A Parallel Branch-and-Bound Algorithm for Thin-Film Optical Systems, with Application to Realizing a Broadband Omnidirectional Antireflection Coating for Silicon Solar Cells

by

Paul Azunre

B.A., Economics (2007), B.S., Engineering (2007), Swarthmore College,
M.Sc., Electrical Engineering and Computer Science, MIT (2009)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2014

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
August 29, 2014

Certified by . . . .
Marc A. Baldo
Professor of Electrical Engineering
Thesis Supervisor

Certified by . . . .
George C. Verghese
Professor of Electrical Engineering
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

A Parallel Branch-and-Bound Algorithm for Thin-Film Optical Systems, with Application to Realizing a Broadband Omnidirectional Antireflection Coating for Silicon Solar Cells

by

Paul Azunre

Submitted to the Department of Electrical Engineering and Computer Science
on August 29, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

For the class of nondispersive, nonabsorbing, multilayer thin-film optical systems, this thesis work develops a parallel branch-and-bound computational system on Amazon's EC2 platform, using the Taylor model mathematical/computational system due to Berz and Makino to construct tight rigorous bounds on the merit function on subsets of the search space (as required by a branch-and-bound algorithm). This represents the first, to the best of our knowledge, deterministic global optimization algorithm for this important class of problems, i.e., the first algorithm that can guarantee that a global solution to an optimization problem in this class has been found. For the particular problem of reducing reflection using multilayer systems, it is shown that a gradient index constraint on the solution can be exploited to significantly reduce the search space and thereby make the algorithm more practical. This optimization system is then used to design a broadband omnidirectional antireflection coating for silicon solar energy. The design is experimentally validated using RF sputtering, and shows performance that is competitive with existing solutions based on impractical sophisticated nano-deposition techniques, as well as the more practical but also more narrowly applicable solutions based on texturing. This makes it arguably the best practical solution to this important problem to date. In addition, this thesis develops a mathematical theory for cheaply (in the computational sense) and tightly bounding solutions to parametric weakly-coupled semilinear parabolic (reaction-diffusion) partial differential equation systems, as motivated by the design of tandem organic solar cell structures (which are governed by the drift-diffusion-Poisson system of equations). This represents the first theoretical foundation, to the best of our knowledge, to enable guaranteed global optimization of this important class of problems, which includes, but is broader, than many semiconductor design problems. A serial branch-and-bound algorithm implementation illustrates the applicability of the bounds on a pair of simple examples.

Thesis Supervisor: Marc A. Baldo
Title: Professor of Electrical Engineering

Thesis Supervisor: George C. Verghese
Title: Professor of Electrical Engineering

For my Mother

Acknowledgements

I am grateful to one of my thesis supervisors, Marc Baldo, for funding this project and the unyielding motivation despite the fact that most of the mathematical theoretical and computational aspects of this work lay outside of his area of expertise. The confidence shown in my ability to solve this problem independently is much appreciated. I am also grateful to Marc for getting me excited about the thin-film application area on which this dissertation hinges. It turned out to be a better domain for my birth as a device engineer than the semiconducting applications I was interested in when I started this work and which spawned the mathematical theoretical contribution of this work. Now that I have been born as a device engineering researcher in the domain of multilayer systems, I am appropriately prepared to tackle the semiconducting applications. I am grateful to my other thesis supervisor, George Verghese, for the emotional support, patience and intellectual engagement, could not h ave kept going without it. He truly is one of the biggest reasons why I successfully finished this epic journey, the voice in my head reminding me that I can do this and do it well, even when all other factors suggest otherwise. I am grateful to my thesis reader, Steven G. Johnson, for sharing his immense expertise on the subject in a series of meetings at crucial junctures of this project. Without this, the thesis could have never been completed. I am grateful to everyone in the Soft Semiconductor group (including, but not limited to, Nick Thompson, Dan Congreve, Markus Einzinger, Jean Anne Currivan, Tony Wu, Matthias Bhalke, Carmel Rotschild, Amador Velázquez, David Ciudad, Sumit Dutta and Phil Reusswig) for listening to many of my progress reports and providing feedback from the perspective of a device engineer even though my presentations were often far out of their area of interest and/or expertise. I am particularly grateful to Catherine Bourgeois, Phil Reusswig, Hiroshi Mendoza, Leonard Azeem, Anton Voinov, Johan de Borst, Inessa Lurye, Khendi White, Demba Ba, Abe Jmilli and David Kwabi for social support. I am also particularly grateful to Nick Thompson, Johanna Engel, Tim McClure, Dr. Shiahn Chen, Patrick Boisvert, Kurt Broderick, Dan Congreve and Joel Jean for extensive help on the experimental front of this work. I am

grateful to MIT for a number of reasons. First, I came here with approximately zero mathematics, real software development and experimental experience, now I have tons of all three. Perhaps more importantly, MIT taught me a number of life lessons which an ultra-liberal place like Swarthmore never could and which were extremely painful to digest and fully assimilate. Thanks to these lessons, I am in a much better position to take on the world. My undergraduate supervisor, Carr Everbach, is the one who set me on this research path and has remained in touch and a source of inspiration and support. I am truly grateful to him. I am indebted to special lady Diana Chiyangwa for pulling me out of the depths of PhD depression and setting me back on track. I can't thank her enough for listening to my incessant monologues about my work well beyond what any sane person can handle, and supporting me through the emotional ups and downs of completing this journey. I suspect she knows nearly as much about my research as myself at this point. My family (Mom, Rich, Lamisi, Florence, Grandmas, Madam Priscilla, Gideon and Gifty) has always been and remains a source of motivation to reach even greater heights, for which I am forever grateful. This material is based upon work supported as part of the Center for Excitonics, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award Number DE-SC0001088.

Contents

List of Figures

Chapter 1

Introduction

This work represents a modest set of theoretical and applied contributions to deterministic nonconvex programming algorithmics, more specifically the branch-and-bound framework, arising from my interest in developing an overview and an assessment to self of the state of the theory and applicability of such methods at the time of this writing. In particular, this interest arose from my sense of the potentially large impact such an overview could have given the prevalent opinion in modern engineering system design textbooks (and among practitioners) that engineering system design is an art. The potential of such an algorithm to locate a global solution to an engineering design problem with surety (to within a user-specified tolerance that is based on model accuracy) could be harnessed to convert engineering system design into a science. Of course, inverse problems, i.e., fitting models to data, is another place where such methods could be extremely useful. For instance, when doing hypothesis testing, it is crucial to make conclusions based on a true global optimum rather than some local optimum. When calibrating models to data while optimizing deposition conditions for device fabrication, the frustration and cost of being deceived by local minima is well-known (to me personally through the fabrication experience in this work). Below I outline my understanding of the contributions of this work.

This work takes the development of such an algorithm from the very beginning, i.e., the development of the underlying pure mathematics, through nontrivial algorithm/software engineering and to the very end, i.e., application to an important engineering system. For a class of problems for which the critical mathematical theoretical component, namely the construction of sufficiently cheap and tight bounds on any given subset of the search space, does not exist, it is created. The class of problems chosen for this purpose is the class of weakly-coupled semilinear parabolic partial differential equation systems (PDEs) (traditionally referred to as reaction-diffusion systems in the literature) as motivated by design and model calibration of both organic and inorganic semiconducting devices. More specifically, this work was motivated by my interest in optimizing tandem organic solar cells. The contribution of this mathematical theory is potentially broader than this class of problems, the novel approach to developing the theory being chosen to potentially serve as a solution program for other classes of differential equations by drawing from a large set of analogous results available through the method of lower and upper solutions. For a class of problems for which such an algorithm has not been engineered but the necessary mathematical and computational tools exist, namely nondispersive nonabsorbing multilayer thin-film optical systems, a parallel branch-and-bound computational system is developed on Amazon's EC2 platform using the Taylor model mathematical/computational system due to Berz and Makino. For the particular problem of reducing reflection using multilayer systems, it is shown that the special structure on the solution, namely the gradient index constraint, can be exploited to significantly reduce the search space and thereby make the algorithm much more practical (an argument that naturally generalizes to other important gradient index systems, including, but not limited to, gradient index optical fibers and gradient index lenses). The branch-and-bound computational system is then used to design an important engineering system, namely a broadband omnidirectional antireflection coating for silicon solar energy, the theoretical guarantee of global optimality on the obtained solution providing some interesting and arguably useful conclusions. This result is experimentally validated using RF sputtering, and is competitive with existing solutions that employ sophisticated nano-deposition techniques, as well as the more practical but also more narrowly applicable solutions based on texturing. This makes it arguably the best practical solution to this problem to date.

The rest of this work is organized as follows. Minimal relevant background is presented at the beginning of each chapter. The background is kept minimal and high quality sources are cited for the interested reader so as to focus the write-up on the novel contributions of this work rather than prior art. The chapter that follows describes the parallel branch-and-bound algorithm for nondispersive nonabsorbing multilayer systems that was developed using the Taylor Arithmetic bounding framework due to Berz and Makino on Amazon's EC2 platform. In the same chapter it is demonstrated how the gradient index constraint on the solution to the antireflection design problem can be exploited to reduce the search space and make the algorithm more practical from a computational perspective. The chapter that follows describes the broadband omnidirectional antireflection coating for silicon solar cells that was experimentally demonstrated based on the theory that came out of the branch-and-bound work, arguably the best practical solution to that problem to date. How does one develop a theoretical bounding foundation for a class of problems for which one does not exist? This question is addressed next for the class of semilinear parabolic PDEs, as motivated by my interest in optimizing tandem organic solar cells (which are governed by the drift-diffusion-Poisson parabolic system of equations). The thesis concludes with a chapter briefly outlining future directions, many of which are already in progress.

Chapter 2

Parallel Branch-and-Bound Optimization of Multilayer Optical Systems, and Application to Broadband Omnidirectional Antireflection Coating Design for Silicon

2.1 Overview

In this chapter, a deterministic (alternatively verified or rigorous) global optimization algorithm for thin-film optical filters is developed for a grid computing parallel environment. This algorithm enables locating a global solution to a thin-film optical filter optimization problem with surety (to within a user-specified tolerance, which is governed by model accuracy). More specifically, the algorithm is a parallel branch-and-bound method that requires a lower bound for the cost function (some function of the front-interface reflectance computed using the transfer-matrix model) on subintervals of the search space. An interval bound is constructed using Taylor arithmetic. Moreover, the algorithm is characterized by best-bound subinterval and relative-width bisection direction selection rules, midpoint evaluation for incumbent search and a hybrid scheduling approach in the parallelization process (static scheduling at the level of servers and dynamic scheduling via a single work queue within each server). An implementation of the algorithm on Amazon's EC2 parallel computing platform is applied to two antireflection coating design problems. To the best of our knowledge, this is the first demonstration that sufficiently interesting thin-film optical filters can be optimized in a verified fashion at a reasonable cost. Thus, contrary to the opinion of some designers, this chapter shows that it may be feasible to find with surety a global solution to a multilayer filter design problem. Said somewhat differently, this chapter shows that the design of multilayer filters, an activity popularly regarded as an art, can be made into a science using an algorithm of the class described. It is hoped that the demonstration in this chapter will stimulate further research into more efficient algorithms of this kind.

2.2 Introduction

Multilayer filters are an integral component of modern optical systems. They function to determine the spectral composition and intensity of light reflected and/or transmitted by an optical system. A thin film is generally considered to vary between a fraction of a nanometer and a couple micrometers in thickness [20]. Thin films were first discovered in the late 1600s by Robert Hooke in the phenomenon known as Newton's rings (these were later named after Isaac Newton who was the first to provide an analysis). The first antireflection coatings were made by Josef von Fraunhofer in 1817. He noticed that corroded glasses reflected less light, making his devices the precursor to today's surface texturing approaches to reducing reflection. In 1880 Lord Rayleigh was the first to recognize the potential of gradient index antireflection coatings, by making an analogy to the way sunlight travels through the atmosphere. This is captured by his statement that "No one would expect a ray of light to undergo reflection in passing through the earth's atmosphere as a consequence of the gradual change of density with elevation." [35] The modern era of thin film manufacturing began in the 1930s with the invention of reliable vapor deposition techniques, such as electron beam deposition and sputter deposition.

A schematic representation of a multilayer filter is shown in Figure 2-1. In the figure, $\mathcal{R}$ represents the front-interface reflectance (ratio of reflected to incident intensity at any given incident wavelength λ, incident angle θ and polarization s or p) while d_k and η_k respectively denote the physical thickness and the complex refractive index of the k^{th} layer. The complex refractive index of the k^{th} layer is explicitly written as

$$\eta_k = n_k + j\kappa_k, \tag{2.1}$$

with n_k and κ_k respectively denoting the phase speed and extinction coefficient of the k^{th} layer. In general n_k is wavelength dependent, but all problems considered in this work are nondispersive so that n_k is not a function of wavelength. Moreover, all examples considered feature approximately nonabsorbing materials, so that κ_k is $0\,\forall k$.

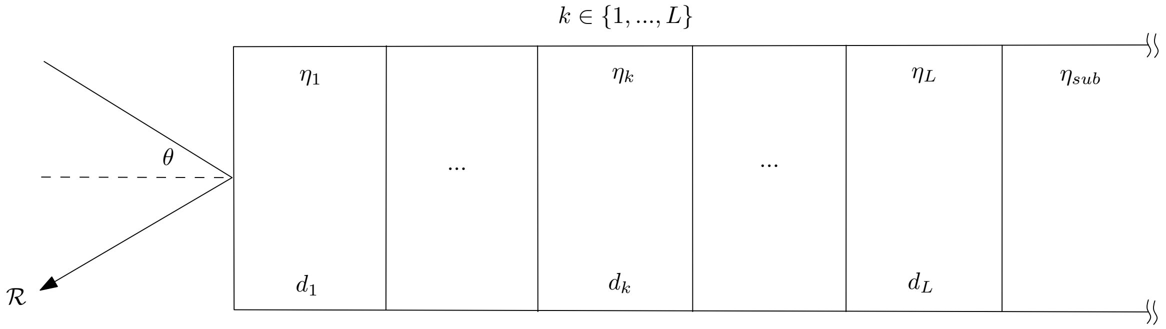

Figure 2-1: *A schematic representation of a thin-film optical filter.*

The task of optimizing a thin-film optical filter generally involves finding a system configuration that most closely approximates the desired response (reflectance over the relevant bandwidth, incident angle range and specified polarization). For a fixed number of layers L, this task may be posed as

$$\mathbf{p}_{opt} = \underset{\mathbf{p}\in P}{\arg\min} \left(O \left(\left\{ \left\{ \mathcal{R}_p \left(\mathbf{p}, \lambda_i, \theta_j \right), \mathcal{R}_s \left(\mathbf{p}, \lambda_i, \theta_j \right) \right\}_{i=1}^{m} \right\}_{j=1}^{n}, L \right) \right). \tag{2.2}$$

Here, the design variable (or parameter) vector $\mathbf{p}$ is specified by some subset of $\{d_k, \eta_k\}_{k=1}^{L}$ (i.e., by some subset of the thicknesses and complex refractive indices of the layers in the stack) and the subscripts of $\mathcal{R}$ denote the polarization of incident light. The cost (alternatively objective or merit) function O measures how closely a given configuration $\mathbf{p}$ approximates the ideal response and perhaps penalizes more complex designs. It is usually a numerical approximation for a definite integral over specified wavelength and incident angle ranges. The objective function O should be increasing in the number of layers L to reflect the fact that a simpler system is preferable (since it is less expensive to manufacture, for instance, but also since it is less prone to manufacturing errors and failure due to high tensile stress). If O does not explicitly penalize a more complex system through dependence on L, one can do better (i.e., achieve a lower global solution $O\left(\left\{\left\{\mathcal{R}_p\left(\mathbf{p}_{opt}, \lambda_i, \theta_j\right), \mathcal{R}_s\left(\mathbf{p}_{opt}, \lambda_i, \theta_j\right)\right\}_{i=1}^{m}\right\}_{j=1}^{n}\right)$) by increasing the number of degrees of freedom through increasing L (and thereby the dimension of the search space P). When using the algorithm outlined in this chapter to address this case, one should start with as few layers as is reasonable and repeat the optimization process for incrementally more layers until an acceptable performance is achieved or until adding more layers does not appreciably improve performance (the issue of diminishing returns with increasing number of layers is well-known). Hence, we define the global solution to be the solution corresponding to the number of layers immediately after diminishing returns set in. Technically, this is a Mixed Integer Nonlinear Program (MINLP) since O is nonlinear in $\mathbf{p}$ and the entries of $\mathbf{p}$ may be continuous (i.e., take values on intervals) or be discrete/integers (i.e., be restricted to a finite number of choices, as in the case where some library of materials is available in the laboratory). However, in this work attention is restricted to continuous problems for simplicity.

Many methods for solving the problem (2.2) have been used. Design methods can be broken into two broad classes, refinement and synthesis. Refinement methods are basically local optimization techniques that improve an initial guess by a designer. Everything depends on the intuition of the designer since it drives the initial guess and thereby the quality of the final solution. Synthesis techniques create a design to some specification using some arbitrary procedure, no initial guess is required (some people refer to these as "black magic", rightfully so, as it is often not clear why one would expect such a procedure to produce good results). Workers in this field tend to call mathematical optimization techniques "digital design methods". A comparative analysis of popular algorithms is presented in [7]. The problem is well-known to be nonconvex on P (nonconvexity is demonstrated in the context of the first numerical example), motivating the use of global optimization algorithms to avoid suboptimal local minima being adopted as the solution. Global optimization algorithms are divided into two general classes, namely stochastic (e.g., genetic algorithms, simulated annealing, variants of multistart optimization) and deterministic (e.g., branch-and-bound, inner approximation, outer approximation, sums-of-squares). Stochastic global optimization algorithms have been used extensively to optimize thin-film optical filters (see [23] for an example application of genetic algorithms, [9] for an example application of Multi Level Single Linkage, abbreviated as MLSL, an intelligent variant of the multistart algorithm which is also used here to benchmark our novel algorithm). However, convergence of this class of algorithms to a global solution is only guaranteed asymptotically but not in practice (a maximum number of iterations or function evaluations is set, or the user decides that the current solution is "good enough" and terminates execution). Thus, although these algorithms can do well in avoiding suboptimal local minima, one can never know for sure whether a global solution has been found in practice.

Problem-specific heuristics, such as the gradient index profile concept for designing antireflection coatings (the problem we will be interested in, in this work) are also available and can do quite well. Based on Lord Rayleigh's analogy, by varying the refractive index gradually from that of the free medium to that of the substrate, maximum transmission can be attained. In the context of inhomogeneous layers, theory has been developed to find an index profile which is optimal [35]. Inhomogeneous layers are unfortunately not practical so in practice discrete layers are used to approximate these profiles. Using oblique angle deposition of SiO_2 and co-sputtering of SiO_2 and TiO_2, it is possible to vary refractive indices continuously in the interval $[1.09, 2.60]$ [19], striking very closely any index needed to approximate a selected profile. Oblique angle deposition varies the porosity of the material and thereby varies its index. Co-sputtering varies the index between the indices of the two materials being co-sputtered by varying relative deposition rates.

In this work, minimizing average reflection from silicon over the incident angle range $[0, 60]$ degrees and the wavelength range $[400, 1600]$ nm is the problem of particular interest. Untreated silicon normal incidence reflection is greater than 30%. The goal here is to search for a "perfect" antireflection coating for a silicon solar cell, one that can transmit all/most of the light that is incident on it to maximize the efficiency of the solar cell in its relevant wavelength and incident angle range of operation. Although silicon absorption is approximately 0 above $1100nm$, reducing reflection there may still be important for upconversion and silicon photonic applications. There is a good solution to this problem in terms of average reflection (it seems to be the lowest value reported in the literature for this problem) that achieves an average reflection of 3.79%, a seven layer design that approximates the quintic profile

$$n\left(z\right) = n_{\min} + \left(n_{\max} - n_{\min}\right)\left(10z^3 - 15z^4 + 6z^5\right). \tag{2.3}$$

This result, whose SEM image from the original reference [19] is shown in Figure 2-2, although performs well, is not a practical solution because the porous films are not weather resistant (when it rains, water gets into the pores and modifies the desired properties of the material). The slanted (oblique angle deposited) nanorods of SiO_2

can be seen in the first two layers of that image. Arguably the most widely applicable practical solution in use today on a large scale is a single layer quarter-wave antireflection coating, typically made of silicon nitride (Si_3N_4), achieving normal incidence reflection of approximately 15%. Surface texturing solutions also exist, minimizing reflections by "roughening" the surface of the substrate (by etching pyramids into it, for instance), thereby increasing the chance that reflected light can be reabsorbed. These typically quote figures of around 2% at normal incidence (see, for instance, [45] for texturing based on subwavelength surface Mie resonating nanocylinders etched into the silicon surface). We note, briefly, that figures this low also include a single quarter-wave layer, typically made of silicon nitride, on top of the textured surface. The texturing approach, unfortunately, has some disadvantages. For instance, it requires corrosive chemicals, such as the notoriously toxic hydrofluoric acid, during etching of the silicon surface (even if we can do this, should we, given the inherent environmental costs and workplace hazards?). Perhaps more importantly, texturing works well only for a small fraction of the solar market, i.e., monocrystalline silicon such as the one manufactured by the popular company SunPower. It is well-known that polycrystalline silicon, which forms the majority of the solar cell industry at the time of this writing, doesn't texture as easily. For these reasons, we think a better solution is needed, which motivated this work. We believe the theory developed here points to such a solution.

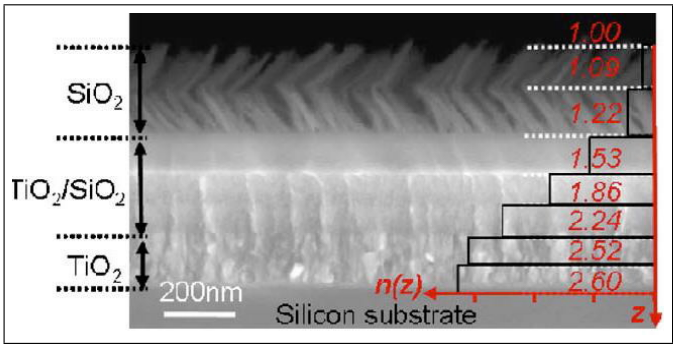

Figure 2-2: *Presently best broadband omnidirectional antireflection coating for silicon solar energy, SEM image.*

Unlike any of the previously outlined algorithms, deterministic global optimization algorithms can provide a guarantee that a global solution to an optimization problem has been found to within a user-specified tolerance. In this chapter, a deterministic (alternatively verified or rigorous) global optimization algorithm for thin-film optical filters is developed. More specifically, the algorithm is a parallel branch-and-bound method that requires a lower bound for the cost function (some function of the front-interface reflectance computed using the transfer-matrix model) on subintervals of the search space. The general framework of branch-and-bound was first proposed in 1960 by Land and Doig [15]. In our algorithm, an interval bound is constructed using Taylor arithmetic. Moreover, the algorithm is characterized by best-bound subinterval and relative-width bisection direction selection rules, midpoint evaluation for incumbent search and a hybrid scheduling approach in the parallelization process (static scheduling at the level of servers and dynamic scheduling via a single work queue within each server). An implementation of the algorithm on Amazon's EC2 parallel computing platform is applied to two antireflection design problems.

To the best of our knowledge, this is the first time that a problem within the class of thin-film optical filters has been optimized in a verified fashion. Thus, contrary to the opinion of some designers, this chapter shows that it may be feasible to find with surety a global solution to a multilayer filter design problem. Said differently, this chapter shows that the design of multilayer filters, an activity popularly regarded as an art (see the Foreword in [20] for an example of a recurring view in standard references on design being as much an art as a science, due to the necessity for a good initial guess for refinement by local optimization algorithms and the arbitrariness of most synthesis methods), can be made into a science using an algorithm of the class described. This chapter can also be viewed to provide an overview and an assessment of the current state of knowledge within the evolving field of global optimization for this important class of problems, an overview that thin-film engineers could find useful.

The rest of the chapter is organized as follows. The algorithm and the bounding procedure are described in Section 2.3. Section 2.4 presents the numerical examples.

The numerical results are discussed and the chapter is concluded in Section 2.5.

2.3 Deterministic Global Optimization Algorithm

In this section, the branch-and-bound algorithm is presented. First, the parallelized method is outlined. Then, a procedure for bounding functions of the front-interface reflectance (assuming all parameters are interval-valued) on subintervals of P, as required by the branch-and-bound method, is described. We briefly mention how the bounding procedure can be used for uncertainty analysis, an approach superior to other approaches which have been used for uncertainty analysis of multilayer filters in the literature. For a thorough exposition of deterministic global optimization theory the interested reader is referred to [15]. Simple examples used to illustrate concepts in this section are well-known in the deterministic global optimization literature.

2.3.1 Parallel Branch-and-Bound Method

Choose the search space P to enclose all physically realizable system configurations or choose a smaller search space of interest (to enclose all of the best-known historical designs, for instance). If good prior solution information is available (from a database of notable historical designs, or as an output of a stochastic global optimization method, for instance) initiate the candidate global solution (referred to as the *incumbent solution*) to this. This is not done in this work though, since we are interested in the performance of our algorithm without prior knowledge. Create a stack on each running process in the parallel environment of choice (in this work, we use Amazon's EC2 platform - other services, notably Microsoft's Windows Azure, and ProfitBricks are also available, more tightly coupled supercomputers such as IBM's Blue Gene Q are also common in academic environments and large companies) to hold subintervals/partitions of the search space along with corresponding merit lower bound, merit upper bound and level values. The level of an interval is defined as the number of times the original space was bisected to arrive at it. The concept of level is illustrated for a unidimensional search space in Figure 2-3. The procedure we use for computing the merit lower bound for each partition is described in the next subsection. We consider the simplest procedure for computing the merit upper bound for each partition (computation of upper bounds could also be referred to as *incumbent search* since the minimum upper bound observed up to any iteration corresponds to the incumbent solution). This is to evaluate the merit at the midpoint of each partition. Choose the absolute convergence tolerance ε to reflect practical considerations (beyond which point does optimizing the merit make no practical significance?). For all the results presented in this work, ε is set to 0.001 (optimizing problems considered in this work beyond this point, or 0.1%, amounts to optimizing within modeling error, as demonstrated in the context of the second numerical example).

Statically schedule a region of the search space to each server. On each server, discretize the assigned partition and assign each running process one of the resulting subintervals (this process of assigning chunks of the search space to each process is usually referred to as *ramp up* in the branch-and-bound literature). In our algorithm, this is done by running a serial version of the branch-and-bound method until the number of partitions becomes equal to the square of the number of running processes on each server times the number of servers (the reason for this particular number will become clear in the remainder of the description of the algorithm). Then, each process on each server is assigned the number of running processes on each server partitions at iteration one.

Once this initial static assignment is performed, each server works independently (the reason why this is necessary is clarified later on in this write-up). At every iteration (with the exception of the first, where subintervals from the initial discretization are assigned to a process and all relevant quantities corresponding to that interval are computed, as described in the previous paragraph) select a subinterval for each process from the stacks of the processes on that given server (we henceforth refer to these as the *global stack* for that server) to bisect along a parameter into 2 new subintervals. We considered three rules for selecting subintervals for bisection. The first involves picking the first subintervals on the global stack corresponding to the least remaining lower bound (LRLB) of the partitions on the global stack. This rule is referred to as the *best-bound* rule in the deterministic global optimization literature. The second involves picking the first subintervals on the global stack corresponding to the least remaining upper bound of the partitions on the global stack. This rule is referred to as the *best-estimate* rule in the literature. The third rule involves picking the first subintervals on the global stack corresponding to the least level of the subintervals on the global stack. This rule is referred to as the *breadth-first* rule in the literature. Call the selected intervals $\{I_i\}_{i=1}^{NPROCESSES} \subset P$. Here, $NPROCESSES$ is the number of processes on any given server (in our case, this number will be 16). The selected intervals we refer to as *active intervals* for any given iteration. They are selected as follows.

On any given process (on any given server), select $NPROCESSES$ candidate active intervals (in case all active intervals for the next iteration on that server are on the present process) according to the active selection rule. Communicate these across all processes on that server for a total of $NPROCESSES^2$ candidate active intervals. Select the best $NPROCESSES$ candidates to be the active intervals for the next iteration according to the active selection rule. This routine is the reason why we ramp up to the square of the number of running processes (i.e., $NPROCESSES^2$) on each server times the number of servers (so that there are $NPROCESSES$ intervals on each process at the first iteration, to serve as sufficient number of candidate active intervals for the second iteration).

We considered two rules for selecting which direction of the selected subintervals to bisect on, on every process, on every server, at every iteration. The first involves bisecting on a uniformly randomly selected direction and the second involves selecting the direction $i \in \{1, 2, ..., 2L-1, 2L\}$ that maximizes the quantity

$$\frac{w\left(I_i\right)}{w\left(P_i\right)}. \tag{2.4}$$

This quantity we call the *relative-width*. The width w is defined as the difference between the upper and lower bounds of the corresponding interval and the normalization by the width of the original space handles dimension differences between different parameters (always normalizing possible measures into the interval [0, 1] regardless of dimension). We found empirically that the best-bound rule coupled to the relative-width rule worked best (we choose to leave out details of this comparison, for brevity). The best-bound rule is motivated by the desire to increase least lower bound value as fast as possible since this is critical to convergence (another motivation might be the potential of finding a good candidate solution on such a subinterval) while the relative-width rule is motivated by the desire to divide things up in a uniform way. The bisection of a simple nonconvex univariate objective is illustrated in Figure 2-4 to aid visualization.

Place each of the two resulting subintervals (with the exception of the first iteration, when a subinterval from the initial discretization is processed instead) on the stack local to the process where bisection occurred along with computed merit lower bound and merit upper bound information. If either upper bound on the new subintervals is better (i.e., lower) than the incumbent, update the incumbent accordingly. Then, communicate local incumbent information across all processes on that server and extract the global incumbent information on each process. Execute pruning of partitions from the stack, where partitions with lower bound greater than the global least upper bound (i.e., the incumbent) are eliminated from the stack (the reason being that the global solution clearly cannot exist on such a partition). This exclusion step alleviates the so-called "curse of dimensionality" (i.e., the exponential explosion of computational cost with increasing problem complexity). This step is also the reason why branch-and-bound is more efficient than what is sometimes called "scanning" (alternatively "brute-force" search or "grid search"), a nonrigorous global optimization approach involving fine discretization of the search space into a grid along which the merit is evaluated and the best merit value taken as the solution (if all the parameters were discrete, evaluating every single point on a grid representing every possible parameter combination would be called "exhaustive enumeration"). Also make sure to get rid of the subinterval that was bisected while pruning. Extract the LRLB on each process. Communicate this across processes on this server and extract the global (least) LRLB value. If the global incumbent is closer than ε to the global LRLB, then convergence has been achieved on that server. In this case, terminate execution of every process on that server and return the global incumbent as a globally optimal system configuration on that server. Otherwise, begin another iteration. Once every server has converged to the global solution, pick the minimum and return that as the global solution for the problem. If at any point of the execution, the LRLB on any server becomes greater than the best incumbent among all servers, terminate execution on that server - the chunk of the search space which was initially statically assigned to it is infeasible.

It should be clear that this procedure represents a constructive proof for identifying a solution sufficiently close to a global solution. We see that we need to lower bound the objective to be able to globally minimize it in a verified fashion. An applicable lower bounding procedure is outlined in the next subsection. The algorithm is visualized thoroughly in the context of the first numerical example. Before concluding this subsection, we note that this parallelization strategy is classified as static scheduling to a single dynamically scheduled work queue per server, with prioritization based on the best-bound rule on every such queue. Hence, it is a mixed static-dynamic parallelization strategy, with the static component being made necessary by the limitations of the loosely-coupled EC2 parallel environment coupled to the limitations on the parallelization tools available in the environment needed to lower bound our merit function. This is touched on in more detail at the beginning of Section 2.4.

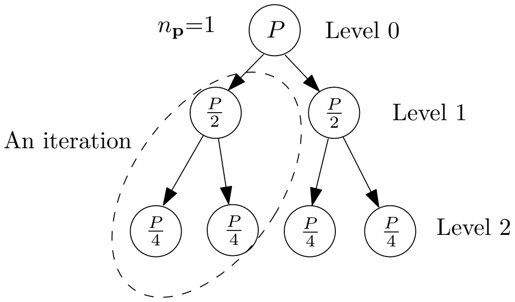

Figure 2-3: *An illustration of the concept of level for a one-dimensional search space.*

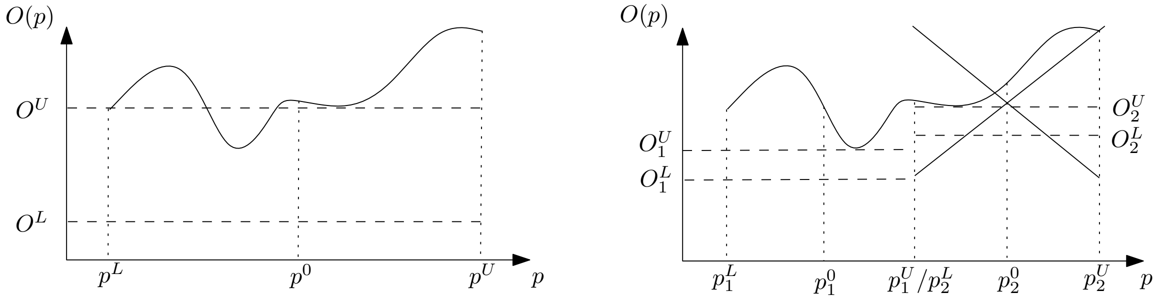

Figure 2-4: *The bisection of a simple univariate objective.*

2.3.2 Lower Bounding Procedure

Taylor Arithmetic

The solution to the problem of bounding the ranges of functions on intervals began with interval arithmetic (which originated with the work of Ramon E. Moore [28] in the 1960s). Interval arithmetic has been automated in many software packages, such as the MATLAB toolbox INTLAB [40] and the C++ library C-XSC [14]. It is applicable to factorable functions, i.e., functions which can be computed in a finite number of simple steps. The function of relevance here is some function of the front-interface reflectance $\mathcal{R}$, which for any given wavelength and incident angle is a function of $\mathbf{p}$ composed entirely of the binary operations $+, -, \times, /$ and trigonometric functions sin and cos, polynomial functions and roots (see [20], for instance, for the general expression). This function is made available by the transfer matrix model (the solution to Maxwell's equations relevant for the thin-film optical filter architecture). Interval arithmetic employs a set of rules corresponding to each step in the computation of the function, e.g.,

$$\left[a^L, a^U\right] + \left[b^L, b^U\right] = \left[a^L + b^L, a^U + b^U\right], \tag{2.5}$$

and can be used to compute an interval bound $\left[\mathcal{R}^L, \mathcal{R}^U\right]$ of $\mathcal{R}$ on P. As a simple example, consider the expression $p^2 + p$, where p can take values on the interval $[-1, 1]$. Employing the simple rule that the lower and upper bounds of the square of an interval are respectively the square of the smallest and largest absolute values in the interval, together with the addition rule (2.5), interval arithmetic can be applied to bound the expression as follows:

$$[-1, 1]^2 + [-1, 1] = [0, 1] + [-1, 1] = [-1, 2]. \tag{2.6}$$

These bounds, together with the expression, are plotted in Figure 2-5.

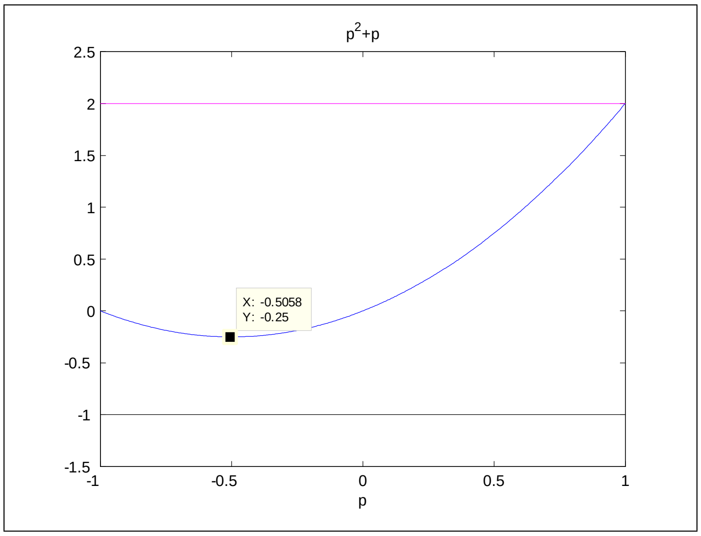

Figure 2-5: *Illustration of the non-exactness of interval arithmetic bounds for a simple expression.*

Note that the lower bound, although valid rigorously and hence theoretically applicable, is not exact (-1 is significantly lower than the true/exact lower bound of $-\frac{1}{4}$). Now, rearrange the expression as $\left(p + \frac{1}{2}\right)^2 - \frac{1}{4}$. Applying interval arithmetic to this yields exact bounds:

$$\left[-1+\frac{1}{2}, 1+\frac{1}{2}\right]^2 - \frac{1}{4} = \left[0, \frac{9}{4}\right] - \frac{1}{4} = \left[-\frac{1}{4}, 2\right]. \tag{2.7}$$

In general, it can be shown that rearranging the expression such that each interval-valued variable appears only once yields exact bounds. The following extreme example further illustrates this issue:

$$\left[p^L, p^U\right] - \left[p^L, p^U\right] = \left[p^L, p^U\right] + \left[-p^U, -p^L\right] = \left[p^L - p^U, p^U - p^L\right] \neq 0. \tag{2.8}$$

In other words, even though we know that the expression above should always be zero (as the difference between a quantity and itself), naive application of interval arithmetic yields a significantly wider interval bound for the expression. Rearranging this expression such that the interval-valued variable appears only once eliminates this problem (using a self-explanatory interval arithmetic rule for the product of an interval by a nonnegative scalar):

$$\left[p^L, p^U\right] \times (1-1) = \left[p^L, p^U\right] \times 0 = \left[p^L \times 0, p^U \times 0\right] = [0, 0]. \tag{2.9}$$

For most practical expressions, such a rearrangement is unfortunately not entirely possible. This issue is referred to as the dependency problem in the deterministic global optimization literature. This terminology reflects the fact that the problem arises from the inability of interval arithmetic to account for the dependency (or sensitivity) of each term on the underlying independent variables.

The dependency problem is a serious issue for thin-film optical filters. Consider the closed-form expression for reflectance in the relatively simple case of a two layer device at normal incidence (assuming all materials are nonabsorbing so that the refractive indices are real):

$$\mathcal{R}\left(\mathbf{p}\right) = \frac{\begin{array}{c}\left(\left(1-n_{sub}\right)\cos\delta_1\cos\delta_2-\left(\frac{n_2}{n_1}-n_{sub}\frac{n_1}{n_2}\right)\sin\delta_1\sin\delta_2\right)^2\\+\left(\left(\frac{n_{sub}}{n_1}-n_1\right)\sin\delta_1\cos\delta_2+\left(\frac{n_{sub}}{n_2}-n_2\right)\cos\delta_1\sin\delta_2\right)^2\end{array}}{\begin{array}{c}\left(\left(1+n_{sub}\right)\cos\delta_1\cos\delta_2-\left(\frac{n_2}{n_1}+n_{sub}\frac{n_1}{n_2}\right)\sin\delta_1\sin\delta_2\right)^2\\+\left(\left(\frac{n_{sub}}{n_1}+n_1\right)\sin\delta_1\cos\delta_2+\left(\frac{n_{sub}}{n_2}+n_2\right)\cos\delta_1\sin\delta_2\right)^2\end{array}}. \tag{2.10}$$

Here,

$$\delta_k = \frac{2\pi n_k d_k}{\lambda} \tag{2.11}$$

is the phase change experienced by electromagnetic radiation in passing through layer k. Note multiple occurrences of the refractive index and thickness variables, which cannot be eliminated by simply rearranging the expression (at least not in a way we are aware of). However the following simple rearrangement is possible to reduce it, and this is done in our implementation.

By Snell's law,

$$\sin\theta_k = \frac{n_0\sin\theta_0}{n_k}. \tag{2.12}$$

Then,

$$\cos\theta_k = \sqrt{1-\frac{n_o^2\sin^2\theta_0}{n_k^2}}. \tag{2.13}$$

But since there are several appearances of the term $n_k\cos\theta_k$ (see expression in [20], for instance, if not convinced), always computing it as

$$n_k\cos\theta_k = \sqrt{n_k^2 - n_o^2\sin^2\theta_0} \tag{2.14}$$

reduces dependency (since n_k appears only once).

To alleviate the dependency problem, one can employ an approach referred to as Taylor arithmetic and automated in the (extensively verified) system COSY INFINITY [4] that is based on Fortran 77 (other techniques, mentioned in Section 2.5, can also be applied to alleviate the dependency problem, but Taylor arithmetic appears to be the most mature technique for this purpose, being automated in an extensively verified software package. For this reason, Taylor arithmetic is today the "state-of-the-art" for globally optimizing an arbitrary engineering system (characterized by a model given by an explicit expression of sufficient differentiability) in a verified fashion and so is a natural choice for the study in this work. It applies Taylor's theorem to bound an $o+1$ times continuously partially differentiable (on the interval under consideration) function f of the interval-valued variable $\mathbf{p}$ by applying interval arithmetic to the following expression (the square brackets [] denote an interval bound for the enclosed quantity):

$$[f]\left([\mathbf{p}]\right) = f\left(\mathbf{p}_0\right) + \sum_{i=1}^{o} \frac{1}{i!} D^i f\left(\mathbf{p}_0\right)\left([\mathbf{p}] - \mathbf{p}_0\right)^i + [r]\left([\mathbf{p}], \mathbf{p}_0\right), \tag{2.15}$$

with $D^i f\left(\mathbf{p}_0\right)$ being the i^{th} order partial derivative of f at $\mathbf{p}_0$. An explicit expression for the remainder is available for such functions, it being the main tool for obtaining $[r]$ (the interested reader is referred to the COSY INFINITY manual [4] and the references therein for further mathematical detail). In other words, after expressing the function as the sum of its Taylor expansion of some specified order o around some reference point $\mathbf{p}_0$ and a remainder term r, interval arithmetic is applied to that expression. In an intuitive sense, the reason this works in reducing the dependency problem is that (2.15) attempts to explicitly express the function in terms of its dependencies (the derivatives, at $\mathbf{p}_0$) on $\mathbf{p}$. Derivatives are computed using automatic differentiation [11] (the automatic computational equivalent of the forward rule of differentiation). Again, consider the simple extreme example (2.8) for illustration purposes. For $o = 0$, the expression (2.15) is written as follows (the second part comes from the remainder term):

$$f\left(p_0\right) + \frac{df}{dp}\left(p_0\right)\left([p] - p_0\right). \tag{2.16}$$

For $p \in [-1, 1]$, choosing $p_0 = 1$ one obtains exact bounds as follows:

$$(1-1) + \overbrace{(1-1)}^{\frac{df}{dp}=0}([-1,1]-1) = [0,0]\,. \tag{2.17}$$

In general, higher o yields tighter bounds at a higher computational cost and is chosen empirically (we choose it arbitrarily in this work, as reported in Section 2.4), while $\mathbf{p}_0$ may be chosen arbitrarily (in this work, we henceforth choose the midpoint of the interval for this purpose). A set of rules is then defined for binary operations on a pair of such representations, allowing one to build up such representations for complex expressions computed in an iterated fashion in long code lists. Beyond simple application of interval arithmetic to (2.15), more intelligent use of the Taylor expansion can lead to tighter bounds. In this work, we employ the Linear Dominated Bounder (LDB) as such an intelligent alternative (the interested reader is referred to the COSY INFINITY manual [4] for details pertaining to the LDB, as well as any other details of interest, including other such intelligent alternatives).

We must now check whether differentiability requirements are satisfied. These requirements are satisfied by merit functions which are smooth functions of $\mathcal{R}$ (in technical lingo, of class $\mathcal{C}^\infty$) for any o, since the composition of smooth functions is a smooth function, and both the numerator and the denominator of $\mathcal{R}$ are built up entirely from smooth function of $\mathbf{p}$ (polynomial functions, the trigonometric functions and roots) and binary operations which preserve smoothness. Since both terms are positive, the lower bound on the result of the division is obtained as the division of the lower bound on the numerator and the upper bound of the denominator. The upper bound on the result of the division is obtained as the division of the upper bound on the numerator and the lower bound of the denominator. This can then be used as an interval to construct a bound on the merit function (we note that this is not necessary for the examples we look at in this work, since both involve minimizing reflection, so that only the lower bound of this interval is required).

COSY INFINITY has been used for rigorous global optimization of some abstract functions which have traditionally been used for testing optimization algorithms (e.g., the Beale function in [3]) and some problems in charged-particle optics (e.g., a triple bend achromat located at Lawrence Livermore National Laboratory in [22]), but not on multilayer filter problems and using a branch-and-bound algorithm differing in many details of its development (details of that implementation can be found in [3] and the references therein). COSY INFINITY has been used for a variety of other purposes by over 1000 people worldwide, including, but not limited to, high-order multivariate automatic differentiation, solution of ODEs (also validated solution of ODEs, where a rigorous bound on the ODE solution is constructed pointwise in time) and advanced particle beam dynamics simulations. Note that all bounds are guaranteed to be rigorous, despite the roundoff errors that are inherent in finite machine arithmetic.

Incorporating Natural Bounding Information

The last-but-one step in the computation of $\mathcal{R}$ involves the application of the square function (division of two such squares being the final step). Algebraically then $\mathcal{R}$ cannot be negative (which corresponds to the physical observation that reflectance cannot be negative). However, because the square in the current version of COSY INFINITY is evaluated as multiplication by self [4], a negative lower bound on $\mathcal{R}$ is actually possible. As an illustration of this fact, consider the square of the interval $[-1, 1]$. The lower bound of that quantity is certainly 0, but interval arithmetic would evaluate the lower bound of $[-1, 1] * [-1, 1]$ to be -1! Thus, it is important to truncate the lower bound on $\mathcal{R}$ to 0 following computation, whenever possible, to yield a tighter bound in general.

Experimental Uncertainty Analysis

Techniques for bounding ranges of functions on intervals are natural tools for performing experimental uncertainty analysis. Given an interval range for the expected experimental variability in thickness/refractive indices, one can readily compute an interval bound for the variation in the merit function that one can expect experimentally. This conceptually simple approach is superior to sampling techniques which have been used for uncertainty analysis of multilayer filters in the literature. These typically involve generating random device architectures in the thickness/refractive index interval and statistically analyzing these samples (see [46], [20] for example applications of this approach). Such an approach is of course not rigorous and significantly more computationally expensive than simply computing a merit function bound using Taylor arithmetic. We have not seen anyone make a reference to rigorous bounding tools for uncertainty analysis in the multilayer system literature, which is why we do so here. We do not discuss this approach further for now.

2.4 Numerical Examples

The parallel branch-and-bound algorithm was implemented using Amazon's EC2 platform and the COSY INFINITY system (in particular, single work queue dynamic scheduling components of the algorithm on any given server were implemented using the scheduling construct `PLOOP` made recently available by COSY INFINITY's authors, this construct providing an interface to the Message Passing Interface or MPI but restricted to all-to-all communication between processes).

It is emphasized, briefly, that because this construct only allows all-to-all communication between running processes, the communication and synchronization costs prohibit dynamic scheduling of tasks across multiple servers. In fact, when we attempted doing this, we found the communication cost to be several times larger than computational time (showing negative speedup in moving from a single server to multiple servers, details of this test are presented in the context of the first numerical example). Traditional approaches to reducing communication cost, by granulating communication (communicating more intervals to each process to be bisected at each iteration, so as to communicate less often) do not work here because doing so merely introduces a trade-off with synchronization cost under the all-to-all communication paradigm (it is now more likely that some processes are done with their work significantly earlier than others, and since communication cannot happen until all processes are done, this translates into a higher synchronization time). Suggestions for future work to circumvent this constraint are outlined in the final chapter of this thesis.

The instance type we used on Amazon was the `cc2.8xlarge` high memory (80 GB) instance. These instances possess 16 physical cores each and come with hyperthreading enabled for a total of 32 virtual cores. However, our tests indicated that our algorithm suffers a slowdown from this feature (a factor of about 2.5 slower when moving from 16 to 32 threads on any one server, we omit the details of this test for brevity). This is consistent with the experience of many workers in high performance computing (see for instance [25]). Hence, we disabled hyperthreading and so have only 16 threads per `cc2.8xlarge` server.

Models were tested against analytic examples in the literature while also being validated against real data (see the second numerical example for details of the comparison to real data, other details are omitted for brevity). Merit lower bounding code was tested for consistency (a notion from deterministic global optimization theory, theoretical guarantees of convergence require merit lower bounds to possess this property), i.e., it was checked that the bounds become tighter (meaning larger, given that we focus here on minimization problems) as the parameter interval on which the lower bound is computed is made smaller, and that the merit value is attained on a thin/degenerate interval. This allows us to conjecture that our algorithm is provably convergent (details of this test are omitted for now for brevity). All materials are assumed to be nondispersive, so that their refractive indices are assumed to be constant over the wavelength ranges of interest. We believe this to be reasonable, since it is well-established that the refractive indices of most dielectrics do not change appreciably over the wavelength ranges considered in this work. Thin-film materials are assumed to be nonabsorbing as well, for the design problems considered in this work, this is a valid assumption (an antireflection coating material must transmit all or at least most of the incident radiation to the substrate). Moreover, the model is tested against real data in the context of the second numerical example and found to be fairly accurate, which validates our modeling assumptions. In the computation, the substrate is assumed to be semi-infinite. Concurrent dollar costs of running each example using the EC2 service are reported along with solution times. Comparison with stochastic global optimization methods is made where appropriate, the goal of such a comparison being to gauge what advantage this algorithm has over "state-of-the-art" existing methods. We choose to compare with variants of multistart optimization, where randomly sampled initial guesses for the optimal system configuration are repeatedly locally optimized for some user-specified number of iterations, and the best solution found over all iterations reported in the end. The reason for choosing this class of stochastic methods (over, say, genetic algorithms or simulated annealing) is that there is a prevalent belief among experts in the literature [39] that this is the most promising subclass of stochastic methods. Comparisons employs the C++ interface to the package NLopt (written by Steven G. Johnson, and freely available at `http://ab-initio.mit.edu/nlopt`) for the intelligent Multi Level Single Linkage (MLSL) multistart algorithm [39] (which employs a clustering heuristic to reduce redundancy in the initial guess choice at every iteration, rather than just picking a uniformly randomly selected one each time, and is guaranteed to find all local minima in a finite number of iterations). In particular, we use the derivative-based sequential quadratic programming (SQP) algorithm for the local optimization component, with the gradient being derived explicitly (again, details of this are omitted for brevity). Termination criteria for MLSL is maximum number of merit function evaluations and an absolute convergence tolerance on the merit function value (whichever is reached first, specific values selected for these are given in the numerical examples below). The midpoint of the search space is used as an initial guess for the search. In each case, specific values for these termination criteria are given. CPU times reported for serial stochastic optimization tests are given for an Intel CPU with clock speed of 1.79 GHz and 2 GB of RAM. All results are reported to within 3 significant figures (except for inherently integer quantities, such as the number of iterations for convergence, which are reported without any approximation).

2.4.1 Broadband Normal Incidence Antireflection Coating for Silicon Solar Cells

We first consider the problem of designing an antireflection coating for silicon solar cells, with the goal of minimizing average normal incidence reflectance over a broad range of wavelengths ($\lambda \in [400, 1600]\,nm$). This is captured by minimizing the objective

$$\begin{aligned} O\left(\mathbf{p}\right) &= \frac{1}{1200}\int_{400nm}^{1600nm} \mathcal{R}\left(\mathbf{p}, \lambda, 0\right) d\lambda, \\ &\approx \frac{1}{1200}\frac{1200}{10}\sum_{i=1}^{i=10} \mathcal{R}\left(\mathbf{p}, \lambda_i, 0\right), \quad \lambda_i = 400 + \left(i-1\right)\frac{1200}{10}nm, \end{aligned} \tag{2.18}$$

where the numerical approximation for the definite integral is performed using the rectangle method (using 10 rectangles and the top-left corner approximation, corresponding to $m = 10$ and $n = 1$ in (2.2)). We use a 3^{rd} order Taylor expansion (chosen arbitrarily) for constructing the lower bound on the merit function for this example. Thicknesses and refractive indices of every layer are used as design variables (thereby specifying the design vector $\mathbf{p}$). Thicknesses are assumed variable in $[5, 500]\,nm$, which we believe to be representative of configurations reliably realizable on our sputtering system (described in detail in the next chapter). Refractive indices are assumed to be variable in the interval $[1.09, 2.60]$. This is consistent with the recent demonstration of refractive index variability achievable through oblique angle deposition of SiO_2 and co-sputtering of SiO_2/TiO_2 [19]. The refractive index of Silicon is obtained by averaging refractive index data (obtained from `http://www.filmetrics.com/`) over the wavelength range of interest on a uniform grid of 10^3 points. This yields a refractive index of 3.73 (with the imaginary part being 0.02). Absolute convergence tolerance is fixed at 0.1%, justification for which will be presented in the next design example by comparison to real data.

Before proceeding, we would like to demonstrate that our algorithm is correct. We do this by thoroughly visualizing its behavior in the context of the one layer (i.e., two parameter) problem. With only two parameters, it is possible to visualize the merit function and pick the global optimum approximately visually. We show this plot in Figure 2-6. It is clear that this problem is highly nonconvex, even in this small dimensional case, the issue can be expected to become much worse in larger dimensions. We see that the global solution is approximately $\mathbf{p}_{opt} = \begin{bmatrix} 1.95 & 145 \end{bmatrix}$ which corresponds to a merit function value of 10.6%. This problem is simple enough for our branch-and-bound algorithm to solve relatively quickly on a single process. Doing this yields the solution $\mathbf{p}_{opt} = \begin{bmatrix} 1.93 & 148 \end{bmatrix}$ and a corresponding merit function value of 10.6%, in 2424 iterations, 6.91 seconds CPU time and 7.07 seconds wall clock time. This design can be realized approximately experimentally using a material such as yttrium oxide (Y_2O_3). The convergence information (the incumbent and the least remaining lower bound evolution) is shown in Figure 2-7. This exercise validates our code.

We further visualize algorithm behavior on multiple processes in this simple example. We run the algorithm on 16 processes (for convergence in 3.92 seconds wall clock time) and visualize in three dimensions what the algorithm is doing on every process. This is shown in Figure 2-8. Corresponding two dimensional convergence information is shown in Figure 2-9. Finally, we perform a full visualization of the stack evolution for the refractive index interval [1.09 1.50] and the thickness interval [5 50] nanometers in Appendix A. For this proper subset of the search space, it only takes the algorithm 8 iterations to converge on two processes, making it ideal for such a detailed evaluation (however, only the first four iterations are shown).

Next, we increase the complexity of the problem to two layers (i.e., 4 parameters) but increase the absolute convergence tolerance to 2%. This yields a problem that is complex enough to analyze for scaling with number of processes, but simple enough for this to be done in reasonable time. Results of scaling tests are reported in Table 2.1. Efficiency of parallelization is measured as

$$E = \frac{T_1}{NPT_{NP}}. \tag{2.19}$$

Here, T_1 is the serial CPU time, T_{NP} is the wall clock time for execution on NP processors. It is well-established that ideal (linear) scalability would be represented by efficiency of 1 $\forall NP$, but many practical algorithms show an efficiency that declines with larger NP due to more effort spent on synchronization and communication (with efficiency reaching 0 for an infinite number of processors) [21]. Efficiency numbers greater than one indicate superlinear speedup. We see that our algorithm is arguably quite efficient on a single server. In moving from a single process to multiple processes on a single server (i.e., up to 16 processes) we see efficiency numbers greater than 1. This means we are experiencing superlinear speedup. In moving from one to 12 processes, for instance, the speedup is $\frac{T_1}{T_{12}} \approx 28$ which is much larger than 12 (expected speedup under ideal circumstances). This effect is less dramatic when considering moving from 8 to 16 processes, for instance (in that case, speedup is only negligibly superlinear, being only slightly larger than 2). Indeed, if the serial run is eliminated, the speedup appears to be slightly superlinear most of the time. When considering speedup based on iteration number, it is always slightly superlinear, but approximately linear. Superlinear speedup is a well-documented phenomenon in the parallel computing community. Combined with the only slight corresponding superlinear speedup in iteration number, we can conclude that the massive superlinear speedup in moving from one to multiple processes is due to memory effects, i.e., on multiple processes the space complexity on any one server is smaller, making any smaller piece of memory easier to access and search [21]. The slight superlinear speedup the rest of the time can be attributed to another well-known effect. With a higher number of processes, the tree is searched in a different order leading to faster convergence [21]. This effect can be directly exploited by simulating multiple threads via oversubscription, provided the context switching cost is not significantly larger than the benefit from searching the tree in a different order. In our case, since the superlinear effect is only slight, we found that this exercise yields no benefit. In moving from 16 processes on one server to 32 on two servers we observe a sublinear speedup (approximately 1.61 which is less than the 2 we expect in the ideal scenario). This can be easily attributed to the static assignment of work to each server. The decision tree is unbalanced and one server finishes earlier than the second one, some computing power is wasted sitting idly by while work remains to be done (in this case one server took 187 seconds while the second took 136 seconds).

We next try the harder two layer problem on one server (with the absolute convergence tolerance set back at 0.1%). We solve this problem in 179098 iterations and 84080 seconds (23.4 hours, costing \$7.2) wall clock time. The solution is 0.0462 and the solution vector is $\begin{bmatrix} 1.56 & 2.38 & 100 & 66.1 \end{bmatrix}$. We then try to statically schedule to 16 servers to see if we can significantly speed it up. Doing this yields the same solution in 161854 iterations and 81381 seconds (22.6 hours) wall clock time. This represents a saving of approximately 3.21% in wall clock time. While this represents a positive speedup, it is very low due to the necessity for static scheduling to servers made necessary by the all-to-all communication constraint on the parallelization construct in our software environment. Statically scheduling to a multitude of servers would allow one to leverage vast amounts of computing power and use this tool as a dynamic mesh search engine by simply using it in nonverified fashion to search for candidate solutions, with the lower bounding component hopefully providing a useful lower bound (not necessarily large enough to ensure convergence to a global solution). We do not concern ourselves with this further for now though, and all results are henceforth reported for a run on 16 processes (i.e., on one server). In other words, for the remainder of this work, the dynamically scheduled component of the described algorithm is treated as a shared memory algorithm being tested on a small scale.

Results of branch-and-bound algorithm execution for two layers with varying upper bound for thicknesses are shown in Table 2.2. We vary this to study the sensitivity of convergence and solution information to the thickness interval. Absolute convergence tolerance is set at 0.1%. Problems with greater numbers of layers are not executed due to what we categorize as diminishing returns in solution quality in moving from two to three layers. Dollar costs could have been computed at concurrent EC2 on-demand instance price of \$2.0 per 16 processes per hour, but we actually used the flexible spot instance pricing model, so this cost is the one that is presented. This flexible pricing model is market based, i.e., price varies depending on demand and supply of instances in this pricing pool. At the time we ran our tests, the market worked out to approximately \$0.3 per 16 processes per hour, which appears to be the lower bound on price overall at the time of this writing.

Results of MLSL runs are reported in Table 2.3 for comparison. The termination criterion for this test was an absolute convergence tolerance 10^{-10} and 10^5 maximum function evaluations (which ever one is reached first). This was selected to obtain a solution that we felt was "reasonably good" in a "reasonable" amount of time, as is often done in practice with stochastic global optimization tools. We see that the stochastic methods find the global solution as well, even if this is done without the guarantee of global optimality.

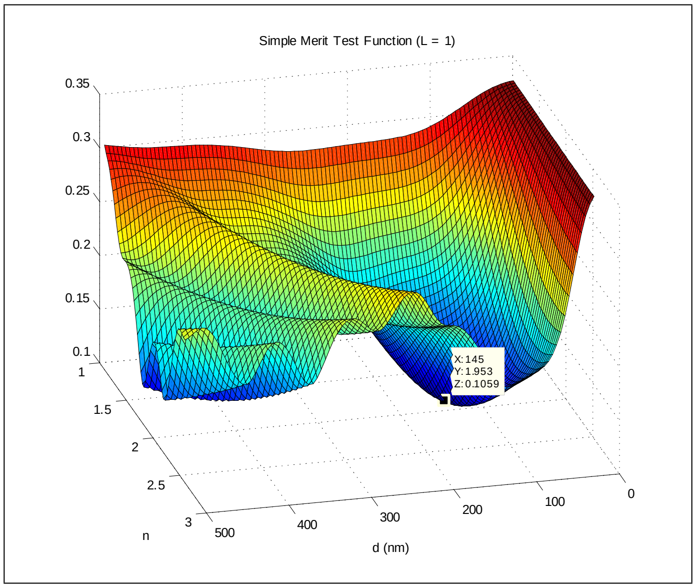

Figure 2-6: *Simple test function for deterministic algorithm. One layer normal incidence problem visualized showing the approximate global optimum.*

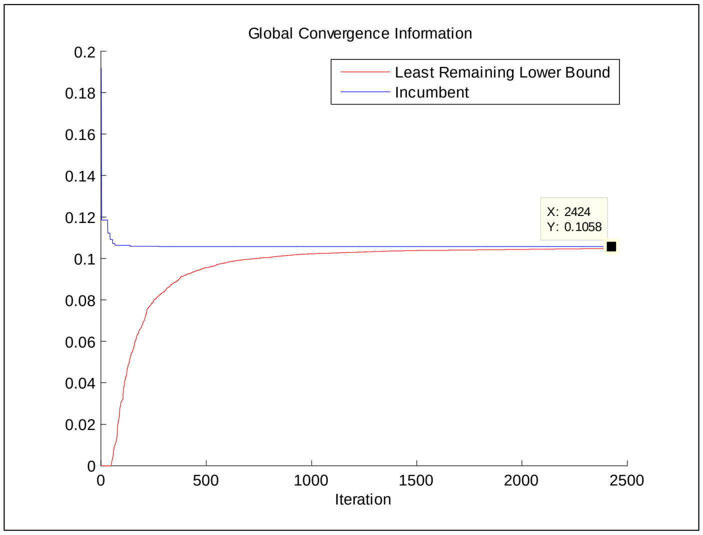

Figure 2-7: *Simple serial test example (L=1) convergence information.*

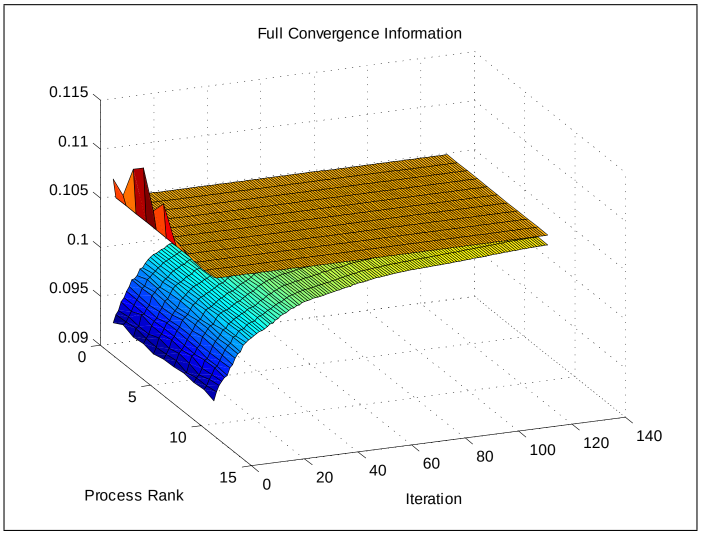

Figure 2-8: *Simple serial test example (L=1) 3D convergence information on 16 processes.*

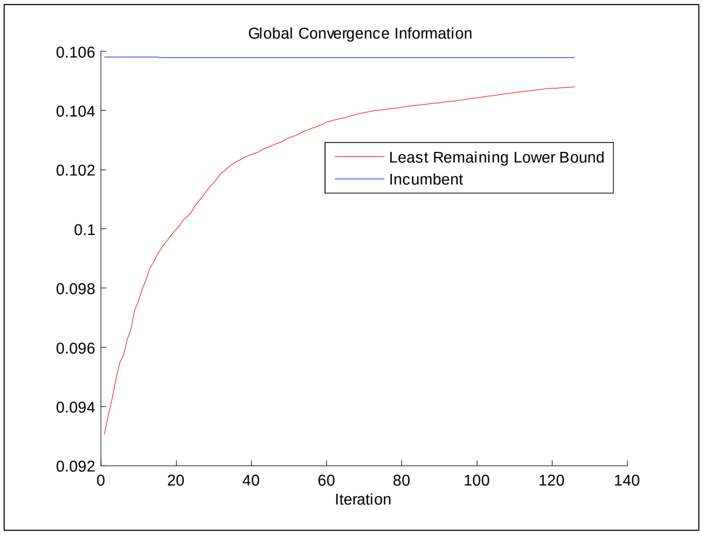

Figure 2-9: *Simple serial test example (L=1) corresponding 2D convergence information on 16 processes.*

$NPROCESSES$	Wall Clock (sec)	Efficiency	Iterations	Solution (optimal merit, $\mathbf{p}_{opt}$)
1	10800	1	127731	0.0465, $\begin{bmatrix} 1.57 & 2.42 & 102 & 63.0 \end{bmatrix}$
4	1410	1.92	31930	0.0465, $\begin{bmatrix} 1.57 & 2.42 & 102 & 63.0 \end{bmatrix}$
8	608	2.22	15960	0.0465, $\begin{bmatrix} 1.57 & 2.42 & 102 & 63.0 \end{bmatrix}$
12	384	2.35	10634	0.0465, $\begin{bmatrix} 1.57 & 2.42 & 102 & 63.0 \end{bmatrix}$
16	302	2.24	7969	0.0465, $\begin{bmatrix} 1.57 & 2.42 & 102 & 63.0 \end{bmatrix}$
32	187	1.81	5717	0.0465, $\begin{bmatrix} 1.57 & 2.42 & 102 & 63.0 \end{bmatrix}$

Table 2.1: *Deterministic algorithm scaling test.*

2.4.2 Broadband Omnidirectional Antireflection Coating for Silicon Solar Cells

Next, we consider a more important but harder practical design problem, the problem of minimizing average reflectance from a silicon solar cell over a broad range of incident angles $\left(\left[0, \frac{\pi}{3}\right]\right)$ in addition to wavelengths $\left([400, 1600]\, nm\right)$. This is captured by minimizing the objective

$$O\left(\mathbf{p}\right) = \frac{3}{\pi}\frac{1}{1200}\int_0^{\frac{\pi}{3}}\int_{400nm}^{1600nm}\mathcal{R}\left(\mathbf{p}, \lambda, \theta\right) d\lambda,$$

where the numerical approximation for the definite integral is again performed using the rectangle method (using 10 rectangles for each independent variable and the top-left cor-

L	d^U	Wall Clock Time	Iterations	Solution (merit, $\mathbf{p}_{opt}$)	EC2 Cost
1	500	7.07	2424	0.106, $\begin{bmatrix} 1.93 & 148 \end{bmatrix}$	\$0.30
2	500	84080/23.4	179098	0.0462, $\begin{bmatrix} 1.57 & 2.38 & 100 & 65.9 \end{bmatrix}$	\$6.90
2	250	29866/8.30	90764	0.0462, $\begin{bmatrix} 1.57 & 2.38 & 100 & 65.9 \end{bmatrix}$	\$2.48
3	250	1047237/291/12.1	808148	0.0136, $\begin{bmatrix} 1.32 & 1.86 & 2.60 & 120 & 77.5 & 60.7 \end{bmatrix}$	\$87.3

Table 2.2: *Deterministic algorithm solution information. Wall clock time is presented in the format seconds/hours/days, with some of that information ommitted whenever it is redundant. The thickness upper bound d^U is given in the units of nanometers.*

L	Convergence Time (sec)	Solution (optimal merit, $\mathbf{p}_{opt}$)
1	11.9	0.106, $\begin{bmatrix} 1.93 & 148 \end{bmatrix}$
2	42.1	0.0462, $\begin{bmatrix} 1.57 & 2.38 & 100 & 65.9 \end{bmatrix}$
3	104	0.0135, $\begin{bmatrix} 1.32 & 1.86 & 2.60 & 120 & 77.5 & 60.8 \end{bmatrix}$

Table 2.3: *Stochastic algorithm solution information for the first design problem.*

ner approximation, corresponding to $m = 10$ and $n = 10$ in (2.2)). We use a 3^{rd} order Taylor expansion (chosen arbitrarily) for constructing the lower bound on the merit function for this example, as for the last example. Thicknesses and refractive indices of every layer are used as design variables (thereby specifying the design vector $\mathbf{p}$). Thicknesses are assumed variable in $[5, 500]\,nm$, which we believe to be representative of configurations reliably realizable on our sputtering system (described in detail in the next chapter), although we do make the thickness interval narrower for the harder problems (observe that in moving from a thicker interval to a smaller one in the previous example, as in this example, the global solution does not change provided the original solution was contained in it). As before, refractive indices are assumed to be variable in the interval $[1.09, 2.60]$. For this merit function, we compare our model to data from the literature (the result in [19]) and found the discrepancy to be only 0.13% (3.66% versus 3.79% measured in that work). This suggests that our modeling error is less than 0.2% and leads us to set the absolute convergence tolerance to 0.1%. Deterministic algorithm solution information is shown in Table 2.4 and stochastic in Table 2.5. Note again that the stochastic tool finds the solution, albeit without any guarantee. Also, observe that the three layer solution can be approximated fairly closely using the practical materials MgF_2, Y_2O_3 and the rutile phase of TiO_2 (these exact refractive indices have been reported in the literature for these materials). MgF_2 could be substituted by other fluorides such as LiF and NaF, Y_2O_3 by other oxides such as HfO_2 (hafnium oxide) and TiO_2 by high index materials such as ZnS.

L	d^U	Wall Clock Time	Iterations	Solution (merit, $\mathbf{p}_{opt}$)	EC2 Cost
1	500	22.5	136	0.112, $\begin{bmatrix} 1.93 & 153 \end{bmatrix}$	\$0.30
2	500	118513/32.9	202134	0.0526, $\begin{bmatrix} 1.55 & 2.37 & 109 & 68.3 \end{bmatrix}$	\$6.90
2	250	44093/12.2	110744	0.0526, $\begin{bmatrix} 1.55 & 2.37 & 109 & 68.4 \end{bmatrix}$	\$3.90
3	200	1663253/462/19.3	935599	0.0182, $\begin{bmatrix} 1.31 & 1.85 & 2.60 & 131 & 80.8 & 61.9 \end{bmatrix}$	\$159

Table 2.4: *Deterministic solution information for the second numerical example. Wall clock time is presented in the format seconds/hours/days, with some information omitted when redundant. The thickness upper bound d^U is given in the units of nanometers.*

L	Convergence Time (sec)	Solution (optimal merit, $\mathbf{p}_{opt}$)
1	66.6	0.112, $\begin{bmatrix} 1.93 & 153 \end{bmatrix}$
2	295	0.0526, $\begin{bmatrix} 1.55 & 2.37 & 109 & 68.4 \end{bmatrix}$
3	918	0.0182, $\begin{bmatrix} 1.31 & 1.85 & 2.60 & 131 & 81.0 & 61.5 \end{bmatrix}$

Table 2.5: *Stochastic algorithm solution information for the second design problem.*

2.5 Discussion and Conclusion

This chapter engineered the first ever, to the best of our knowledge, deterministic global optimization algorithm for thin-film optical systems. Two important broadband problems pertaining to reducing reflection from silicon were looked at. In both the normal incidence and the omnidirectional cases, the global solutions could be realized using practical (i.e., naturally occurring) materials, thereby not requiring the sophisticated nano-deposition technology that has been used to realize the best performing devices for this problem in recent times. This means that the use of sophisticated nanotechnology is not necessary for this particular problem. Moreover, the solution does not require any toxic chemicals and would work equally well for both polycrystalline and monocrystalline silicon, unlike solutions based on texturing. Current state-of-the-art stochastic global optimization tools find the solution anyway, in every case we have looked at to date, but having the rigorous guarantee provided by the algorithm developed in this work enables one to make interesting rigorous theoretical statements (like the one just made previously/above). It also must be emphasized that the stochastic tools do need to be used correctly to find the global optimum. One can always construct a use case scenario where a global solution has not been found when the stochastic algorithm terminates, for instance by picking a termination criterion that is too loose. As problem complexity increases, we expect the guarantee of global optimality to become more important (making it less likely that a global solution was missed because the stochastic tool was terminated too early, for instance). Moreover, different design problems in this class, of similar complexity but a higher degree of nonconvexity, may benefit from improved solution information as well, not just the guarantee. This work has demonstrated that it is possible to optimize important practical problems in this class in a verified fashion, with relatively small branch-and-bound implementations. With further work (some potential directions are outlined in the final chapter) it may become possible to have bigger implementations of more efficient algorithms, thereby solving even harder problems and making this class of algorithms increasingly important.

We note also that all global solutions observed are gradient-index ones, i.e., the refractive indices increase monotonically and thicknesses similarly decrease monotonically from air to substrate. This has been widely conjectured to be true for antireflection coatings in the literature (provided the bandwidth is sufficiently wide, see for instance [33]). This structural feature can be exploited to make the algorithm more efficient as follows. Create for every variable, other than those associated with the first layer, a variable to lie on the interval $[0, 1]$. Call these, $\forall i > 1$, $p_i^{n_{gi}}$. Then, $\forall i > 1$, set $n_{i+1} = p_i^{n_{gi}} n_i$. An analogous constraint can be implemented for the thickness variables. In doing this, the domain is reduced to a fraction that is $\frac{1}{2^{2\left(\sum_{i=1}^{L-1} i\right)}}$ of the original space. Here, the factor of 2 in the exponent of the denominator is due to the fact that such conditions hold for both the thickness and refractive index variables, while the summation term in the exponent of the denominator is due to the fact that for every inequality in the monotonicity constraint, the space is reduced to half of its original size. This fraction is equal to 1, $\frac{1}{4}$ and $\frac{1}{64}$ for one, two and three layer problems respectively, promising significant reduction in convergence time. The refractive index fraction is shown for the three layer case, to aid in visualization, in Figure 2-10. The inequalities in question here are

$$n_1 \leq n_2 \leq n_3.$$

Three planes can be seen in the Figure, one for each inequality between the refractive index variables (hence the reduction is $\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$ due to the refractive indices and thereby $\frac{1}{64}$ overall). For four layers, consider that this reduction is $\frac{1}{4096}$ overall. This exponential domain reduction mechanism promises to provide for very efficient algorithms for gradient-index systems. Consider, also, that there are other classes of optical systems (for instance, gradient-index fibers and gradient-index lenses) where this constraint is imposed a priori for practical reasons, so that this domain reduction mechanism may be more widely applicable than thin-film antireflection coatings. Finally, observe that the domain reduction can be used by any optimization algorithm, not just a branch-and-bound or even a deterministic one.

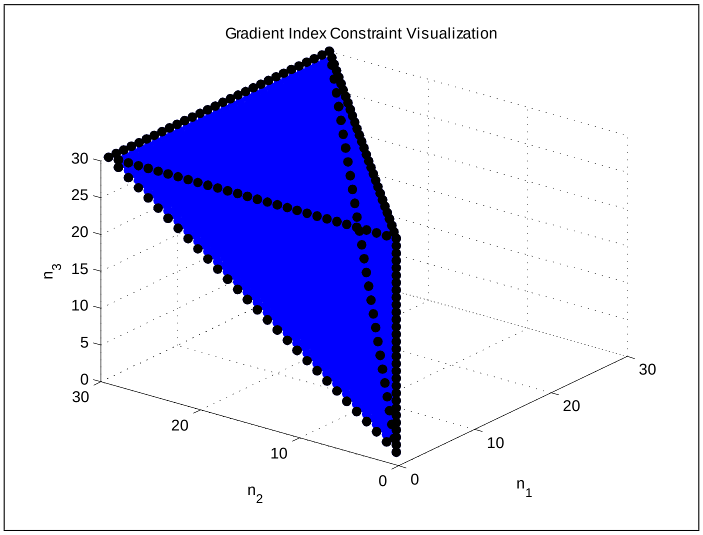

Figure 2-10: *Visualization of domain reduction mechanism of refractive index subset of the search space for the three layer case. For each inequality constraint, there is a reduction of the original domain by a factor of two, so in this case there are three factors leading to a reduction by a factor of one-eighth.*

Chapter 3

Experimental Realization of Optimized Broadband Omnidirectional Antireflection Coatings for Silicon

3.1 Overview

In this chapter, an important engineering system is experimentally demonstrated using the algorithms and theory developed in the previous chapter as a guide. More specifically, some efficient broadband omnidirectional antireflection coatings for silicon are demonstrated.

To harness most of the solar energy over the course of the day, the surface of a silicon solar cell must achieve low reflection at all relevant angles of incidence and over the entire wavelength range relevant to the operation of the cell. For design purposes in this chapter, the incident angle range is selected to be between 0 and 60 degrees since at higher angles, the intensity of sunlight is low and the light is more likely to be blocked from reaching the panel by obstacles such as trees. The wavelength range is selected to be between

400nm and 1600nm. Although silicon absorbs little to no radiation above 1100nm, higher wavelengths might be important for solar upconversion applications. Moreover, wavelength below 1600nm but above 1100nm may also be important for silicon photonic applications, such as reducing injection losses into silicon waveguides. We however also specialize our result for wavelengths between 400nm and 1100nm for the case of silicon photodetectors exclusively, while also weighing different incident angles and wavelengths differently depending on energy content of incident sunlight, for the case of silicon solar cells exclusively. The reader is referred to the beginning of the previous chapter for some discussion of current state-of-the-art in solving that problem.

In this chapter, a practical coating, approximating the global solution to this problem, is demonstrated to achieve average reflection that is competitive with the best devices that have been realized prior to this work. We believe this coating is the most widely applicable one of the lot, since it should work well for both monocrystalline and polycrystalline silicon. It is also realized using materials that have widely been reported to be susceptible to low-cost atmospheric deposition techniques such as spray-coating (vacuum systems are used in this work, mostly due to convenience).

The rest of this chapter is organized as follows. A representative sample of the apparatus used in the experiment is documented in the first section. Then, the experimental procedure is detailed in the second section. The experimental results obtained are presented and discussed in the third section. Some modifications to the design problem and corresponding experimental results are addressed in the subsequent section before the chapter is concluded.

3.2 Apparatus

The figures below visually catalog a representative sample of the apparatus used in the experiment. Figure 3-1 shows the control panel of the AJA International Inc ATC ORION sputtering system used in the majority of the experiment, located in the Microsystems

Technology Laboratories (MTL) at MIT (two other systems, one located in the Crystal Physics and Electroceramics Laboratory and the other in the Organic and Nanostructured Electronics Laboratory, were used as supplements). Figure 3-2 shows the vacuum chamber of the system. Figure 3-3 shows a close-up view of the inside of the vacuum chamber during deposition. Material targets (bottom) are bombarded by high energy particles (visible as plasma in the image) leading to the ejection of particles from the target and unto the rotating silicon wafer substrate. The setup used to characterize wide angle reflection (between 20 degrees and 70 degrees, the bounds being hardware constraints on the equipment) from the device prototype is shown in Figure 3-4. This setup is a VARIAN Cary 500 Spectrophotometer, fitted with one of its variable angle specular reflectance accessories (VASRA), located in the Center for Material Science and Engineering (CMSE) at MIT. A p16 stylus profilometer used to determine the thicknesses of single films mechanically (for refractive index characterization, once a step edge had been appropriately created), located in CMSE, is shown in Figure 3-5. A Filmetrics reflectometer (model F20), used to extract refractive index values from single films whose thickness had already been determined using profilometry (mathematically, this step amounts to reducing the number of parameters for the relevant inverse problem thereby making it more identifiable), located in MTL, is shown in Figure 3-6 (a model F40 located in CMSE was used as a supplement). It performs this extraction by fitting a model to normal incidence reflection from the film. This tool was also used to characterize the normal incidence reflection from the device, as documented in the section after the next. A spectrophotometer made by Aquila instruments (nkd-8000), which is not shown, was used to check for consistency in wide angle measurement data from the Cary spectrophotometer at 30 degrees incident radiation for both polarizations of light.

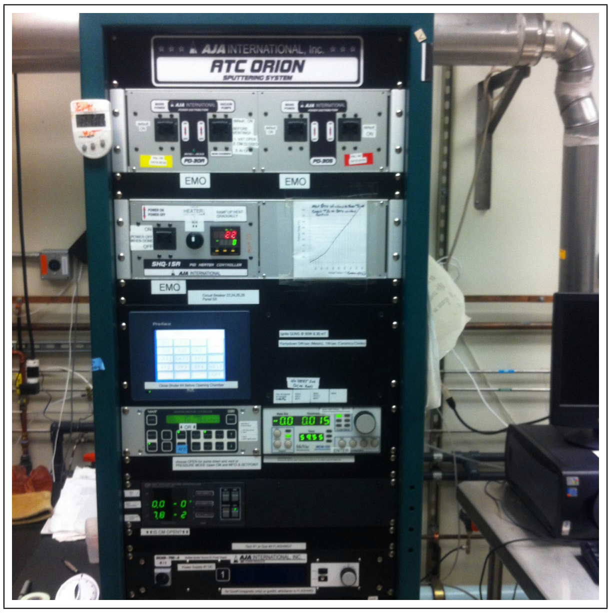

Figure 3-1: *AJA sputtering system control panel.*

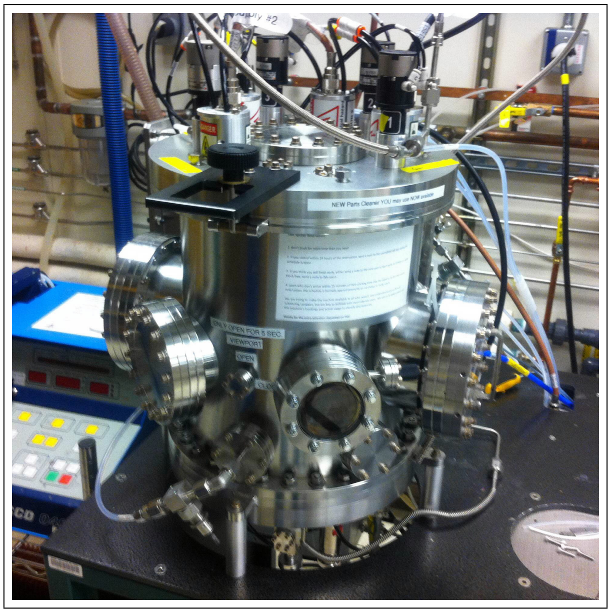

Figure 3-2: *AJA sputtering system vacuum chamber.*

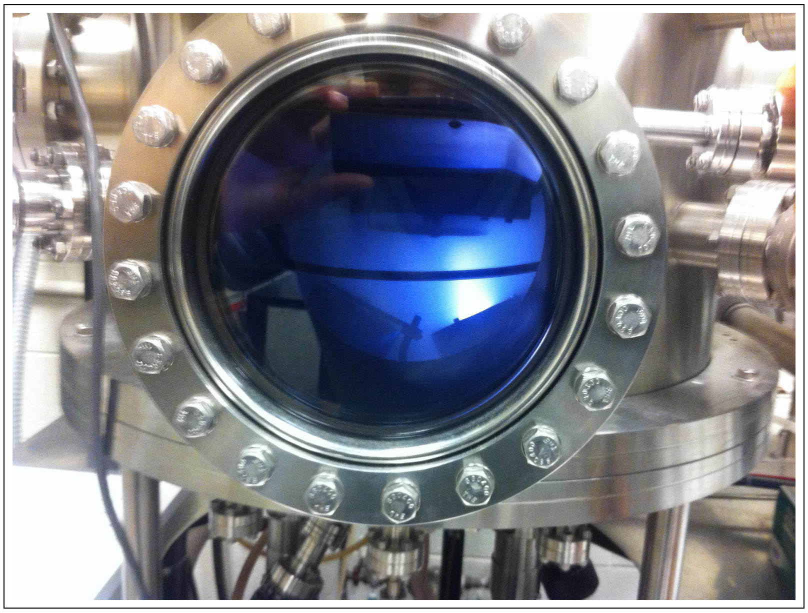

Figure 3-3: *Closeup view of a vacuum chamber during deposition, showing plasma emanating from target onto a silicon substrate.*

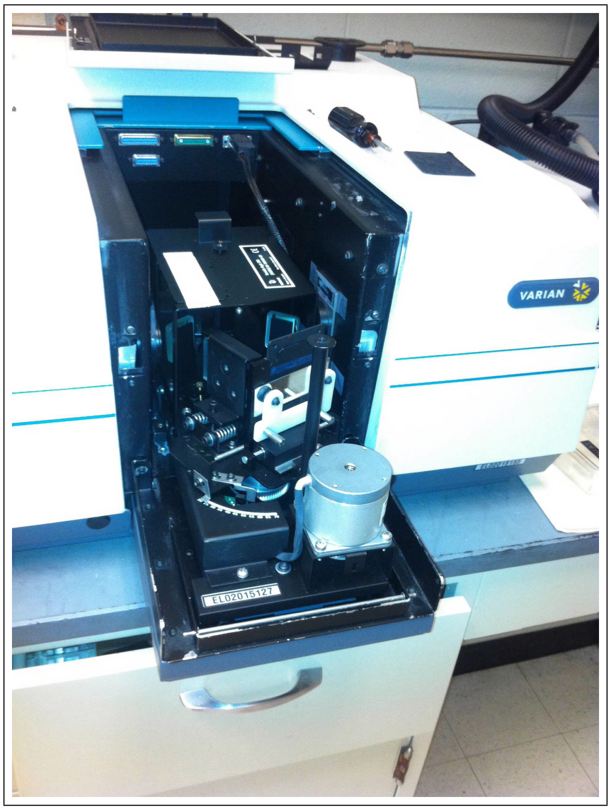

Figure 3-4: *VARIAN Cary 500 wide angle measurement setup.*

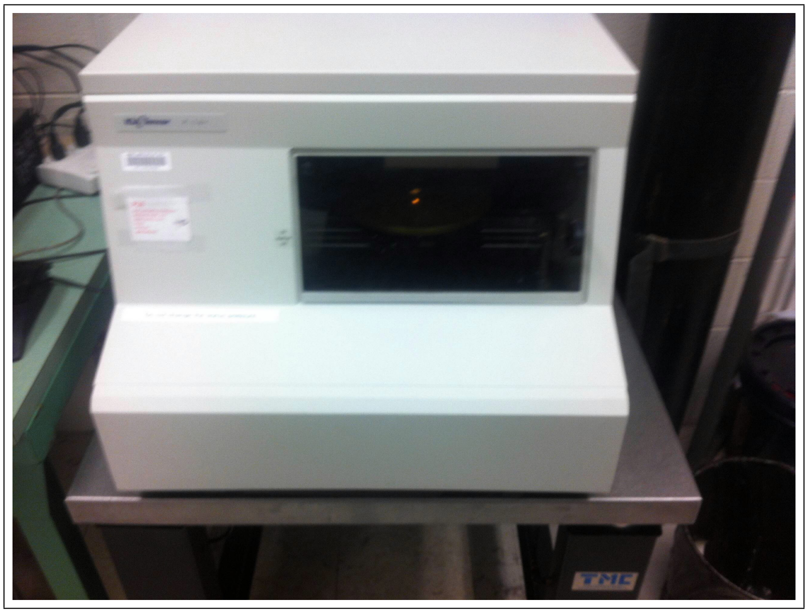

Figure 3-5: *p16 stylus profilometer.*

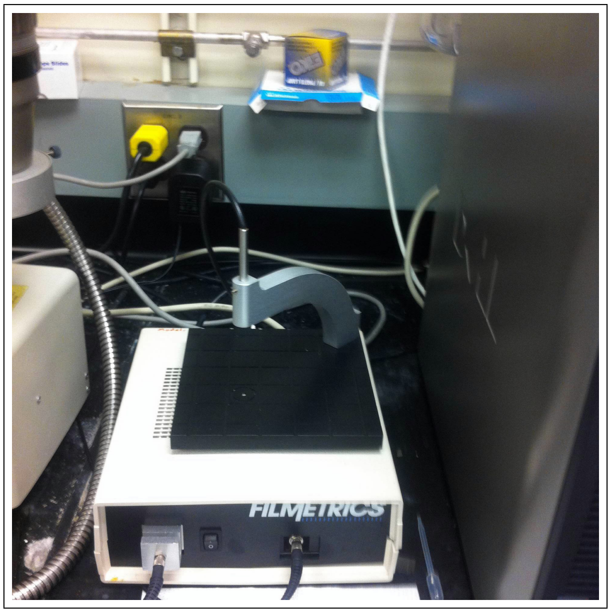

Figure 3-6: *Filmetrics reflectometer.*

3.3 Experimental Procedure

Material targets (Y_2O_3, TiO_2 and MgF_2) were circular with radius of 25mm. These were obtained from the Kurt J. Lesker company and had a 99.9% purity. Silicon wafers were obtained from MTL. They were not Piranha cleaned with HF prior to deposition as simulations revealed that the approximately 2nm native oxide layer only leads to around 0.2% change in reflection, which was considered to be negligible given big picture values attainable and not worth the HF safety hazard and extra cost. Targeted deposition times were determined iteratively using profilometry on single layers. TiO_2 was deposited in Argon at 500 degrees Celsius substrate temperature (this was done to achieve a significantly higher index than would be achieved at room substrate temperature, i.e., 2.33) and at 160 Watts applied power. This was done for 46 minutes to achieve the required thickness on the system described (the targeted thicknesses are given precisely in the previous chapter). During the characterization process, the materials were assumed to be nondispersive and nonabsorbing, consistent with the modeling assumptions in the previous chapter. Refractive index of the deposited TiO_2 was measured to be 2.55. Y_2O_3 was deposited in Argon at room substrate temperature for 64 minutes at 140 Watts applied power to achieve the desired thickness. Index was measured to be 1.90. MgF_2 was deposited in 25% O_2 and 75% Argon at 180 Watts at room substrate temperature for a duration of 7 hours and 18 minutes. The index was measured to be 1.40. It is hereby noted that the introduction of oxygen is necessary to eliminate absorption of sputtered MgF_2 (without oxygen, the extinction coefficient was measured to be approximately 0.13, which is significant), but it does increase the deposition time and thereby processing cost. Accepting some absorption for a lower cost might be worthwhile in practical scenarios when sputtering needs to be used. It is also worthwhile to note that relatively high deposition power is necessary, to keep targets at a relatively high temperature and induce sputtering in the form of molecules [16], otherwise sputtering in the form of atoms may occur leading to a reaction of magnesium with oxygen and the formation of MgO (the resulting film has a significantly higher index than what is needed, which was experienced during process development described in this work). During the characterization process, all three materials were found to be nonabsorbing.

3.4 Results and Discussion

In Figure 3-7 can be seen the normal incidence reflection from our device. There is some discrepancy between the experimental data and model-based prediction of the experimental result (labelled as "Simulated Experimental Reflection" in that figure - this is the model prediction for refractive index figures listed in the previous section together with the thicknesses $130nm$, $86nm$ and $46nm$, as determined by reflectometry and confirmed by transmission electron microscopy, as shown in Figure 3-8). This difference can probably be attributed to some dispersion in the materials and some characterization error (recall that the thin-film materials were treated as nondispersive nonabsorbing in the previous chapter). There is clearly a discrepancy between experimental reflection of 3.84% at normal incidence and the theoretically attainable 1.51%. This difference can be attributed to the significant discrepancy between targeted and attained refractive indices, as well as the significant error in thicknesses (the worst thickness error, in the TiO_2 layer thickness, is $15nm$). Note, however, that the performance of 3.84% is competitive with the nanotech result of 3.79% (recall also that the nanotech solution is less practical). This appears to suggest that the design is quite robust to experimental error, i.e., it has good sensitivity properties and shows good performance despite significant experimental error.

The wide angle reflection measurements are shown in Figures 3-9 and 3-10. This is done between angles of 20 degrees and 70 degrees (not 0 and 60) as dictated by constraints on the wide angle characterization hardware. This data shows that the excellent performance of the coating is maintained at even higher angles than were designed for. Note the exceptionally good performance for p polarized incident light. An average reflection of 4.2% is achieved. In contrast, the seven layer nanoporous coating achieves

3.79% reflection on average. These figures are strongly competitive with currently best-known solutions, such as [36], which quotes figures around 3.9% at normal incidence in the range 400 to 700 nm and 6.1% in the same range but over incident angles between 40 and 90 degrees. Again, this competing three layer design is made with impractical materials. Finally, we show what the silicon surface looks like visually in Figure 3-11, confirming that it looks "dark" to the eye, relative to untreated silicon, indicating low reflection visually.

While these preliminary results are promising, we clearly need to improve them. The first obvious factor that needs to be addressed is thickness control. The second clear factor is the discrepancy in matching practical refractive indices to the ideal ones. The third factor is the nondispersive assumption of the modeling process in the previous chapter. Moreover, we want to see how much more performance can be attained by narrowing the wavelength range to $[400nm, 1100nm]$ and weighting different wavelengths and incident angles differently depending on energy content. The next section reports on experiments we did to address these issues.

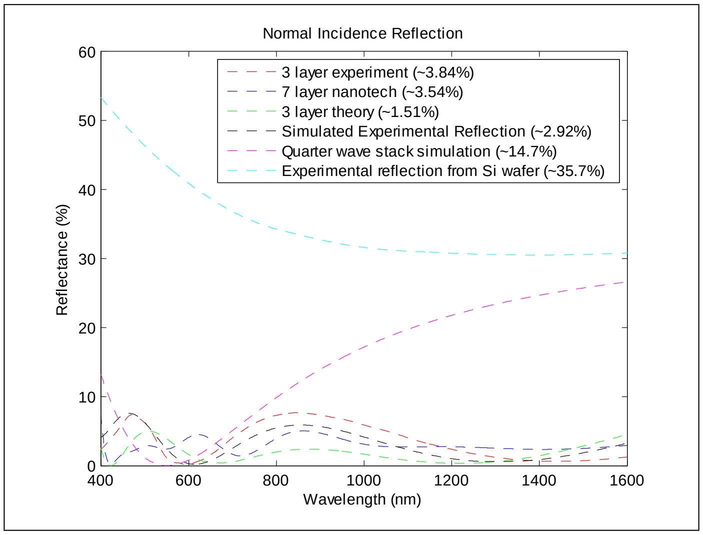

Figure 3-7: *Normal incidence reflection from antireflection coating and relevant comparisons.*

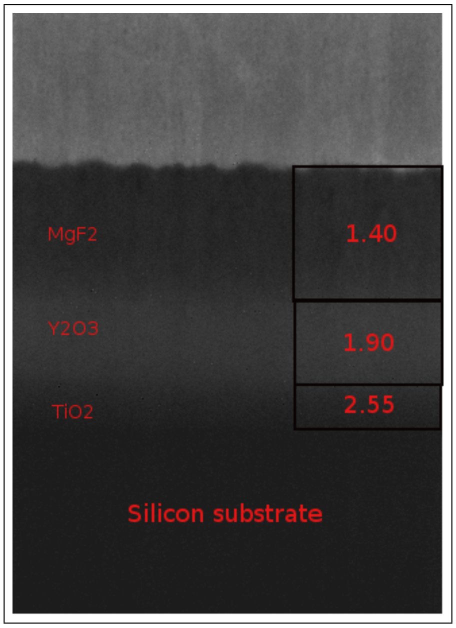

Figure 3-8: *Transmission electron microscopy cross-section of coating.*

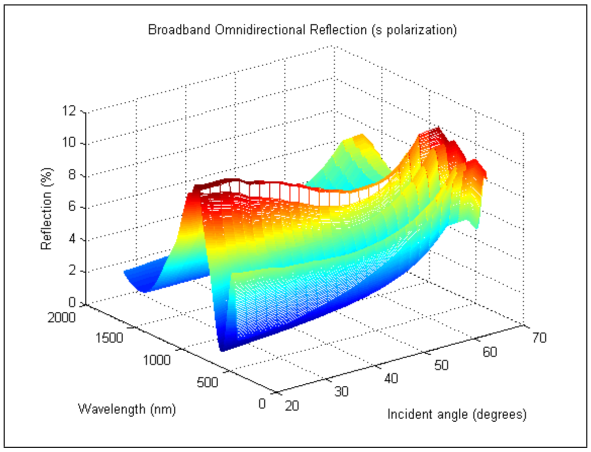

Figure 3-9: *Wide angle reflection from antireflection coating for s polarization of incident radiation.*

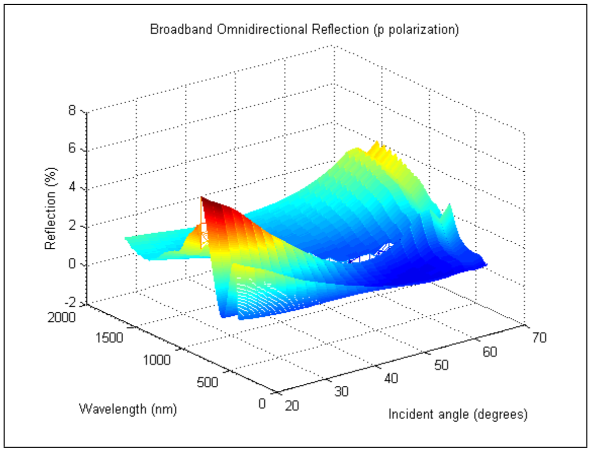

Figure 3-10: *Wide angle reflection from antireflection coating for p polarization of incident radiation.*

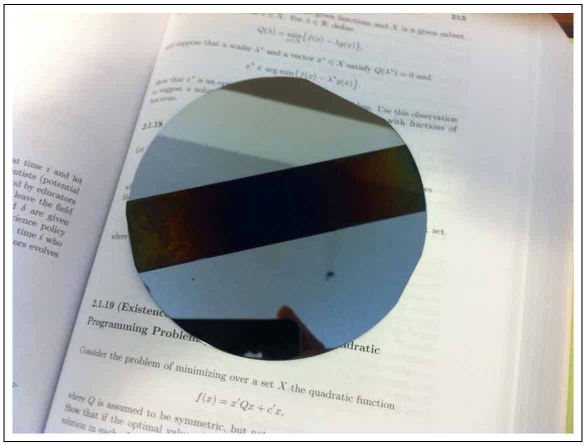

Figure 3-11: *Image of surface of a silicon wafer coated with first prototype. Note how dark the surface looks compared to the uncoated silicon wafer underneath.*

3.5 Modifications and Improvements

Wavelengths were first limited to the range between $400nm$ and $1100nm$, for the purposes of a general silicon photodetector, and the optimization procedure was reran. This yielded the solution vector

$$\mathbf{p}_{opt} = \begin{bmatrix} 1.15 & 1.66 & 2.60 & 139 & 87.3 & 56.2 \end{bmatrix},$$

which corresponds to the average reflectance value of 1.02%. Matching practical materials with closest refractive indices, we get MgF_2, Al_2O_3 and rutile TiO_2. In order to achieve better thickness control, fabrication was moved to the sputtering system located in the Organic and Nanostructured Electronics Laboratory. This system is equipped with a quartz crystal monitor (QCM) for finer thickness monitoring but does not possess a substrate heating capability. To address this drawback, we attempted post deposition annealing of TiO_2 at a variety of temperatures (100, 200, 300, 400 and 500 degrees Celsius), with the maximum index attained at 300 degrees. MgF_2 was deposited using thermal evaporation to explore better ways to control absorption. Refractive indices were characterized as a function of wavelength to account for dispersion using a spectroscopic ellipsometer. Before annealing, the index of TiO_2 was measured to be 2.49 on average, increasing to 2.52 after annealing. Clearly, the index of TiO_2 was found to be relatively high on this system (which is probably why annealing only had a marginal effect). The indices of MgF_2 and Al_2O_3 were measured to be 1.38 and 1.68 on average respectively. Every material took under an hour to deposit. All materials were found to be nonabsorbing. Annealing was eliminated from the process to save cost and time. The optimization was reran with indices constrained to these values (only the thicknesses varied) to yield the solution vector

$$\mathbf{p}_{opt} = \begin{bmatrix} 1.38 & 1.68 & 2.49 & 79.2 & 42.4 & 53.7 \end{bmatrix},$$

and a merit function (average reflection) value of 2.13%. A stack was deposited to approximate this, its normal incidence reflection and wide angle reflection (for a subset of the wavelength range, for practical reasons) shown in Figures 3-12 and 3-13 respectively. Visual inspection is shown in Figure **??**. While the surface indeed looks very dark to the naked eye, hints of violet are consistent with the rapid increase in reflectance below 460 nm. Thicknesses were measured to be $\begin{bmatrix} 74.9 & 44.0 & 58.8 \end{bmatrix}$. It is clear that while this device is significantly better than the first prototype described in the previous section, in terms of thickness control (maximum error being under 5nm) and match between theory and experiment, we can do even better. Modeling was adjusted to account for dispersion. The indices for the three materials, characterized as a function of wavelength using spectroscopic ellipsometry, are shown in Figure 3-14. The design was specialized for solar energy as follows. Wavelengths were weighted using the well-known AM1.5 photon flux spectrum to account for different levels on energy in terrestrial sunlight and incident angles were weighted using the benchmark SOLIS model [1]. The SOLIS model incorporates the sinusoidal variation of energy during the day, with light at higher angles traveling through a longer atmospheric path length and thereby being even further attenuated. With all this incorporated, the problem was again reoptimized. This yielded the thickness vector $\begin{bmatrix} 83.5 & 39.8 & 51.6 \end{bmatrix}$ with the corresponding weighted merit function value being 2.43%. The stack was deposited, with the QCM tooled immediately prior to the deposition, and the resulting thicknesses were measured using spectroscopic ellipsometry and confirmed using stylus profilometry. The resulting thickness vector was found to be $\begin{bmatrix} 84.6 & 39.8 & 51.3 \end{bmatrix}$. Normal incidence reflection is shown in Figure 3-16, broadband omnidirectional reflection in Figure 3-17 and visual appearance in Figure 3-18. Near-perfect fit between theory and experiment is attained and the surface of the silicon (quite literally) looks black visually. Note that the violet coloring is gone since the peak is significantly sharper. Moreover, the rapid rise contributes little to the merit function, since the AM1.5 spectrum decreases rapidly in that regime. Next, we compare our device experimentally to the SunPower texture, which is the state-of-the art texturing solution for solar energy. To give a sense of the texture shape, we show an atomic force microscopy (AFM) image of the surface roughness in Figure 3-19. Then, we measure diffuse reflection from the textured surface using an integrating sphere, and compare it to the experimental reflection from our device at 30 degree incident radiation. The result is shown in Figure 3-20, showing that our result is significantly better than the state-of-the-art texturing approach. Finally, we reoptimize theoretically up to 90 degrees incident radiation. This yields the thickness vector $\begin{bmatrix} 83.7 & 41.3 & 51.7 \end{bmatrix}$ and a corresponding merit function value of 2.86%. This shows that higher angles are only marginally important.

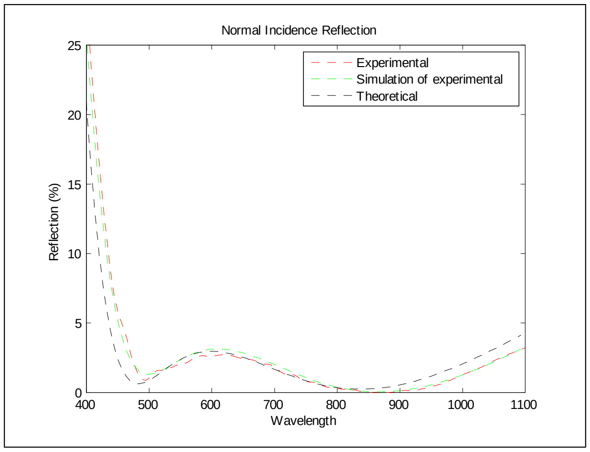

Figure 3-12: *Normal incidence reflection from next-to-final prototype.*

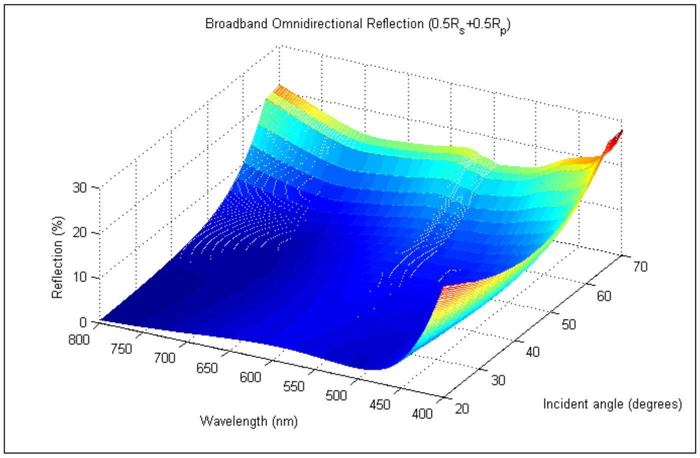

Figure 3-13: *Broadband omnidirectional reflection from next-to-final prototype.*

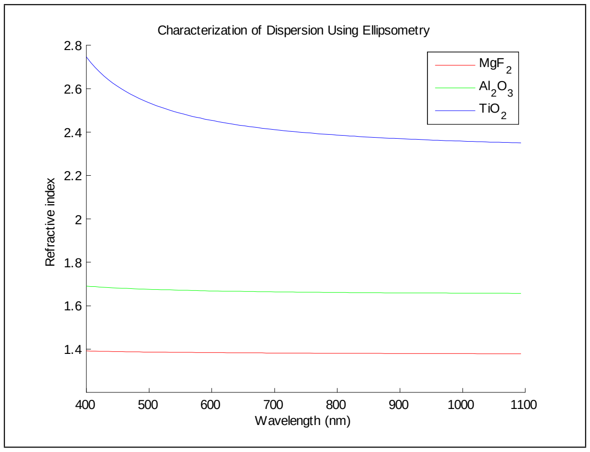

Figure 3-14: *Characterization of dispersion using ellipsometry.*

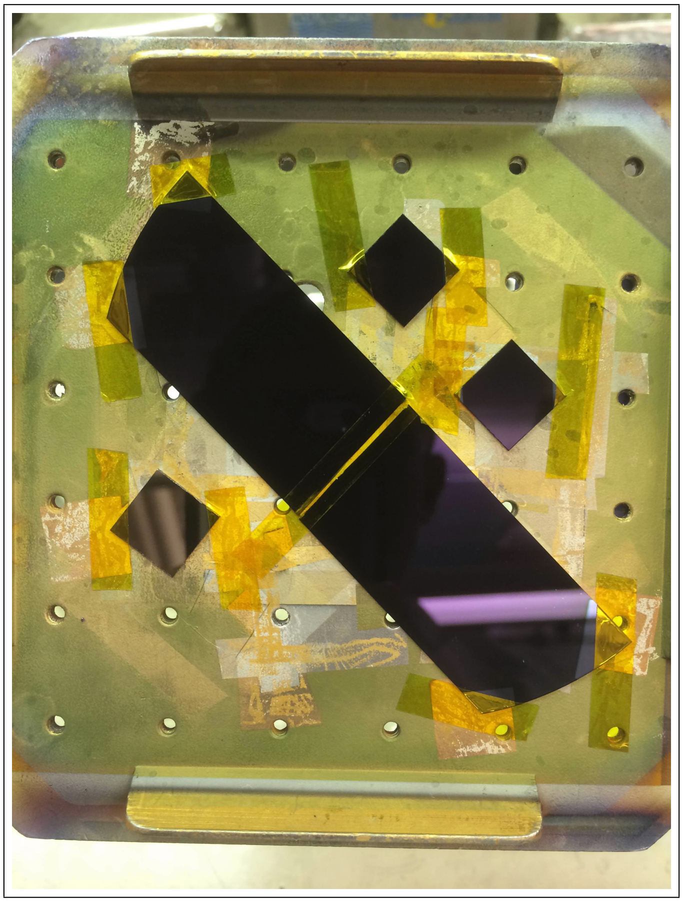

Figure 3-15: *Image of surface of a silicon wafer coated with next-to-final prototype.*

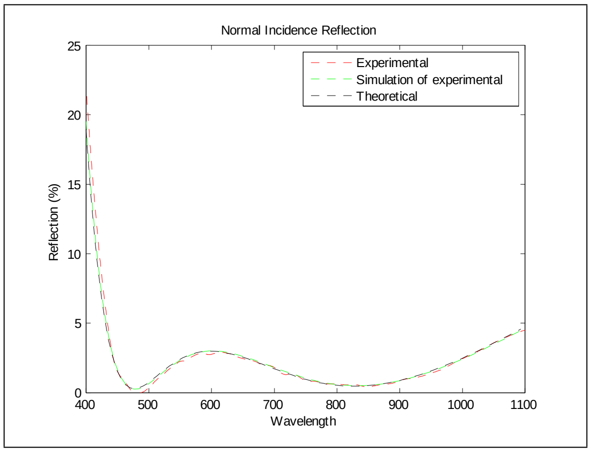

Figure 3-16: *Normal incidence reflection from final prototype.*

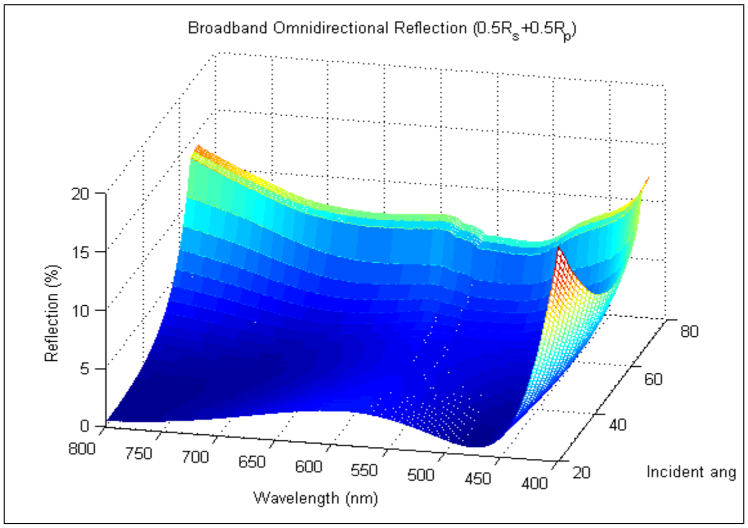

Figure 3-17: *Broadband omnidirectional reflection from final prototype.*

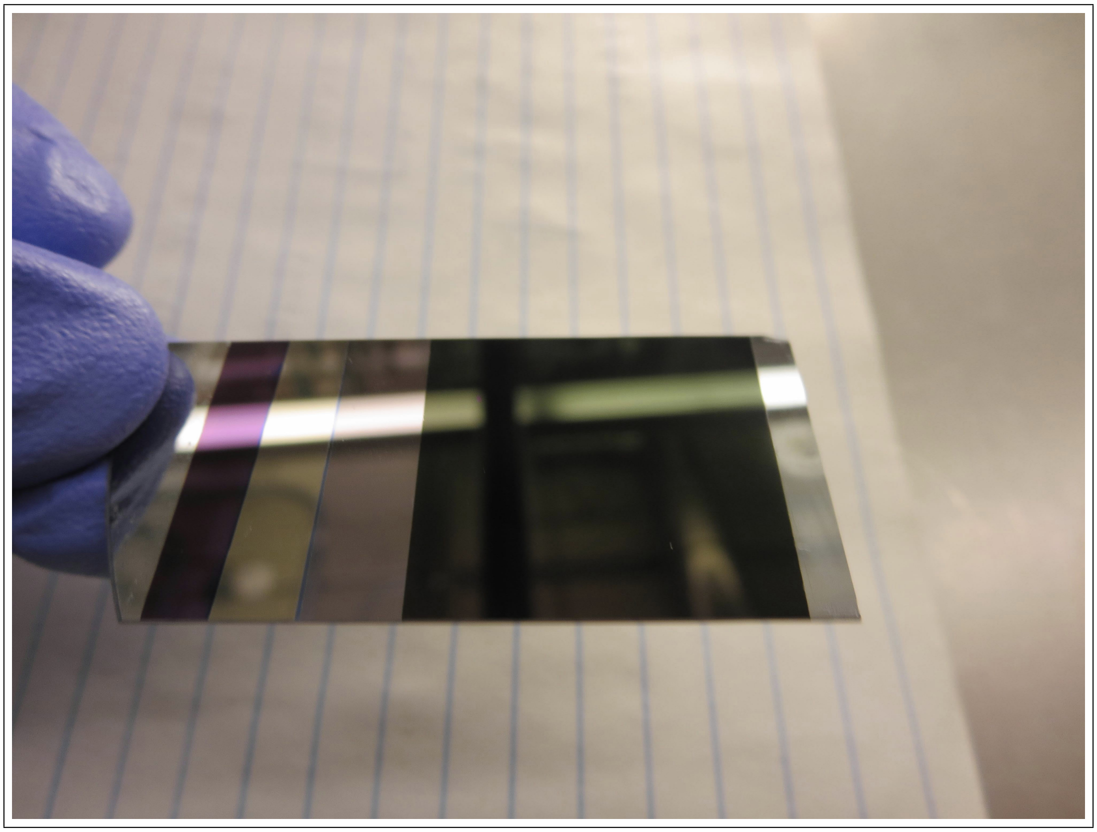

Figure 3-18: *Image of surface of a silicon wafer coated with final prototype. Note how dark the surface (on the right) looks compared to the silicon wafer underneath. On the left is the step edge system used to compare thicknesses of each layer mechanically using stylus profilometry*

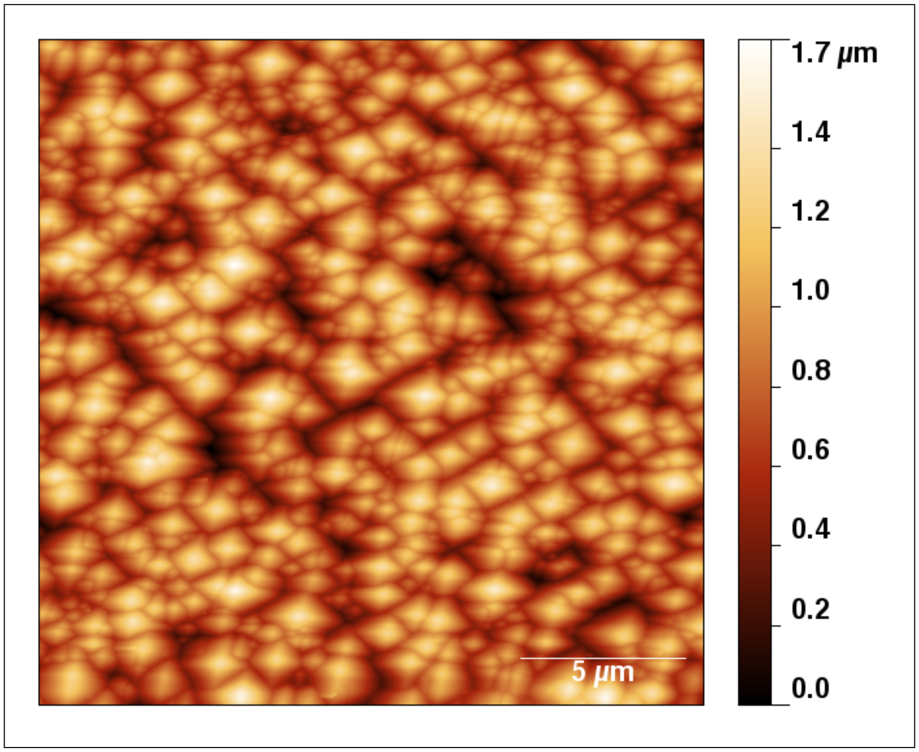

Figure 3-19: *AFM of SunPower texture*

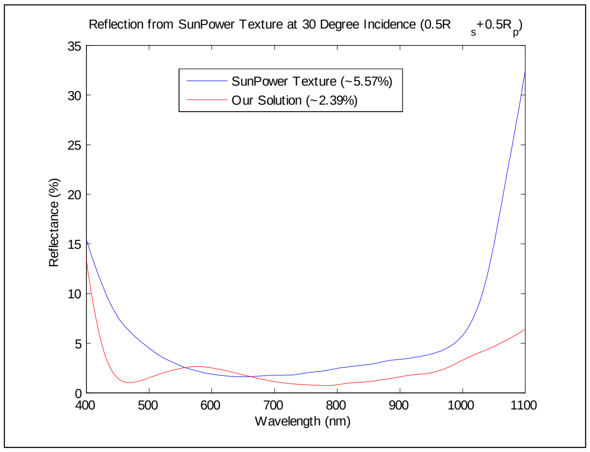

Figure 3-20: *Comparison of SunPower texture reflection to the experimental reflection from our device, at the incident angle of 30 degrees.*

3.6 Conclusion

Arguably the best broadband omnidirectional antireflection coating technology for silicon solar energy to date has been demonstrated in this chapter. This claim can be made because this device is the best *practical* result in the literature. Moreover, it appears that the coating is relatively robust to experimental error, the competitiveness with existing solutions is maintained despite significant experimental error. Not only is this device practical in the sense that it uses real materials, the realization technology was chosen to be a mixture of RF sputtering and thermal evaporation because it (together with the experimental process that was developed) scales up readily industrially and is susceptible to factory automation. Thus, the industrial infrastructure necessary to build this device on real solar cells on a large scale is already in place. The experimental process can be optimized further to lower both the cost and the reflection value. One way to achieve this is to increase the index of sputtered TiO_2 further by increasing the substrate temperature during deposition (indices as high as 2.8 have been reported at 600 degrees substrate temperature [26]). Cost can be lowered by optimizing the experimental process to lower sputtering time for the MgF_2, by omitting the introduction of O_2 in its deposition and accepting some absorption. Alternatively, thermally evaporated films, deposited relatively quickly, do not appear to be absorbing.

Chapter 4

Bounding the Solutions of Parametric Weakly Coupled Semilinear Parabolic Partial Differential Equation Systems

4.1 Overview

In this chapter, two novel techniques for bounding the solutions of parametric weakly-coupled second-order semilinear parabolic PDEs are developed. The first provides a theorem to construct interval bounds while the second provides a theorem to construct lower bounds convex and upper bounds concave in the parameter vector. The convex/concave bounds (which we alternatively refer to as *relaxations*) can be significantly tighter than the interval bounds due to the wrapping effect suffered by interval analysis in dynamic systems. Both types of bounds are computationally cheap to construct, requiring solving auxiliary systems twice and four times larger than the original system respectively. Illustrative numerical examples of bound construction and use for deterministic global optimization within a simple serial branch and bound algorithm, implemented numerically using interval arithmetic and a generalization of McCormick's relaxation technique respectively, are presented. The bounds may also be applicable to rigorously quantifying parametric uncertainty of problems within this class. To the best of our knowledge, this is the first example of such bounds for this class of problems. The particular motivation that drove this work is rigorous optimization of semiconductor problems (based on the drift-diffusion-Poisson system of equations). Particular examples of such problems include recovery of inorganic semiconductor doping profiles from data, design of inorganic semiconductor doping profiles to minimize leakage currents and thickness optimization of bulk heterojunction organic photovoltaic devices. More generally, problems within the important class of *reaction-diffusion* systems with diffusion coefficients that are not state dependent may be optimized rigorously with these tools.

4.2 Introduction

Reaction-diffusion systems with diffusion coefficients that are not state dependent can be modeled using semilinear parabolic partial differential equations (PDEs) (state dependence of diffusion coefficients would render the system *quasilinear*). An important and well-known example is the heat equation with source term nonlinear in the temperature. One may be faced with the task of fitting such a model to experimental data by formulating an optimization problem. The resulting optimization problem is typically nonconvex, making it desirable to develop a global optimization method for problems involving this class of differential equations, to ensure that the best possible fit can be obtained and that the descriptive power of this class of important models can be robustly evaluated. Alternatively, one may be interested in coming up with a global solution to a design problem involving this important class of differential equations. Unlike stochastic global optimization methods, such as genetic algorithms and simulated annealing, deterministic global optimization using a branch-and-bound algorithm can provide a guarantee that the global optimum has been identified to within a finite tolerance (governed by practical considerations such as modeling error). This is achieved when the algorithm converges, since it represents a constructive procedure for locating the global optimum. The critical component of such an algorithm is the construction of parametric bounds on the PDE solution.

The problem of bounding the solutions of parametric ordinary differential equations (ODEs) has received much attention in the literature. Harrison [12] described a technique to construct interval pointwise in time bounds on the solutions of parametric ODEs. He used an existence-comparison result due to Walter [47] to achieve this. Unfortunately, most real systems do not satisfy a stringent condition of quasimonotonicity (which requires the off-diagonal entries of each source function's Jacobian to preserve its sign in its domain), in which case these bounds are often too weak to be useful. This is due to the wrapping effect of interval analysis [12] [28], the difficulty stemming from employing bounds parallel to the coordinate axes. This motivated the demonstration of the construction of affine in the parameter bounds on the solutions of parametric ODEs in [44]. These bounds employ McCormick's relaxation technique, are significantly stronger under nonquasimonotonicity and are trivially both convex and concave in the parameter. Unfortunately, they include arbitrary user-specified components that directly influence the quality of the bounds and may be unsuitable under high nonlinearity in the parameter [42]. This motivated the construction of nonlinear convex lower and concave upper in the parameter bounds for ODEs using a generalization of McCormick's relaxation technique in [42]. It is also possible to suppress the wrapping effect using validated integrators based on Taylor models [30], but such an approach is significantly more computationally expensive than the aforementioned methods (due to the relatively high computational cost of Taylor models), which only require the regular integration of auxiliary systems four times larger than the original system.

In this chapter, two novel techniques for bounding the solutions of parametric weakly-coupled second-order semilinear parabolic PDEs are developed. The first provides a theorem to construct interval bounds while the second provides a theorem to construct lower bounds convex and upper bounds concave in the parameter. The convex/concave bounds can be significantly tighter than the interval bounds due to the wrapping effect suffered by interval analysis in dynamic systems. Both types of bounds are computationally cheap to construct, requiring solving auxiliary systems twice and four times larger than the original system respectively. Illustrative numerical examples of bound construction and use for deterministic global optimization within a simple serial branch and bound algorithm, implemented numerically using interval arithmetic and a generalization of McCormick's relaxation technique, are presented. The bounds may also applicable to quantifying parametric uncertainty of problems within this class. To the best of our knowledge, this is the first example of such bounds for this class of problems. The particular motivation that drove this work is optimization of semiconductor problems (which are based on the drift-diffusion-Poisson system of equations). Particular examples of such problems include recovery of inorganic semiconductor doping profiles from data, design of inorganic semiconductor doping profiles to minimize leakage currents (this is arguably the most important problem in the semiconductor industry) and thickness optimization of bulk heterojunction organic photovoltaic devices (we hereby note that bilayer organic photovoltaic devices are described by a quasilinear parabolic PDE system due to the electric field dependence of the diffusion coefficients, whereas in the bulk heterojunction case detailed simulations and some experimental evidence have revealed that electric field remains constant in the device [18] [17]). More generally, problems within the important class of reaction-diffusion systems, where the diffusion coefficients are not state dependent, may be optimized with these tools.

We note, before proceeding, that all the results we use to prove the theorems in this chapter, are taken from the well-studied method of lower and upper solutions (much like what Harrison did, but broader since we do not address interval bounds only but also relaxations). This was done on purpose, we believe that the immense body of work associated with this method, coupled with the solution strategies we develop here, can provide for a solution program to extend these bounds systematically into other classes of differential equations.

The rest of this chapter is organized as follows. In Section 4.3, the problem is defined mathematically and the construction of the different types of bounds is motivated by outlining how they are used by a simple deterministic branch-and-bound global optimization procedure. In Section 4.4, theorems for constructing the different types of bounds are formulated and proved and simple illustrative analytic examples are presented. Finally, Section 4.5 presents a pair of simple numerical examples.

4.3 Preliminaries

4.3.1 Parametric Semilinear Parabolic PDE System

Theory is developed in one spatial dimension for simplicity, but the same results can be directly extended to multiple spatial dimensions. Denote the spatial coordinate by $x \in \mathbb{R}$. Let Ω be a bounded or an unbounded open spatial domain in $\mathbb{R}$, with boundary $\partial\Omega$ and closure $\bar{\Omega}$. For any $t_f > 0$, denote the temporal domain by $T \equiv (0, t_f]$ and the temporospatial domain by $Q \equiv \Omega \times T$, denoting its closure by $\bar{Q}$. Let $\Gamma \equiv \partial\Omega \times T$ and let $\mathbf{p} \in P \equiv \left[\mathbf{p}^L, \mathbf{p}^U\right]$ denote the parameter vector. Intervals between vector functions are componentwise and pointwise in their domain. Denote by $C\left(\bar{Q}\right)$ the space of functions continuous in $\bar{Q}$, and by $C^{m,l}\left(Q\right)$ the space of functions with derivatives up to m^{th} order with respect to (w.r.t.) x and up to l^{th} order w.r.t. t continuous in Q.

For each $\mathbf{p} \in P$, and every $i \in I_{\mathbf{u}} \equiv \{1, ..., n_{\mathbf{u}}\}$, define an operator as follows:

$$L_i u_{i,\mathbf{p}}\left(x,t\right) \equiv a_i\left(x,t\right) \frac{\partial^2 u_{i,\mathbf{p}}}{\partial x^2}\left(x,t\right) + b_i\left(x,t\right) \frac{\partial u_{i,\mathbf{p}}}{\partial x}\left(x,t\right), \forall \left(x,t\right) \in Q. \tag{4.1}$$

Then, define an operator as follows:

$$\mathcal{L}_i u_{i,\mathbf{p}}\left(x,t\right) \equiv \frac{\partial u_{i,\mathbf{p}}}{\partial t}\left(x,t\right) - L_i u_{i,\mathbf{p}}\left(x,t\right), \forall \left(x,t\right) \in Q. \tag{4.2}$$

If, for every $i \in I_{\mathbf{u}}$, a_i is positive in Q, the operators L_i and $\mathcal{L}_i$ are said to be elliptic and parabolic respectively (in multiple spatial dimensions, this requirement dictates that the corresponding matrix be positive-definite). Then, the coupled system of a finite number $n_{\mathbf{u}}$ of equations

$$\begin{aligned} \mathcal{L}_i u_{i,\mathbf{p}}(x,t) &= f_i\left(\mathbf{u}_{\mathbf{p}}(x,t), x, t, \mathbf{p}\right), \forall (x,t) \in Q, \\ \mathcal{B}_i u_{i,\mathbf{p}}(x,t) &= h_i(x,t,\mathbf{p}), \forall (x,t) \in \Gamma, \\ u_{i,\mathbf{p}}(x,0) &= u_{i,0}(x,\mathbf{p}), \forall x \in \Omega. \end{aligned} \tag{4.3}$$

is parabolic. This system is weakly-coupled in the sense that the coupling source function $\mathbf{f}$ does not depend on state spatial derivatives. Dependence of the state variable $\mathbf{u}_{\mathbf{p}} \in \mathbb{R}^{n_{\mathbf{u}}}$ on $\mathbf{p}$ is denoted by the subscript to indicate that it is implicit. $\mathcal{B}_i$ denotes a linear boundary operator of the form

$$\mathcal{B}_i u_{i,\mathbf{p}}(x,t) \equiv \alpha_i(x,t) \frac{\partial u_{i,\mathbf{p}}}{\partial \nu}(x,t) + \beta_i(x,t) u_{i,\mathbf{p}}(x,t), \forall (x,t) \in \Gamma, \tag{4.4}$$

with $\frac{\partial u_{i,\mathbf{p}}}{\partial \nu}(\cdot,t)$ for each $t \in T$ denoting the outward normal spatial derivative of $u_{i,\mathbf{p}}(\cdot,t)$ on $\partial\Omega$, and

$$\alpha_i(x,t) \geq 0,\ \beta_i(x,t) \geq 0,\ \alpha_i(x,t) + \beta_i(x,t) > 0,\ \forall (x,t) \in \Gamma. \tag{4.5}$$

Well-known assumptions required by the existence of a classical solution to (4.3) are made, the interested reader being referred to Section 2.1.1 of [34], for instance, for a detailed discussion. These are a set of continuity and consistency assumptions on the various functions involved, among which Hölder continuity in Q of order in $(0,1)$ is particularly important. For notational brevity, it is convenient to rewrite (4.3) as follows:

$$\begin{aligned} \mathcal{L}_i u_{i,\mathbf{p}} &= f_i\left(\mathbf{u}_{\mathbf{p}}, x, t, \mathbf{p}\right) \text{ in } Q, \\ \mathcal{B}_i u_{i,\mathbf{p}} &= h_i\left(x, t, \mathbf{p}\right) \text{ on } \Gamma, \\ u_{i,\mathbf{p}}\left(x, 0\right) &= u_{i,0}\left(x, \mathbf{p}\right) \text{ in } \Omega. \end{aligned} \tag{4.6}$$

4.3.2 A Simple Serial Branch-and-Bound Method

Here, a procedure for deterministic branch-and-bound global optimization is outlined, to motivate the constructions of the bounds on $\mathbf{u}_{\mathbf{p}}$. The classic reference for branch-and-bound theory is [15]. Consider an optimization problem of the form

$$\mathbf{p}_{opt} = \arg\min_{\mathbf{p}\in P}\left\{O\left(\mathbf{p}\right) = \sum_{(x,t)\in Q^m} \phi\left(\mathbf{u}_{\mathbf{p}}\left(x,t\right), x, t, \mathbf{p}\right)\right\}. \tag{4.7}$$

Here, O denotes a potentially nonconvex on P objective function, e.g. least squares or maximum likelihood in parameter estimation problems. Standard optimization software (e.g. $fmincon$ in MATLAB) can only yield a locally optimal solution $O\left(\mathbf{p}_{loc}\right)$ in the vicinity of the initial guess. A branch-and-bound algorithm can determine the globally optimal solution $O\left(\mathbf{p}_{opt}\right)$ to within some finite ϵ_O tolerance, by recursively bounding the solution on progressively smaller subintervals of the parameter space. A local optimizer can be used to obtain an upper bound by initializing it anywhere on the subinterval (another approach is to just evaluate the objective function anywhere on the subinterval). The corresponding lower bound can be obtained in one of two ways. If a pointwise in $\bar{Q}$ interval bound $\left[\mathbf{u}^L, \mathbf{u}^U\right]$ on $\mathbf{u}_{\mathbf{p}}$ is available, standard interval arithmetic [28] can be used to propagate it (along with P) through ϕ to obtain a corresponding interval bound $\left[O^L, O^U\right]$ on O. If a pointwise in $\bar{Q}$ bound $\left[\mathbf{u}_{\mathbf{p}}^{CV}, \mathbf{u}_{\mathbf{p}}^{CC}\right]$ on $\mathbf{u}_{\mathbf{p}}$ is available, with $\mathbf{u}_{\mathbf{p}}^{CV}$ being convex and $\mathbf{u}_{\mathbf{p}}^{CC}$ being concave on P, a generalization of McCormick's relaxation technique (which is discussed later on in this chapter) can be applied to obtain an interval bound $\left[O^{CV}(\mathbf{p}), O^{CC}(\mathbf{p})\right]$ on $O(\mathbf{p})$ for each $\mathbf{p} \in P$, with O^{CV} being convex and O^{CC} being concave on P. O^{CV} can then be locally optimized on the subinterval to yield the lower bound (this uses the well-known fact that the local minimum of a convex function is its global minimum). To aid visualization, the different types of bounds are illustrated in Figure 4-1 for a univariate objective function. A generalization of McCormick's relaxation technique will be used to construct $\left[\mathbf{u}_{\mathbf{p}}^{CV}, \mathbf{u}_{\mathbf{p}}^{CC}\right]$ later on in this paper, and will be discussed in more detail at that time. If the objective lower bound on any subinterval is higher than the least upper bound known so far (LUB, or incumbent), the global solution cannot exist on it and the subinterval is excluded from further consideration. If the least lower bound on the remaining subintervals (LRLB) is not within ϵ_O of the LUB, one subinterval is bisected on a uniformly randomly selected parameter into 2 intervals to be bounded and added to the active interval list. The process is initiated with P and continued until the LRLB is within ϵ_O of LUB. At this point, an optimal solution is available as the parameter corresponding to the LUB (this solution, by construction, is known to be within ϵ_O of the global solution). A sample illustrative iteration of the procedure, with the convex bound being employed to lower bound the objective on each subinterval, is shown in Figure 4-2. We see that we need to construct bounds for the PDE solution that are either intervals or convex lower and concave upper in the parameter (bounds also typically referred to as convex and concave relaxations, or simply relaxations, in the global optimization literature).

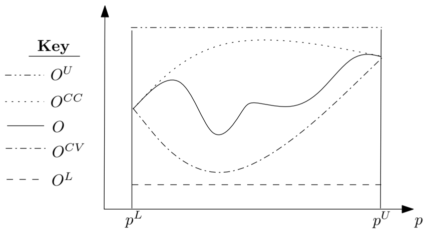

Figure 4-1: *Different types of bounds for the objective function O.*

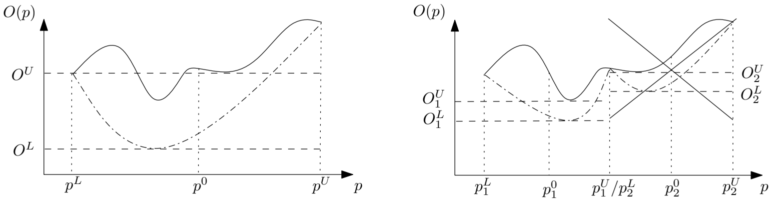

Figure 4-2: *An iteration of the branch-and-bound procedure.*

4.4 Bounds

4.4.1 Theorems

Here, key theorems used to construct the bounds, are stated and proved by drawing on some rather well-known results in the literature. The following existence-comparison theorem is key for the construction of the interval bounds. Unless made explicit otherwise, the order relation $\leq$ should henceforth be taken to be componentwise and pointwise in the domain for vector functions.

Theorem 1 *Consider (4.6) for some* $\mathbf{p} \in P$ *and assume that a pair of functions* $\mathbf{v}$ *and* $\mathbf{w}$ *in* $C^{2,1}(Q) \cap C(\bar{Q})$ *satisfy the following inequality* $\forall i \in I_{\mathbf{u}}$*:*

$$\begin{aligned} \mathcal{L}_i v_i &\leq f_i(\mathbf{z}, x, t, \mathbf{p})\big|_{\mathbf{z}\in[\mathbf{v},\mathbf{w}],\, z_i=v_i} \text{ in } Q, \\ \mathcal{B}_i v_i &\leq h_i(x, t, \mathbf{p}) \text{ on } \Gamma, \\ v_i(x, 0) &\leq u_{i,0}(x, \mathbf{p}) \text{ in } \Omega, \\ \mathcal{L}_i w_i &\geq f_i(\mathbf{z}, x, t, \mathbf{p})\big|_{\mathbf{z}\in[\mathbf{v},\mathbf{w}],\, z_i=w_i} \text{ in } Q, \\ \mathcal{B}_i w_i &\geq h_i(x, t, \mathbf{p}) \text{ on } \Gamma, \\ w_i(x, 0) &\geq u_{i,0}(x, \mathbf{p}) \text{ in } \Omega, \end{aligned} \tag{4.8}$$

with it being assumed that each f_i *is continuously differentiable in the state (with the derivative being bounded in* Q*), which implies that it is Lipschitz in the state, on* $[\mathbf{v}, \mathbf{w}]$*,* *i.e.,* $\exists K_i \in \mathbb{R}_+$ *such that*

$$\left|f_i\left(\mathbf{y}^1, x, t, \mathbf{p}\right) - f_i\left(\mathbf{y}^2, x, t, \mathbf{p}\right)\right| \leq K_i \sum_{j=1}^{n_{\mathbf{u}}} \left|y_j^1 - y_j^2\right|, \forall \left(\mathbf{y}^1, \mathbf{y}^2\right) \in [\mathbf{v}, \mathbf{w}] \times [\mathbf{v}, \mathbf{w}]. \quad (4.9)$$

Then, there exists a unique solution $\mathbf{u}_{\mathbf{p}}$ *to (4.6), and it is ordered as* $\mathbf{v} \leq \mathbf{u}_{\mathbf{p}} \leq \mathbf{w}$.

Proof. *Implicit in the statement of the theorem is that such a pair of functions is necessarily ordered as* $\mathbf{v} \leq \mathbf{w}$. *Hence, we first show that such a pair of functions is necessarily ordered as* $\mathbf{v} \leq \mathbf{w}$. *For this purpose, subtract the top half of (4.8) from the lower half to obtain the following inequality* $\forall i \in I_{\mathbf{u}}$*:*

$$\begin{aligned} \mathcal{L}_i\left(w_i - v_i\right) &\geq \left. f_i\left(\mathbf{z}, x, t, \mathbf{p}\right)\right|_{\mathbf{z} \in [\min(\mathbf{v},\mathbf{w}), \max(\mathbf{v},\mathbf{w})],\, z_i = w_i} \\ &\quad - \left. f_i\left(\mathbf{z}, x, t, \mathbf{p}\right)\right|_{\mathbf{z} \in [\min(\mathbf{v},\mathbf{w}), \max(\mathbf{v},\mathbf{w})],\, z_i = v_i} \ \textit{in } Q, \\ \mathcal{B}_i\left(w_i - v_i\right) &\geq 0 \ \textit{on } \Gamma, \\ \left(w_i - v_i\right)(x, 0) &\geq 0 \ \textit{in } \Omega. \end{aligned} \quad (4.10)$$

Then, observe that the following inequality is implied:

$$\begin{aligned} \mathcal{L}_i\left(w_i - v_i\right) &\geq \left.\begin{pmatrix} \left. f_i\left(\mathbf{z}, x, t, \mathbf{p}\right)\right|_{z_i = w_i} \\ - \left. f_i\left(\mathbf{z}, x, t, \mathbf{p}\right)\right|_{z_i = v_i} \end{pmatrix}\right|_{\mathbf{z} \in [\min(\mathbf{v},\mathbf{w}), \max(\mathbf{v},\mathbf{w})]} \ \textit{in } Q, \quad (4.11) \\ \mathcal{B}_i\left(w_i - v_i\right) &\geq 0 \ \textit{on } \Gamma, \\ \left(w_i - v_i\right)(x, 0) &\geq 0 \ \textit{in } \Omega. \end{aligned}$$

Whenever $w_i = v_i$, *the right hand side of 4.11 is* 0. *Now, whenever* $w_i \neq v_i$, *apply the mean value theorem to deduce the following inequality:*

$$\begin{aligned} \mathcal{L}_i\left(w_i - v_i\right) &\geq \left.\left.\frac{\partial f_i}{\partial z_i}\left(\mathbf{z}, x, t, \mathbf{p}\right)\right|_{z_i = \eta} \left(w_i - v_i\right)\right|_{\mathbf{z} \in [\min(\mathbf{v},\mathbf{w}), \max(\mathbf{v},\mathbf{w})]} \ \textit{in } Q, \quad (4.12) \\ \mathcal{B}_i\left(w_i - v_i\right) &\geq 0 \ \textit{on } \Gamma, \\ \left(w_i - v_i\right)(x, 0) &\geq 0 \ \textit{in } \Omega, \end{aligned}$$

where η is an intermediate value (pointwise) between w_i and v_i for each $\mathbf{z}$ in $[\min(\mathbf{v},\mathbf{w}),\max(\mathbf{v},\mathbf{w})]$. Now, since $\frac{\partial f_i}{\partial z_i}$ is bounded in Q for each $\mathbf{z}$ in $[\min(\mathbf{v},\mathbf{w}),\max(\mathbf{v},\mathbf{w})]$. Then, by a positivity lemma for (4.6) (see Lemma 2.2.1 in [34]) that is itself a consequence of the maximum principle for the parabolic operator (4.2), it follows that $w_i - v_i \geq 0, \forall i \in I_{\mathbf{u}}$, i.e., that $\mathbf{v} \leq \mathbf{w}$. Having shown this, the existence of a unique solution $\mathbf{u_p}$ ordered as $\mathbf{v} \leq \mathbf{u_p} \leq \mathbf{w}$ is a direct consequence of a theorem due to C.V. Pao (see Theorem 8.9.3 in [34]). That theorem says that if $\mathbf{v} \leq \mathbf{w}$, then a unique solution $\mathbf{u_p}$ ordered as $\mathbf{v} \leq \mathbf{u_p} \leq \mathbf{w}$ exists, i.e., having shown the order $\mathbf{v} \leq \mathbf{w}$ here, the order $\mathbf{v} \leq \mathbf{u_p} \leq \mathbf{w}$ for the unique solution $\mathbf{u_p}$ follows from C.V. Pao's theorem. He proved that theorem by constructing a monotone sequence of functions, with $\mathbf{v}$ and $\mathbf{w}$ as the initial condition for the iteration, to converge to $\mathbf{u_p}$ from above and below. ■

The following theorem is key for the construction of the convex/concave bounds.

Theorem 2 *Consider the pair of scalar PDEs:*

$$\begin{aligned} \mathcal{L}_i u_{i,\mathbf{p}} &= f_i(x,t,\mathbf{p}) \ in\, Q, \\ \mathcal{B}_i u_{i,\mathbf{p}} &= h_i(x,t,\mathbf{p}) \ \ on\ \Gamma, \\ u_{i,\mathbf{p}}(x,0) &= u_{i,0}(x,\mathbf{p}) \ \ in\ \Omega, \end{aligned} \tag{4.13}$$

i.e., $i \in \{1,2\}$. Assume that, for some $\mathbf{p} \in P$, the following inequality holds:

$$\begin{aligned} f_1(x,t,\mathbf{p}) &\leq f_2(x,t,\mathbf{p}) \ in\, Q, \\ h_1(x,t,\mathbf{p}) &\leq h_2(x,t,\mathbf{p}) \ on\, \Gamma, \\ u_{1,0}(x,\mathbf{p}) &\leq u_{2,0}(x,\mathbf{p}) \ in\, \Omega. \end{aligned} \tag{4.14}$$

Then, the solutions are ordered as $u_{1,\mathbf{p}} \leq u_{2,\mathbf{p}}$.

Proof. *See Theorem 2.2.1 in [34].* ■

4.4.2 Interval Bound

We next construct an interval bound on the solution to (4.6).

Theorem 3 *Consider a pair of functions* $\mathbf{u}^L$ *and* $\mathbf{u}^U$ *satisfying the following inequality* $\forall i \in I_{\mathbf{u}}$*:*

$$\begin{aligned}
\mathcal{L}_i u_i^L &\leq \inf_{\mathbf{u}^L(x,t)\leq \mathbf{z}\leq \mathbf{u}^U(x,t),\ z_i=u_i^L(x,t),\ \mathbf{p}\in P} \left\{ f_i\left(\mathbf{z},x,t,\mathbf{p}\right)\right\} \ in\, Q, \\
\mathcal{B}_i u_i^L &\leq \inf_{\mathbf{p}\in P} \left\{ h_i\left(x,t,\mathbf{p}\right)\right\} \ on\, \Gamma, \\
u_i^L\left(x,0\right) &\leq \inf_{\mathbf{p}\in P} \left\{ u_{i,0}\left(x,\mathbf{p}\right)\right\} \ in\, \Omega, \\
\mathcal{L}_i u_i^U &\geq \sup_{\mathbf{u}^L(x,t)\leq \mathbf{z}\leq \mathbf{u}^U(x,t),\ z_i=u_i^U(x,t),\ \mathbf{p}\in P} \left\{ f_i\left(\mathbf{z},x,t,\mathbf{p}\right)\right\} \ in\, Q, \\
\mathcal{B}_i u_i^U &\geq \sup_{\mathbf{p}\in P} \left\{ h_i\left(x,t,\mathbf{p}\right)\right\} \ on\, \Gamma, \\
u_i^U\left(x,0\right) &\geq \sup_{\mathbf{p}\in P} \left\{ u_{i,0}\left(x,\mathbf{p}\right)\right\} \ in\, \Omega,
\end{aligned} \tag{4.15}$$

with it being assumed that each f_i *is continuously differentiable in the state (with the derivative being bounded in* Q*), which implies that it is Lipschitz continuous in the state, on* $\left[\mathbf{u}^L, \mathbf{u}^U\right]$*. Then, there exists a unique solution* $\mathbf{u}_{\mathbf{p}}$ *to (4.6) for each* $\mathbf{p} \in P$*, ordered as* $\mathbf{u}^L \leq \mathbf{u}_{\mathbf{p}} \leq \mathbf{u}^U$*.*

Proof. For each $\mathbf{p} \in P$, $\mathbf{u}^L$ and $\mathbf{u}^U$ satisfy the hypotheses of Theorem 1. ■

Consider the following simple example application of this theorem.

Example 4 *Consider the following scalar PDE for some* $p \in P$*:*

$$\frac{\partial u_p}{\partial t}\left(x,t\right) - \frac{\partial^2 u_p}{\partial x^2}\left(x,t\right) = e^{p^3},\ \forall \left(x,t\right) \in Q, \tag{4.16}$$

with the boundary and initial conditions not being parameter dependent. An auxiliary *system satisfying (4.15) is obtained as follows:*

$$\begin{aligned}\frac{\partial u^L}{\partial t}(x,t) - \frac{\partial^2 u^L}{\partial x^2}(x,t) &= e^{\left(p^L\right)^3}, \forall (x,t) \in Q, \\ \frac{\partial u^U}{\partial t}(x,t) - \frac{\partial^2 u^U}{\partial x^2}(x,t) &= e^{\left(p^U\right)^3}, \forall (x,t) \in Q,\end{aligned} \tag{4.17}$$

with the same initial and boundary conditions as the original PDE. Here, we have used the fact that the source function is monotonically increasing in p to deduce that:

$$e^{p^3} \in \left[e^{\left(p^L\right)^3}, e^{\left(p^U\right)^3}\right], \forall p \in P. \tag{4.18}$$

In general, standard interval arithmetic can be used to obtain an interval bound for the range of the right-hand side, thereby obtaining a valid auxiliary system of the form (4.15), with a variety of software tools (e.g., INTLAB for MATLAB [40]) being available to automate the process.

4.4.3 Relaxations

In this subsection, it is assumed that an interval bound $\left[\mathbf{u}^L, \mathbf{u}^U\right]$ has been constructed as specified in the previous subsection (this also establishes the existence of a unique solution $\mathbf{u}_{\mathbf{p}}$ for each $\mathbf{p} \in P$). We are interested in constructing convex and concave relaxations of each $u_{i,\mathbf{p}}$ on P, i.e., a pair of functions $u^{cv}_{i,\mathbf{p}}$ and $u^{cc}_{i,\mathbf{p}}$ that are respectively convex and concave on P pointwise in $\bar{Q}$, and respectively lower bounds and upper bounds pointwise in $\bar{Q}$ for each $\mathbf{p} \in P$. The following theorem is used to achieve this.

Theorem 5 *For some $\mathbf{p} \in P$, consider a pair of functions $\mathbf{u}^{cv}_{\mathbf{p}}$ and $\mathbf{u}^{cc}_{\mathbf{p}}$ satisfying the* *following equality* $\forall i \in I_{\mathbf{u}}$*:*

$$\begin{aligned} \mathcal{L}_i u_{i,\mathbf{p}}^{cv} &= f_i^{CV}\left(\mathbf{u}_{\mathbf{p}}^{cv}, \mathbf{u}_{\mathbf{p}}^{cc}, x, t, \mathbf{p}\right) \; in\, Q, \\ \mathcal{B}_i u_{i,\mathbf{p}}^{cv} &= h_i^{CV}\left(x, t, \mathbf{p}\right) \; on\, \Gamma, \\ u_{i,\mathbf{p}}^{cv}\left(x, 0\right) &= u_{i,0}^{CV}\left(x, \mathbf{p}\right) \; in \; \Omega, \\ \mathcal{L}_i u_{i,\mathbf{p}}^{cc} &= f_i^{CC}\left(\mathbf{u}_{\mathbf{p}}^{cv}, \mathbf{u}_{\mathbf{p}}^{cc}, x, t, \mathbf{p}\right) \; in\, Q, \\ \mathcal{B}_i u_{i,\mathbf{p}}^{cc} &= h_i^{CC}\left(x, t, \mathbf{p}\right) \; on\, \Gamma, \\ u_{i,\mathbf{p}}^{cc}\left(x, 0\right) &= u_{i,0}^{CC}\left(x, \mathbf{p}\right) \; in\, \Omega. \end{aligned} \tag{4.19}$$

Here, the superscripts CV and CC should be taken to mean that these functions are respectively valid convex relaxations and concave relaxations of the original right-hand side functions, provided $\mathbf{u}_{\mathbf{p}}^{cv}$ *and* $\mathbf{u}_{\mathbf{p}}^{cc}$ *are respectively valid convex and concave relaxations of* $\mathbf{u}_{\mathbf{p}}$*. Moreover, assume that each* f_i^{CV} *and* f_i^{CC} *is globally Lipschitz in* $\mathbf{u}_{\mathbf{p}}^{cv}$ *and* $\mathbf{u}_{\mathbf{p}}^{cc}$*, i.e.,* $\exists K_i^{CV} \in \mathbb{R}_+$ *and* $\exists K_i^{CC} \in \mathbb{R}_+$ *such that*

$$\begin{aligned} & \left|f_i^{CV}\left(\mathbf{y}^1, \mathbf{y}^3, x, t, \mathbf{p}\right) - f_i^{CV}\left(\mathbf{y}^2, \mathbf{y}^4, x, t, \mathbf{p}\right)\right| \\ \leq \; & K_i^{CV}\left(\sum_{j=1}^{n_{\mathbf{u}}}\left|y_j^1 - y_j^2\right| + \sum_{j=1}^{n_{\mathbf{u}}}\left|y_j^3 - y_j^4\right|\right), \forall\left(\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4\right) \in \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R}^{n_{\mathbf{u}}}, \\ & \left|f_i^{CC}\left(\mathbf{y}^1, \mathbf{y}^3, x, t, \mathbf{p}\right) - f_i^{CC}\left(\mathbf{y}^2, \mathbf{y}^4, x, t, \mathbf{p}\right)\right| \\ \leq \; & K_i^{CC}\left(\sum_{j=1}^{n_{\mathbf{u}}}\left|y_j^1 - y_j^2\right| + \sum_{j=1}^{n_{\mathbf{u}}}\left|y_j^3 - y_j^4\right|\right), \forall\left(\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \mathbf{y}^4\right) \in \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R}^{n_{\mathbf{u}}}. \end{aligned} \tag{4.20}$$

Then, $\mathbf{u}_{\mathbf{p}}^{cv}$ *and* $\mathbf{u}_{\mathbf{p}}^{cc}$ *are valid convex relaxations and concave relaxations of* $\mathbf{u}_{\mathbf{p}}$ *respectively.*

Proof. For each $\mathbf{p} \in P$, under the assumed global Lipschitz continuity of each f_i^{CV} and f_i^{CC} in $\mathbf{u}_{\mathbf{p}}^{cv}$ and $\mathbf{u}_{\mathbf{p}}^{cc}$, the sequence in $C^{2,1}\left(Q\right) \cap C\left(\bar{Q}\right)$, with successive iterates defined by

$$\begin{aligned}
\mathcal{L}_i u_{i,\mathbf{p}}^{cv,k+1} &= f_i^{CV}\left(\mathbf{u}_{\mathbf{p}}^{cv,k}, \mathbf{u}_{\mathbf{p}}^{cc,k}, x, t, \mathbf{p}\right) \text{ in } Q, \\
\mathcal{B}_i u_{i,\mathbf{p}}^{cv,k+1} &= h_i^{CV}\left(x, t, \mathbf{p}\right), \forall \left(x, t\right) \in \Gamma, \\
u_{i,\mathbf{p}}^{cv,k+1}\left(x,0\right) &= u_{i,0}^{CV}\left(x, \mathbf{p}\right), \forall x \in \Omega, \\
\mathcal{L}_i u_{i,\mathbf{p}}^{cc,k+1} &= f_i^{CC}\left(\mathbf{u}_{\mathbf{p}}^{cv,k}, \mathbf{u}_{\mathbf{p}}^{cc,k}, x, t, \mathbf{p}\right) \text{ in } Q, \\
\mathcal{B}_i u_{i,\mathbf{p}}^{cc,k+1} &= h_i^{CC}\left(x, t, \mathbf{p}\right), \forall \left(x, t\right) \in \Gamma, \\
u_{i,\mathbf{p}}^{cc,k+1}\left(x,0\right) &= u_{i,0}^{CC}\left(x, \mathbf{p}\right), \forall x \in \Omega,
\end{aligned} \tag{4.21}$$

$\forall i \in I_{\mathbf{u}}$ converges to the unique solution $\mathbf{u}_{\mathbf{p}}^{cv}$, $\mathbf{u}_{\mathbf{p}}^{cc}$ to (4.19) from any initial estimate in $C^{2,1}\left(Q\right) \cap C\left(\bar{Q}\right)$. See Theorem 8.9.1 in [34] for proof. The formal reason behind this is that the mapping between successive iterates is a contraction mapping on the Banach space $C^{2,1}\left(Q\right) \cap C\left(\bar{Q}\right)$. For every $\mathbf{p} \in P$, choose $\mathbf{u}_{\mathbf{p}}^{cv,0}$ and $\mathbf{u}_{\mathbf{p}}^{cc,0}$ to be $\mathbf{u}^L$ and $\mathbf{u}^U$ respectively. Assume that the following inequalities hold at step k for any distinct parameter pair $\mathbf{p}_1$, $\mathbf{p}_2 \in P$ and any $\lambda \in (0,1)$:

$$\begin{aligned}
\mathbf{u}_{\mathbf{p}}^{cv,k} &\leq \mathbf{u}_{\mathbf{p}} \leq \mathbf{u}_{\mathbf{p}}^{cc,k}, \forall \mathbf{p} \in P, \\
\mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cv,k} &\leq \lambda \mathbf{u}_{\mathbf{p}_1}^{cv,k} + \left(1-\lambda\right)\mathbf{u}_{\mathbf{p}_2}^{cv,k}, \\
\lambda \mathbf{u}_{\mathbf{p}_1}^{cc,k} + \left(1-\lambda\right)\mathbf{u}_{\mathbf{p}_2}^{cc,k} &\leq \mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cc,k}.
\end{aligned} \tag{4.22}$$

Note that these are valid at $k = 0$. These inequalities capture the fact that $\mathbf{u}_{\mathbf{p}}^{cv,k}$, $\mathbf{u}_{\mathbf{p}}^{cc,k}$ are valid relaxations of $\mathbf{u}_{\mathbf{p}}$, and hence $f_i^{CV}, h_i^{CV}, u_{i,0}^{CV}, f_i^{CC}, h_i^{CC}$ and $u_{i,0}^{CC}$ are valid relaxations of their respective functions at step k. Simultaneously, consider the following sequence:

$$\begin{aligned}
\mathcal{L}_i u_{i,\mathbf{p}}^{k+1} &= f_i\left(\mathbf{u}_{\mathbf{p}}^{k}, x, t, \mathbf{p}\right) \text{ in } Q, \\
\mathcal{B}_i u_{i,\mathbf{p}}^{k+1} &= h_i\left(x, t, \mathbf{p}\right) \text{ on } \Gamma, \\
u_{i,\mathbf{p}}^{k+1}\left(x, 0\right) &= u_{i,0}\left(x, \mathbf{p}\right) \text{ in } \Omega,
\end{aligned} \tag{4.23}$$

$\forall i \in I_{\mathbf{u}}$, initiated at $\mathbf{u}_{\mathbf{p}}$ such that it remains there $\forall k$. Some algebra implies from (4.21) that the following:

$$\begin{aligned}
\mathcal{L}_i\left(\lambda u_{i,\mathbf{p}_1}^{cv,k+1}+(1-\lambda)\,u_{i,\mathbf{p}_2}^{cv,k+1}\right) &= \lambda f_i^{CV}\left(\mathbf{u}_{\mathbf{p}_1}^{cv,k},\mathbf{u}_{\mathbf{p}_1}^{cc,k},x,t,\mathbf{p}_1\right) \\
&\quad+(1-\lambda)\,f_i^{CV}\left(\mathbf{u}_{\mathbf{p}_2}^{cv,k},\mathbf{u}_{\mathbf{p}_2}^{cc,k},x,t,\mathbf{p}_2\right)\ \text{in } Q, \\
\mathcal{B}_i\left(\lambda u_{i,\mathbf{p}_1}^{cv,k+1}+(1-\lambda)\,u_{i,\mathbf{p}_2}^{cv,k+1}\right) &= \lambda h_i^{CV}(x,t,\mathbf{p}_1)+(1-\lambda)\,h_i^{CV}(x,t,\mathbf{p}_2)\ \text{on } \Gamma, \\
\left(\lambda u_{i,\mathbf{p}_1}^{cv,k+1}+(1-\lambda)\,u_{i,\mathbf{p}_2}^{cv,k+1}\right)(x,0) &= \lambda u_{i,0}^{CV}(x,\mathbf{p}_1)+(1-\lambda)\,u_{i,0}^{CV}(x,\mathbf{p}_2)\ \text{in } \Omega,
\end{aligned} \tag{4.24}$$

and the following:

$$\begin{aligned}
\mathcal{L}_i u_{i,\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cv,k+1} &= f_i^{CV}\left(\mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cv,k},\mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cc,k},x,t,\lambda\mathbf{p}_1+(1-\lambda)\,\mathbf{p}_2\right)\ \text{in } Q, \\
\mathcal{B}_i u_{i,\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cv,k+1} &= h_i^{CV}(x,t,\lambda\mathbf{p}_1+(1-\lambda)\,\mathbf{p}_2)\ \text{on } \Gamma, \\
u_{i,\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cv,k+1}(x,0) &= u_{i,0}^{CV}(x,\lambda\mathbf{p}_1+(1-\lambda)\,\mathbf{p}_2)\ \text{in } \Omega,
\end{aligned} \tag{4.25}$$

equalities are valid for any distinct parameter pair $(\mathbf{p}_1,\, \mathbf{p}_2) \in P \times P$ and any $\lambda \in (0,1)$. Analogous equalities are valid for the concave overestimating portion. Then, simply comparing right-hand side values between (4.24) and (4.25) using Theorem 2, implies that $\mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cv,k+1} \le \lambda\mathbf{u}_{\mathbf{p}_1}^{cv,k+1}+(1-\lambda)\,\mathbf{u}_{\mathbf{p}_2}^{cv,k+1}$. Similarly, comparing right-hand side values between (4.23) and the top half portion of (4.21), also using Theorem 2, implies that $\mathbf{u}_{\mathbf{p}}^{cv,k+1} \le \mathbf{u}_{\mathbf{p}}$, $\forall \mathbf{p} \in P$. Analogous comparisons guarantee $\mathbf{u}_{\mathbf{p}} \le \mathbf{u}_{\mathbf{p}}^{cc,k+1}$, $\forall \mathbf{p} \in P$, and that $\lambda\mathbf{u}_{\mathbf{p}_1}^{cc,k+1}+(1-\lambda)\,\mathbf{u}_{\mathbf{p}_2}^{cc,k+1} \le \mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cc,k+1}$ for any distinct parameter pair $(\mathbf{p}_1,\, \mathbf{p}_2) \in P \times P$ and any $\lambda \in (0,1)$. Thus, we know by induction that $\mathbf{u}_{\mathbf{p}}^{cv}$ and $\mathbf{u}_{\mathbf{p}}^{cc}$ are valid relaxations, i.e., that

$$\begin{aligned}
\mathbf{u}_{\mathbf{p}}^{cv} &\le \mathbf{u}_{\mathbf{p}} \le \mathbf{u}_{\mathbf{p}}^{cc},\ \forall \mathbf{p} \in P, \\
\mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cv} &\le \lambda\mathbf{u}_{\mathbf{p}_1}^{cv}+(1-\lambda)\,\mathbf{u}_{\mathbf{p}_2}^{cv}, \\
\lambda\mathbf{u}_{\mathbf{p}_1}^{cc}+(1-\lambda)\,\mathbf{u}_{\mathbf{p}_2}^{cc} &\le \mathbf{u}_{\lambda\mathbf{p}_1+(1-\lambda)\mathbf{p}_2}^{cc},
\end{aligned} \tag{4.26}$$

for any distinct parameter pair $(\mathbf{p}_1, \mathbf{p}_2) \in P \times P$ and any $\lambda \in (0,1)$. ■

The above theorem is readily implemented using a generalization of McCormick's relaxation technique [43], which was used to construct relaxations of ODE solutions in [42] and required the same conditions as we require here. We next briefly describe the technique in the remainder of this section. The following theorem, originally proved by McCormick in 1976 [24], but presented here in the form of [27], is the cornerstone of this technique.

Theorem 6 *(McCormick's composition theorem) Let $Z \subset \mathbb{R}^n$ and $X \subset \mathbb{R}$ be nonempty convex sets. Consider the composite function $g = F \circ o$, where $o : Z \longrightarrow \mathbb{R}$ is continuous, $F : X \longrightarrow \mathbb{R}$, and let $o(Z) \subset X$. Suppose that a convex relaxation $o^{CV} : Z \longrightarrow \mathbb{R}$ and a concave relaxation $o^{CC} : Z \longrightarrow \mathbb{R}$ of o on Z are known. Let $F^{CV} : X \longrightarrow \mathbb{R}$ be a convex relaxation of F on X, let $F^{CC} : X \longrightarrow \mathbb{R}$ be a concave relaxation of F on X, let $x^{\min} \in X$ be a point at which F^{CV} attains its infimum on X, and let $x^{\max} \in X$ be a point at which F^{CC} attains its supremum on X. Then, $g^{CV} : Z \longrightarrow \mathbb{R}$,*

$$g^{CV}(\mathbf{z}) = F^{CV}\left(\mathrm{mid}\left(o^{CV}(\mathbf{z}), o^{CC}(\mathbf{z}), x^{\min}\right)\right), \forall \mathbf{z} \in Z, \tag{4.27}$$

is a convex relaxation of g on Z, and $g^{CC} : Z \longrightarrow \mathbb{R}$,

$$g^{CC}(\mathbf{z}) = F^{CC}\left(\mathrm{mid}\left(o^{CV}(\mathbf{z}), o^{CC}(\mathbf{z}), x^{\max}\right)\right), \forall \mathbf{z} \in Z, \tag{4.28}$$

is a concave relaxation of g on Z. Here, mid *is just the median of the three input values.*

The well-known fact that the sum of two convex functions is convex provides a rule for relaxing binary sums. A specific rule also exists for relaxing binary products, but it is not presented explicitly here in the interest of brevity (refer to [24], [27] and [43] for details). These three rules define McCormick's relaxation technique. Any function which can be expressed as a finite recursive composition of binary sums, binary products and a given library of intrinsic functions, referred to as a factorable function, may be relaxed using this technique. This is a rather general class of functions, including nearly all functions that can be represented finitely on a computer [27].

The first step in applying this technique is to decompose a given function into a finite recursive composition of binary sums, binary products and intrinsic functions. Then, the interval-valued independent variable in which relaxations are to be constructed is treated as a function with an interval bound for its range specified by the interval on which the independent variable can take its values and both convex and concave relaxations specified by the independent variable value. At each stage of this composition, standard interval arithmetic is used to obtain an interval bound for the range of that stage, and McCormick's relaxation rules are used to relax it (making sure to truncate relaxations to fall on the interval bound after they have been constructed). At the final stage, an interval bound for the range of the original function and its relaxations on the independent variable interval are available. Consider the following simple example to fix this idea.

Example 7 *Consider the following function:*

$$g\left(p\right) = e^{p^3} + p^3,\ \forall p \in P = \left[-1, 1\right]. \tag{4.29}$$

We are interested in obtaining a convex relaxation of g on P, g^{CV}. Then, consider the finite recursive composition (the intrinsic functions here being power and exponential):

$$\nu_1 = p,\ \nu_2 = \nu_1^3,\ \nu_3 = e^{\nu_2},\ \nu_4 = \nu_3 + \nu_2, \tag{4.30}$$

with ν_4 representing the original function g. Use the specified interval for the interval-valued independent variable p to specify:

$$\nu_1^L = -1,\ \nu_1^U = 1,\ \nu_1^{CV}\left(p\right) = p,\ \nu_1^{CC}\left(p\right) = p. \tag{4.31}$$

Since ν_2 is monotonically increasing in ν_1 deduce:

$$\nu_2^L = \left(\nu_1^L\right)^3 = -1,\ \nu_2^U = \left(\nu_1^U\right)^3 = 1. \tag{4.32}$$

The αBB relaxation rule [2] can be used to obtain the following relaxations for ν_2:

$$\nu_2^{cv}\left(\nu_1\right) = \nu_1^3 + 3\left(\nu_1^2 - 1\right), \nu_2^{cc}\left(\nu_1\right) = \nu_1^3 - 3\left(\nu_1^2 - 1\right). \tag{4.33}$$

Truncate these to lie on $\left[\nu_2^L, \nu_2^U\right]$ (recalling that max *and* min *functions preserve convexity and concavity respectively) as follows:*

$$\begin{aligned} \nu_2^{CV}\left(\nu_1\right) &= \max\left(\nu_2^L, \nu_2^{cv}\left(\nu_1\right)\right) = \max\left(-1, \nu_1^3 + 3\left(\nu_1^2 - 1\right)\right), \\ \nu_2^{CC}\left(\nu_1\right) &= \min\left(\nu_2^U, \nu_2^{cc}\left(\nu_1\right)\right) = \min\left(1, \nu_1^3 - 3\left(\nu_1^2 - 1\right)\right). \end{aligned} \tag{4.34}$$

Since the exponential function is convex, its convex relaxation on its domain is also the exponential function. Moreover, since it is a monotonically increasing function, it attains its infimum at the lower bound of its domain. Then, McCormick's composition theorem can be applied to relax ν_3 as follows:

$$\begin{aligned} \nu_3^{CV}\left(\nu_1\right) &= e^{\operatorname{mid}\left(\nu_2^{CV}\left(\nu_1\right), \nu_2^{CC}\left(\nu_1\right), \nu_2^L\right)} \\ &= e^{\operatorname{mid}\left(\max\left(-1, \nu_1^3+3\left(\nu_1^2-1\right)\right), \min\left(1, \nu_1^3-3\left(\nu_1^2-1\right)\right), -1\right)}. \end{aligned} \tag{4.35}$$

Finally, the convex relaxation of ν_4 is obtained as follows:

$$\begin{aligned} \nu_4^{CV}\left(\nu_1\right) &= \nu_3^{CV}\left(\nu_1\right) + \nu_2^{CV}\left(\nu_1\right) \\ &= e^{\operatorname{mid}\left(\max\left(-1, \nu_1^3+3\left(\nu_1^2-1\right)\right), \min\left(1, \nu_1^3-3\left(\nu_1^2-1\right)\right), -1\right)} + \max\left(-1, \nu_1^3 + 3\left(\nu_1^2 - 1\right)\right). \end{aligned} \tag{4.36}$$

In other words, the convex relaxation of g in p is given by the following:

$$g^{CV}(p) = e^{\mathrm{mid}\left(\max\left(-1,p^3+3\left(p^2-1\right)\right),\min\left(1,p^3-3\left(p^2-1\right)\right),-1\right)} \tag{4.37}$$

$$+\max\left(-1,p^3+3\left(p^2-1\right)\right). \tag{4.38}$$

The validity of this relaxation is illustrated in Fig. 4-3.

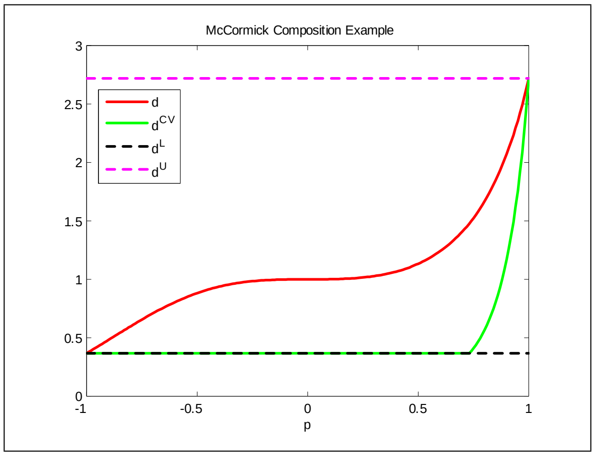

Figure 4-3: *McCormick's relaxation technique example.*

The generalization of McCormick's relaxation technique developed in [43] is analogous to the original McCormick relaxation technique, the exception being that one is also allowed to treat dependence of the function to be relaxed on intermediate variables that are known to be functions of the independent variable in which relaxations are to be constructed (with this dependence not being known explicitly). Assuming that a valid interval bound for the intermediate variable, along with valid relaxations, on the independent variable interval is available, by treating each intermediate variable as a function with the given interval bound and relaxations (making sure the relaxations have been truncated to lie on the interval bound), one is able to obtain valid relaxations of the function in the independent variable using the standard McCormick relaxation rules similarly. Consider the following simple example to fix this idea.

Example 8 *Consider the following function:*

$$g\left(p\right) = e^{p^3} + u_p,\, \forall p \in P = \left[-1, 1\right]. \tag{4.39}$$

Here, u_p is a function whose dependence on p is not known explicitly, but whose range is known to be bounded on P by $\left[u^L, u^U\right]$ and whose relaxations are known to be u_p^{cv} and u_p^{cc}. Truncate these relaxations to lie on $\left[u^L, u^U\right]$ as $u_p^{CV} = \max(u_p^{cv}, u^L)$ and $u_p^{CC} = \min(u_p^{cc}, u^U)$ (recalling that max *and* min *functions preserve convexity and concavity respectively). We are interested in obtaining a convex relaxation of g on P, g^{CV}. Then, consider the finite recursive composition:*

$$\nu_1 = p,\, \nu_2 = \nu_1^3,\, \nu_3 = e^{\nu_2},\, \nu_4 = u_p,\, \nu_5 = \nu_3 + \nu_4, \tag{4.40}$$

with ν_5 representing the original function g. Given that ν_1, ν_2 and ν_3 are the same as in Example 7, they are not treated explicitly again. Then, the convex relaxation of ν_5 is:

$$\begin{aligned} \nu_5^{CV}\left(\nu_1\right) &= \nu_3^{CV}\left(\nu_1\right) + \nu_4^{CV}\left(\nu_1\right) \\ &= e^{\mathrm{mid}\left(\max\left(-1, \nu_1^3 + 3\left(\nu_1^2 - 1\right)\right), \min\left(1, \nu_1^3 - 3\left(\nu_1^2 - 1\right)\right), -1\right)} + \nu_4^{CV}\left(\nu_1\right). \end{aligned} \tag{4.41}$$

In other words, the convex relaxation of g in p, g^{CV}, is given by the following:

$$g^{CV}\left(p\right) = e^{\mathrm{mid}\left(\max\left(-1, p^3 + 3\left(p^2 - 1\right)\right), \min\left(1, p^3 - 3\left(p^2 - 1\right)\right), -1\right)} + \max(u_p^{cv}, u^L). \tag{4.42}$$

In order to employ this generalization of McCormick's relaxation technique to apply Theorem 5, $\mathbf{u_p}$ is treated as an intermediate variable with lower bound, upper bound, convex relaxation and concave relaxation on P being $\mathbf{u}^L$, $\mathbf{u}^U$, $\mathbf{u_p}^{CV} = \mathrm{mid}\left(\mathbf{u}^L, \mathbf{u}^U, \mathbf{u_p}^{cv}\right)$ and $\mathbf{u_p}^{CC} = \mathrm{mid}\left(\mathbf{u}^L, \mathbf{u}^U, \mathbf{u_p}^{cc}\right)$ respectively. This is used to obtain the relaxations f_i^{CV} and f_i^{CC} for each f_i. In [43], it is established that these relaxations are Lipschitz in $\mathbf{u_p}^{CV}$ and $\mathbf{u_p}^{CC}$ on $\left[\mathbf{u}^L, \mathbf{u}^U\right]$ (given a minimal set of assumptions on the various basic elements of the construction, all of which are satisfied by the $C++$ library libMC [27] which automates this construction for a given library of intrinsic functions). Then, f_i^{CV} and f_i^{CC} are indeed each globally Lipschitz in $\mathbf{u}_{\mathbf{p}}^{cv}$ and $\mathbf{u}_{\mathbf{p}}^{cc}$ (this fact is also used by the analogous ODE relaxation theory in [42]), satisfying the key hypothesis of Theorem 5. Note that since in the proof to Theorem 5 $\mathbf{u}_{\mathbf{p}}^{cv,k} \leq \mathbf{u}_{\mathbf{p}} \leq \mathbf{u}_{\mathbf{p}}^{cc,k}$, $\forall \mathbf{p} \in P$, for all k, i.e., $\mathbf{u}_{\mathbf{p}}^{cv,k} \leq \mathbf{u}^U$ and $\mathbf{u}^L \leq \mathbf{u}_{\mathbf{p}}^{cc,k}$, $\forall \mathbf{p} \in P$, for all k, we may redefine $\mathbf{u}_{\mathbf{p}}^{CV} = \max\left(\mathbf{u}^L, \mathbf{u}_{\mathbf{p}}^{cv}\right)$ and $\mathbf{u}_{\mathbf{p}}^{CC} = \min\left(\mathbf{u}^U, \mathbf{u}_{\mathbf{p}}^{cc}\right)$ in the generalized McCormick relaxation construction (as opposed to $\mathbf{u}_{\mathbf{p}}^{CV} = \text{mid}\left(\mathbf{u}^L, \mathbf{u}^U, \mathbf{u}_{\mathbf{p}}^{cv}\right)$ and $\mathbf{u}_{\mathbf{p}}^{CC} = \text{mid}\left(\mathbf{u}^L, \mathbf{u}^U, \mathbf{u}_{\mathbf{p}}^{cc}\right)$). The relaxations $h_i^{CV}, u_{i,0}^{CV}, h_i^{CC}$ and $u_{i,0}^{CC}$ are constructed using the standard McCormick relaxation technique since no implicit parameter dependence is involved. Once $\mathbf{u}_{\mathbf{p}}^{CV}$ and $\mathbf{u}_{\mathbf{p}}^{CC}$ have been solved for, the generalized McCormick relaxation technique is used to obtain an interval bound $\left[O^{CV}(\mathbf{p}), O^{CC}(\mathbf{p})\right]$ on $O(\mathbf{p})$ for each $\mathbf{p} \in P$, with O^{CV} being convex and O^{CC} being concave on P (recall Equation (4.7)). Consider the following simple example to help fix this idea.

Example 9 *Consider the following PDE for each $p \in P = [-1, 1]$:*

$$\frac{\partial u_p}{\partial t}(x,t) - \frac{\partial^2 u_p}{\partial x^2}(x,t) = e^{p^3} + u_p, \; \forall (x,t) \in Q, \tag{4.43}$$

with initial and boundary conditions that do not carry parameter dependence, and assume that an interval bound $\left[u^L, u^U\right]$ has already been constructed for u_p using Theorem 1. The convex relaxation of u_p for each $p \in P$ can then be obtained by solving the following PDE:

$$\begin{aligned} & \frac{\partial u_p^{cv}}{\partial t}(x,t) - \frac{\partial^2 u_p^{cv}}{\partial x^2}(x,t) \\ = \; & e^{\text{mid}\left(\max\left(-1, p^3+3\left(p^2-1\right)\right), \min\left(1, p^3-3\left(p^2-1\right)\right), -1\right)} + \max(u_p^{cv}, u^L), \; \forall (x,t) \in Q, \end{aligned} \tag{4.44}$$

with the same initial and boundary conditions. Here, we have used the generalized convex McCormick relaxation of the source function obtained in Example 8. Note that in general, however, that the equations for u_p^{cv} and u_p^{cc} will be coupled.

4.4.4 The Case of State Spatial Derivatives Coupled to Parameter Dependence

When parameter dependence is coupled to the state spatial derivatives, i.e., when the elliptic operator in (4.1) takes the following form for each $\mathbf{p} \in P$:

$$L_i u_{i,\mathbf{p}}\left(x,t,\mathbf{p}\right) \equiv a_i\left(x,t,\mathbf{p}\right)\frac{\partial^2 u_{i,\mathbf{p}}}{\partial x^2}\left(x,t\right) + b_i\left(x,t,\mathbf{p}\right)\frac{\partial u_{i,\mathbf{p}}}{\partial x}\left(x,t\right),\ \forall\left(x,t\right)\in Q, \tag{4.45}$$

the construction of interval bounds does not change much, i.e., one solves the PDE system defined by the following inequalities:

$$\begin{gathered}
\tfrac{\partial u_i^L}{\partial t}\left(x,t\right) - \inf_{\mathbf{p}\in P}\left\{a_i\left(x,t,\mathbf{p}\right)\tfrac{\partial^2 u_i^L}{\partial x^2}\left(x,t\right) + b_i\left(x,t,\mathbf{p}\right)\tfrac{\partial u_i^L}{\partial x}\left(x,t\right)\right\}\\
\leq \inf_{\mathbf{u}^L(x,t)\leq\mathbf{z}\leq\,\mathbf{u}^U(x,t),\ z_i=u_i^L(x,t),\ \mathbf{p}\in P}\left\{f_i\left(\mathbf{z},x,t,\mathbf{p}\right)\right\},\ \forall\left(x,t\right)\in Q,\\
\mathcal{B}_i u_i^L\left(x,t\right) \leq \inf_{\mathbf{p}\in P}\left\{h_i\left(x,t,\mathbf{p}\right)\right\},\ \forall\left(x,t\right)\in\Gamma,\\
u_i^L\left(x,0\right) \leq \inf_{\mathbf{p}\in P}\left\{u_{i,0}\left(x,\mathbf{p}\right)\right\},\ \forall x\in\Omega,\\
\tfrac{\partial u_i^U}{\partial t}\left(x,t\right) - \sup_{\mathbf{p}\in P}\left\{a_i\left(x,t,\mathbf{p}\right)\tfrac{\partial^2 u_i^U}{\partial x^2}\left(x,t\right) + b_i\left(x,t,\mathbf{p}\right)\tfrac{\partial u_i^U}{\partial x}\left(x,t\right)\right\}\\
\geq \sup_{\mathbf{u}^L(x,t)\leq\mathbf{z}\leq\,\mathbf{u}^U(x,t),\ z_i=u_i^U(x,t),\ \mathbf{p}\in P}\left\{f_i\left(\mathbf{z},x,t,\mathbf{p}\right)\right\},\ \forall\left(x,t\right)\in Q,\\
\mathcal{B}_i u_i^U\left(x,t\right) \geq \sup_{\mathbf{p}\in P}\left\{h_i\left(x,t,\mathbf{p}\right)\right\},\ \forall\left(x,t\right)\in\Gamma,\\
u_i^U\left(x,0\right) \geq \sup_{\mathbf{p}\in P}\left\{u_{i,0}\left(x,\mathbf{p}\right)\right\},\ \forall x\in\Omega,
\end{gathered} \tag{4.46}$$

$\forall i \in I_{\mathbf{u}}$ in place of 4.15. Maximal and minimal over the spatial domain spatial homogeneity can be employed to handle this case for constructing valid relaxations, i.e., for any $\mathbf{p} \in P$ one solves the ODE system defined by the following equalities:

$$\begin{aligned}
\frac{du_{i,\mathbf{p}}^{cv}}{dt}(t) &= \min\begin{pmatrix} \min\limits_{x\in\Omega}\left\{f_i^{CV}\left(\mathbf{u}_{\mathbf{p}}^{cv}(t), \mathbf{u}_{\mathbf{p}}^{cc}(t), x, t, \mathbf{p}\right)\right\}, \\ \min\limits_{x\in\Gamma}\left\{\frac{\partial}{\partial t}\frac{h_i^{CV}}{\beta_i}(x, t, \mathbf{p})\right\}\end{pmatrix}, \forall t \in T, \\
\frac{du_{i,\mathbf{p}}^{cc}}{dt}(t) &= \max\begin{pmatrix} \max\limits_{x\in\Omega}\left\{f_i^{CC}\left(\mathbf{u}_{\mathbf{p}}^{cv}(t), \mathbf{u}_{\mathbf{p}}^{cc}(t), x, t, \mathbf{p}\right)\right\}, \\ \max\limits_{x\in\Gamma}\left\{\frac{\partial}{\partial t}\frac{h_i^{CC}}{\beta_i}(x, t, \mathbf{p})\right\}\end{pmatrix}, \forall t \in T, \qquad (4.47) \\
u_{i,\mathbf{p}}^{cv}(0) &= \min_{x\in\bar{\Omega}}\left\{u_{i,0}^{CV}(x, \mathbf{p})\right\}, \\
u_{i,\mathbf{p}}^{cc}(0) &= \max_{x\in\bar{\Omega}}\left\{u_{i,0}^{CC}(x, \mathbf{p})\right\},
\end{aligned}$$

$\forall i \in I_{\mathbf{u}}$ in place of (4.19). The proofs for the validity of these bounds are analogous to what was presented for the case of no parameter dependence coupled to state spatial derivatives, so there is no need to present them in detail.

4.5 Numerical Demonstration

First, a numerical note. Systems are solved using the method of lines (MOL), discretizing them on the spatial domain using the three-point-centered finite difference scheme on a uniform grid of spatial nodes and integrating the resulting coupled ODE system forward in time using the $C++$ CVODES ODE solver [13]. McCormick relaxations for the right-hand sides are constructed by the open source $C++$ library libMC [27]. Interval arithmetic is performed using a combination of libMC and INTLAB. The local optimizer used is the $fmincon$ optimizer in MATLAB. All C++ code was linked to MATLAB using its *mex* interface (we are interested in rapid prototyping here, rather than efficiency). Note that since in both examples the source function is polynomial in the state, the continuous differentiability hypothesis of Theorem 3 is trivially true.

The following parameter estimation example involves a semilinear parabolic PDE system of two equations, coupled through a quasimonotone function, so attention is restricted to the interval bounds. This is an example from chemical kinetics, a simplified version of the Belousov-Zhabotinskii reaction-diffusion system. Section 12.2 of [34] discusses the physical meaning behind each state variable as it pertains to the underlying chemical reaction network.

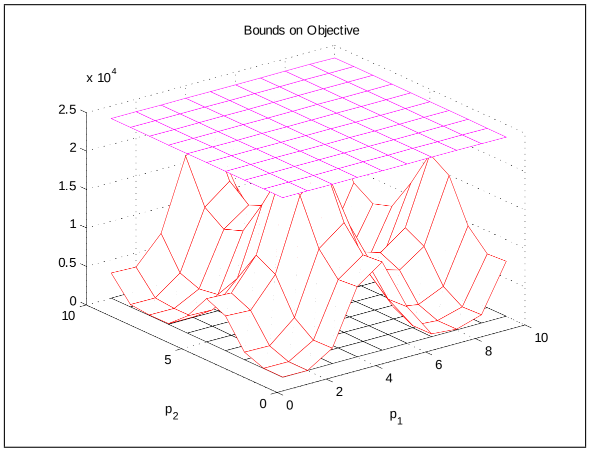

Figure 4-4: *Objective and bounds over parameter interval for the first numerical example.*

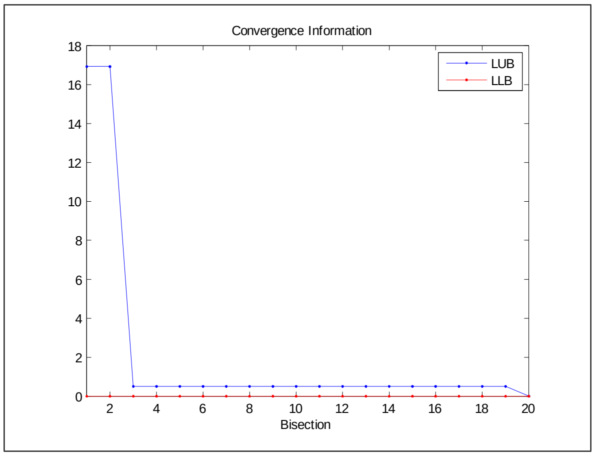

Figure 4-5: *Convergence information of a sample run of the deterministic global optimization procedure for the first numerical example.*

Example 10 *Consider the following least squares parameter estimation problem:*

$$\min_{\mathbf{p}\in P}\left\{\sum_{(x,t)\in\bar{Q}^m}\left(u_1^m(x,t)-u_{1,\mathbf{p}}(x,t)\right)^2+\sum_{(x,t)\in\bar{Q}^m}\left(u_2^m(x,t)-u_{2,\mathbf{p}}(x,t)\right)^2\right\}, \tag{4.48}$$

involving the following system:

$$\begin{gathered}\tfrac{\partial u_{1,\mathbf{p}}}{\partial t}(x,t) = \tfrac{\partial^2 u_{1,\mathbf{p}}}{\partial x^2}(x,t) + u_{1,\mathbf{p}}(x,t)\left(p_1 - p_2 u_{1,\mathbf{p}}(x,t) - 2u_{2,\mathbf{p}}(x,t)\right), \\ \forall (x,t) \in (0,1) \times (0, 0.01], \\ \tfrac{\partial u_{2,\mathbf{p}}}{\partial t}(x,t) = \tfrac{\partial^2 u_{2,\mathbf{p}}}{\partial x^2}(x,t) - 2u_{1,\mathbf{p}}(x,t)\, u_{2,\mathbf{p}}(x,t), \; \forall (x,t) \in (0,1) \times (0, 0.01],\end{gathered} \tag{4.49}$$

subject to the following time-independent boundary conditions:

$$\begin{aligned} u_{1,\mathbf{p}}(0,t) &= \sin(p_1) + 1, \; u_{2,\mathbf{p}}(0,t) = \sin(p_2) + 1, \; \forall t \in (0, 0.01], \\ u_{1,\mathbf{p}}(1,t) &= \sin(p_2) + 1, \; u_{2,\mathbf{p}}(1,t) = \sin(p_3) + 1, \; \forall t \in (0, 0.01], \end{aligned} \tag{4.50}$$

and initial conditions specified as a line between these boundary values. P is taken to be $[1,10] \times [1,10] \times [1,10]$. The sample $u_{1,\mathbf{p}}$ trajectory corresponding to all parameters set to 2 is used as the data to be fitted, $u_1^m(x,t)$ and $u_2^m(x,t)$, $\bar{Q}^m$ being specified by the temporospatial grid on which the PDE is solved, so that this is known to be the global solution a priori. Visualizing the objective over P (with p_3 fixed at 2), along with the bounds constructed using Theorem 3 as shown in Fig. 4-4 illustrates their validity. Convergence information for a sample run of the deterministic global optimization procedure is shown in Fig. 4-5.

The following example involves a semilinear parabolic PDE in two variables, coupled through a nonquasimonotone function, so relaxations are constructed along with the interval bounds.

Example 11 *Consider the following design problem:*

$$\min_{\mathbf{p} \in P} \left\{ \sum_{x \in \bar{\Omega}^m} u_{1,\mathbf{p}}(x, t_f) \right\}, \tag{4.51}$$

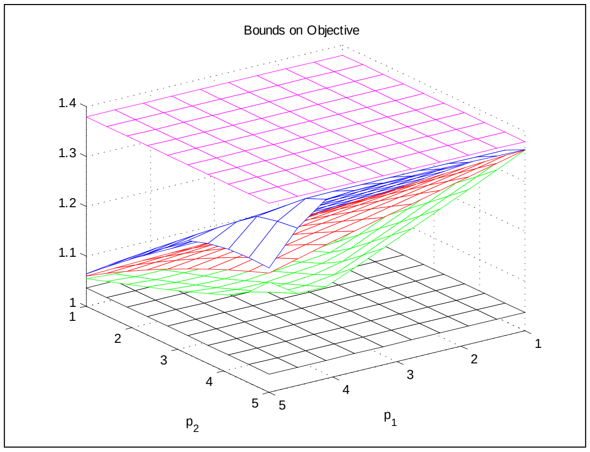

Figure 4-6: *Objective and its bounds for the second numerical example (the objective is red).*

involving the coupled parabolic system defined for each $\mathbf{p} \in P$ *by the following equations:*

$$\begin{aligned} \frac{\partial u_{1,\mathbf{p}}}{\partial t}(x,t) - \frac{\partial^2 u_{1,\mathbf{p}}}{\partial x^2}(x,t) &= p_1 u_{1,\mathbf{p}}(x,t), \, \forall (x,t) \in (0,1) \times (0,1], \\ \frac{\partial u_{2,\mathbf{p}}}{\partial t}(x,t) - \frac{\partial^2 u_{2,\mathbf{p}}}{\partial x^2}(x,t) &= -p_2 \left(u_{1,\mathbf{p}}(x,t) - u_{2,\mathbf{p}}(x,t) + \frac{(u_{2,\mathbf{p}}(x,t))^3}{3} \right), \quad (4.52) \\ \forall (x,t) &\in (0,1) \times (0,1], \end{aligned}$$

(i.e., $t_f = 1$*). Time-independent boundary conditions are specified by the following:*

$$\begin{aligned} u_{1,\mathbf{p}}(0,t) &= 1, \, u_{2,\mathbf{p}}(0,t) = 1, \, \forall (t,\mathbf{p}) \in (0,1] \times P, \quad (4.53) \\ u_{1,\mathbf{p}}(1,t) &= 1, \, u_{2,\mathbf{p}}(1,t) = 1, \, \forall (t,\mathbf{p}) \in (0,1] \times P, \end{aligned}$$

and initial conditions specified as a line between these boundary values. P *is specified as* $[1,5] \times [1,5]$. $\bar{\Omega}^m$ *is specified by the uniform spatial grid of* 100 *points on which the PDE is solved. The objective is visualized, along with its interval bounds and relaxations, on* P*, in Fig. 4-6. We see that all bounds are valid, and that the relaxations are significantly better than the interval bounds.*

Chapter 5

Conclusions and Future Directions

This chapter addresses future directions presently under exploration to expand on the algorithmic and mathematical work that has been done in this thesis. For future directions pertaining to the antireflection coating device, please refer to the conclusion section of Chapter 3. First, issues pertaining to the multilayer system branch-and-bound solver are discussed.

As discussed in Chapter 2, the parallelization strategy that we were limited to by the parallelization tools built into COSY INFINITY is better suited for a shared memory system. It is prudent then to test algorithm performance on other parallel infrastructures. These include a tightly coupled IBM Blue Gene Q supercomputer, for instance, and better suited for this algorithm, than Amazon, distributed-memory (or perhaps more appropriately, grid-computing) systems like ProfitBricks. Any one server provided by this service possesses 64 cores (for a total of 128 threads if hyperthreading were enabled), which should make it possible to run the algorithm with dynamic scheduling in an environment that is four times larger. This latter service is also better because it is subject to lower communication latency due to InfiniBand technology used for communication between servers (which has been widely been reported to be up to eight times faster than the ethernet paradigm of services such as Amazon). Developing better scheduling techniques is a necessary next step that should permit a more efficiently parallelized algorithm independent of the nature of the parallel infrastructure being used. The well-known scheduling technique of work-stealing might be ideal for this application.

Many ideas have been discussed regarding further directions that may improve the serial branch-and-bound solver, e.g., testing other less mature techniques developed to address the dependency problem, such as Hansen's generalized interval arithmetic [10], affine arithmetic [6], a more thorough comparison of selection for bisection rules (in particular, testing gradient-based rules for choosing bisection directions [37] may be a fruitful exercise). The incumbent search procedure can be improved by locally optimizing from the midpoint of any given subinterval, for instance (COSY INFINITY does have some local optimizers built into it). Another popular approach to bounding the range of a function is αBB [2], if the function is twice-differentiable (the resulting lower bound is convex, and must be locally optimized to obtain the lower bound for the merit on any given subinterval). It has been used to predict protein structure [8]. This technique requires bounds on entries of the Hessian (the matrix of the second-order partial derivatives w.r.t. $\mathbf{p}$) of the function, which has traditionally being obtained using interval arithmetic coupled to automatic differentiation techniques. Yet another approach to bounding the range of a function is McCormick's relaxation technique [24], applicable to factorable functions (the resulting lower bound is convex). It is similar to interval arithmetic in that there is a rule for every simple step in the computation of the function, and each such step is coupled to an interval arithmetic step. It should then be clear that both αBB and McCormick's relaxation technique are also plagued by the dependency problem in their traditional form. Exploring the applicability of a hybridization of αBB with Taylor arithmetic, where Taylor arithmetic is employed to obtain bounds on the Hessian rather than interval arithmetic coupled to automatic differentiation (the derivation of an explicit expression for the transfer-matrix model Hessian would be necessary for this purpose), may be a fruitful exercise (how would such bounds compare with bounds from Taylor arithmetic?). We have numerical evidence that αBB exhibits significantly better convergence than interval bounds (albeit when interval arithmetic, not Taylor arithmetic, is used in the comparison, using the toolbox INTLAB to bound the Hessian). Hybridizing McCormick's relaxation technique with Taylor arithmetic (with Taylor arithmetic replacing interval arithmetic) may similarly be a fruitful exercise. Moreover, a popular set of domain reduction techniques [41] may further improve the algorithm. Makino's range reduction technique (see Theorem 9.1 in [31]), in particular, is capable of making branch-and-bound algorithms based on Taylor arithmetic significantly more efficient. Finally, it is stressed that if a coordinate transformation can be found to eliminate or reduce the dependency problem prior to applying interval arithmetic (another approach that has been used to alleviate the dependency problem in the literature, and something that we explored with no success), the algorithm may become significantly better since such an approach would analytically exploit problem structure to reduce the dependency problem before applying the significantly cheaper (than Taylor arithmetic) interval arithmetic (an approach of this type was developed for linear static structural mechanics problems in [29], although in the context of an interval Finite Element Method). Other analytic approaches for alleviating the dependency problem prior to applying interval arithmetic, such as those outlined in [32], some drawing on the analytic properties the transfer-matrix model may have, may also be possible. Some workers have found that multisection (a partition being subdivided into more than two subintervals at every iteration), rather than bisection, can have an acceleration effect on the convergence of branch-and-bound algorithms, perhaps warranting investigation in this context. Another recently popularized approach to accelerating branch-and-bound algorithms is exploiting graphical processing units (GPUs), an approach that we hope to explore in the future (Amazon offers GPU enhanced instances, for instance). More thorough comparison of selection rules (with regards to space complexity, for instance) may be warranted. Depth-first search, for instance, is well-known to take much less memory in a variety of applications, even when it corresponds to a slower convergence time.

We also aim to explore the intersection of machine learning and branch-and-bound. In particular, explanation-based machine learning algorithms for learning structure of arbitrary problems and thereby alleviating the inherent worst-case exponential complexity of branch-and-bound algorithms [38]. The word arbitrary here is used in the sense that the special structure of the said problem cannot be exploited analytically at the time the optimization problem is being tackled, the way in this thesis it was shown one can improve the efficiency of an optimization algorithm for gradient-index systems by reducing the domain accordingly. These algorithms work by learning which control information leads to faster convergence (which portion of the search space to bisect into at any given step of the algorithm and along which direction?). One can imagine first solving a few small instances of any given problem to learn this information and then using this problem solving experience to solve larger, practical instances of the problem. Alternatively, one can imagine having a lot of users looking at a problem and sharing this information through a server that aggregates it, learns from it and shares it to all workers to help improve their solution process. This (machine-learning) we believe to be the future of the field of deterministic nonconvex programming and is something we hope to explore with future research. We believe this to be the future because it would enable the exploitation of structure of arbitrary problems without the costly scientific process of discovering structural features for narrow problem subclasses as is done today (and of which the gradient-index discussion in this thesis is a particular example).

The algorithm developed in this work can be used on a variety of other thin-film design problems. Antireflection coatings alone present many potential applications, e.g., reducing glare from medical glasses. As we have shown that for gradient index systems, efficient deterministic algorithms can be developed, a natural next step is to extend these ideas to other gradient-index optical systems, such as gradient-index lenses and gradient-index optical fibers used in fiber optic communication.

Now, we discuss extensions to the mathematical theory developed within these pages. The mathematical theory that was developed in this work needs to be used on a specific practical problem to fully demonstrate its usefulness. We are presently exploring the application of this theoretical tool to the optimization of the power conversion efficiency of homo-tandem organic solar cells. This problem corresponds to determining how many layers to compose the device of and how thick to make each layer, given some promising donor-acceptor combination composing each layer. In particular the combination of the fullerene C70 and DBP has recently been shown to provide power conversion efficiencies approaching 7% in a single bulk heterojunction layer [48]. This is an emerging solar technology with a lot of potential niche applications - the flexibility and low cost of these devices makes them suitable for novel applications such as placement in everyday clothes. However, low efficiencies limit the broad adoption of this technology. It is generally believed that an efficiency of about 15% is needed for wide scale adoption, while presently a record efficiency of 12% has been achieved by the first commercial outfit in this domain, the German company Heliatek. A properly conducted mathematical optimization study coupled to experiments, which has not yet been done, to the best of our knowledge, could be the difference. Another very important specific problem that can be addressed by these mathematical tools is the design of inorganic semiconductors. This problem is also based on the drift-diffusion-Poisson system of equations. In particular, arguably the most important question in the semiconductor industry is how to choose the spatial doping profile to minimize leakage current. There is reason to believe that one can engineer very efficient deterministic algorithms for this problem, as the Karush-Kuhn-Tucker conditions were recently shown to partially decouple [5]. The ability to fabricate and characterize these devices is available in MTL at MIT. It is prudent to reiterate that this thesis provides the first theoretical foundation, to the best of our knowledge, to allow rigorous optimization of semiconductor problems like these two. Theoretically speaking, we intend to back up the conjecture made at the beginning of the thesis that the mathematical theory can serve as a solution program for bounding other classes of differential equations by extending it accordingly.

Appendix A

Detailed Stack Evolution for a Simple Example

In this Appendix, stack evolution for the one layer normal incidence problem for a subset of the search space, i.e., where the refractive index varies in the interval [1.09 1.50] and thickness varies in the interval [5 50] nanometers, is completely visualized on two processes. The reason this is done is to make sure there is no ambiguity in the description of the algorithm, following these numbers should help the reader confirm that our algorithm is correct in finding the global optimum with a guarantee (assuming rigor of the lower bounds). The reason we choose only a subset of the search space is in order to make convergence relatively fast (only 8 iterations, of which the first four iterations are shown), so as for the amount of information to be analyzed to be limited and tractable.

Before proceeding, we first show the evolution of the incumbent at every iteration, presenting the numbers in detail so that the reader can match with the numbers on the stack:

0.1951627287405621

0.1889819981410793

0.1826910528334099

0.1797446180783397

0.1797446180783397

0.1797446180783397

0.1797446180783397

0.1797446180783397

Next, we show the evolution of the least remaining lower bound globally at every iteration:

0.1490836923045543

0.1537899704751392

0.1613456666575919

0.1616551786607268

0.1722584925402127

0.1725051390631530

0.1779414179602290

0.1791074874851764

Then, we show the evolution of the stack for the first four iterations on the first process:

```
***********************************************************
Preliminary debug stuff...
pL:
1.090000 5.000000
pU:
1.500000 100.0000
dL:
5.000000000000000
dU:
100.0000000000000
***********************************************************
Stack after ramp up...
nps:
4.000000000000000
UVAL:
0.3042129694840464
0.2383290232911963
0.2349836471142173
0.1951627287405621
LVAL:
0.1490836923045543
0.1632466322657859
0.1930907180510398
0.1510579154202665
pLstack:
```

```
1.090000 5.000000
1.295000 52.50000
1.397500 52.50000
1.397500 76.25000
pUstack:
1.295000 100.0000
1.397500 100.0000
1.500000 76.25000
1.500000 100.0000
```

```
Iteration number:
1.000000000000000
************************************************************
Stack before pruning...
nps:
2.000000000000000
UVAL:
0.3042129694840464
0.2383290232911963
LVAL:
0.1490836923045543
0.1632466322657859
pLstack:
1.090000 5.000000
1.295000 52.50000
pUstack:
1.295000 100.0000
1.397500 100.0000
```

```
active interval indices:
0.000000 0.000000
active intervals:
pL:
0.000000 0.000000
pU:
0.000000 0.000000
****************************************************************
Stack after pruning...
nps:
2.000000000000000
UVAL:
0.3042129694840464
0.2383290232911963
LVAL:
0.1490836923045543
0.1632466322657859
pLstack:
1.090000 5.000000
1.295000 52.50000
pUstack:
1.295000 100.0000
1.397500 100.0000
```

```
Iteration number:
2.000000000000000
```

```
indexing into global stack to extract active intervals on 1st process
```

```
partitions extracted on this process->
active interval indices:
1.000000 1.000000
active intervals:
pL:
1.090000 5.000000
pU:
1.295000 100.0000
```

```
Iteration number:
2.000000000000000
***************************************************************
Stack before pruning...
nps:
4.000000000000000
UVAL:
0.3042129694840464
0.2383290232911963
0.3237282798328219
0.2808789513449836
LVAL:
0.1490836923045543
0.1632466322657859
0.2704156837017569
0.1920898887374683
pLstack:
1.090000 5.000000
1.295000 52.50000
```

```
1.090000 5.000000
1.090000 52.50000
pUstack:
1.295000 100.0000
1.397500 100.0000
1.295000 52.50000
1.295000 100.0000
active interval indices:
1.000000 1.000000
active intervals:
pL:
1.090000 5.000000
pU:
1.295000 100.0000
****************************************************************
Stack after pruning...
nps:
1.000000000000000
UVAL:
0.2383290232911963
LVAL:
0.1632466322657859
pLstack:
1.295000 52.50000
pUstack:
1.397500 100.0000
```

```
Iteration number:
```

```
3.000000000000000
```

```
indexing into global stack to extract active intervals on 1st process
partitions extracted on this process->
active interval indices:
1.000000 1.000000
active intervals:
pL:
1.295000 52.50000
pU:
1.397500 100.0000
```

```
Iteration number:
3.000000000000000
*****************************************************************
Stack before pruning...
nps:
3.000000000000000
UVAL:
0.2383290232911963
0.2577287614722094
0.2227154654695186
LVAL:
0.1632466322657859
0.2210993143197687
0.1834596469663544
pLstack:
1.295000 52.50000
```

```
1.295000 52.50000
1.295000 76.25000
pUstack:
1.397500 100.0000
1.397500 76.25000
1.397500 100.0000
active interval indices:
1.000000 1.000000
active intervals:
pL:
1.295000 52.50000
pU:
1.397500 100.0000
***********************************************************
Stack after pruning...
nps:
0.000000000000000
UVAL:
LVAL:
pLstack:
pUstack:
```

```
Iteration number:
4.000000000000000
************************************************************
Stack before pruning...
nps:
2.000000000000000
```

```
UVAL:
0.2577287614722094
0.2227154654695186
LVAL:
0.2210993143197687
0.1834596469663544
pLstack:
1.295000 52.50000
1.295000 76.25000
pUstack:
1.397500 76.25000
1.397500 100.0000
active interval indices:
2.000000 1.000000
active intervals:
pL:
1.295000 52.50000
pU:
1.397500 100.0000
**************************************************************
Stack after pruning...
nps:
0.000000000000000
UVAL:
LVAL:
pLstack:
pUstack:
```

Finally we show the evolution of the stack for the first four iterations on the second process:

```
***************************************************************
Preliminary debug stuff...
pL:
1.090000 5.000000
pU:
1.500000 100.0000
dL:
5.000000000000000
dU:
100.0000000000000
***************************************************************
Stack after ramp up...
nps:
4.000000000000000
UVAL:
0.3042129694840464
0.2383290232911963
0.2349836471142173
0.1951627287405621
LVAL:
0.1490836923045543
0.1632466322657859
0.1930907180510398
0.1510579154202665
pLstack:
1.090000 5.000000
```

```
1.295000 52.50000
1.397500 52.50000
1.397500 76.25000
pUstack:
1.295000 100.0000
1.397500 100.0000
1.500000 76.25000
1.500000 100.0000
```

```
Iteration number:
1.000000000000000
********************************************************************
Stack before pruning...
nps:
2.000000000000000
UVAL:
0.2349836471142173
0.1951627287405621
LVAL:
0.1930907180510398
0.1510579154202665
pLstack:
1.397500 52.50000
1.397500 76.25000
pUstack:
1.500000 76.25000
1.500000 100.0000
active interval indices:
```

```
0.000000 0.000000
active intervals:
pL:
0.000000 0.000000
pU:
0.000000 0.000000
******************************************************
Stack after pruning...
nps:
2.000000000000000
UVAL:
0.2349836471142173
0.1951627287405621
LVAL:
0.1930907180510398
0.1510579154202665
pLstack:
1.397500 52.50000
1.397500 76.25000
pUstack:
1.500000 76.25000
1.500000 100.0000
```

```
Iteration number:
2.000000000000000
```

```
indexing into global stack to extract active intervals on 2nd process
partitions extracted on this process->
```

```
active interval indices:
2.000000 2.000000
active intervals:
pL:
1.397500 76.25000
pU:
1.500000 100.0000
```

```
Iteration number:
2.000000000000000
***************************************************************
Stack before pruning...
nps:
4.000000000000000
UVAL:
0.2349836471142173
0.1951627287405621
0.2016385899662755
0.1889819981410793
LVAL:
0.1930907180510398
0.1510579154202665
0.1682811872638599
0.1537899704751392
pLstack:
1.397500 52.50000
1.397500 76.25000
1.397500 76.25000
```

```
1.448750 76.25000
pUstack:
1.500000 76.25000
1.500000 100.0000
1.448750 100.0000
1.500000 100.0000
active interval indices:
2.000000 2.000000
active intervals:
pL:
1.397500 76.25000
pU:
1.500000 100.0000
**************************************************************
Stack after pruning...
nps:
2.000000000000000
UVAL:
0.2016385899662755
0.1889819981410793
LVAL:
0.1682811872638599
0.1537899704751392
pLstack:
1.397500 76.25000
1.448750 76.25000
pUstack:
1.448750 100.0000
```

```
1.500000 100.0000
```

```
Iteration number:
3.000000000000000
```

```
indexing into global stack to extract active intervals on 2nd process
partitions extracted on this process->
active interval indices:
2.000000 2.000000
active intervals:
pL:
1.448750 76.25000
pU:
1.500000 100.0000
```

```
Iteration number:
3.000000000000000
****************************************************************************
Stack before pruning...
nps:
4.000000000000000
UVAL:
0.2016385899662755
0.1889819981410793
0.1967231422981335
0.1826910528334099
LVAL:
0.1682811872638599
```

```
0.1537899704751392
0.1766074510748749
0.1613456666575919
pLstack:
1.397500 76.25000
1.448750 76.25000
1.448750 76.25000
1.448750 88.12500
pUstack:
1.448750 100.0000
1.500000 100.0000
1.500000 88.12500
1.500000 100.0000
active interval indices:
2.000000 2.000000
active intervals:
pL:
1.448750 76.25000
pU:
1.500000 100.0000
************************************************************
Stack after pruning...
nps:
3.000000000000000
UVAL:
0.2016385899662755
0.1967231422981335
0.1826910528334099
```

```
LVAL:
0.1682811872638599
0.1766074510748749
0.1613456666575919
pLstack:
1.397500 76.25000
1.448750 76.25000
1.448750 88.12500
pUstack:
1.448750 100.0000
1.500000 88.12500
1.500000 100.0000
```

```
Iteration number:
4.000000000000000
```

```
indexing into global stack to extract active intervals on 2nd process
partitions extracted on this process->
active interval indices:
2.000000 1.000000
active intervals:
pL:
1.397500 76.25000
pU:
1.448750 100.0000
```

```
Iteration number:
4.000000000000000
```

```
indexing into global stack to extract active intervals on 2nd process
partitions extracted on this process->
active interval indices:
2.000000 3.000000
active intervals:
pL:
1.448750 88.12500
pU:
1.500000 100.0000
```

```
Iteration number:
4.000000000000000
***************************************************************
Stack before pruning...
nps:
5.000000000000000
UVAL:
0.2016385899662755
0.1967231422981335
0.1826910528334099
0.1857196447674095
0.1797446180783397
LVAL:
0.1682811872638599
0.1766074510748749
0.1613456666575919
0.1684226171666425
```

```
0.1616551786607268
pLstack:
1.397500 76.25000
1.448750 76.25000
1.448750 88.12500
1.448750 88.12500
1.474375 88.12500
pUstack:
1.448750 100.0000
1.500000 88.12500
1.500000 100.0000
1.474375 100.0000
1.500000 100.0000
active interval indices:
2.000000 3.000000
active intervals:
pL:
1.448750 88.12500
pU:
1.500000 100.0000
************************************************************
Stack after pruning...
nps:
3.000000000000000
UVAL:
0.1967231422981335
0.1857196447674095
0.1797446180783397
```

```
LVAL:
0.1766074510748749
0.1684226171666425
0.1616551786607268
pLstack:
1.448750 76.25000
1.448750 88.12500
1.474375 88.12500
pUstack:
1.500000 88.12500
1.474375 100.0000
1.500000 100.0000
```

This completes our “complete” stack evolution visualization exercise!

Bibliography

[1] A broadband simplified version of the solis clear sky model. *Solar Energy 82*, 8 (2008), 758 – 762.

[2] Adjiman, C., and Floudas, C. A. Rigorous convex underestimators for general twice–differentiable problems. *Journal of Global Optimization 9* (1996), 23–40.

[3] Berz, M., and Makino, K. Rigorous global search using taylor models. In *Proceedings of the 2009 conference on Symbolic numeric computation* (New York, NY, USA, 2009), SNC '09, ACM, pp. 11–20.

[4] Berz, M., and Makino, K. *(COSY INFINITY) 9.1 Programmer's Manual*, 2011.

[5] Burger, M., Pinnau, R., and Wolfram, M. On/off-state design of semiconductor doping models. *Commun. Math. Sci 6*, 4 (Dec. 2008), 799–1095.

[6] Comba, J. L. D., and Stolfi, J. Affine arithmetic and its applications to computer graphics. In *Proceedings of VI SIBGRAPI 1993.* (1993), pp. 9–18. Recife, Brazil, October 1993.

[7] Dobrowolski, J., and Kemp, R. Refinement of optical multilayer systems with different optimization procedures. *Applied Optics 29*, 19 (1990), 2876–2893.

[8] Eyrich, V. A., Standley, D. M., Felts, A. K., and Friesner, R. A. Protein tertiary structure prediction using a branch and bound algorithm. *Proteins 35*, 1 (1999), 41–57.

[9] Ghebrebrhan, M., Bermel, P., Avniel, Y., Joannopoulos, J. D., and Johnson, S. G. Global optimization of silicon photovoltaic cell front coatings. *Optics Express 17*, 9 (2009), 7505–7518.

[10] Hansen, E. A generalized interval arithmetic. In *Interval Mathematics*, K. Nickel, Ed., vol. 29 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1975, pp. 7–18.

[11] Hansen, E., and Walster, G. W. *Global optimization using interval analysis*, vol. 264 of *Monographs and Textbooks in Pure and Applied Mathematics.* Marcel Dekker Inc., 2004.

[12] Harrison, G. Dynamic models with uncertain parameters. *In Proceedings of the First International Conference on Mathematical Modeling 1* (1977), 295–304.

[13] Hindmarch, A. C., and Serban, R. *User Documentation for CVODES v2.6.0.*

[14] Hofschuster, W., and Kramer, W. *C-XSC 2.0: A C++ Library for Extended Scientific Computing*, 2001.

[15] Horst, R., and Tuy, H. *Global optimization: deterministic approaches.* Springer, 1993.

[16] Kawamata, K., Shouzu, T., and Mitamura, N. K-m-s (keep-molecules sputtering) deposition of optical mgf2 thin films. *Vacuum 51*, 4 (1998), 559 – 564.

[17] Koster, L. J. A., Smits, E. C. P., Mihailetchi, V. D., and Blom, P. W. M. Device model for the operation of polymer/fullerene bulk heterojunction solar cells. *Phys. Rev. B 72* (Aug 2005), 085205.

[18] Kumar, P., Jain, S. C., Kumar, V., Chand, S., and Tandon, R. P. A model for the currentvoltage characteristics of organic bulk heterojunction solar cells. *Journal of Physics D: Applied Physics 42*, 5 (2009), 055102.

[19] Kuo, M., Poxson, D., Kim, Y., Mont, F., Kim, J., Schubert, E., and Lin, S. Realization of a near-perfect antireflection coating for silicon solar energy utilization. *Optics Letters 33*, 21 (2008), 2527–2529.

[20] Liddell, H. M. *Computer-aided Techniques for the Design of Multilayer Filters.* 1981.

[21] Lin, C., and Snyder, L. *Principles of Parallel Programming*, 1st ed. Addison-Wesley Publishing Company, USA, 2008.

[22] Makino, K., and Berz, M. Optimal correction and design parameter search by modern methods of rigorous global optimization. *Nuclear Instruments & Methods in Physics Research Section A-accelerators Spectrometers Detectors and Associated Equipment 645* (2011), 332–337.

[23] Martin, S., Rivory, J., and Schoenauer, M. Synthesis of optical multilayer systems using genetic algorithms. *Applied Optics 34*, 13 (1995), 2247–2254.

[24] McCormick, G. P. Computability of global solutions to factorable nonconvex programs: Part i - convex underestimating problems. *Mathematical Programming 10* (1976), 147–175.

[25] McGeoch, C. C., and Wang, C. Experimental evaluation of an adiabiatic quantum system for combinatorial optimization. In *Proceedings of the ACM International Conference on Computing Frontiers* (New York, NY, USA, 2013), CF '13, ACM, pp. 23:1–23:11.

[26] Miao, L., Jin, P., Kaneko, K., Terai, A., Nabatova-Gabain, N., and Tanemura, S. Preparation and characterization of polycrystalline anatase and rutile tio2 thin films by rf magnetron sputtering. *Applied Surface Science 212*, 213 (2003), 255 – 263.

[27] Mitsos, A., Chachuat, B., and Barton, P. I. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization 20*, 2 (2009), 573–601.

[28] Moore, R. *Interval Analysis.* 1966.

[29] Muhanna, R. L. Combined axial and bending stiffness in interval finite-element methods. *Journal of Structural Engineering 133* (2007), 9–43.

[30] Neher, M., Jackson, K., and Nedialkov, N. On taylor model based integration of odes. *SIAM Journal on Numerical Analysis 45*, 1 (2007), 236–262.

[31] Neumaier, A. Taylor forms - use and limits. *Reliable Computing 2003* (2002), 9–43.

[32] Neumaier, A. Improving interval enclosures, 2009.

[33] Oskooi, A., Mutapcic, A., Noda, S., Joannopoulos, J. D., Boyd, S. P., and Johnson, S. G. Robust optimization of adiabatic tapers for coupling to slow-light photonic-crystal waveguides. *Opt. Express 20*, 19 (Sep 2012), 21558–21575.

[34] Pao, C. *Nonlinear Parabolic and Elliptic Equations.* 1992.

[35] Poitras, D., and Dobrowolski, J. A. Toward perfect antireflection coatings. 2. theory. *Appl. Opt. 43*, 6 (Feb 2004), 1286–1295.

[36] Poxson, D. J., Schubert, M. F., Mont, F. W., Schubert, E. F., and Kim, J. K. Broadband omnidirectional antireflection coatings optimized by genetic algorithm. *Opt. Lett. 34*, 6 (Mar 2009), 728–730.

[37] Ratz, D., and Csendes, T. On the selection of subdivision directions in interval branch-and-bound methods for global optimization. *J. Global Optimization 7* (1995), 183–207.

[38] Realff, M. J., and Stephanopoulos, G. On the application of explanation-based learning to acquire control knowledge for branch and bound algorithms. *INFORMS J. on Computing 10*, 1 (Jan. 1998), 56–71.

[39] Rinnooy Kan, A. H. G., and Timmer, G. T. Stochastic global optimization methods. part 1: clustering methods. *Math. Program. 39*, 1 (1987), 27–56.

[40] Rump, S. INTLAB - INTerval LABoratory. In *Developments in Reliable Computing*, T. Csendes, Ed. Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104.

[41] Sahinidis, N. V. *(BARON) User's Manual Version 4.0*, 1999.

[42] Scott, J. K., Chachuat, B., and Barton, P. I. Nonlinear convex and concave relaxations for the solutions of parametric ordinary differential equations. *Journal of Global Optimization* (2010).

[43] Scott, J. K., Stuber, M. D., and Barton, P. I. Generalized McCormick relaxations. *Journal of Global Optimization* (2010).

[44] Singer, A. B., and Barton, P. I. Bounding the solutions of parameter dependent nonlinear ordinary differential equations. *SIAM J. Sci. Comput. 27*, 6 (2006), 2167–2182.

[45] Spinelli, P., Verschuuren, M., and Polman, A. Broadband omnidirectional antireflection coating based on subwavelength surface mie resonators. *Nat Commun 3* (2012).

[46] Thelen, A. Design of a hot mirror: contest results. *Appl. Opt. 35*, 25 (1996), 4966–4977.

[47] Walter, W. *Differential and Integral Inequalities.* 1970.

[48] Zheng, Y.-q., Potscavage, W. J., Komino, T., and Adachi, C. Highly efficient bulk heterojunction photovoltaic cell based on tris[4-(5-phenylthiophen-2-

yl)phenyl]amine and c70 combined with optimized electron transport layer. *Applied Physics Letters 102*, 15 (2013), 102–105.