A NOVEL GEOMETRY PARAMETERIZATION, OPTIMIZATION AND SIMULATION OF REALISTIC AL_2O_3-BASED OPTICAL WAVEGUIDE Y-BRANCHES

Master of Science Thesis

Jıhad AWAD

Eskişehir 2019

A NOVEL GEOMETRY PARAMETERIZATION, OPTIMIZATION AND SIMULATION OF REALISTIC AL_2O_3-BASED OPTICAL WAVEGUIDE Y-BRANCHES

Jıhad AWAD

MASTER OF SCIENCE THESIS

Electrical and Electronics Department

Supervisor: Assoc. Prof. Dr. Feridun AY

Eskişehir

Eskişehir Technical University

Institute of Graduate Programs

July 2019

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), project no 114E594.

FINAL APPROVAL FOR THESIS

This thesis titled "A Novel Geometry Parameterization, Optimization and Simulation of Realistic Al_2O_3-Based Optical Waveguide Y-Branches" has been prepared and submitted by Jıhad AWAD in partial fulfillment of the requirements in "Eskişehir Technical University Directive on Graduate Education and Examination" for the Degree of Master of Science in Electrical and Electronics Engineering Department has been examined and approved on 02/07/2019

Committee Members	**Title Name Surname**	**Signature**
Member (Supervisor)	: Assoc. Prof. Dr. Feridun AY	.....
Member	: Prof. Dr. Atilla AYDINLI	.....
Member	: Assoc. Prof. Dr. Nihan KOSKU PERKGÖZ	.....

Prof. Dr. Murat TANIŞLI

Director of Institute of Graduate Programs

ABSTRACT

A NOVEL GEOMETRY PARAMETERIZATION, OPTIMIZATION AND SIMULATION OF REALISTIC AL_2O_3-BASED OPTICAL WAVEGUIDE Y-BRANCHES

Jıhad AWAD

Department of Electrical and Electronics Engineering

Eskişehir Technical University, Institute of Graduate Programs, July 2019

Supervisor: Assoc. Prof. Dr. Feridun AY

Integrated optics are expected to exhibit a growth rate similar to that exhibited by integrated electronics several decades ago. However, in order to maintain its growth and to produce highly functional, densely integrated optical devices, advanced design methods should be developed.

In this thesis, we showcased how to merge an optimization technique with a powerful electromagnetic solver to form a design tool which is capable of finding optimal designs while fulfilling some constraints. Namely, we used Particle Swarm Optimization algorithm combined with Finite-Difference Time-Domain electromagnetic solver to find the optimal geometry of a single-mode optical waveguide Y-branch while ensuring design fabricability within a 1 μm-resolution standard lithography. The proposed design has a mean total insertion loss of 0.45 dB over wavelengths ranging from 1450 to 1580 nm with a footprint of around 40 by 10 μm. Novel aspects of our work include the introduction of a new design parameterization and the usage of a weakly guiding material system, namely Al_2O_3 over SiO_2 which is more challenging than strongly guiding material systems. Finally, all simulations presented in this thesis are based on a realistic waveguide geometry which proved to be extremely important for simulations to reflect reality.

In this thesis, amorphous Al_2O_3 grown using Atomic Layer Deposition technique was chosen because it is very promising for the realization of active and passive integrated optical components.

Keywords: Y-branch, Photonic design optimization, Particle Swarm Optimization, Al_2O_3, Waveguide, Integrated optics, FDTD.

ÖZET

GERÇEKÇİ Al_2O_3 TABANLI Y-DALLI OPTİK DALGA KILAVUZLARINDA YENİ BİR GEOMETRİ PARAMETERİZASYONU, OPTİMİZASYONU VE SİMÜLASYONU

Jıhad AWAD

Elektrik Elektronik Mühendisliği Bölümü

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Temmuz 2019

Danışman: Doç. Prof. Dr. Feridun AY

Entegre optiklerin, birkaç on yıl önce gömülü elektroniklerin gösterdiğine benzer bir büyüme oranı göstermesi beklenmektedir. Bununla birlikte, büyümesini sürdürmek ve oldukça işlevsel, yoğun biçimde entegre edilmiş optik cihazlar üretmek için gelişmiş tasarım yöntemleri geliştirilmelidir. Bu tezde, bir optimizasyon yönteminin güçlü bir elektromanyetik çözücü ile nasıl birleştirileceğini, bu teknikle bazı sınırlamaları yerine getirirken optimal tasarımları bulabilecek bir tasarım aracı oluşturulabileceğini gösterdik. Bunu elde etmek için, Sonlu Fark Zaman Çizelgesi elektromanyetik çözücüsü ile 1 μm çözünürlükteki standart litografi tasarımlarının üretimini sağlarken, tek modlu, optik Y-Dallı dalga kılavuzunun ideal geometrisini tanımlamak için bir parçacık tarama optimizasyon algoritması kullandık. Önerilen tasarım 1450 ila 1580 nm dalga boyları aralığında 0.45 dB'lik ortalama toplam ekleme kaybına sahiptir ve yaklaşık olarak 40 ila 10 μm bir ayak izi vardır. Araştırmamızın genel bakış açısı yeni tasarım parametrelerinin uygulanmasını ve ağır yol gösterici malzeme işlemlerinden daha zor olan zayıf yönlendirilmiş bir malzeme sisteminin, yani SiO_2 üzerinde Al_2O_3'ün kullanılmasını içermektedir. Son olarak, bu tezde sunulan tüm simülasyonlar, simülasyonların gerçeği yansıtması için son derece önemli olduğu kanıtlanan gerçekçi bir dalga kılavuzu geometrisine dayanmaktadır.

Bu tezde, aktif ve pasif entegre optik bileşenlerin gerçekleştirilmesi için çok umut verici olduğu olan Atomik Katman Kaplama biriktirme tekniği kullanılarak büyütülen amorf Al_2O_3 seçilmiştir.

Anahtar Kelimeler: Y-Dallı, Fotonik Tasarım Optimizasyonu, Parçacık Sürüsü Algoritması, Al_2O_3, Dalga Kılavuzu, Tümleşik Optik, FDTD.

02/07/2019

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Jıhad AWAD

.....

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

Page

LIST OF ABBREVIATIONS

CMOS	: Complementary Metal-Oxide-Semiconductor
LIDAR	: Light Detection and Ranging
PSO	: Particle Swarm Optimization
FDTD	: Finite-Difference Time-Domain
CAD	: Computer Aided Design
BPM	: Beam Propagation Method
MoM	: Method of Moments
FEM	: Finite-Element Method
EME	: Eigenmode Expansion method
TMM	: Transfer Matrix Method
1D	: 1 Dimensional
2D	: 2 Dimensional
3D	: 3 Dimensional
VCSEL	: Vertical-Cavity Surface-Emitting Laser
LED	: Light Emitting Diode
OLED	: Organic Light Emitting Diode
GSA	: Gravitational Search Algorithm
COA	: Chaotic Optimization Algorithm
SA	: Stimulated Annealing
TSA	: The Tabu Search Algorithm
ICA	: Imperialistic Competition Algorithm
GA	: Genetic Algorithm
EP	: Evolutionary Programming
ES	: Evolutionary Strategy
DE	: Differential Evolution
ACO	: Ant Colony Optimization
BFO	: Bacterial Foraging Optimization
ABC	: Artificial Bee Colony
ALD	: Atomic Layer Deposition
MIDAS	: Micro/Nano Devices And Systems
FOM	: Figure of Merit

1. INTRODUCTION

Gordon Moor, a Co-founder of Intel, predicted in 1965 that the number of transistors per unit area will double about every two years. His prediction held true for decades even though it is not based on a scientific law. Fast forward to today's state-of-the-art electronics industry, it seems that the famous law is about to end due to physical limitations. As transistor's dimensions become smaller and smaller, it reaches a point where quantum mechanics and related phenomena like electron tunneling start taking over, which is detrimental to the transistor's performance. Therefore, researchers are looking for alternatives that offer similar performance to silicon-based integrated electronics. One possible candidate is integrated optics which is the field of science that is concerned with developing optical devices and circuits with high functionality.

Numerous material systems and technology platforms are developed or being developed to be used in integrated optics. These material systems include III-V semiconductors, high index glasses, nitrides, Lithium niobates, polymers, Silicon ...etc. [1]. Among all of the previously mentioned material systems, Silicon is especially promising due to three factors; first, Silicon is arguably the most studied element of all time. Thus, its physical, chemical and optical properties are well known. Second, Silicon is very abundant and cheap with purification process mastered throughout decades. Third, the manufacturing techniques and tools developed for integrated electronics can be repurposed for manufacturing integrated optics. If Silicon is used in the realization of integrated optics then it is called "Silicon photonics" [1]. Silicon photonics offers well-established manufacturing techniques which can potentially reduce the costs of production and increase the production volume. Another advantage of using Silicon in integrated optics is the high refractive index contrast between it and its oxide resulting in a strongly guiding material system which in turn allows for sub-micrometer waveguides and features and lower radii bends. Therefore offering a more efficient use of the chip area.

Photonics and electronics are anticipated to merge into one field where it will be possible to create electro-photonic circuits on the same chip. This is turn would have a big impact globally. Many applications would benefit from such a technology most notably is the data communications-related devices and systems. However, such an integration comes with its own set of problems and challenges; e.g. every attempt to

integrate a photonic functionality directly into a Complementary Metal-Oxide-Semiconductor (CMOS) chip without changing the process has either failed or resulted in poor performing devices [2].

1.1. Applications of Silicon Photonics

Although silicon photonics are mostly in research and development phase with limited volume production, a number of applications are slowly emerging. Data communications is a major sector which is expected to benefit the most from silicon photonics. Applications such as high-speed short-reach, advanced modulation schemes and coherent long range communications are anticipated to incorporate silicon photonic technologies in the near future [2]. Other fields which can benefit from silicon photonics include biosensing [3, 4], gas sensors [5], novel light sources [6, 7], optical gyroscopes [8, 9], nonlinear optics [10], Light Detection and Ranging (LIDAR) systems [11] and radio frequency integrated optoelectronics [12, 13], on chip optical connections [14] to name just a few.

1.2. Outline of This Thesis

This thesis revolves around optimization algorithm-aided optical component design process. This is a relatively new concept and have been implemented with various optimization algorithms and electromagnetic solvers. Our approach involves using the Particle Swarm Optimization (PSO) algorithm with the Finite-Difference Time-Domain (FDTD) electromagnetic solver to find the optimal geometry of an optical component. Although we demonstrated this approach by optimizing a single-mode optical waveguide Y-branch, most components can be designed with the same approach (after introducing the relevant parameterization). While this thesis reports the design parameterization and optimization of a single-mode optical waveguide Y-branch, it provides a general insight on the optical component design and optimization processes.

In section 2, some of the numerical methods used for solving electromagnetic problems are presented briefly. FDTD solver theory was explained in detail since it was used in this work.

Section 3 provides a brief overview on the optimization algorithms and their classifications. Then the focus is drown towards PSO and its implementation is Lumerical.

In section 4, optical waveguide design choices were presented in general, the design process for a single-mode, ridge waveguide made from Al_2O_3 over SiO_2 substrate was explained. Then we focused on Al_2O_3 and its role as a host material for rare-earth elements such as Erbium.

In section 5, all pieces of this thesis come together where a Y-branch parameterization and optimization process in Lumerical environment is explained in detail. A conclusion of this work was drawn at the end on this section.

2. NUMERICAL METHODS FOR SOLVING MAXWELL'S EQUATIONS

Running simulations is much faster and cheaper than fabricating actual prototypes and then making experimental measurements. It is especially important in the case of trying to evaluate a number of different designs before the actual fabrication process. This holds true for many fields in science and engineering and integrated optics is not an exception. Currently, many Computer Aided Design (CAD) software packages are used for simulating photonic structures and circuits with each one of them relying on a numerical method for solving Maxwell's equations. Most of these software packages have some sort of a graphical user interface, a scripting language and some other functionalities like parameter sweeps and optimizations.

The design of photonics often requires an understanding of how light will propagate in the structure understudy [2]. Therefore is extremely important for the designer to find the electrical and magnetic fields in the structure. These fields can be found by solving Maxwell's equations. However, the exact analytical solution of Maxwell's equations is limited to simple and symmetrical structures. For more complex structures it is more convenient to solve them numerically. Several numerical schemes for solving these equations such as Beam Propagation Method (BPM), Method of Moments (MoM) [15], the Finite-Element Method (FEM) [16], Eigenmode Expansion method (EME), Transfer Matrix Method (TMM) and Finite-Difference Time-Domain method (FDTD) [17] were proposed. Each one of the previously mentioned methods has its own strengths and weaknesses and therefore is tailored toward solving a category of electromagnetic problems.

The BPM predates FDTD and it provides a scalar approximate solution for Maxwell's equations. Originally, this method was suitable for slowly varying structures with small refractive index contrasts. Its usage was restricted for paraxial (small angle) forward-only propagation [2]. It was later improved to find vectorial solutions with forward and backward and wider angle propagations [2]. It is successfully implemented in numerous software packages like Synopsys RSoft BeamPROP [18] which was used by the author at the beginning of this research but was later replaced by the superior FDTD.

Eigenmode Expansion method (EME) is a rigorous, frequency domain, full vectorial method [19]. In EME, the geometry is divided into cells. Then the propagating

field is degenerated into modes (here they are called “super modes”) which are calculated at the interface of adjacent cells. Scattering matrices (s-parameters) for each section are then formulated by matching the tangential E and H fields at the cell boundaries [19]. Super modes coupling from one cell to another is calculated with the aid of the S-parameters between them. Therefore, this method is inherently bi-directional [2]. The accuracy of this method is related to the number of super modes used; higher number of super modes increases accuracy and computational cost. However, in this method, the computational cost scales exceptionally well with propagation length, which makes it more efficient at simulating long structures than FDTD. Therefore, this method is suitable for simulating long tapers, gratings and periodic structures [2]. Unlike BPM, EME can simulate propagation at large angles and is not limited to low refractive index contrast material systems [19]. This method simulates propagation at only one wavelength which means that many simulations are required to find the wavelength response of a given design.

Transfer Matrix Method (TMM) is a fast and simple technique suitable for devices that have varying refractive index profile [2]. It requires the structure to be approximated to be 1 Dimensional (1D) therefore its usage is limited. It can be employed to simulate thin-film reflectors and Distributed Bragg Reflectors in Vertical Cavity Lasers (VCSELs) [2].

The numerical methods for solving Maxwell’s equations mentioned above are just a small number of the existing methods, going through all of them is out of the scope of this thesis. Instead, we will focus on the FDTD method and its underlying physics.

2.1. Finite-Difference Time-Domain

The Finite-Difference Time-Domain (FDTD) method was first proposed by Kane S. Yee in 1966 [17]. In his original paper, Yee described a way to solve Maxwell’s equations based on the central difference approximations of the spatial and temporal derivatives of the curl-equations. However, the main issue with his method was the lack of sufficient computing power at that time, furthermore the FDTD method is computationally expensive even in today’s standards.

The FDTD method is a simple yet vary powerful technique capable of solving extremely complex electromagnetic problems which is a key factor for its popularity [20].

FDTD method makes no assumptions about the direction of light propagation. It has no approximations other than the finite sized mesh and finite sized time step, therefore it is very accurate. Finally, the FDTD algorithm scales well with parallelization, so it is well suited for modern, multi-core and multi-processor computers as well as high performance computing clusters [20].

Because FDTD is such a versatile technique, it can address a wide range of applications. These include photonic crystals, plasmonics, CMOS image sensors, nanoparticle scattering and absorption, nano-patterned solar cells, OLEDS and LEDs, gratings and integrated optics to name just a few [19]. It is implemented in more than 30 different software packages [20]. The FDTD is a time-domain method, i.e. it excites the system with an electromagnetic pulse in time (typically on the order of femto-seconds) this excitation plus has a broad frequency range (i.e. contains many wavelengths), therefore, the system's wavelength response can be obtained from a single simulation [2]. It is also a fully-vectorial simulation method since it solves for all vector components of the electric and magnetic fields. FDTD is typically used when the feature size is on the order of the wavelength [2]. This wavelength scale regime where diffraction, interference, coherence and other similar effects play a critical role is called "wave optics" [19]. If the feature size is much larger than the wavelength used then it is better to use other methods such as ray tracing for more efficient simulations.

2.1.1. Theory

The main idea of the FDTD is to numerically solve Maxwell's equations on a discrete grid in both space and time, and derivatives are replaced with finite differences. For the sake of simplicity, we will explain the FDTD method in a one dimensional propagation; namely a plane wave propagating in free space which is a linear, isotropic, homogenous, non-conducting and charge free medium. Maxwell's curl equations can be written as follows

$$\frac{\partial \mathbf{E}}{\partial \mathrm{t}} = \frac{1}{\varepsilon_0} \nabla \times \mathbf{H} \tag{2.1}$$

$$\frac{\partial \mathbf{H}}{\partial \mathrm{t}} = \frac{-1}{\mu_0} \nabla \times \mathbf{E} \tag{2.2}$$

However, for one-dimensional case where a plane wave is traveling towards the positive z direction in free space, equations (2.1), (2.2) can be rewritten as

$$\frac{\partial E_x}{\partial t} = \frac{-1}{\varepsilon_0} \cdot \frac{\partial H_y}{\partial z} \tag{2.3}$$

$$\frac{\partial H_y}{\partial t} = \frac{-1}{\mu_0} \cdot \frac{\partial E_x}{\partial z} \tag{2.4}$$

The idea behind the FDTD method is to discretize E_x and H_y in both space and time. i.e. E_x and H_y are shifted in space by half spatial step (length) and in time by half a time step when approximating the derivatives with a central difference. The left part of equation (2.3) will be rewritten as an example in order to clarify the idea furthermore

$$\frac{\partial E_x^{n\Delta t}}{\partial t} = \frac{E_x^{(n+0.5)\Delta t}(k.\Delta z) - E_x^{(n-0.5)\Delta t}(k.\Delta z)}{\Delta t} \tag{2.5}$$

The super script denotes the time step with Δt representing time resolution, Δz denotes the spatial resolution (sometimes called mesh resolution or spatial step). k, n are an integers. In other words, Equation (2.5) indicates that the derivative of the electric field at time $n\Delta t$ can be expressed as a central difference of its values at times $(n+0.5).\Delta t$ and $(n-0.5).\Delta t$.

As for the right part of equation (2.3), it approximates the derivative of the magnetic field at point $k.\Delta z$ as a central difference of its values at points $(k+0.5).\Delta z$ and $(k-0.5).\Delta z$. The right part of equation (2.3) can be rewritten as

$$\frac{-1}{\varepsilon_0} \cdot \frac{\partial H_y^{n\Delta t}}{\partial z} = \frac{-1}{\varepsilon_0} \cdot \frac{H_y^{n\Delta t}\big((k+0.5).\Delta z\big) - H_y^{n\Delta t}\big((k-0.5).\Delta z\big)}{\Delta z} \tag{2.6}$$

From equations (2.5) and (2.6), it is possible to rewrite equation (2.3) as

$$\frac{E_x^{(n+0.5)\Delta t}(k.\Delta z) - E_x^{(n-0.5)\Delta t}(k.\Delta z)}{\Delta t} = \frac{-1}{\varepsilon_0} \cdot \frac{H_y^{n\Delta t}\big((k+0.5).\Delta z\big) - H_y^{n\Delta t}((k-0.5).\Delta z)}{\Delta z} \tag{2.7}$$

By applying the same logic on equation (2.4), it is possible to rewrite it as

$$\frac{H_y^{(n+1)\Delta t}\big((k+0.5).\Delta z\big) - H_y^{n\Delta t}\big((k+0.5).\Delta z\big)}{\Delta t} = \frac{-1}{\mu_0}\cdot\frac{E_x^{(n+0.5)\Delta t}\big((k+1).\Delta z\big) - E_x^{(n+0.5)\Delta t}(k.\Delta z)}{\Delta z} \qquad (2.8)$$

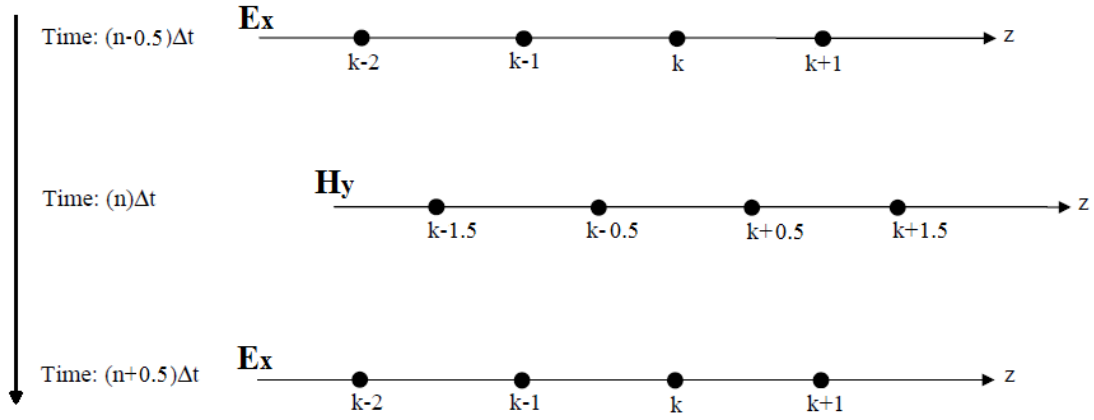


Figure 2.1. *A schematic view of the FDTD algorithm to clarify discretization in space and time*

In this scheme, all magnetic field values are first calculated followed by the electrical field values. In other words, the magnetic field values at points …, (k-1.5).Δz, (k-0.5).Δz, (k+0.5).Δz, … and times …, (n-1).Δt, n.Δt, (n+1).Δt, … are first calculated then the electric field values at points …, (k-1).Δz, k.Δz, (k+1).Δz, … and times …, (n-1.5).Δt, (n-0.5).Δt, (n+0.5).Δt, … it is better expressed in table 2.1.

Table 2.1. *The positions and times at which H_y and E_x are calculated. Note that H_y and E_x are never known for the same position. This table explains how space and time discretization is done in FDTD.*

	Position	Time
H_y	(k-1.5).Δz, (k-0.5).Δz, (k+0.5).Δz, …	(n-1).Δt, n.Δt, (n+1).Δt, …
E_x	(k-1).Δz, k.Δz, (k+1).Δz, …	(n-1.5).Δt, (n-0.5).Δt, (n+0.5).Δt, …

The explicit FDTD update equations can be derived from (2.7) and (2.8) as follows

$$E_x^{(n+0.5)\Delta t}(k.\Delta z) =$$

$$E_x^{(n-0.5)\Delta t}(k.\Delta z) + \frac{\Delta t}{\varepsilon_0.\Delta z} \cdot (H_y^{n\Delta t}\big((k-0.5).\Delta z\big) - H_y^{n\Delta t}((k+0.5).\Delta z)) \quad (2.9)$$

$$H_y^{(n+1)\Delta t}((k+0.5).\Delta z) =$$

$$H_y^{n\Delta t}((k+0.5).\Delta z) + \frac{\Delta t}{\mu_0.\Delta z} \cdot (E_x^{(n+0.5)\Delta t}(k.\Delta z) - E_x^{(n+0.5)\Delta t}((k+1).\Delta z)) \quad (2.10)$$

Although these equations may seem very complicated but in reality, they are not. They can be easily implemented using a computer code. A key factor that determines the accuracy of the FDTD simulation is the spatial resolution Δz and the time resolution Δt. Overestimating them would increase the simulation time significantly (simulation time is proportional to $1/\Delta z^4$ [2]) while underestimating them produces inaccurate results. Therefore a balance between the desired accuracy and the simulation time must be struck. Δz is usually defined by the number of points per wavelength in the material. As a rule of thumb, initial FDTD simulations are performed with with 6 points per wavelength. The obtained results will be within 10 or 20% of the correct answer [19]. If the number of points per wavelength is to be increased to 10, then the obtained results will be within 1 to 2% of the correct result. High number of points per wavelength beyond 20 is rarely needed and is used for special cases like simulating plasmonic effects where high light confinement can occur. In this case very high resolution is required to resolve geometric features [19]. It is also important to note that the points per wavelength should be defined with respect to the wavelength in the medium (not the free space wavelength), therefore, for similar accuracy a higher number of points per wavelength should be used in materials with higher refractive index [19].

Lumerical's FDTD Solutions provides a simple mesh accuracy setting that targets a minimum points per wavelength in all regions of the simulation and automatically adapts to the refractive index of the different materials. The mesh accuracy of 1 through 8 corresponds to points per wavelength targets of 6, 10, 14, 18, … up to 34. The default value is 2 which is appropriate for most initial simulations.

As for the time step Δt, for stability reasons, a field component must not propagate more than one Δz in a time step. This is expressed in equation (2.11) as follows

$$\Delta t \leq \frac{\Delta z}{C_0} \tag{2.11}$$

Where C_0 is the speed of light. Typically, to keep a safe margin, half of the number resulted from equation (2.11) is used as a time step.

2.1.2. Weaknesses of Finite-Difference Time-Domain method

Although FDTD method has many advantages and strengths, it also has weaknesses. For example, one of its major weaknesses is its requirement for full discretization of the electric and magnetic fields throughout the entire simulation volume [20], such discretization forces the FDTD to simulate large and empty volumes with no inhomogeneities. Another problem comes from large geometries that contain very fine features which demands smaller mesh size and consequently an inordinately smaller time step [20]. Therefore, larger number of time iterations is required which leads to higher simulation time. Furthermore, simulating a light trapping structure (e.g. a ring resonator) using FDTD may take very long time until a steady state solution is reached. Finally, FDTD suffers from the stair-casing effect especially on the edges of the geometry understudy [20].

3. THE NEED FOR OPTIMIZATION ALGORITHMS

Grid search or parameter sweep is a very simple method to search for the optimal parameters of a given model. For the sake of simplicity, let the model be represented by the function F which has only one parameter x as follows.

$$F(x) = y \tag{3.1}$$

The aim is to find $x_{(optimal)}$ at which F outputs an optimal value $y_{(optimal)}$, this dictates that a systematic scan of the possible values of x must be done followed by the evaluation of the corresponding values of F and how they are compared to the desired optimal value $y_{(optimal)}$. This process can be efficient and practical for models that have low parameter count n (typically less than three) with small ranges. However, the amount of the required evaluations grows exponentially (It becomes in the order of C^n, where C is a constant) as the model's parameters and their respective ranges increase. As a result, the required computational resources and time become a serious limitation especially if the evaluation process is computationally expensive. Therefore, methods other than grid search should be used to overcome the aforementioned limitation.

The need for optimization algorithms comes from the fact that exploring the parameter space of a model which has more than a few parameters using grid search is a highly non efficient process in terms of computational resources needed and time required. Therefore it is often dismissed due to being impractical. In fact, studies have shown empirically and theoretically that, for a given model, the random search for the optimal parameters yields good or better results than the grid search for a fraction of the computation time [21]. Optimization algorithms, on the other hand, are designed to overcome the aforementioned problem by exploring the parameter space "intelligently" which leads to a substantial decrease in the required computational resources and time. This however, comes at a cost; it is much more difficult to implement an optimization algorithm for a particular problem compared to implementing a simple grid search. The added layer of complexity includes choosing the most suitable optimization algorithm for the particular problem understudy and formulating the problem accordingly.

3.1. Optimization Algorithm Types

Optimization algorithms can be broadly divided into three main categories; physics-based, geography-based and biology-based algorithms [22]. They can also be categorized differently but it is out of the scope of this thesis.

3.1.1. Physics-based optimization algorithms

Physics-based optimization algorithms imitate the physical properties and physical behavior of matter [22]. For example, Stimulated Annealing (SA) optimization algorithm mimics the real physical annealing process [23] in which, crystallinity level of a material is changed by a series of heating and cooling cycles until the optimal material properties (e.g., hardness) is reached. When given the right conditions, this algorithm asymptotically converges to the global optimum solution of the problem understudy [24]. Other examples include the Gravitational Search Algorithm (GSA) which employs laws similar to that of gravity and motion in order to find the optimal solution [25] and the Chaotic Optimization Algorithm (COA) which is suitable for problems resembling chaotic systems [26]; i.e., a small change in the initial parameters lead to very different future behavior. It also employs mechanisms to prevent the convergence to local optima [22].

3.1.2. Geography-based optimization algorithms

Geography-based optimization algorithms tend to view the parameter space as if it is a map. Possible solutions “travel” through the map according to a set of rules until a satisfactory solution is reached. The Tabu Search Algorithm (TSA) [27] and Imperialistic Competition Algorithm (ICA) [28] are examples of this kind of methods.

3.1.3. Biology-based optimization algorithms

Biology-based optimization algorithms are derived from the analogy with the natural evolution and biological activities [22]. There are two main sub-categories of biology-based optimization algorithms; namely evolution-based and swarm-based algorithms [22].

Evolution-based optimization algorithms are stochastic search methods which imitate the processes of evolution, natural selection and social behavior of various life forms and animal species [22]. There are a number of evolution-based optimization algorithms such as the Genetic Algorithm (GA) in which the mechanisms of natural selection are imitated. In GA the possible solutions called “population” are first initialized

in the parameter space with a number of "chromosomes". Each individual is then evaluated. Then new individuals are created by applying selection, cross over and mutation on the population. The newly created individuals may be (and hopefully) better than their parents [29]. The process is repeated for a number of generations until a satisfactory solution has been found. It is worth mentioning that GA has numerous issues like poor and slow parameter convergence, it is not guaranteed that the found solution is the optimal one and sometimes it losses the best solution after finding it [30]. Other evolution-based optimization algorithms include Evolutionary Programming (EP), Evolutionary Strategy (ES) and Differential Evolution (DE) algorithm …etc.

Swarm-based optimization algorithms, as the name dictates, employ a swarm which consists of a large number of simple entities (sometimes called "agents") that have the ability to interact with each other and with their environment. There are two main principles that are persistent in this family of optimization algorithms; first, each entity in the swarm (representing one possible solution) is in itself a simple entity with limited capabilities. However, the collaboration between these entities creates some sort of collective "intelligence" for the swarm as a whole. Second, there is no central control mechanism for the swarm but rather some laws that must be obeyed by all entities in the swarm. This family of optimization algorithms have many good merits; they are capable of finding solutions with low computational cost, they are fast and are able to find robust solutions for several complex problems [22]. To put things into perspective, entities in the swarm are analogous to animals (or other life forms) in their respective groups. Examples of this kind of algorithms which were originally inspired by nature include Ant Colony Optimization (ACO) [31], Bacterial Foraging Optimization (BFO) [32, 33], Artificial Bee Colony (ABC) [34] and Particle Swarm Optimization (PSO) [35] …etc.

PSO was reported to have a total number of related publications that is higher than the sum of all publications related to the other nature-inspired optimization algorithms [36]. The aforementioned rigorous study indicates how popular PSO became and reports its application in a wide variety of scientific and engineering fields.

3.2. Particle Swarm Optimization

In 1995, J. Kennedy and R. Eberhart published the first paper about Particle Swarm Optimization [35] which was inspired by the social behavior of birds in their flocks. Ever

since its introduction, PSO became one of the most useful and popular optimization algorithms. It was used to solve various optimization problems in various fields such as electrical and electronics engineering, mechanical engineering, communication theory, automation, control systems, medicine, chemistry and biology [36].

PSO is a popular optimization algorithm. It owes its popularity to its simplicity and to being a heuristic optimization algorithm [36] which means that PSO have few or no assumptions about the problem being optimized. In addition, it is suitable for exploring a large, multi-dimensional parameter space at a low computational cost. Unlike classic optimization algorithms such as gradient descent, PSO does not require a differentiable problem which adds to its simplicity. PSO overcomes most of the limitations encountered by the Genetic Algorithm [37] in that more interaction in the group enhances rather than impedes the progress toward the solution. Furthermore, a PSO has some sort of memory embedded within it whereas GA does not have any memory due to the destruction of the previous knowledge of the problem while creating the next generation. In PSO if a particle fly past an optima, it is likely to get back to it [37].

3.2.1. The standard particle swarm optimization theory

A large number of "particles" called "swarm of particles" each of which represents a potential solution of the problem are first initialized with random positions and velocities in the parameter space, then they move around according to a set of rules.

In PSO, there is no central control mechanism that dictates how each particle should behave but rather, particles move independently following a set of rules which incorporates a certain degree of randomness. The interactions between these independent, unintelligent particles lead to the emergence of a collective "intelligent" behavior of the swarm.

Each particle has three properties; i) The particle's current coordinates vector in the parameter space denoted by $\boldsymbol{X_n}$, which is updated according to Equation 3.2. ii) The particle's current velocity vector, denoted by $\boldsymbol{V_n}$ and updated according to Equation 3.3. iii) is the coordinates and value of the best obtained solution so far and it is called personal best or "*pbest*". The "global best" or *gbest* which is the best overall solution obtained by any particle in the swarm and its position can be seen by all particles in the swarm and it is stored separately.

$$X_n(t+1) = X_n(t) + V_n(t+1) \quad (3.2)$$

Where $\boldsymbol{X_n}$ and $\boldsymbol{V_n}$ are the particle's coordinates and velocity in the n^{th} dimension. T is the time step. The velocity is updated according to the following equation

$$V_n(t+1) = w.V_n(t) + C_1.R_1\big(pbest(n,t) - X_n(t)\big) + C_2.R_2(gbest(t) - X_n(t)) \quad (3.3)$$

Where *R1* and *R2* are random variables which have uniformly distributed values between 0 and 1. *w* is the inertia weight used to balance the global exploration and local exploitation, *C1* and *C2* are positive constant parameters called "acceleration coefficients" [36]. In equation 3.2, there are three main terms; the first one is called "inertia" and provides the particle with "momentum" in order for it to move around in the parameter space. The second one is called the "cognitive" component, which resembles the particles' thinking of other particles. This term encourages the particle to move towards *pbest*. The third term is the "cooperation" component which represents the collaborative effect of the particles to find the global optimal solution [36].

A detailed flow chart of how PSO works is shown in Figure 3.1. It is worth mentioning that the fitness function represents the function *F* in equation 3.1. In other words, the fitness function is the same as the function representing the model (problem) understudy and sometimes it is called the cost function or the Figure of Merit (FOM) function. In this thesis we used the FOM term in section 5.

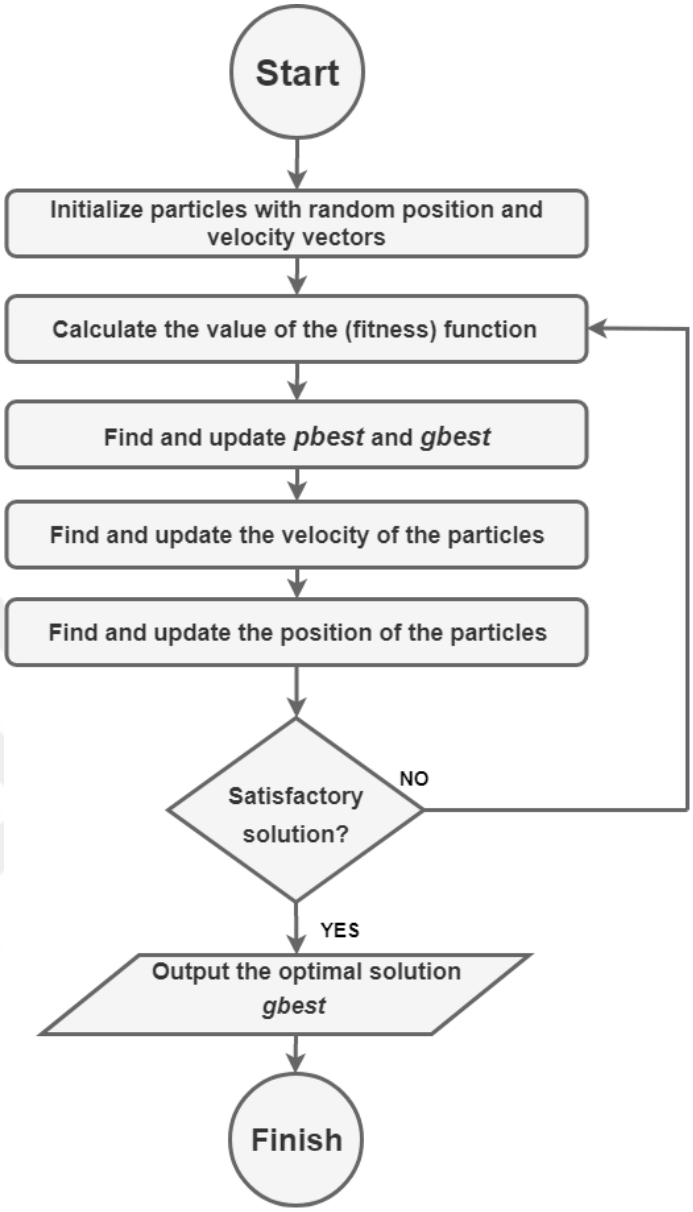


Figure 3.1. *A flow chart of the particle swarm algorithm. The algorithm starts by initializing the particles in the parameters space with random position and velocity vectors, then it calculates the value of the fitness function for each particle. For all particles, if the obtained fitness values are better that the current pbest then update pbest to the newly found fitness value. Compare between pbest of all particles, if there is any of them is better than the current gbest then update gbest to the highest found pbest. Update the particles' velocity and position vectors according to equations 3.2 and 3.3 respectively. Evaluate gbest, if it is a sufficient solution then output it and declare the end of the algorithm, otherwise go to step 2 and recalculate the fitness function for all particles in their new positions*

3.3. Lumerical's Implementation of Particle Swarm Optimization

Lumerical FDTD solutions® (V. 8.19.1584) has a built-in PSO functionality. For convenience, a graphical user interface window as shown in figure 3.2 is used to setup the optimization. It is worth mentioning that Lumerical allows the user to write a custom code for any optimization in the advanced tab. In this thesis however, we used the built-in PSO which is slightly more advanced than the standard PSO described in section 3.2.1. It offers either to minimize or to maximize the FOM function. The swarm population is set by the Generation size tab and the number of time steps is set by the Maximum generations tab. The total number of simulations is the multiplication of these two numbers. An option of resetting the random number generator is also available. When checked, it ensures consistent results between runs. Tolerance tab allows early termination of the optimization if the change in FOM is lower than the assigned tolerance. The user can add or remove parameters from the parameter tab. The ranges of these parameters must be assigned along with their type and unit. Finally, the user must add a FOM which can be a direct result of the simulation or can be defined by the user using the scripting language of Lumerical.

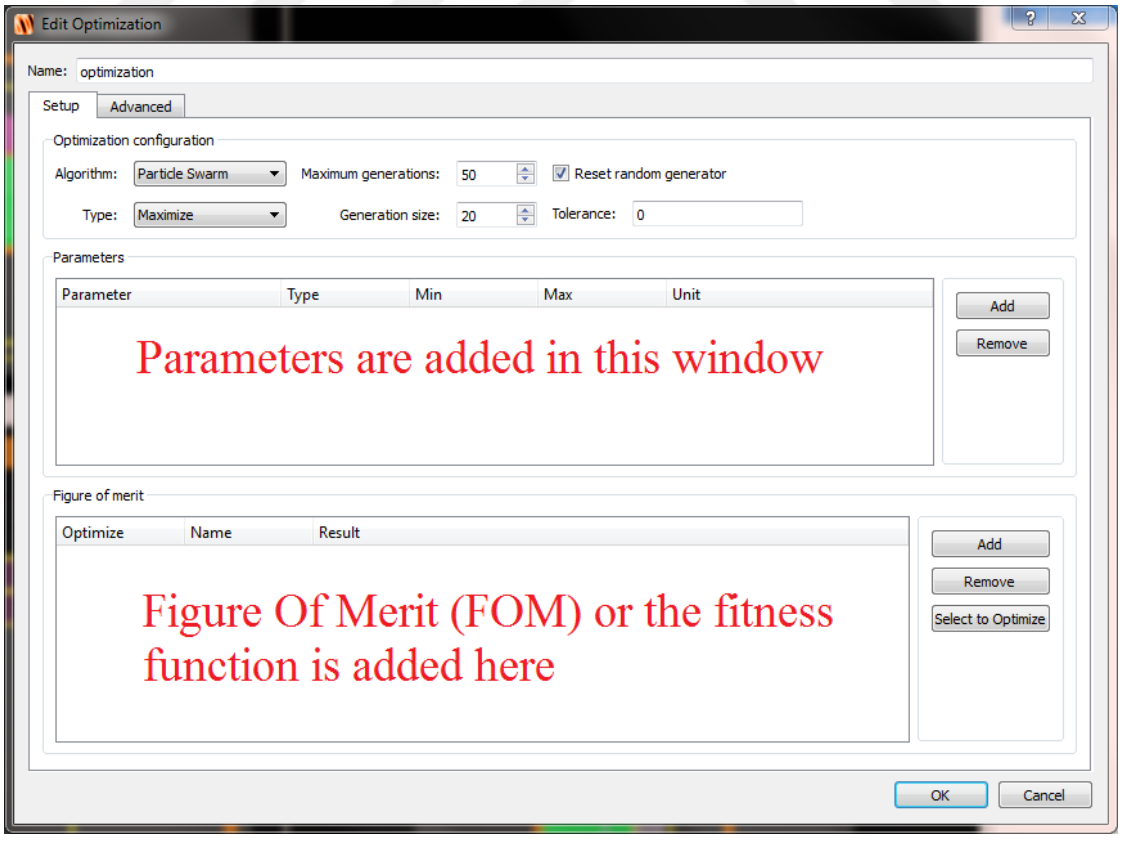


Figure 3.2. *Lumerical's graphical user interface window for setting up optimizations*

4. OPTICAL WAVEGUIDES

Optical waveguides are essential optical components since they are the foundation on which optical circuits and devices are realized. They are the backbone of the optical circuits since they provide a low-loss path for light energy to travel through. Optical waveguides can be categorized in many different ways. For example they can be classified according to their geometry as planar or non-planar waveguides, or according to their refractive index profile as step index or gradient index. Optical waveguides can also be classified based on their modal behavior as single-mode or multi-mode, or according to the material platform used to create them such as polymers, glasses or semiconductors … etc. [38].

For this thesis, our interest is to investigate planar, step index, single-mode waveguides made from Al_2O_3 grown using Atomic Layer Deposition (ALD) over SiO_2 substrate due to the following reasons; First, planar waveguides were chosen because we aim to use these waveguides for realizing integrated optical devices and typically this is done using waveguides with planar geometry as opposed to circular geometry used in optical fibers. Second, we chose waveguides that have step index profile since they are easier to fabricate and in-line with other research carried or being carried out in Micro/Nano Devices And Systems (MIDAS) clean room in Eskişehir technical university. Third, our interest is to have single-mode waveguides due to their importance for interferometry-based devices and sensors. Finally, we chose amorphous Al_2O_3 grown using ALD over SiO_2 substrate as our material system due to its high potential and functionality, low cost and availability. Amorphous materials exhibit isotropic properties, i.e., their electrical and optical properties are non-directional. Therefore, they are commonly used in modern communications as base materials for building functional optical devices. Other advantages include good dielectric constant, good stability, high hardness, and it is an excellent host of rare-earth elements [39, 40]. Moreover, Al_2O_3-based waveguides have been successfully fabricated and characterized in MIDAS clean room in Eskişehir Technical University [40].

Possible planar waveguides geometry include (but not limited to) buried channel waveguides, strip-loaded, ARROW, rib, ridge, diffused, slot …etc. as shown in figure 4.1.

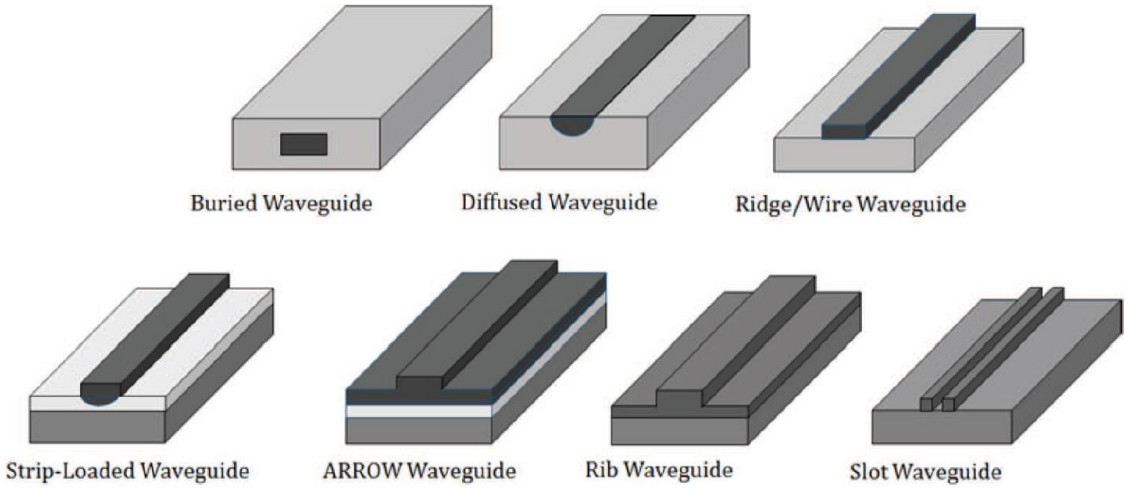


Figure 4.1. *Different waveguide geometries. (Adopted from S. Kumar et. al. [38])*

Of these geometries, ridge, rip and buried channel waveguides are of our interest. Both the rip and buried channel geometries were fabricated from Al_2O_3 (grown using sputtering) over SiO_2 and their fabrication and characterization is reported in detail in [41]. Each one of the aforementioned geometries has its own strengths and weaknesses.

The buried channel waveguides has many advantages such as lower optical losses than the ridge waveguides because of the refractive index contrast between the core and the surroundings. In the case of the buried channel waveguide, the cladding is surrounding the waveguide's core from all directions (except for the propagation direction) and the refractive index contrast between the core and cladding is lower than that of the core and air as in ridge waveguides therefore less optical scattering occur. In addition, having the upper cladding helps to prevent physical damage to the waveguide's core [41].

The rip geometry has a small ridge on top which is fabricated by means of shallow etching a patterned Al_2O_3 film. However, the thickness of this ridge (on the order of a few tens of nanometers) is a very influential parameter for this geometry. Therefore, even the slightest of variations in the ridge's thickness heavily affect the performance leading to inconsistency issue. On the other hand, this geometry offers lower scattering-induced losses due to etching compared to the ridge geometry [41]. It is worth mentioning that the rip geometry has a low lateral confinement of light energy which restricts its ability to form low-loss tight bends [41].

The ridge geometry is simple, easy to fabricate and allows tighter bends than that of the rip. One drawback is that its confinement factor is lower than that of the rip [41]. This is especially important if the Al_2O_3 is to be doped with Erbium (Er^{+3}) ions where maximum interaction between the light and the Er^{+3} ions is required for lasing to occur.

4.1. Rare-Earth-Doped Waveguides

Despite the efforts to develop an active light emitting functionality in silicon, a limited success was achieved due to the low emission efficiency of silicon which is the result of the silicon's indirect bandgap [41]. The attention was shifted towards other materials which can be electrically pumped such as III-IV material platforms to create integrated lasers and waveguide amplifiers [41]. However, the added complexity of integrating such materials and the associated costs were too high which encouraged the search for alternatives. One possible solution is to grow a host material using silicon platform and to dope this host with a rare-earth element which can be pumped optically [41]. Amongst rare-earth elements, Erbium stands out since its emission spectrum is in-line with the second communication window at 1550 nm wavelengths while it can be pumped optically at 1480 nm. Amongst possible host materials, amorphous a-Al_2O_3 (or simply Al_2O_3) stands out as a very good choice due to having high refractive index contrast with SiO_2 allowing for higher integration density. In addition, Al_2O_3 films can be deposited directly on SiO_2 substrates. Moreover, Al_2O_3 can be used as-is to create passive devices or it can be doped with rare-earth elements to create active devices. On top of that Al_2O_3 have excellent physical and chemical stability and its transparency at visible and near infrared wavelengths is very good as well [42].

Amorphous Al_2O_3 makes a decent host for Er^{+3} ions since it fulfills the following requirements; i) Al_2O_3 can provide Er^{+3} ions with suitable bonding sites, ii) high Er^{+3} solubility in Al_2O_3 without clustering, iii) Al_2O_3 has low propagation loss at the characteristic emission of Er^{+3} wavelength (at around 1550 nm) [41]. Moreover, ALD can be used to grow Al_2O_3 films with higher concentrations of Er^{+3} while allowing a better control on the film thickness.

In this thesis, the material system is Al_2O_3 over SiO_2, the reason for not considering doped Al_2O_3 is due to the fact that doping with Er^{+3} ions has almost a negligible effect on the refractive index of the Al_2O_3 [41].

4.2. The Design of Al_2O_3 Based Waveguides

Based on what have been provided since the beginning of section 4, we decided to design and simulate a waveguide made from Al_2O_3 over SiO_2 with the following considerations; i) the geometry of the waveguide is a ridge, ii) the operational wavelength is the band between 1450 nm and 1580 nm (to ensure that the pump wavelength of Er^{+3} at 1480 nm and its characteristic emission at around 1550 nm are both included in the operational wavelength band), iii) the waveguide should have a single mode operation for the all wavelengths in the operational wavelengths range, iv) It is imperative to design waveguides with the largest possible cross-sectional area. However this task is not simple since it is impractical to grow very thick Al_2O_3 films using ALD also, enlarging the cross sectional area compromises the single-mode operation of the waveguide. v) Bent waveguides created using the same geometry should be checked for single-mode operation within the wavelength range of interest. This procedure had to be performed since bent waveguides tend to have a different modal behavior than their straight counterparts. For example a straight waveguide may be able to support a single mode, however if the same waveguide is to be bent then it may have no modes supported, vi) the simulated design should be as close to reality as possible. Therefore real world measurements of the refractive index of the ALD grown Al_2O_3 films was used (reported in appendix A). In addition, the sidewalls angle should be included. J. Bradly [41] reported sidewalls angles between 56° to 68° while it was around 50° for the waveguides created in MIDAS clean room. Ideal 90^{o} (perpendicular) sidewalled waveguides are not fabricable thus simulating them will not reflect the real world performance accurately. In section 5.8 this effect is explained in greater detail. In all of the following simulations, a 50^{o} angled sidewalls are assumed. The approximate shape of the waveguide is shown in figure 4.2.

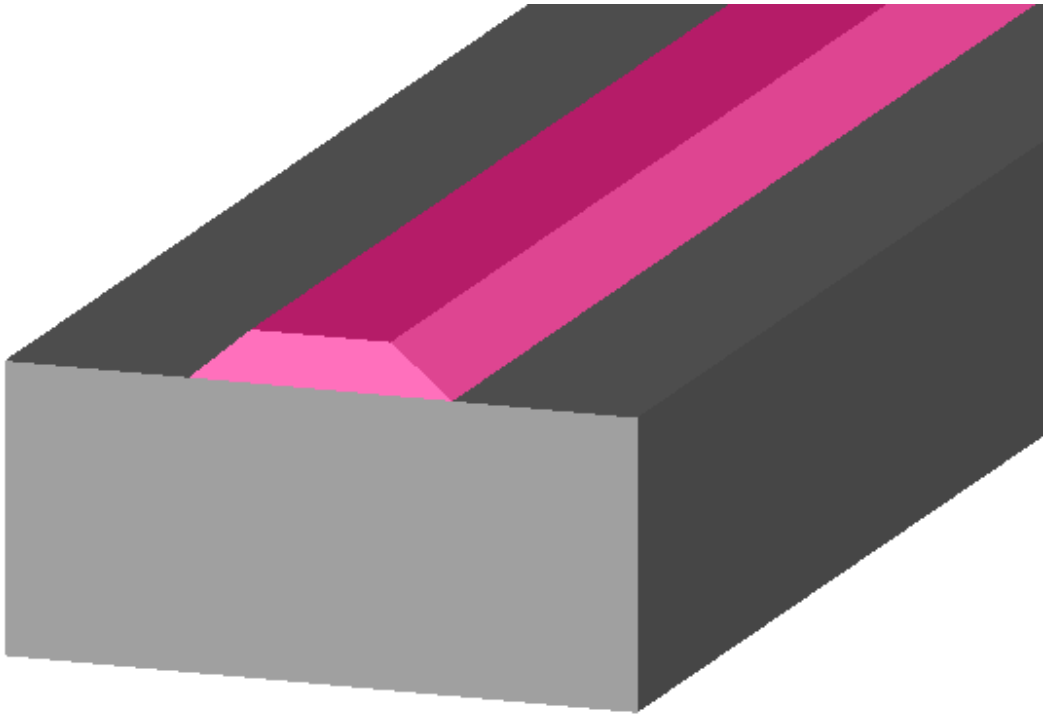


Figure 4.2. *An approximate 3D view of the ridge waveguide whose dimensions are not yet determined. Al_2O_3 and SiO_2 are represented in pink and grey respectively*

4.3. Simulating Al_2O_3 Waveguides in Lumerical

In order to simulate structures made from Al_2O_3 in Lumerical, Al_2O_3 must be first added to the Lumerical's material database. This is done by opening the material tab in Lumerical's user interface, then pressing on add button. Numerous material type choices are available. However sampled 3d data is most suited in this case. Data points of wavelength, real part of the refractive index and imaginary part of the refractive index are then entered. Figure 4.3 shows the material database tab with the added "*Alumina official*" material representing the amorphous Al_2O_3 with the entered data points. Only wavelengths ranging from 1.45 to 1.58 μm were considered with the imaginary part of the refractive index is zero across all wavelengths.

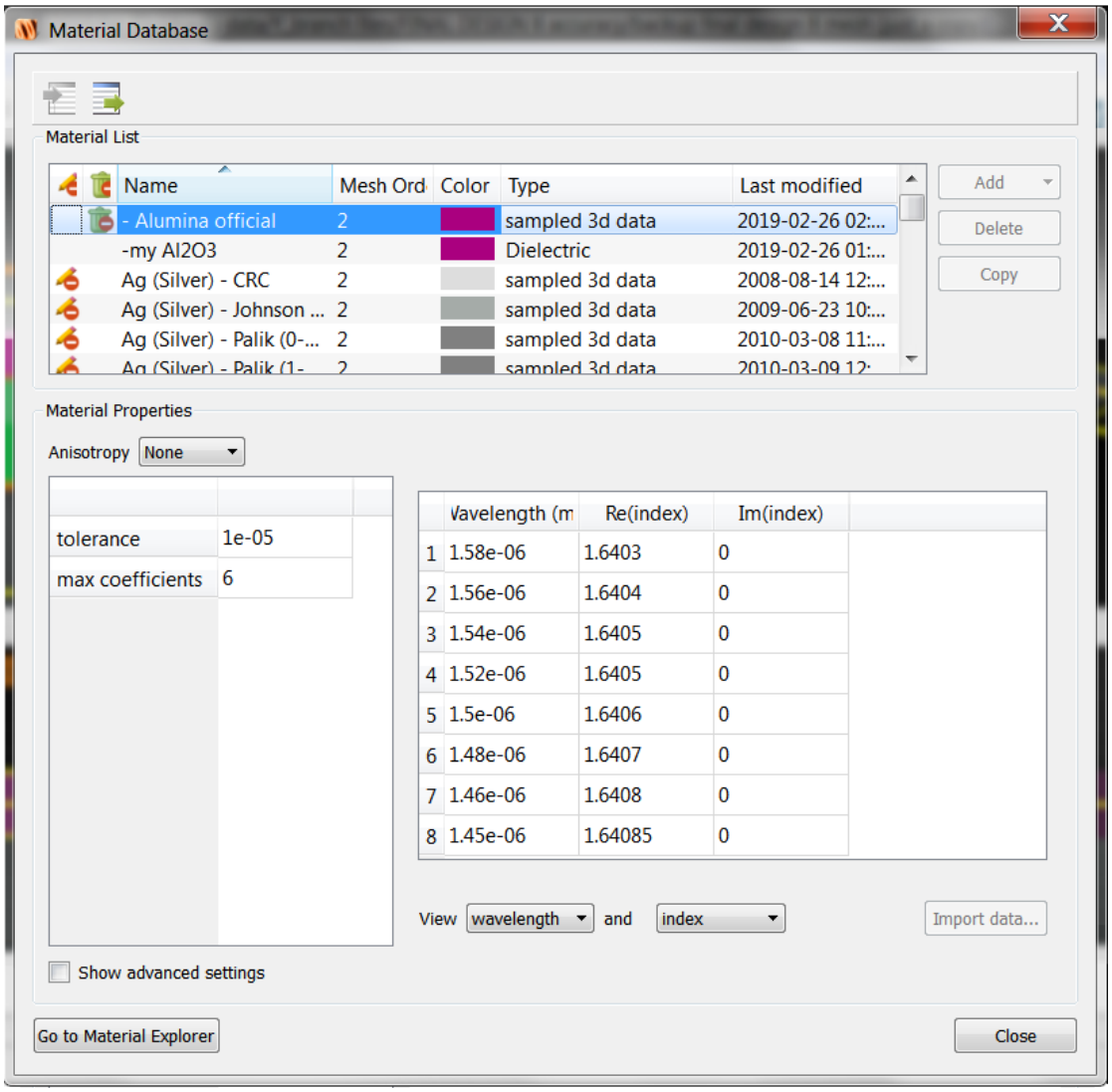


Figure 4.3. *The graphical user interface of the material tab in Lumerical FDTD solutions. Al_2O_3 data points are shown in material properties box*

Since the entered data consists of discrete points, Lumerical needs to interpolate them to form a continuous function. In order to make sure that Lumerical interpolates these data points correctly, Material explorer should be opened and the entered data should be fitted and plotted as figure 4.4 shows.

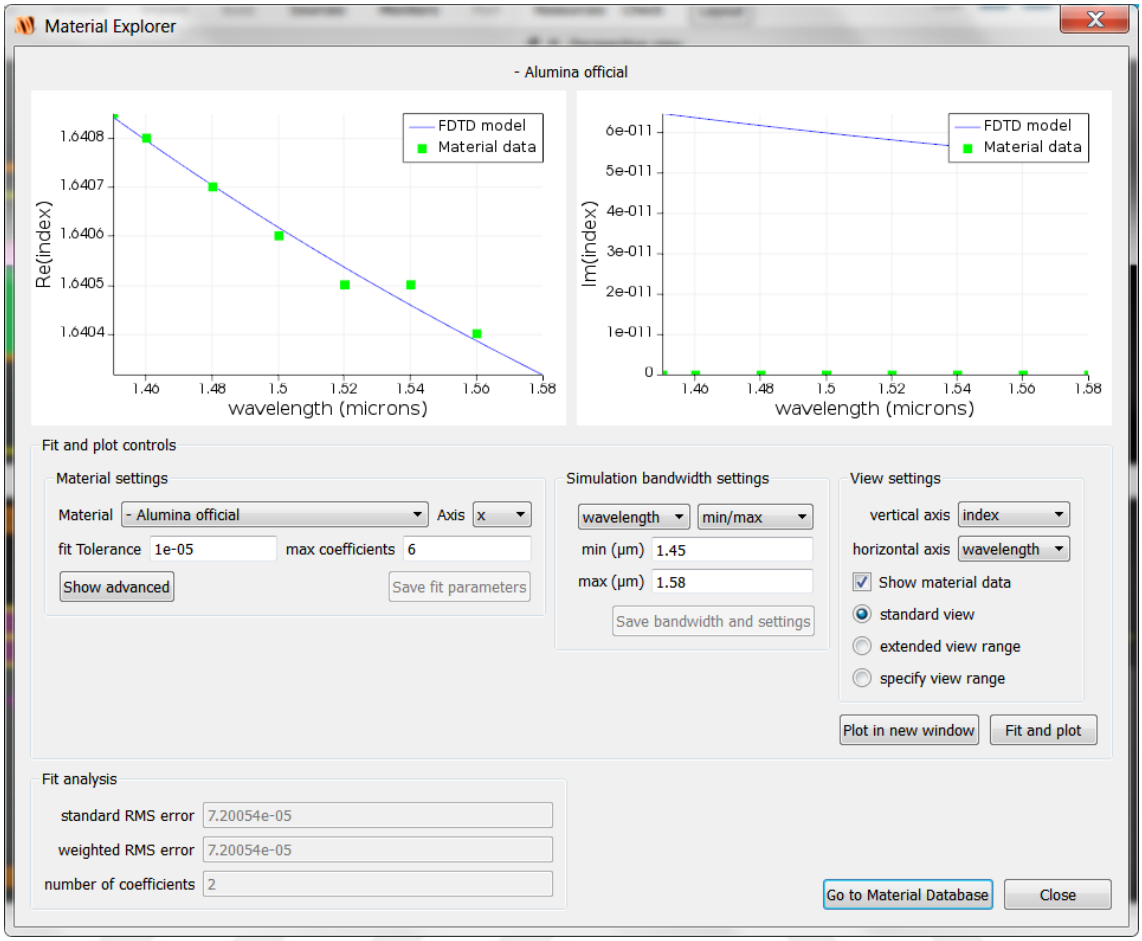


Figure 4.4. *Lumerical's Material Explorer tab. On the upper right portion of the window is a plot of the fitted real part of refractive index of Al_2O_3 (blue line) and the original data points (in green). On the upper left portion is a plot of the fitted imaginary part of refractive index of Al_2O_3. It is worth mentioning that while all data points is the second plot are at y=0 line, the fitted model shows a non-zero line and it is probably related to avoid numerical instability (the line has extremely low values close to zero)*

We used Lumerical's Mode® Solutions to find the supported modes. The aim was to find thickness range and the waveguide's base width within which the waveguide would only support single-mode operation for all operational wavelengths and for bend waveguides with 50 μm radius.

4.3.1. Simulation results of the ridge waveguide

As it turns out, the thickness can be varied between 0.8 and 1.1 μm (a safety margin is incorporated) and the best value for the waveguide's base width is 2.5 μm (a safety margin is also incorporated). Figure 4.5 shows a cross-sectional view of the designed waveguide with it parameters. Figure 4.6 shows a contour plot of the electrical field

intensity of the fundamental TE_0 mode supported within the designed waveguide. Waveguide's thickness, base width and operating wavelength are set to 0.87 μm, 2.5 μm and 1.55 μm respectively.

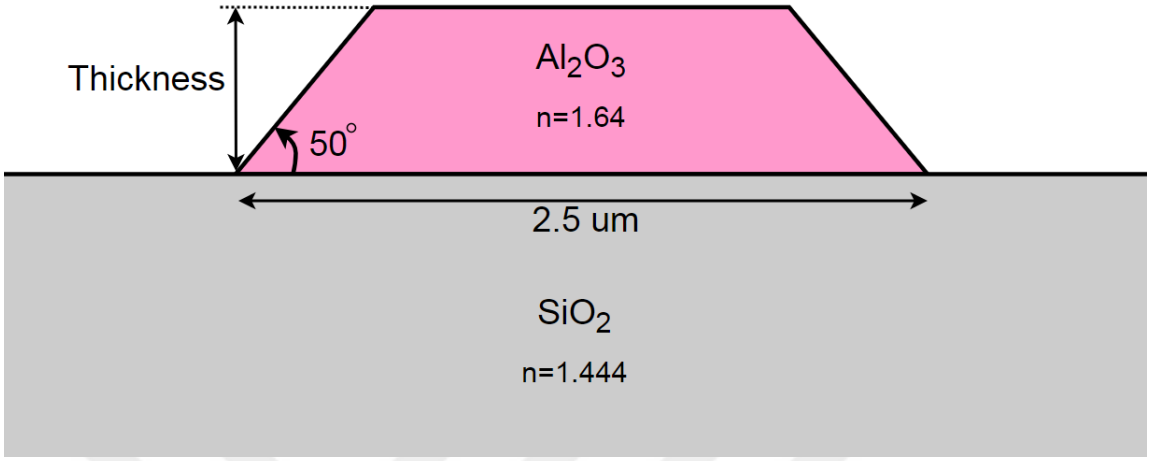


Figure 4.5. *Cross-sectional view of a realistic, single-mode ridge waveguide made of Al_2O_3, n=1.64 over SiO_2 n=1.444 substrate. The waveguide has a base width of 2.5μm and a 50˚ sloped sidewall. Thickness ranges from 0.8 to 1.1 μm. This geometry is suitable for wavelengths between 1450 to 1580 nm*

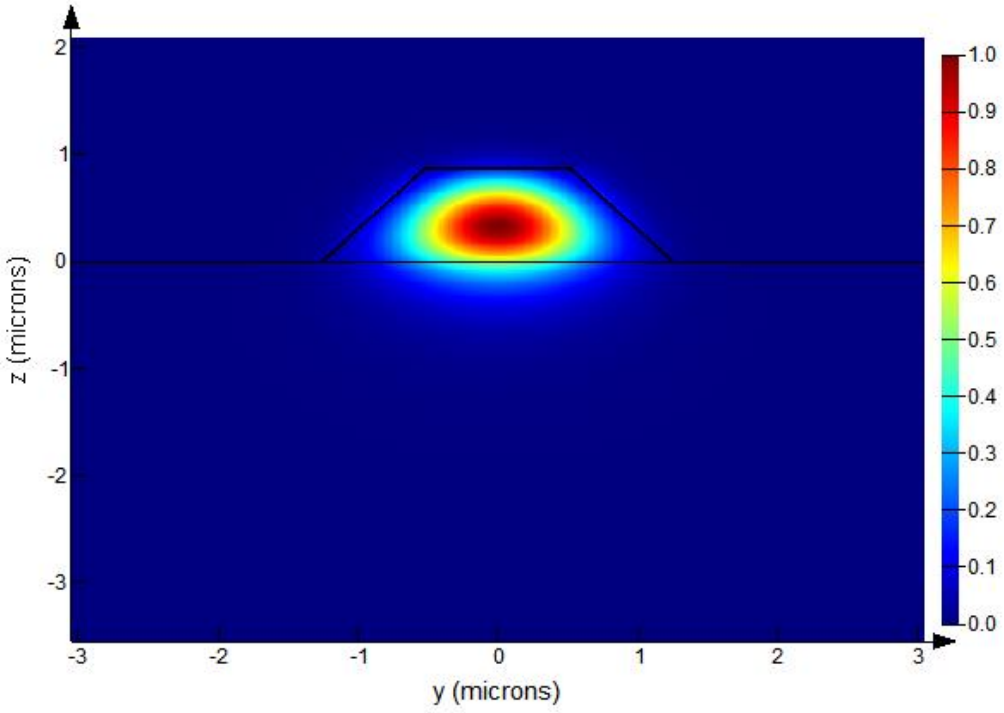


Figure 4.6. *A contour plot of the electrical field intensity of the fundamental mode TE_0 for a waveguide with 2.5 μm base width and with a thickness of 0.87 μm at 1.55 μm. The mode turned out to have an effective refractive index of 1.497. No other mode exists*

5. Y-BRANCH DESIGN AND OPTIMIZATION PROCESS

In this section, the design and optimization process of a realistic single-mode Y-branch is reported. The design is based on the waveguide geometry and material described in section 4.3.1 while the simulation was carried out using the FDTD method described in section 2.1. PSO (described in section 3.1) was employed to find the optimal geometry for the Y-branch. Finally, the effect of the structure's sidewalls angle on the fidelity of the simulation was investigated.

5.1. Motivation and Background

Y-branches are one of the most extensively studied photonic components in literature since they are basic building blocks for creating more complex photonic circuits. The design of a low-loss Y-branch which is insensitive to fabrication artifacts, temperature changes, wavelength and polarization direction while being easily fabricable remains elusive. Traditionally, Y-branch designers start with an analytically valid design, then they modify it to suit their particular application. This attitude is shown in the work of Sakai et al. [43]. Novel Y-branch design approaches based on refractive index tapering [44], total reflection [45], integrated prisms [46], [47], expanded truncated structural Y-branch [48] and wavefront matching [49], [50] were demonstrated with various degrees of performance. The introduction of the optimization algorithms to the photonic design brought novel and high-performance designs which often have inconceivable shapes to a human designer since they are generated by an algorithm. The Y-branch demonstrated by Lalau-Keraly et al. using the adjoint shape optimization method [51] illustrates this fact. Furthermore, the three-channel wavelength demultiplexer demonstrated by Su et al. [52] had arbitrary sub-micrometer holes which raised concerns about the fabricability of such designs using standard lithography. Therefore, it is imperative to develop a mechanism to enforce fabricability onto these algorithms. Zhang et al. designed a Y-branch waveguide using the particle swarm algorithm [53], our work builds on that implementation with some further modifications such as the development of new design parameters for a better controlled design with the area between the output waveguides also being parameterized. The material system was changed to a weakly guiding one consisting of amorphous Al_2O_3 over SiO_2 instead of silicon over SiO_2. In order to perform simulations which better reflect reality, structures with sloped sidewalls were used instead

of the ideal vertical ones. A minimum feature size of 1.5 μm was enforced to ensure fabricability and maximum yield within a 1 μm-resolution standard lithography process.

5.2. The Optimization Process of a Y-branch

The CAD environment in Lumerical FDTD solutions® [19] was used to parameterize, simulate and optimize a Y-branch that is based on the Al_2O_3 waveguide described in section 4.3.1. Lumerical's 2 Dimensional FDTD solver was employed to find the electrical and magnetic fields and other data which are then used to evaluate the Y-branch's performance. Lumerical's built-in PSO algorithm function was used to find the optimal values of the design parameters. The design parameters had initial values assigned by the author, then the PSO took over and started providing values to these parameters automatically. Each set of the generated parameters were applied to the design then evaluated until a certain number of iterations was reached. The next subsections are dedicated to explain the logic behind our parameterization and the way it was implemented in Lumerical.

A good geometry description through parameters or "design parameterization" plays a key role in helping the optimization algorithm to converge faster to the global optima. However, it is not always a straight-forward task and it needs some creativity from the designer's side. In our suggested design (see Figure 5.1), the body or the "taper" section of the Y-branch is defined by spline interpolation of 5 polynomials, *W1* to *W5*. *W1* is set to 2.5 μm to match the input waveguide's width which in turn prevents the formation of sharp corners. The gap area is defined by spline interpolation of 4 polynomials, *E1* to *E4* and terminated by a circle whose diameter *D* equals to *E1*. Finally, thickness, body length, and gap length are added as individual parameters. Special attention is given to the relations between these parameters.

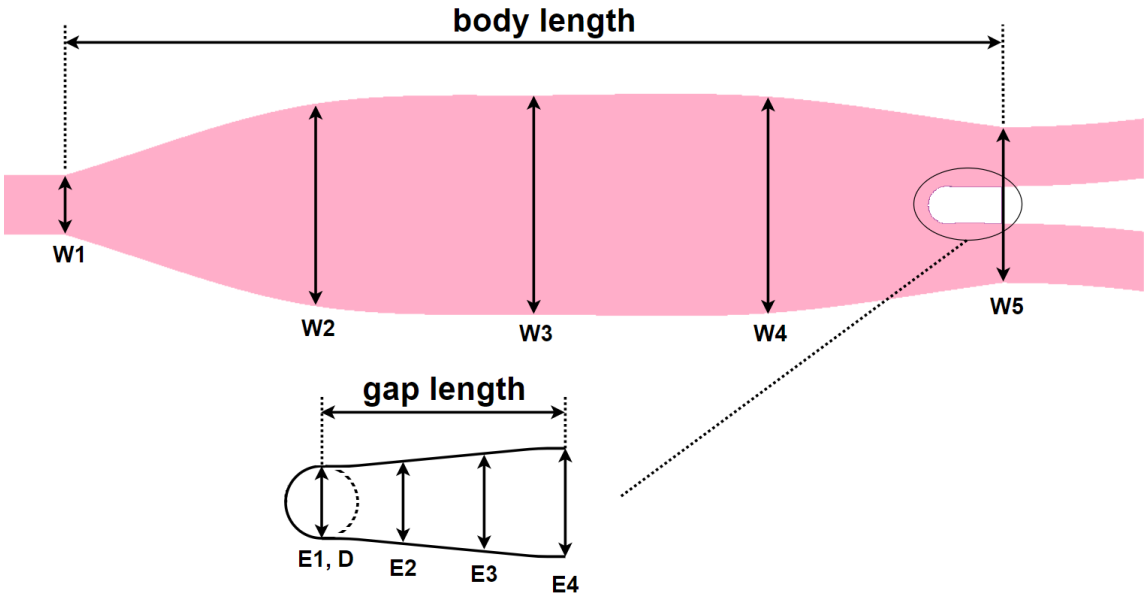


Figure 5.1. *Y-branch general layout and relevant parameters*

The output waveguides are rings with 50 μm radius to minimize losses associated with bent waveguides.

5.3. Y-branch Design Parameterization in Lumerical

In order to implement the parameterization described in the previous subsection in Lumerical, a combination of structure groups and scripts were employed. The process will be explained step by step starting from Lumerical's Object Tree.

5.3.1. Lumerical's Object Tree

Most CAD programs implement the idea of having different design objects that are combined to form a model and Lumerical is no exception. As Figure 5.2 shows, there are many objects in the Object Tree each with its own options and settings. The highest object in the hierarchy is called "model" and it contains some general information about the simulated model as a whole, it can also change the properties of other objects. As will be seen in the next section, it essential to write a custom script inside the model and the structure group objects to achieve the simulation's objectives.

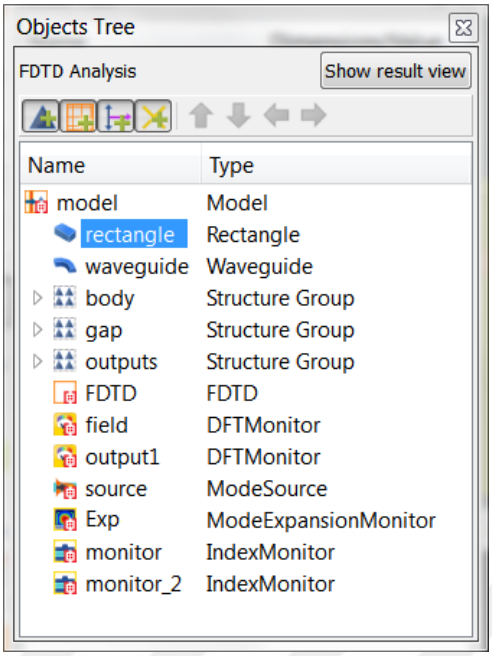


Figure 5.2. *Lumerical's FDTD Solutions Object Tree window*

5.3.1.1. Object Tree; rectangle

"Rectangle" object is right rectangular prism (3D object). Its dimensions, materials and other options are set by right clicking on it and changing the corresponding setting manually. It can also be created and/or modified by writing scripts commands on the script prompt directly or by writing the scripting commands on the "model" tab. For this specific simulation, this object resembles the SiO_2 substrate. Its dimensions were set to have large enough area to accommodate for the possible designs.

5.3.1.2. Object Tree; waveguide

The object "waveguide" is, as the name dictates, is a waveguide. However, its cross-section can be set to resemble a trapezoid as in figure 4.5. It is defined by its base width, sidewall angle, height and material. It is worth mentioning that the waveguide object does not have to be straight; it can be routed by setting its "poles". The sidewalls of this type of objects is smooth with no imperfections at all.

5.3.1.3. Object Tree; body

The "body" object is a structure group which means that it contains a number of objects within. Figure 5.3 shows the properties tab.

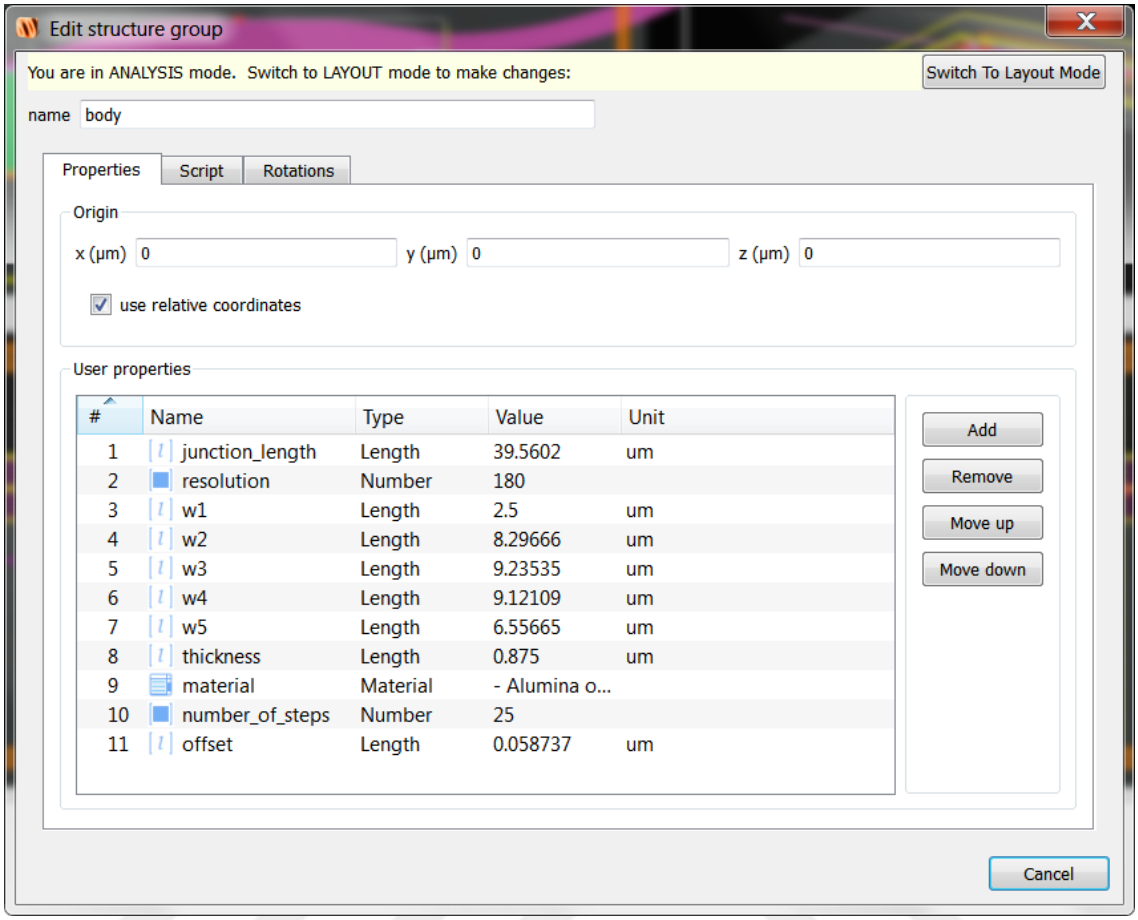


Figure 5.3. *The user interface for adding and/or removing different properties for the "body" structure group*

As previously mentioned, *W1* to *W5*, *junction_length* (which is body length) and thickness are the Y-branch's design parameters. All of the properties shown in Figure 5.3 are used in the Script tab which contains script code instructions. It is worth mentioning that Lumerical doesn't have a definite way of creating arbitrary shaped structures with sloped sidewalls (other than waveguides with constant width). Therefore, the sloped sidewalls are approximated by a staircase-like shape as shown in figure 5.5. This is achieved by offsetting the perimeter of the body of the Y-branch by a small amount to the inside 25 times starting from the base up. The resulting staircase would have little to no effect on the fidelity of the simulation (especially through scattering loss) since the slice height is much less than one-quarter of the minimum operating wavelength.

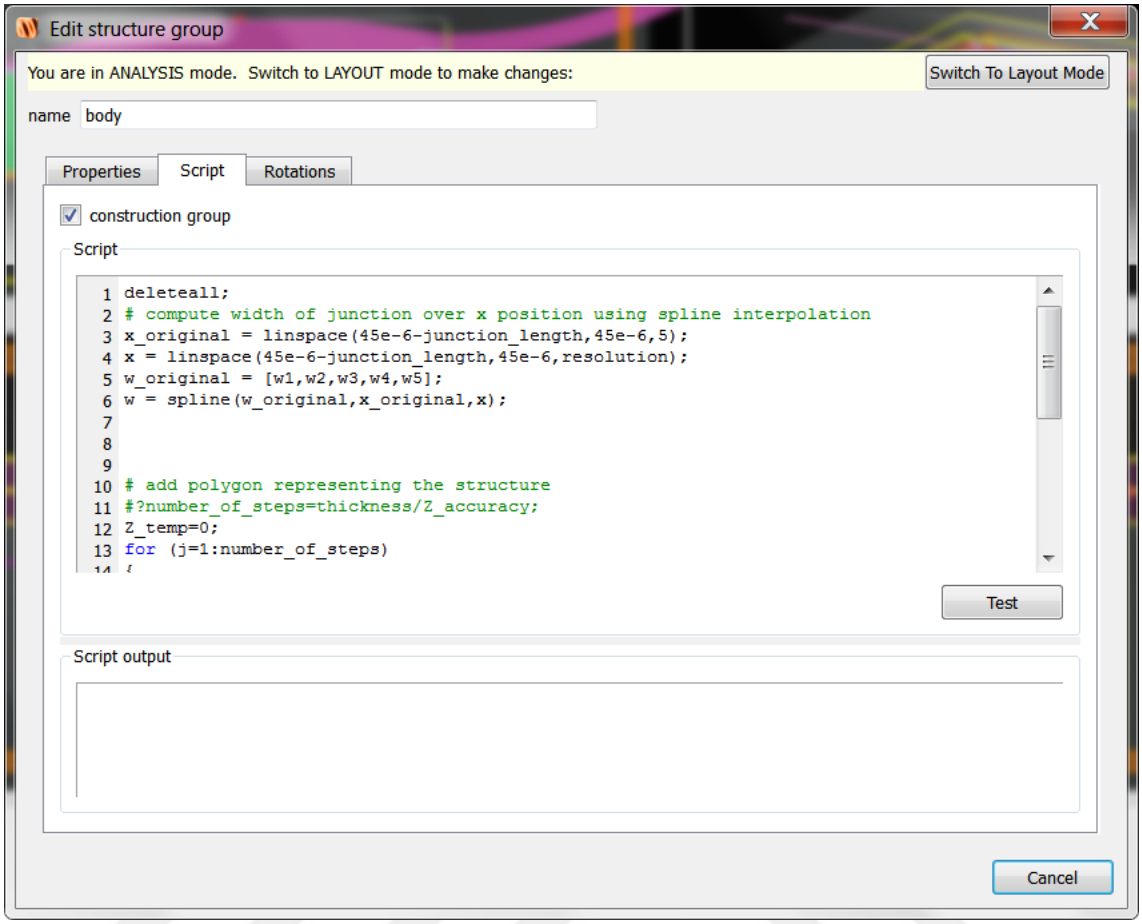


Figure 5.4. *The script tab in the edit structure group window*

The particular purpose of this code (can be found in appendix B) is to interpolate the polynomials *W1* to *W5* along the body length to create the base slice of the Y-branch's main body then the polynomials are reduced by subtracting *offset* value from each one of them. The reduced polynomials are then interpolated to create the next slice in the body and so on. The process is repeated 25 times (can be set by changing the *number_of _steps* property). Both *number_of_steps* and *offset* properties have values such that the resulting body has a 50 degree angled sidewall. Furthermore, a measure was undertaken to keep the angle of the sidewalls constant at 50 degrees regardless of the *thickness* property value. Figure 5.5 shows the resulting body emphasizing on the sidewalls staircase-like profile.

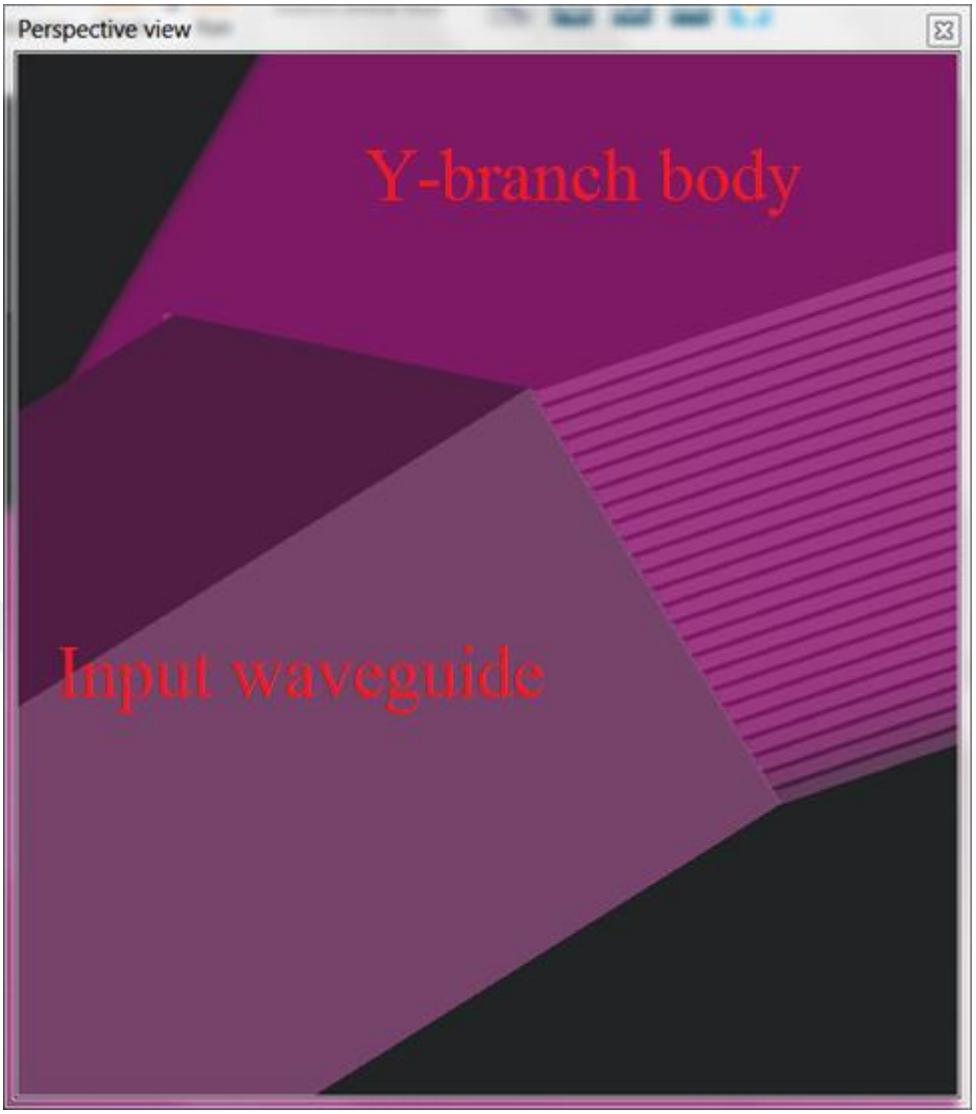


Figure 5.5. *A close view of the input waveguide and the main body of the Y-branch. The input waveguide has a flat sidewall surface while the body has a staircase-like surface profile. Both sidewalls are angled at 50 degrees*

5.3.1.4. Object Tree; gap

The "gap" structure group resembles the critical area between the body and the output of the Y-branch and it is a novel aspect of the proposed design. Even though it is a structure group, it does the exact opposite; it is designed to remove matter in the volume enclosed within it. In Lumerical, this is done by assigning "etch" material to it, which has a refractive index of one and has the higher priority over other materials when overlapping. The properties tab which is similar to that of the body is shown in figure 5.6

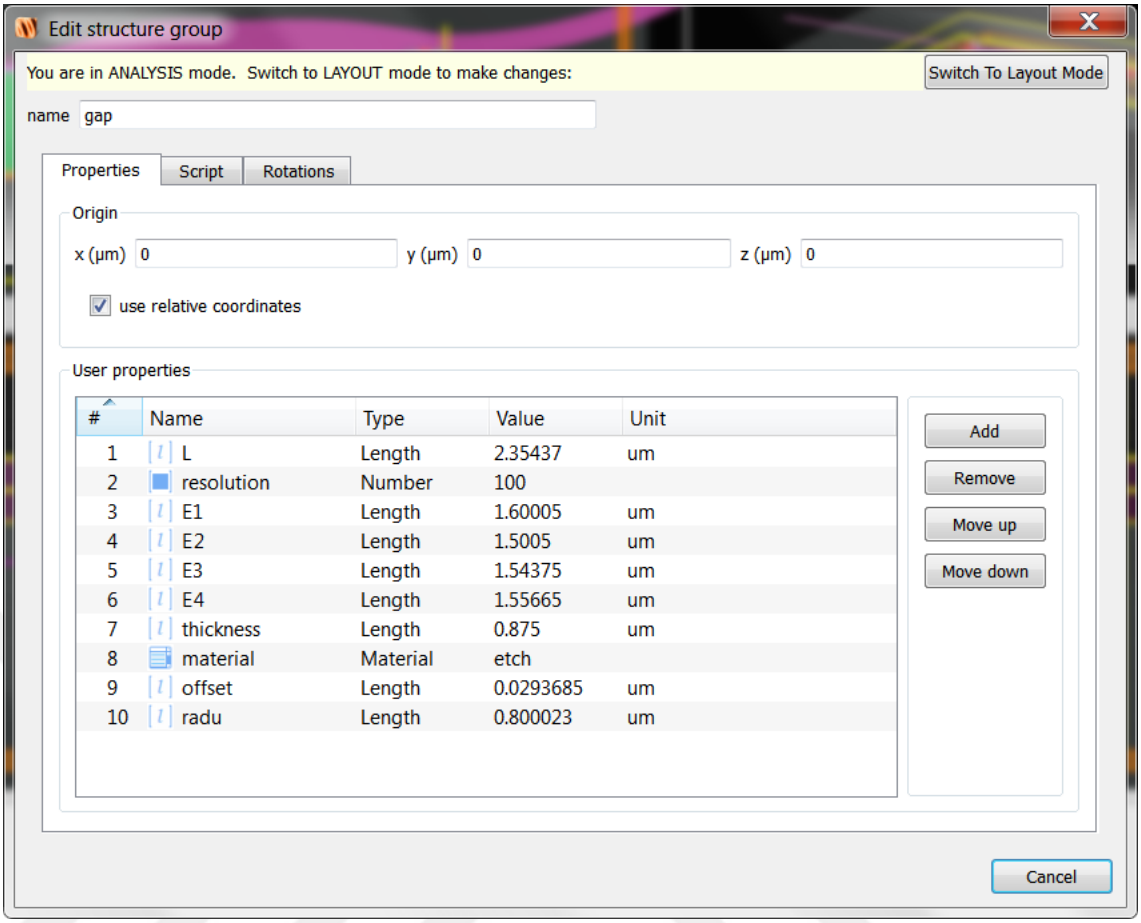


Figure 5.6. *The properties tab of the "gap" structure group*

Properties *E2*, *E3* and *thickness* are used in the optimization as-is. However, properties *E1* and *radu* are combined in the script to form a single parameter called "gap diameter", note that *radu* property refers to the radius of the circle that terminate the gap area and its value is always half that of *E1*. Also, *L*, is represents the "gap length". It is also worth mentioning that since this structure group subtracts materials from the "body", its sidewalls should have a reversed slope, as a result the base slice is the smallest and it gets larger and larger as more slices are added. The complete script code in this structure group can be found in appendix C.

5.3.1.5. Object Tree; outputs

The "outputs" object is the last structure group in the Object Tree. It consists of 25 concentric ring slices which are combined to form the upper output waveguide of the Y-branch. The resulting structure is a waveguide which has 50µm radius and the same cross-section as in Figure 4.5. As shown in figure 5.7, the *output_center* property is

combined with *W5* and *E4* to form a single parameter called "gap opening distance". However the script code responsible for that is written in the model script tab.

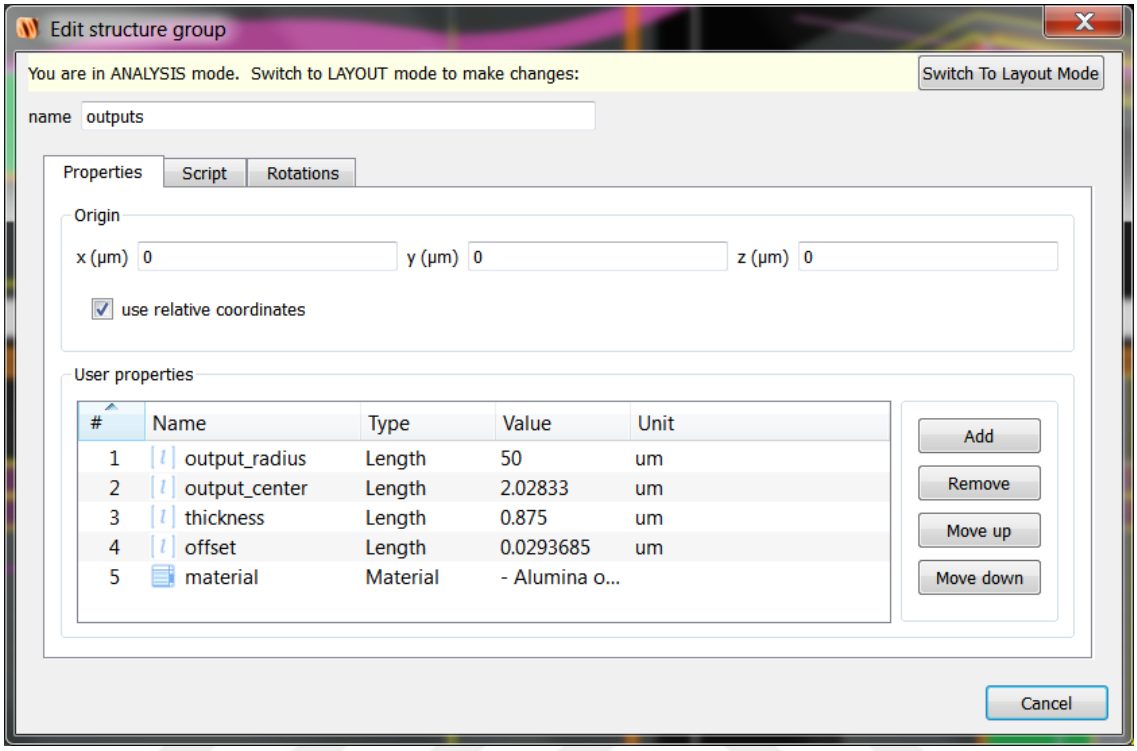


Figure 5.7. *The properties tab of the "outputs" structure group*

It is worth mentioning that there is no need to include another structure group for the lower output waveguide since the symmetry of the model in the propagation axis was exploited. The script code in this structure can be found in appendix D.

5.3.1.6. Object Tree; FDTD

This object represents the area (or volume) in which the simulation is going to be performed. Setting its options correctly is extremely important for a successful simulation. It is shown in figure 5.8 as an orange box.

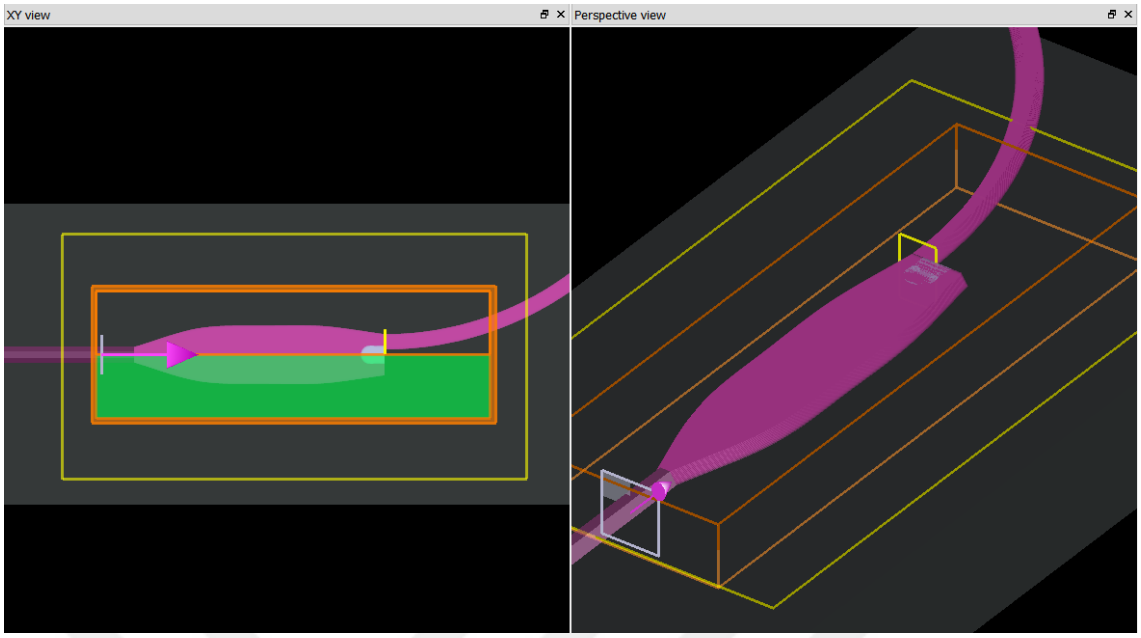


Figure 5.8. *A top view and a perspective view of the Y-branch in Lumerical's interface. Al_2O_3 which forms the input waveguide, the body and the output of the Y-branch is the pink colored object, the grey rectangle is the SiO_2 substrate. The simulation is performed only inside the orange box (which is defined from FDTD object). The yellow squares are electric and magnetic field monitors. The white area represents the gap. The green area is not simulated (structure's symmetry was exploited to reduce simulation time)*

When opening the options of the FDTD object, the first tab that appears controls the simulation type (1, 2 or 3 dimensional), simulation time in femtosecond, background index and simulation temperature. During the optimization process, the simulation type is set to 2D. A 3D simulation is used to double check the obtained results. The simulation time should be set to be "large enough". Increasing this parameter is important when simulating light trapping structures like ring resonators for example. However, in this particular case it is fine to be as low as 2000 femtosecond. Over estimating this value does not affect the simulation fidelity at all and it does not increase the required simulation time. The background index is set to 1 (to resemble air) and the simulation temperature is set to 300k (room temperature). The second tab controls the geometry of the simulation area which is very important. The boundaries should be far enough from any feature (except from the input and output waveguides) several tests were taken to ensure the fields' values are negligible near the boundaries. If the boundaries are close enough to interfere with the fields then the simulation results are invalid. On the other hand, if the boundaries are placed further than necessary then it would be a waste of time and

computational resources since the fields would be calculated in unimportant areas. Over estimating the boundaries distance is mostly fine for 2D simulations but very problematic for 3D simulations (the required simulation time increases from a couple of hours to days). Since the parameters of the Y-branch are to be changed automatically by the PSO algorithm, it is possible to place the boundaries based on the highest allowed parameter values but this would be a waste of time when the optimization algorithm chooses smaller designs. To overcome this issue, a part of the script code in the "model" was dedicated to assign the boundary's optimal distance for each set of the parameters automatically. The corresponding code can be found in appendix E.

On the third tab, time step and mesh settings for the Yee cell are present (see figure 5.9). The mesh type can be set to be uniform where the user manually enters the step size on all axes but this can be a waste of resources since it has the same resolution everywhere. The next choice is auto non-uniform mesh type which solves that problem by reducing the mesh size (and therefore increasing the resolution) near light guiding areas while increasing it elsewhere. The result is either a more accurate result at a comparable simulation time or a comparable accuracy at lower simulation time. Auto non-uniform mesh with the highest possible mesh accuracy of 8 (out of 8) is selected for all of the simulations reported throughout this thesis. More information about mesh and accuracy can be found in section 2.1.1.

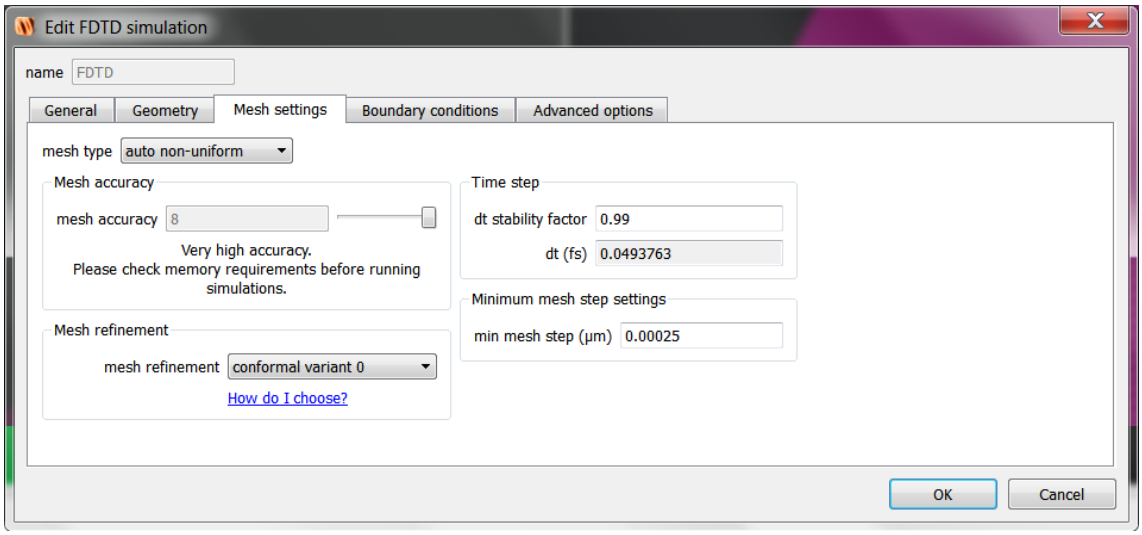


Figure 5.9. *Spatial and temporal mesh settings in Lumerical*

In mesh refinement option, conformal variant 0 is chosen. Conformal mesh methods try to account for sub-cell features by solving Maxwell's integral equations near structure

boundaries resulting in faster simulations without compromising accuracy more information about this topic can be found on Lumerical website. Time step setting was not changed from the default value. However, as a precautionary measure the final simulation was also performed with reduced time step but the results matched that of the default value.

The fourth tab is concerned with boundary conditions and symmetry. Figure 5.10 shows this tab and the options that can be changed using the user interface. As mentioned before the boundaries' positions are very important to be set up properly. Moreover, setting up the boundary condition type is equally as important. Two types are considered for the Y-branch understudy. The first one is metal boundary conditions which allows the simulation to be faster at the cost of accuracy. However, the most suited boundary condition for the Y-branch is the Perfectly Matched Layer (PML) which ideally absorbs all of the incident optical power on it. It is worth mentioning that in reality there are some inevitable reflections from PML boundary conditions. The absorption performance of the PML can be enhanced by increasing the number of the layers leading to more accurate but slower simulations. There is a need to increase the number of PML layers especially for boundaries that have features passing through them. For the Y-branch reported in this thesis, the light is propagating in the positive x direction and the output waveguides cut through the boundary on the maximum x axis. Therefore, it is sensible to place 32 layers of PML at the maximum x axis boundary. It also makes sense to increase the number of the PML layers for the minimum x axis boundary condition since the Y-branch may generate a considerable amount of back-reflection.

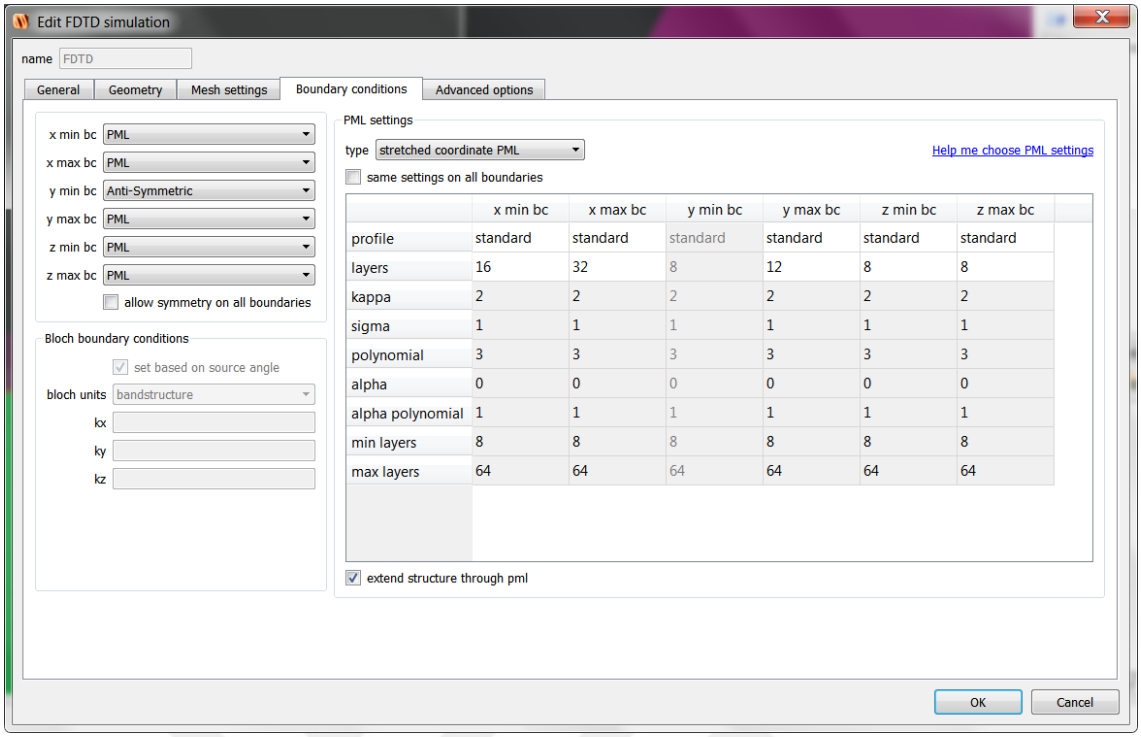


Figure 5.10. *Boundary conditions and PML settings tab*

As an excellent time saving measure, the symmetry of the Y-branch in its propagation axis allows the designer to exploit it to reduce the required simulation time. By assigning the minimum boundary condition to be anti-symmetric, only one half of the simulation area will be simulated while the other is deduced from the simulated half. Refer to figure 5.8, the green area is not simulated. Furthermore, due to symmetry, only one output waveguide is created since there is no need to create the other.

5.4. The Figure of Merit Function

The goal of any optimization is to either minimize or maximize a function whose value is dependent on the parameters of the design. This function is called Figure of Merit (FOM). FOM can be the amount of power reflected from a surface, the amount of power coupled into a certain mode or the amount of optical power absorbed within a certain volume…etc.

For symmetric single-mode Y-branches, FOM should be the amount of power coupled from the input waveguide into the fundamental mode of each of the output waveguides and it should be maximized [44–51]. Conversely, FOM can be chosen to be the insertion loss of the Y-branch across the wavelength band of interest [19], therefore it is to be minimized. To implement this, first, the power in the TE_0 of one output

waveguide is calculated by an overlap integral between TE_0 mode of the output waveguide and the recorded field at the output [53] as in equation (5.1).

$$\text{Power in } \mathrm{TE}_{0\ (\text{at one output})} = \int \mathrm{TE}^{*}_{0\ (\text{calculted mode})} \cdot \mathrm{E}_{(recorded\ field)} \cdot ds \tag{5.1}$$

Then the insertion loss (in dB) is calculated by equation (5.2)

$$\text{Insertion loss} = -10.\log_{10}(2.\text{mean}(\text{power in } \mathrm{TE}_{0(\text{at the output})})) \tag{5.2}$$

The mean function is used since power in TE_0 mode is recorded for the entire wavelength band (from 1450 to 1580 nm). However, if the designer wants to emphasize on a particular wavelength (1550 nm for example), it is possible to include that wavelength only. The reason for multiplying by 2 is due to the fact that only one output waveguide is considered in equation (5.1). Therefore, in order to account for power at both output waveguides we simply multiply by 2 since the Y-branch is symmetric. It is worth mentioning that some power will inevitably couple into higher order modes of the output waveguides which is considered as power loss. Therefore it is inappropriate to set FOM to be the total power recorded at the output since it is not a meaningful performance metric [53].

5.4.1. Figure of Merit implementation in Lumerical

In order to implement the FOM described in the previous section in Lumerical, monitors and code scripts were used. Monitors are objects that record data related to the simulation within a predefined volume, area or along a line. There are many monitor types in Lumerical depending on the data they collect. Frequency-domain field and power monitors are used to collect the electric field (E), magnetic field (H), Poynting vector (P) and power transmission (T) as functions of position and wavelength. Two monitors of this type were employed as seen in the Object Tree (figure 5.2). The first one is named "field" and it was used to record the E field across a horizontal plane that cuts the Y-branch in mid-height which is useful for demonstrating how E field is distributed within the Y-branch. It provides a rough estimation of the design's performance. For example, if it shows standing waves in the Y-branch then it is determined that there is a non-negligible portion of the input power is being reflected back to the input side. Improper propagation and wobbling of the fields are also signs of poor performing design. The

recorded E field using this monitor is shown in figure 5.13 and in figure 5.17. The second monitor is called “output1” and it is a 2D monitor that is placed perpendicular to the direction of propagation, just outside the body of the Y-branch and at the beginning of the upper output waveguide. This monitor has the ability to find the total power coupled into the output waveguide (as previously mentioned, total power coupled into the output is not a valid performance metric).

A Mode Expansion Monitor is used for analyzing the fraction of power transmitted into any mode(s) of a non-absorbing waveguide [19]. In the Object Tree it is named “Exp” and it is used in combination with “output1” to calculate the amount of power coupled from the body of the Y-branch into the fundamental mode of upper output waveguide. In other words, this monitor calculates the overlap integral as expressed in equation 5.1. Setting this monitor up requires assigning a mode and a Frequency-domain field and power monitor from the mode expansion tab. In our particular case, the fundamental TE mode and the “output1” monitor was chosen. It is important to note that the mode is calculated in the area of the “Exp” monitor and it should be placed on the same cross section that has “output1” monitor. Furthermore, the output waveguides are bent therefore the corresponding options should be assigned correctly as in Figure 5.11.

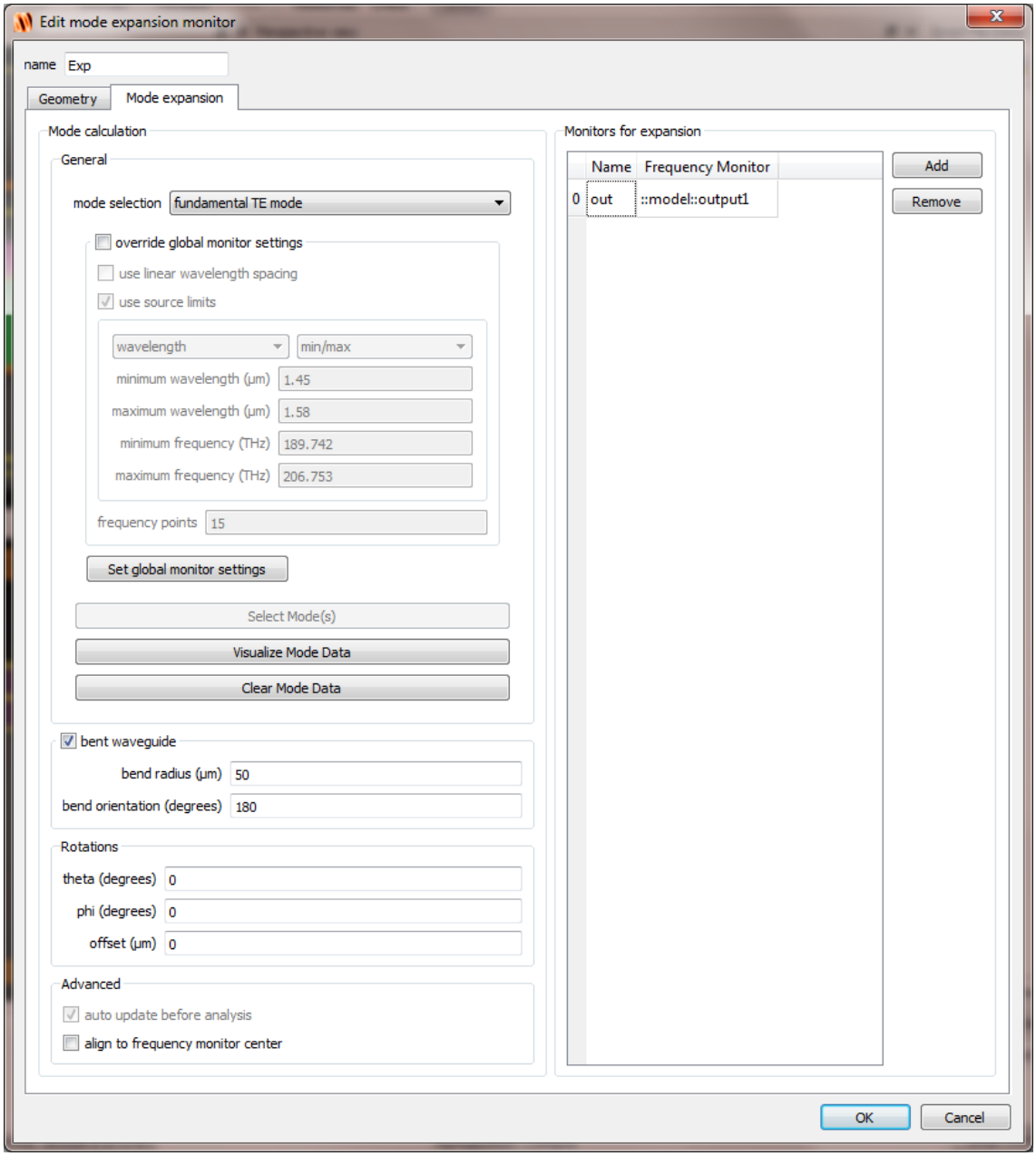


Figure 5.11. *The expansion monitor settings tab*

After the simulation is done, the results of the Mode Expansion Monitor become available. The value T_net represents the answer of equation 5.1 and it is a real value between 0 and 1 (due to normalization to the source power).

The script code written in "model" can control many aspects of the simulation. For this particular simulation, a code was written in the script tab of the "Analysis" tab in "model" object (it is reported in appendix E). It calculates the mean value of the total insertion loss as expressed in equation 5.2 by accessing the results of the Mode Expansion Monitor "Exp" and obtaining T_net then performing the subsequent arithmetic operations

on it. Moreover, in our implementation of the code, a variable named “IL” is added in the results section of the “Variables” tab in the “model” object. “IL” holds the answer to equation 5.2.

5.5. Launching the Y-branch optimization in Lumerical

In order to achieve acceptable results in a timely manner, the simulation is set to the less accurate 2D FDTD which takes around a minute per simulation (much less than a full 3D FDTD simulation which takes around 10 minutes per simulation). The mesh is set to auto non-uniform with mesh accuracy of 4 out of 8. A perfectly matched layer is set to be the simulation boundary on all sides. Since the Y-branch is symmetric around the propagation axis, it is possible to simulate only on half of the simulation region and deduce the other one which is a very effective measure to reduce the resources needed for the optimization. 50 generations with 30 simulations per generation (1500 simulation in total) were performed.

It is important to point out that not all parameters are used in the optimization. W1, for example, is excluded since its value should be 2.5 μm (constant). W5 and E4 are combined using Lumerical's scripting language to form a single parameter which we called gap opening distance. The resulting new parameter is, in fact, the minimum distance between the output waveguides. Changing this parameter will change W5 and E4 in a way that maintains a constant width of 2.5 μm for the output waveguides. Finally, E1 and D are forced to be equal to ensure a smooth and continuous gap. They are also replaced by a single parameter we called gap diameter. The complete list of included parameters in the optimization can be found in Table 5.1. The thickness range is chosen such that single-mode operation is not violated in the wavelength range under study.

Table 5.1. *The complete list of the included parameters in the optimization and their initial values and allowed ranges*

Parameter	Minimum value allowed	Maximum value allowed	Initial value
W2	4 μm	15 μm	8 μm
W3	2 μm	10 μm	4 μm
W4	4 μm	15 μm	8 μm
Body length	17 μm	70 μm	50 μm
E2	1.5 μm	3 μm	1.5 μm
E3	1.5 μm	3 μm	1.5 μm
Gap length	0 μm	10 μm	5 μm
Gap opening distance	1.5 μm	3 μm	3 μm
Gap diameter	1.5 μm	3 μm	1.5 μm
Thickness	0.8 μm	1.1 μm	1 μm

5.6. Results and Discussion

The obtained total insertion loss was found to be 0.083 dB at the end of the optimization. The optimal parameters and their values are shown in Table 2. Figure 5.12 shows the FOM (insertion loss in this case) as a function of generation number. A 3D FDTD simulation with the highest mesh accuracy of 8 was performed to double check the results. Surprisingly, the total insertion loss was in fact 0.45 dB instead of 0.083 dB. The reason for this difference in the obtained results is that the 2D FDTD doesn't take into account the angled sidewalls correctly.

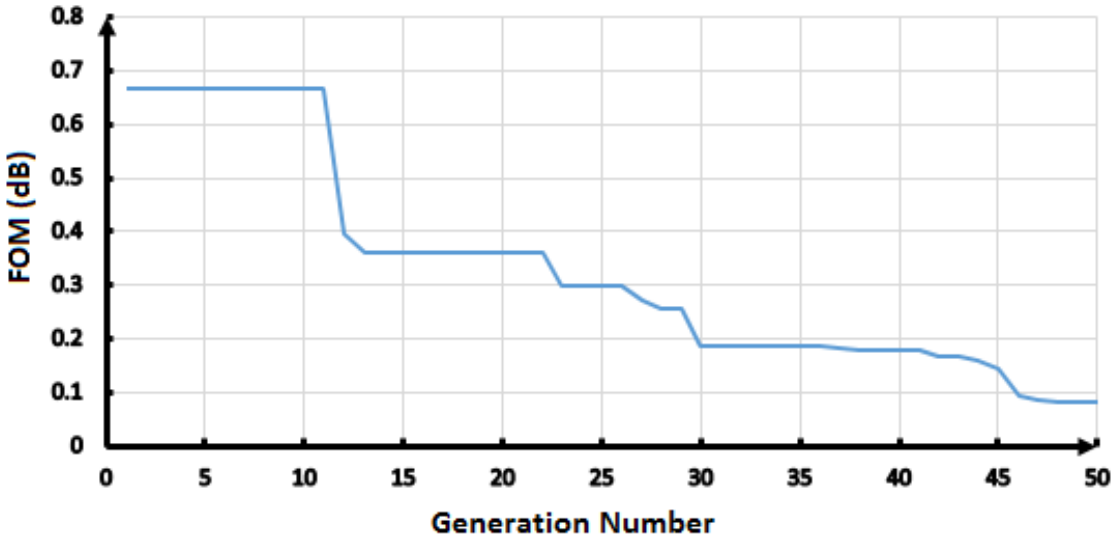


Figure 5.12. *The insertion loss in dB as a function of generation number*

A top view of the electrical field distribution in the Y-branch shows a very smooth propagation and transition towards the output. The absence of standing waves indicates that the reflected power to the input side is marginally small as illustrated in Figure 5.13. A 3D view of the optimized Y-branch is shown in Figure 5.14.

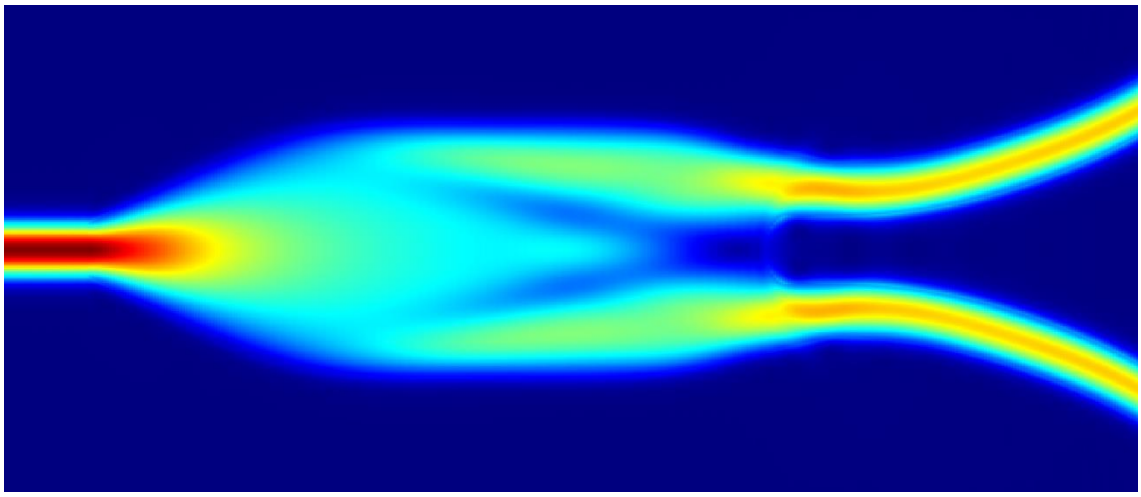


Figure 5.13. *A contour plot of the electrical field distribution inside the optimized Y-branch. It shows a smooth propagation of optical power towards the output waveguides. The plot is recorded at half thickness*

Table 5.2. *Parameters' optimal values*

Parameter	Optimum value
W2	8.30 µm
W3	9.24 µm
W4	9.12 µm
Body length	39.56 µm
E2	1.50 µm
E3	1.54 µm
Gap length	2.35 µm
Gap opening distance	1.56 µm
Gap diameter	1.60 µm
Thickness	0.80 µm

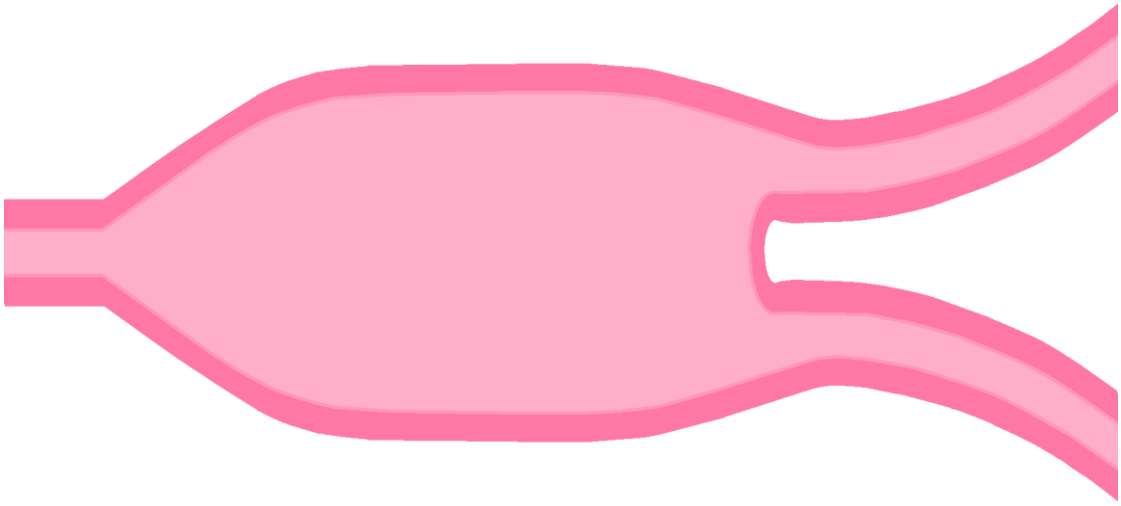


Figure 5.14. *Shows a 3D view of the Y-branch with the output waveguides. Dark areas are sloped*

The total transmission has a mean value of around 90% and it changes within less than 2% across the wavelength range of interest indicating that the optimized Y-branch is wavelength insensitive in that range as Figure 5.15 shows. It is worth mentioning that the highest transmission of 90.8% occurs at wavelengths around 1.49 µm. Conversely,

from a total insertion loss perspective, the optimized design has a mean value of 0.45 dB which acceptable given the enforcement of a minimum feature size of 1.5 μm and use of a weakly guiding material system. Total insertion loss is shown in Figure 5.16. If the optimized Y-branch is to be doped with a rare earth element like Erbium whose pumping wavelength band ranges from 1480 to 1500 nm with an emission wavelength band ranging from 1520 to 1570 nm, a compact lossless devices can be fabricated without effecting the single-mode characteristics.

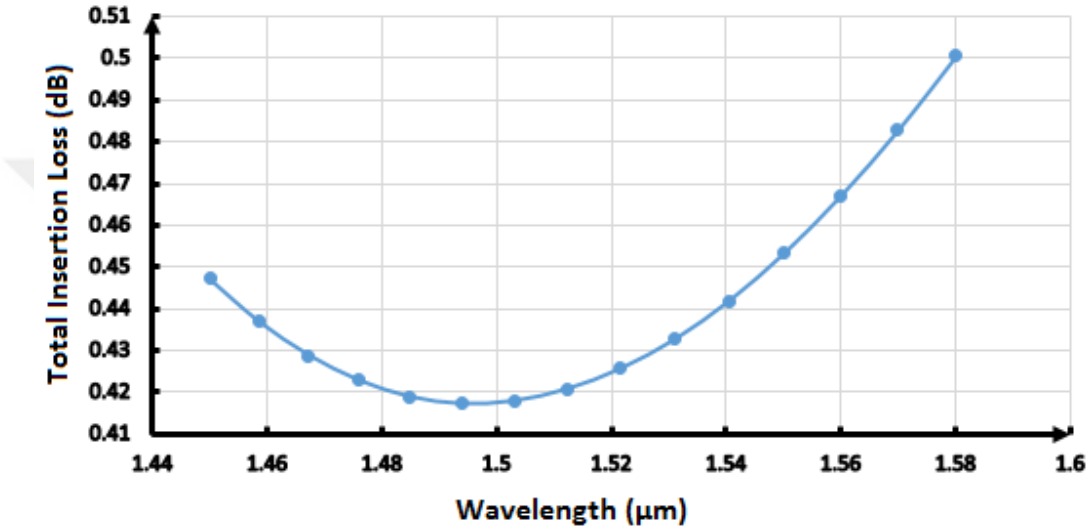


Figure 5.15. *The normalized total transmission of power across wavelengths ranging from 1450 to 1580 nm. The mean total transmission is 90%*

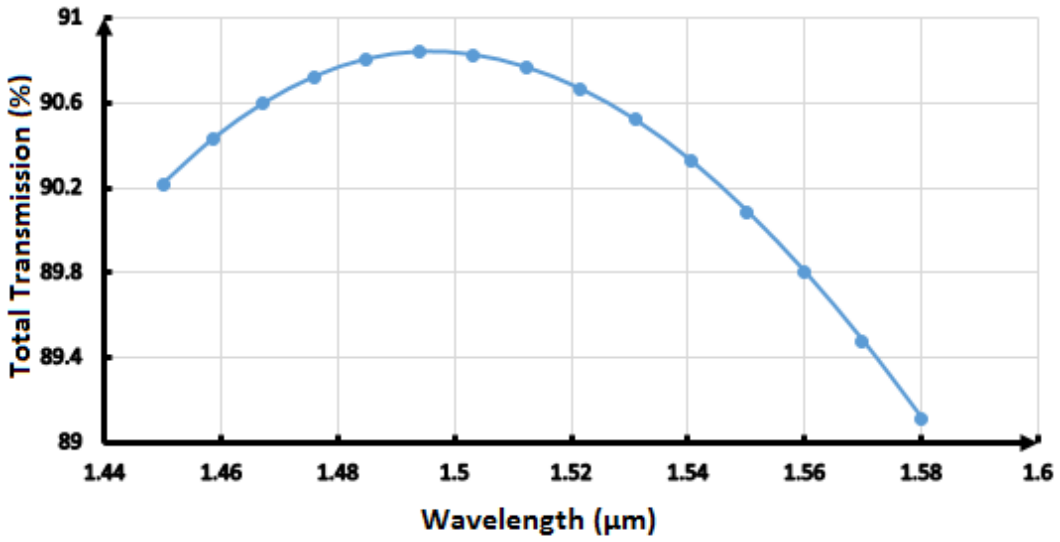


Figure 5.16. *The total insertion loss across wavelengths ranging from 1450 to 1580 nm. The mean value is 0.45 dB*

5.7. Simulation Results of Non-Realistic Y-branch Geometry

In order to investigate the importance of simulating a realistic structure with sloped sidewalls, a 3D FDTD simulation with the highest mesh accuracy of 8 was performed on the optimized Y-branch only with ideal vertical sidewalls (instead of 50° sloped sidewalls). The mean total insertion loss was found to be 1.98 dB which is more than 4 times of that of the realistic case. In other words if the optimization was carried out using structures with vertical sidewalls, then the resulting optimized design would be useless in the real world since real structures have sloped sidewalls.

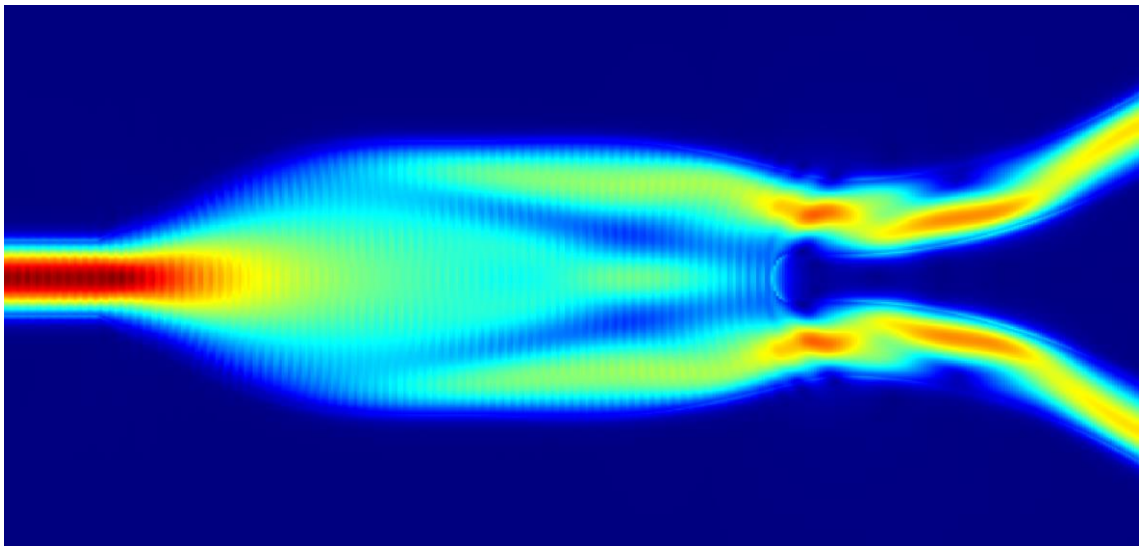


Figure 5.17. *A contour plot (top view) of the electric field distribution inside the optimized Y-branch but has ideal vertical sidewalls instead of the realistic sloped ones. Rough transition towards the output and back reflections are present. The plot is recorded at half thickness*

Figure 5.17 shows how the propagation within the Y-branch suffers from rough transition from the body of the Y-branch to the output waveguides in addition to back reflections indicated by the presence of standing waves within the body and input sections.

5.8 Did PSO converge to The Absolute Optimal Design?

Particle swarm algorithm is considered a heuristic optimization method, which means that it needs relatively large number of simulations to converge to an acceptable solution. However there is no guarantee that the solution found is in fact the optimal solution for the given parameters. Increasing the number of simulations should be considered in order to overcome this issue since performing more simulations dictates that the parameter space is better explored. To the author's surprise, it was found by

coincidence that the optimal design reported in section 5.6 is not the absolute optimal design. This discovery was made while performing a simple parameter sweep of the *thickness* parameter which seemed to be influential over the Y-branch as a whole. The total insertion loss as a function of thickness is plotted in figure 5.18.

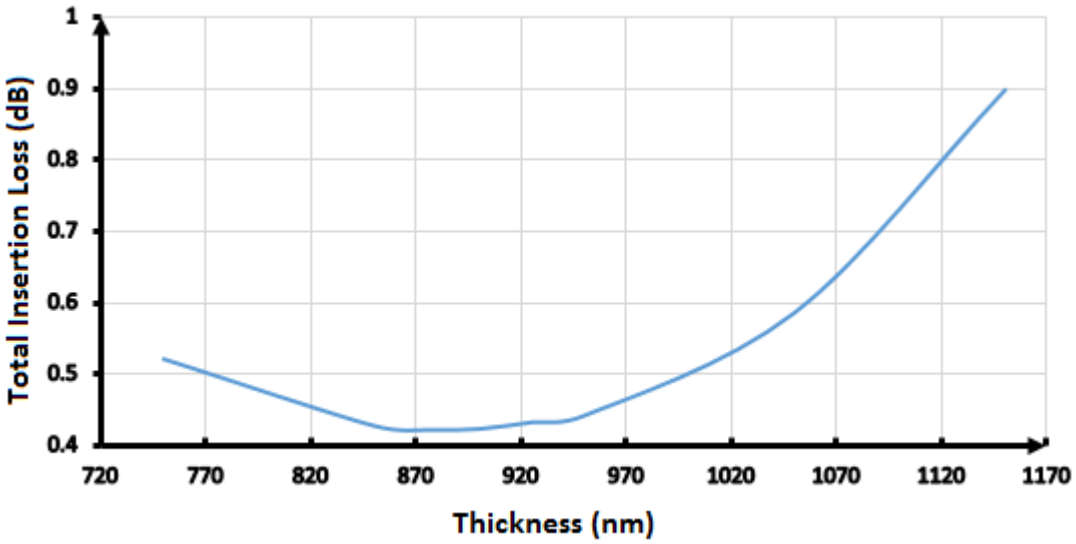


Figure 5.18. *Total insertion loss as a function of thickness. It shows a minimal insertion loss of 0.43dB at thicknesses around 0.87μm*

Although the difference is only 0.02 dB between the optimal design found by the optimization algorithm and the one found after the parameter sweep, is it nonetheless a difference. This discovery further emphasizes the nature of the PSO as a stochastic optimization algorithm.

It is worth noting that the optimization was found to be somewhat sensitive to the initial parameters set by the user where different initial parameter combinations cause the algorithm to converge to different designs with different performance levels. Therefore it is imperative that the designer tries different combinations of the parameters.

Another aspect of using this algorithm is that adding more parameters does not necessarily improve the results. Up to 10 polynomials for defining the body of the Y-branch were tested. However with trial and error, it was found that 5 polynomials are enough for acceptable results. Moreover, with the addition of more parameters, the optimization algorithm would need more simulations to converge to a satisfactory

solution. Therefore a balance between the number of parameters and the level of performance needed must be struck.

5.9. Conclusion

We created a single-mode waveguide geometry suitable for Al_2O_3 which is a very promising material in the field of integrated optics, then we proposed a novel Y-branch parameterization based on the work of Zhang et al. [53] while emphasizing on fabricability within a 1 μm lithography process. Particle swarm algorithm combined with a 2D FDTD solver were employed to find the optimal parameters. The proposed optimal design is wavelength insensitive and has a 0.45 dB of mean total insertion loss between wavelengths ranging from 1450 to 1580 nm. We also found that for a ridge waveguide-based devices, it is imperative for the designer to include the angle of the sidewalls in the optimization. Otherwise the obtained results may not reflect reality. PSO stochastic nature was also investigated.

Our work serves as either a guideline for using optimization algorithms in photonic design or a case study of the material system and/or the algorithm used for creating the Y-branch.

REFERENCES

[1] W. Bogaerts and L. Chrostowski (2018). Silicon photonics circuit design: methods, tools and challenges. *Laser & Photonics Reviews,* 12 (4), 1-29.

[2] L. Chrostowski and M. Hochberg (2015). Introduction. In Silicon Photonics Design: From Devices to Systems. *Cambridge University Press,* 1, 2, 5, 7, 8, 71, 72.

[3] J. Hu, X. Sun, A. Agarwal, and L. Kimerling (2009). Design guidelines for optical resonator biochemical sensors. *Journal Optics Society America B,* 26, 1032–1041.

[4] M. Iqbal, M. Gleeson, B. Spaugh, et al (2010). Label-free biosensor arrays based on silicon ring resonators and high-speed optical scanning instrumentation. *IEEE Journal of Selected Topics in Quantum Electronics,* 16 (3), 654–661.

[5] R. B. Wehrspohn, S. L. Schweizer, T. Geppert, et al. (2006). Chapter 12. Application of Photonic Crystals for Gas Detection and Sensing. Advances in Design, Fabrication, and Characterization, K. Busch, S. Lalkes, R. B. Wehrspohn, and H. Fall (eds.), *Photonic Crystals: Wiley-VCH Verlag GmbH.*

[6] R. Camacho-Aguilera, Y. Cai, N. Patel, et al (2012). An electrically pumped germanium laser. *Optics Express*, 20, 316–320.

[7] H. Y. Liu, T. Wang, Q. Jiang, et al (2011). Long-wavelength InAs/GaAs quantum-dot laser diode monolithically grown on Ge substrate. *Nature Photonics,* 5 (7), 416–419.

[8] A. A. Trusov, I. P. Prikhodko, S. A. Zotov, A. R. Schofield, and A. M. Shkel (2010). Ultra-high Q silicon gyroscopes with interchangeable rate and whole angle modes of operation. *Proc. IEEE Sensors*, 864–867.

[9] M. Guilln-Torres, E. Cretu, N.A.F. Jaeger, and L. Chrostowski (2012). Ring resonator optical gyroscopes – parameter optimization and robustness analysis. *Journal of Lightwave Technology*, 30 (12), 1802–1817.

[10] M A. Turner, M. Lipson, and A. Gaeta (2008). Nonlinear optics in photonic nanowires. *Optics Express*, 16 (2), 1300–1320.

[11] J. K. Doylend, M. J. R. Heck, J. T. Bovington, et al (2011). Two-dimensional free-space beam steering with an optical phased array on silicon-on-insulator. *Optics Express,* 19 (22), 21 595–21 604.

[12] J. Capmany and D. Novak (2007). Microwave photonics combines two words. *Nature Photonics*, 1 (6), 319–330.

[13] Maurizio Burla, Luis Romero Cortés, Ming Li, et al (2013). Integrated waveguide Bragg gratings for microwave photonics signal processing. *Optics Express*, 21, 25120– 25147.

[14] A. Shacham, K. Bergman, and L. P. Carloni (2008). Photonic networks-on-chip for future generations of chip multiprocessors. *IEEE Transactions on Computers*, 57 (9), 1246–1260.

[15] A. F. Peterson, S. L. Ray, and R. Mittra (1998). Computational Methods for Electromagnetics. *New York: IEEE Press*.

[16] J. M. Jin (2002). The Finite Element Method in Electromagnetics, *2nd edition. New York: John Wiley & Sons, Inc*.

[17] K. S. Yee (1966). Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 302–307.

[18] RSoft Products – Synopsys Optical Solutions. url: (last accessed on 03/05/2019) http://optics.synopsys.com/rsoft/

[19] Solutions, F. D. T. D., 2018, Lumerical Solutions Inc. url: (last accessed on 03/05/2019) https://www.lumerical.com/

[20] S. D. Gedney (2011). Introduction to the Finite-Difference Time-Domain (FDTD) Method for Electromagnetics. *Morgan & Claypool*.

[21] J. Bergstra, Y. Bengio (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13, 281-305.

[22] S. Behera, S. Sahoo, B.B.Pati (2015). A review on optimization algorithms and application to wind energy integration to grid. *Renewable and Sustainable Energy Reviews*, 48, 214–227.

[23] Kirkpatrick S, Gelatt CD, Vecchi MP (1983). Optimization by simulated annealing. *Sci. J.*, 220 (4598), 671–680.

[24] A. Tiwari, V. Ajjarapu (2008). Optimal allocation of dynamic VAR for enhancing stability and power quality. *Proceedings of 2008 IEEE power and energy society general meeting—conversion and delivery of electrical energy in the 21st century*, 1–7.

[25] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi (2009). GSA a gravitational search algorithm. *Inf. Sci.*, 179, 2232-2248.

[26] X.F. Yan, D.Z. Chen, S.X. Hu (2003). Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. *Comput. Chem. Eng*., 27 (10), 1393-1404.

[27] F. Glover (1977). Heuristics for integer programming using surrogate con-straints. *Decis. Sci.,* 8, 156-166.

[28] EA. Gargari, C. Lucas (2007). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *Proceedings of IEEE congress on evolutionary computation*, 4661.

[29] M.S. Saad, H. Jamaluddin, I.Z. Darus (2012). PID controller tuning using evolutionary algorithms. *WSEAS Trans. Syst. Control*, 7 (4), 139-149.

[30] M. Vanitha, K. Thanushkodi (2011). Solution to economic dispatch problem by differential evolution algorithm considering linear equality and inequality constraints. *Int J. Res. Rev. Electr. Comput. Eng*, 1 (1), 21-26.

[31] M. Dorigo (1992). Optimization, learning and natural algorithms. *[Ph.D. thesis]*, *Politecnico di Milano,Milano*, Italy.

[32] S.Muller, S. Airaghi, J. Marchetto, and P. Koumoutsakos (2000). Optimization algorithms based on amodel of bacterial chemotaxis. *Proceedings of the 6th International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, 375‑384.

[33] K.M. Passino (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22 (3), 52‑67.

[34] D. Karaboga and B. Basturk (2007). Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems. *Proceedings of the Foundations of Fuzzy Logic and Soft Computing*, 789–798.

[35] J. Kennedy and R. Eberhart (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4.

[36] Y. Zhang, S. Wang, and G. Ji (2015). A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Mathematical Problems in Engineering*, 2015.

[37] R. Eberhart and J. Kennedy (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 39-43.

[38] S. Kumar, S. and P. Sethi (2018). Review on Optical Waveguides. *Emerging Waveguide Technology*, Kok Yeow You, IntechOpen.

[39] M. Demirtaş et al (2016). Extensive mode mapping and novel polarization filter design for ALD grown Al_2O_3 ridge waveguides. *Opt. Quant. Electron.*, 48, 357.

[40] M. Demirtaş et al (2018). Low Loss Atomic Layer Deposited Al_2O_3 Waveguides for Applications in On-Chip Optical Amplifiers. *IEEE Journal of Selected Topics in Quantum Electronics*, 24 (4).

[41] J. Bradley (2009). Al_2O_3:Er^{+3} as a gain platform for integrated optics. *[Ph.D Thesis]*, Enschede, The Netherlands.

[42] S. Shi, S. Qian, X. Hou, J. Mu, J. He, and X. Chou (2018). Structural and Optical Properties of Amorphous Al_2O_3 Thin Film Deposited by Atomic Layer Deposition. *Advances in Condensed Matter Physics*, 2018, Article ID 7598978.

[43] A. Sakai, T. Fukazawa and T. Baba (2002). low loss ultra-small branches in a silicon photonic wire waveguide. *IEICE Transactions on Electronics*, E85-C (4), 1033–1038.

[44] M. H. Hu, J. Z. Huang, R. Scarmozzino, M. Levy, and R. M. Osgood, Jr (1997). A Low-Loss and Compact Waveguide Y-Branch Using Refractive-Index Tapering. *IEEE Photonics Technology Letters*, 9 (2), 203–205.

[45] Y. Shibata, S. Oku, M. Ikeda, and Y. Kadota (1992). An Asymmetric Y-Branching Circuit Using Total Reflection for Semiconductor Integrated Optical Circuits. *IEEE Photonics Technology Letters*, 4 (11), 1253–1255.

[46] R. Lu, Y. Liao, and W. Wang (1998). Design of Symmetric Y-Branch with a Substrate Prism and Two Tapered Output Waveguides on $LiNbO_3$. *IEEE Photonics Technology Letters*, 10 (9), 1274–1276.

[47] H. Lin, R. Cheng, and W. Wang (1994). Wide- Angle Low-Loss Single-Mode Symmetric Y-Junctions. *IEEE Photonics Technology Letters*, 6 (7), 825–827.

[48] H.P. Chan, S.Y. Cheng and P.S. Chung (1996). Low loss wide-angle symmetric Y-branch waveguide. *IEEE Electronics Letters*, 32 (7), 652–654.

[49] Y. Sakamaki, T. Saida, M. Tamura, T. Hashimotoand, and H. Takahashi (2007). Low-loss Y-branch waveguides designed by wavefront matching method and their application to a compact 1x 32 splitter. *IEEE Electronics Letters*, 43 (4).

[50] Y. Sakamaki et al (2006). Y-Branch Waveguides With Stabilized Splitting Ratio Designed by Wavefront Matching Method. *IEEE Photonics Technology Letters*, 18 (7), 817–819.

[51] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch (2013). Adjoint shape optimization applied to electromagnetic design. *Optics Express*, 21 (18).

[52] L. Su, A. Y. Piggott, N. V. Sapra, J. Petykiewicz, and Jelena Vučković (2017). Inverse Design and Demonstration of a Compact on-Chip Narrowband Three-Channel Wavelength Demultiplexer. *ACS Photonics*, 5 (2), 301−305.

[53] Y. Zhang, S. Yang, A. E. Lim, G. Lo, C. Galland, T. Baehr-Jones, and M. Hochberg (2013). A compact and low loss Y-junction for submicron silicon waveguide. *Optics Express*, 21 (1).

APPENDIX A

Table of the ellipsometerically measured Al_2O_3 refractive index as a function of wavelength. The film was grown using ALD.

Wavelength (nm)	Re(n)	Im(n)
1450	1.64085	0
1460	1.6408	0
1480	1.6407	0
1500	1.6406	0
1520	1.6405	0
1540	1.6405	0
1560	1.6404	0
1580	1.6403	0

APPENDIX B

The script code in the “body” structure group. It is supposed to create 25 “slices” of the body of the Y-branch. The base slice is the largest one on top of it is a slightly smaller one followed by yet smaller one and so on. The perimeter shape is defined by interpolating W1, W2, … , W5.

```
deleteall;
x_original = linspace(45e-6-junction_length,45e-6,5);
x = linspace(45e-6-junction_length,45e-6,resolution);
w_original = [w1,w2,w3,w4,w5];
w = spline(w_original,x_original,x);
# add polygon representing the structure
#?number_of_steps=thickness/Z_accuracy;
Z_temp=0;
for (j=1:number_of_steps)
{
if(Z_temp==thickness){break;}
addpoly;
set("z min",Z_temp);
set("z max",Z_temp+(thickness/number_of_steps));
set("material",material);
x_positions = [x;flip(x,1)];
y_positions = [w/2;flip(-w/2,1)];
V=[x_positions,y_positions];
set("vertices",V);
set("color opacity",0.4);
Z_temp=Z_temp+(thickness/number_of_steps);
w = w - offset;
}
```

APPENDIX C

The script code in the “gap” structure group. It is supposed to create 25 “slices” of the gap of the Y-branch. The base slice is the smallest one on top of it is a slightly larger one followed by yet another larger one and so on. The perimeter shape is defined by interpolating E1, E2, E3 and E4.

```
deleteall;
junction_length=L;
meow=-junction_length+45e-6;
x_original = linspace(meow,45e-6,4);
x = linspace(meow,45e-6,resolution);
w_original = [E1,E2,E3,E4];
w = spline(w_original,x_original,x);
# add polygon representing the structure
#?number_of_steps=thickness/Z_accuracy;
Z_temp=0;
#radu=0.75e-6;
number_of_steps=25;
for (j=1:number_of_steps)
{
if(Z_temp==thickness){break;}
addpoly;
set("z min",Z_temp);
set("z max",Z_temp+(thickness/number_of_steps));
set("color opacity",0.4);
set("material",material);
x_positions = [x;flip(x,1)];
y_positions = [w/2;flip(-w/2,1)];
V=[x_positions,y_positions];
set("vertices",V);
w = w + offset*2;
addcircle;
```

```
set("z min",Z_temp);
set("z max",Z_temp+(thickness/number_of_steps));
set("material",material);
set("y",0);
set("x",45e-6-junction_length);
set("radius",radu);
radu=radu+offset;
Z_temp=Z_temp+(thickness/number_of_steps);
set("color opacity",0.4);
}
```

APPENDIX D

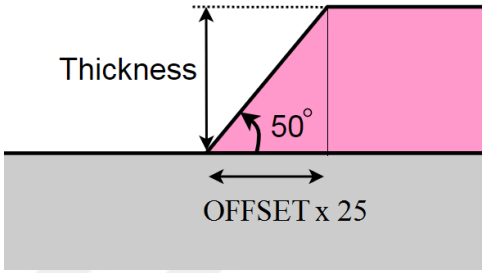
The script code in the “outputs” structure group. It is responsible of creating 25 concentric ring slices, which form the output upper waveguide. The slice in the base is the largest.

```
deleteall;
Z_temp = 0;
rate=offset;
number_of_steps=25;
inner=output_radius-1.25e-6;
outer=output_radius+1.25e-6;
for (j=1:number_of_steps)
{
addring;
set("name", "r"+num2str(j));
set("inner radius",inner);
inner=inner+rate;
set("outer radius",outer);
outer=outer-rate;
set("y",output_radius+output_center);
set("x", 45e-6);
set("z min",Z_temp);
set("z max",Z_temp+(thickness/number_of_steps));
Z_temp=Z_temp+(thickness/number_of_steps);
set("theta start", -90);
set("theta stop", -40);
set("material",material);
set("color opacity",0.4);
}
```

APPENDIX E

The script written in the script tab in the setup tab of the "model" object has the ability to change, modify and access the properties of all structure groups. Therefore, it controls the whole parameterization process. The first line of this code calculates the offset (for a single slice) needed to get 50 degree sloped sidewall regardless of the thickness. This is done by dividing the thickness over a fixed number which comes from the tangent equation as follows

$$25 \times \tan(50^{\circ}) = \frac{Thickness}{OFFSET} = 29.79383981$$




```
# The main code of the model
OFFSET= THICKNESS / (29.79383981);
gap_opening_distance=gap_opening_distance/2;
output_center=(gap_opening_distance+1.25e-6);
select("outputs");
set("thickness",THICKNESS);
set("offset",OFFSET);
set("output_center",output_center);
unselectall;
# creating the slpoed ring
select("output1");
set("y",output_center);
set("y span",output_center*2 - 2e-7);
unselectall;
select("Exp");
set("y",output_center);
set("y span",output_center*2 - 2e-7);
unselectall;
select("body");
set("thickness",THICKNESS);
set("offset",OFFSET*2);
set("w5",(output_center+1.25e-6)*2);
```

```
body_length = get("junction_length");
xstart= 45e-6 - body_length;
unselectall;
select("source");
set("x",xstart-5e-6);
unselectall;
select("FDTD");
set("x min",xstart-6e-6);
unselectall;
select("gap");
set("thickness",THICKNESS);
set("offset",OFFSET);
set("E4",(output_center-1.25e-6)*2);
set("E1",get("radu")*2);
if( gap_length > body_length -2e-6 )
{gap_length = body_length - 2e-6;
set("L",gap_length);
}
else
{set("L",gap_length);}
unselectall;
select("waveguide");
set("base height",THICKNESS);
set("z",THICKNESS/2);
unselectall;
select("field");
set("z",THICKNESS/2);
unselectall;
```

APPENDIX F

The script found here is written in the script tab in the “Analysis” tab of the “model” object. The second line calculates the total insertion loss in dB as in equation 5.2.

```
EXP=getresult("Exp","expansion for out");
IL=-10*log10(2*mean(EXP.T_net));
```