattering in a large portion of the rectangular domain to a value on the order of $10^{-4}$. Without a cloak the domain has a scatter value on the order of one, demonstrating that our cloaks are successful at manipulating the incoming incident wave to minimize scattering in our rectangular domain and inducing cloaking ability. We also observe that the final integer cloak is quite similar to the relaxation, which indicates the relaxation is a good initial guess.



**(a)** Solution: 20×20.

**(b)** Solution: 40×40.

**(c)** Gradient at solution: 20 × 20.All components of the gradient are on the order of $10^{-4}$.

**(d)** Gradient at solution: 40×40. All components of the gradient are on the order of $10^{-5}$

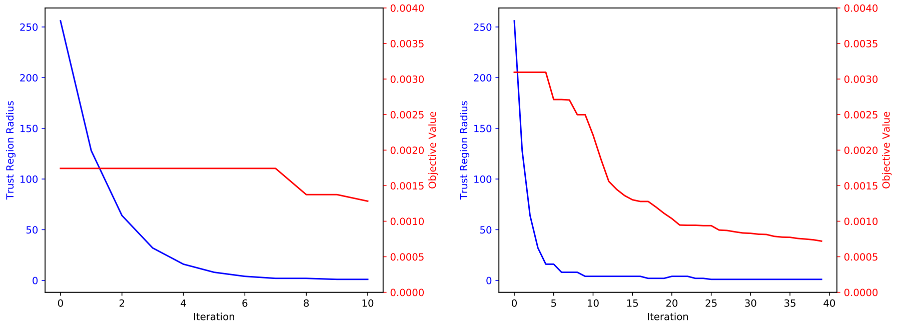
**(e)** $\log_{10}$ 20×20 Scattering Field.

**(f)** $\log_{10}$ 40× 40 Scattering Field.

**Figure 4.7** Results of the trust-region method applied to the rounded relaxation for $\theta = \frac{\pi}{2}$ for cloaking domain $D_0^1$.

We also compare the final discrete design in Figure 4.7 to the solution of the continuous relaxation in Figure 4.5. We note that the two designs are very similar, which we take as a further indication that our approach is able to find good-quality cloaking designs.

In Figure 4.8 we show the progress of our trust-region method in terms of objective function values and trust-region radii as a function of iteration for the 20×20 and 40×40 control meshes. We observe that our trust-region method is successful at reducing the incumbent objective function for the small trust-region radii regime, making steady improvement.



**(a)** 20×20.

**(b)** 40×40.

**Figure 4.8** Trust-region progress for 20×20 and 40×40 control mesh.

In Tables 4.2 and 4.3 the first column shows the domain and the second column shows the size of the control mesh. We report the CPU time in seconds for our trust-region method in Tables 4.2 and 4.3 for the nominal angle $\theta = \frac{\pi}{4}$ and $\theta = \frac{\pi}{2}$, respectively. The tables summarize the computational effort required to solve the MIPDECO. The first three measures show the CPU time of the relaxation solve, the CPU time of the forward (state), adjoint and gradient computation, and the CPU time of the knapsack solve. The CPU time for PDEs/gradient includes computational effort required to build $w(v)$. We include this to highlight the additional computational time required going from $20 \times 20$ to $40 \times 40$ cloak elements, because of the size of each of the respective gradients.

We observe from Tables 4.2 and 4.3 that the computation time for solving the knapsack problems, denoted as MIPs, is negligible. We also observe that the number of trust-region iterations required to meet our termination condition is modest. The trust-region methods time is divided into the amount of time it takes to solve the PDEs, build the gradient, and then solve the knapsack problem. We note that the number of iterations required to solve (4.28) grows only moderately as we increase the control mesh from 20×20 to 40×40, even though the design space grows exponentially.

In Table 4.4 we report numerical values of the objective associated with the nominal angles $\theta = \frac{\pi}{4}$ and $\theta = \frac{\pi}{2}$. The table contains three values: the relaxed objective, $\hat{J}(u^{\text{relax}})$, the rounded

**Table 4.2** CPU times(s) for nominal relaxation and PDE solves, as well as trust-region computation on the $128 \times 128$ mesh with $\theta = \frac{\pi}{4}$.

| Problem Instance | | Solution CPU Time(s) | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|
| Domain | Control | Relaxation | PDEs/Gradient | MIPs | Iterations | Figure |
| $D_0^1$ | $20 \times 20$ | 302.48 | 98.23 | 0.26 | 15 | 1 |
| $D_0^2$ | $20 \times 20$ | 125.93 | 122.85 | 0.59 | 20 | 5 |
| $D_0^3$ | $20 \times 20$ | 35.27 | 121.39 | 0.42 | 20 | 9 |
| $D_0^1$ | $40 \times 40$ | 1497.04 | 420.39 | 0.51 | 20 | 3 |
| $D_0^2$ | $40 \times 40$ | 141.06 | 1789.25 | 1.48 | 79 | 7 |
| $D_0^3$ | $40 \times 40$ | 196.24 | 1405.30 | 1.42 | 68 | 11 |

**Table 4.3** CPU times(s) for nominal relaxation and PDE solves, as well as trust-region computation on $128 \times 128$ mesh with $\theta = \frac{\pi}{2}$.

| Problem Instance | | Solution CPU Time(s) | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|
| Domain | Control | Relaxation | PDEs/Gradient | MIPs | Iterations | Figure |
| $D_0^1$ | $20 \times 20$ | 76.53 | 60.31 | 0.16 | 11 | 13 |
| $D_0^2$ | $20 \times 20$ | 41.77 | 84.77 | 0.22 | 17 | 17 |
| $D_0^3$ | $20 \times 20$ | 31.43 | 89.12 | 0.17 | 18 | 21 |
| $D_0^1$ | $40 \times 40$ | 202.40 | 917.72 | 1.06 | 40 | 15 |
| $D_0^2$ | $40 \times 40$ | 86.24 | 671.36 | 0.72 | 41 | 19 |
| $D_0^3$ | $40 \times 40$ | 56.55 | 872.90 | 0.98 | 50 | 23 |

relaxation objective function value, $\hat{J}(u^{\text{round}})$, and the trust-region objective function value $\hat{J}(u^{\text{trust}})$. In all simulations we are able to reduce the objective for the circle and square cases to a reasonably small value. The objective associated with the rectangle is much larger by comparison, which is due to its having a larger area. We see in several cases that our trust-region method is able to reduce the objective value. We observe that the relaxed objective function value does not always give a lower bound on the solution. In fact, in some cases, simply rounding already reduces the objective function value. We believe that this behavior is a result of the nonconvexity of our problem.

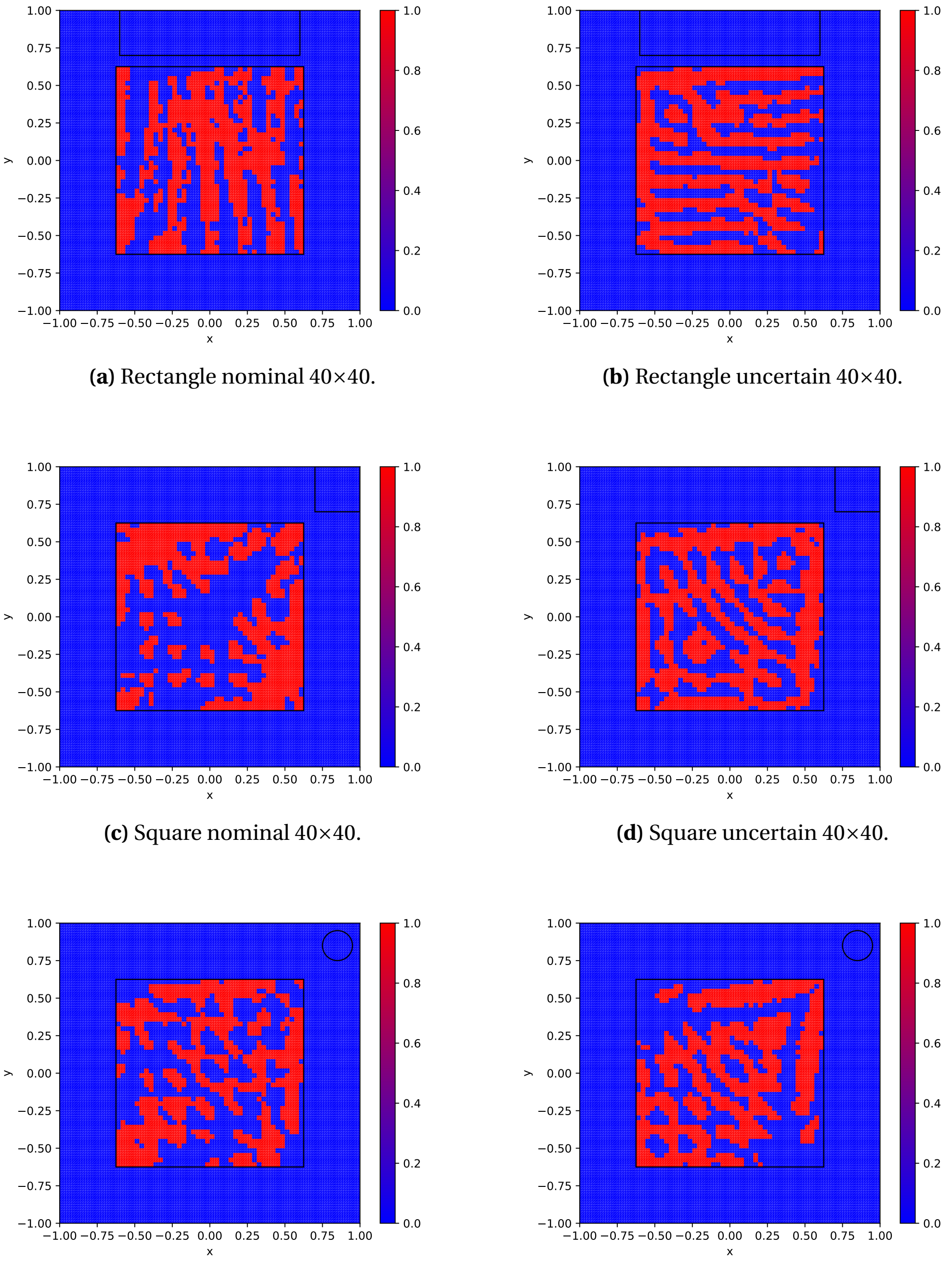
**Table 4.4** Objective function values for nominal problem.

| Problem Instance | | $\theta = \frac{\pi}{4}$ | | | $\theta = \frac{\pi}{2}$ | | |
|---|---|---|---|---|---|---|---|
| Domain | Control | Relaxed Objective | Relaxed Rounded Objective | Trust Region Objective | Relaxed Objective | Relaxed Rounded Objective | Trust Region Objective |
| $D_0^1$ | $20 \times 20$ | 0.0142 | 0.0321 | 0.0168 | 0.0010 | 0.0017 | 0.0012 |
| $D_0^2$ | $20 \times 20$ | 0.0054 | 0.0077 | 0.0052 | 0.0032 | 0.0097 | 0.0036 |
| $D_0^3$ | $20 \times 20$ | 0.0015 | 0.0014 | 0.0011 | 0.0002 | 0.0126 | 0.0017 |
| $D_0^1$ | $40 \times 40$ | 0.0168 | 0.0219 | 0.0163 | 0.0029 | 0.0030 | 0.0007 |
| $D_0^2$ | $40 \times 40$ | 0.0068 | 0.0077 | 0.0032 | 0.0039 | 0.0187 | 0.0031 |
| $D_0^3$ | $40 \times 40$ | 0.0019 | 0.0016 | 0.0010 | 0.0010 | 0.0126 | 0.0008 |

#### 4.2.6.4 Results for the Uncertainty Problem

We now present our results for (4.29). In Figure 4.9 we compare the design found for the 40×40 instance for a nominal angle of $\theta = \frac{\pi}{4}$ and the design found from solving the uncertainty problem for 15 angles on $\left[0, \frac{\pi}{2}\right]$ for $D_0^1, D_0^2$, and $D_0^3$. Similar results for the nominal angle of $\theta = \frac{\pi}{2}$ can be found in the electronic supplement .1. We see a clear difference in design for the rectangle geometry.

In 4.5 and 4.6 we repeat the computational effort required to solved the uncertainty simulation for 15 angles. We again take $v^{(0)} = 0.5$ as a starting guess in TAO. As in the nominal case, solving the relaxation takes the most computational effort compared with the trust-region method. Even though more angles are present in the uncertainty problem, the number of trust-region iterations does not increase drastically. As in the nominal case, the computational time in the trust-region method is dominated by solving the PDEs and constructing the gradient, not by solving the knapsack problems. Another similarity is the number of iterations required by our trust-region method, which increases modestly as we go from optimizing over $20 \times 20$ integer variables to $40 \times 40$. The number of trust-region iterations is not dependent on the number of angles since we see similar iteration counts between a single angle and 15 angles. In the electronic supplement .1 we include plots for all



(**a**) Rectangle nominal 40×40.

(**b**) Rectangle uncertain 40×40.

(**c**) Square nominal 40×40.

(**d**) Square uncertain 40×40.

(**e**) Circle nominal 40×40.

(**f**) Circle uncertain 40×40.

**Figure 4.9** Cloak designs due to 15 uniformly distributed $\theta \in \left[0, \frac{\pi}{2}\right]$ and nominal angle $\frac{\pi}{4}$.

numerical simulations.

**Table 4.5** CPU times(s) for uncertain relaxation, PDE/gradient calculation and knapsack problem on $128 \times 128$ mesh for 15 angles where $\theta \in \left[0, \frac{\pi}{2}\right]$.

| Problem Instance | | Solution CPU Time(s) | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|
| Domain | Control | Relaxation | PDEs/Gradient | MIPs | Iterations | Figure |
| $D_0^1$ | $20 \times 20$ | 1094.43 | 690.49 | 0.31 | 19 | 2 |
| $D_0^2$ | $20 \times 20$ | 158.04 | 762.72 | 0.34 | 23 | 6 |
| $D_0^3$ | $20 \times 20$ | 121.82 | 418.73 | 0.11 | 13 | 10 |
| $D_0^1$ | $40 \times 40$ | 1996.71 | 1005.43 | 0.23 | 18 | 4 |
| $D_0^2$ | $40 \times 40$ | 355.67 | 2444.14 | 0.77 | 44 | 8 |
| $D_0^3$ | $40 \times 40$ | 306.21 | 2664.15 | 0.79 | 48 | 12 |

**Table 4.6** CPU times(s) for uncertain relaxation, PDE/gradient calculation and knapsack problem on $128 \times 128$ mesh for 10 angles where $\theta \in \left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.

| Problem Instance | | Solution CPU Time(s) | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|
| Domain | Control | Relaxation | PDEs/Gradient | MIPs | Iterations | Figure |
| $D_0^1$ | $20 \times 20$ | 2093.23 | 430.97 | 0.18 | 17 | 14 |
| $D_0^2$ | $20 \times 20$ | 137.17 | 430.98 | 0.16 | 17 | 18 |
| $D_0^3$ | $20 \times 20$ | 135.26 | 361.15 | 0.13 | 14 | 22 |
| $D_0^1$ | $40 \times 40$ | 2495.73 | 3900.51 | 0.91 | 61 | 16 |
| $D_0^2$ | $40 \times 40$ | 326.36 | 3715.97 | 0.89 | 60 | 20 |
| $D_0^3$ | $40 \times 40$ | 62.42 | 5986.25 | 1.64 | 105 | 24 |

In Table 4.7, we show the solution quality for the uncertain case. We see in many cases, similar to the single angle case, that we are able to reduce the objective function value below the relaxed objective function value with our trust-region method. This is not unexpected because, as demonstrated in Figure 4.3 for a single angle, our problem is nonconvex.

We also compare nominal design compares with the uncertainty design over a range of angles for the circular domain. We define the function

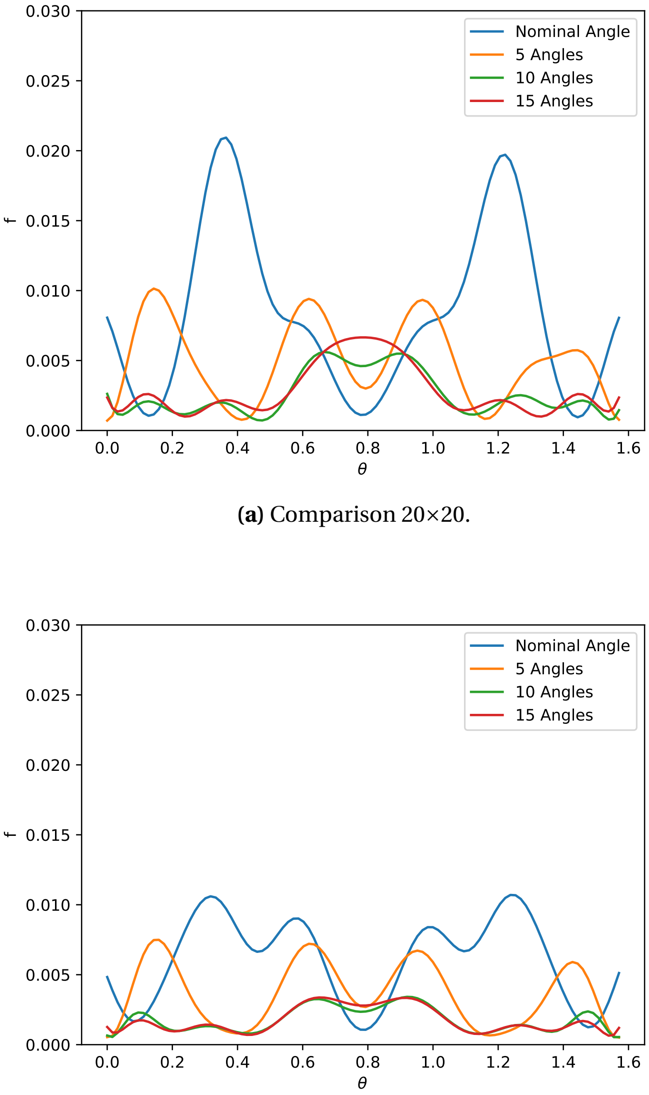
$$f(\tilde{\theta}) = \hat{J}^h(\tilde{\theta}; v^{trust}). \tag{4.37}$$

We take the solution of the trust-region method from the nominal and the uncertainty case with a given $\tilde{\theta}$, solve the state, and evaluate the objective value to obtain $f(\tilde{\theta})$. We compare the nominal

**Table 4.7** Objectives for uncertain problem with 15 angles.

| Problem Instance | | $\theta \in \left[0, \frac{\pi}{2}\right]$ | | | $\theta \in \left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$ | | |
|---|---|---|---|---|---|---|---|
| Domain | Control | Relaxed Objective | Relaxed Rounded Objective | Trust Region Objective | Relaxed Objective | Relaxed Rounded Objective | Trust Region Objective |
| $D_0^1$ | $20 \times 20$ | 0.0927 | 0.0941 | 0.0902 | 0.0773 | 0.0770 | 0.0762 |
| $D_0^2$ | $20 \times 20$ | 0.0106 | 0.0107 | 0.0083 | 0.0196 | 0.0210 | 0.0190 |
| $D_0^3$ | $20 \times 20$ | 0.0042 | 0.0034 | 0.0028 | 0.0086 | 0.0093 | 0.0082 |
| $D_0^1$ | $40 \times 40$ | 0.0818 | 0.0818 | 0.0077 | 0.0614 | 0.0647 | 0.0597 |
| $D_0^2$ | $40 \times 40$ | 0.0111 | 0.0101 | 0.0071 | 0.0220 | 0.0220 | 0.0183 |
| $D_0^3$ | $40 \times 40$ | 0.0043 | 0.0030 | 0.0017 | 0.0197 | 0.0126 | 0.0090 |

design for $\theta = \frac{\pi}{4}$ with the uncertainty design found using 5, 10, and 15 angles from the interval $\left[0, \frac{\pi}{2}\right]$ on the 20×20 and 40×40 control meshes, respectively. We evaluate $f$ on $\tilde{\theta} = \left[0, \frac{\pi}{2}\right]$ by breaking the interval into 100 sample points and then plot the resulting objective value as a function of $\theta$, as shown in Figure 4.10. We observe that the nominal design produces a lower objective value at $\frac{\pi}{4}$ when compared to the uncertain designs, which is expected. However, when we move away from the nominal angle, $\theta = \frac{\pi}{4}$, we observe that the uncertain design preforms better than the nominal design. We built the nominal design to cloak for a single angle, so it makes sense that it is superior to the uncertain design at $\theta = \frac{\pi}{4}$, which covers several angles. Similarly, because the uncertain design was developed for several angles, it performs better across a range of angles.



**(a)** Comparison 20×20.

**(b)** Comparison 40×40.

**Figure 4.10** Comparison of nominal versus robust cloak design over a range of angles $\theta$.

## 4.3 Conclusions

In this chapter, we introduced a steepest descent trust-region method for MIPDECO using an adjoint-based gradient. Our approach to MIPDECO is systematic and to our knowledge the first OTD approach for MIPDECO. We first solve the continuous relaxation of the MIPDECO, then we use a rounding heuristic to produce a feasible control for the MIPDECO. We use this feasible control as a starting point for our trust-region method. We have shown that our approach produces minimizers with lower objective function value when compared to the objective function value of minimizers produced by random feasible starting guesses. Our numerical solutions have demonstrated that our algorithm produces quality minimizers for our application problem and is, more importantly, computationally problem. In fact the computational effort required to solve the integer portion of the MIPDECO is negligible to the overall effort required to execute the trust-region algorithm.

CHAPTER

# 5

# A MIXED-INTEGER PDE-CONSTRAINED OPTIMIZATION FORMULATION FOR THE CONSTRUCTION OF ELECTROMAGNETIC CLOAKS WITH SEVERAL MATERIALS IN 2D

In this chapter we introduce a generalization of the cloaking problem from chapter 4. In this problem we allow the cloak to be made out of several materials and constrain the maximum mass the cloak. We study how the introduction of these constraints on the problem affect our trust-region algorithm and the computational effort to solve the MIPDECO. We follow the same blueprint provided for the single material problem. We introduce a rounding heuristic that maintains feasibility for this application and we also introduce two continuations methods to improve our cloak designs. We then present numerical results to show the effectiveness of the cloaks generated by our trust-region and continuation approach, and compare these approaches.

## 5.1 Introduction

The recent work of Haslinger [41] introduced a topology optimization problem for constructing electromagnetic cloaks. The formulation assumes that the cloak is made out of a single material. In our previous work in chapter 4, we posed a mixed-integer partial differential equation constrained optimization (MIPDECO) formulation based on this topology optimization problem to design an electromagnetic cloak. Our formulation accounted for uncertainty in the angle of attack of the electromagnetic wave. Using a reduced-space formulation of MIPDECO for binary electromagnetic cloaking, our method solved the continuous relaxation, rounded the solution, and used the rounded solution as an initial starting point in a steepest-descent trust-region method. We chose to solve the reduced-space problem using the optimize-then-discretize approach where we took advantage of the adjoint equation associated with the MIPDECO to compute the gradient. In each iteration of the trust-region method, only the weak state and adjoint equation were solved to update the gradient for the trust-region problem. The trust-region subproblem is a binary knapsack problem that can be solved quickly. Hence, our approach for solving MIPDECO problems can be viewed as sequentially solving PDEs and mixed-integer programs. Our numerical results showed that our trust-region method was computationally tractable and that the solutions provided effective cloaking ability.

Here, we introduce a generalized topology optimization problem based on Haslinger's formulation and on our previous work in chapter 4 that allows the electomagnetic cloak to be designed from several different materials. We extend the single-material formulation from chapter 4 in two important ways: (1) we allow multiple materials in the cloak with different material properties; and (2) we add a mass constraint. The material used to build a cloak directly impacts the cost of the cloak, which can create a financial incentive to build less massive cloaks. If the quality of the cloaking ability is not adversely impacted, then there is no reason to build more massive cloaks. We add a mass constraint to place an upper bound on the mass of the electromagnetic cloak.

To design an electromagnetic cloak from multiple materials, we let $D \subset \mathbb{R}^2$ be the computational domain and $D_0 \subset D$ be the part of the domain we wish to cloak. We let $\widehat{D} \subset D - D_0$ be the subregion in which we can design the cloak and $\hat{\Omega} \subset \widehat{D}$ be the topology of the cloak. Our interest is to minimize the scattering that occurs in $D_0$ to induce optical invisibility. We introduce the following topology optimization problem to construct an optimal electromagnetic cloak out of $p \geq 1$ materials while also enforcing an upper bound on the mass of the cloak:

$$\begin{aligned}
\underset{u,\, q,\, \hat{\Omega}\subset\widehat{D}}{\text{minimize}} \quad & \frac{1}{2}\|u+\bar{u}_\theta\|^2_{2,D_0} \\
\text{subject to} \quad & -\Delta u - k_0^2\big(1+q\mathbb{1}_{\hat{\Omega}}\big)u = k_0^2 q\mathbb{1}_{\hat{\Omega}}\bar{u}_\theta && \text{in } D, \\
& \frac{\partial u}{\partial n} - ik_0 u = 0 && \text{on } \partial D, \\
& (q(x,y),\rho(x,y)) \in \{(0,0),(q_1,\rho_1),\ldots,(q_p,\rho_p)\}, \\
& \int_{\hat{\Omega}} \rho(x,y)d\hat{\Omega} \le C,
\end{aligned} \tag{5.1}$$

where $i=\sqrt{-1}$, $\partial D$ is the boundary of the computational domain and

$$\bar{u}_\theta(x,y) = e^{(ik_0(x\cos(\theta)+y\sin(\theta)))} \tag{5.2}$$

is the incidence wave in direction $d=(\cos(\theta),\sin(\theta))$, and $k_0$ is the wave number. The angle $\theta$ at which the incidence wave approaches the cloak is measured from the x-axis and we refer to $\theta$ as the angle of attack. The state variable $u$ represents the electromagnetic field intensity. In this formulation we allow the cloak to be made out of $p$ materials. Each material has a relative permittivity $q_i$, with corresponding density $\rho_i$ for $i=1,\ldots,p$. We measure the relative permittivity to zero which corresponds to the uniform permittivity found outside of the cloak in $D-\hat{\Omega}$. The parameter $C$ serves as an upper bound for the cloaks mass. Because $u,\bar{u}_\theta:\mathbb{R}^2\to\mathbb{C}$ is complex valued, we split $u$ and $\bar{u}_\theta$ into its real and imaginary parts, $u=u_{\text{Re}}+iu_{\text{Im}}$ and $\bar{u}_\theta=\bar{u}_{\text{Re}}+i\bar{u}_{\text{Im}}$. We view the problem in term of the real and imaginary parts instead of the complex valued quantities $u$ and $\bar{u}_\theta$.

We study how the introduction of several materials can improve cloaking ability and how much of an impact the mass constraint has on the quality of the cloaking ability. The introduction of this mass constraint makes the design problem more complex by adding an additional resource constraint. In some instances, we have observed inconsistent optimal solutions as we vary the upper bound on the mass, due to the nonconvexity of the objective function. Hence, we introduce a continuation scheme to create a heuristic that improves the electromagnetic cloaks in a sequential manner. We show that by using our continuation method, we obtain cloaks with improved cloaking ability when compared to producing cloaks by means of only solving the relaxation, rounding, and then applying our trust-region method. We provide numerical experiments and cloak designs obtained from our approach and report the computational effort.

**Chapter Outline:** In Section 5.2, we define the infinite-dimensional MIPDECO for electromagnetic cloaking. This formulation allows multiple materials to be used in the cloaking device and sets an upper bound on the mass of the cloak. In Section 5.3, we derive the optimality conditions to produce an adjoint-based gradient in infinite dimensions using the formal Lagrangian approach and we discuss the discretization of the state and adjoint equation and the gradient. In Section 5.4, we introduce a rounding scheme that enforces our mass constraint. We then discuss a trust-region method for MIPDECO with constraints on the control variable. Next we introduce our continuation method. We provide numerical experiments in Section 5.5 to demonstrate the cloaking ability of cloaks produced by our trust-region method and we comment on the computational effort to solve our problems. We then conclude by summarizing our results in Section 5.6.

## 5.2 Multi-Material Electromagnetic Cloak

For our MIPDECO based on the topology optimization problem introduced in (5.1), we assume that the cloak $\hat{\Omega}$ is a box. We then partition the cloak $\hat{\Omega}$ into a finite number of non-overlapping uniform squares, $\hat{\Omega}_n$, for $n = 1, \ldots, N$. We refer to this partition as the *control mesh* and assume that

$$\hat{\Omega} = \bigcup_{n=1}^{N} \hat{\Omega}_n \quad \text{and} \quad \left(\hat{\Omega}_n \setminus \partial\hat{\Omega}_n\right) \cap \left(\hat{\Omega}_m \setminus \partial\hat{\Omega}_m\right) = \emptyset \text{ for } n \neq m.$$

We define two vectors, the relative permittivity vector $q \in \mathscr{R}^p$, $p \geq 1$, which is relative to the uniform permittivity found outside the cloak, $D - \hat{\Omega}$, and the density vector $\rho \in \mathscr{R}^p$:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_p \end{bmatrix}, \quad \rho = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_p \end{bmatrix}. \tag{5.3}$$

We define a decision vector $v \in \{0,1\}^{pN}$ that defines the design of our cloak as

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}, \quad \text{where each} \quad v_n = \begin{bmatrix} v_n^1 \\ v_n^2 \\ \vdots \\ v_n^p \end{bmatrix}, \tag{5.4}$$

for $n = 1, \ldots, N$ control elements. We illustrate a sample design in Figure 5.1, where we have two materials, yellow and red, and purple corresponds to no material. In this formulation we associate the binary vector $v_n$ with the $n$-th control element $\hat{\Omega}_n$ by defining a material function, $w(x, y)$,

$$w = \sum_{n=1}^{N} \left(q^T v_n\right) \mathbb{1}_{\hat{\Omega}_n \setminus \partial\hat{\Omega}_n} = \sum_{n=1}^{N} \sum_{i=1}^{p} q_i v_n^i \mathbb{1}_{\hat{\Omega}_n \setminus \partial\hat{\Omega}_n}. \tag{5.5}$$

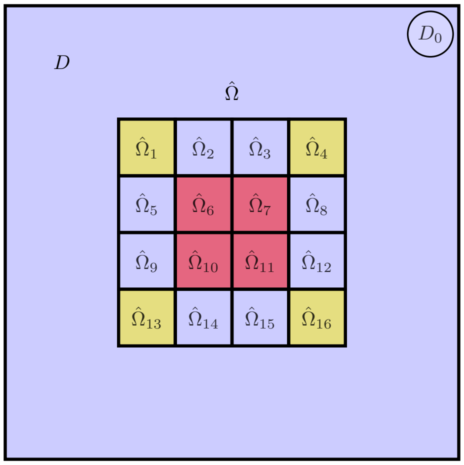


**Figure 5.1** An illustration of a 4×4 grid of control elements in which the goal is to cloak the region $D_0$. In this case $D_0$ is a rectangle in the top region domain. We choose from two materials build our cloak in this illustration.

To ensure that at most one type of material is selected in each control element, we add special-order-set of type one (SOS1) constraints [30, 99, 100]:

$$\sum_{i=1}^{p} v_n^i \leq 1 \qquad \forall n = 1, \ldots, N. \tag{5.6}$$

If $v_n = 0$ then we we place no material in the control element $\hat{\Omega}_n$. This type of constraint has been studied extensively and the structure of this constraint can be exploited when preforming branch-and-bound.

In addition, we add a constraint on the the mass of the cloak. We compute the mass of the cloak as the sum of the individual mass contributions from each $\hat{\Omega}_n$:

$$m(v) = \sum_{n=1}^{N} \left( \int_{\hat{\Omega}_n} \sum_{i=1}^{p} \rho_i v_n^i d\hat{\Omega}_n \right) = \sum_{n=1}^{N} \left( \int_{\hat{\Omega}_n} \rho^T v_n d\hat{\Omega}_n \right) = \sum_{n=1}^{N} \rho^T v_n |\hat{\Omega}_n|, \tag{5.7}$$

where

$$|\hat{\Omega}_n| = \int_{\hat{\Omega_n}} 1 d\hat{\Omega}_n, \tag{5.8}$$

and we add the constraint $m(v) \leq C$. We experiment with different values of $C$ to illustrate the effect the mass constraint has on the optimal design and its cloaking ability. We now summarize our new

MIPDECO formulation:

$$
\begin{aligned}
\underset{u,v,w}{\text{minimize}} \quad & \mathscr{J}(u,v,w) = \frac{1}{2|D_0|}\|u+\bar{u}_\theta\|^2_{2,D_0} & & \\
\text{subject to} \quad & -\Delta u - k_0^2(1+w)u = k_0^2 w\,\bar{u}_\theta & & \text{in } D \\
& \frac{\partial u}{\partial n} - i k_0 u = 0 & & \text{on } \partial D \\
& w = \sum_{n=1}^{N}\left(q^T v_n\right)\mathbb{1}_{\hat{\Omega}_n\setminus\partial\hat{\Omega}_n} & & \\
& v_n \in \{0,1\}^p & & \forall n = 1,\dots,N, \\
& \sum_{i=1}^{p} v_n^i \le 1 & & \forall n = 1,\dots,N \\
& m(v) \le C & & \forall n = 1,\dots,N,
\end{aligned}
\tag{5.9}
$$

where $C$ is an upper bound on the mass of the cloak. We note that (5.9) is a full-space formulation of the MIPDECO. We established in chapter 4 that for any given $v$ both the state $u$ and control function $w$ are uniquely defined. Therefore, we solve a reduced-space formulation where instead of optimizing over $(u, v, w)$, we optimize over $v$. The objective function of this reduced approach can be viewed as $\mathscr{J}(v) = \mathscr{J}(u(v), v, w(v))$. We can obtain the gradient efficiently using the adjoint-based gradient [97]. We also introduce a scaling parameter $|D_0| = \int_{D_0} 1 dD_0$ so that the objective function value is normalized to accurately measure the quality of our cloak.

## 5.3 Optimality Conditions and Discretization

In chapter 4, we outlined how to obtain the gradient of the objective, $\nabla_v \mathscr{J}(v)$, by means of the adjoint-based approach [97]. We omit the details here and provide a summary instead.

To build the gradient, we pose the partial Lagrangian associated with (5.9):

$$
\begin{aligned}
&\mathscr{L}(u_{\text{Re}}, u_{\text{Im}}, v, \lambda_{\text{Re}}, \lambda_{\text{Im}}; \theta) \\
&= \frac{1}{2|D_0|}\int_{D_0} \left(u_{\text{Re}} + \bar{u}_{\text{Re}}\right)^2 + \left(u_{\text{Im}} + \bar{u}_{\text{Im}}\right)^2 dD_0 \\
&+ \int_D \nabla u_{\text{Re}} \cdot \nabla \lambda_{\text{Re}} dD + \int_{\partial D} k_0 u_{\text{Im}} \lambda_{\text{Re}} d\partial D \\
&\qquad - \int_D \left[\lambda_{\text{Re}}\left[k_0^2(1+w)u_{\text{Re}} + k_0^2 w \bar{u}_{\text{Re}}\right]\right] dD \\
&+ \int_D \nabla u_{\text{Im}} \cdot \nabla \lambda_{\text{Im}} dD - \int_{\partial D} k_0 u_{\text{Re}} \lambda_{\text{Im}} d\partial D \\
&\qquad - \int_D \left[\lambda_{\text{Im}}\left[k_0^2(1+w)u_{\text{Im}} + k_0^2 w \bar{u}_{\text{Im}}\right]\right] dD,
\end{aligned}
\tag{5.10}
$$

where $u_{\text{Re}}, u_{\text{Im}}$ are the solution to the real and imaginary part of the state equation, and $\lambda_{\text{Re}}, \lambda_{\text{Im}}$ are the adjoint (multiplier) variables associated with the real and imaginary state equation, respectively. The reader may be curious why this Lagrangian does not take into account the SOS1 and mass constraints. The reason is because these constraints do not contain the state variables, so they have no contribution to the total derivative $\nabla_v \mathscr{J}$ and we treat these constraints explicitly in the reduced-space optimization problem. We follow our approach from chapter 4 to derive the weak state and adjoint equations for our new formulation (5.9). The weak state equations are

$$
\begin{aligned}
0 = \mathscr{L}_{\lambda_{\text{Re}}}[\tilde{\lambda}_{\text{Re}}] = &\int_D \nabla u_{\text{Re}} \cdot \nabla \tilde{\lambda}_{\text{Re}} dD + \int_{\partial D} k_0 u_{\text{Im}} \tilde{\lambda}_{\text{Re}} d\partial D \\
&- \int_D \tilde{\lambda}_{\text{Re}}\left[k_0^2(1+w)u_{\text{Re}} + k_0^2 w \bar{u}_{\text{Re}}\right] dD \qquad \forall \tilde{\lambda}_{\text{Re}} \in H^1(D),
\end{aligned}
\tag{5.11}
$$

$$
\begin{aligned}
0 = \mathscr{L}_{\lambda_{\text{Im}}}[\tilde{\lambda}_{\text{Im}}] = &\int_D \nabla u_{\text{Im}} \cdot \nabla \tilde{\lambda}_{\text{Im}} dD - \int_{\partial D} k_0 u_{\text{Re}} \tilde{\lambda}_{\text{Im}} d\partial D \\
&- \int_D \tilde{\lambda}_{\text{Im}}\left[k_0^2(1+w)u_{\text{Im}} + k_0^2 w \bar{u}_{\text{Im}}\right] dD, \qquad \forall \tilde{\lambda}_{\text{Im}} \in H^1(D),
\end{aligned}
\tag{5.12}
$$

and the weak adjoint equations are

$$
\begin{aligned}
0 = \mathscr{L}_{u_{\mathrm{Re}}}[\tilde{u}_{\mathrm{Re}}] = \frac{1}{|\Theta(s)|}\int_{D_0} \big(u_{\mathrm{Re}} + \bar{u}_{\mathrm{Re}}\big)\tilde{u}_{\mathrm{Re}}\, dD_0 & \\
+ \int_D \nabla \tilde{u}_{\mathrm{Re}} \cdot \nabla \lambda_{\mathrm{Re}}\, dD - \int_D \lambda_{\mathrm{Re}} k_0^2 (1 + qw)\tilde{u}_{\mathrm{Re}}\, dD & \\
- \int_{\partial D} k_0 \tilde{u}_{\mathrm{Re}} \lambda_{\mathrm{Im}}\, d\partial D, & \qquad \forall \tilde{u}_{\mathrm{Re}} \in H^1(D),
\end{aligned}
\tag{5.13}
$$

$$
\begin{aligned}
0 = \mathscr{L}_{u_{\mathrm{Im}}}[\tilde{u}_{\mathrm{Im}}] = \frac{1}{|\Theta(s)|}\int_{D_0} \big(u_{\mathrm{Im}} + \bar{u}_{\mathrm{Im}}\big)\tilde{u}_{\mathrm{Im}}\, dD_0 & \\
+ \int_D \nabla \tilde{u}_{\mathrm{Im}} \cdot \nabla \lambda_{\mathrm{Im}}\, dD - \int_D \lambda_{\mathrm{Im}} k_0^2 (1 + qw)\tilde{u}_{\mathrm{Im}}\, dD & \\
+ \int_{\partial D} k_0 \tilde{u}_{\mathrm{Im}} \lambda_{\mathrm{Re}}\, d\partial D, & \qquad \forall \tilde{u}_{\mathrm{Im}} \in H^1(D).
\end{aligned}
\tag{5.14}
$$

Once we have solved for the state and adjoint variables we build each component of the gradient, $v_n^i$, by calculating the integral:

$$
\nabla_{v_n^i} \mathscr{J}(v) = -\int_{\hat{\Omega}_n} \Big( k_0^2 q_i (u_{\mathrm{Re}}\lambda_{\mathrm{Re}} + u_{\mathrm{Im}}\lambda_{\mathrm{Im}}) + k_0^2 q_i (\bar{u}_{\mathrm{Re}}\lambda_{\mathrm{Re}} + \bar{u}_{\mathrm{Im}}\lambda_{\mathrm{Im}}) \Big) d\hat{\Omega}_n,
$$

for $n = 1, \dots, N$ and $i = 1, \dots, p$.

To solve the weak state and adjoint equations numerically, we discretize the infinite dimensional equations using the finite-element method (FEM). We represent each solution to the weak state and adjoint equations as a linear combination of piecewise linear basis functions, with a mesh discretization parameter of $h$ in both the $x$ and $y$ dimension. We denote these solutions for a fixed $h$ as $u^h_{\mathrm{Re}}, u^h_{\mathrm{Im}}, \lambda^h_{\mathrm{Re}}, \lambda^h_{\mathrm{Im}}$. We choose piecewise linear elements to obtain second-order accuracy in the state and adjoint variable approximations, as discussed in chapter 4. Once we have solved for the FEM approximation of the state and adjoint equations we produce a discretized gradient by applying a two-point Gaussian quadrature rule to the following integral using the FEM solutions:

$$
\nabla_{v_n^i} J^h(v) = -\int_{\hat{\Omega}_n} \Big( k_0^2 q_i (u^h_{\mathrm{Re}}\lambda^h_{\mathrm{Re}} + u^h_{\mathrm{Im}}\lambda^h_{\mathrm{Im}}) + k_0^2 q_i (\bar{u}^h_{\mathrm{Re}}\lambda^h_{\mathrm{Re}} + \bar{u}^h_{\mathrm{Im}}\lambda^h_{\mathrm{Im}}) \Big) d\hat{\Omega}_n,
\tag{5.15}
$$

for $n = 1, \dots, N$ and $i = 1, \dots, p$. We call $\nabla_v J^h$ the approximation to the true gradient $\nabla_v \mathscr{J}$.

## 5.4 Solving the Multi-material Cloaking Problem

In chapter 4, we introduced a steepest-descent trust-region method. In this approach we first solve a continuous relaxation of (5.9). This relaxation of (5.9) relaxes $v$ to be continuous, $0 \leq v \leq 1$. Next we take the relaxed solution and apply a rounding heuristic to obtain a feasible solution for (5.9). Then we apply a discrete steepest-descent trust-region method to improve this rounded solution. A continuation scheme is then provided to further improve the cloak designs.

### 5.4.1 A Rounding Heuristic for the Relaxation

Here, we present a deterministic rounding heuristic inspired by the feasibility pump, a heuristic for obtaining good initial guesses for mixed-integer linear programming (MILP) and mixed-integer nonlinear programming (MINLP) problems [1, 9, 11, 31, 33]. Let $\tilde{v} \in [0,1]^{pN}$ and $\tilde{w} \in [0, q_p]^N$ be the solution of the relaxation of (5.9), and construct an integer feasible point, $v \in [0,1]^{pN}$ and $w \in \{0, q_1, \ldots, q_p\}^N$, that is as close as possible to $(\tilde{v}, \tilde{w})$. Letting $w_n = q^T v_n$, where $w_n$ is the constant value of $w$ on $\hat{\Omega}_n$, we formulate this problem as the following MILP:

$$\begin{aligned} \underset{v,w}{\text{minimize}} \quad & ||w - \tilde{w}||_1 \\ \text{subject to} \quad & w_n = q^T v_n \\ & \sum_{n=1}^{N} \rho^T v_n |\hat{\Omega_n}| \leq C \\ & \sum_{i=1}^{p} v_n^i \leq 1 \qquad \forall n = 1, \ldots, N \\ & v_n \in \{0,1\}^p \qquad \forall n = 1, \ldots, N \\ & w \in \mathbb{R}^N \\ & v \in \{0,1\}^{pN}. \end{aligned} \tag{5.16}$$

We let our initial guess for our trust-region method be the solution $(v, w)$ of (5.16), which is the point that has a feasible, discrete cloaking pattern defined by $\bar{w}$ that is as close as possible to the continuous relaxation.

We could use the $\ell_2$-norm typically used in the feasibility pump in the objective. Unlike the full feasibility pump, however, we solve only a single MILP to obtain our rounded initial guess instead of a mixed-integer quadratic program that could take significantly more effort to solve.

### 5.4.2 Steepest-Descent Trust-Region Method for MIPDECO

We generalize our trust-region method from chapter 4 to include additional constraints with Algorithm 2. We decompose solving the MIPDECO into iteratively solving PDEs to obtain the gradients and a binary linear program.

```
Given an initial trust-region radius, $\Delta_0 = \bar{\Delta} \geq 1$, and an initial guess $v^{(0)} \in \{0,1\}^N$
Select a acceptance step parameter $\bar{\rho} > 0$, and set $k \leftarrow 0$
Evaluate the objective function $J_h^{(k)} = J^h(v^{(k)})$ and the gradient $g_h^{(k)} = \nabla_v J^h(v^{(k)})$;
while $\Delta_k \geq 1$ do
    Solve the trust-region subproblem for $\widehat{v}$:
        $\widehat{v} = \underset{v}{\text{minimize}} \quad g_h^{(k)^T}\left(v - v^{(k)}\right) + J_h^{(k)}$
        $\text{subject to} \quad \|v - v^{(k)}\|_1 \leq \Delta_k$
        $\sum_{i=1}^{p} v_n^i \leq 1 \quad \forall n = 1, \ldots, N$
        $m(v) \leq C$
        $v \in \{0,1\}^{pN}$
    Evaluate the objective $J_h(\widehat{v}, u(\widehat{v})$ by solving the state equations with $\widehat{v}$;
    Compute the ratio of actual over predicted reduction: $\rho_k = \frac{J^{(k)} - J(\widehat{v}, u(\widehat{v}))}{-\left(g^{(k)}\right)^T \left(\widehat{v} - v^k\right)}$;
    if $\rho_k > \bar{\rho}$ then
        Accept the step: $v^{(k+1)} = \widehat{v}$ and evaluate the gradient $g^{(k+1)} = \nabla_v J^h(v^{(k+1)})$
        if $\|v^{(k+1)} - v^{(k)}\|_1 = \Delta_k$ then increase the trust-region radius $\Delta_{k+1} = 2\Delta_k$ ;
    else if $\rho_k > 0$ then
        Accept the step $v^{(k+1)} = \widehat{v}$ and evaluate the gradient $g^{(k+1)} = \nabla_v J^h(v^{(k+1)})$
        Keep trust-region radius unchanged $\Delta_{k+1} = \Delta_k$
    else
        Reject the step, set $v^{(k+1)} = v^{(k)}$ and copy the gradient $g^{(k+1)} = g^{(k)}$
        Reduce the trust-region radius $\Delta_{k+1} = \text{floor}\left(\frac{\Delta_k}{2}\right)$;
    Set $k \leftarrow k + 1$;
```

**Algorithm 2:** Steepest-Descent Trust-Region Algorithm.

In every iteration of our trust-region method we begin by solving the discretized state equation given a $v^{(k)}$. Next we build the discretized adjoint system by substituting the approximate state solution $u_{\mathrm{Re}}^h$, $u_{\mathrm{Re}}^h$ and solve the adjoint system for $\lambda_{\mathrm{Re}}^h, \lambda_{\mathrm{Re}}^h$. Once we have the FEM approximation to the state and adjoint equations, we build the gradient (5.15) by applying a quadrature rule. We use this gradient in our trust-region subproblem. The constraints in our application problem are linear, so the trust-region subproblem is still a mixed-integer linear program (MILP).

### 5.4.3 Continuation Algorithm

We introduce a continuation method in Algorithm 3 that we use to attempt to improve on the design of our cloaks. We do this for $s$ different mass upper bounds $C_1, \ldots, C_s$ that are ordered from least to greatest. We introduce a sequence of $l \geq 1$ control meshes $\hat{\Omega}_1, \ldots, \hat{\Omega}_l$, in which the $k$-th control mesh $\hat{\Omega}_k$ corresponds to a $2^{k-1}\tilde{N} \times 2^{k-1}\tilde{N}$ mesh. We choose $\tilde{N}$ with the intention that this size would produce the coarsest of the control meshes to be examined in our continuation method. We note that this ordering implies the following two propositions:

*Proposition* 1. Any feasible solution of (5.9) on the control mesh $\hat{\Omega}_k$ is a feasible solution on $\hat{\Omega}_{k+1}$.

*Proof.* The control mesh $\hat{\Omega}_{k+1}$ is a relaxation of the control mesh $\hat{\Omega}_k$. Therefore a feasible solution for $\hat{\Omega}_k$ is a feasible solution for $\hat{\Omega}_{k+1}$. □

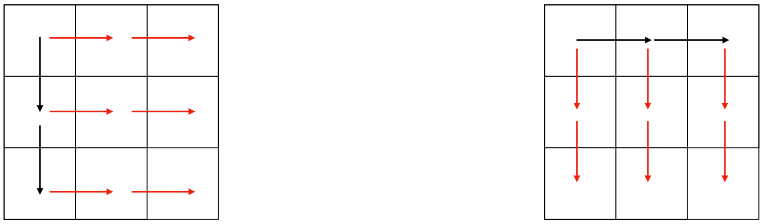
*Proposition* 2. Any feasible solution of (5.9) with mass upper bound $C_l$ is feasible for a mass upper bound $C_{l+1} \geq C_l$.

*Proof.* A feasible solution $v$ satisfies $m(v) \leq C_l \leq C_{l+1}$, therefore $m(v) \leq C_{l+1}$. □

We start the algorithm by first solving the relaxation of (5.9), rounding with (5.16), and executing our trust-region method on the coarsest mesh, with $C = C_1$. We note there are many possible paths for continuation, such as first along the control mesh and then along the mass upper bound, or first along the mass upper bound and then along the control mesh. We illustrate these two continuation paths in Figure 5.2. We choose the following orientation while performing continuation: moving left to right corresponds to refining the control mesh, and moving from top to bottom corresponds to increasing the mass upper bounds. We define a general list $L$:

$$L = ((C_l, \hat{\Omega}_k)) \quad \text{for some } l, k, \tag{5.17}$$

which defines the sequence of problems to be solved. The order of the sequence determines the continuation path taken and any order is acceptable as long as the feasible region of the previous problem is a subset of the feasible region of the current problem. We assume the first problem in the list corresponds to the coarsest mesh and the smallest mass upper bound.



**(a)** Stage 1 continuation on the mass, then in stage 2 continuation on the control mesh. **(b)** Stage 1 continuation on the control mesh, then in stage 2 continuation on the mass.

**Figure 5.2** 5.2a: Continuation method with emphasis on continuation in control mesh size. Stage one of the continuation is represented with black arrows and stage two of the continuation is represented by red lines. 5.2b: Continuation method with emphasis on continuation on the mass upper bound. Stage one is represented with black arrows and stage two is represented with red arrows.

## 5.5 Numerical Experiments

We consider cloaking of an object in the region

$$D_0 = \{x, y \mid -0.6 \le x \le 0.6,\ 0.7 \le y \le 1\} \qquad \text{(Rectangle).} \tag{5.18}$$

We run all our experiments using a MacOS desktop with a four core Intel processor working at 3.3GHz. We use the FEniCS version 2017.2.0 package in Python [5] to solve the state and adjoint PDEs using piecewise linear basis functions. We solve all the PDEs on a 128 × 128 FEM mesh using triangular elements. In addition we use FEniCS's default two-point Gauss quadrature method to build the components of the gradient. To solve the relaxation we use the "trust-constr" solver in SciPy [52], which is a trust-region method for optimization problems with constraints. For this solver, we set the termination condition to be once the two-norm of the gradient is $10^{-4}$. We use CPLEX 12.8.0.0 when solving our trust-region subproblem and when applying our rounding heuristic (5.16) to the relaxed solution of (5.9). In our trust-region method, we use $\bar{\rho} = 0.75$, as recommended by [80] and an initial trust-region radius of $\Delta_0 = 256$. All numerical experiments use the following set of common parameters: wave number $k_0 = 6\pi$ and cloak location, $\widehat{D} \subset D$, where

$$D = \left[-1, 1\right] \times \left[-1, 1\right] \quad \text{and} \quad \widehat{D} = \left[-\frac{5}{8}, \frac{5}{8}\right] \times \left[-\frac{5}{8}, \frac{5}{8}\right]. \tag{5.19}$$

In Table 5.1, we list the material relative permittivity, density, and the color that we use for means of plotting. If we choose to not place material in a region we color the cloak element blue. We let the incidence wave approach the domain at angle $\theta = \frac{\pi}{2}$ for all simulations. For our numerical

```
Given list $L$ of problems, $(C_1, \hat{\Omega}_1) \leftarrow \mathrm{pop}(L)$   // select first element of list
Using the trust-region method 2, solve (5.9) using the mass upper bound and control mesh
  $(C_1, \hat{\Omega}_1)$ to obtain trust-region solution $v_{1,1}$.
while $L$ is not empty do
    $(C_l, \hat{\Omega}_k) \leftarrow \mathrm{pop}(L)$ // select next element of the list
    if $v_{l,k-1}$ solved then
        Initialize the trust-region method starting point to be $v_{l,k-1}$ and solve (5.9) using
          $(C_l, \hat{\Omega}_k)$ to produce trust-region solution $v_{l,k}$. // continuation in the
          control mesh
    else
        Initialize the trust-region method starting point to be $v_{l-1,k}$ and solve (5.9) using
          $(C_l, \hat{\Omega}_k)$ to produce trust-region solution $v_{l,k}$. // continuation in the mass
          upper bound
```

**Algorithm 3:** Continuation Algorithm.

**Table 5.1** Material relative permittivity, density and plotting color.

| $q_i$ | $\rho_i$ | Color |
|---|---|---|
| 0.1875 | 1/6 | Green |
| 0.3750 | 1/2 | Yellow |
| 0.5625 | 2/3 | Black |
| 0.7500 | 1.0 | Red |

experiments we use mass upper bounds $C_1 = 0.30$, $C_2 = 0.40$, and $C_3 = 0.45$, and we examine $10 \times 10$, $20 \times 20$, and $40 \times 40$ control meshes.

We experiment with both a single material and four materials. For the one material case, we use the red material, and for the four material case we use all the materials in Table 5.1. In Section 5.5.1, we establish if it is advantageous to build a cloak out of several materials versus a single material.

### 5.5.1 One Material

We first validate our model by comparing our results with the results from [41]. Next we conduct a computational experiment that solves (5.9) with a single material using the approach by solving the relaxation of (5.9), rounding the relaxation using (5.16), and performing our trust-region method using the rounded relaxation as an initial starting point. We compare the objective values of solving this problem with the mass constraint to our results in chapter 4, which did not constrain the mass of the cloak, but did use the same material. We also report in this table the mass of the cloak. When we examine the objective function values in Table 5.2 to the objective function values in chapter 4, we observe that the objective values in Table 5.2 are an order of magnitude higher. Our conclusion

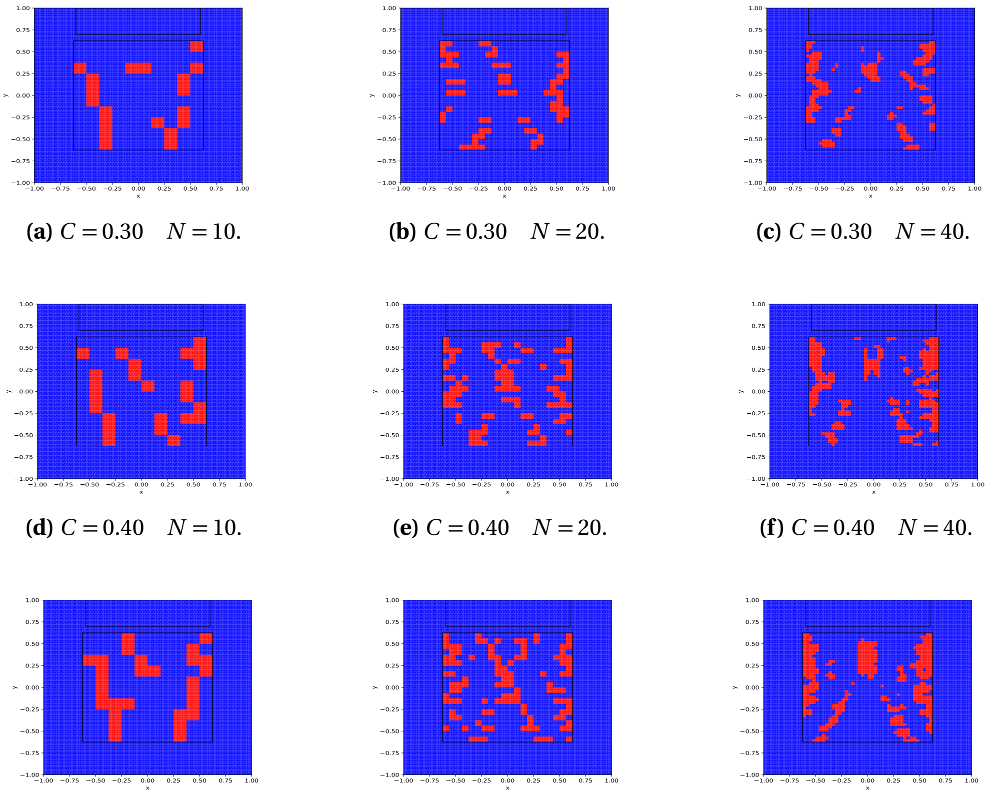
**Table 5.2** The trust-region solution objective function value for a given mass upper bound and control mesh and the mass of the cloak. We solve the problem by solving the relaxation of (5.9) using one material, then start our trust-region method with an initial guess generating by applying the rounding heuristic (5.16).

| $C$ | $10 \times 10$ | | $20 \times 20$ | | $40 \times 40$ | |
|---|---|---|---|---|---|---|
| | Objective | Constraint | Objective | Constraint | Objective | Constraint |
| 0.30 | 0.1001 | 0.2968 | 0.0379 | 0.2968 | 0.0194 | 0.2998 |
| 0.40 | 0.0549 | 0.3750 | 0.0278 | 0.3984 | 0.0080 | 0.3994 |
| 0.45 | 0.0541 | 0.4375 | 0.0175 | 0.4492 | 0.0139 | 0.4404 |

is that the introduction of the mass constraint can negatively affect the cloaking ability of the cloak. We observe that for $C = 0.30$ and $C = 0.40$ the mass constraint is almost active. We make several conclusions from Table 5.2. First, when $C = 0.45$, the mass constraint is not active in the local solutions we find. As we solve the problem on more refined control meshes, we find lower objective function values. This indicates that on more refined control meshes, where there is more choice in the construction of the cloak, we are able to drive the objective function value lower. Second, we find that when the upper bound for the mass is increased, there is an improvement in the cloaking ability. In Figure 5.3, we show the designs produced by the trust-region method and in Figure 5.4 we show the integrand of the objective function value of (5.9) on the computational domain for different mass upper bounds and control meshes. We see that cloaking ability is improved as we refine the control mesh. In Table 5.3 we report the computational effort required to solve the problem. When we compare the computational effort incurred by the introduction of the mass constraint with our results in chapter 4, we do not see a significant increase and the trust-region iterations are comparable. This indicates that our trust-region method for MIPDECO is still computationally tractable with linear constraints.

**Table 5.3** CPU times in seconds when solving from scratch with one material including the PDE/gradient calculations and knapsack problems on a $128 \times 128$ FEM mesh where $\theta = \frac{\pi}{2}$.

| | | Solution CPU Time(s) | | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|---|
| $C$ | Control | Relaxation | PDEs | Gradient | MIPs | Iterations | Figure |
| 0.30 | $10 \times 10$ | 707.64 | 40.13 | 26.03 | 0.23 | 11 | 5.3a,5.4a |
| 0.30 | $20 \times 20$ | 12432.87 | 74.45 | 172.13 | 1.62 | 21 | 5.3b,5.4b |
| 0.30 | $40 \times 40$ | 3750.88 | 231.97 | 2078.87 | 14.24 | 66 | 5.3c,5.4c |
| 0.40 | $10 \times 10$ | 564.90 | 47.77 | 31.15 | 0.29 | 13 | 5.3d,5.4d |
| 0.40 | $20 \times 20$ | 10072.67 | 70.88 | 163.72 | 1.23 | 20 | 5.3e,5.4e |
| 0.40 | $40 \times 40$ | 82182.54 | 223.39 | 1999.44 | 12.40 | 64 | 5.3f,5.4f |
| 0.45 | $10 \times 10$ | 393.83 | 43.21 | 28.79 | 0.27 | 12 | 5.3h,5.4h |
| 0.45 | $20 \times 20$ | 10791.47 | 102.02 | 234.94 | 1.704 | 29 | 5.3g,5.4g |
| 0.45 | $40 \times 40$ | 67686.60 | 170.82 | 1563.98 | 8.44 | 49 | 5.3i,5.4i |

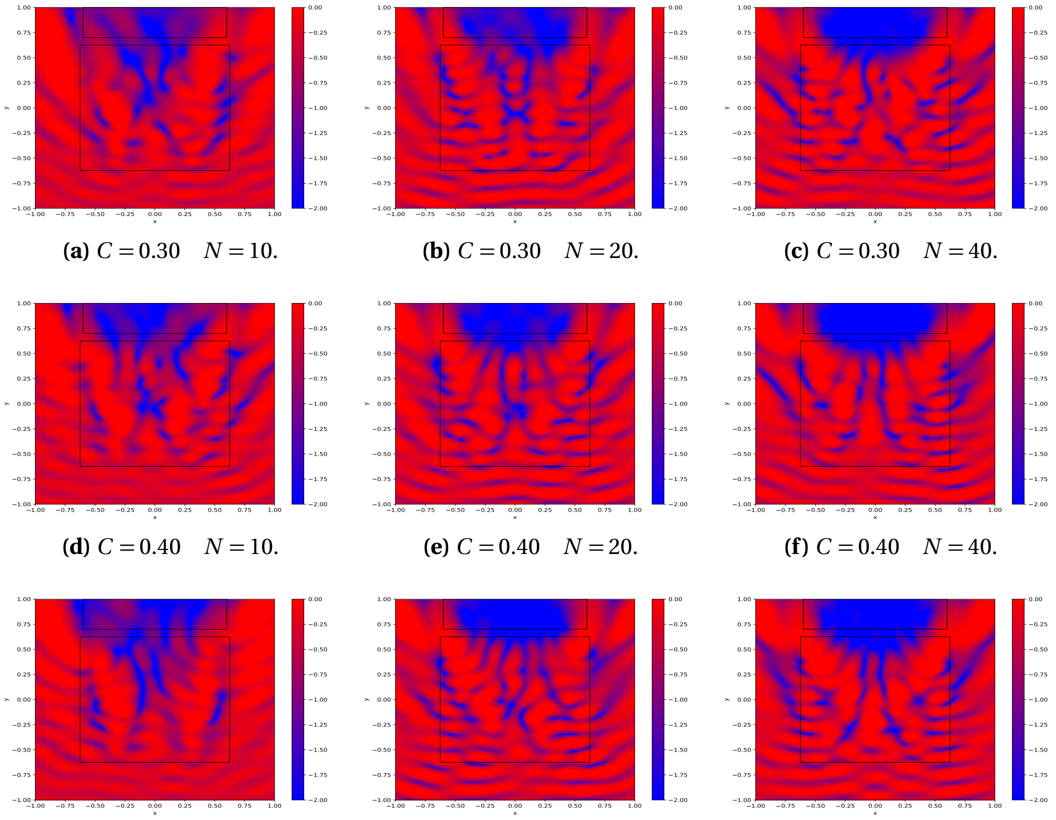


**(a)** $C = 0.30 \quad N = 10.$ **(b)** $C = 0.30 \quad N = 20.$ **(c)** $C = 0.30 \quad N = 40.$

**(d)** $C = 0.40 \quad N = 10.$ **(e)** $C = 0.40 \quad N = 20.$ **(f)** $C = 0.40 \quad N = 40.$

**(g)** $C = 0.45 \quad N = 10.$ **(h)** $C = 0.45 \quad N = 20.$ **(i)** $C = 0.45 \quad N = 40.$

**Figure 5.3** Cloak designs obtained when solving from scratch with one material.

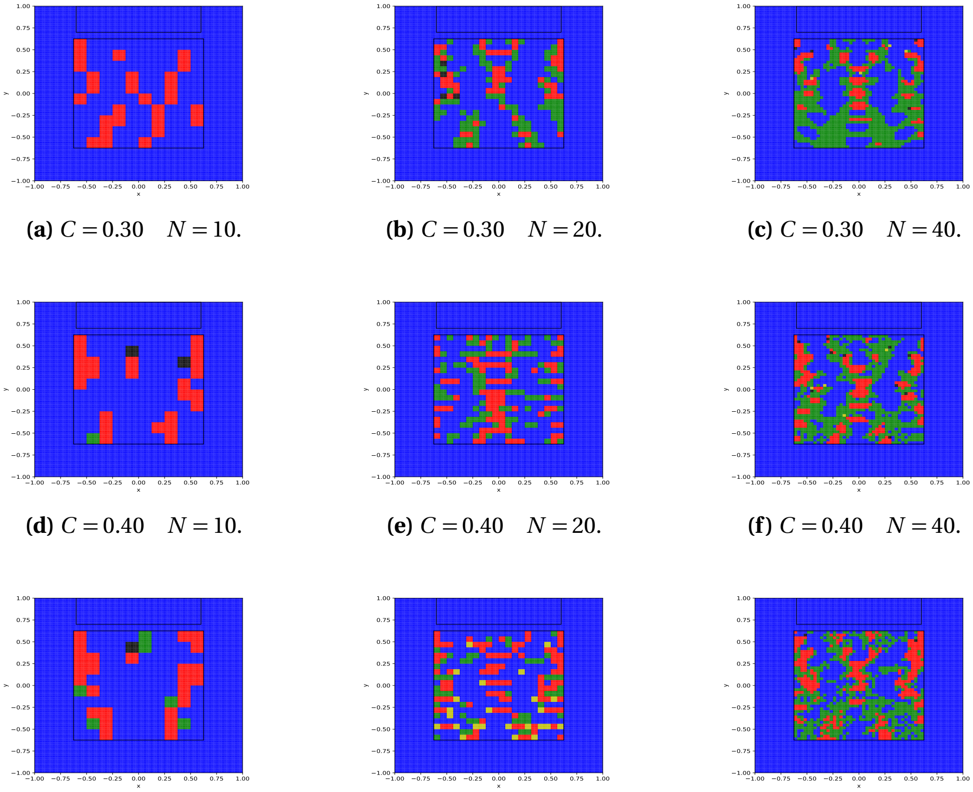


**(a)** $C = 0.30 \quad N = 10.$ **(b)** $C = 0.30 \quad N = 20.$ **(c)** $C = 0.30 \quad N = 40.$

**(d)** $C = 0.40 \quad N = 10.$ **(e)** $C = 0.40 \quad N = 20.$ **(f)** $C = 0.40 \quad N = 40.$

**(g)** $C = 0.45 \quad N = 10.$ **(h)** $C = 0.45 \quad N = 20.$ **(i)** $C = 0.45 \quad N = 40.$

**Figure 5.4** Scattering field in $\log_{10}$ obtained when solving from scratch with one material.



**(a)** $C = 0.30 \quad N = 10.$ **(b)** $C = 0.30 \quad N = 20.$ **(c)** $C = 0.30 \quad N = 40.$

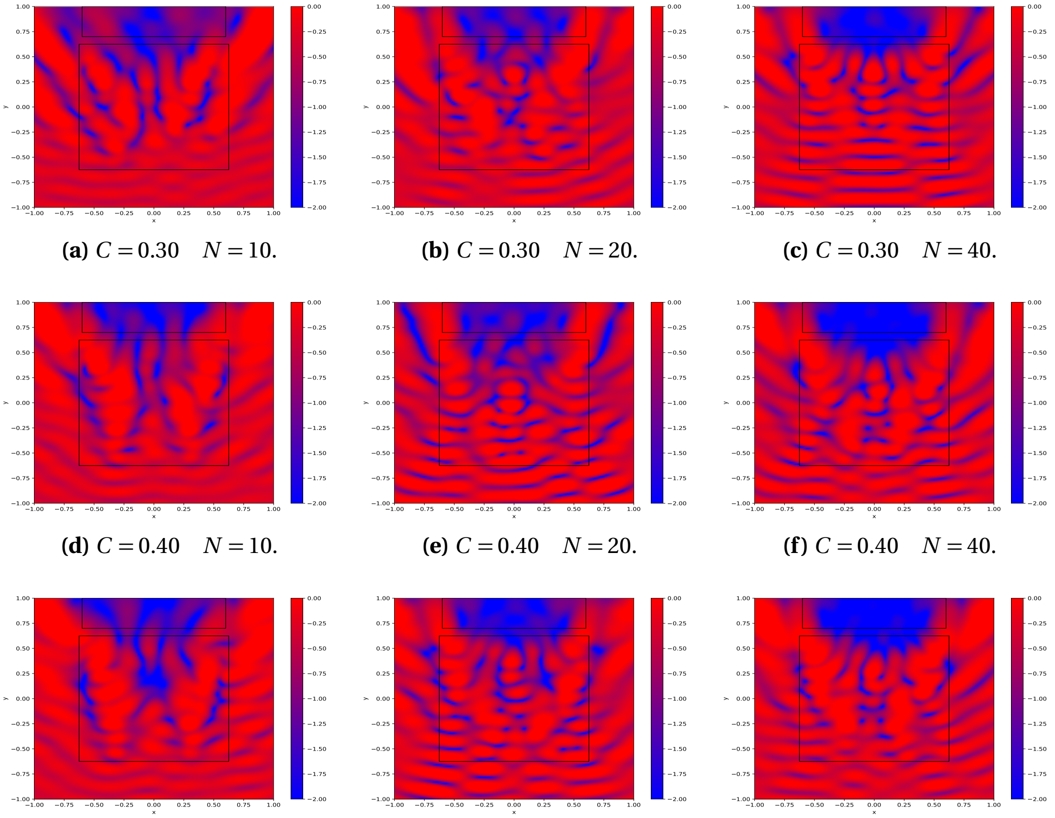
**(d)** $C = 0.40 \quad N = 10.$ **(e)** $C = 0.40 \quad N = 20.$ **(f)** $C = 0.40 \quad N = 40.$

**(g)** $C = 0.45 \quad N = 10.$ **(h)** $C = 0.45 \quad N = 20.$ **(i)** $C = 0.45 \quad N = 40.$

**Figure 5.5** Cloak designs obtained when solving from scratch with four materials.

### 5.5.2 Four Materials

In our next experiment, we examine solving (5.9) with four materials and compare the results with the previous experiment. We compare Table 5.5, which reports the objective function value and mass of the final cloak for four materials, to Table 5.3 that reports the same for one material. We see that the objective function value for the four material case is slightly worse than the one material case, but is on the same order of magnitude. This indicates that it is not necessarily advantageous to build a cloak out of several materials, at least from the approach of solving the relaxation, rounding, and applying the trust-region method. In Figures 5.5 and 5.6, we report the designs and scattering field associated with using four materials. Our results follow a similar trend with our observations and conclusions for one material, mainly that cloaking ability improves as we either refine the control mesh or allow for a larger mass upper bound. The computational time for solving the problem using different mass upper bounds and control meshes is reported in Table 5.5. Because of the way we



**(a)** $C = 0.30 \quad N = 10$. **(b)** $C = 0.30 \quad N = 20$. **(c)** $C = 0.30 \quad N = 40$.

**(d)** $C = 0.40 \quad N = 10$. **(e)** $C = 0.40 \quad N = 20$. **(f)** $C = 0.40 \quad N = 40$.

**(g)** $C = 0.45 \quad N = 10$. **(h)** $C = 0.45 \quad N = 20$. **(i)** $C = 0.45 \quad N = 40$.

**Figure 5.6** Scattering field in $\log_{10}$ obtained when solving from scratch with four materials.

**Table 5.4** The trust-region solution objective function value for a given mass upper bound and control mesh and the mass of the cloak. We solve the problem by solving the relaxation of (5.9) using one material, then start our trust-region method with an initial guess generating by applying the rounding heuristic (5.16).

| $C$ | $10 \times 10$ | | $20 \times 20$ | | $40 \times 40$ | |
|---|---|---|---|---|---|---|
| | Objective | Constraint | Objective | Constraint | Objective | Constraint |
| 0.30 | 0.0981 | 0.2968 | 0.0606 | 0.2994 | 0.0334 | 0.2996 |
| 0.40 | 0.0608 | 0.3984 | 0.0558 | 0.3951 | 0.0162 | 0.3995 |
| 0.45 | 0.0599 | 0.4166 | 0.0312 | 0.4498 | 0.0153 | 0.4497 |

**Table 5.5** CPU times in seconds when solving from scratch with four materials including the PDE/gradient calculations and knapsack problems on a 128 × 128 FEM mesh where $\theta = \frac{\pi}{2}$.

| | | Solution CPU Time(s) | | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|---|
| $C$ | Control | Relaxation | PDEs | Gradient | MIPs | Iterations | Figure |
| 0.30 | $10 \times 10$ | 721.05 | 45.72 | 104.57 | 1.07 | 12 | 5.5a,5.6a |
| 0.30 | $20 \times 20$ | 3196.71 | 358.80 | 2990.04 | 19.43 | 96 | 5.5b,5.6b |
| 0.30 | $40 \times 40$ | 112647.39 | 377.80 | 12528.02 | 84.12 | 100 | 5.5c,5.6c |
| 0.40 | $10 \times 10$ | 3260.25885 | 47.66 | 107.44 | 1.35 | 13 | 5.5d,5.6d |
| 0.40 | $20 \times 20$ | 3124.09 | 76.65 | 727.46 | 6.81 | 20 | 5.5e,5.6e |
| 0.40 | $40 \times 40$ | 117127.11 | 360.20 | 12486.30 | 88.30 | 100 | 5.5f,5.6f |
| 0.45 | $10 \times 10$ | 667.87 | 45.92 | 108.52 | 0.48 | 12 | 5.5g,5.6g |
| 0.45 | $20 \times 20$ | 3892.93 | 114.48 | 972.96 | 5.39 | 31 | 5.5h,5.6h |
| 0.45 | $40 \times 40$ | 107674.43 | 206.25 | 7029.62 | 41.39 | 56 | 5.5i,5.6i |

have introduced the MIPDECO (5.9), the four-material problem has four times as many variables as its one-material counterpart. When we take this into account, we see that the computational effort required to construct the gradient for the trust-region method for the four material problem on a given run is about four times greater than its one material counterpart. The computational effort of building the gradient for the multi-material problem is just a multiple of building the gradient for the one material problem. The multiplication factor that determines the effort is the number of materials. This experiment shows that the designs found for the refined control meshes are cloaks made out of mostly red and green material on the most refined control meshes. It is not surprising that this is the case, because this green material is the least massive material and has the smallest relative permittivity. When compared to the one material case, we replace more massive red material with less massive green material to create a cloak. Even though the cloaking ability is slightly worse, one can imagine the advantage of such a cloak.

### 5.5.3 Continuation

Solving the problem from scratch with four materials in some instances of $C$ and control mesh size produced cloaks whose cloaking ability were slightly worst than their one material counterparts. This makes multi-material cloaking look less attractive. We address this concern with our continuation approach and show that the continuation produces superior cloaks. One benefit of the continuation methods is that we only solve one problem from scratch, which means only one relaxation must be solved.

In Figure 5.7, we provide the upper bound on the mass and the control meshes we use in our continuation method. In Figure 5.9, we present the designs from performing the continuation on the control mesh, and in Figure 5.10, we present the corresponding scattering effect. We observe

| | | |
|---|---|---|
| $(0.30, 10 \times 10)$ | $(0.30, 20 \times 20)$ | $(0.30, 40 \times 40)$ |
| $(0.40, 10 \times 10)$ | $(0.40, 20 \times 20)$ | $(0.40, 40 \times 40)$ |
| $(0.45, 10 \times 10)$ | $(0.45, 20 \times 20)$ | $(0.45, 40 \times 40)$ |

**Figure 5.7** Parameters for the continuation method. The first parameter is the value of $C$ and the second parameter is the control mesh size.

the same behavior as in our previous experiments. The cloaking ability is enhanced as either we refine the control mesh, or lessen the restriction on the mass upper bound and refine the mesh simultaneously. Even though we can choose from several materials, our continuation solutions favor the red material. We note that the objective function values we find from our continuation scheme in Table 5.6 are better than the objective function values obtained from solving each instance of the problem from scratch with four materials in Table 5.4 or one material in Table 5.2. Moreover, when we compare the objective function values in Table 5.7 with our previous results in chapter 4, the objective functions values are on the same order of magnitude or better. Even with a mass constraint, we can build a cloak out of four materials that is competitive with a cloak made out of one material without a mass constraint. We also observe that when we look at the computational effort of performing the continuation in Table 5.7, the average trust-region iterations required are lower than solving the problem from scratch.

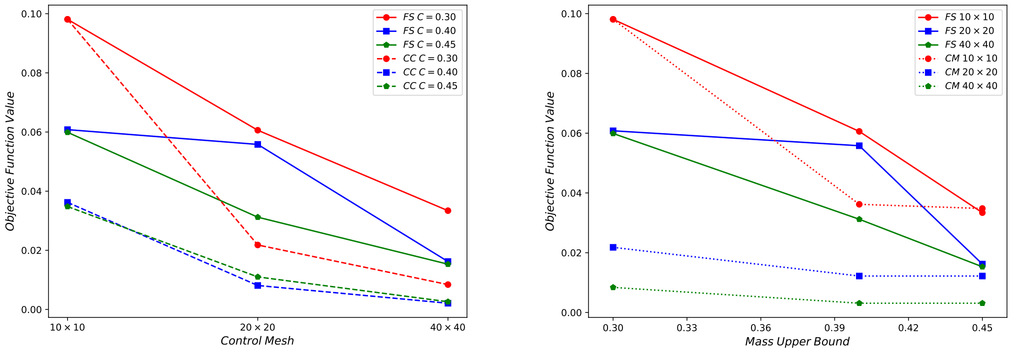
Next we present the designs in Figure 5.11 and the scattering fields in Figure 5.12 for continuation on the mass upper bound. We observe that the refinement of the control mesh is helpful at reducing the objective function value and therefore creating better cloaking ability. The cloaks obtained from the continuation method still favor red material. The computational effort required for performing this continuation in Table 5.9 is comparable.

We compare the effectiveness between solving the four material problem from scratch, using control mesh continuation, and using cloak mass continuation in Figure 5.8. Between the control mesh continuation and the cloak mass continuation, they are comparable in the quality of cloaks

**Table 5.6** Objective function value and cloak mass for control mesh continuation.

| $C$ | $10 \times 10$ | | $20 \times 20$ | | $40 \times 40$ | |
|---|---|---|---|---|---|---|
| | Objective | Constraint | Objective | Constraint | Objective | Constraint |
| 0.30 | 0.0981 | 0.2968 | 0.0218 | 0.2994 | 0.0084 | 0.2999 |
| 0.40 | 0.0362 | 0.3906 | 0.0081 | 0.3984 | 0.0021 | 0.3979 |
| 0.45 | 0.0348 | 0.4218 | 0.0110 | 0.4335 | 0.0026 | 0.4335 |

produces. The objective function values of the two continuation schemes are on the same order of magnitude. The control mesh continuation produces slightly lower objective values on the more refined control meshes, $20 \times 20$ and $40 \times 40$. This makes a strong case that control mesh refinement plays a powerful role at enhancing the cloaking ability of a cloak. In both cases, the objective function values from conducting continuation in either Tables 5.6 or 5.8 are better than when solving from scratch in Tables 5.2 and 5.4, and comparable to the objective function values found in chapter 4.



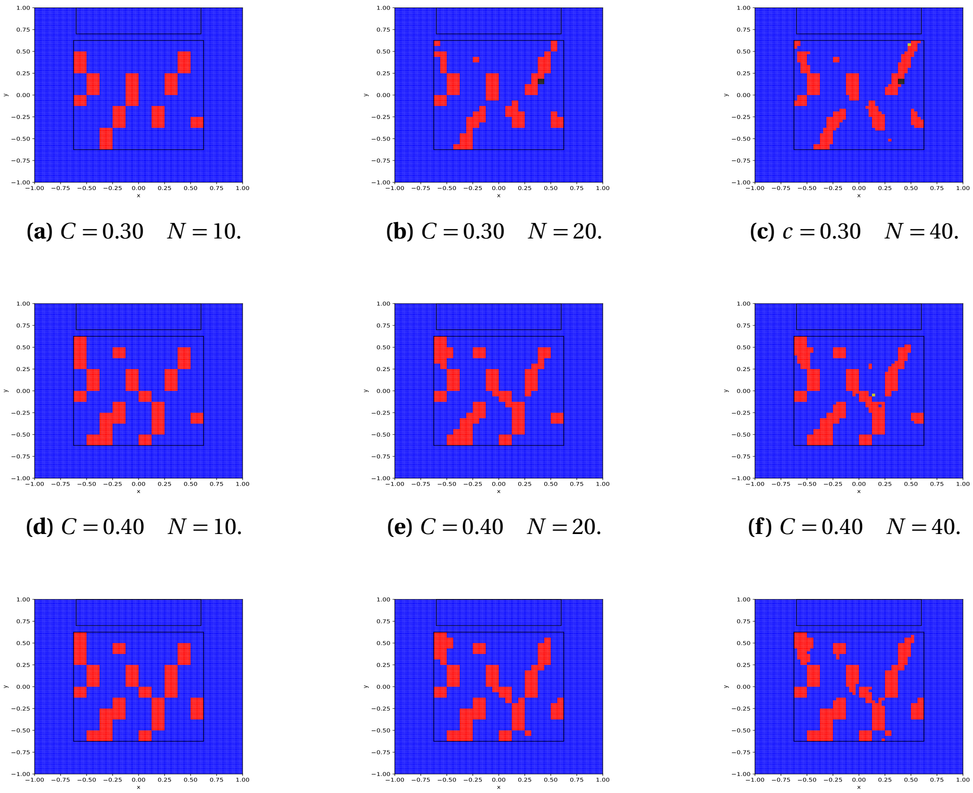
**(a)** Control mesh continuation.

**(b)** Mass continuation.

**Figure 5.8** 5.8a: Comparison between the control mesh continuation and solving the problem from scratch with several different mass upper bounds. We use FS to denote the from scratch experiments and CC to denote the control mesh continuation experiments. 5.8b: Comparison between the mass continuation and solving the problem from scratch for several different control meshes. We use CM to denote mass continuation experiments.

## 5.6 Conclusion

In this chapter we introduced a MIPDECO formulation for designing electromagnetic cloaks using multiple materials, while also enforcing a constraint on the mass of the cloak. We introduced a



**(a)** $C = 0.30 \quad N = 10.$

**(b)** $C = 0.30 \quad N = 20.$

**(c)** $c = 0.30 \quad N = 40.$

**(d)** $C = 0.40 \quad N = 10.$

**(e)** $C = 0.40 \quad N = 20.$

**(f)** $C = 0.40 \quad N = 40.$

**(g)** $C = 0.45 \quad N = 10.$

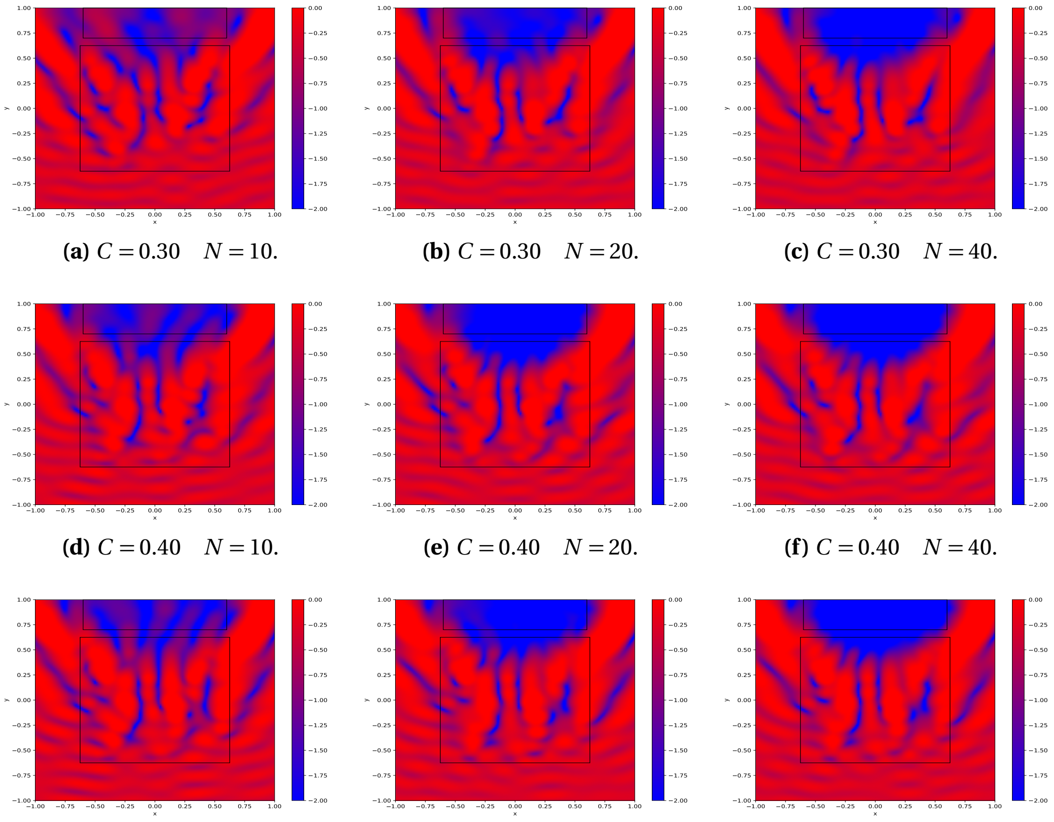
**(h)** $C = 0.45 \quad N = 20.$

**(i)** $C = 0.45 \quad N = 40.$

**Figure 5.9** Cloak designs obtained with control mesh continuation.

**Table 5.7** CPU times in seconds from control mesh continuation including the PDE/gradient calculations and knapsack problems on a $128 \times 128$ FEM mesh where $\theta = \frac{\pi}{2}$.

| | | Solution CPU Time(s) | | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|---|
| $C$ | Control | Relaxation | PDEs | Gradient | MIPs | Iterations | Figure |
| 0.30 | $10 \times 10$ | 288.29 | 88.36 | 199.70 | 1.55 | 23 | 5.9a,5.10a |
| 0.30 | $20 \times 20$ | - | 69.70 | 651.68 | 6.78 | 19 | 5.9b,5.10b |
| 0.30 | $40 \times 40$ | - | 88.79 | 3188.87 | 20.60 | 24 | 5.9c,5.10c |
| 0.40 | $10 \times 10$ | - | 65.54 | 155.88 | 1.40 | 17 | 5.9d,5.10d |
| 0.40 | $20 \times 20$ | - | 55.97 | 515.63 | 2.98 | 22 | 5.9e,5.10e |
| 0.40 | $40 \times 40$ | - | 70.34 | 2481.67 | 12.78 | 25 | 5.9f,5.10f |
| 0.45 | $10 \times 10$ | - | 33.09 | 78.31 | 0.34 | 10 | 5.9g,5.10g |
| 0.45 | $20 \times 20$ | - | 54.15 | 473.93 | 2.12 | 15 | 5.9h,5.10h |
| 0.45 | $40 \times 40$ | - | 70.45 | 2405.10 | 12.09 | 19 | 5.9i,5.10i |



**(a)** $C = 0.30 \quad N = 10.$

**(b)** $C = 0.30 \quad N = 20.$

**(c)** $C = 0.30 \quad N = 40.$

**(d)** $C = 0.40 \quad N = 10.$

**(e)** $C = 0.40 \quad N = 20.$

**(f)** $C = 0.40 \quad N = 40.$

**(g)** $C = 0.45 \quad N = 10.$

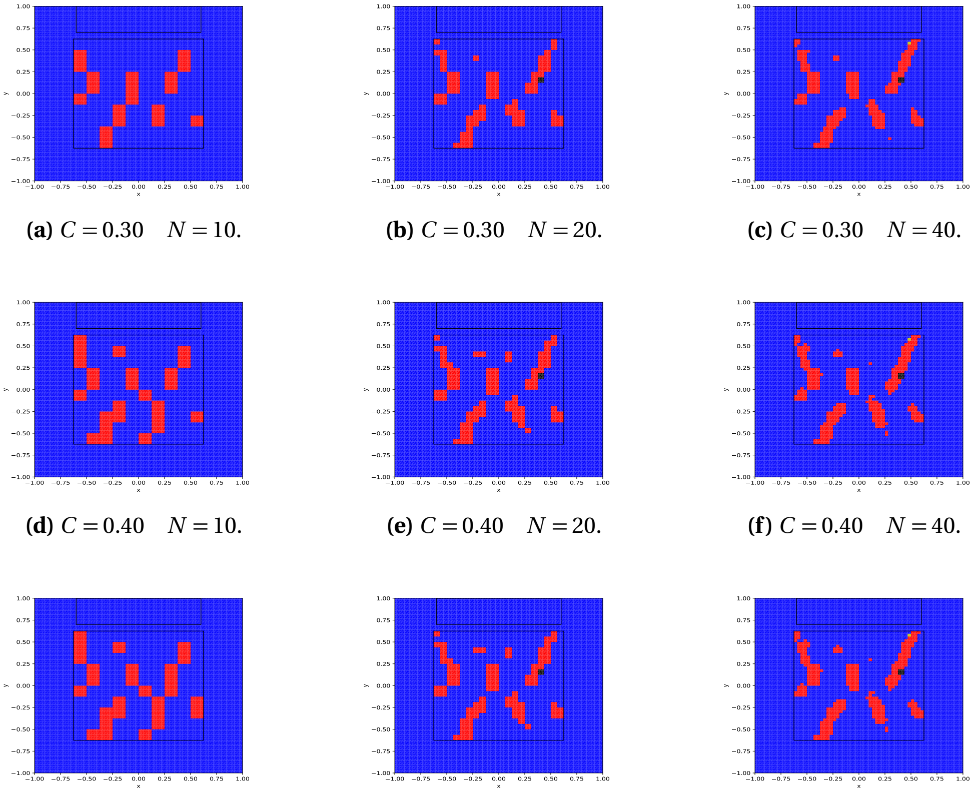
**(h)** $C = 0.45 \quad N = 20.$

**(i)** $C = 0.45 \quad N = 40.$

**Figure 5.10** Scattering field in $\log_{10}$ with control mesh continuation.

**Table 5.8** Objective function value and cloak mass for mass continuation.

| $C$ | $10 \times 10$ | | $20 \times 20$ | | $40 \times 40$ | |
|---|---|---|---|---|---|---|
| | Objective | Constraint | Objective | Constraint | Objective | Constraint |
| 0.30 | 0.0981 | 0.2968 | 0.0218 | 0.2994 | 0.0084 | 0.2999 |
| 0.40 | 0.0362 | 0.3906 | 0.0122 | 0.3307 | 0.0031 | 0.3273 |
| 0.45 | 0.0348 | 0.4218 | 0.0122 | 0.3307 | 0.0031 | 0.3273 |



**(a)** $C = 0.30 \quad N = 10.$ **(b)** $C = 0.30 \quad N = 20.$ **(c)** $C = 0.30 \quad N = 40.$

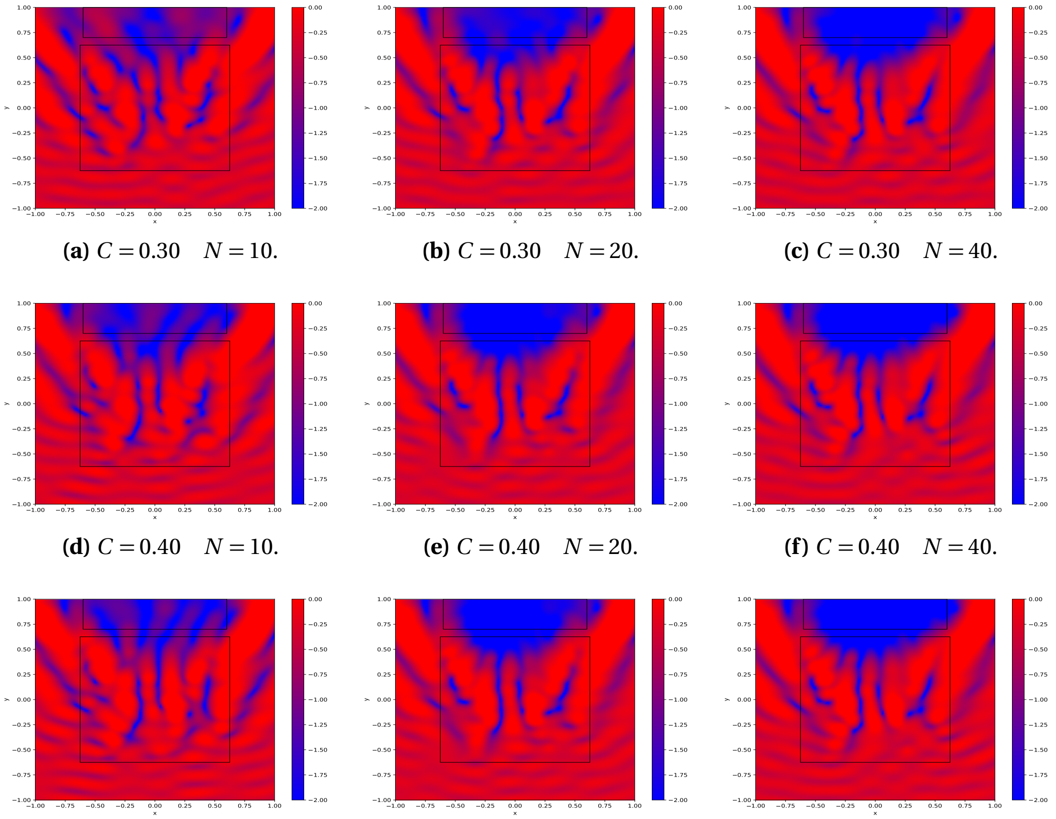
**(d)** $C = 0.40 \quad N = 10.$ **(e)** $C = 0.40 \quad N = 20.$ **(f)** $C = 0.40 \quad N = 40.$

**(g)** $C = 0.45 \quad N = 10.$ **(h)** $C = 0.45 \quad N = 20.$ **(i)** $C = 0.45 \quad N = 40.$

**Figure 5.11** Cloak designs obtained with mass continuation.

**Table 5.9** CPU times in seconds from mass continuation including the PDE/gradient calculations and knapsack problems on a 128 × 128 FEM mesh where $\theta = \frac{\pi}{2}$.

| | | Solution CPU Time(s) | | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|---|
| $C$ | Control | Relaxation | PDEs | Gradient | MIPs | Iterations | Figure |
| 0.3 | 10 × 10 | 272.46 | 87.47 | 206.15 | 1.81 | 23 | 5.11a,5.12a |
| 0.3 | 20 × 20 | - | 69.20 | 659.69 | 6.72 | 19 | 5.11b,5.12b |
| 0.3 | 40 × 40 | - | 89.04 | 3333.24 | 19.53 | 24 | 5.11c,5.12c |
| 0.40 | 10 × 10 | - | 68.54 | 150.36 | 1.57 | 18 | 5.11d,5.12d |
| 0.40 | 20 × 20 | - | 37.65 | 361.92 | 1.51 | 10 | 5.11e,5.12e |
| 0.40 | 40 × 40 | - | 70.30 | 2742.36 | 13.61 | 19 | 5.11f,5.12f |
| 0.45 | 10 × 10 | - | 32.74 | 76.38 | 0.46 | 9 | 5.11g,5.12g |
| 0.45 | 20 × 20 | - | 33.54 | 339.41 | 1.00 | 9 | 5.11h,5.12h |
| 0.45 | 40 × 40 | - | 33.59 | 1295.91 | 5.98 | 9 | 5.11i,5.12i |



**(a)** $C = 0.30 \quad N = 10.$

**(b)** $C = 0.30 \quad N = 20.$

**(c)** $C = 0.30 \quad N = 40.$

**(d)** $C = 0.40 \quad N = 10.$

**(e)** $C = 0.40 \quad N = 20.$

**(f)** $C = 0.40 \quad N = 40.$

**(g)** $C = 0.45 \quad N = 10.$

**(h)** $C = 0.45 \quad N = 20.$

**(i)** $C = 0.45 \quad N = 40.$

**Figure 5.12** Scattering field in $\log_{10}$ with mass continuation.

trust-region method to solve this problem. The trust-region method was shown in our numerical experiments to be computationally tractable, even with the mass constraint. Instead of solving the our problem with the branch-and-bound method, which could be computationally intractable in practice [34, 102], our method only requires the solutions of PDEs and mixed-integer linear programming problems. Our numerical experiments demonstrated the effect the mass constraint and the control mesh refinement had on the effectiveness of the electromagnetic cloak. We showed that solving the MIPDECO problem from scratch is not as favorable as applying a continuation method. The continuation method we introduced was successful at producing superior cloaks when compared to solving our problem from scratch in each instance.

CHAPTER

# 6

# A MIXED-INTEGER PDE-CONSTRAINED OPTIMIZATION FORMULATION FOR THE CONSTRUCTION OF ELECTROMAGNETIC CLOAKS WITH A SINGLE MATERIAL IN 3D

In this chapter we introduce a model for binary electromagnetic cloaking in 3D. This new model is a direct generalization of the 2D cloaking problem with a single material. The problem itself naturally introduces a large number of integer variables. We study how to build a $20 \times 20 \times 20$ cube cloak. We show that even with 8000 integer variables present in the problem, our trust-region method is successful in producing quality local solutions. Furthermore we show that like in the 2D case, the bottleneck of MIPDECO from our perspective is not the number of integer variables present in the problem. In fact, solving the integer portion of the MIPDECO during the execution of our trust-region method adds negligible time compared to the time taken to solve the PDEs and building the gradient.

## 6.1 Electromagnetic Cloaking Design for a Single Material in 3D

In this section we study the binary electromagnetic cloaking MIPDECO in 3D. The nature of the 3D problem introduces a large amount of integer variables. The purpose of this section is to illustrate that our trust-region method for MIPDECO is computationally tractable when a large number of integer variables are present. We show this with numerical experiments.

The remainder of the section will proceeds as follows: we first pose 3D formulation based on the 2D formulation presented in Section 4.2. We then present numerical experiments that demonstrate the success of our trust-region method when a large amount of integer variables are present. We note that there are many similarities between solving the 2D problem and 3D. Because of this fact we omit repetitive details that are shared between the two sections.

### 6.1.1 A Formulation for 3D Binary Electromagnetic Cloaking

We first revisit our model based on [41] and presented in Section 4.2 for a single angle of attack. We let $D \subset \mathbb{R}^3$ be the computational domain and $D_0 \subset D$ be the part of the domain that we wish to cloak. We let $\widehat{D} \subset D - D_0$ be the subregion in which we can design the cloak and $\hat{\Omega} \subset \widehat{D}$ the cloak. The only significant difference in the 2D and the 3D model is that the incident wave can now travel is a three dimensional space instead of a two dimension space. We define the traveling (incident) wave to be $\hat{u}_{\psi,\theta}$ and define the 3D MIPDECO (with scaling) as

$$\begin{array}{lll} \underset{u,v,w}{\text{minimize}} & \mathscr{J}(u,v,w) = \dfrac{1}{2|D_0|}\|u - \hat{u}_{\psi,\theta}(\psi,\theta)\|_{2,D_0}^2 & \\ \text{subject to} & -\Delta u - k_0^2(1+q\,w)u = k_0^2 q\,w\,\hat{u}_{\psi,\theta} & \text{in } D \\ & \dfrac{\partial u}{\partial n} - i k_0 u = 0 & \text{on } \partial D \\ & w = \displaystyle\sum_{n=1}^{N} v_n \mathbb{1}_{\hat{\Omega}_n \setminus \partial\hat{\Omega}_n} & \\ & v_n = \{0,1\} & \forall n = 1,\ldots,N, \end{array} \tag{6.1}$$

where

$$\hat{u}_{\psi,\theta}(\psi,\theta) = \exp(i k_0(x\sin(\psi)\cos(\theta) + y\sin(\psi)\sin(\theta) + z\cos(\psi))) \tag{6.2}$$

is the traveling wave with respect to which we are cloaking with normalized wave direction $(\sin(\psi)\cos(\theta), \sin(\psi)\sin(\theta), \cos(\psi))$, and wave number $k_0$. We note that we will enforce the notation for the spherical representation of the direction at which the wave travels. The quantity $\psi$ measures the angle between the positive z axis and $\theta$ measures the angle in the $(x, y)$ plane from the positive x axis. The quantity $q$ is the electromagnetic permittivity between and the cloaked device region and the outside the cloak device region. The state variable $u$ represents the electromagnetic field intensity and $\mathbb{1}$ is the indicator function of the cloak. We introduce the scaling factor $\frac{1}{|D_0|}$, as seen in the multiple material problem to ensure that the objective function cannot be made arbitrarily small.

In this formulation, we generalize how we conceptualize $\hat{\Omega}$, $\hat{\Omega}_n$ and $D_0$. In the 2D problem, $\hat{\Omega}$ was a square subdomain. In this formulation $\hat{\Omega}$ is a cube. In the 2D formulation, $\hat{\Omega}_n$ were squares, in this formulation they are now cubes. We generalize the cloaking domains we used for the 2D formulation, which were a circle, square and rectangle, to become a sphere, cube, and prism respectively.

We note that the derivations and analysis conducted in 4.2 is still valid in this current formulation. The state, adjoint and gradient equation in 4.2.3 still hold for this problem. The only difference is that the integrals are in $\mathbb{R}^3$ instead of $\mathbb{R}^2$.

### 6.1.2 Numerical Experiments

In this section we describe our numerical experiments. We apply our trust-region method (Algorithm 1) to the 3D problem.

#### 6.1.2.1 Experimental Setup

We consider the cloaking of three different domains $\bar{D}_0^1$, $\bar{D}_0^2$, and $\bar{D}_0^3$ given by

$$\begin{aligned}
\bar{D}_0^1 &= \{x, y, z \mid -0.6 \leq x \leq 0.6 \; 0.7 \leq y \leq 1 \; 0.7 \leq z \leq 1\} && \text{(Rectangular Prism)}\\
\bar{D}_0^2 &= \{x, y, z \mid 0.7 \leq x \leq 1 \; 0.7 \leq y \leq 1 \; 0.7 \leq z \leq 1\} && \text{(Cube)}\\
\bar{D}_0^3 &= \{x, y, z \mid (x-0.85)^2 + (y-0.85)^2 + (z-0.85)^2 \leq (.1)^2\} && \text{(Sphere).}
\end{aligned} \tag{6.3}$$

We note that we use the same computational resources to solve this problem as we used in 4.2.6. All numerical experiments use the following set of common parameters: wave number $k_0 = 6\pi$, material constant $q = 0.75$, spectral angles $\theta = \frac{\pi}{4}$ and $\psi = \frac{\pi}{4}$, and cloak location $\widehat{D} \subset D$, where

$$D = \left[-1, 1\right] \times \left[-1, 1\right] \times \left[-1, 1\right] \quad \text{and} \quad \widehat{D} = \left[-\frac{5}{8}, \frac{5}{8}\right] \times \left[-\frac{5}{8}, \frac{5}{8}\right] \times \left[-\frac{5}{8}, \frac{5}{8}\right]. \tag{6.4}$$

We solve the state and adjoint equation on a 32 × 32 × 32 FEM mesh and will limit ourselves to a 20 × 20 × 20 cloak; this instance of a cloaking mesh admits a MIPDECO that has 8000 integer variables respectively. The goal of our experiment is to show that our trust-region method can solve problems with a large number of integer variables. The focus of this experiment is not to solve the state to high accuracy. Because this is an elliptic problem, the FEM matrices (for both the state and adjoint operator) will become progressively more ill-conditioned as we refine the mesh. The difficulties associated with solving the Helmholtz equation numerically as the space dimension increases is well known. A summary of these difficulties, and approaches to address them is discussed in [27]. Building a preconditioner (which will be dependent on the binary variable $v$) for both the relaxation and the electromagnetic cloaking MIPDECO is nontrivial exercise and is beyond the scope of this thesis.

In Table 6.1 we show the computational effort required to execute our trust-region algorithm. We provide solution Figures for each run in the appendix. The solution figures can be interpreted as “x-y slices” of the cloak. Level zero is the bottom of the cloak and we move to the top of the cloak which is denoted as level twenty. We note that considerable computational effort is required to solve the relaxation. To alleviate this cost, a feasible starting guess may be used, however in computational experiments we have observed that we obtain better objective function values using the rounded relaxation as a starting guess for our trust-region method than using a randomly generated feasible starting guess. We also report the computational effort to preform the three components of our trust-region method: solve the state and adjoint equations, build the gradients and solve a knapsack problems. We observe that that for our experiments the computational effort is dominated by building the gradient. In this instance, because we used a coarse mesh ($32 \times 32 \times 32$) the cost to solve the state and adjoint is negligible compared to gradient computation. In real application problems, where the mesh would be expected to be at least $128 \times 128 \times 128$, we would see solving the state and adjoint equation would take the most amount of time. The amount of time to solve the knapsack problem, denoted “MIPs”, takes negligible time in the overall computational expense of executing the trust-region method. Even with many more integer variables present in the 3D MIPDECO versus the 2D MIPDECO, we see a similar amount of trust-region steps required to locate a local minimum. As we observed in the 2D problem, the computational effort for solving MIPDECO is not hindered by the presence of a large amount of integer variables, which is an extremely encouraging observation; because this means that our approach scales well with the number of integer variables present in the problem. We do not observe an exponential like time increase when comparing the effort required to solve the 2D single material problem versus the 3D single material problem.

**Table 6.1** CPU times(s) for 3D relaxation, PDE/gradient calculation and knapsack problem on a $32 \times 32 \times 32$ FEM mesh where $\theta = \frac{\pi}{4}$, $\psi = \frac{\pi}{4}$.

| Problem Instance | | Solution CPU Time(s) | | | | Trust-Region | Solution |
|---|---|---|---|---|---|---|---|
| Domain | Control | Relaxation | PDEs | Gradient | MIPs | Iterations | Figure |
| $\bar{D}_0^1$ | $20 \times 20 \times 20$ | 91569.07 | 2129.87 | 34022.25 | 1.54 | 31 | .2.1 |
| $\bar{D}_0^2$ | $20 \times 20 \times 20$ | 51535.83 | 3223.75 | 51119.53 | 3.28 | 44 | .2.2 |
| $\bar{D}_0^3$ | $20 \times 20 \times 20$ | 22122.63 | 1781.35 | 28734.25 | 1.36 | 26 | .2.3 |

In Table 6.2 we report the objective function values for the relaxed solution, the rounded solution and the trust-region solution. In Figure 6.1, 6.2, 6.3 we show the progress of our trust-region method for the three different cloaking regions. We plot the trust-region radius and objective function value as a function of iteration count. We observe that our trust-region method is successful at significantly reducing the objective function value in a reasonable amount of iterations.

**Table 6.2** Objectives for 3D problem.

| Problem Instance | | $\theta = \frac{\pi}{4}$, $\psi = \frac{\pi}{4}$ | | |
|---|---|---|---|---|
| Domain | Control | Relaxed Objective | Relaxed Rounded Objective | Trust Region Objective |
| $\bar{D}_0^1$ | $20 \times 20 \times 20$ | 0.0775 | 0.1489 | 0.0805 |
| $\bar{D}_0^2$ | $20 \times 20 \times 20$ | 0.2642 | 0.3286 | 0.2691 |
| $\bar{D}_0^3$ | $20 \times 20 \times 20$ | 0.0062 | 1.0060 | 0.0112 |

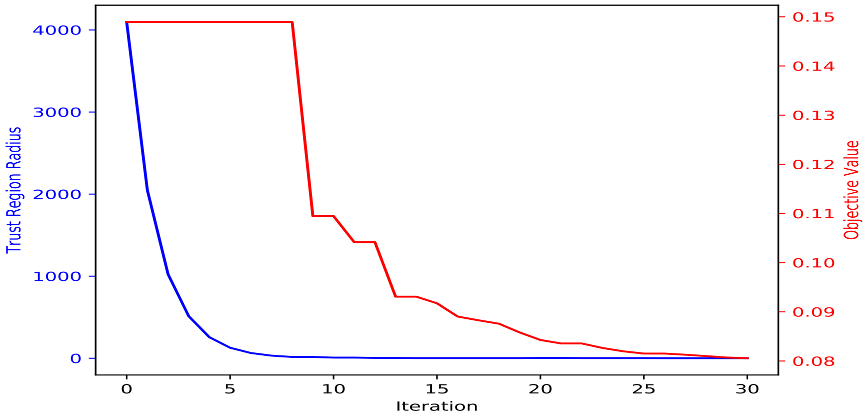


**Figure 6.1** Trust-region progress on rectangular prism experiment.

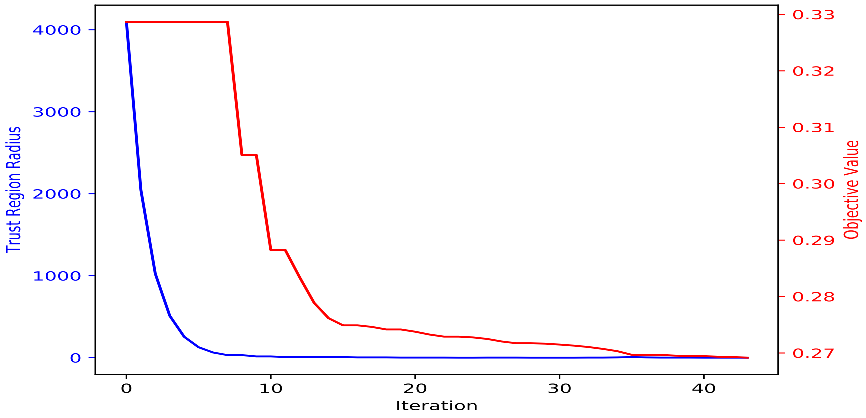


**Figure 6.2** Trust-region progress on cube experiment.

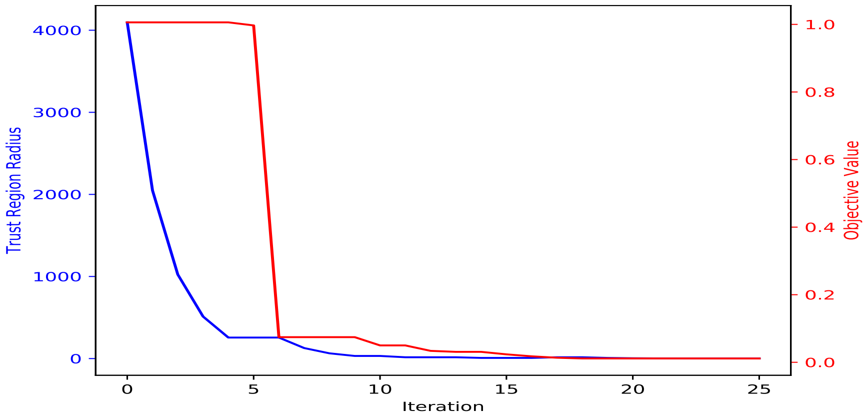


**Figure 6.3** Trust-region progress on sphere experiment.

### 6.1.3 Conclusion

In this chapter we posed a model that describes 3D binary electromagnetic cloaking with a single material which results in a reduced space MIPDECO problem. The model is a generalization of the 2D problem posed in Section 4.2.1. We derived the optimality conditions to produce a reduced gradient and applied our trust-region method. We showed that even with 8000 integer variable present in the problem, which is considered a large amount of integer variables, our trust-region method is successful at finding a local minimum to the problem in a reasonable amount of iterations. The fact that our method is still computationally tractable, even for a 3D PDE, is very promising. This indicates that our method can be applied to many problems in the class of MIPDECO.

CHAPTER

# 7

# FUTURE WORK

In this chapter we highlight directions for the future research regarding the topics discussed in this thesis.

### 7.0.1 The Immersed Interface Method and PDE Constrained Optimization Problems

A first natural direction for future work is to solve Maxwell's equations in three dimensions. We can use the tools developed for the one and two dimensional problem as a template to develop and analyze methods for Maxwell's equations in three dimensions. In this thesis we focused on a finite difference approach to solve interface problems, using the FDTD method as a template which we generalized. When using finite difference methods for interface problems, we assumed that the solution has sufficient regularity away from the interface. That is to say we search for solutions to satisfy Maxwell's equations in a classical sense. We can redo this analysis, using the IIM with the finite-element time domain (FETD) method, in order to find solutions that satisfy the equations in a weak sense.

Another direction is studying optimal control of Maxwell equations with interfaces. In general PDE optimization problems have been solved using the finite-element approach. Of course interface control problems can be solved with with our finite-difference approach, but there needs to be sufficient regularity in the state and control; in the absence of such regularity we need to search for non classical solutions to Maxwell's equations. This is a natural reason to derive the IIM using the FETD approach. We can derive the gradient and Hessian by using adjoint and second order adjoint approach, then apply an use either trust-region or line search to solve the problem. To our knowledge interface control problems have not been solved for Maxwell equations.

### 7.0.2 Mixed-Integer Partial-Differential Equation Constrained Optimization Theory

In this thesis we focused on using a trust-region method for solving MIPDECO. We constructed a gradient by solving the state and adjoint equation. In our method we do not use the Hessian. We can easily derive the Hessian by using the formal Lagrangian approach. Of course including Hessian information will lead to quicker convergence to a local minimum (both for the relaxation and trust-region solve), but it is not clear if this approach is viable for MIPDECO. The main reason is because the introduction of the Hessian makes our trust-region sub problem, which is a linear integer program (knapsack problem) in the steepest descent case, into a quadratic integer program, which is significantly harder to solve. These questions need to be studied further. Other methods that are classical to optimization can be modified for MIPDECO such as line search (Newton or quasi-Newton) and the augmented Lagrangian approach.

The electromagnetic cloaking problem studied in this thesis can be viewed as a single stage binary risk-neutral PDE optimization problem. An interesting direction would be to study the general case - multi-stage binary risk adverse-PDE optimization (MBRAPDEO). The combination of our trust-region method and techniques from stochastic programming can be combined to numerically solve this class of problems. Such problems have not been studied in depth due to the computational intractability of solving MIPDECO.

Another direction, which is purely theoretical, is motivated by the study and development of cuts in finite dimensional integer programming. A popular technique in integer programming is to introduce these cuts to the feasible region in order to make the problem more computationally tractable. The motivation is to introduce possibly several additional inequality constraints to the original integer program that isolates local minimums. Even thought the introduction of these additional constraints can cause more computational effort to solve the relaxation, the hope is branch and bound terminates “quickly” and does not take exponential time. Our interest is to introduce “cuts” in infinite dimension, for convex problems, which isolate the infinite dimensional minimums of the reduced space formulation. These “cuts” can be constructed with the intention to produce quality controls for a given MIPDECO. This can help both the OTD or DTO approach for MIPDECO. In the OTD case the cuts would appear in our knapsack sub problem. In the OTD the infinite dimensional cuts would be discretized, which then could easily be passed to a MINLP. Because these cuts are introduced in the infinite dimensional space rather than the discretized space they maintain some physical proprieties of the control (which in finite dimensions can be lost).

A final direction is applying our methodology to mixed-integer optimal control. In optimal control, bang-bang problems have already been viewed with tools from integer programming. The DTO approach is often used to solve these problems and there has been no study on the OTD approach to our best knowledge. We believe an alternative formulation of these problems combined with our algorithm can produce good local solutions, while also maintaining computational tractability.

# Bibliography

[1] Achterberg, T. & Berthold, T. “Improving the Feasibility Pump”. *Discrete Optimization* **4**.1 (2007), pp. 77–86.

[2] Achterberg, T. “SCIP: solving constraint integer programs”. *Mathematical Programming Computation* **1**.1 (2009), pp. 1–41.

[3] Adams, J. C., Swarztrauber, P. N. & Sweet, R. “FISHPACK90: Efficient FORTRAN Subprograms for the Solution of Separable Elliptic Partial Differential Equations”. *Astrophysics Source Code Library* (2016).

[4] Alexanderian, A., Petra, N., Stadler, G. & Ghattas, O. “Mean-variance risk-averse optimal control of systems governed by PDEs with random parameter fields using quadratic approximations”. *SIAM/ASA Journal on Uncertainty Quantification* **5**.1 (2017), pp. 1166–1192.

[5] Alnæs, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M. E. & Wells, G. N. “The Project Version 1.5”. *Archive of Numerical Software* **3**.100 (2015).

[6] Atamtürk, A. & Savelsbergh, M. W. “Integer-programming software systems”. *Annals of operations research* **140**.1 (2005), pp. 67–124.

[7] Ball, D. W. “Maxwell’s equations of electrodynamics: an explanation”. SPIE. 2012.

[8] Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J. & Mahajan, A. “Mixed-integer nonlinear optimization”. *Acta Numerica* **22** (2013), pp. 1–131.

[9] Bertacco, L., Fischetti, M. & Lodi, A. “A feasibility pump heuristic for general mixed-integer problems”. *Discrete Optimization* **4**.1 (2007), pp. 63–76.

[10] Bonami, P., Cornuéjols, G., Lodi, A. & Margot, F. “A Feasibility Pump for Mixed Integer Nonlinear Programs”. *Mathematical Programming* **119** (2009), pp. 331–352.

[11] Bonami, P., Cornuéjols, G., Lodi, A. & Margot, F. “A feasibility pump for mixed integer nonlinear programs”. *Mathematical Programming* **119**.2 (2009), pp. 331–352.

[12] Borzi, A. & Schulz, V. *Computational optimization of systems governed by partial differential equations*. Vol. 8. SIAM, 2011.

[13] Brucker, P., Jurisch, B. & Sievers, B. “A branch and bound algorithm for the job-shop scheduling problem”. *Discrete applied mathematics* **49**.1-3 (1994), pp. 107–127.

[14] Cai, W. *Computational Methods for Electromagnetic Phenomena: electrostatics in solvation, scattering, and electron transport*. Cambridge University Press, 2013.

[15] Cai, W., Chettiar, U. K., Kildishev, A. V. & Shalaev, V. M. “Optical cloaking with metamaterials”. *Nature photonics* **1**.4 (2007), p. 224.

[16] Campolattaro, A. A. “Generalized Maxwell equations and quantum mechanics. I. Dirac equation for the free electron”. *International Journal of Theoretical Physics* **29**.2 (1990), pp. 141–155.

[17] Cannon, P. D. & Honary, F. “A GPU-accelerated finite-difference time-domain scheme for electromagnetic wave interaction with plasma”. *IEEE Transactions on Antennas and Propagation* **63**.7 (2015), pp. 3042–3054.

[18] Clausen, J. “Branch and bound algorithms-principles and examples”. *Department of Computer Science, University of Copenhagen* (1999), pp. 1–30.

[19] Constantin, P. & Foias, C. *Navier-stokes equations*. University of Chicago Press, 1988.

[20] Cui, T. J., Smith, D. R. & Liu, R. *Metamaterials*. Springer, 2010.

[21] Cummer, S. A., Popa, B.-I., Schurig, D., Smith, D. R. & Pendry, J. “Full-wave simulations of electromagnetic cloaking structures”. *Physical Review E* **74**.3 (2006), p. 036621.

[22] Dener, A., Munson, T., Sarich, J., Wild, S., Benson, S. & McInnes, L. C. *TAO 3.9 Users Manual*. Argonne Technical Memorandum ANL/MCS-TM-242. http://www.mcs.anl.gov/petsc. Mathematics and Computer Science Division, Argonne National Laboratory, revised April 2018.

[23] Deng, S. “On the immersed interface method for solving time-domain Maxwell’s equations in materials with curved dielectric interfaces”. *Computer physics communications* **179**.11 (2008), pp. 791–800.

[24] Deng, S., Ito, K. & Li, Z. “Three-dimensional elliptic solvers for interface problems and applications”. *Journal of Computational Physics* **184**.1 (2003), pp. 215–243.

[25] Ditkowski, A, Dridi, K & Hesthaven, J. S. “Convergent Cartesian grid methods for Maxwell’s equations in complex geometries”. *Journal of Computational Physics* **170**.1 (2001), pp. 39–80.

[26] Erdemli, Y., Reddy, C. & Volakis, J. L. “AWE technique in frequency domain electromagnetics”. *Journal of Electromagnetic Waves and Applications* **13**.3 (1999), pp. 359–378.

[27] Erlangga, Y. A., Vuik, C. & Oosterlee, C. W. “On a class of preconditioners for solving the Helmholtz equation”. *Applied Numerical Mathematics* **50**.3-4 (2004), pp. 409–425.

[28] Evans, L. “Computer-Aided Design: Advances in Process Flowsheeting Systems”. *Recent Developments in Chemical Process and Plant Design*. Wiley, 1987, pp. 261–287.

[29] Feinerman, E, Bresler, E & Dagan, G. “Optimization of a spatially variable resource: An illustration for irrigated crops”. *Water resources research* **21**.6 (1985), pp. 793–800.

[30] Fischer, T. & Pfetsch, M. E. “Branch-and-cut for linear programs with overlapping SOS1 constraints”. *Mathematical Programming Computation* **10**.1 (2018), pp. 33–68.

[31] Fischetti, M., González, J. & Toth, P. “Solving the Orienteering Problem through Branch-and-Cut”. *INFORMS Journal on Computing* **10** (1998), pp. 133–148.

[32] Fischetti, M. & Lodi, A. “Local Branching”. *Mathematical Programming* **98** (2002), pp. 23–47.

[33] Fischetti, M., Glover, F. & Lodi, A. “The feasibility pump”. *Mathematical Programming* **104**.1 (2005), pp. 91–104.

[34] Garmatter, D., Porcelli, M., Rinaldi, F. & Stoll, M. “Improved Penalty Algorithm for Mixed Integer PDE Constrained Optimization (MIPDECO) Problems”. *arXiv preprint arXiv:1907.06462* (2019).

[35] Geihe, B., Lenz, M., Rumpf, M. & Schultz, R. “Risk averse elastic shape optimization with parametrized fine scale geometry”. *Mathematical Programming* **141**.1-2 (2013), pp. 383–403.

[36] Gunzburger, M. D. *Perspectives in flow control and optimization*. Vol. 5. Siam, 2003.

[37] Hahn, M., Sager, S. & Leyffer, S. *Binary Optimal Control by Trust-Region Topological Steepest Descent*. Tech. rep. In preparation. Argonne National Laboratory, 2018.

[38] Hahn, M., Leyffer, S. & Zavala, V. M. *Mixed-Integer PDE-Constrained Optimal Control of Gas Networks*. Tech. rep. ANL/MCS-P7095-0817. Argonne National Laboratory, Mathematics and Computer Science Division, 2017.

[39] Hante, F. M. “Relaxation methods for hyperbolic PDE mixed-integer optimal control problems”. *Optimal Control Applications and Methods* **38**.6 (2017), pp. 1103–1110.

[40] Hante, F. M. “Mixed-Integer optimal control for PDEs: Relaxation via differential inclusions and applications to gas network optimization”. *Mathematical Modelling, Optimization, Analytic and Numerical Solutions*. Springer, 2020, pp. 157–171.

[41] Haslinger, J. & Mäkinen, R. A. “On a topology optimization problem governed by two-dimensional Helmholtz equation”. *Computational Optimization and Applications* **62**.2 (2015), pp. 517–544.

[42] Herzog, R. & Kunisch, K. “Algorithms for PDE-constrained optimization”. *GAMM-Mitteilungen* **33**.2 (2010), pp. 163–176.

[43] Hicken, J. & Alonso, J. “Comparison of reduced-and full-space algorithms for PDE-constrained optimization”. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2013, p. 1043.

[44] Hoffman, K. L. & Padberg, M. “Solving airline crew scheduling problems by branch-and-cut”. *Management science* **39**.6 (1993), pp. 657–682.

[45] Holmberg, K. “Cross decomposition applied to integer programming problems: Duality gaps and convexification in parts”. *Operations research* **42**.4 (1994), pp. 657–668.

[46] Hunter, J. K., Li, Z. & Zhao, H. “Reactive autophobic spreading of drops”. *Journal of Computational Physics* **183**.2 (2002), pp. 335–366.

[47] Inan, U. S. & Marshall, R. A. *Numerical electromagnetics: the FDTD method*. Cambridge University Press, 2011.

[48] Ito, K. & Kunisch, K. *Lagrange multiplier approach to variational problems and applications*. Vol. 15. Siam, 2008.

[49] Ito, K., Kunisch, K. & Li, Z. “Level-set function approach to an inverse interface problem”. *Inverse problems* **17**.5 (2001), p. 1225.

[50] Jackson, J. D. *Classical electrodynamics*. 1999.

[51] Jin, S. & Wang, X. “Robust numerical simulation of porosity evolution in chemical vapor infiltration: II. two-dimensional anisotropic fronts”. *Journal of Computational Physics* **179**.2 (2002), pp. 557–577.

[52] Jones, E., Oliphant, T. & Peterson, P. “SciPy: Open source scientific tools for Python” (2001).

[53] Kelley, C. T. *Iterative methods for optimization*. SIAM, 1999.

[54] Kelley, C. T. *Solving nonlinear equations with Newton’s method*. Vol. 1. Siam, 2003.

[55] Kelley Jr, J. E. “The cutting-plane method for solving convex programs”. *Journal of the society for Industrial and Applied Mathematics* **8**.4 (1960), pp. 703–712.

[56] Kihlstrom, R. E. & Laffont, J.-J. “A general equilibrium entrepreneurial theory of firm formation based on risk aversion”. *Journal of political economy* **87**.4 (1979), pp. 719–748.

[57] Langacker, P. & Pi, S.-Y. “Magnetic monopoles in grand unified theories”. *Physical Review Letters* **45**.1 (1980), p. 1.

[58] Lawler, E. L. & Woods, D. E. “Branch-and-bound methods: A survey”. *Operations Research* **14**.4 (1966), pp. 699–719.

[59] Lee, J. & Leyffer, S. *Mixed integer nonlinear programming*. Vol. 154. Springer Science & Business Media, 2011.

[60] LeVeque, R. J. & Leveque, R. J. *Numerical methods for conservation laws*. Vol. 3. Springer, 1992.

[61] Li, D. & Sun, X. *Nonlinear integer programming*. Vol. 84. Springer Science & Business Media, 2006.

[62] Li, J & Huang, Y. *Time-domain finite element methods for Maxwell's equations in metamaterials*. Springer Series in Computational Mathematics, 2013.

[63] Li, Z. "A fast iterative algorithm for elliptic interface problems". *SIAM Journal on Numerical Analysis* **35**.1 (1998), pp. 230–254.

[64] Li, Z. & Ito, K. *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*. Vol. 33. Siam, 2006.

[65] Li, Z., Ji, H. & Chen, X. "Accurate solution and gradient computation for elliptic interface problems with variable coefficients". *SIAM journal on numerical analysis* **55**.2 (2017), pp. 570–597.

[66] Li, Z., Qiao, Z. & Tang, T. *Numerical Solution of Differential Equations: Introduction to Finite Difference and Finite Element Methods*. Cambridge University Press, 2017.

[67] Li, Z. & Song, P. "An adaptive mesh refinement strategy for immersed boundary/interface methods". *Communications in computational physics* **12**.2 (2012), pp. 515–527.

[68] Li, Z. & Song, P. "Adaptive mesh refinement techniques for the immersed interface method applied to flow problems". *Computers & structures* **122** (2013), pp. 249–258.

[69] Li, Z., Zhao, H. & Gao, H. "A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid". *Journal of Computational Physics* **152**.1 (1999), pp. 281–304.

[70] Linderoth, J. T. & Savelsbergh, M. W. "A computational study of search strategies for mixed integer programming". *INFORMS Journal on Computing* **11**.2 (1999), pp. 173–187.

[71] Logan, J. D. *Applied partial differential equations*. Springer, 2014.

[72] Mahajan, A., Leyffer, S., Linderoth, J., Luedtke, J. & Munson, T. *Minotaur: A Mixed-Integer Nonlinear Optimization Toolkit*. Tech. rep. ANL/MCS-P8010-0817. Argonne National Laboratory, Mathematics and Computer Science Division, 2017.

[73] Martí, R. "Multi-start methods". *Handbook of metaheuristics*. Springer, 2003, pp. 355–368.

[74] Maxwell, J. C. *A treatise on electricity and magnetism*. Vol. 1. Clarendon press, 1881.

[75] Meyer, C. D. *Matrix analysis and applied linear algebra*. Vol. 71. Siam, 2000.

[76] Min, M. & Teng, C. "The instability of the Yee scheme for the magic time step". *Journal of Computational Physics* **166**.2 (2001), pp. 418–424.

[77] Mitten, L. "Branch-and-bound methods: General formulation and properties". *Operations Research* **18**.1 (1970), pp. 24–34.

[78] Morefield, C. “Application of 0-1 integer programming to multitarget tracking problems”. *IEEE Transactions on Automatic Control* **22**.3 (1977), pp. 302–312.

[79] Nannicini, G., Belotti, P. & Liberti, L. *A local branching heuristic for MINLPs*. `http://arxiv.org/abs/0812.2188`. 2008.

[80] Nocedal, J. & Wright, S. J. *Numerical Optimization*. New York: Springer-Verlag, 1999.

[81] Osher, S., Fedkiw, R. & Piechor, K. “Level set methods and dynamic implicit surfaces”. *Appl. Mech. Rev.* **57**.3 (2004), B15–B15.

[82] Pendry, J. B., Schurig, D. & Smith, D. R. “Controlling electromagnetic fields”. *science* **312**.5781 (2006), pp. 1780–1782.

[83] Petra, N. & Stadler, G. *Model variational inverse problems governed by partial differential equations*. Tech. rep. University of Texas at Austin, Institute for Computational Engineering and Sciences, 2011.

[84] Reyes, J. C. De los. *Numerical PDE-constrained optimization*. Springer, 2015.

[85] Sakurai, J. J. & Commins, E. D. *Modern quantum mechanics, revised edition*. 1995.

[86] Salsa, S. *Partial differential equations in action: from modelling to theory*. Vol. 99. Springer, 2016.

[87] Schurig, D., Mock, J. J., Justice, B., Cummer, S. A., Pendry, J. B., Starr, A. F. & Smith, D. R. “Metamaterial electromagnetic cloak at microwave frequencies”. *Science* **314**.5801 (2006), pp. 977–980.

[88] Schurig, D., Mock, J., Justice, B., Cummer, S. A., Pendry, J. B., Starr, A. & Smith, D. “Metamaterial electromagnetic cloak at microwave frequencies”. *Science* **314**.5801 (2006), pp. 977–980.

[89] Selvanayagam, M. & Eleftheriades, G. V. “Experimental demonstration of active electromagnetic cloaking”. *Physical review X* **3**.4 (2013), p. 041011.

[90] Sen, S. & Higle, J. L. “The C 3 theorem and a D 2 algorithm for large scale stochastic mixed-integer programming: Set convexification”. *Mathematical Programming* **104**.1 (2005), pp. 1–20.

[91] Sethian, J. A. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Vol. 3. Cambridge university press, 1999.

[92] Shapiro, A., Dentcheva, D. & Ruszczyński, A. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[93] Sklarz, S. E. & Tannor, D. J. “Loading a Bose-Einstein condensate onto an optical lattice: An application of optimal control theory to the nonlinear Schrödinger equation”. *Physical Review A* **66**.5 (2002), p. 053619.

[94] Takahashi, R. “Arrival time differences between gravitational waves and electromagnetic signals due to gravitational lensing”. *The Astrophysical Journal* **835**.1 (2017), p. 103.

[95] Tawarmalani, M. & Sahinidis, N. V. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*. Vol. 65. Springer Science & Business Media, 2013.

[96] Trefethen, L. N. “Finite difference and spectral methods for ordinary and partial differential equations” (1996).

[97] Tröltzsch, F. *Optimal Control of Partial Differential Equations*. Vol. 112. Graduate Studies in Mathematics. Providence: American Mathematical Society, 2010.

[98] Turkel, E & Yefet, A. “On the construction of a high order difference scheme for complex domains in a Cartesian grid”. *Applied Numerical Mathematics* **33**.1-4 (2000), pp. 113–124.

[99] Vielma, J. P. & Nemhauser, G. L. “Modeling disjunctive constraints with a logarithmic number of binary variables and constraints”. *International Conference on Integer Programming and Combinatorial Optimization*. Springer. 2008, pp. 199–213.

[100] Vielma, J. P. & Nemhauser, G. L. “Modeling disjunctive constraints with a logarithmic number of binary variables and constraints”. *Mathematical Programming* **128**.1-2 (2011), pp. 49–72.

[101] Weiland, T. “Time domain electromagnetic field computation with finite difference methods”. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* **9**.4 (1996), pp. 295–319.

[102] Wesselhoeft, C. “Mixed-Integer PDE-Constrained Optimization”. PhD thesis. Imperial College London, 2017.

[103] West, D. B. et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, 2001.

[104] Wolsey, L. A. *Integer programming*. Vol. 52. John Wiley & Sons, 1998.

[105] Yee, K. “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media”. *IEEE Transactions on antennas and propagation* **14**.3 (1966), pp. 302–307.

[106] Yefet, A. & Petropoulos, P. G. “A staggered fourth-order accurate explicit finite difference scheme for the time-domain Maxwell’s equations”. *Journal of Computational Physics* **168**.2 (2001), pp. 286–315.

[107] Zhang, C. & LeVeque, R. J. “The immersed interface method for acoustic wave equations with discontinuous coefficients”. *Wave motion* **25**.3 (1997), pp. 237–263.

[108] Zhang, C. & LeVeque, R. J. “The immersed interface method for acoustic wave equations with discontinuous coefficients”. *Wave motion* **25**.3 (1997), pp. 237–263.

[109] Zhao, S. “High order matched interface and boundary methods for the Helmholtz equation in media with arbitrarily curved interfaces”. *Journal of Computational Physics* **229**.9 (2010), pp. 3155–3170.

[110] Zhao, S. & Wei, G. “High-order FDTD methods via derivative matching for Maxwell’s equations with material interfaces”. *Journal of Computational Physics* **200**.1 (2004), pp. 60–103.

[111] Zhao, Y., Ding, D. & Chen, R. “A discontinuous Galerkin time-domain integral equation method for electromagnetic scattering from PEC objects”. *IEEE Transactions on Antennas and Propagation* **64**.6 (2016), pp. 2410–2417.

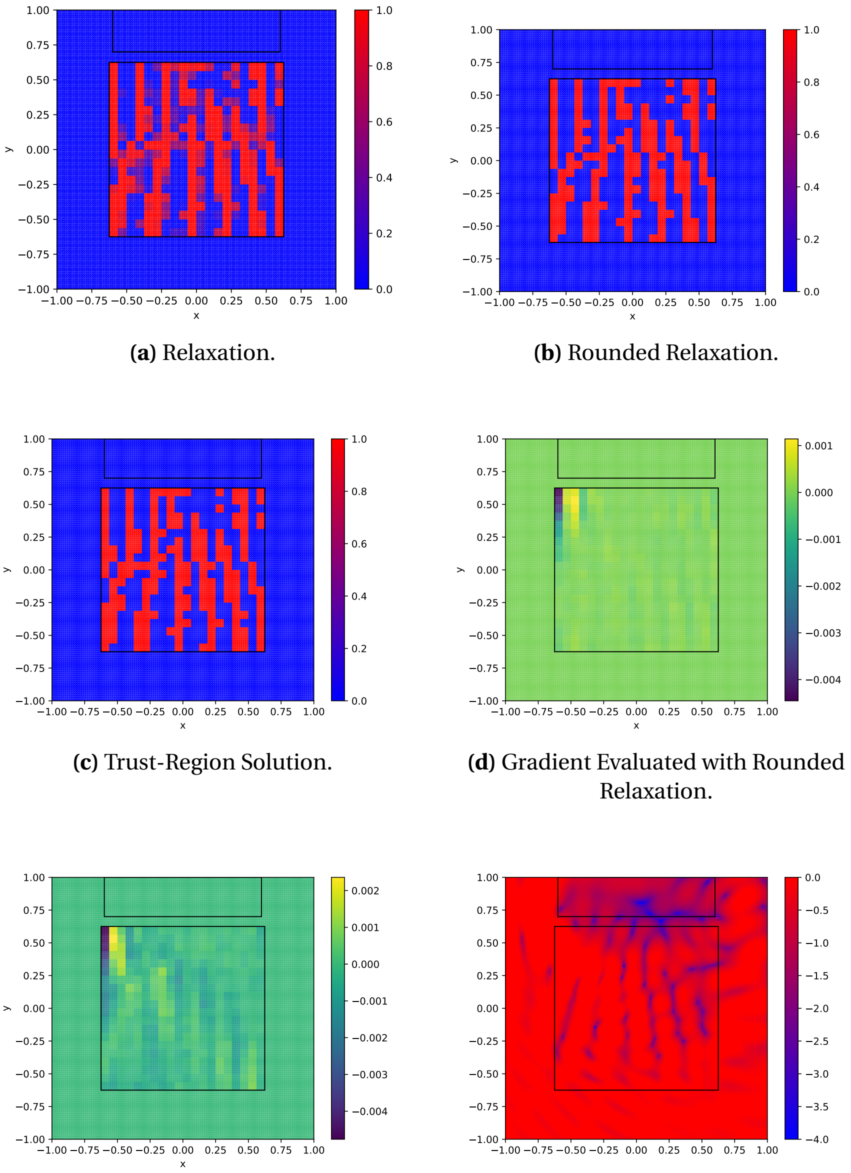
[112] Zhu, C., Byrd, R. H., Lu, P. & Nocedal, J. “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization”. *ACM Transactions on Mathematical Software (TOMS)* **23**.4 (1997), pp. 550–560.

# APPENDICES

## .1 Simulations for 2D Binary Electromagnetic Cloaking using a Single Material

This portion of the appendix shows the solution of the relaxation, its rounding, the final trust-region design, the final gradients, and the scattering field for the different test problems defined in Section 4.2.6.

### .1.1 Nominal angle $\theta = \frac{\pi}{4}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

**(d)** Gradient Evaluated with Rounded Relaxation.

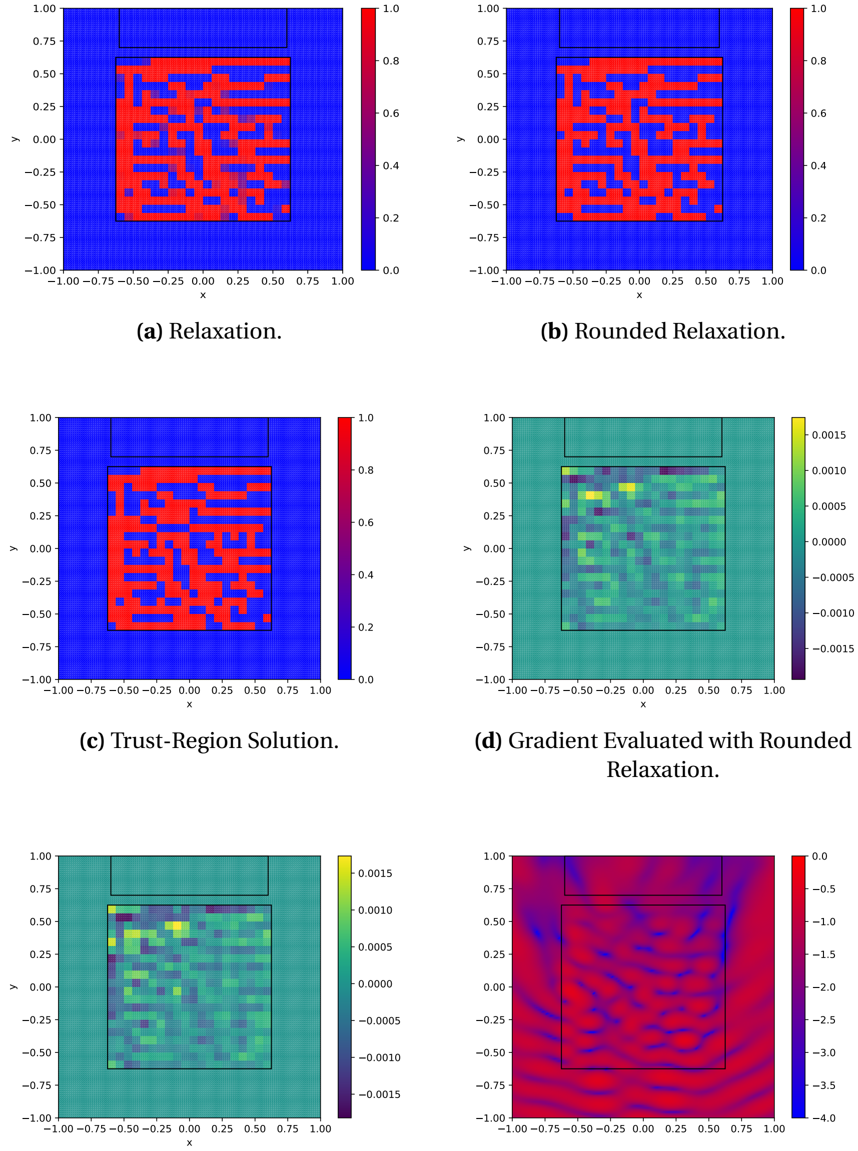
**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 1** Rectangle 20x20: Nominal angle $\theta = \frac{\pi}{4}$.

### .1.2 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$

### .1.3 Nominal angle $\theta = \frac{\pi}{4}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

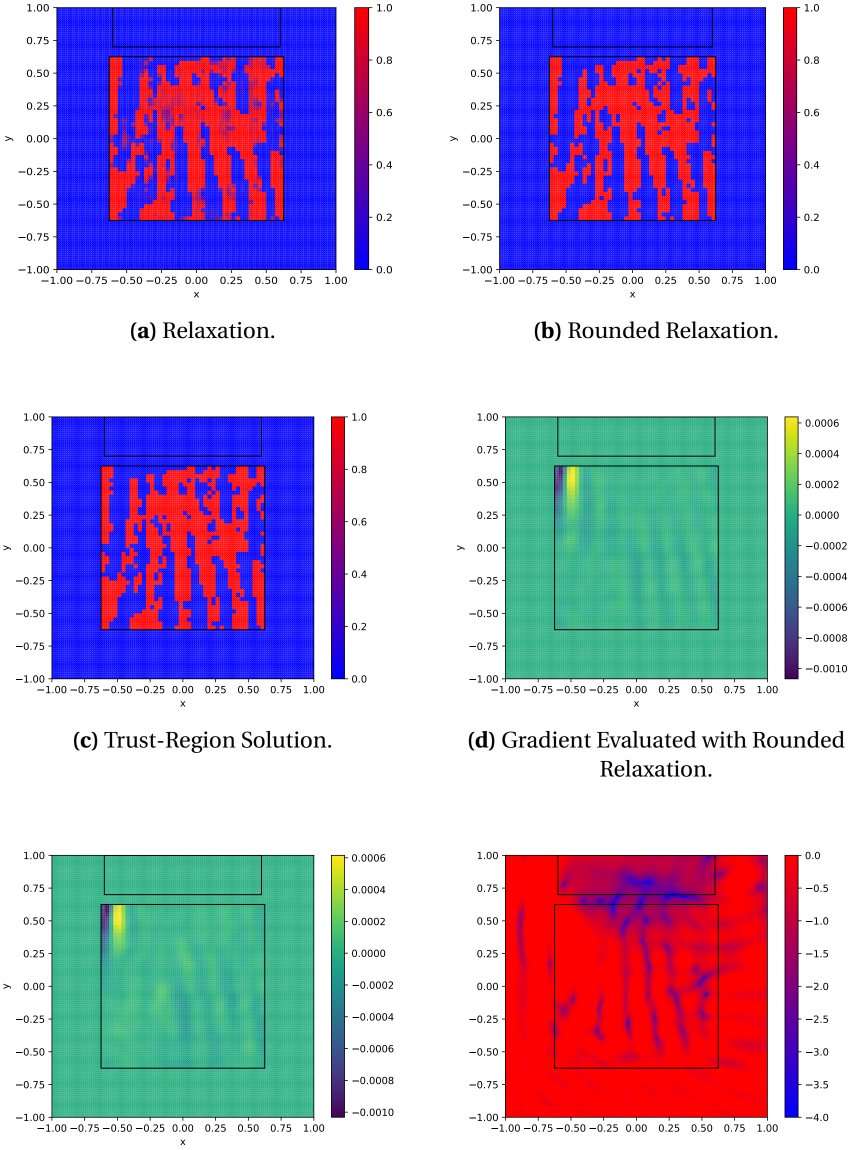
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 2** Rectangle 20x20: 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$.

### .1.4 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

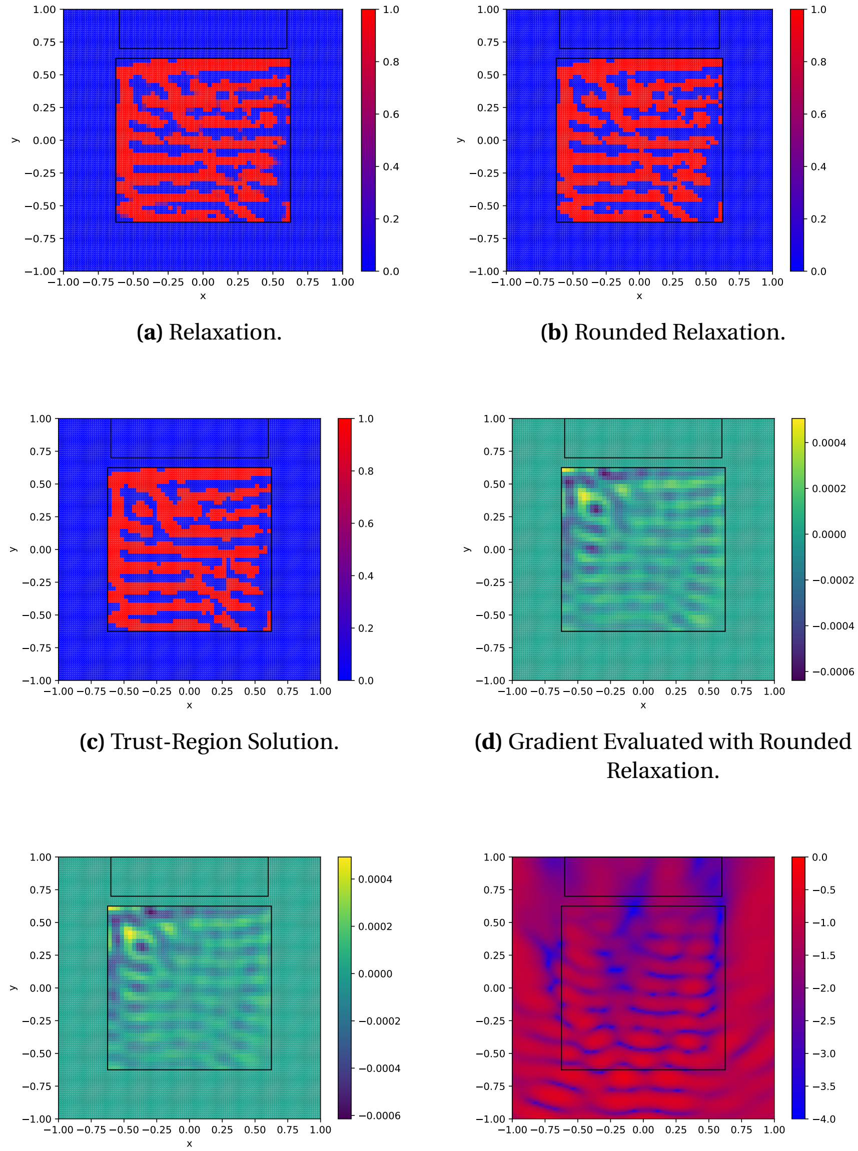
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 3** Rectangle 40x40: Nominal angle $\theta = \frac{\pi}{4}$.

## .1.5 Nominal angle $\theta = \frac{\pi}{4}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

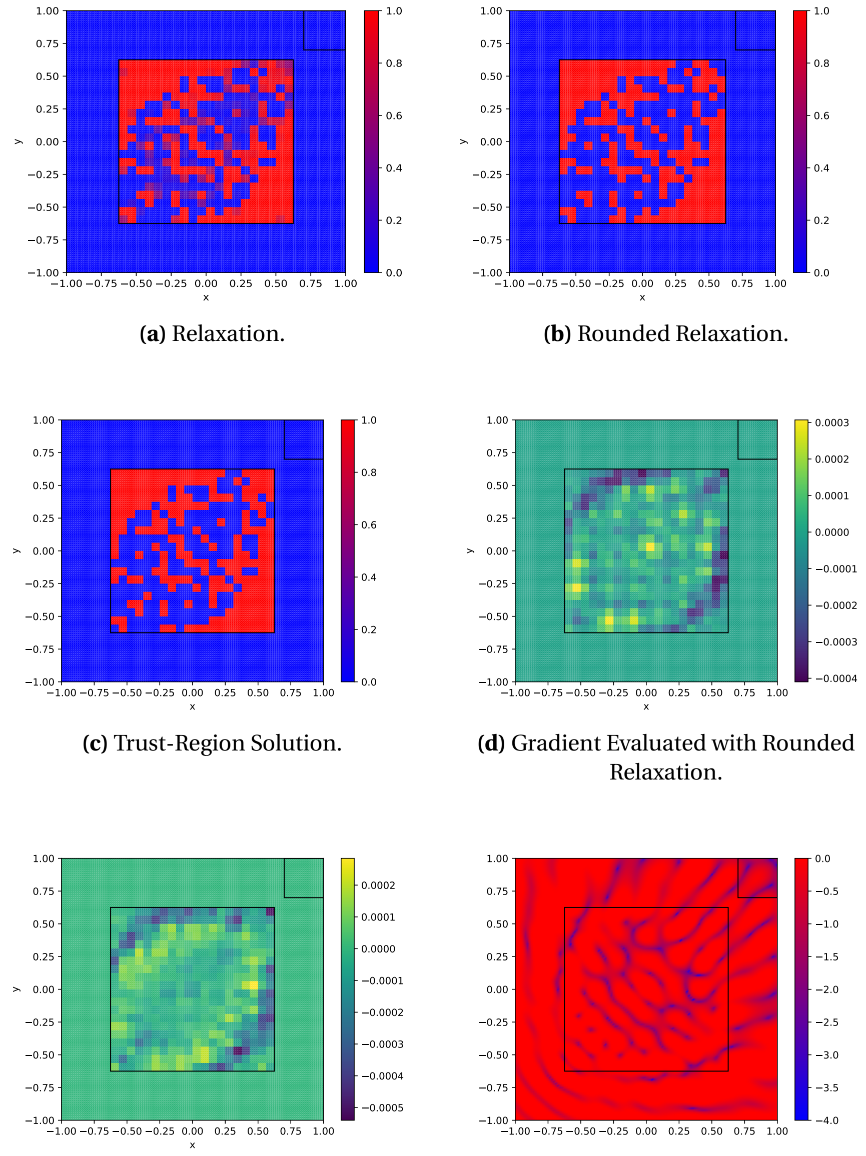
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 4** Rectangle 40x40: 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$.

### .1.6 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

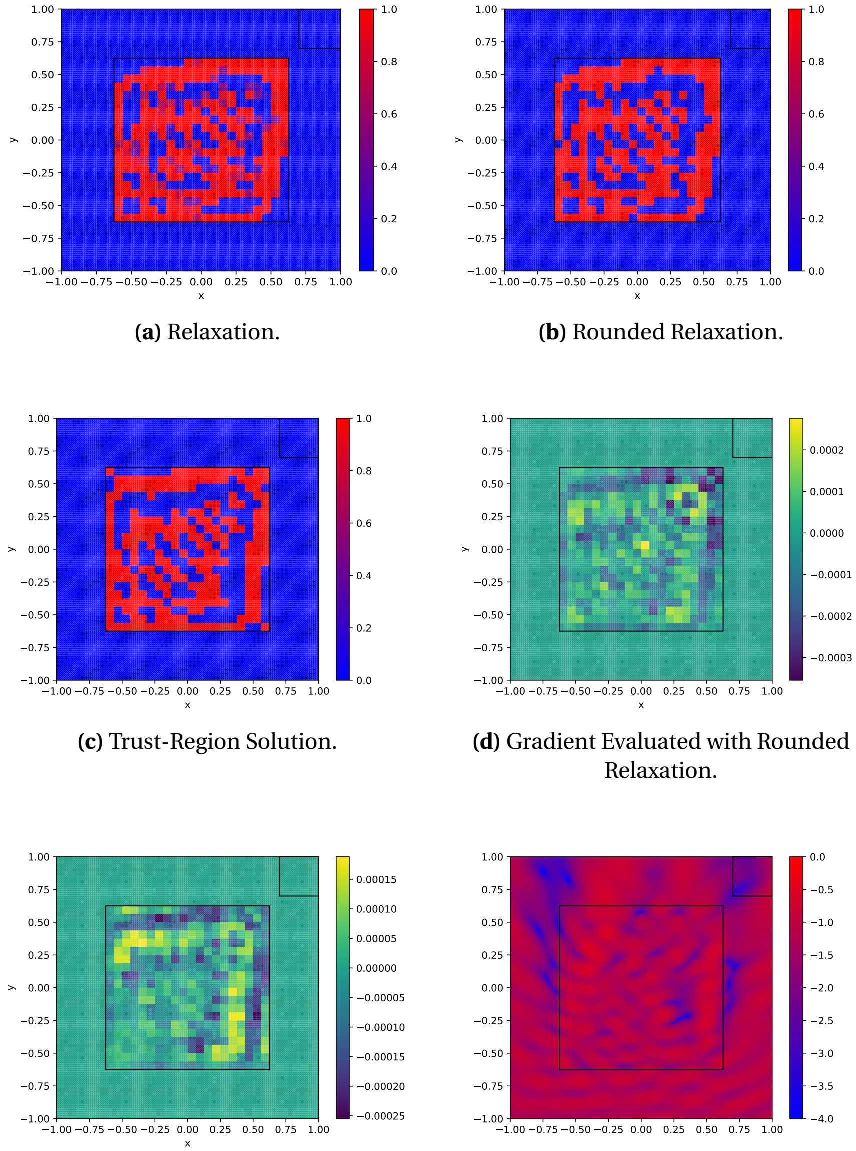
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 5** Square 20x20: Nominal angle $\theta = \frac{\pi}{4}$.

## .1.7 Nominal angle $\theta = \frac{\pi}{4}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

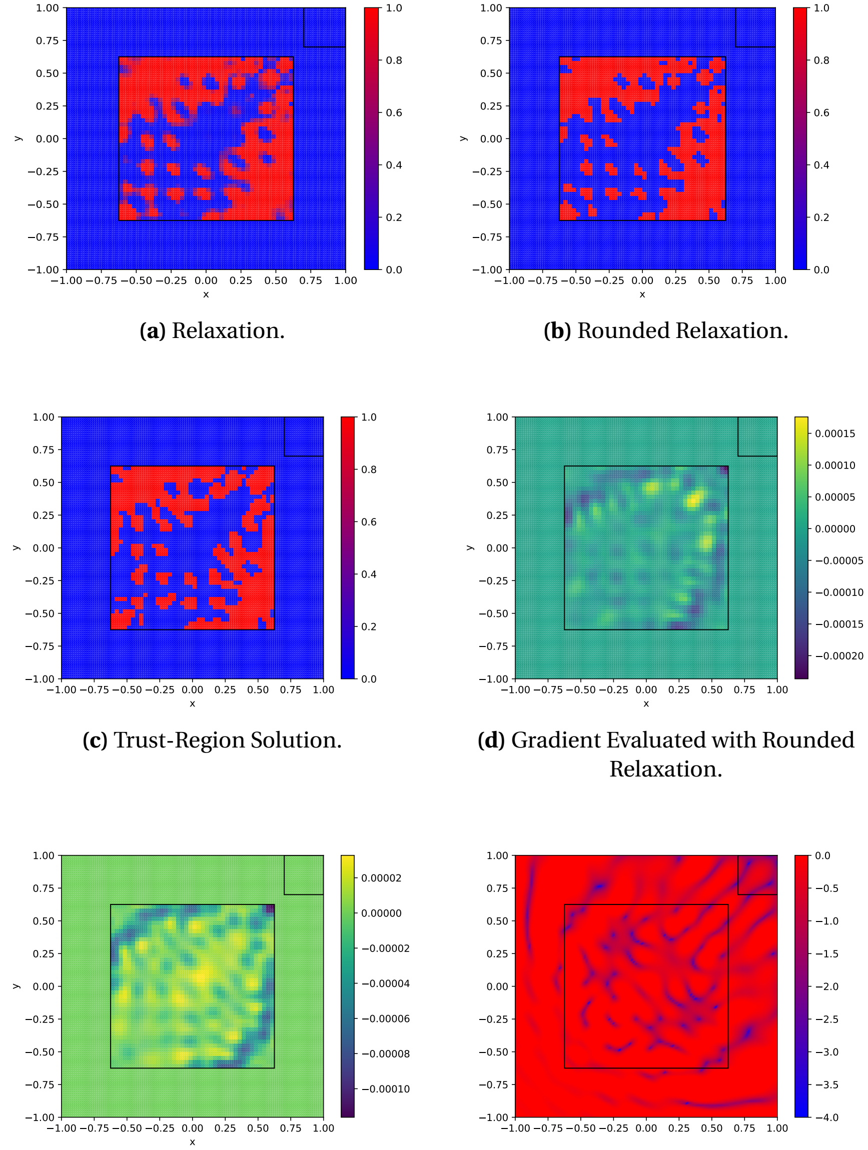
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 6** Square 20x20: 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$.

### .1.8 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

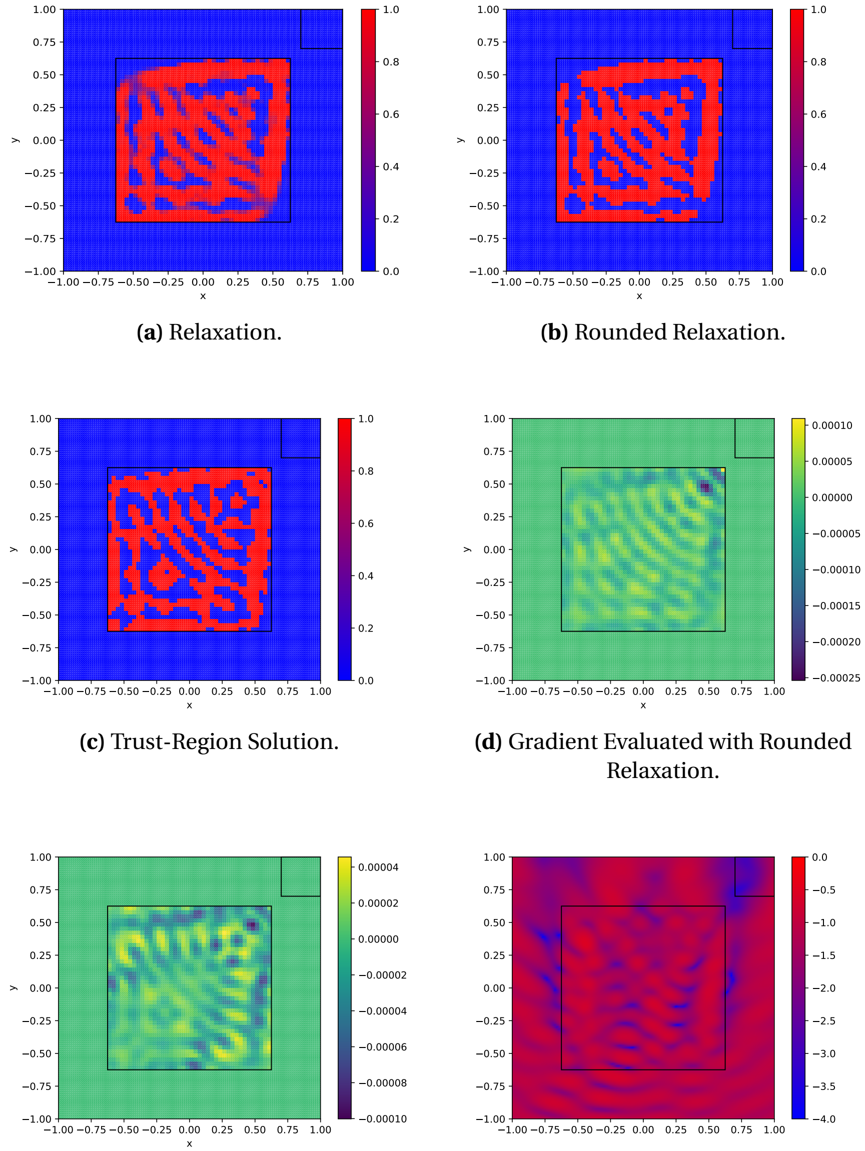
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 7** Square 40x40: Nominal angle $\theta = \frac{\pi}{4}$.

### .1.9 Nominal angle $\theta = \frac{\pi}{4}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

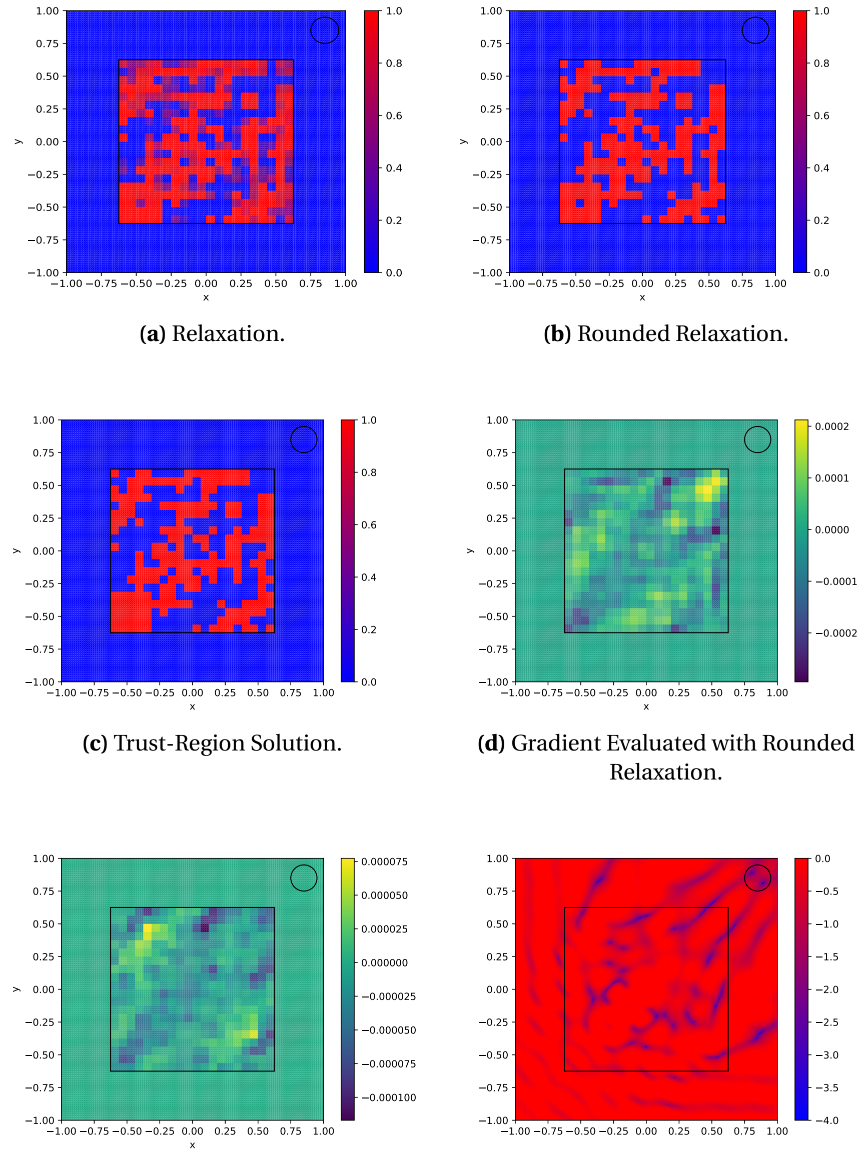
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 8** Square 40x40: 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$.

### .1.10 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

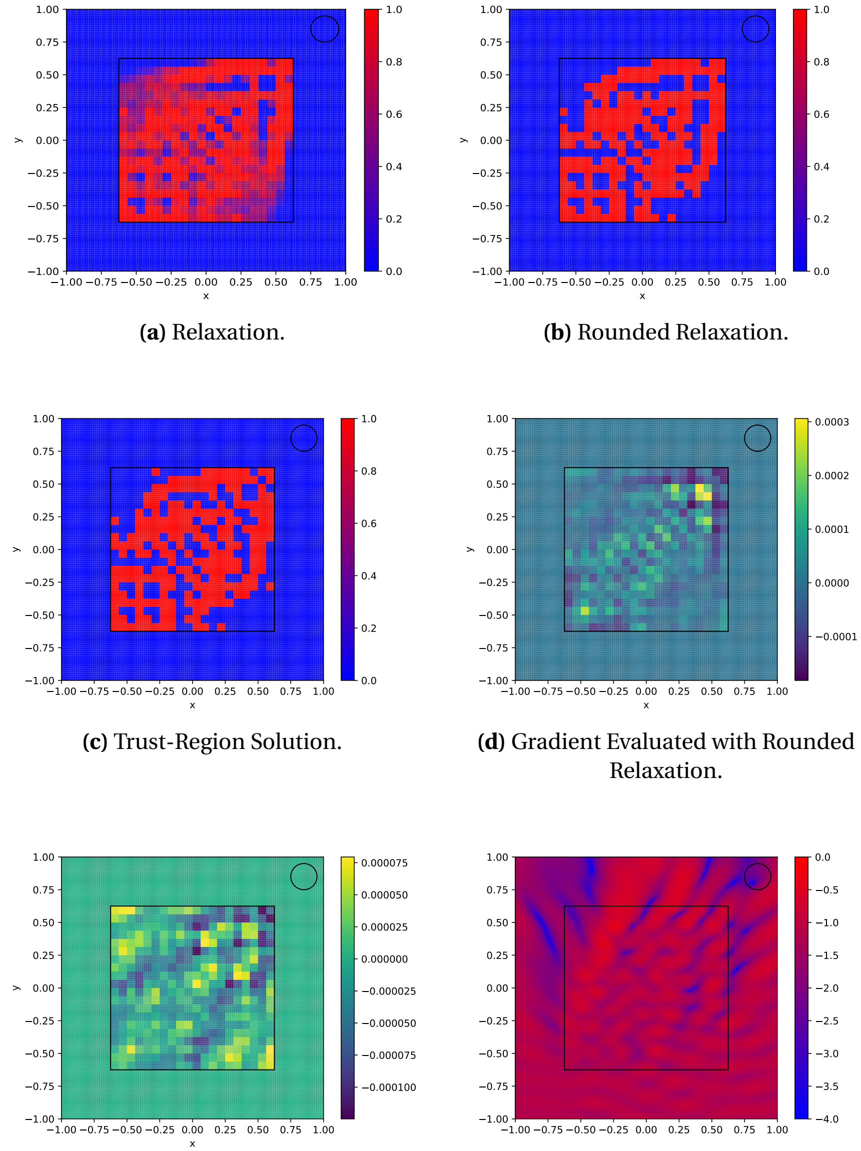
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 9** Circle 20x20: Nominal angle $\theta = \frac{\pi}{4}$.

### .1.11 Nominal angle $\theta = \frac{\pi}{4}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

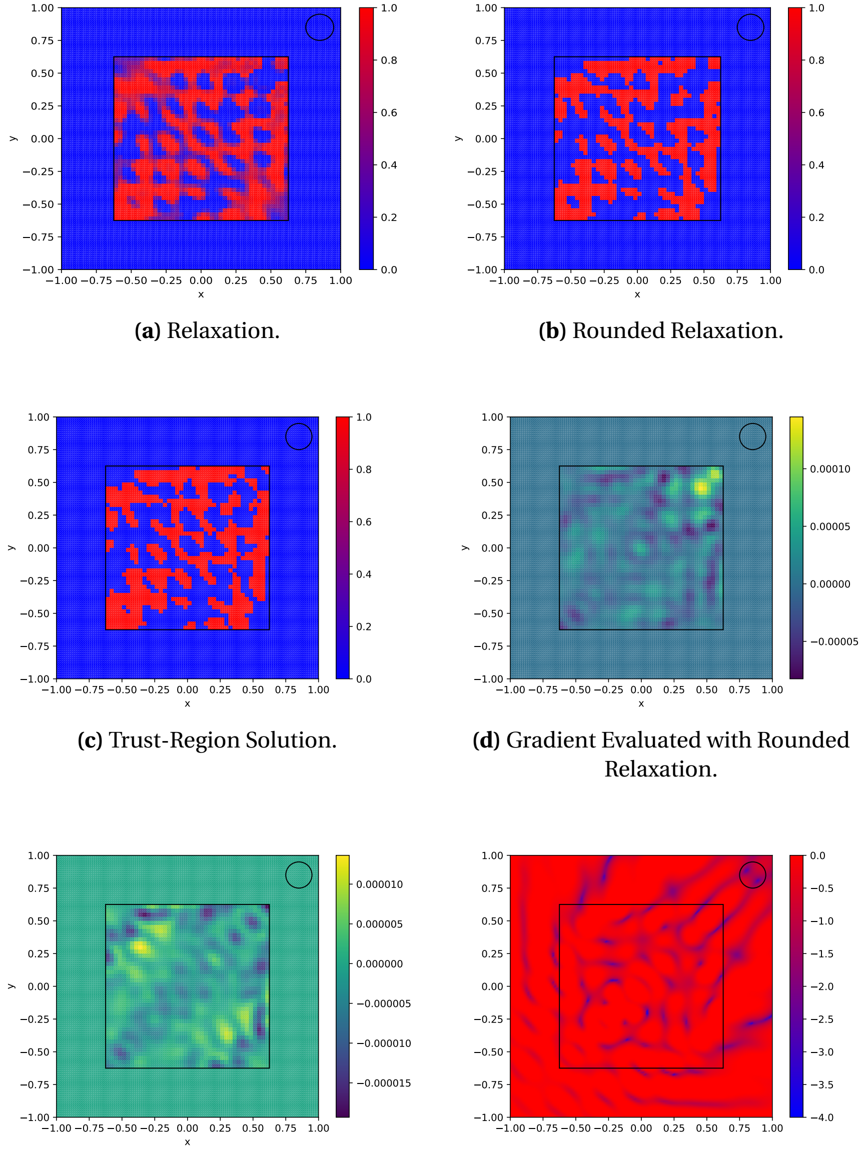
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 10** Circle 20x20: 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$.

### .1.12 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

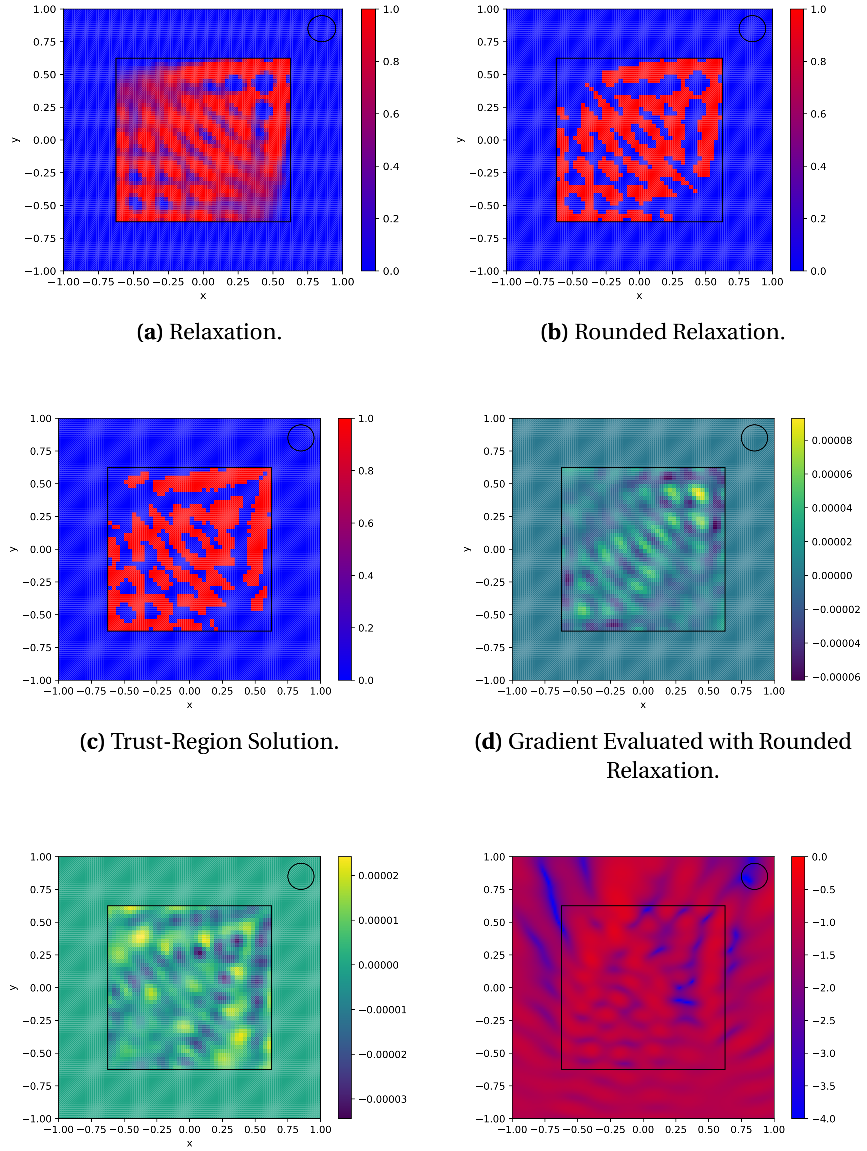
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 11** Circle 40x40: Nominal angle $\theta = \frac{\pi}{4}$.

### .1.13 Nominal angle $\theta = \frac{\pi}{2}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

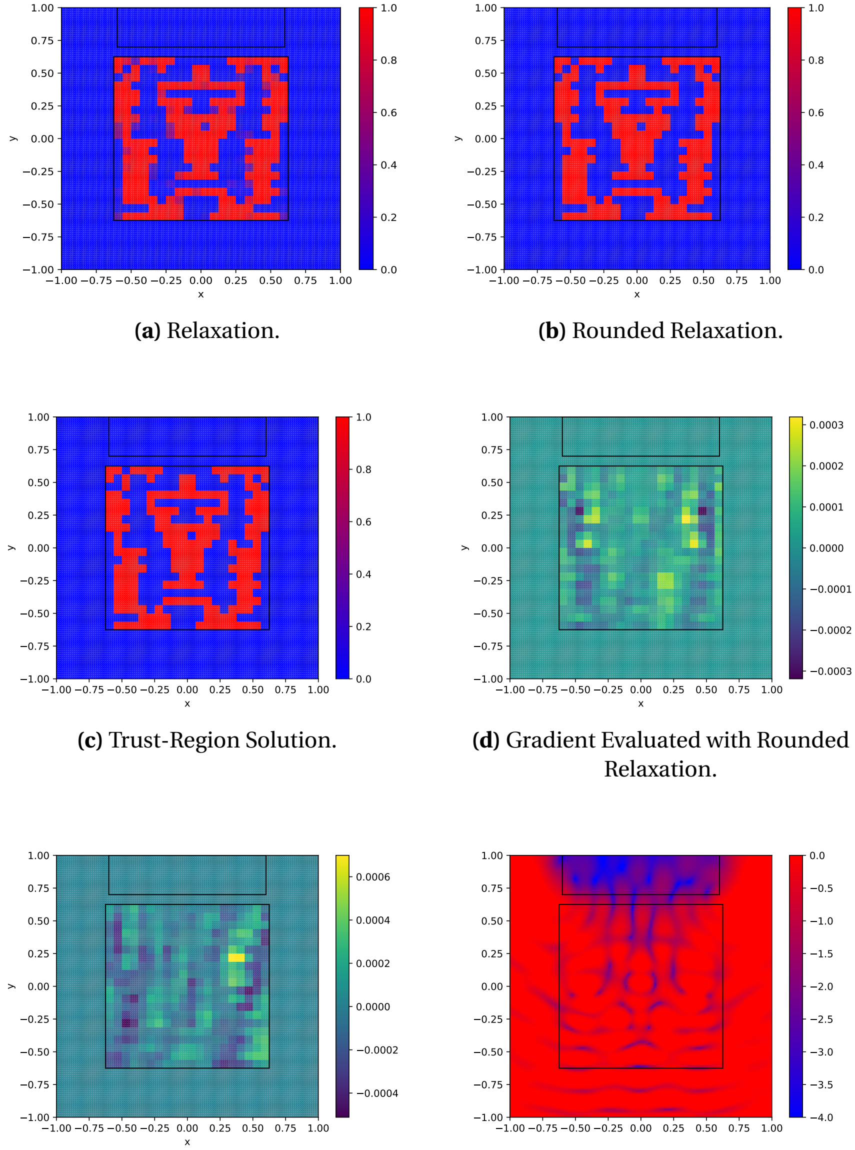
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 12** Circle 40x40: 15 Angles $\theta \in \left[0, \frac{\pi}{2}\right]$.

### .1.14 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

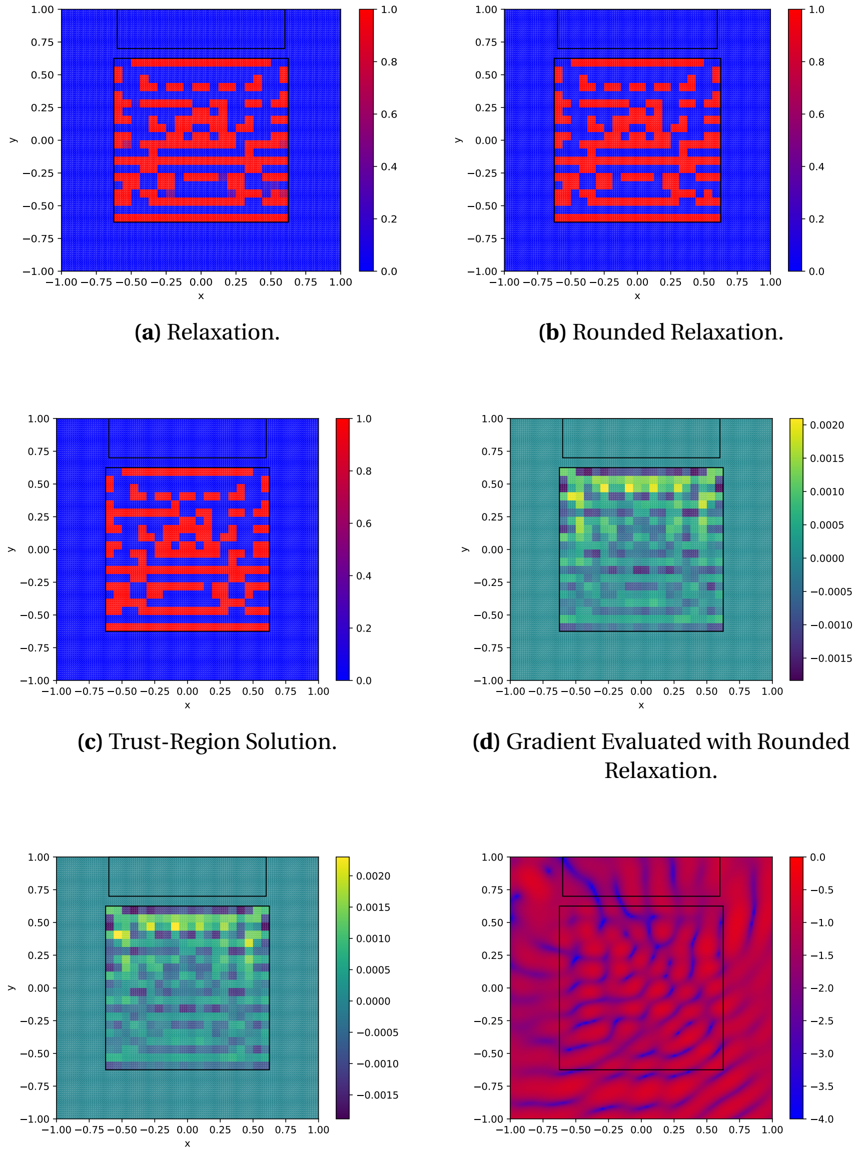
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 13** Rectangle 20x20: Nominal angle $\theta = \frac{\pi}{2}$.

### .1.15 Nominal angle $\theta = \frac{\pi}{2}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

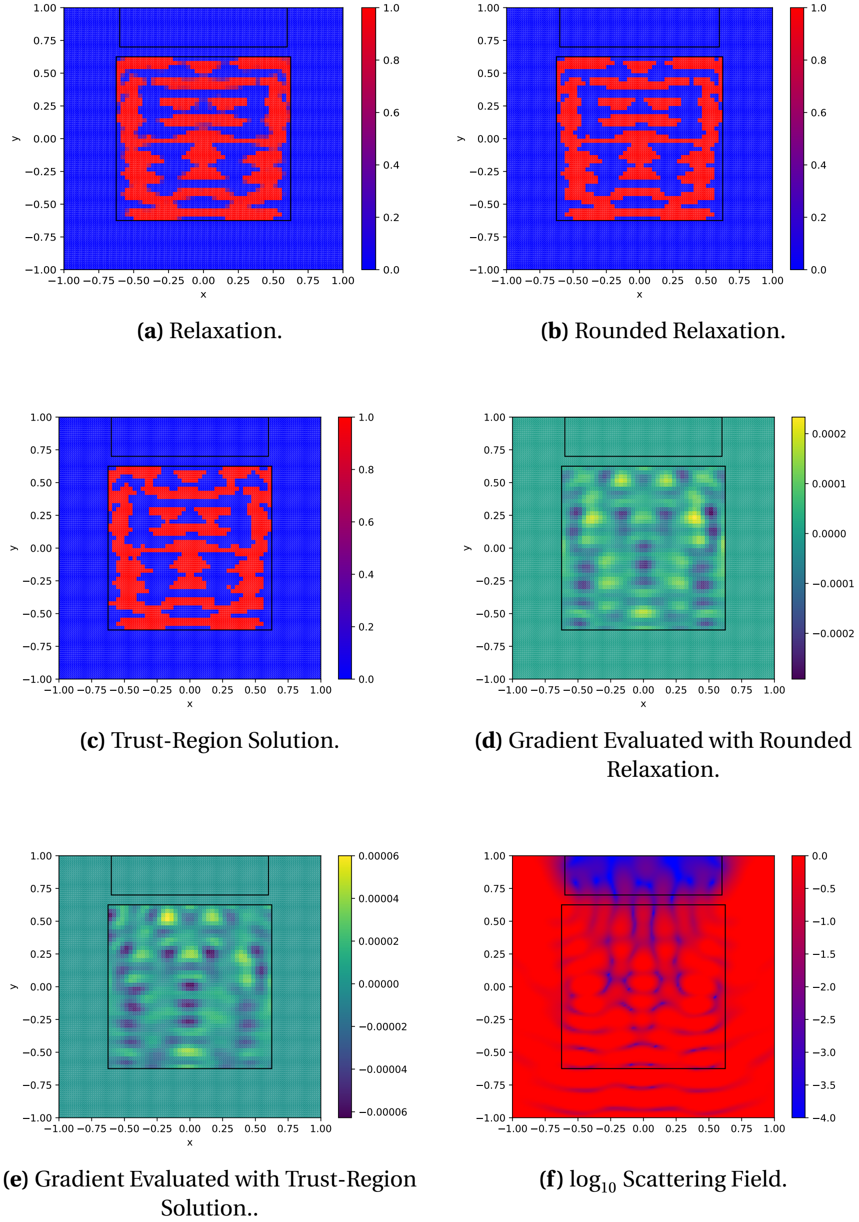
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 14** Rectangle 20x20: 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.

### .1.16 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

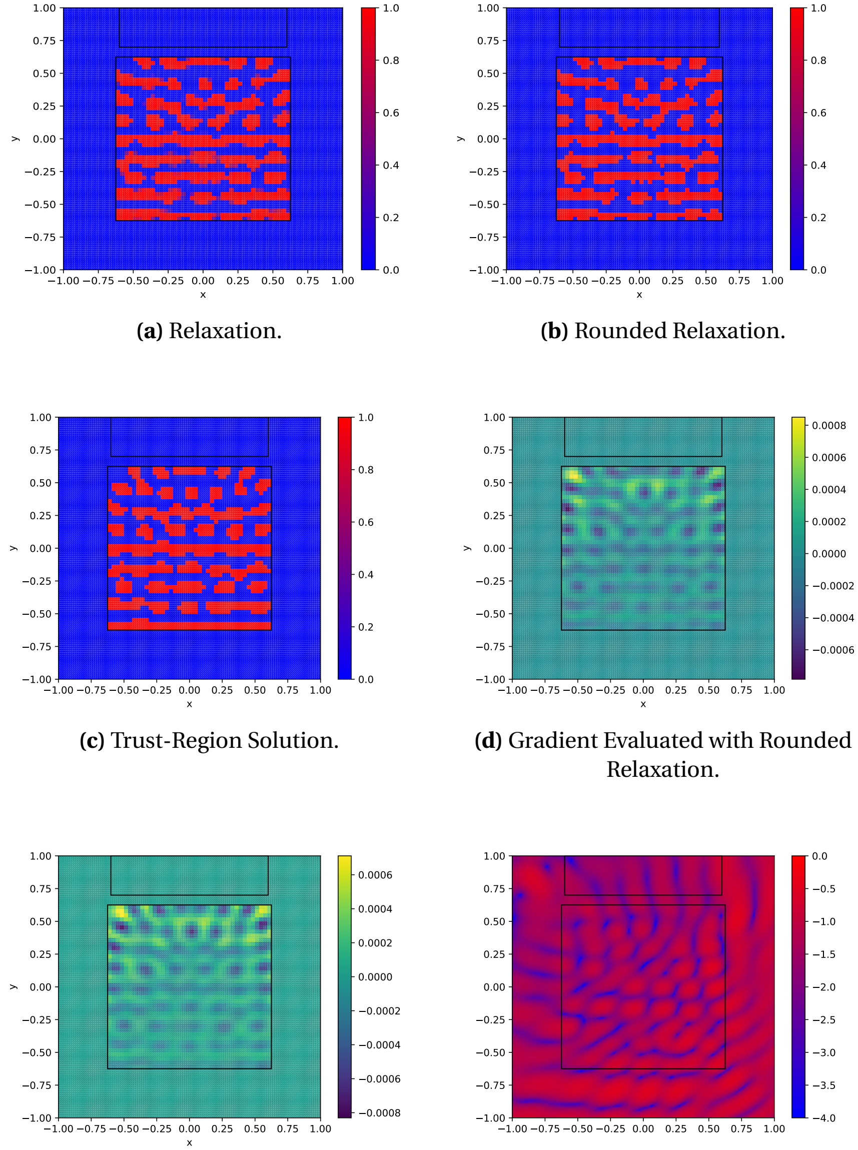
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 15** Rectangle 40x40: Nominal angle $\theta = \frac{\pi}{2}$.

### .1.17 Nominal angle $\theta = \frac{\pi}{2}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

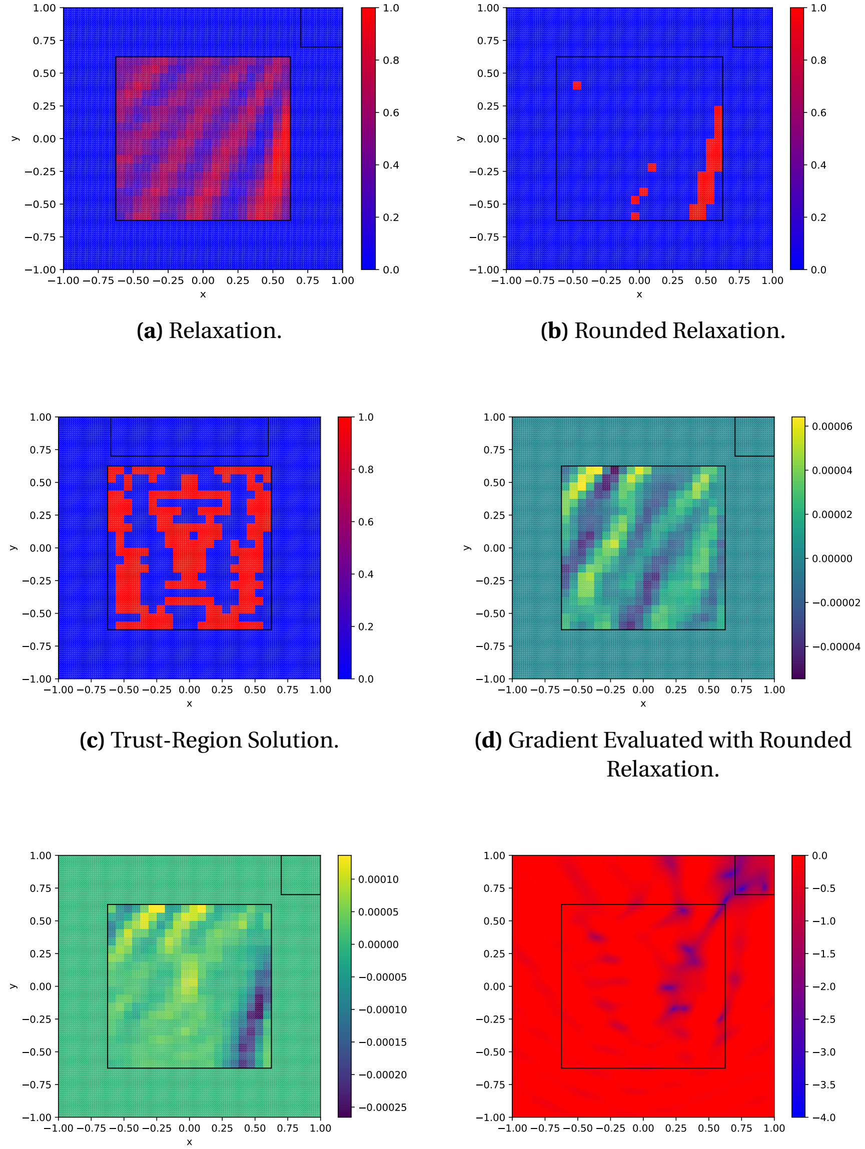
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 16** Rectangle 40x40: 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.

### .1.18 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

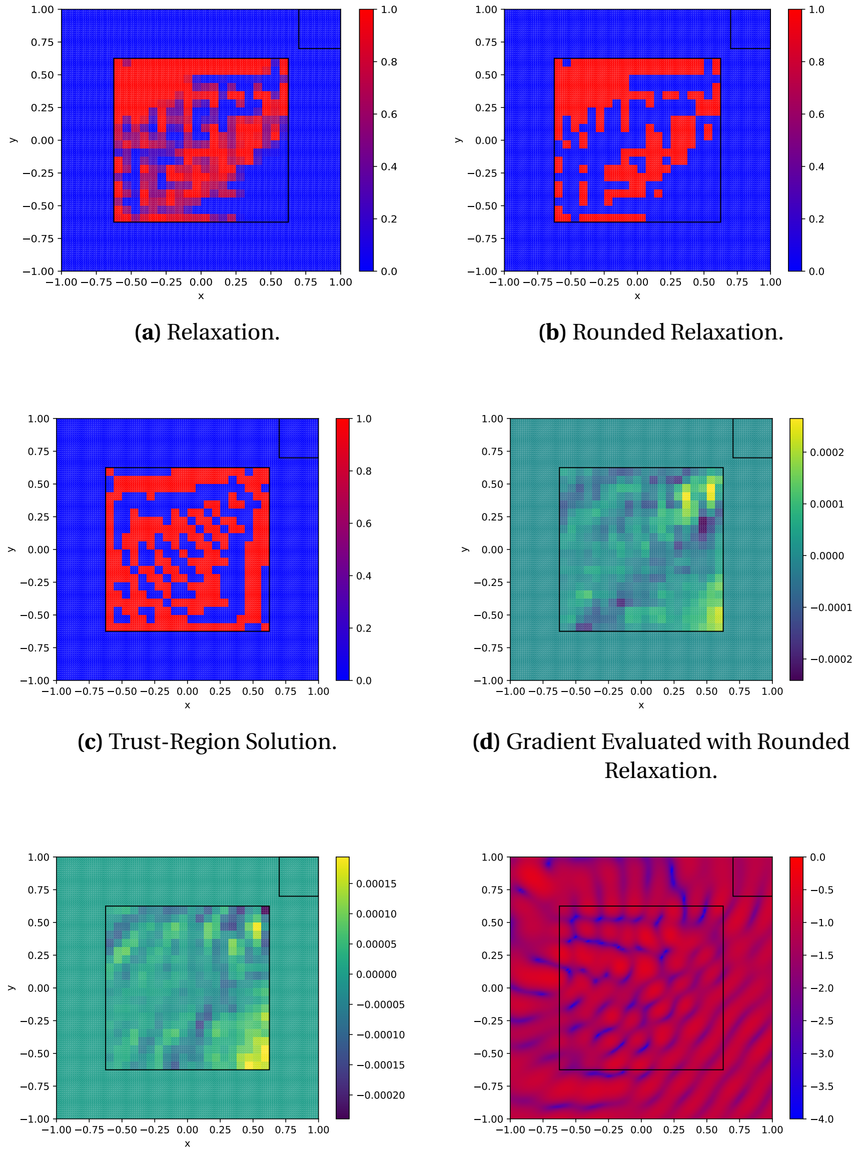
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 17** Square 20x20: Nominal angle $\theta = \frac{\pi}{2}$.

### .1.19 Nominal angle $\theta = \frac{\pi}{2}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

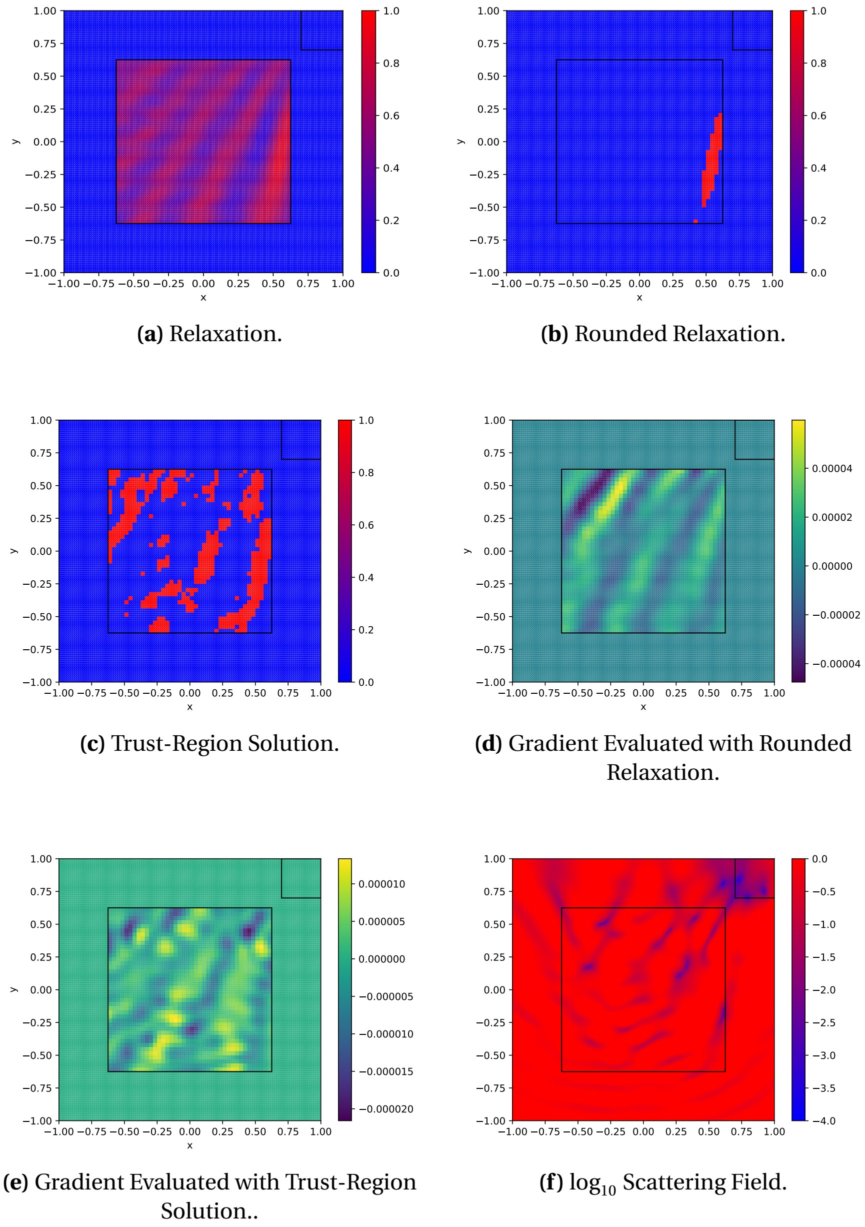
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 18** Square 20x20: 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.

### .1.20 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

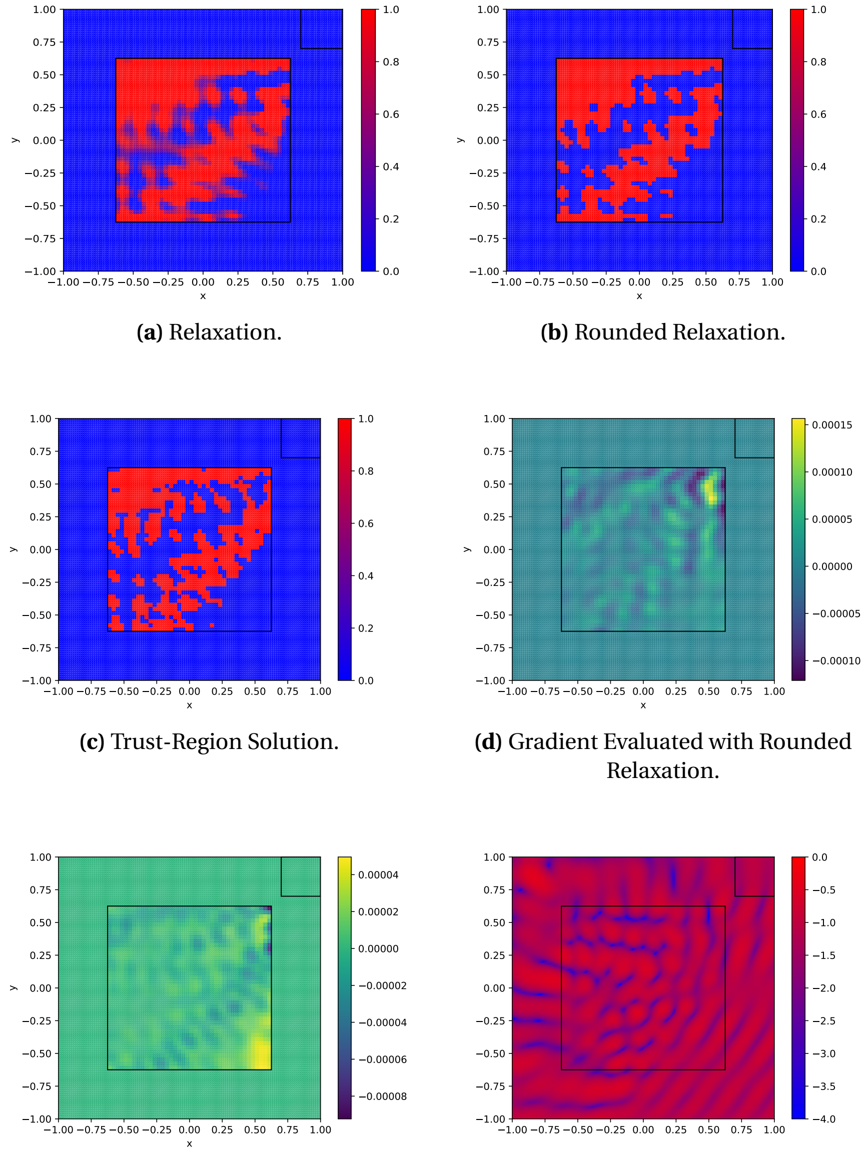
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 19** Square 40x40: Nominal angle $\theta = \frac{\pi}{2}$.

### .1.21 Nominal angle $\theta = \frac{\pi}{4}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

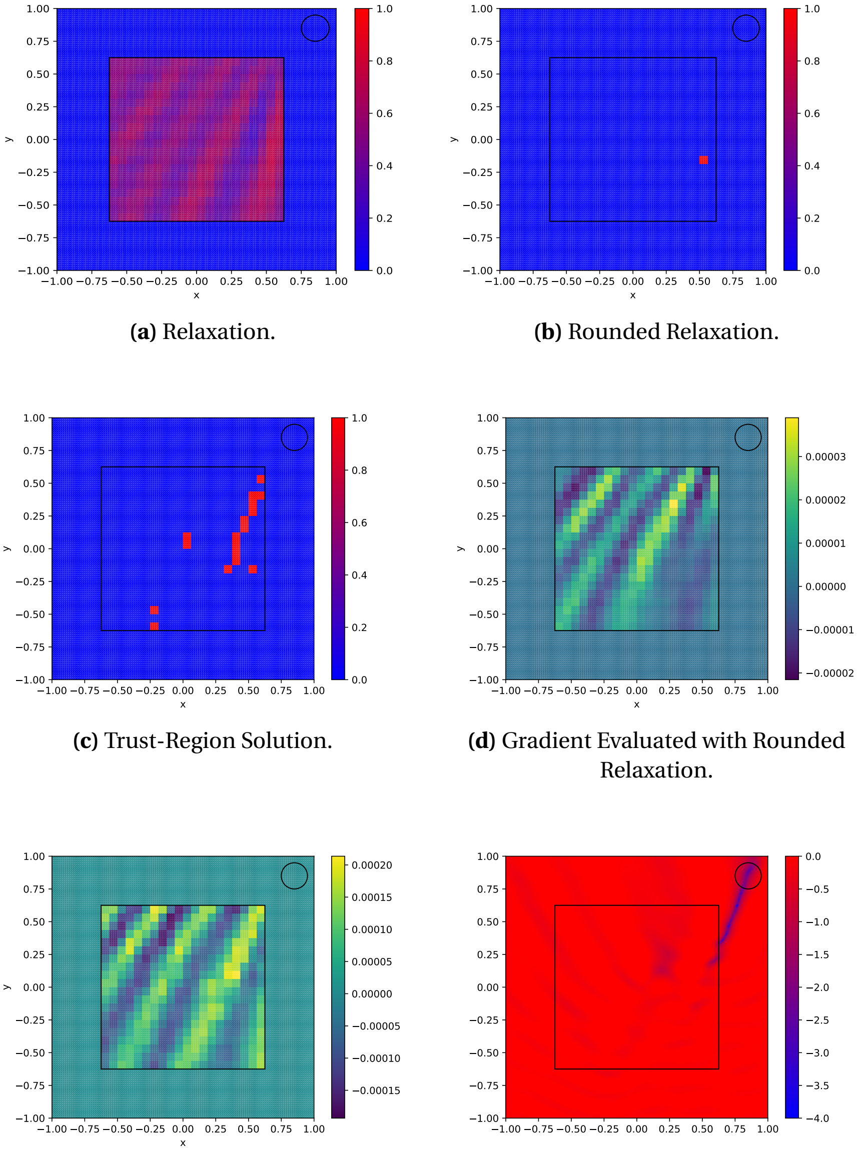
**(c)** Trust-Region Solution.

**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 20** Square 40x40: 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

**(d)** Gradient Evaluated with Rounded Relaxation.

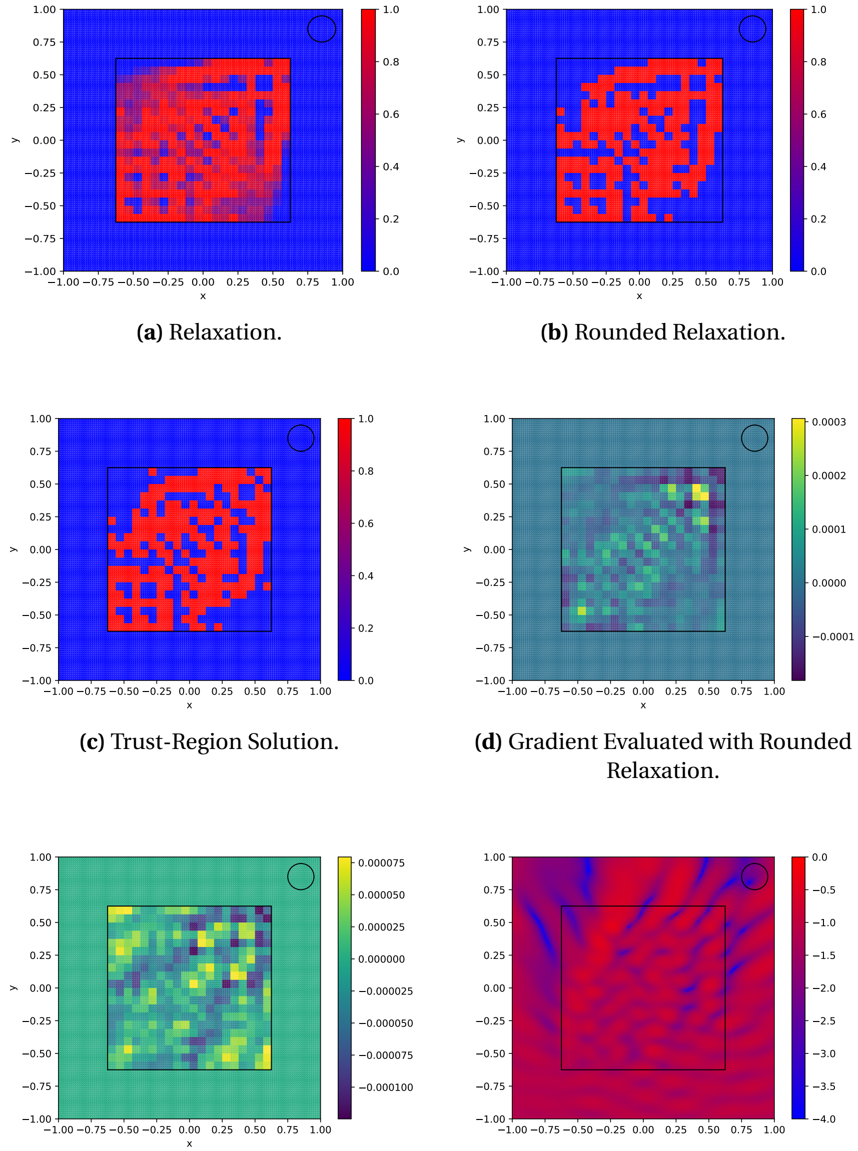
**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 21** Circle 20x20: Nominal angle $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.

### .1.22 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$

### .1.23 Nominal angle $\theta = \frac{\pi}{2}$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

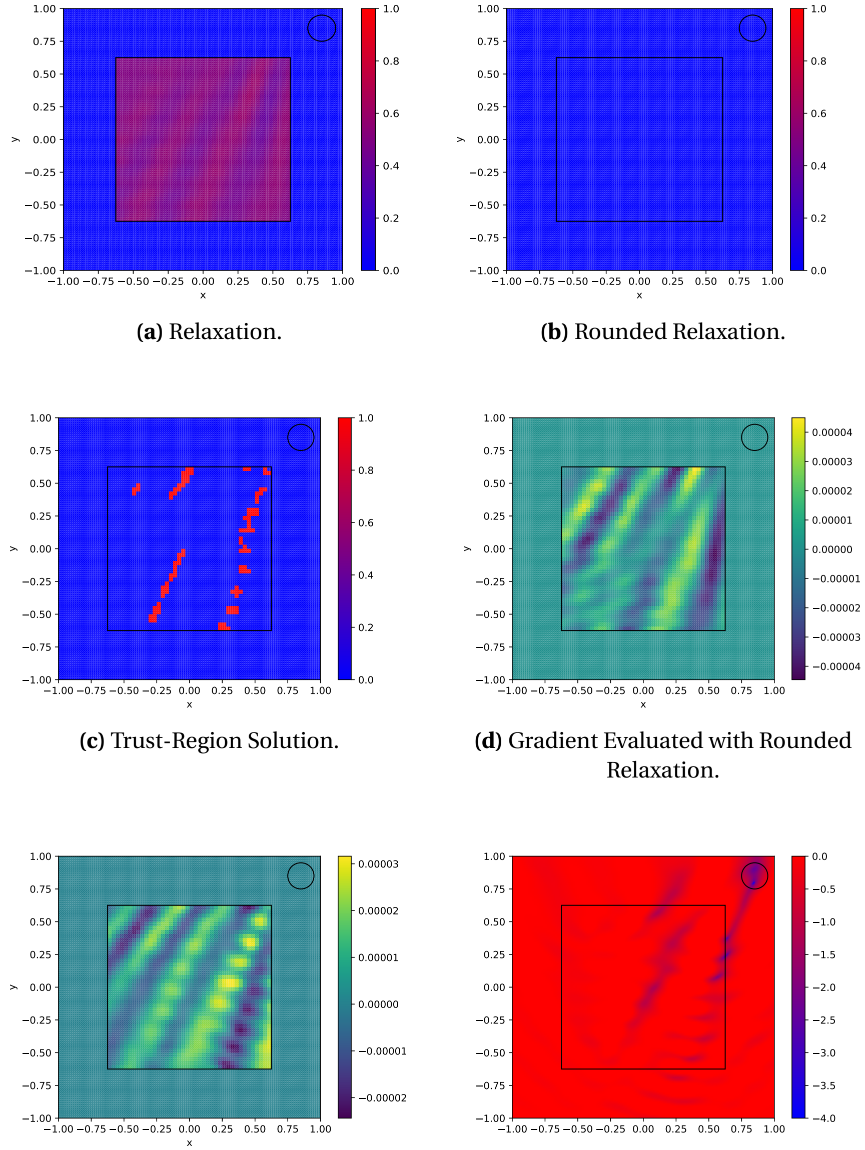
**(c)** Trust-Region Solution.

**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 22** Circle 20x20: 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

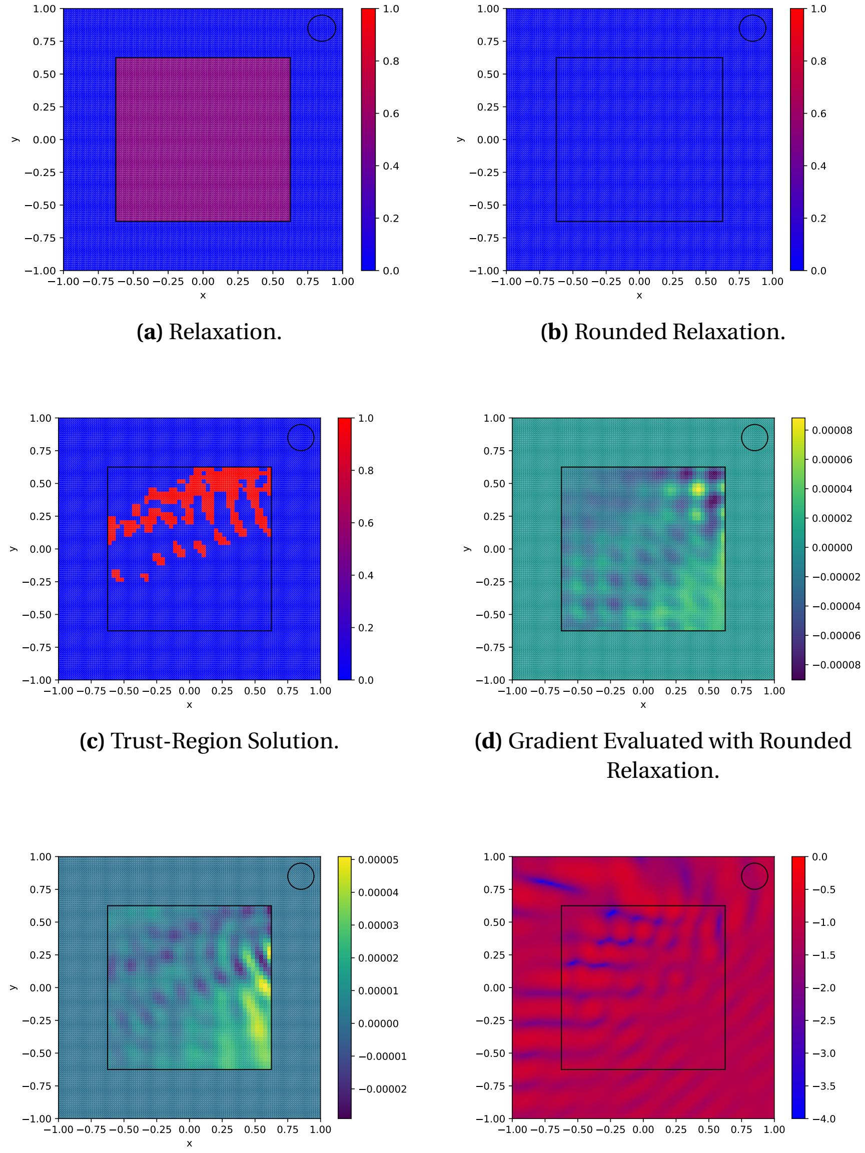
**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

**(f)** $\log_{10}$ Scattering Field.

**Figure 23** Circle 40x40: Nominal angle $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.

### .1.24 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$



**(a)** Relaxation.

**(b)** Rounded Relaxation.

**(c)** Trust-Region Solution.

**(d)** Gradient Evaluated with Rounded Relaxation.

**(e)** Gradient Evaluated with Trust-Region Solution..

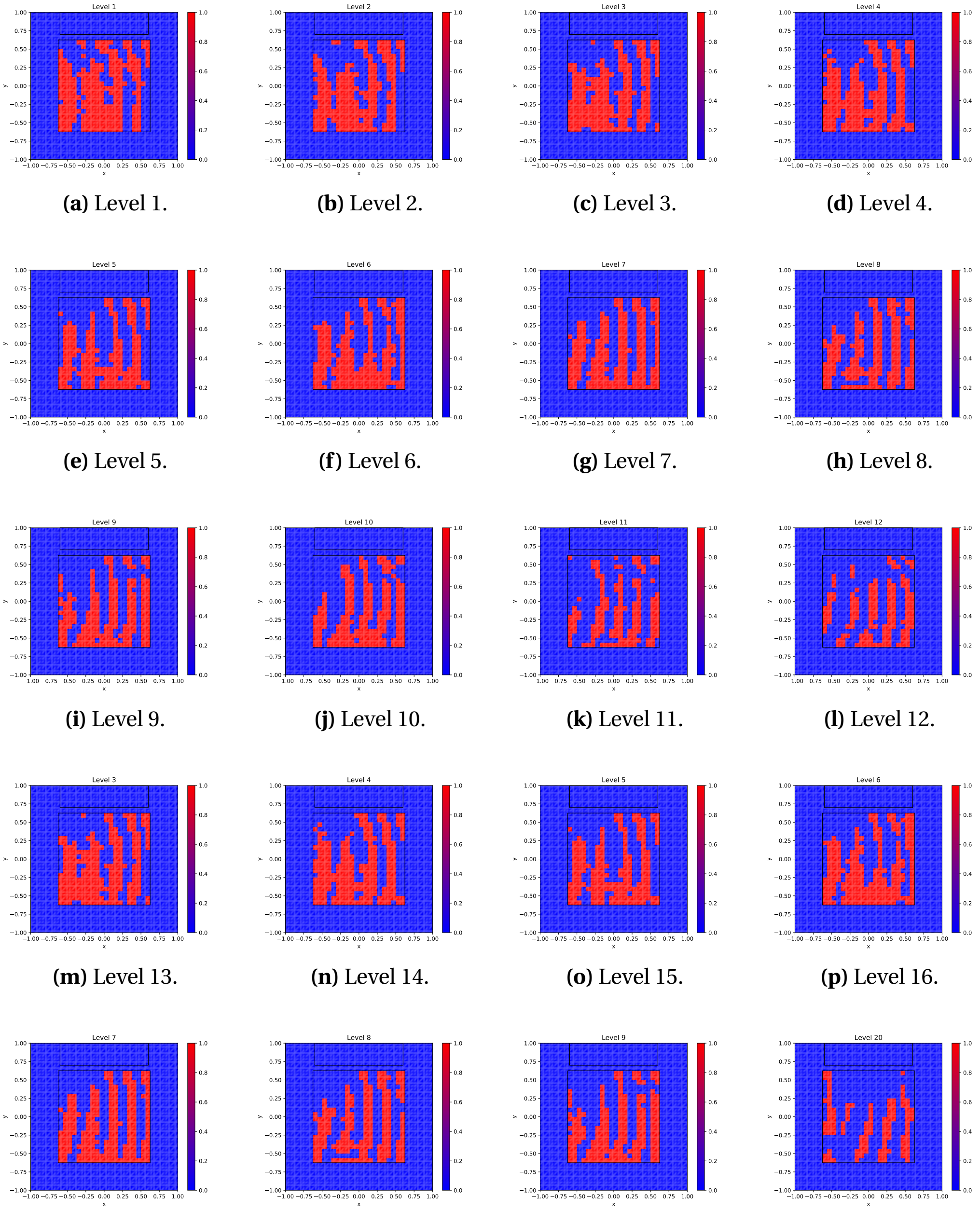
**(f)** $\log_{10}$ Scattering Field.

**Figure 24** Circle 40x40: 15 Angles $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$.

## .2 Simulations for 3D Electromagnetic Cloaking using a Single Material

This portion of the appendix shows the solutions yielded by our trust-region method in Section 6.1.2.

### .2.1 Rectangle: 20x20x20 discretization $\theta = \frac{\pi}{4}$, $\psi = \frac{\pi}{4}$



**(a)** Level 1. **(b)** Level 2. **(c)** Level 3. **(d)** Level 4.

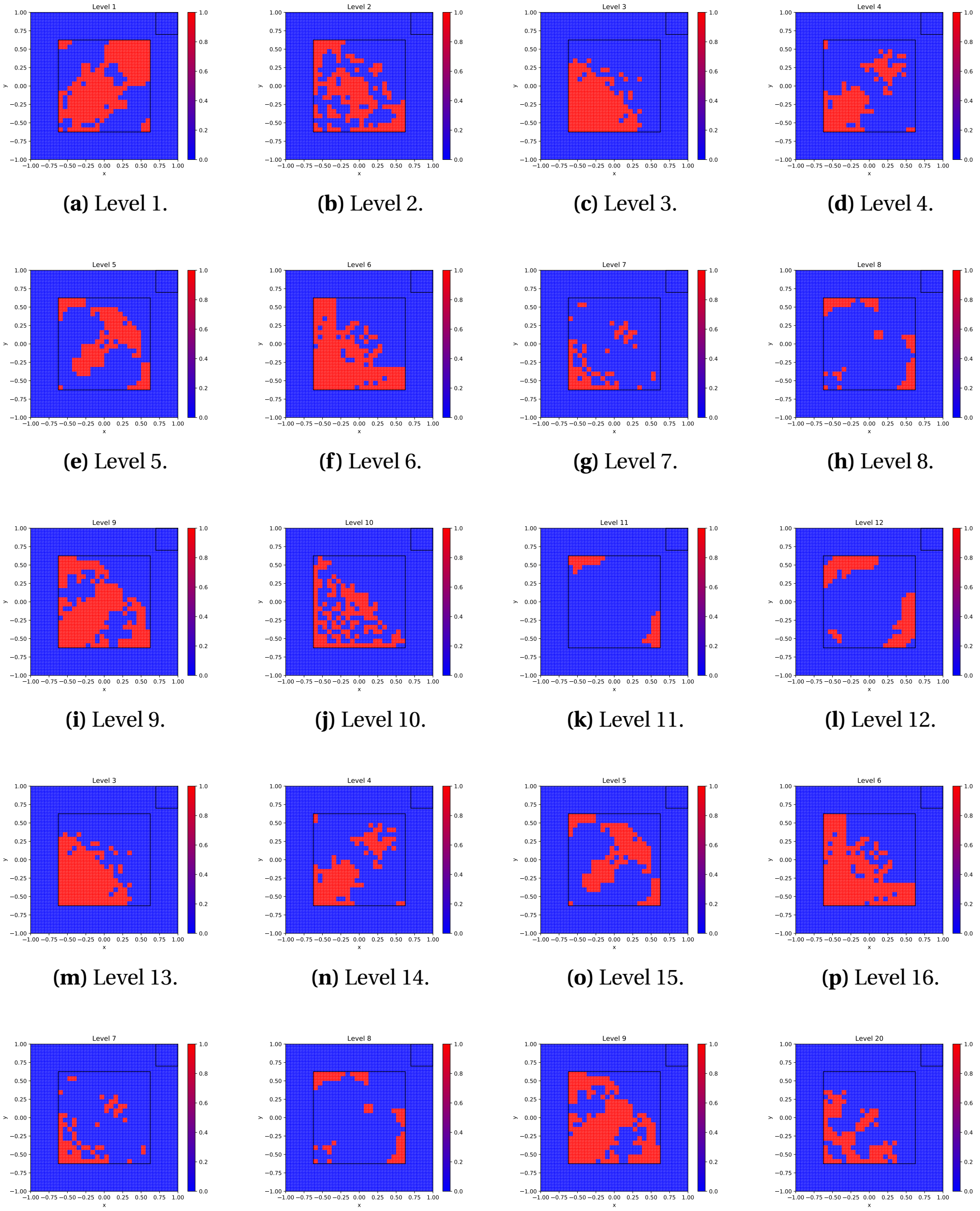
**(e)** Level 5. **(f)** Level 6. **(g)** Level 7. **(h)** Level 8.

**(i)** Level 9. **(j)** Level 10. **(k)** Level 11. **(l)** Level 12.

**(m)** Level 13. **(n)** Level 14. **(o)** Level 15. **(p)** Level 16.

**(q)** Level 17. **(r)** Level 18. **(s)** Level 19. **(t)** Level 20.

### .2.2 Square: 20x20x20 discretization $\theta = \frac{\pi}{4}, \psi = \frac{\pi}{4}$



**(a)** Level 1. **(b)** Level 2. **(c)** Level 3. **(d)** Level 4.

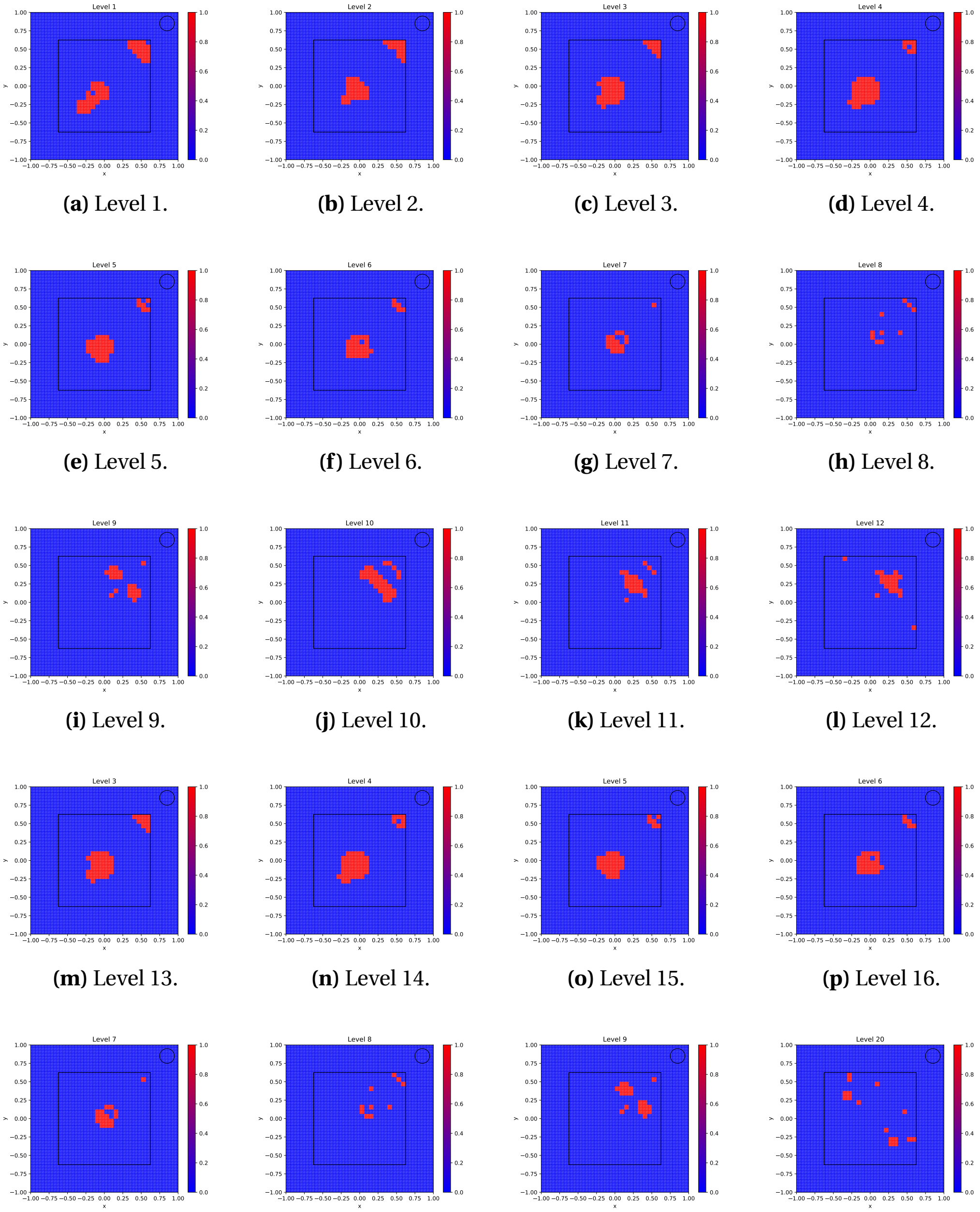
**(e)** Level 5. **(f)** Level 6. **(g)** Level 7. **(h)** Level 8.

**(i)** Level 9. **(j)** Level 10. **(k)** Level 11. **(l)** Level 12.

**(m)** Level 13. **(n)** Level 14. **(o)** Level 15. **(p)** Level 16.

**(q)** Level 17. **(r)** Level 18. **(s)** Level 19. **(t)** Level 20.

### .2.3 Circle: 20x20x20 discretization $\theta = \frac{\pi}{4}$, $\psi = \frac{\pi}{4}$



**(a)** Level 1.

**(b)** Level 2.

**(c)** Level 3.

**(d)** Level 4.

**(e)** Level 5.

**(f)** Level 6.

**(g)** Level 7.

**(h)** Level 8.

**(i)** Level 9.

**(j)** Level 10.

**(k)** Level 11.

**(l)** Level 12.

**(m)** Level 13.

**(n)** Level 14.

**(o)** Level 15.

**(p)** Level 16.

**(q)** Level 17.

**(r)** Level 18.

**(s)** Level 19.

**(t)** Level 20.