Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for elektroniske systemer

Petter Wiker

Optimization and Fabrication of Free-Standing Waveguides for Sensing Applications

Masteroppgave i MTNANO
Veileder: Astrid Aksnes
Medveileder: Jana Jágerská
Juli 2023

NTNU
Kunnskap for en bedre verden

Petter Wiker

Optimization and Fabrication of Free-Standing Waveguides for Sensing Applications

Masteroppgave i MTNANO
Veileder: Astrid Aksnes
Medveileder: Jana Jágerská
Juli 2023

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for elektroniske systemer

NTNU
Kunnskap for en bedre verden

Problem Description

The master's project was carried out under supervision from Professor Astrid Aksnes at NTNU and guidance from Associate Professor Jana Jágerská at the Ultrasound, Microwaves, and Optics (UMO) group at UiT The Arctic University of Norway. It was initiated in February and carried out separately from the specialization project that normally precedes master's projects at NTNU. The work was introduced as part of a project aiming to miniaturize sensors for trace gas sensing based on tunable diode laser absorption spectroscopy. Development of sensor systems for trace gas sensing is of widespread interest for applications in climate research and monitoring, biomedicine, industrial environments, and more. Miniaturization of trace gas sensors are of special interest as on-chip integration allows for bulk- and cost-reductions that facilitate their extensive deployment. A key area of research for such miniaturized sensor systems is the development of low-loss waveguides optimized for high on-chip light-analyte interaction.

The goals of the master's project could be summarized as:

- Investigate and determine a set of optimal fabrication parameters for a free-standing waveguide design for sensing applications using two-dimensional Finite-Difference Method simulation techniques.
- Based on the obtained parameters, fabricate a set of free-standing waveguide prototypes in the NTNU NanoLab cleanroom.
- If time and project progression allows, experimentally measure the performance of the waveguide prototypes at the Department of Physics at the University of Tromsø.

The master project work required a literature search, learning the advanced simulation tool Lumerical MODE, simulating the novel free-standing waveguide with grating structure, being trained on NTNU NanoLab equipment (EBL, ICP-RIE, SEM, HF-etching, etc), optimizing recipes for underetched/free-standing waveguides, fabricating the simulated free-standing waveguide, learning how to use the structural characterization equipment, and characterizing and coupling light into the fabricated waveguides to assess performance. In addition a one week research stay in Tromsø in May was organized in order to optically characterize the fabricated waveguides.

Abstract

Development of sensor systems for trace gas sensing is of widespread interest for applications in climate research and monitoring, biomedicine, industrial environments, and more. Miniaturization of trace gas sensors are especially interesting as on-chip integration allows for bulk- and cost-reductions that facilitate their extensive deployment. Emissions are often local in nature and part of complex chemical and biological pathways, and increasing sensor mobility and cost-effectiveness allow for accurate quantification and monitoring of trace gases over time in the regions of emission. One such emission source is the release of methane from Arctic soil due to thawing of the permafrost. Such production and release of methane has been linked to complex metabolic processes in bacteria inhabiting the soil, and on-chip trace gas sensors are a key enabling technology to study these chemical processes in their natural enviroment. Similarly, low-cost gas sensors are also showing promise for detection and tracking of biomarkers for a number of health conditions, and are thus a candidate for inclusion in medical diagnostics and screening programs.

In order to faciliate miniaturization of conventional optical gas sensors, development of low-loss waveguides lithographically defined in the chip itself has been a key area of research. Waveguide based sensors would rely on interaction between the evanescent field of the waveguide and the analyte, and would need to be carefully designed to balance propagation losses with analyte interaction. The free-standing waveguide structures with sub-wavelength cladding that are the focus of this thesis have previously been identified as promising designs for sensing applications. This master's project aimed to investigate and determine optimal fabrication parameters for such free-standing waveguide designs with emphasis on maximizing the air confinement factor, a figure of merit for normalized light-analyte interaction in optical sensing systems, and minimizing propagation losses. This was done through extensive simulations using the two-dimensional Finite-Difference Method (FDM), fabrication of the structures in the cleanroom at NTNU NanoLab, and subsequent structural and optical characterization at NTNU and UiT.

The magnitude of the approximated air confinement factor was in simulations found to be highly sensitive to variations of the waveguide core dimensions and the air duty cycle of the sub-wavelength grating that acts as a cladding to the core. Optima for the approximated air confinement factor were found to emerge due to a competing relation between the localization of the normalized electrical field energy density in the air, and the increase of the mode group index with larger field localizations in constitutents of high refractive index. The magnitude of the optima were found to increase with lower duty cycles and were found to be above unity in all designs with air duty cycles below 20 %. With a basis in preliminary simulation results, several batches of free-standing waveguides were successfully fabricated at NTNU NanoLab with measured duty cycles of 15 %, 10 %, and 7 %. Analysis of loss simulations revealed the propgating modes in these to be susceptible to significant lateral and vertical leakage losses due to their low core confinement and their wide evanescent field distributions. Additionally, overlap analysis assuming end-fire coupling revealed the presence of large coupling losses.

Optical characterization was in this work limited to measurements of the propagation loss of the fabricated waveguides. In general, the propagation loss was measured to be significantly higher than suggested by the simulated loss results, likely caused by a combination of factors pertaining to deviations from the set design parameters following fabrication, scattering losses, and uncertainties in the approximations required for two-dimensional FDM simulations.

To address the large propagation and coupling losses, which is believed to be the limiting factor for the practical operation of the waveguides, a number of suggestions for future continuation of the work have been presented. While the operation of lower duty cycle waveguide designs will be limited in practice by the vertical losses of relatively thin silica insulator layer, successfully fabricating low-loss free-standing waveguides of above-unity air confinement factor should be readily possible by targeting structures of duty cycles in the 15 % – 20 % range.

Sammendrag

Utvikling av nye sensorssystemer for måling av sporgasser er av stor interesse blant annet innen forskning på klimautslipp, innen arbeid med biomedisin og -teknologi, og i overvåking av utslipp i industrielle miljø. Spesielt interessant er integrasjon av systemene i små mikrobrikker. Ved å senke produksjonskostnader, størrelse, og vekt, tillater slik miniatyrisering økt utplassering av sensorene i områder hvor utslipp foregår. Utslipp er ofte del av komplekse kjemiske og biologiske vekselvirkninger som kan variere stort også innen mindre geografiske områder. Produksjon og utslipp av klimagassen metan i Arktis grunnet økt tining av permafrosten er et eksempel på en slik utslippskilde. Den er blitt knyttet til komplekse metabolske prosesser i jordlevende bakterier, og utviklingen og utplasseringen av miniatyrsensorer kan bidra til å studere disse i deres lokale miljø. Slike gassensorer har også vist stort potensial for å kunne brukes til å detektere og overvåke biomarkører indikative for rekke helsetilstander, og er derfor lovende for bruk i medisinsk diagnostikk.

Utviklingen av bølgeledere med lavt forplantningstap er et nødvendig ledd i miniatyrisering av konvensjonelle optiske systemer for gassmåling. Bølgelederbaserte sensorer bygger på interaksjonen mellom de ledede bølgene og analytten, og må designes med omhu for å oppnå ønsket om lave tap og høy interaksjon mellom lys og analytt. De frittstående bølgelederstrukturene som er fokus for dette arbeidet har tidligere vist seg å være lovende kandidater til dette formålet. Denne masteroppgaven tok sikte på å undersøke og bestemme optimale fabrikasjonsparametere for slike bølgelederdesign med hensyn på å maksimere lys-analytt interaksjonen og minimere forplantningstap. Dette ble gjort ved bruk av endelig-differanse metoden (Finite-Difference Method) til todimensjonale simuleringer av bølgelederdesignet, fabrikasjon av strukturene i renrommet ved NTNU NanoLab, og påfølgende strukturell og optisk karakterisering ved både NTNU og Universitet i Tromsø (UiT).

Størrelsen på interaksjonen mellom lys og analytt ble gjennom simuleringer funnet til å være svært avhengig av bredden på bølgelederens kjerne og driftssyklusen til gitteret som omkler kjernen. Optimale parametere for maksimering av interaksjonen mellom lys og analytt oppsto som følge av et konkurrerende forhold mellom den normaliserte elektriske energifelttettheten til lyset i den analyttholdige luften og materialenes påvirkning på lysets gruppeindeks. Størrelsen på toppunktene for interaksjonen mellom lys og luft økte med reduksjon i gitterets driftssyklus. I design med lavere driftssyklus enn 20 % var interaksjonen mellom lys og analytt større enn det som er mulig å oppnå med en uledet lystråle i luftfylt rom. Flere sett med frittstående bølgeledere med driftssykluser 15 %, 10 %, og 7 %, ble vellykket fabrikert ved NTNU NanoLab. Tapssimuleringer avdekket at bølgelederene var utsatt for betydelige forplantningstap grunnet horisontal og vertikal spredning av lyset grunnet bred intensitetspredning. I tillegg avslørte videre analyse at det trolig ville forekomme store tap ved innkobling av lys direkte inn i bølgelederne.

Optisk karakterisering av de fabrikerte bølgelederne er i dette arbeidet begrenset til målinger av forplantningstapet deres. Forplantningstapet ble generelt sett målt til å være betydelig større enn det som ble anslått i tapssimuleringene. Årsaken knytter seg trolig til flere faktorer, blant annet avvik mellom de designede og fabrikerte strukturene, spredningstap, og usikkerhet knyttet til approksimasjonene brukt som grunnlag for simuleringsprosessen. De store forplantnings- og innkoblingstapene er trolig de begrensende faktorene for funskjonsevnen til bølgelederne, og en rekke punkter har blitt foreslått for å imøtekomme årsakene til disse. Tas disse til følge vil det trolig raskt være mulig å fabrikere funksjonsdyktige frittstående bølgeledere med høy interaksjon mellom lys og analytt og minimale tap.

Acknowledgements

I would like to thank my supervisor Professor Astrid Aksnes for her excellent supervision, support and discussion throughout the master's project. I am also grateful to UiT associate professor Jana Jágerská for her supervision and willingness to bring me onto the project. I consider myself fortunate to have been allowed to work and extend my competencies as an engineer on such an ambitious and inspiring project, and for the continued confidence that was placed in me during the course of the work.

I would like to thank process engineer Jens Høvik at NTNU NanoLab for sharing from his knowledge and experience of waveguide fabrication and taking the time to assist me with planning, execution, and analysis of the waveguide fabrication. I am also grateful for the excellent training courses provided by the other staff at NanoLab.

I would also like to specially thank UiT postdoctoral research fellow Roman Zakoldaev for his guidance and supervision during optical measurements of the fabricated waveguides, as well as making sure that my research stay at UiT was gainful. I also appreciate the interactions with the other team members in Tromsø, and would like to specifically thank PhD candiate Henock Demessie Yallew for his assistance and availability during planning of the waveguide fabrication.

The Research Council of Norway is acknowledged for the support to the Norwegian Micro- and Nano-Fabrication Facility, NorFab, project number 295864.

Contents

1 Introduction

1.1 Background and Motivation

1.1.1 Trace Gas Sensing

This master's thesis has been part of a project aiming to miniaturize sensors for trace gas sensing based on tunable diode laser absorption spectroscopy. Development of sensor systems for trace gas sensing is of widespread interest for applications in climate research and monitoring, biomedicine, industrial environments, and more. Miniaturization of trace gas sensors are of special interest as on-chip integration allows for bulk- and cost-reductions that facilitate their extensive deployment.

The term *trace gas* denotes any gas that is only present in a relatively small amount in a given environment. On earth, the atmosphere is mainly made up of nitrogen (78.1 %), oxygen (20.9 %), argon (0.934 %), and water vapour, while the remaining approximate 0.1 % is made up of a variety of trace gases. Out of the trace gases, carbon dioxide (419 ppm), helium (18.2 ppm), neon (5.3 ppm), methane (1.9 ppm), hydrogen (0.56 ppm), and nitrous oxide (0.33 ppm) are the most abundant [1][2]. Along with hydrochlorofluorocarbons (HCFCs), hydrofluorocarbons (HFCs) and ozone, carbon dioxide, methane, and nitrous oxide fall into a category of chemical compounds known as greenhouse gases. The greenhouse gases all share the characteristic trait of absorbing and dispersing heat radiation as it propagates from the surface of the earth and into space, thus contributing to keeping the temperature of the atmosphere.

Greenhouse gases play a critical role in regulating the temperature of the earth and have provided the prerequisites for life on earth as we know it. However, since the advent of the industrial revolution, the rapid increase in atmospheric concentration of carbon dioxide, methane, and nitrous dioxide have been found to be a well-documented contributor to global warming and climate change. The climate change provided by anthropogenic emissions are expected to have large consequences for soceties and eco-systems as temperatures rise and weather systems shift from their historic equilibriums. For instance, sea levels are expected to rise as polar ice caps melt, agricultural safety diminishes as droughts become more severe and frequent, bio-diversity will suffer as ecosystems are perturbed, and global trade may face upheavals due to reduced availability of commodities. Resource scarcity and mass migration in affected areas may also give rise to political and military tensions.

Large scale deployment of sensors are required to monitor and track the extent of greenhouse gas emissions and identify and employ effective measures to control them. Emissions are often local in nature and part of complex chemical and biological pathways, and it is thus useful to be able to accurately quantify and monitor presence of trace gases over in time the regions of emission. One such example is the release of methane from arctic soil due to thawing of the permafrost [3]. This effect has been a central field of research for CAGE, the centre for arctic gas hydrate, environment, and climate at the University of Tromsø. Of particular interest is the investigation of greenhouse gas metabolism and production of bacteria living in the soil [4]. Studies of metabolic processes in living organisms are possible by using a technique called istopic labeling, in which ^{13}C-labeled molecules can be tracked through chemical processes[5]. This is also a promising method for detection of biomarkers for a number of health conditions ranging from diabetes to certain forms of cancer, and low-cost gas sensor are thus a candidate for inclusion in medical diagnostics and screening programs[6] [7].

The inaccesibility of the vast arctic makes it challenging to monitor emissions over time using traditional bulky gas sensors. By reducing their size, mass, and unit cost, sensors can be deployed in larger numbers over increased areas. Their increased mobility gives flexibility to the method of introducing the sensors into the environment. The sensors could for example be integrated into

the sensor suite of research drones and airplanes, or carried into remote areas by foot. Advances in energy-friendly microcontrollers and the Internet-of-Things (IoT) facilitiate integration of trace gas sensors into clusters that can transfer data wirelessly and be monitored remotely in real-time.

Such integrated sensor systems also have significant use cases for leak detection and health and safety monitoring in industrial environments where hazardous gases are used or have the potential to form. Examples of such are in laboratories, chemical plants, in industrial farms and agricultural settings, and in oil and gas refineries. One could also hypothesise that sensors could be employed in cities for high-resolution real time monitoring of air pollutants and irritants that periodically affect people with asthma and other respiratory conditions. Furthermore, contingency plans for handling ecological catastrophies, chemical spills, and deliberate release of chemical and biological agents could include distribution of low-cost on-chip trace gas sensors to the population in order to increase emergency preparedness.

1.1.2 On-chip Integration of TDLAS Based Methane Sensors

Many methods for gas detection are available, each with their own advantages and drawbacks, for example in measurement resolution, specificity, power consumption, and size. A thorough review of the principles and techniques governing the field of optical gas sensing has been made by Hodgkinson et al. in [8] and McManus et al. in [9]. Commercially available systems capable of sensing trace amounts of gases are bulky, large, expensive, and thus hard to deploy in remote areas and in large numbers. The most sensitive intrumentation often rely on a technique called tunable-diode laser absorption spectroscopy, which works on the principle of interaction between a laser beam and the molecules of the gas analyte. All molecules are characterized by different molecular properties such as their constituent atoms, chemical bonds, and their geometry, which enables the molecules to absorb light at certain wavelengths. The magnitude of absorption is determined by the wavelength dependent absorption coefficient $\alpha(\lambda)$, and the specific set of frequencies that are absorbed give rise to specific absorption spectra, often called *molecular fingerprints*, that are unique to each molecule. By tuning the frequency of the laser diode over a specific frequency range, the resulting absorption spectrum can be detected and compared to database absorption values for different analytes in order to yield an estimate for the analyte concentration.

The intensity of a transmitted signal in a gaseous molecular medium of low concentration is attenuated following the Lambert-Beer law

$$I(\lambda) = I_0(\lambda)e^{\alpha_m(\lambda)L}, \tag{1.1}$$

where $I_0(\lambda)$ denotes the inital intensity of the beam, $\alpha_m(\lambda)$ the effective absorption coefficient of the medium, and L the interaction length. To achieve high sensitivty and low limits of detection it is necessary have large frequency-specific signal attenuation. In conventional TDLAS systems this is done by increasing the interaction length through the introduction of an analyte filled free-space multipass gas cell in which the beam can travel back and forth between low-loss mirrors before detection. The most precise systems can feature interaction lengths in the hundreds of meters. Additionally, the operating laser frequency range is chosen to include distinctive frequencies of arge absorption coefficients.

The interaction cell is typically the largest part of conventional TDLAS systems, as the laser diode and detector often already already are fabricated on a scale that is suited for on-chip integration. As the overarching goal of the project in which this work belongs is to develop fully on-chip methane sensors for trace gas sensing with sensitivities in the ppm to ppb range, it is necessary to integrate the interaction volume onto the chip. The most promising way to do this is by making use of waveguides, which confines light to propagating modes in structures lithograpically defined in the chip itself. Waveguide based sensors would rely on interaction between the evanescent field

of the waveguide and the analyte, and would need to be carefully designed to balance propagation losses with analyte interaction.

Both straight and spiral waveguides of significant lengths in the order of meters have previously been fabricated with extremely low losses [10][11] of below 0.1 dB/cm. However, in waveguides designed specifically for optical transmission, the modes are generally confined to the waveguide cores and do not typically interact significantly with the surrounding mediums. Trace gas sensing requires evanescent fields of extensive spatial distribution and energy confinement in order for the mode to adequately interact with the gaseous analyte. Obtaining the evanescent field interaction needed for sensing purposes typically leads to significant losses in the order of several dB/cm [12] as the propagating mode scatters on surface roughness and imperfections at the boundaries of material interfaces between the waveguide constituents and the analyte. Development of extremely sensitive on-chip methane sensors have thus been focused on exploring materials, designs, and processing techniques that allow for balancing the propagation loss and the analyte interaction.

1.2 Thesis Scope

1.2.1 Objectives and Structure of the Thesis

Free-standing waveguides have previously been shown to be a promising design for sensing applications [13][14][15][16][17]. Elevating the waveguide core from the substrate medium allows for increased light-analyte interaction while also serving to reduce absorption losses and leakage losses in the substrate. The design that will be investigated in this master's thesis features a free-standing silicon waveguide core cladded by a silicon subwavelength grating produced by lithographic patterning and subsequent etching of a Silicon-on-Insulator (SOI) wafer. The grating structure allows for increased light-analyte interaction in between the silicon grating tethers, while also reducing lateral leakage into the silicon device layer. The subwavelength grating also acts as one-dimensional photonic crystal that slows down the group velocity of the light, increasing the potential light-analyte interaction beyond what is achieveable for a beam propagating through free-space.

The free-standing waveguides have been designed for operation in the mid-infrared (MIR) in order to both reduce the effect of scattering losses, which generally scales with $1/\lambda^4$, and make use of the increased values of the methane absorption coefficient around 3.25 µm. Additonally, the constituents of the SOI wafer, silicon and silica, are practically transparent at this wavelength, which allows for the use of processes compatible with common semiconductor manufacturing. This significantly eases fabrication.

From the overarching thesis scope, a clear set of research objectives can be separated out and defined. The work presented in this thesis aims to:

- Investigate and determine a set of optimal fabrication parameters for a free-standing waveguide design for sensing applications using two-dimensional Finite-Difference Method simulation techniques.
- Based on the obtained parameters, fabricate a set of free-standing waveguide prototypes at the NTNU NanoLab cleanroom.
- If time and project progression allows, experimentally measure the performance of the waveguide prototypes at the Department of Physics at the University of Tromsø.

In addition to the research objectives outlined above, the project work has included development of extensive Python code enviroments as tools for automated batch execution and analysis of ANSYS Lumerical MODE simulations, lithographic mask design, and post-processing of propagation loss measurement results. The code enviroments have been written with an emphasis on generalization and modularization in order to facilitate their use in paralell and subsequent research projects.

The thesis will continue with an introduction into the fundamental principles of light propagation in waveguides and how computational methods are leveraged to solve for the properties of more complex waveguide designs. Then, key technologies and techniques enabling the fabrication of silicon-based integrated photonic circuits are treated in brief, followed by an overview of the simulation, fabrication, and characterization of the free-standing waveguides, ultimately leading up to a presentation and discussion of the results and insights gained.

Unless explicitly stated otherwise, all written text, code, simulation results, fabrication, images, and figures are the product of the author's own work on the project.

2 Theoretical Background

2.1 Optical Waveguides

This section aims to provide the fundamentals of guided-wave optics that is necessary to form a basis for the understanding and discussion of the simulated and measured behaviour of the treated free-standing waveguides. It will start off by introducing the principles governing simple planar-mirror waveguides, before expanding the scope to consider more complex waveguide geometries and field distributions, ultimately leading up to a discussion of the free-standing waveguide design and the special considerations that must be made in terms of loss analysis, coupling, and geometry when considering its fabrication. The material for the section is largely sourced from *"Fundamentals of Photonics"* [18] and the excellent PhD thesis of Marek Vlk [19]. The reader is encouraged to consult these works for a more comprehensive approach to the topic of guided-wave optics.

2.1.1 Fundamentals of Light Propagation in Simple Optical Waveguides

The perhaps simplest and most easy to conceptualize waveguide consists of two paralell mirrors in space oriented in the x-z plane separated by a distance d in the direction of the y-axis. Consider an optical ray representig a plane wave traveling at an angle θ with the z-axis. The optical ray is reflected at the top mirror, and travels at an angle $-\theta$ before intersecting the bottom mirror, reflecting at an angle θ, and once again continuing again toward the top mirror. For each reflection, a phase shift of π is applied to the phase of the ray. It can be shown that in order for the reflections of the plane wave in such a roundtrip to constructively interfere, the so-called self-consistency conditon

$$\frac{2\pi}{\lambda} 2d \sin\theta = 2\pi m, \tag{2.1}$$

must be fulfilled. Here λ denotes the wavelength of the plane wave and m the so-called *mode number*, an integer equal to or larger than one. It is clear that at a fixed wavelength and mirror spacing, the self-consistency relation is only satisfied for certain pairs of bounce angles θ_m and mode numbers m.

$$\sin\theta_m = m\frac{\lambda}{2d}. \tag{2.2}$$

The same principle shown for a planar mirror waveguide applies to planar dielectric waveguides but the mirrors are replaced by dielectric materials of refractive indices n_{tc} and n_{bc} corresponding to the top and bottom cladding, respectively, and the previously empty gap between the mirrors now contains a dielectric material of refractive index $n_1 > max\{n_{bc}, n_{tc}\}$. For such a case, self consistency dictates that the relation

$$\frac{2\pi}{\lambda} 2d \sin\theta - 2\phi_r = 2\pi m, \tag{2.3}$$

must be satisfied, where ϕ_r denotes the phase shift associated with a reflection at the dielectric boundary. Furthermore, to achieve a state of total internal reflection (TIR), where the entire intensity of the propagating wave is reflected at the dielectric boundaries, it follows from Snell's law that the bounce angle θ_m of a propagating mode satisfying Equation 2.3 must be smaller than the critical angle θ_c dependent upon the refractive indices of the waveguide constituents

$$\sin\theta_c = \frac{min\{n_{bc}, n_{tc}\}}{n_1}. \tag{2.4}$$

Figure 2.1 provides an illustration of total internal reflection in a planar dielectric waveguide.

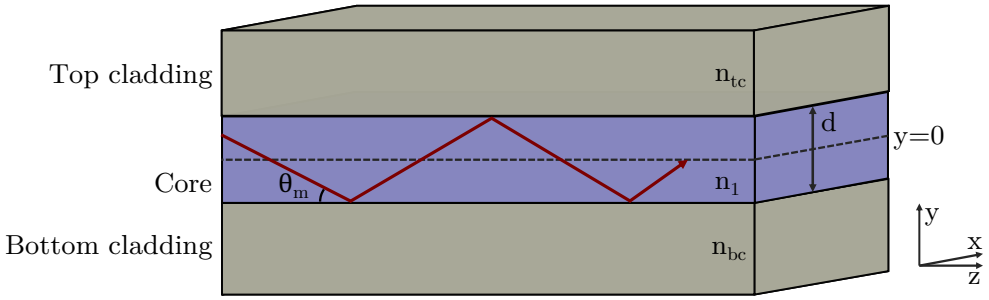


Figure 2.1: Illustration of a propagating mode in a planar dielectric waveguide where the refractive index of the core, n_1, is larger than both the refractive index of the bottom cladding, n_{bc}, and the refractive index of the top cladding, n_{tc}. The thickness of the core is denoted by d, while $\theta_m \leq \theta_c$ denotes the bounce angle of the mode

Thus, the constraints put on the guided modes by Equations 2.3 and 2.4 give rise to a finite set of propagating modes. Each guided mode is characterized by an effective index n_{eff} or the propagation constant β_m given by

$$\beta_m = n_{eff,m}k_0 = n_1 k_0 \cos\theta_m, \tag{2.5}$$

where k_0 denotes the free-space angular wavenumber of the propagating mode. It follows from the above equations that a structure will be unable to support any propagating modes at a certain operating wavelength if the bounce angle θ_m of the fundamental mode is larger than the critical angle θ_c of the structure. By extension, it can be seen from Equation 2.5 that supported modes must have a propagation constant larger than the so-called cut-off propagation constant β_c, given by

$$\beta_c = max\{n_{bc}, n_{tc}\}k_0, \tag{2.6}$$

which implies that the effective index of the fundamental mode must be larger than the refractive indices of both the top and bottom claddings for the mode to propagate.

In order to understand the field distributions across the waveguides a more comprehensive electromagnetic approach is required. Derivation of the field distributions is covered in detail in [18] and [19] and will not be reiterated here. For TE modes, which will be the focus of simulations and measurements in this project, the electric-field complex amplitude $E_x(y,z)$ of a mode m propagating in a planar dielectric waveguide can be shown to be modulated along the thickness of the slab according to

$$u_m \propto \begin{cases} \cos\left(\frac{2\pi\sin(\theta_m)y}{\lambda}\right), & m = 0,2,4,... \\ \sin\left(\frac{2\pi\sin(\theta_m)y}{\lambda}\right), & m = 1,3,5,..., \end{cases} \quad -\frac{d}{2} \leq y \leq \frac{d}{2}, \tag{2.7}$$

and

$$u_m \propto \begin{cases} e^{-\gamma_m y}, & y > \frac{d}{2} \\ e^{\gamma_m y}, & y < -\frac{d}{2} \end{cases} \tag{2.8}$$

where γ_m denotes the so-called extinction coefficient of the mode. Equations 2.7 and 2.8 describe the shape of the electric-field distribution inside the core and the claddings, respectively. Dielectric boundary conditions ensure that the internal field must match the external field at the boundaries between core and cladding. The exponentially decaying field in the cladding is called the evanescent field and is obtained from substitution of $E_x(y, z)$ into the Helmholtz equation. It follows that the extinction coefficient of a mode relates to its propagation constant and the properties of the cladding by

$$\gamma_m^2 = \beta_m^2 - n_c^2 k_0^2, \tag{2.9}$$

where n_c denotes the refractive index of the cladding. By substitution of Equation 2.5 into Equation 2.9 it can be seen that the extinction coefficent decreases with a decrease in the effective index of the mode. This general result is important for understanding the associated spatial intensity distributions of the simulated modes in Section 5.1.

2.1.2 Expansion to More Complex Waveguide Geometries

The principles of the one-dimensional planar waveguides treated in Section 2.1.1 are readily expandable to higher dimensions and waveguides of increased complexity. Modes in such structures will be partitioned into light of both TE- and TM-polarization as boundary conditions need to be satisfied on boundaries both parallel and perpendicular to the x-direction. However, structures can be engineered to support modes of predominantly one polarization, and these quasi-TE and quasi-TM modes are often just referred to as TE and TM.

Although several analytical methods such as Marcatili's method and the effective index method exist to solve for the spatial intensity distribution and effective index of propagating modes mode, complex designs necessitates the use of approximations that may jeopardize the accuracy of the results. With the increase in computation power that has become available in recent years, techniques based on the finite-difference method (FDM) and finite element method (FEM) are generally much more feasible and computationally efficient. The principles and use of FDM to solve for the modal properties of free-standing waveguides will be covered in much more detail in Section 2.2 and Section 4.2, respectively.

2.1.3 The Confinement Factor

The overall bulk gain/loss coefficient α_m of a waveguide is related to the individual gain/loss coefficients α_i of its constituent materials by the confinement factor Γ_i, which serves as a measure of the interaction between the guided mode and a constituent material. The bulk gain/loss coefficient can be expressed as the sum of individual gain/loss coefficients weighted by their respective confinement factors

$$\alpha_m = \sum_i \Gamma_i \alpha_i \tag{2.10}$$

In the simple case of a wave propagating through space occupied by a single medium, for example air, the entirety of the wave will interact with the medium for an air confinement factor Γ_{air} of 100 % such that $\alpha_m = \alpha_{air}$. For waveguide structures comprised of multiple media, however, the confinement factor of individual constituents may take on a range of values from 0 % to above unity depending on the spatial intensity distribution of the guided mode and the waveguide geometry. As the bulk gain/loss coefficient, and thus confinement factor, is related to light absorption in the waveguide section of length L through the Lambert-Beer law

$$I = I_0 e^{-\sum_i \Gamma_i \alpha_i L}, \tag{2.11}$$

it is evident that the distribution of confinement factors between waveguide constituents is of key importance for sensing applications. Increasing the confinement factor in the analyte yields increased sensitivity to changes in the loss coefficient of the analyte, which in turn is dependent on the analyte concentration. The confinement factors may reach values above unity; in these cases, waveguide-based sensors are more sensitive to variations in the absorption coefficient of the gain medium than their traditional (free path) counterparts [13].

Robinson et al. [20] have shown that the confinement factor in a medium i can be expressed as

$$\Gamma_i = \frac{n_g}{\Re\{n_i\}} \frac{\iint_i \epsilon |\mathbf{E}|^2 dxdy}{\Re\{\iint_{-\infty}^{\infty} \epsilon |\mathbf{E}|^2 dxdy\}} = \frac{n_g}{\Re\{n_i\}} \frac{\iint_i \epsilon_i |\mathbf{E}|^2 dxdy}{\Re\{\sum_j \iint_j \epsilon_j |\mathbf{E}|^2 dxdy\}}, \tag{2.12}$$

where n_g is the group index of the mode, n_i and ϵ the complex refractive index and permittivity of the medium, respectively, and $\mathbf{E}$ the electric field intensity. The first part of the fraction is related to the properties of the material, while the second fraction is a measure of the normalized electrical field energy density in the medium. In a bulk medium the confinement factor will be limited to a maximum of 100 %, but carefully optimized waveguide designs can reach confinement factors in the gain medium of well above unity. This can be achieved by optimizing the mode energy distribution across the waveguide such that the group index increases above the refractive index of the gain medium, and the group velocity is reduced compared to the light velocity in the bulk medium. Thus, the light-matter interaction in the gain medium is increased as the light has more time to interact with the gain medium per unit length of propagation. Distributing the mode energy does come at the cost of a reduction to the normalized field energy density in the gain medium, so an optimal balance must be achieved between the two fractions in Equation 2.12. Section 5.1.1 will treat in much more detail implications of this competitive relation on the air confinement factor optima of the free-standing waveguide design.

The group index at a wavelength λ can be calculated as

$$n_g = n_{eff}(\lambda) - \lambda \frac{d}{d\lambda} n_{eff}(\lambda), \tag{2.13}$$

where n_{eff} denotes the effective mode index. Furthermore, to estimate the group index from a limited number of simulation results, the rough forward difference approximation

$$\frac{d}{d\lambda} n_{eff}(\lambda) = \frac{n_{eff}(\lambda + \Delta\lambda) - n_{eff}(\lambda)}{\Delta\lambda} \tag{2.14}$$

can be used for the wavelength dependent derivative of the effective mode index at small values of $\Delta\lambda$.

By means of the Poynting theorem, Visser et al [21] have also been able to express the confinement factor as

$$\Gamma_i = \frac{c\epsilon_0 \Re\{n_i\} \iint_i |\mathbf{E}|^2 dxdy}{\Re\{\iint_{-\infty}^{\infty} (\mathbf{E} \times \mathbf{H}^*) \cdot \mathbf{e}_z dxdy\}}, \tag{2.15}$$

where c denotes the free-space velocity of light, ϵ the free space permittivity, $\mathbf{H}$ the magnetic field intensity, and $\mathbf{e}_z$ the normal vector in the propagation direction. Equation 2.12 will prove useful in the interpretation of the large air confinement factors in Section 5.1.1, while Equation 2.15 lends itself nicely for the Lumerical MODE implementation of the confinement factor described in Section 4.2.4.

2.1.4 Overview of General Loss Mechanisms in Optical Waveguides

Losses in optical waveguides refer to the attenuation of the optical signal propagating along the waveguide, resulting in a reduction in the amplitude and power of the signal. There are various mechanisms that contribute to these losses, with absorption loss, substrate loss, and bending loss being some of the most significant.

Absorption loss occurs when the light in the waveguide is absorbed by the material through which it is propagating. This can be due to intrinsic material properties such as impurities or defects, or extrinsic factors such as surface contamination or metal contacts. The amount of absorption loss depends on the wavelength of the light and the material properties, with some materials exhibiting higher absorption at certain wavelengths.

Substrate loss, on the other hand, occurs when the optical signal interacts with the substrate on which the waveguide is fabricated. This can be due to scattering, reflection, or absorption of the signal by the substrate material. The amount of substrate loss depends on the refractive index contrast between the waveguide and the substrate, as well as the material properties of the substrate.

Bending loss occurs when the waveguide is curved or bent, causing the light to scatter or leak out of the waveguide. This can be due to the mode mismatch between the straight and curved sections of the waveguide, or due to the finite size of the waveguide. The amount of bending loss depends on the curvature radius, the waveguide dimensions, and the refractive index contrast between the core and cladding.

2.1.5 Design Considerations for Free-Standing Silicon Waveguides

The material composition and geometry of the free-standing waveguides that are the focus of this thesis necessitate closer treatment of some considerations related to their losses, grating structure, effective properties, and options for interfacing with other optical components. A schematic illustration of the free-standing waveguide design is presented in Figure 2.2. First of all, losses due to material absorption in the silicon or silica are expected to be negligble, as both constituents are considered to be transparent around the expected laser operating wavelength of 3.274 µm. Silicon is transparent in the wavelength range 1.1 µm – 7 µm [22], while pure silica is transparent at 300 nm – 3.5 µm [23]. However, OH-groups embedded in amorphous oxides readily absorb light in the mid-infrared [24], and may contribute to increased losses in conjunction with water molecules adsorbed to the waveguide surface. Ensuring high silica purities is therefore an important part of the Silicon-On-Insulator (SOI) fabrication that will be considered further in Section 3.1.

Some scattering loss is expected due to imperfections in the constituents of the SOI wafers and in the fabricated structures. Examples of the latter are surface roughness induced during reactive

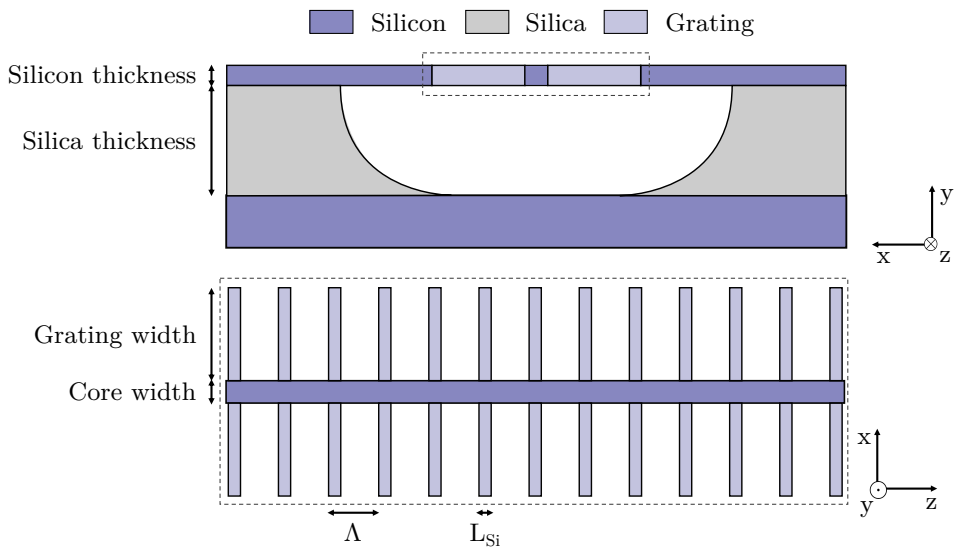


Figure 2.2: Schematic drawing of the free-standing waveguide structure with labeled dimensions. The top figure shows a cross-sectional view of the SOI-based waveguide while the bottom figure shows a top view of the silicon waveguide core and grating structure. The grating duty cycle is defined as the ratio of the width of the spacing between grating tethers, $\Lambda - L_{Si}$, to the grating period, Λ, so that $f = 1 - L_{Si}/\Lambda$.

ion etching (Section 3.3.1) of the silicon device layer, or inhomogenities in the sub-wavelength grating. As obtaining large air confinement factors generally requires a strong evanescent field that can interact with surrounding air, the structure is particularily sensitive to losses caused by interaction between the evanescent field and the silicon substrate and device layer, whose overlap may cause significant substrate losses (also termed vertical leakage losses) and lateral leakage losses. The lateral leakage losses may be mitigated by increasing the width of the sub-wavelength grating, while reduction of the vertical leakage loss is restricted by the insulator thickness. High air confinement factor designs might be expensive to fabricate due to the limited availability of SOI wafers with thick insulator layers.

To avoid radiative losses, the period Λ of a waveguide consisting of a periodically varying grating must adhere to the so-called Bragg condition

$$\Lambda \leq \frac{\lambda}{2n_{BF}}, \tag{2.16}$$

where λ denotes the laser operating wavelength and n_{BF} the fundamental Bloch-Floquet mode of the structure [25]. Such gratings are said to be *subwavelength*. In the free-standing waveguide design presented in this thesis, the effective mode index n_{eff} of propagating modes during single mode operation is equal to n_{BF}.

In order to be able to model the free-standing waveguide structure using two-dimensional simulation techniques, the three-dimensional properties of the grating must be collapsed into two dimensions. An effective refractive index is introduced to describe the average behavior of light propagating through a composite material with varying refractive index, such as the periodic sub-wavelength grating of the free-standing waveguides. Depending on the material, the effective index can be determined based on a variety of effective medium approximations found in effective index theory, in which the composite material is treated as a homogenous metamaterial with an effective refractive index that accounts for the complex behavior of the material.

The effective refractive index n_{SWG} of the sub-wavelength grating could roughly be approximated

through the simple effective medium approximation

$$n_{SWG} = f n_{Si}^2 + (1 - f) n_{clad}^2, \tag{2.17}$$

where f denotes the grating air duty cycle, and n_{Si} and n_{clad} the refractive indices of the silicon and surrounding medium, respectively [14] [26] [27]. The grating duty cycle is defined as the ratio of the width of the spacing between grating tethers, $\Lambda - L_{Si}$, to the grating period, Λ, so that $f = 1 - L_{Si}/\Lambda$.

Effective index theory has been successfully used to model a variety of metamaterial structures, including photonic crystals, plasmonic structures, and negative index materials. It provides a relatively simple and computationally efficient way to model the optical properties of these materials, and is a useful tool for designing new metamaterial devices and structures. However, it has limitations in modeling more complex structures and phenomena, such as strong coupling and nonlinear effects, which require more sophisticated methods. The results provided by approximations should be verified in three-dimensional simulations and experiments.

Designing purpose-built couplers could be necessary to couple light efficiently into the structures. While the waveguides fabricated in this work will make use of end-fire coupling, a form of coupling where light is focused directly on the end facet of the waveguide, the associated losses could be reduced by introduction of coupler structures that faciliate transfer of optical energy from free space to the waveguides. Examples of such designs are the inverse tapered coupler and the forked waveguide coupler. Implementation of such designs would require further research and investigation, as the subwavelength grating would need to be integrated into such couplers and a method would need to be deviced for precise cleaving of the chip at the location of the coupler. Integration of couplers is thus outside the scope of this project, but discussed further in Section 5.1.2 and Section 5.4.

2.2 Simulating Optical Waveguides with the Finite-Difference Eigenmode Method

FDE stands for Finite-Difference Eigenmode, which is a numerical method used to solve for the modes of electromagnetic waves in optical waveguides. In FDE simulations, the waveguide structure is discretized into a grid of finite-difference cells, and Maxwell's equations are solved numerically to obtain the modal properties of the waveguide, such as the mode profile, effective refractive index, propagation constant, and confinement factor.

FDE simulations can be used to study a wide range of waveguide structures, including planar waveguides, strip waveguides, rib waveguides, photonic crystal waveguides, and more. They are used extensively in the design and optimization of photonic devices, such as lasers, modulators, detectors, and sensors, as well as in the analysis of waveguide-based optical circuits and systems. The contents of this section has largely been based on [28] and [29].

2.2.1 Principles of the Finite-Difference Eigenmode Method

The FDE method is particularly well-suited for waveguide structures, which can be modeled as a two-dimensional cross-section with a finite thickness along the propagation direction. The FDE method typically begins by defining the geometry and material properties of the waveguide structure. This can include the refractive index of the waveguide material, the width and height of the waveguide core, the thickness of the cladding layers, and any other relevant structural details such as slots or gratings.

Once the waveguide structure is defined, the FDE method discretizes the structure into a grid of points, or mesh, which is used to solve for the electromagnetic field distribution in the waveguide. The FDE method then applies appropriate boundary conditions to the waveguide structure, which can include perfectly matched layers (PMLs) to absorb outgoing waves and prevent reflections, or other boundary conditions such as symmetries or anti-symmetries to reduce the size of the simulation domain.

The FDE method then solves for the electromagnetic field distribution in the waveguide using a matrix formulation. This involves discretizing the waveguide into a series of small sections and solving for the field in each section using a matrix equation. The matrix equation is then solved using a numerical differentiation method such as the finite-difference method or the finite-element method [28].

The principle of the finite-difference method is the numerical approximation of differential equations by finite difference equations that can be calculated by a computer. The derivative of a function $f(x)$ can be approximated by the forward, backward, and central difference, or

$$\frac{df(x)}{dx} = \frac{f(x+h) - f(x)}{h} + O(h), \tag{2.18}$$

$$\frac{df(x)}{dx} = \frac{f(x) - f(x-h)}{h} + O(h), \tag{2.19}$$

$$\frac{df(x)}{dx} = \frac{f(x+h) - f(x-h)}{2h} + O(h^2), \tag{2.20}$$

respectively, for sufficiently small step-sizes h [30]. The big O notation corresponds to the difference between the approximated and true derivative, and has a varying dependence on h. These first-order differences can in an analagous way be combined to form higher order differences that are subsequently used to solve for the field distribution and properties of propagating modes.

Once the electromagnetic field distribution is solved, the FDE method can extract the effective properties of the waveguide structure, such as the effective index, the mode field distribution, the propagation constant, and other modal properties. Convergence testing is important to ensure that the numerical results are accurate and reliable.

2.2.2 Convergence Testing

The accuracy and usefulness of FDE simulations is contingent on using the correct solver parameters. Employing a too coarse mesh, or spanning over too little of a simulation region, may lead to inaccurate and varying results as the solver algorithms struggle to find a stable – or *converging* – solution to the governing set of equations. On the other hand, increasing the solver resolution far past the point of convergence very quickly ramps up simulation time while at best only providing marginal benefits to accuracy. Convergence testing can be described as the process of identifying the simulation resolution and spatial extent that is needed to be able to reduce errors from solver instability to an acceptable level while also keeping simulation time at a minimum.

To perform convergence testing, the user varies the mesh dimensions and records the calculated results of the simulation, such as the modal effective index, confinement factor, and propagation loss. The user can then compare the results of each simulation and evaluate whether the results are converging to a stable solution. Convergence testing is often time-consuming, especially when simulating complex structures. However, it is an essential step in making sure that the accuracy and reliability of the simulation results are adequate.

2.2.3 Limitiations of the Finite-Difference Eigenmode Method

One common error that can arise in FDE simulations is numerical dispersion. Numerical dispersion refers to the discrepancy between the numerical results and the exact solution of the differential equation due to the finite difference approximation. This can lead to inaccuracies in the calculated dispersion relations and mode profiles of the simulated waveguides. In order to reduce numerical dispersion, higher-order finite difference schemes can be employed.

Another possible error is the assumption of a perfectly smooth waveguide and straight profile, which may not be realistic in fabricated devices. Surface roughness or imperfections can lead to scattering losses which are not accounted for in FDE simulations. This can lead to discrepancies between the simulated and actual modal properties of the waveguide.

Furthermore, FDE simulations assume that the material parameters are uniform and isotropic throughout the simulation region, which may not be the case in real-world devices. Anisotropy or spatial variations in the refractive index can lead to deviations in the simulated modal properties. Another common source of error is related to the choice of boundary conditions. Improper boundary conditions can lead to reflections and distortions in the simulation results, or failure to identify modes that are supported in actual structures.

It is important to note that the simulation results obtained using FDE methods may differ from the actual behavior of the waveguide. This is due to several factors, including the assumptions made in the model, such as the use of linear materials and the neglect of nonlinear effects. Additionally, the simulation may not capture all of the imperfections and variations present in the physical waveguide. Therefore, experimental verification is necessary to validate the accuracy of the simulation results.

3 Fabrication of Photonic Integrated Circuits

Photonic integrated circuits require a wide array of microfabrication and -processing techniques for successful manufacture and on-chip incorporation of components. This section will in brief treat some of the key technologies and methods enabling the fabrication of silicon-based free-standing waveguides. The contents within are largely based on the extensive coverage of modern physical and chemical methods for nanofabrication provided in *"Nanostructures and Nanomaterials"* [31]. The reader is encouraged to consult the provided literature for a more comprehensive introduction to nanofabrication and silicon microprocessing.

3.1 Silicon-On-Insulator as a Basis for Silicon Photonics

Silicon-on-insulator (SOI) describes a class of semiconductor fabrication where wafers are engineered from a stack of semiconductors and dielectric materials. An active silicon device layer is typically deposited onto an insulator, often silica, which has been layered on top of a thicker silicon substrate. Silicon-on-insulator wafers are broadly used in silicon photonics, where the refrative index contrast between the insulator and the silicon can be leveraged to fabricate integrated optical components such as waveguides, filters, optical amplifers, and lasers. A comprehensive overview of the history, use, and fabrication of SOI wafers is found in [32].

A variety of techniques exist for the purpose of creating well-formed and uniform SOI wafers. For commercial applications mainly two processes are in use. Separation by implantation of oxygen (SIMOX) was first introduced in the late 1970s [33], and works on the principle of implanting oxygen ions into a silicon wafer, followed by annealing to grow a buried layer of SiO_2. The thickness and structure of the layer is controlled by the ion dose, annealing temperature and annealing duration, while the depth of the layer in the silicon wafer is controlled by the kinetic energy of the accelerated ions. Annealing is also important to reduce damage in the silicon device layer caused during implantation by recrystallizing the silicon.

The second technique commonly used in commercial SOI synthesis utilizes a process called wafer bonding. In a wafer bonding process, a film of silica is first grown or deposited on a silicon wafer by means of a physical or chemical method. The insulator thin film is then bonded to the polished surface of another silicon wafer. The thickness of the silicon device layer may then be reduced through a variety of techniques, such as mechanical grinding, etching, and splitting, followed by leveling and polishing.

Production of SOI wafers can be quite expensive depending on the desired thicknesses of the device and insulator layers, as well as the required tolerances for their intended application. Integrated photonic components in the mid-infrared are sensitive to scattering losses and reflective losses caused by imperfections, and good SOI quality is needed to ensure their successful operation. Additionally, components operating in the mid-infrared are sensitive to absorption losses caused by entrapped water molecules in the buried oxide [24], and care must be taken during fabrication to reduce embedded water contents.

3.2 Lithography

Lithography is in the context of micro- and nanofabrication the process of transferring a pattern into a layer of reactive polymer called a *resist*. Following pattern transfer into the resist, the goal is to transfer the pattern into or onto an underlying structure, either by means of a top-down approach such as etching, or a bottom-up technique such as metallization and subsequent lift-off.

The resist layer can be patterned using a variety of radiation sources, such as photons, x-rays, ions, electrons, and atoms, but methods for direct pattern transfer using physical contact masks have also been developed. While different lithography techniques share the same underlying principle of operation, they will have different advantages and drawbacks depending on the dimensions and resolution of the desired pattern, involved materials, and time and cost considerations [34]. For mass production of modern integrated circuits, UV photolithography is the most widely used technique due to the relatively high throughput volume of wafers that can be exposed within a short timespan. The relatively large wavelength of the photons does limit the resolution of the structures, so for prototype fabrication of nanostructures in research settings where the need for high patterning rates is smaller, electron beam lithography (EBL) is often a good alternative when extremely fine feature dimensions on the nanometer scale are required.

Electron beam lithography builds on the principle of targeting a resist with a finely focused beam of accelerated electrons. The resist is sensitive to electron radiation, and is termed either *positive* or *negative*, depending on whether the resist weakens or hardens following exposure. Weakened resist is removed following exposure by application of a selective solvent termed a *developer*. Due to the extremely small De Broglie wavelength of the accelerated electrons, the pattern resolution is not diffraction limited like in the case of UV lithography, but rather limited by beam broadening as the electrons interact with the resist and substrate. Electrons may either collide elastically, where it changes directions without imparting energy, or elastically, where it imparts its energy into the resist. Elastic collisons cause the beam energy to be departed over a wider lateral area than intended, causing the subsequent pattern of exposed resist to widen. To reduce the effect of this phenomen, care should be taken when designing the lithographic mask and when considering the resist, its thickness, the applied electron dose, and the properties of the beam. Dose tests and subsequent structural characterization should be done to compare the results of chosen parameters to the desired exposure profiles and dimensions. Depending on the geometry of the exposed pattern, it is common to calculate a bias width for a certain set of parameters that must be either subtracted or added to features in the lithographic mask. Software is also available to simulate and correct for electron scattering through a process called proximity effect correction (PEC).

3.3 Etching

3.3.1 Reactive Ion Etching

Dry etching is a widespread process in micro- and nanofabrication that involves selectively etching material from a substrate using reactive gases or plasmas. It is employed to fabricate patterns and structures on the surface of microchips with a high degree of precision and accuracy. Dry etching techniques generally offer several advantages over the wet etching methods discussed in 3.3.2, including better control over etching profiles, higher etch rates, and compatibility with smaller feature sizes, but are also generally more expensive and sensitive to variations in process parameters.

Many methods of dry etching are commonly used in fabrication of integrated photonic circuits, including plasma etching and Reactive Ion Etching (RIE). In plasma etching, a plasma is generated by applying an electric field to a reactive gas mixture typically containg fluorine or chlorine based molecules. The plasma chemically reacts with the material to be etched, leading to its removal. In Inductively Coupled Plasma - Reactive Ion Etching (ICP-RIE) a radio-frequency signal is used to generate a high-density plasma due to inductive coupling between the radiation source and the plasma. By applying a DC voltage bias between the target and the RF source, the ionized gases are then accelerated towards the substrate, augmenting the etching process.

3.3.2 Wet Etching

In conjunction with dry etching, wet etching techniques play a significant role in nanofabrication for the selective removal of material layers. Wet etching is commonly employed to define patterns, create features, and substrates in various microchip manufacturing processes.

When silicon dioxide needs to be selectively removed from a substrate, wet etching in hydrofluoric acid is a widely utilized technique. Hydrofluoric acid is highly effective in etching silicon dioxide, while having minimal impact on other materials commonly found in microchip fabrication, such as metals and semiconductors. This selectivity makes it an ideal choice for precise and controlled removal of SiO_2 layers.

The process typically involves immersing the substrate in a hydrofluoric acid solution, which reacts with the silicon dioxide, resulting in its dissolution. Careful attention must be paid to the etching conditions, such as HF concentration, temperature, and etching time, to achieve the desired removal rate and selectivity. To improve process control and prevent peeling of e-beam and photoresist, HF is commonly mixed with a buffering agent such as ammonium fluoride (NH_4F) to form a buffered oxide etchant (BOE).

4 Methods

4.1 Process Overview

This section will treat the process of simulating, fabricating, and structurally and optically characterizing a prototype free-standing waveguide design in line with the research objectives of the work defined in Section 1.2.1, which are to:

- Investigate and determine a set of optimal fabrication parameters for a free-standing waveguide design for sensing applications using two-dimensional Finite-Difference Method simulation techniques.
- Based on the obtained parameters, fabricate a set of free-standing waveguide prototypes at the NTNU NanoLab cleanroom.
- If time and project progression allows, experimentally measure the performance of the waveguide prototypes at the Department of Physics at the University of Tromsø.

While the section may present the different phases of the work in a seemingly stream-lined, sequential manner, it is important to note that the design process was a result of an iterative procedure where insights from latter stages were used as continuous input for repetition of earlier steps in several interations. Figure 4.1 shows a top-level schematic of the process flow. The insights obtained and applied throughout the progression of the depicted feedback loop will be treated in more detail in Section 5.

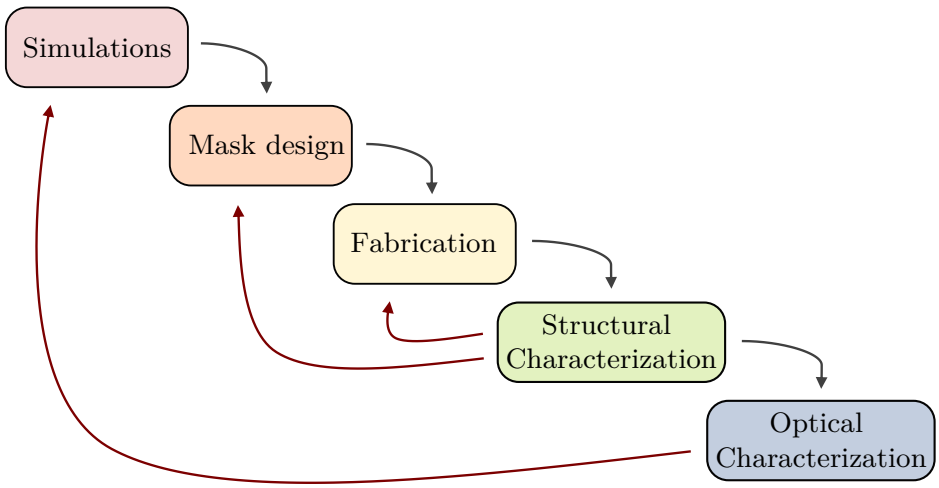


Figure 4.1: Flow-chart showing the iterative nature of the process of simulating, fabricating, and structurally and optically characterizing the free-standing waveguide designs. The black arrows indicate the nominal process flow, while the maroon arrows indicate use of insights gained from latter stages as input for repetition of earlier steps.

4.2 Finite-Difference Eigenmode Simulations for Design Optimization

4.2.1 ANSYS Lumerical MODE

The process of determining the optimal waveguide fabrication parameters heavily relied on simulations carried out in the two-dimensional optical waveguide and coupler solver program ANSYS Lumerical MODE [35], from here on referred to as MODE. MODE enables academic and professional users to precisely model the performance of waveguides and couplers by offering a selection of solvers/engines in a unified and easy-to-use software environment. Notably, the software includes its own scripting language, the aptly named Lumerical Scripting Language, for software control and result analysis. The commands available to the user in the Lumerical Scripting Language are also accessible through a provided Python API, *lumapi*, allowing for extreme user flexibility and automated batch processing of simulation jobs.

4.2.2 Defining the Waveguide Structure Group

The 2D cross-section of the suspended waveguide design was implemented in MODE as a so-called structure group of (n, k)-materials. A (n, k)-material is in MODE a material defined by its frequency dependent real refractive index, n, and imaginary extinction coefficient, k. The implemented structure group is shown in Figure 4.2. The rectangular waveguide core, device layer, and buried substrate have been given the properties of Si [36], while the electrically insulating mid layers on either side of the trench have been given the properties of SiO_2 [36].

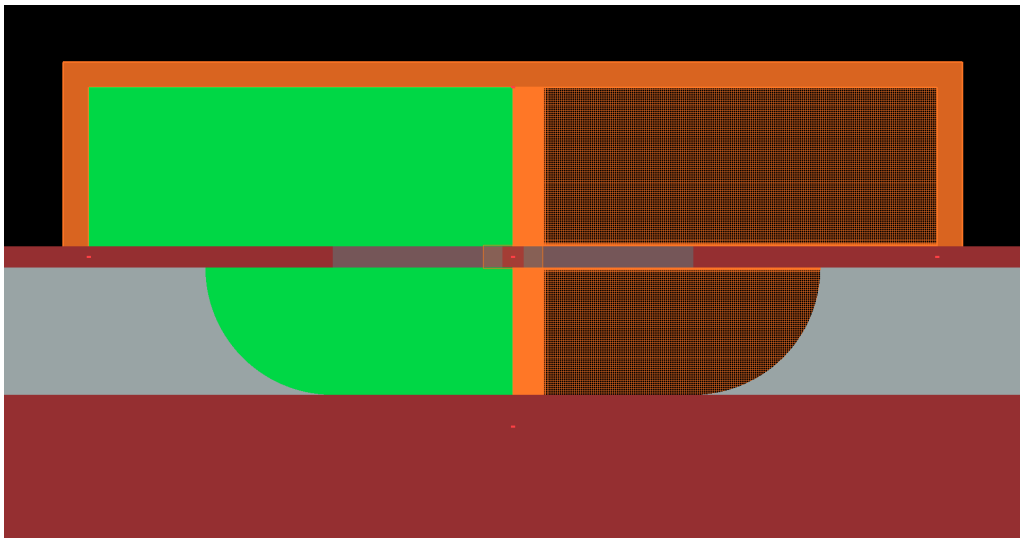


Figure 4.2: Two-dimensional cross-section of the suspended rectangular waveguide design implemented as a structure group in Lumerical MODE. The maroon, light grey, and grey materials correspond to silicon, silica, and the silicon grating, respectively. The figure also features the overlaid FDE solver mesh with perfectly matched layer (PML) boundary conditions and anti-symmetry that is treated in greater detail in Section 4.2.3. The FDE solver mesh is partially hidden by the waveguide design due to the graphical rendering priorities in MODE. The bottom edge of the FDE solver mesh is marked by the small red square visible in the maroon silicon substrate layer.

The periodic waveguide grating that acts as a cladding for the rectangular silicon core has been given the properties of a user-defined metamaterial where n and k are estimated from Equation 2.17, using the set duty cycle and the material properties of the grating constituents. It is important to emphasize that this relation only gives a rough approximation for the effective refractive index

of the sub-wavelength grating, and three-dimensional simulations would be necessary to represent the index properties of the grating more accurately. As shall be explored further in Section 5.1.1, a comparison between the simulated index properties of such a metamaterial in 2D and 3D has been carried out by Zhou et al. [14]. In their work, the index of the propagating modes were found to be lower in the 3D simulations, presumably as a result of a reduced effective refractive index in the grating medium compared to that predicted by Equation 2.17.

The structure has been defined dynamically in a Lumerical Scripting Language script that takes in a set of design parameters defined as variables in the scope of the structure group. The design dimensions have been made accessible to the user, so that changing one parameter, for example the sub-wavelength grating width, will dynamically adapt all other geometries in the structure group to accomodate the parameter update.

4.2.3 Defining and Meshing the Simulation Region

As shown in Figure 4.2 the MODE FDE solver was applied across a region spanning the central portion of the waveguide structure group. Convergence testing of the propagating mode properties as a function of the mesh dimensions found that 50 nm $\times$ 50 nm cells were adequate in most of the simulation region. For the waveguide core and material interfaces, however, mesh dimensions of down to 1 nm $\times$ 5 nm were applied. Mesh refinement in these regions of increased modal intensity was revealed to be necessary following convergence testing of the confinement factor and propagation loss, which exhibited higher sensitivities to changes in the mesh size than the other modal properties of interest. Figure 4.3 clearly displays the mesh size variations imposed by the added reinforcement mesh rib in the immediate vicinity of the waveguide core. The solver span was fixed to 20 µm and 12 µm along the x- and y-axis, respectively, following convergence testing. A more comprehensive account of the applied convergence testing is given in Appendix A.

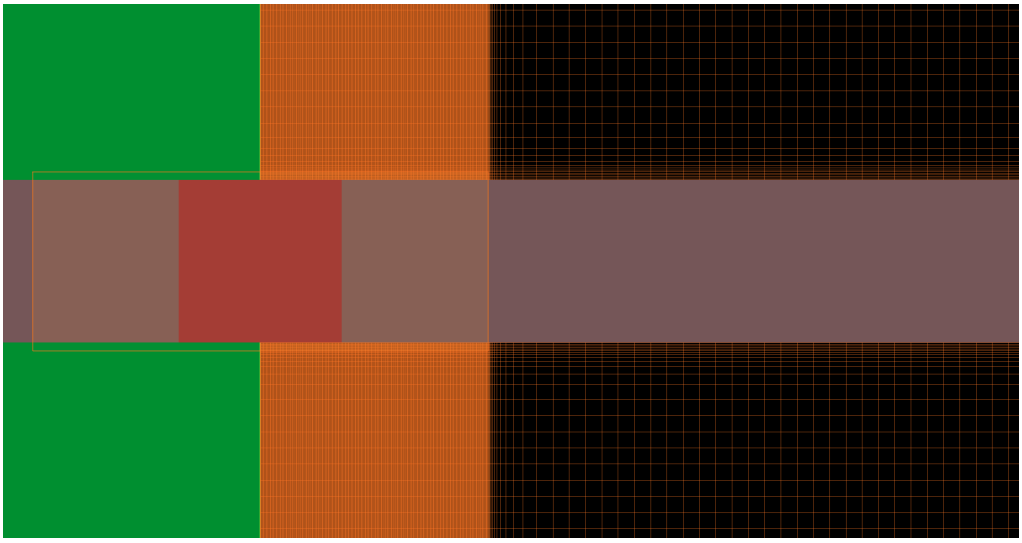


Figure 4.3: A close-up of the two dimensional cross-section of the rectangular waveguide design shown in Figure 4.2 clearly displaying the mesh size variations imposed by the added reinforcement mesh rib in the immediate vicinity of the waveguide core. The FDE solver mesh is partially hidden by the waveguide design due to the graphical rendering priorities in MODE.

The propagation of light outside the simulation region due to the evanescent field is typically assumed to occur with minimal reflections, and therefore, it is feasible to use Perfectly Matched

Layer (PML) boundary conditions. PML boundaries are constructed using an absorbing material that is impedance-matched to the surrounding materials, with the goal of minimizing reflections. Furthermore, anti-symmetric boundary conditions can be applied to the FDE solver to accelerate the simulation time. Specifically, by imposing anti-symmetric boundary conditions to the left x-boundary, the computational resources required can be reduced by a factor of two. This is because the solver can take advantage of the mirror symmetry of the waveguide structure in the x-plane, effectively reducing the number of required mesh cells by a half. Making use of symmetry boundary conditions result in a substantial improvement in computational efficiency while maintaining simulation accuracy.

4.2.4 Setting Eigensolver Analysis Parameters, Running Simulations, and Calculating Mode Properties

Before being able to run simulations, the user must provide the eigensolver with the wavelength of the propagating light and an inital set of parameters telling the solver where to look for the desired mode. Specifically, the user should give an estimate of the expected effective mode index and the amount of trial modes the solver will use when searching for a mode around this provided index. With more intricate designs it is increasingly important to give an accurate value for the expected mode index. The search range can be broadened by increasing the number of trial modes, but due to increased simulation time, this is only advisable if the resulting mode index is hard to predict or the number of modes are expected to be larger than the current number of trial modes. For this reason, only two trial modes were generally used for simulations. The low number of trial modes was made feasible by the single-mode nature of the waveguide structure and the implementation of code for dynamically updating the search index during the parameter sweeps detailed in Section 4.2.5.

The eigensolver also allows for setting the bend radius of the waveguide design, giving the user the oppurtunity to simulate the modal behaviour of curved waveguides. Having set the eigensolver parameters, the user can then simply run the simulation to start the search for possible modes. The solver will then provide a list of propagating modes, if any, and their modal properties such as effective index, propagation loss, TE/TM intensity ratio, and spatial intensity profile. For calculation of custom properties, the simulation environment includes an analysis tab that allow users to apply user-written scripts on individual modes. Calculations of the confinement factor, normalized electrical field energy density, and modal intensity fraction have been implemented in a script available in Appendix D.

4.2.5 Automating Simulation Parameter Sweeps with the Lumerical Python API

A comprehensive Python-based simulation environment was developed to facilitate systematic variation of parameters in the FDE solver, design, and Eigensolver Analysis calculations. The environment provides a user-friendly Python interface for accessing and modifying all relevant simulation parameters. The user can perform sweeps of accessible parameters in one or two dimensions by specifying a range of values and a resolution. In a one-dimensional sweep, a single parameter is varied while others remain constant. For instance, one may vary the waveguide core width from 400 nm to 500 nm in steps of 10 nm. In a two-dimensional sweep, two parameters are varied independently. It can be seen as a series of one-dimensional sweeps where a second parameter is altered for each one-dimensional sweep completion. Multiple sweeps can be defined and executed in a queue, thereby eliminating the need for additional user interaction.

Use of the parameter sweeping environment follows the ensuing steps:

1. **Configuration:** The Python environment is furnished with a configuration file, wherein the user prescribes the fundamental parameters for the FDE solver, design, and Eigensolver Analysis. The user further toggles the extraction of desired data from the sweep for subsequent visualization and analysis, and also determines the custom parameters to be computed using MODE's scripting language for each iteration. This forms the baseline for the simulation environment. The user may then define a list of one- or two-dimensional sweeps to execute. Each sweep is defined by the name of a parameter, a start point value, end point value, number of iterations, and information about the structure group the parameter belongs to in the simulation file.

2. **Execution:** With the contents of the configuration file set, the user may simply run the sweep environment to execute the list of defined sweep jobs. An instance of a MODE simulation environment will be initialized, and the script will access and set all the fundamental parameters. The design's structure group contains a setup script written in the Lumerical scripting language to dynamically transfer the implemented design parameters. This setup script, available in Appendix D, is automatically run whenever the simulation environment is updated. For each iteration of a sweep, the simulation environment is updated with the value of the varied target parameter. Then, the Eigensolver Analysis is tasked to identify any modes that may propagate in the structure. If one or more modes are found, an analysis script also written in the Lumerical Scripting Language is executed to compute custom values (such as the confinement factor) for each existing mode.

3. **Data extraction:** After the Eigensolver Analysis and subsequent analysis script has been run, the properties of the propagating modes are available to be extracted and stored by the Python environment. Parameters are extracted according to the user-defined list in the configuration file. The sweep continues once the data has been extracted. Following completion of a sweep, the data is saved locally to a unique file for further use and analysis.

4. **Visualization:** The Python environment includes functionality for opening such data files and plotting the results of sweeps in both one and two dimensions, as well as functionality for further post-processing of the results.

Experienced MODE users may question the rationale behind opting for a customized Python environment for parameter sweeps, given that MODE supports parameter sweeping to a certain extent. However, employing the parameter sweep cycle as described above offers the significant benefit of reduced memory consumption. In MODE's native sweep implementation, a new simulation environment is defined and saved on disk for each sweep iteration. As a result, memory usage increases linearly with the number of iterations and can quickly escalate to hundreds of gigabytes for concurrent, high-resolution sweeps. In contrast, the Python implementation modifies and overwrites a single simulation environment, maintaining a constant memory usage independent of the number of sweep iterations. Additionally, using a customized environment simplifies the extraction, storage, and visualization of the resulting data. Whether opting for a customized or native sweep implementation, a Python environment would need to be written to efficiently handle this data.

4.2.6 Summary of Baseline Simulation Parameters

Table 4.1 summarizes the baseline design, Eigensolver Analysis, and solver parameters used for the FDE simulations. Unless otherwise stated, it can be assumed that these parameters were used in the acquisition of simulation results.

Table 4.1: Summary of baseline simulation parameters used for FDE parameter sweeps separated by parameter type. In the "mesh"-column, FDE and "Rib" refers to the solver and the reinforcement mesh rib, respectively. The design parameters correspond to the dimensions shown in Figure 2.2.

Design	Baseline	ESA	Baseline	Mesh	Baseline
Core width	460 nm	Wavelength	3.274 µm	FDE x-span	20 µm
Silicon thickness	500 nm	# Trial modes	2	FDE y-span	12 µm
Silica thickness	3 µm	Search near	n=1.39	FDE dx	50 nm
Substrate thickness	20 µm	Bending radius	0 µm	FDE dy	50 nm
Grating width	4 µm			Rib x-span	1.40 µm
Duty cycle	15 %			Rib y-span	0.55 µm
				Rib dx	1 nm
				Rib dy	5 nm

4.3 Device Fabrication

4.3.1 Overview of the Fabrication Process

Having identified the simulated optimal design parameters in ANSYS Lumerical MODE, both straight and curved free-standing rectangular waveguides featuring several subwavelength grating duty cycles were fabricated at NTNU NanoLab. Unless otherwise specified, process parameters were selected based on previous work by UiT PhD candidate Henock Demessie Yallew [37] [38] and NanoLab process engineer Jens Høvik. Their prior work significantly reduced the need for process development, thus speeding up the device fabrication process. Testing and subsequent characterization were done, however, in order to check the continued validity of the process parameters. A rough summary of the fabrication process is iterated below. Each step will be discussed in more detail throughout the section. The fabrication process is roughly comprised of the following steps:

1. Mask design.
2. Electron beam lithography.
3. Dry etching.
4. Wet etching.
5. Characterization.

4.3.2 Lithographic Mask Design

Lithographic mask designs for both straight and curved waveguides were made and used for different batches of fabricated devices. A Python environment was written to dynamically generate

mask files through the use of Nazca Design [39], an open-source framework for photonic integrated curcuit design. The Python environment is available in Appendix E and is built on the foundation of the "spi_lib" python library developed by Marek Vlk [19] and the Ultrasound, Microwaves and Optics (UMO) group at the Department of Physics at UiT. The library provides the mathematical framework for solving the equations governing the parameterized shape and curvature of adiabatic minimum variation curves [40]. Adiabatic curves have been found to minimize transition losses [41], and are thus essential for fabrication of loss-efficient designs built around spirals and S-bends [42] [13] [43].

The written Python environment provides the user with a simple way to specify design parameters in a format similar to that found in the simulation environment, and allows the user to export arrays of waveguides in a GDS format. The environment includes functionality to make free-standing waveguide masks with paths corresponding to any parameterized curve, and can thus be used to easily implement adiabatic spirals, S-bends, and couplers without the need for further modification from the user. To facilitate easy identification and cleaving of samples, the environment has also been written to include functionality for dynamic sample labeling and placement of cleaving marks. Examples of generated masks are shown in Figure 4.4.

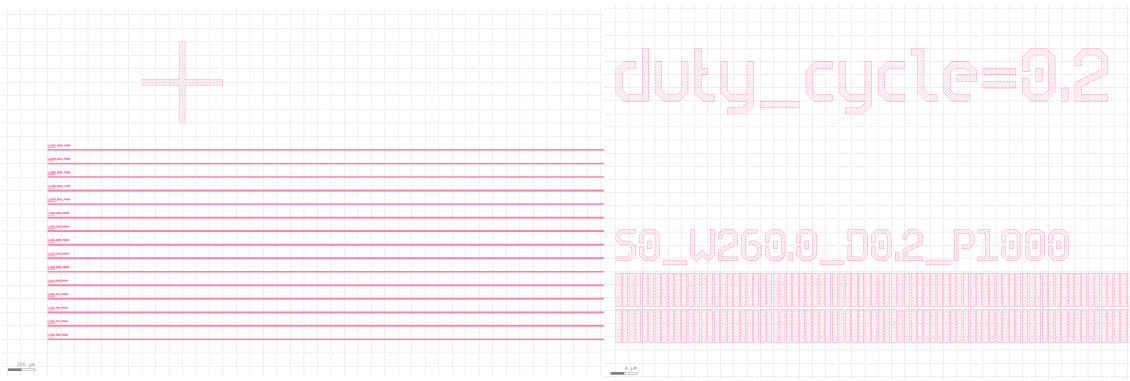


Figure 4.4: Section of an array of straight waveguide masks of various duty cycles and dimensions, complete with labels and cleaving marks, (left), and a detailed view of the subwavelength grating defining a waveguide of 20 % duty cycle (right).

4.3.3 Electron Beam Lithography

Prior to doing electron beam lithography, the generated mask designs were subjected to several post-processing steps in a software called BEAMER [44]:

1. First, through a process called *healing*, the software makes sure that no overlapping regions exist in the mask design. If such regions do exist, it joins the regions into one, thereby keeping the regions from being overexposed.

2. Secondly, the software carries out a proximity error correction (PEC) step. While not strictly necessary, PEC is of great use when dealing with intricate mask designs with small details where high resolution and consistency are desired. The PEC accounts for electrons that impart energy into unintended areas of the resist during exposure due to scattering and reflections within the material stack. This is done by modulating the dose at each exposure point, or *dot*, based on the mask geometry and a simulation of the intensity spread of electrons during exposure made in the complimentary software TRACER [45]. Such a Monte Carlo simulation using one million electrons accelerated at 100 kV was conducted on the

material stack consisting of, from top to bottom, 500 nm CSAR-62 (AR-P 6200.13) e-beam resist, 500 nm silicon, 3 µm SiO_2, and lastly a 60 mm silicon slab. It is important to clarify that the thickness of the latter is set for reasons purely related to simulation and it does not correspond to the actual thickness of the SOI wafer used for fabrication (which measures about 725 µm). The results of the simulation were then applied to the healed mask design.

3. Lastly, BEAMER exports the healed and corrected GDS file in a file format readable by, and native to, the Elionix EBL. These files contain information about the writefield dimensions, number of so-called dots, feed-pitch and scan-pitch, beam current, and exposure time per dot. The exposure time per dot is determined from the number of dots, size of the writefield, feed-pitch, scan-pitch, beam current and desired electron dose. For this project, the writefield size, number of dots, and dose was 500 µm $\times$ 500 µm, 200000, and 300 µC, respectively, while the feed-pitch and scan-pitch was set to 1. This entails a dot size of about 2.5 nm, which should provide more than sufficient dot-resolution for fabricating the free-standing waveguide structures. The beam current was set to 500 pA at an aperture of 120 µm. The relatively low beam current ensures greater exposure homogenity, and thus less sidewall roughness, than the more commonly used 1 nA and 2 nA. This comes at the cost of increased exposure times, however.

Silicon-on-insulator wafer pieces that had previously been cleaved from 8-inch wafers provided by manufacturer WaferPro along the crystallographic axis were then spin-coated with the positive e-beam resist CSAR-62 (AR-P 6200.13) [46]. Prior to being spin-coated, the wafers were cleaned for 5 minutes in acetone followed by an isopropanol (IPA) rinse, then baked at 150 °C for 5 minutes, and ultimately subjected to oxygen plasma cleaning [47] for 5 minutes at 100 %/100 % O_2/power. For the spin-coating a speed of 2750 rpm was applied for 60 s at an acceleration of 1000 rpm/s. The parameters were found through an iterative process of varying the spin speed and checking the results with a Filmetrics F20 reflectometer [48]. The process was completed when the listed parameters provided a resist thickness of 504 nm $\pm$ 3.4 nm. The usage of a relatively thick resist layer was necessary in order to protect the device layer during the subsequent ICP-RIE dry etch. The wafer with the applied resist then underwent soft-baking at 150 °C for 60 s, before commencing the exposure sequence in the Elionix ELS-G100 EBL [49].

Following exposure with the parameters detailed above, the samples are removed and developed in the AR 600-546 developer [50] for 60 s before rinsing in IPA for 60 s. To facilitate improved detachment of the resist, the development was carried out in a sonication bath at medium power. To remove any resist leftovers, the samples are then descummed by oxygen plasma cleaning for 30 s at 50 %/50 % O_2/power. It is highly important to ensure that as much as possible of the resist is removed, especially from the edges between the unexposed resist and the silicon device layer, in order to minimize transfer of roughness into the device layer during ICP-RIE dry etching.

4.3.4 Dry etching

The sample then underwent dry-etching in the Oxford Instruments Plasmalab System 100 ICP-RIE 180 [51] (ICP-RIE Chiller) at NTNU NanoLab. Prior to etching, the samples were adhered to a 4-inch silicon carrier wafer with a thin layer of Fomblin oil, a fluroinated PFPE based lubricant [52]. The oil provides both adhesion and increased thermal conductivity between the sample and the carrier wafer during etching. To prevent formation of undesired gases during the etching process, special care was taken to only coat the backside of the sample and in the process not expose the Fomblin oil to the athmosphere within the ICP-RIE process chamber. Samples were etched for a total of 105 s utilizing SF_6 and CHF_3 chemistry. The Fomblin oil was removed from the sample using acetone. Special emphasis was made on removing the oil without leaving any traces that could ptentially cause contaminate subsequent processing steps.

Following the reactive ion etching, the remainder of the protective resist layer was removed by rinsing the sample in the e-beam resist remover AR 600-71 [53] for 5 minutes in a sonication bath set to medium power. The sample was then thoroughly rinsed in IPA, followed by another oxygen plasma cleaning for 5 minutes at 100 %/100 % O_2/power to remove any remnants of resist stuck to the wafer. The absence of large amounts of leftover resist and other visible contaminants was confirmed by inspection both directly and under an optical microscope at 5×-100× magnification. Early batches where the oxygen plasma cleaning was left out revealed the presence of quite large flakes of resist on the surface, visible both in the microscope and by direct observation as irregularly refractive patches in the otherwise nicely reflective silicon.

4.3.5 Wet etching

Samples were then subjected to hydrofluoric acid (HF) underetching by submersion in a solution of 10:1 buffered oxide etchant (BOE), a mixture reported by provider Sigma-Aldrich to contain 4.4 - 4.9 % HF and 34 - 38 % of the buffering agent ammonium fluoride (NH_4F) [54]. The buffered oxide etchant removes oxides such as SiO_2, thereby etching out the portion of insulator located directly underneath the waveguide, accessed through the silicon grating. An etch duration of 66 minutes was used. Subsequently, the sample was rinsed carefully in water and air-dried. Neither the exact etch time nor the exact HF content is paramount to succesfully perform the underetch, as the operation of the structure is assumed to be insensitive to the associated lateral underetch of the device layer. However, to retain as much structural integrity as possible, it is advisable to keep overetching to a minimum.

4.3.6 Structural Characterization

The fabricated free-standing waveguides were inspected both in the Nikon Eclipse LV150 yellow light microscope [55] and the FEI APREO [56] scanning electron microscope at NTNU NanoLab. To characterize the waveguide cross section, the samples needed to be cleaved. Cleaving was done both through manual scribing and breaking at NanoLab, but also using a LatticeAx 225 [57] indent and cleaving tool with camera magnification available at the cleanroom at UiT in Tromsø. Use of the specialized tool provided much better spatial control over cleaving, which was necessary to identify the etched marker crosses shown in Figure 4.4 and achieve correct chip dimensions for the subsequent optical characterization.

4.4 Optical Characterization

A limited five-day research stay was organized to optically characterize the performance of the fabricated waveguides at the laboratory of Associate Professor Jana Jágerská at the University of Tromsø. Originally, spectroscopic measurements were planned, but as seen in Section 5.3, the signal-to-noise ratios at the waveguide outputs following in-coupling were too small to pursue further characterization of the waveguides. The optical characterization of the fabricated waveguides were therefore limited to measurements of the propagation loss, which were carried out with guidance and supervision from UiT postdoctoral research fellow Roman Zakoldaev.

4.4.1 Experimental Set Up for Propagation Loss Measurements in the Mid-Infrared

Figure 4.5 shows the setup for optical characterization of the propagation loss. The current-tuned nanoplus 3274 nm DFB laser source was regulated by a Thorlabs ITC4001 laser diode controller

[58] and aligned into an aspheric lens of fixed focal length 5.95 mm [59] with the help of a silver mirror [60] for improved control over the beam shape and direction. A Telops M350 scientific thermal infrared camera [61] with a G4x microscope lens was mounted above the output of the aspheric lens to aid in coupling the focused light into the waveguide facets, which were placed on a separate stage downstream of the aspheric lens. The aspheric lens itself was mounted on a 3-Axis NanoMax 300 Flexure Stage [62] for spatial and angular control of the beam focus. The sample stage included the option to mount a custom made gas cell around the samples. The gas cells were fitted with input and output connections for controlled gas flow during measurements. A silver mirror was mounted at an angle of 45 degrees downstream of the chip, so that the output could be observed in the IR camera. Prior to conducting the measurements, samples were annealed at a temperature of 450 °C for approximately 2 hours in an effort to remove adsorbed water molecules at the sample surface. Annealing had in previous projects been found to reduce propagation losses significantly. During the measurements, the gas cells were flushed with N_2 at a flow rate of 100 ml/min to prevent readsorbtion of water molecules on the chip surface.

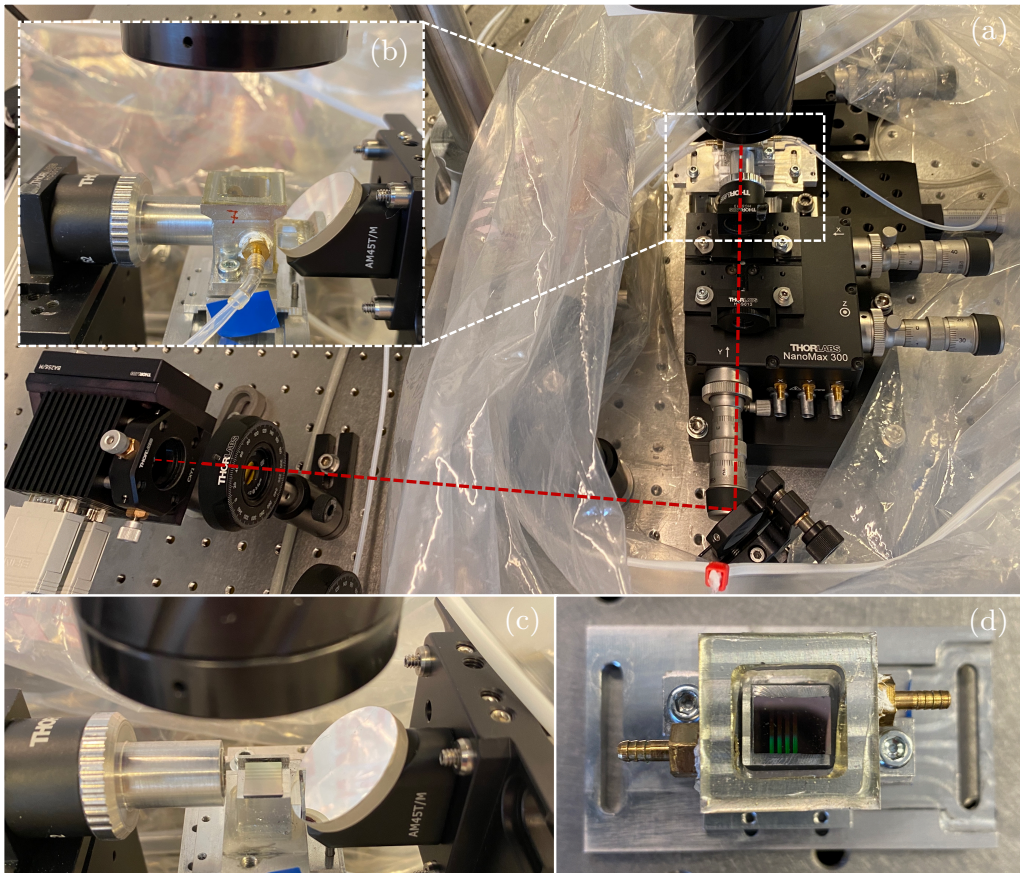


Figure 4.5: Photographs of (a) overview of the setup for optical characterization of the propagation loss with the laser beam path indicated by the red dotted line, (b) close-up of sample mounted in a gas cell with the aspheric lens on the left, silver mirror on the right, and IR camera lens at the top, (c) close-up of a mounted sample without the gas cell attached, and (d) a gas cell seen from above with a mounted sample and the gas input and output clearly visible. Each discernible structure on the sample wafer correspond to an array of 10 waveguides. The plastic bag visible in the background is used as a provisional gas cell by another experiment sharing the same setup for optical characterization.

The proprietary software Reveal IR provided by Telops was used for image aquistion of the waveguides during measurements. Images were then exported and post-processed in a Python environment. The Python environment contains code for creating composite images by stitching

together individual images, as well as a simple algorithm that determines signal attenuation from the composite images. With user-input, the program identifies pixels corresponding to the central portion of the waveguide in the propagation direction. Then, for each row of pixels perpendicular to the waveguide, the algorithm determines the total intensity signal stemming from waveguide losses. Figure 4.6 shows an example of the intensity distribution of the pixels in the vicinity of the waveguide in such a row. The waveguide is approximately 10 µm wide, which corresponds to about 3 pixels, but as the light is scattered, the observed intensity peak is a bit wider.

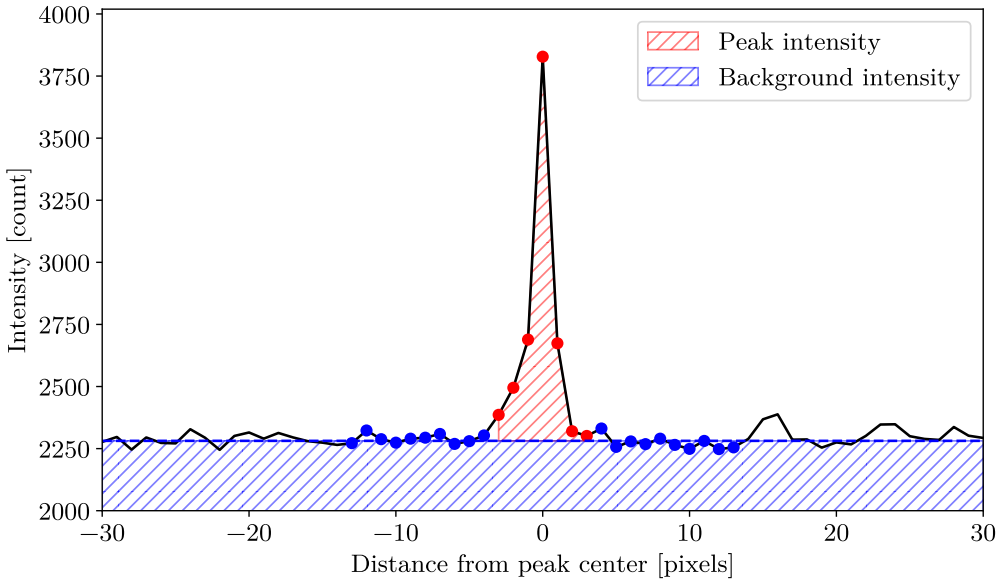


Figure 4.6: Section of the intensity distribution (black line) measured for a row of pixels at a distance of approximately 1.1 mm from the input facet of a free-standing waveguide following successful in-coupling of light. Each pixel corresponds to a 3.75 µm wide section of the chip. Values for n (red points) and m (blue points) is here set to 7 and 10, respectively. The horizontal dashed blue line represents the average background intensity calculated from the intensity values of the $2m$ blue points. The area shaded red corresponds to the calculated total intensity of the loss peak, while the area shaded blue represents the background intensity.

Determining the total peak intensity is a multi-step process. First, the pixel corresponding to the maximum of the peak is identified. Then, the central n pixels around the centre pixel is defined as belonging to the peak. The m pixels on either side of the central n pixels are defined to belong to the background. The average background intensity is then calculated as the average of the $2m$ background points. The total peak intensity is then calculated as the sum of the differences between the peak points and the average background intensity. This operation is then repeated for each row of pixels on a region of the waveguide defined by the user. The total intensity signal of each row is then transformed onto the decibel scale and a linear regression model is applied to get an estimate of the propagation loss. For the measurements presented in Section 5.3.2, values for n and m were set to 5 pixels and 5 pixels, respectively. With the G4x microscope lens mounted on the camera, each pixel corresponds to about 3.75 µm $\times$ 3.75 µm. The measurement region was generally offset about 0.5-1.0 mm from the waveguide facet along the propagation length in order to reduce the effects of reflections from the un-coupled beam on the measurement results.

5 Results and Discussion

5.1 FDE Simulation Results

This subsection will treat simulated properties of the fundamental TE mode in the presented structure as a function of different design dimensions. In addition to identifying optimal parameters for further fabrication of free-standing waveguides for sensing applications, the goal of the simulations is to provide a more general understanding of the behaviour of such waveguides at varying dimensions. The latter will provide an important background for analysis of waveguide performance following fabrication. Unless otherwise stated, parameter sweeps were run on the Lumerical MODE design presented in Section 4.2.2 with the parameters listed in Table 4.1.

5.1.1 Modal Properties as a Response to Varying Waveguide Dimensions

As the thickness of the Silicon device layer is constrained to a nominal 500 nm by the available SOI wafers, focusing on the waveguide core width was a natural starting point for parameter sweeps. Figure 5.1 shows the approximated air confinement factor for structures of duty cycle 5 % to 20 % as a function of waveguide widths in the range 250 nm to 650 nm. The simulation results reveal some clear trends. Firstly, it can be seen that decreasing the duty cycle yields an increased value and an increased sharpness of the approximated air confinement factor maximum. Secondly, decreasing the duty cycles is found to shift the waveguide width corresponding to the peak maximum towards higher values. Table 5.1 summarizes the values of the confinement factor maxima and corresponding waveguide widths for the various duty cycles.

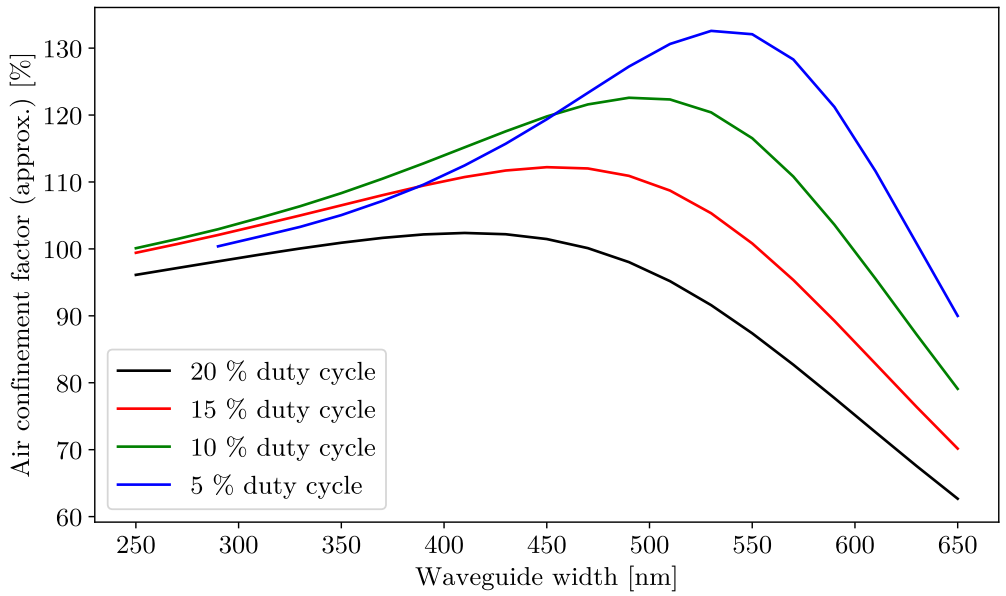


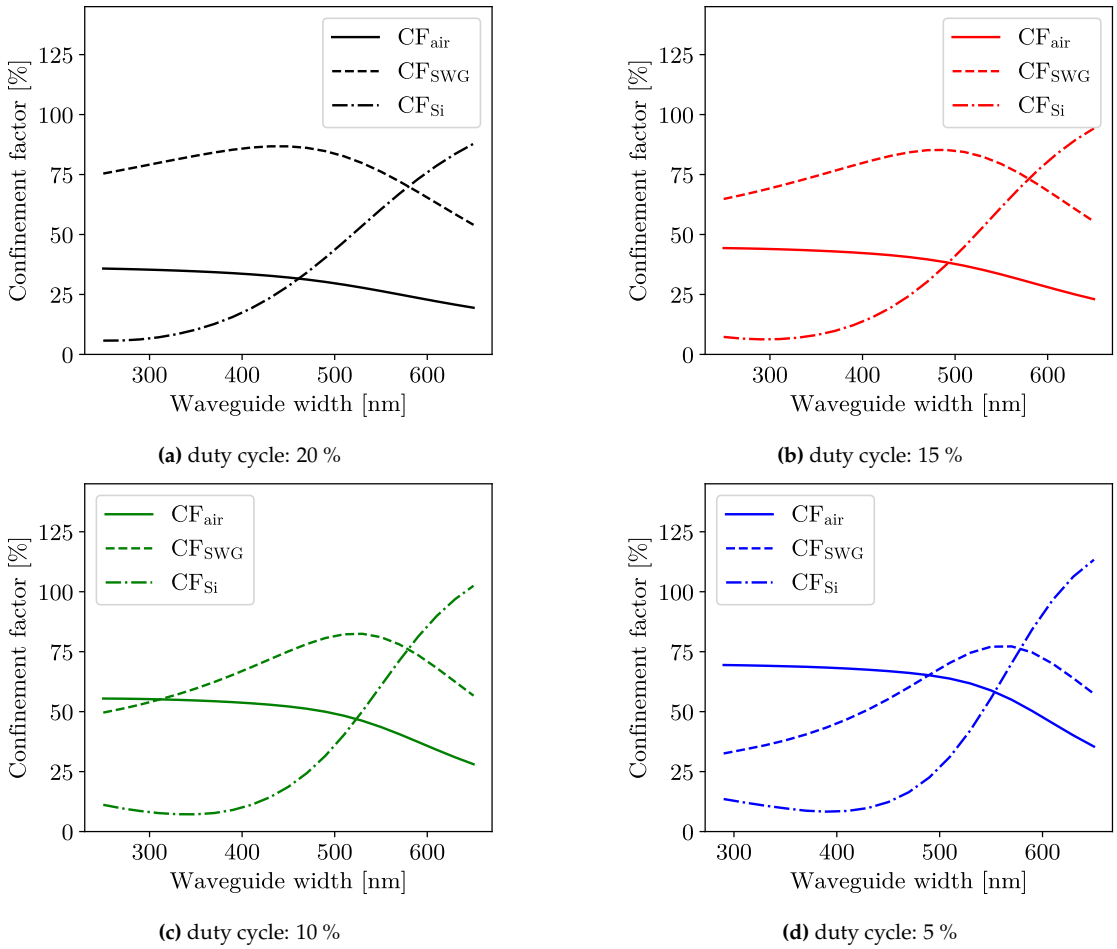
Figure 5.1: Approximated air confinement factor for structures with grating duty cycles in the range 5 % to 20% plotted as a function of the waveguide width. The approximated air confinement factor has been determined by accounting for air confinement in the subwavelength grating estimated through a linear rule-of-mixture relation.

The approximated confinement factor (Γ_{approx}) was calculated by adding a portion of the simulated confinement factor of the subwavelength grating (Γ_{SWG}) to the simulated air confinement factor

(Γ_{air}). The confinement factors of the individual waveguide constituents are shown in Figure 5.2. A simple linear rule-of-mixture relation was used to determine the fraction to be added

$$\Gamma_{approx} = \Gamma_{air} + (1 - f)\Gamma_{SWG},$$

where f denotes the grating duty cycle. A great limitation of applying 2D simulation tools towards the challenge of estimating the actual air confinement is the lack of information about the spatial intensity distribution of the mode in the grating in the direction of propagation. Three-dimensional finite-difference time domain (FDTD) simulations of freestanding slot waveguides (FSSWGs) featuring a similar grating (albeit of larger Silicon duty cycle) performed by Zhou et al. have shown the mode grating intensity profile to be inhomogenous along the propagation direction, with significantly increased intensities in the air compared to the silicon. Zhou also compared the results of their 3D findings to results obtained by the 2D FDE method. The Python sweep environment (which conveniently has been written to also support simulations of FSSWGs) was used to repeat these simulations in 2D. While the effective index of the propagating modes could be reproduced with negligble error, confirming reproducibility, the 2D simulations using the linear estimation was found to underpredict the air intensity fraction obtained from the 3D simulations by about 5 percentage points.



(a) duty cycle: 20 %

(b) duty cycle: 15 %

(c) duty cycle: 10 %

(d) duty cycle: 5 %

Figure 5.2: Calculated confinement factors as a function of the waveguide width for the waveguide constituents at duty cycles ranging from 20 % to 5 %.

Thus, the applied estimation may be slightly conservative. While a more accurate relation for the grating intensity profile would need to be determined through 3D simulations where also its

affinity to variations in duty cycle, grating period, and operating wavelength could be factored in, a simple linear relation may be sufficient for identifying optimums of design parameters and compare air confinement factors for different duty cycles.

As discussed in Section 2.1.3, the confinement factor of a gain medium is determined by the size of two competing fractions contained within Equation 2.12. The first fraction is a measure of the ratio between the mode group index and the refractive index of the medium. The second fraction denotes the normalized field energy density in the medium. By separating the fractions and analyzing how both behave as a function of the waveguide width, it is possible to explain the shape and relative magnitude of the curves shown in Figures 5.1 and 5.2. Figure 5.3 shows the response in the real effective mode index and the group index as a function of varying waveguide widths and grating duty cycles in the ranges 250 nm to 650 nm and 20 % to 5 %, respectively.

The group index was estimated from Equation 2.13 by applying the approximate derivative given in Equation 2.14 following sweeps at wavelengths of 3274 nm and 3275 nm. The results were confirmed to match those of Lumerical MODE's native frequency analysis functionality with negligble error.

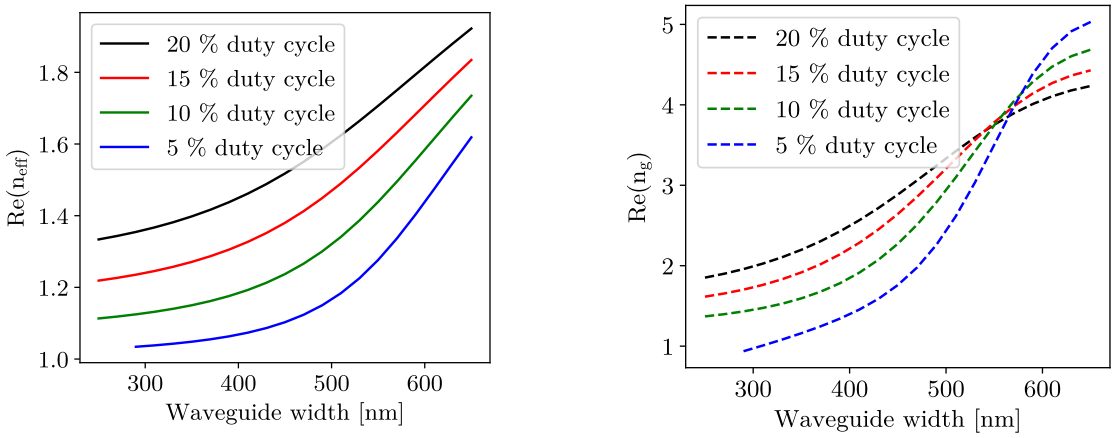
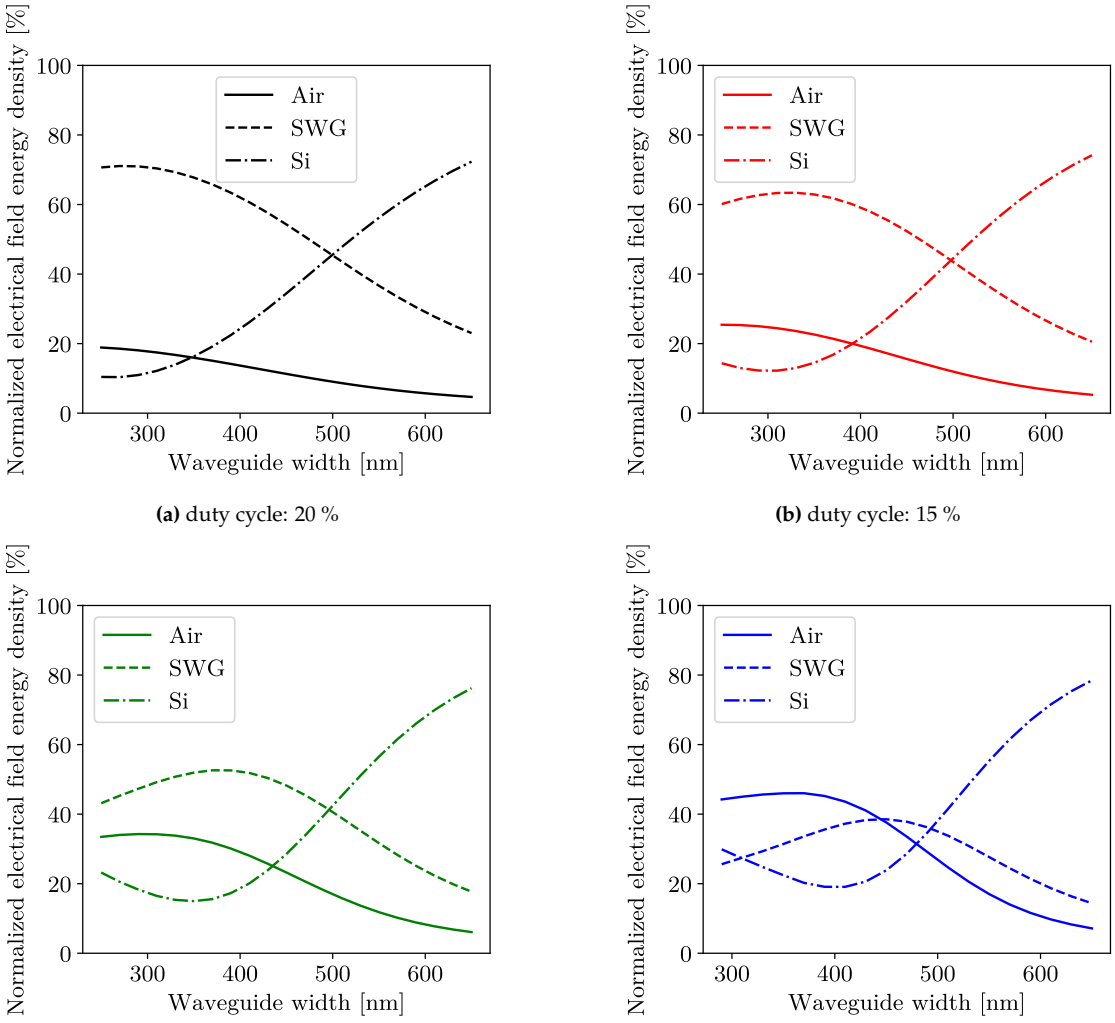


Figure 5.3: Response of the real effective mode index (left) and the group index (right) as a function of varying waveguide widths and grating duty cycles in the ranges 250 nm to 650 nm and 20 % to 5 %, respectively.

A clear trend is observed; the effective mode index is found to increase with increasing waveguide widths while it decreases as the grating duty cycle is reduced. The rate of change in the effective mode index is also significantly more pronounced in designs with lower grating duty cycles. The same behaviour is observed for the group index at waveguide widths below 550 nm. At larger waveguide widths, however, the group index of designs featuring lower duty cycles were found to overtake that of designs with higher duty cycles. This indicates that decreasing the duty cycle also decreases the interaction of the electrical field with the higher index materials of the structure (silicon and the subwavelength grating) at widths below 550 nm, but increases the field interaction at higher widths. This implies that the fundamental modes in such structures are more confined to air at lower widths than in their larger duty cycle counterparts. It also entails that such designs are more dispersive and sensitive to frequency variations.

Two aspects must also be noted in the interpretation of the simulated effective mode indices. First, accurate simulation of the effective mode index and the other properties of the propagating modes is greatly contingent on employing a good model for the effective refractive index of the subwavelength grating. The grating is implemented based on effective medium theory and Equation 2.17. The 2D simulations carried out by Zhou et al. and discussed previously in

this section revealed that 2D FDE simulations overpredicted simulated values of the effective mode index in comparison with results of 3D FDTD simulations. This is possibly linked to the inhomogenous intensity profile in the grating in the propagation direction, as an unproportional intensity distribution in air compared to in the silicon could give a lower effective refractive index compared to that of an inhomogenous medium where the mode energy is homogenously distributed. As a result, the real effective index of a structure of a certain duty cycle is probably shifted towards the value obtained for lower duty cycle structures. As will be seen in subsequent sections, this may have great implications for the loss analysis. Secondly, it should be noted that at the nominal grating period of 1000 nm, the effective mode index must be lower than 1.64 to satisfy the Bragg condition (Equation 2.16) and avoid radiative losses.



(a) duty cycle: 20 %

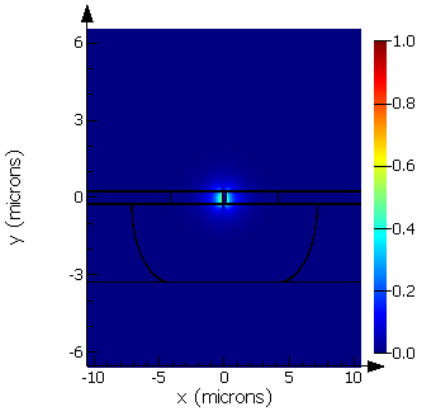
(b) duty cycle: 15 %

(c) duty cycle: 10 %

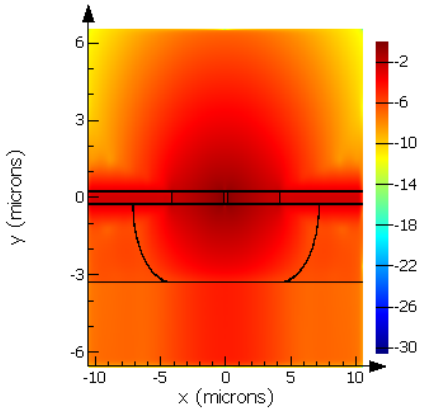
(d) duty cycle: 5 %

Figure 5.4: Simulated normalized electrical field energy density as a function of the waveguide width for the waveguide constituents at duty cycles ranging from 20 % to 5 %.

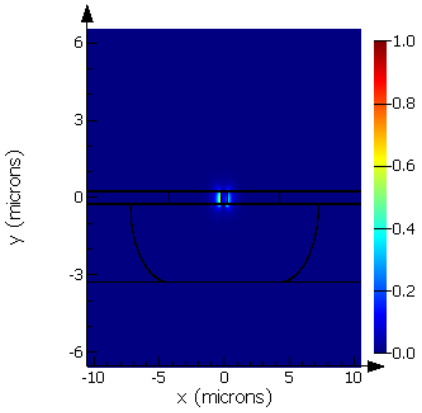
The increased air interaction of lower duty cycle waveguides that can be hypothesised from the group index results are confirmed in Figure 5.4, which shows the calculated normalized electrical field energy density of the waveguide constituents at grating duty cycles varying from 20 % to 5 %. It can be seen that the energy density in air and the grating varies greatly as a function of the duty cycle. Energy density in the silicon gradually increases as the waveguide width is increased.



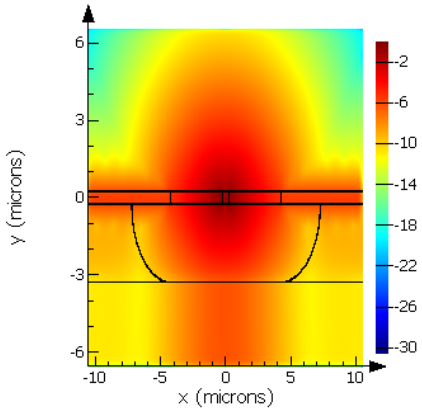
(a) duty cycle: 10 %, width: 300 nm, n_{eff}: 1.13



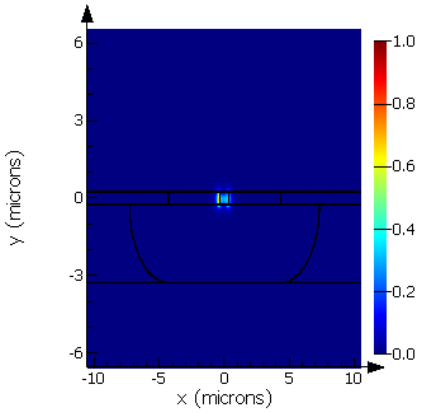
(b) duty cycle: 10 %, width: 300 nm, n_{eff}: 1.13



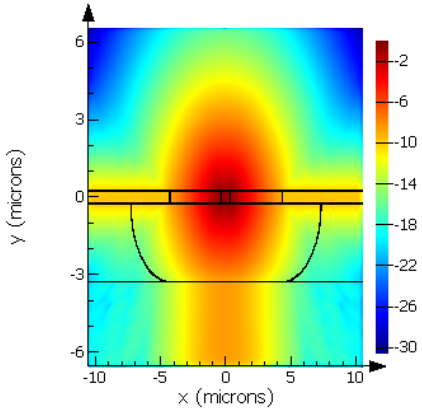
(c) duty cycle: 10 %, width: 500 nm, n_{eff}: 1.32



(d) duty cycle: 10 %, width: 500 nm, n_{eff}: 1.32



(e) duty cycle: 10 %, width: 640 nm, n_{eff}: 1.71



(f) duty cycle: 10 %, width: 640 nm, n_{eff}: 1.71

Figure 5.5: Two-dimensional cross-section of the waveguide with overlaid simulated intensity distributions as a function of waveguide width shown in linear (left) and logarithmic scale (right). Note the failure of the mode shown in the bottom row to satisfy the Bragg condition at grating periods of 1000 nm.

This is due to the mode gradually creeping into the waveguide core as it widens. As hypothesised in the discussion of the group index behaviour, the field energy density in the silicon is elevated

for lower duty cycle designs at waveguide widths above 550 nm. The increased silicon energy densities at waveguide widths around 300 nm observed especially in designs of lower duty cycles are assumed to be due to field localization in the device layer surrounding the waveguide as a result of large lateral leakages caused by poor guiding properties at low core widths. Notably, the cross-over point where the increasing silicon energy density equals the decreasing grating energy density occurs at equal widths at all duty cycles, corresponding to the point where the waveguide width equals the thickness of the device layer.

The sharper air confinement factor peaks observed in Figure 5.1 for designs of lower duty cycles are seen to be a result of the larger rate of change seen in both the group index and normalized electrical field energy densities of these designs. The observed peak shift can be explained by the initial lower group index of the lower duty cycle waveguides in conjunction with their longer (in terms of waveguide width) retention of the field energy density, presumably due to increased refractive index contrast between the silicon core and the adjacent grating.

The observed variation of the normalized electrical field energy density is supported by Figure 5.5, which shows the simulated modal spatial distribution across the cross-section of the design for the waveguide widths of 300 nm, 500 nm, and 640 nm at an intermediate duty cycle of 10 %. At 300 nm, the mode is only weakly confined to the immediate vicinity of the waveguide core, with large field distributions in the surrounding air and significant leakage into the device layer and substrate. As the width is increased to the approximated air confinement maximum at 500 nm, core confinement is still at a rather reduced level, but a larger portion of the mode intensity is confined to the vicinity of the core, with a subsequent reduction in the observed lateral and vertical leakage. From a sensing standpoint, it is understandable that this is an optimal region, as the mode is guided, but largely confined to the air medium. Increasing the waveguide further confines more of the intensity to the core at the expense of air confinement. The associated decrease in group velocity is at this width no longer sufficient to further increase the air confinement factor. Further quantification of the lateral and vertical leakage loss will be conducted in Section 5.1.2.

5.1.2 Simulated Loss

Figure 5.6a shows the total simulated propagation loss at different duty cycles as a function of the waveguide width at a grating width of 4 µm. Figure 5.6b shows a truncated view of Figure 5.6a where the total propagation losses at grating widths of 4 µm and 5 µm are compared. At lower waveguide widths the loss is found to be inpractically large, an effect that is increasingly pronounced in waveguides of decreased duty cycle. The calculated total propagation loss can be explained by the weak mode confinement and wide spatial intensity distribution of the modes as discussed in Section 5.1.1, which causes large leakage losses into the silicon substrate and device layer.

Figure 5.7a presents the lateral leakage loss as a function of the grating width for waveguides of different duty cycles. Figure 5.7b shows the vertical leakage loss as a function of the SiO_2 insulating layer thickness. The waveguide widths of the different designs correspond to their approximated air confinement factor maxima. Interestingly, even though their fundamental modes generally have narrower spatial intensity distributions, designs with larger grating duty cycles are not necessarily found to have lower leakage losses compared to their counterparts with smaller grating duty cycles. This can possibly be explained by the increased normalized electrical field energy densities in the sub-wavelength gratings of the higher duty cycle waveguides. For these structures, the larger energy density in the grating likely facilitates increased leakage through the grating and into the silicon device layer.

The lateral leakage loss was determined in a two-step process. First, the total propagation loss was obtained by varying the grating width of a waveguide design with the nominal 3 µm SiO_2

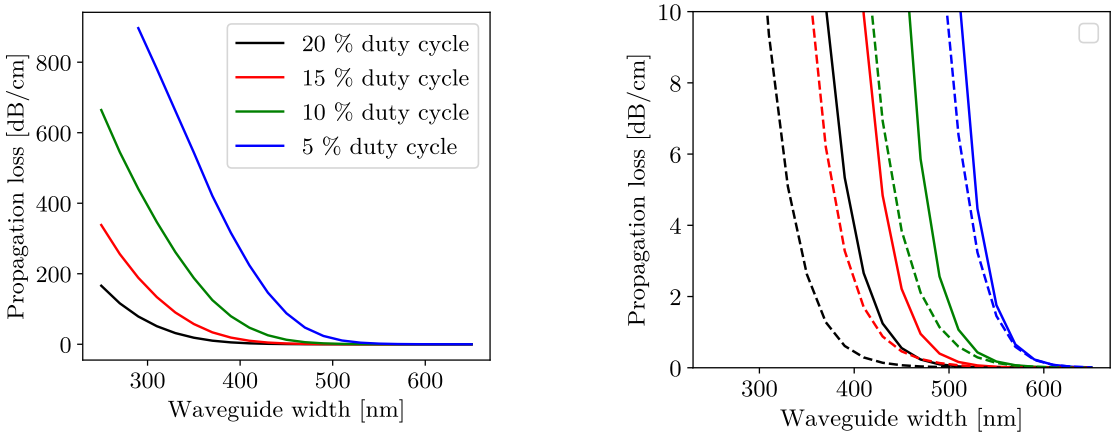


Figure 5.6: Simulated propagation loss as a function of the waveguide width at duty cycles ranging from 20 % to 5 %. Solid and dashed lines represent the performance of designs with 4 µm and 5 µm wide gratings, respectively.

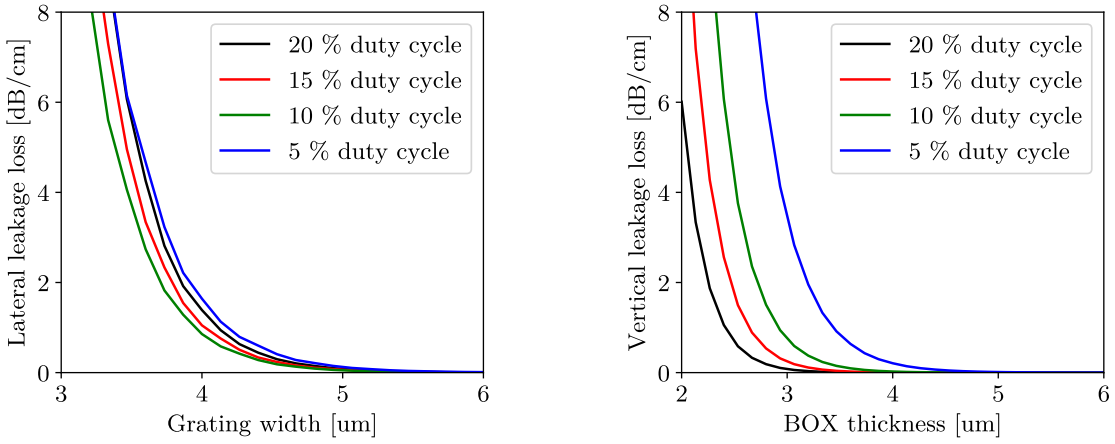


Figure 5.7: Simulated lateral leakage loss as a function of the grating width (left) and simulated vertical leakage loss (right) as a function of the SiO_2 insulator thickness at duty cycles ranging from 15 % to 5 %. The vertical leakage loss was obtained at grating widths of 8 µm. The waveguide widths of the different designs correspond to their approximated air confinement factor maximum.

insulator layer. At grating widths of about 8 µm the propagation loss converges to equal the vertical leakage loss, which was subsequentially subtracted from the total propagation loss to isolate the lateral leakage loss. The vertical leakage loss was obtained by varying the insulator thickness while the grating widths were kept at 8 µm. In this configuration, the lateral leakage losses are negligble, and the vertical leakage loss is approximately equal to the total propagation loss.

The vertical leakage loss simulations present a large challenge for fabrication of functioning waveguides. As the available SOI wafers used in fabrication have a SiO_2 insulator thickness of only 3 µm, fabricated waveguides would experience a minimum propagation loss of 0.09 dB/cm, 0.26 dB/cm, 0.77 dB/cm, and 4.41 db/cm at duty cycles of 20 %, 15 %, 10 %, and 5 %, respectively, not accounting for lateral leakage losses and scattering losses. An oversight during early work

on the project was the lack of a proper analysis of the lateral leakage losses. Therefore, as shown in Table 5.1, free-standing waveguides were fabricated with grating widths of 4 µm and 5 µm, which is clearly not optimal with respect to lateral leakage losses. The simulations show that the grating width should at a minimum be greater than 6 µm to ensure negligble lateral losses at waveguide widths corresponding to the confinement factor peaks of the different designs. At the time, there was a reluctance to increase the grating width further than 5 µm due to concerns for the structural integrity of the grating. However, characterization of the fabricated waveguides covered in Section 5.2.1 shows the larger duty cycle waveguides to exhibit good mechanical stability at grating widths of 5 µm, and increasing the grating width to above 6 µm for future fabrication is not thought to be a significant problem.

End-fire coupling losses were simulated using a Gaussian beam of waist diameter 7.5 µm focused directly at the end facet of the waveguide. Figure 5.8 shows a comparison between the intensity profile of a propagating mode in a waveguide of duty cycle 15 % and the Gaussian beam intensity at the waist. It can be seen that the effective mode area is much smaller than the spatial beam distribution at the waist, which results in quite large coupling losses of around 4-6 dB. Table 5.1 presents the coupling losses for waveguides of different duty cycles and waveguide widths. It can be seen that the lower duty cycle waveguides have smaller associated coupling losses. This can be explained by two factors. Firstly (and most importantly), as seen from Figure 5.4, modes in these waveguides are less confined to the waveguide core, resulting in an increased effective mode area. Secondly, the waveguide width corresponding to the air confinement factor optimum is larger for waveguides of lower duty cycles, thus increasing the area of the waveguide facet. An important focus for future work should be put on the development and implementation of suitable coupling structures to further reduce coupling losses.

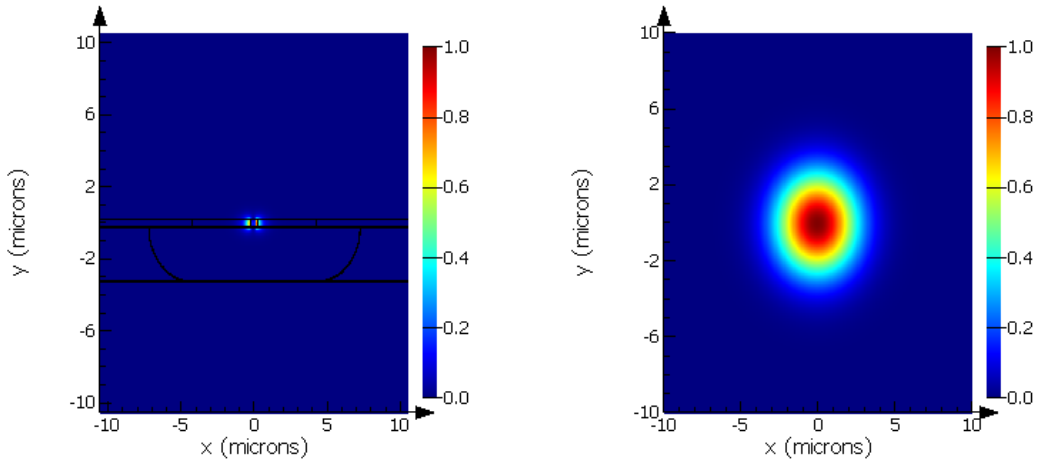


Figure 5.8: Two-dimensional cross-section of the waveguide with overlaid simulated mode intensity distribution in a waveguide of duty cycle 15 % (left) compared to the intensity profile of a Gaussian beam of 7.5 µm waist width centered on the waveguide core (right).

5.1.3 Summary of Simulated Properties for Fabricated Designs

Table 5.1 summarizes the simulated properties of designs that were fabricated. Note the distinction between the set duty cycle and the expected duty cycle. The latter is calculated from bias measurements presented in Section 5.2.1 and reflects the deviation from the set duty cycle that occurs during fabrication. As an example of insights gained during the iterative process described

in Section 4.1, the waveguides have been fabricated using the dimensions corresponding to the confinement factor optimum for the expected duty cycle, not the true duty cycle.

Table 5.1: Summary of set duty cycle (f_{set}), expected duty cycle (f_{exp}) estimated from bias measurements, waveguide width (W_{core}), grating width (W_g), approximated air confinement factor (Γ_{air}), simulated lateral leakage loss (L_{LL}), simulated vertical leakage loss (L_{VL}), simulated total propagation loss (L_{prop}), and simulated coupling loss (L_C), for the different fabricated designs. The expected duty cycle has been estimated from bias measurements detailed in Section 5.2.1. The waveguides have been fabricated using the dimensions corresponding to the confinement factor optimum for the expected duty cycle.

f_{set} (%)	f_{exp} (%)	W_{core} (nm)	t_{Si} (nm)	W_g (µm)	Γ_{air} (%)	L_{LL} (dB/cm)	L_{VL} (dB/cm)	L_{prop} (dB/cm)	L_C (dB)
25*	20	420	500	4	102	1.74	0.09	1.83	6.10
25*	20	420	500	5	102	0.11	0.09	0.20	6.10
20	15	460	500	4	112	1.19	0.26	1.45	5.67
20	15	460	500	5	112	0.05	0.26	0.33	5.67
15	10	500	500	4	123	0.89	0.77	1.66	5.14
15	10	500	500	5	123	0.05	0.77	0.82	5.14
12	7	515	500	4	129	1.32	1.95	3.27	4.55
12*	7	515	500	5	129	0.10	1.95	2.05	4.55
10	5	520	500	4	132	2.53	4.41	6.94	3.95
10*	5	520	500	5	132	0.22	4.41	4.63	3.95

*These designs were not fabricated. Included for summary and comparison of results.

5.2 Device Fabrication

5.2.1 Characterization of Fabricated Waveguides

Through thorough examination utilizing both optical microscopy and scanning electron microscopy (SEM), the majority of the fabricated free-standing waveguides were found to be successfully fabricated. Specifically, waveguides with set duty cycles of 20 % and 15 % were observed to be intact, and the grating structures appeared homogenous and well-formed. However, it was observed that the gratings surrounding the free-standing waveguides with set duty cycles of 10 % had collapsed during the fabrication process, presumably due to lack of structural integrity following the HF wet etching step. Consequently, in subsequent batches, waveguides with duty cycles of 12 % were incorporated as replacements, and upon inspection, they were found to be standing without any signs of structural compromise. Figure 5.9 shows waveguides of set duty cycles of 20 % and 15 % following underetching and subsequent cleaving.

Analysis of the waveguide facets following cleaving demonstrated that the actual waveguide widths of the fabricated waveguides were, on average, 65.5 ± 11.9 nm narrower than their intended nominal values. To address this discrepancy, a decision was made to introduce a bias of 60 nm towards larger waveguide widths in all subsequent mask designs prior to the fabrication process. Experimental measurements conducted on waveguides incorporating the bias during fabrication revealed an approximate increase of 10 nm in width compared to the designated value. Therefore, a more accurate bias for future fabrication was determined to be about 50 nm. Considering the nominal grating period of 1000 nm, this implies that the true duty cycle of the fabricated gratings is approximately 5 percentage points lower than their designated values. This deviation was accounted for in subsequent fabrication by designing waveguides of set duty cycle x % with

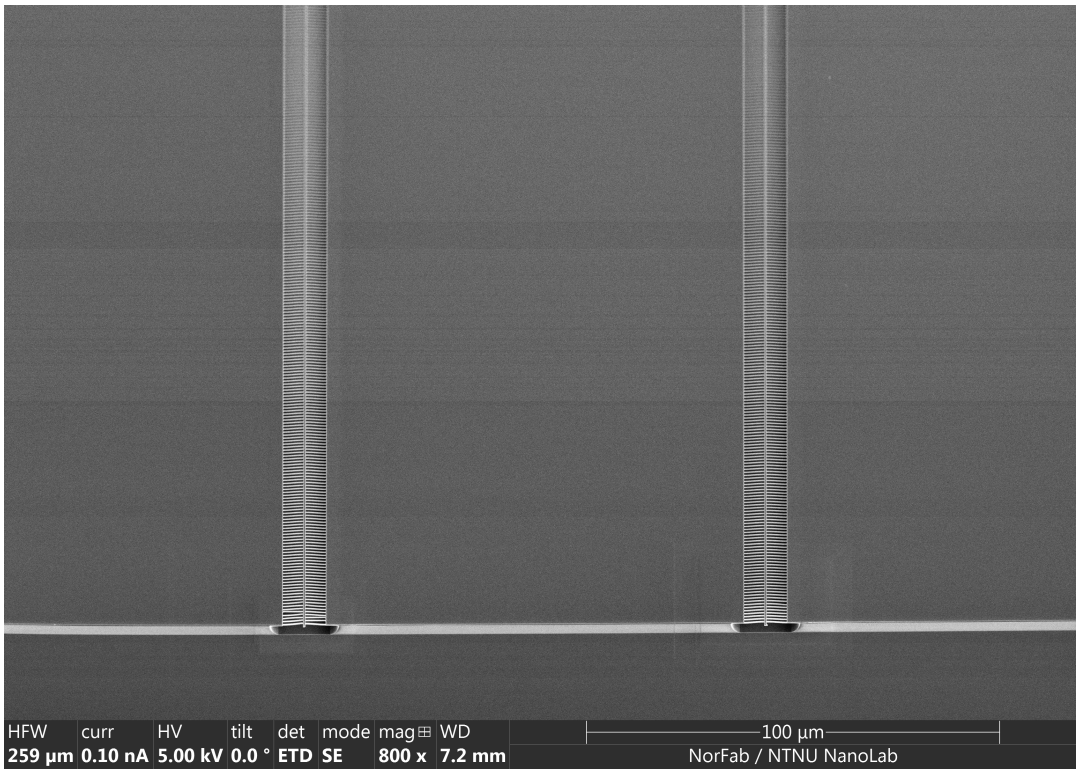


Figure 5.9: SEM micrograph overview of two adjacent free-standing waveguides of set duty cycles and 20 % (left) and 15 % (right). Apart from some damage to the grating near the end facets caused by stresses during wafer cleaving, the waveguides are found to be intact, mechanically stable, and well-formed over significant lengths.

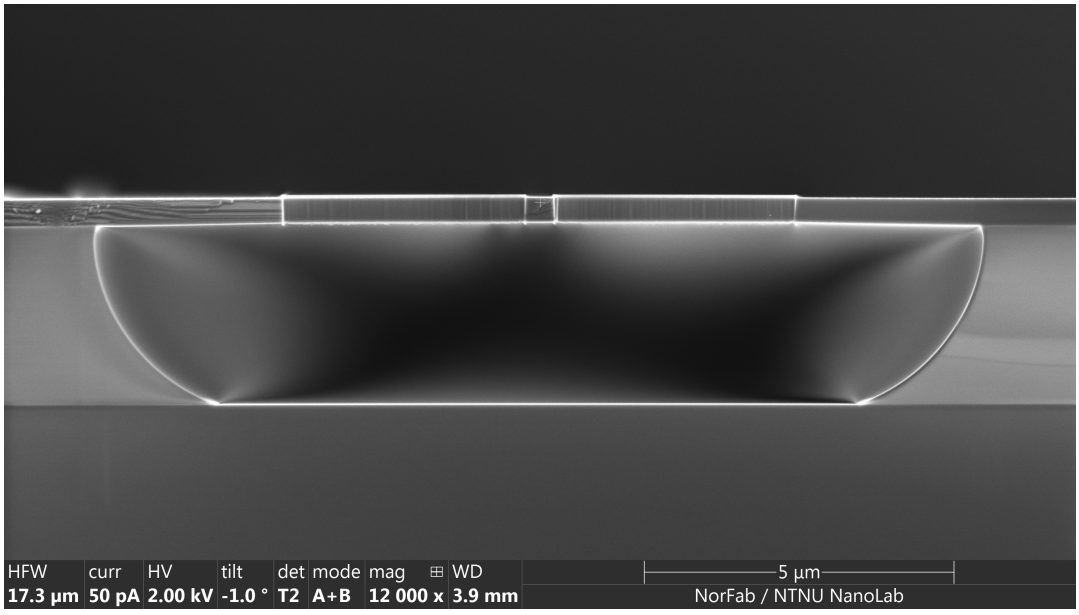


Figure 5.10: SEM micrograph captured following wafer cleaving displaying the cross section of a free-standing waveguide of set duty cycle 15 % with a nominal waveguide width of 500 nm.

the optimal waveguide width for duty cycle (x-5) %. Figure 5.10 shows the cross section of a well-cleaved waveguide with a set duty cycle of 15 %. Wafer cleaving was in general found to be a large challenge, with significant structural integrity problems appearing as a consequence of the relatively large imposed stresses on the rather fragile grating structures. Cleaving is treated in greater detail in Appendix B.3.

The figure shows that the subwavelength gratings feature roughness in the form of vertical lines on the sidewalls of the grating tethers. Similar lines are expected to also appear on the sidewalls of the rectangular waveguide core, and are likely a consequence of a combination of two effects. Firstly, some fine lines will always be present in the sidewalls due to iregularity in the applied electron dose during the electron beam lithography. Dose iregularities will give rise to roughness in the resist layer, which will be transferred down into the silicon device layer during ICP-RIE dry etching. Secondly, rougher lines may be transferred into the silicon device layer if small spots of resist are left behind along the edge between the unexposed resist layer and silicon device layer following development. To counteract this, great care was taken in cleaning the sample following development, both in isopropanol and by descumming in the oxygen plasma cleaner. Some sample batches suffered from extreme surface roughness, however, which were likely a consequence of contamination from an unknown substance or set of substances during the underetching step of the fabrication. This surface roughness was found to degrade the structural integrity of the affected waveguides, rendering them inoperable and likely increasedly prone to large scattering losses. Appendix B.1 covers the effects of the surface roughness in more detail.

Figure 5.11 shows an overview of fabricated waveguides with varying duty cycles and grating widths. While the presented designs are all succesfully free-standing, it should be noted that, unsurprisingly, the lower duty cycle waveguides show signs of buckling grating tethers. Additionally, the lower duty cycle gratings are seen to suffer from reduced in-plane stiffness, which can be observed as horizontal displacements of the tethers. Lower duty cycle waveguides likely also suffer from some droop, which can be seen as the grating on one side of the rectangular core being fully under tension, while the opposing grating is buckling. The associated decrease in distance to the substrate may result in increased vertical leakage losses.

Measurements of the waveguide dimensions following fabrication revealed, somewhat unexpectedly, a significant reduction in the thickness of the device layer from the nominal 500 nm to 410 nm. While the exact cause of the reduction remains unclear, it can be speculated that the thickness reduction may be a result of slow oxidation of silicon to silica which is subsequently removed by the buffered oxide etchant during the wet etching step. The thickness may also have been reduced as a consequence of earlier fabrication steps. It is is also possible, but perhaps not plausible, that the thickness of device layer in the provided SOI wafers deviated from the reported 500 nm $\pm$ 20 nm even at delivery. Understanding of the underlying cause necessitates further methodological investigation.

In conjunction with the 10 nm dicrepancy from nominal width caused by the slightly large bias of 60 nm, the significant thickness reduciton observed in the device layer has a huge effect on the simulated properties of the waveguides. Table 5.2 shows a revised version of Table 5.1 updated to reflect the measured waveguide dimensions of the fabricated designs. It can be seen that the loss increases significantly as a result of the reduced thickness of the device layer, explained by reduced mode confinement in the waveguide core. This is supported by a substantial reduction in the effective mode index for all waveguides. The approximated air confinement factor remains relatively stable, albeit with a slight decrease, as the normalized electrical field energy density increases in the air and grating as the effective mode index, and also the closely related group index, falls. Although the area of the end facet is reduced, reduced confinement of the mode increases the effective mode area. This can in turn explain the observed decrease in coupling loss.

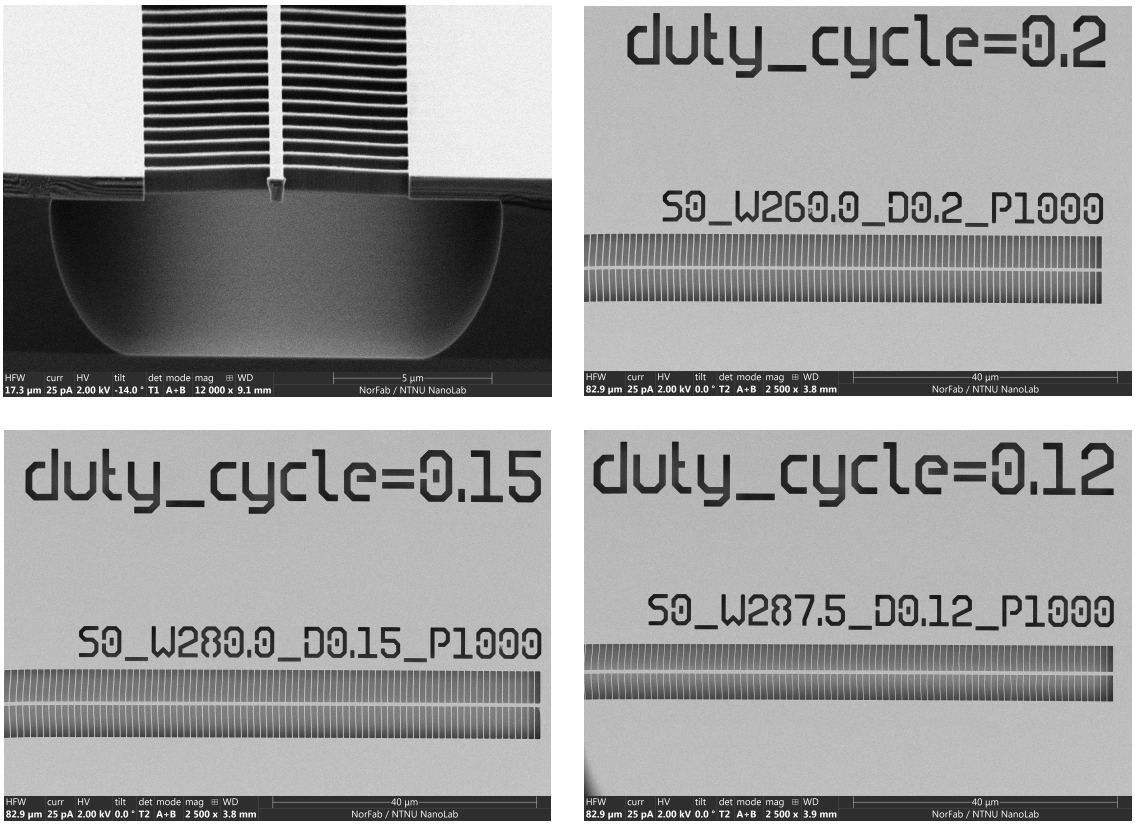


Figure 5.11: SEM micrographs of fabricated free-standing waveguides of varying duty cycles and grating widths. The top left image provides an overview of the cross section of a waveguide of set duty cycle 20 %, while the remaining images show structures of set duty cycles 20 %, 15 %, and 12 %, with grating widths of 5 µm, 5 µm, and 4 µm, respectively. The letters of the etched labeling correspond to the slot size (S), half of the set waveguide width including the bias (W), the duty cycle (D), and the period (P) in nanometers. The reader should note that the top left micrograph was captured at a rather large tilt angle, which results in the vertical axis not appearing to scale.

Table 5.2: Summary of set duty cycle (f_{set}), expected duty cycle (f_{exp}) estimated from bias measurements, measured waveguide width (W_{core}), measured device layer thickness (t_{Si}), grating width (W_g), approximated air confinement factor (Γ_{air}), simulated lateral leakage loss (L_{LL}), simulated vertical leakage loss (L_{VL}), simulated total propagation loss (L_{prop}), and simulated coupling loss (L_C), for the different fabricated designs. The expected duty cycle has been estimated from bias measurements detailed in Section 5.2.1 and confirmed by inspection in SEM.

f_{set} (%)	f_{exp} (%)	W_{core} (nm)	t_{Si} (nm)	W_g (µm)	Γ_{air} (%)	L_{LL} (dB/cm)	L_{VL} (dB/cm)	L_{prop} (dB)	L_C (dB)
20	15	470	410	4	112	5.67	1.21	6.88	9.75
20	15	470	410	5	112	0.52	1.21	1.73	9.75
15	10	510	410	4	121	5.31	3.41	8.72	9.07
15	10	510	410	5	121	0.53	3.41	3.94	9.07
12	7	525	410	4	124	7.89	7.71	15.6	8.30

5.3 Optical Characterization

5.3.1 Observed Modal Properties

Composite images shown in Figure 5.12 display successful in-coupling of light into several of the fabricated free-standing waveguides, visible by clear scattering losses along the lengths of the structures. The scattering losses are periodic in nature with an approximate periodicity of 25 µm separating the observed intensity peaks. These are likely caused by periodic grating inhomogenities due to irregularities in the applied dose at EBL subwritefield boundaries. Appendix B.2 presents a complete coverage of this effect and the measures that have been applied to reduce its consequences on the fabricated samples. Also visible in measurements of some waveguides are periodic intensity peaks thought to possibly stem from stitching errors and dose variations and the boundaries of the larger 500 µm $\times$ 500 µm EBL writefields.

In-coupling was generally found to be difficult, in large part due to the structural challenges commonly associated with cleaving that are treated in Appendix B.3. Identification of suitable waveguides in an optical microscope prior to attempted in-coupling was found to greatly ease the work. Still, in-coupling could be problematic to achieve both due to the large expected coupling losses caused by the relative wide beam waist of the incident Gaussian laser beam, and due to local geometric variations of the cleaved chip surface.

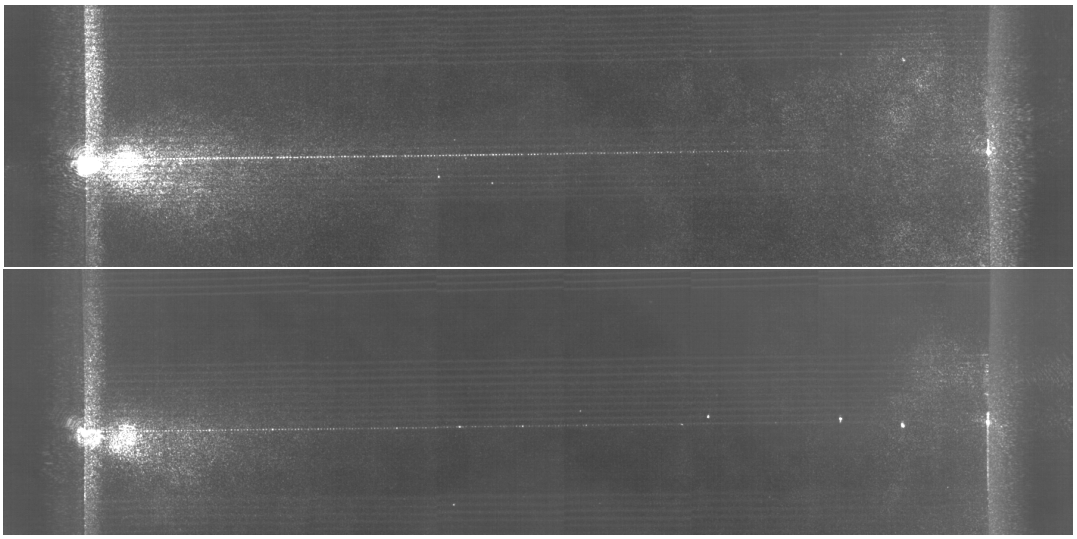


Figure 5.12: Composite images of in-coupled light into free-standing waveguides of set duty cycles 15 % and 5 µm grating width (top), and 12 % and 4 µm (bottom), over a distance of approximately 7.5 mm. The waveguides are clearly visible due to large scattering losses along their propagation length. Out-coupled light from the free-standing waveguides could not be discerned from the output of other modes at the end facet of the chip.

Out-coupled light from the waveguide could not directly be discerned from the background light, even at relatively short propagating lengths of 7.5 mm, which is likely a cause of the large total losses (propagation loss + coupling loss) predicted from the simulations in Section 5.1.2 and the presence of undesired, high-intensity modes propagating in what is effectively planar slab waveguides formed by the silicon device layer on either side of the free-standing waveguides. The output from such planar waveguides can, in addition to background light propagating through the silicon substrate, be seen to make up most of the intensity observed at the output. Thus, it was not possible to isolate the signal from the free-standing waveguidee and perform

spectroscopic measurements. Furthermore, separating the output of the free-standing waveguides from the output of other modes are furthermore made difficult by the much smaller area of the end waveguide facet to those of the planar waveguides, and would require a much higher signal to (what is effectively) noise ratio.

The introduction of an optical polarization filter into the laser path allowed the polarization of the indicident light to be toggled between TE and TM. As expected, clear guiding was observed in TE polarization, while rotating the filter to exclusively allow TM polarized light extinguished the scattering signal. A similar operation was carried out while observing reflections of light originating from the end facet of the chip in the slanted mirror placed downstream of the output facet. Rotating the polarizer to allow TM polarized light extinguished a central portion of the observed output signal, alluding to the presence of successfully outcoupled light from the waveguide, even though it could not be directly discerned. Oscillating cross-coupling between adjacent free-standing waveguides was also observed in some cases.

5.3.2 Propagation Loss Measurements

Figure 5.13 presents the results of propagation loss measurements performed on a set of waveguides by applying the algorithm described in Section 4.3.6. The resultant attenuation was subjected to post-processing in the form of averaging over lengths of about 30 µm for visualization purposes. To better illustrate the total propagation loss, the measured attenuation and the applied linear regression line was also shifted to intercept the plot origin.

Table 5.3 compares the measured propagation losses to the simulated propagation losses obtained in Section 5.2.1. First of all, it can be seen that the measured propagation loss is generally larger than what was simulated in Section 5.1.2 and Section 5.2.1. This may be explained by several factors not accounted for in the simulation environment:

- The presence of periodic inhomogenities in the grating caused by dose variations at EBL writefield and subwritefield boundaries are likely the cause of increased scattering losses. Periodic losses are clearly visible in the images presented in Figure 5.12. Fabricated waveguides will also generally suffer from scattering due to imperfections and sidewall roughness that is challenging to account for in simulations.
- As was briefly discussed in Section 5.1.1, the two-dimensional FDE simulations in all probability overpredict the effective mode index of the simulated designs, meaning that the modes propagating in the physical waveguides are plausibly less confined to the waveguide core than that which is predicted from the simulations. It follows from the discussion in Section 5.1.2 that small reductions in core confinement could rapidly give rise to large increases in lateral and vertical leakage loss. The effect is somewhat alleviated by the wider than intended fabricated waveguide cores, but likely plays a large role in increasing the total propagation loss. Droop of the waveguide core following reductions in structural integrity at low duty cycles bringing the waveguide closer to the substrate may also contribute to increasing vertical leakage loss.
- While the constituent materials of the waveguide design, silicon and silica, remain transparent at the laser operating wavelength of 3274 nm as described in Section 2.1.5, adsorbtion of water molecules to the waveguide core and grating have previously been hypothesised to be a huge contributor to absorption losses and thus increased propagation losses. While annealing was carried out prior to measurements, unsuccesful removal of absorbing molecules or their readsorbtion may have contributed to increasing losses.

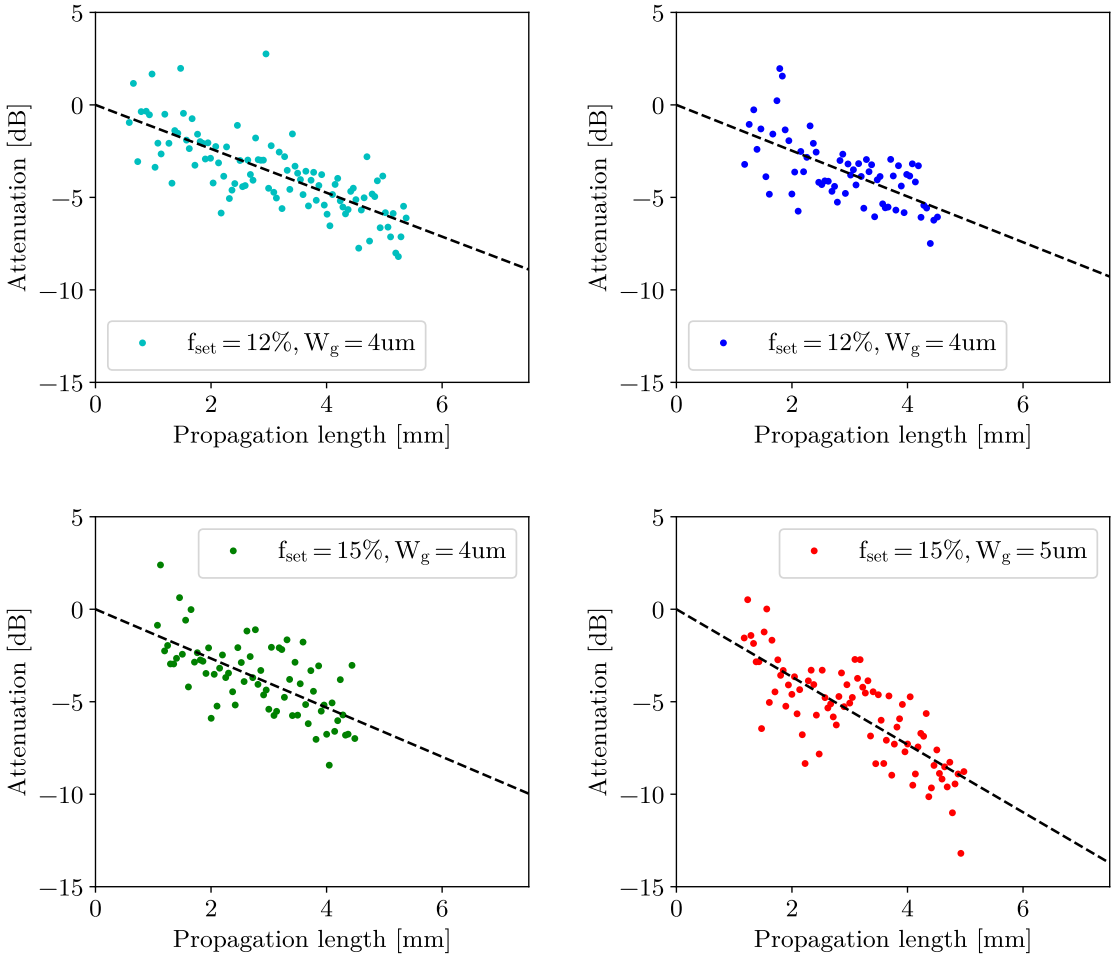


Figure 5.13: Measured attenuation as a function of propagation length for waveguides of set duty cycles 12 % and grating widths 4 µm (top left), 12 % and 4 µm (top right), 15 % and 4 µm (bottom left), and 15 % and 5 µm (bottom right). Their attenuation losses were measured to 11.9 dB/cm, 12.4 dB/cm, 13.3 dB/cm, and 18.3 dB/cm, respectively.

Secondly, no apparent pattern is seen when comparing the loss measured for waveguides of different duty cycles. From the loss simulations in Section 5.1.2, it follows that the losses are expected to be larger in structures of lower duty cycles and grating widths. However, the opposite trend is observed for the measured propagation losses. This effect may in part be related to light absorption in embedded and adsorbed water molecules and -OH groups. As is seen in Figure 5.2, the light-matter interaction is stronger in the silicon core for designs of larger duty cycles, which may contribute to increasing the light-water interaction and the associated absorption losses.

At the same time it should be addressed that the low number of measured samples may be inadequate to form a statistically significant basis for comparison of the samples. Moreover, in addition to the standard error for the measurements of the propagation loss, there are also expected to be uncertainties related to the method of obtaining the propagation loss it self that is challenging to quantify. Hence, with the limited number of measurements, it is not deemed worthwhile to put too much emphasis on neither the exact value nor the comparison of the propagation loss results for the different waveguides, but rather observe the general trend of larger than simulated propagation losses.

Table 5.3: Summary of set duty cycle (f_{set}), expected duty cycle (f_{exp}) estimated from bias measurements, measured waveguide width (W_{core}), measured device layer thickness (t_{Si}), grating width (W_g), total simulated propagation loss ($L_{prop-sim}$), and measured propagation loss ($L_{prop-meas}$), for the different fabricated designs. The expected duty cycle has been estimated from bias measurements detailed in Section 5.2.1.

f_{set} (%)	f_{exp} (%)	W_{core} (nm)	t_{Si} (nm)	W_g (µm)	$L_{prop-sim}$ (dB/cm)	$L_{prop-meas}$ (dB)
20	15	470	410	4	6.88	n.m**
20	15	470	410	5	1.73	n.m**
15	10	510	410	4	8.72	13.3±1.54
15	10	510	410	5	3.94	18.3±1.29
12	7	525	410	4	15.6	12.2±1.22*

*Average of two measured values (11.9 ± 0.86 dB/cm and 12.4 ± 1.57 dB/cm)
**Values for these waveguides have not been measured.

5.4 Suggestions for Future Work

The results discussed in this section has provided a good amount of insights that can be leveraged in continued work and research. A concise list of suggested measures for improvements and additions to the current design is enumerated below:

- The challenge of lateral leakage losses should be addressed by increasing the width of the sub-wavelength grating. From the loss analysis in Section 5.1.2 it is clear that the grating should be expanded to extend for at least 6 µm. To account for lower than expected field confinement due to deviating fabrication dimensions and discrepancies between simulated and real behaviour, an added safety margin of 2 - 3 µm is suggested. Increasing the grating widths should be feasible for structures of higher duty cycles (15 %), but the associated reduction in mechanical stability may pose a challenge for structures of lower grating duty cycles. Reduction of lateral leakage losses will also go a long way towards eliminating the observed cross-coupling. Additionally, it will increase the separation between the core of the free-standing waveguide and the planar waveguides, which should help separate the desired signal from noise.
- The current structures are characterized by mechanical instability in the lateral direction for designs with low duty cycles. To increase mechanical stability, it could be interesting to draw inspiration from the field of photonic crystals to further vary the grating metamaterial. For example, the grating could be replaced by a periodic lattice of etched holes or otherwise interconnected tethers, or the grating tethers could feature a tapered design that is narrower close to the waveguide core (where the field intensity is the strongest) and wider at the connection point with the device layer. The latter configuration could be designed to keep the field air interaction found at lower duty cycles while significantly increasing mechanical stability.
- To further investigate discrepancies between observed and simulated waveguide properties, a natural continuation of the simulation work would be to carry out three-dimensional finite-difference time-domain simulations using the ANSYS Lumerical FDTD solver. Expanding the simulation space from two to three dimensions carries with it a large increase in simulation times, making wide paramater sweeps less feasible. To increase efficiency, the Python environment designed for 2D FDE batch simulations could be employed to identify designs of interest, trends, and patterns, while 3D FDTD simulations could be run on a smaller selection of chosen designs.

- To address the large coupling losses associated with end-fire coupling, specialized couplers should be further researched and implemented. The Python enviroment for lithographic mask design is written to accommodate a large variety of couplers with only small changes required to the code.
- Future work can continue this projects work on fabricating spiral and bent waveguides of spatially offset input and output. While curved waveguides not only yield the increased optical path lengths on limited sample surface areas that are needed for analyte interaction comparable with available TDLAS instrumentation, offsetting the input and output will also serve to reduce the background noise in optical measurements. The Python environment for lithographic mask design includes functionality for fabricating spiral and curved waveguides of arbitrary dimensions. While spiral waveguides were fabricated during the course of this project, measurements on such structures were limited by the surface roughness challenges described in Appendix B.1 that rendered the structures inoperable.

6 Conclusion

Finite-difference eigenmode simulations were carried out on cross-sections of the free-standing waveguide designs in order to identify optimal parameters for further fabrication of sub-wavelength grating supported waveguides for sensing applications, as well as to obtain a general understanding of the effects governing the behaviour of such structures. Parameter sweeps were made for structures of varying geometric dimensions. The magnitude of the approximated air confinement factor was found to be highly sensitive to variations of the waveguide core dimensions and the air duty cycle of the sub-wavelength grating that acts as a cladding to the core. Optimums for the approximated air confinement factor were found to emerge due to a competing relation between the localization of the normalized field energy density in the air, and the increase of the mode group index with larger localizations in constitutents of high refractive index. The magnitude of the optimums were found to increase with lower duty cycles, presumably due to the associated increased effective refractive index contrast between the silicon core and the sub-wavelength grating decreasing the core confinement of the propagating mode. At the optimums, free-standing waveguides of duty cycles 20 %, 15 %, 10 %, and 7 % were found to feature propagating modes yielding approximated air confinement factors of 102 %, 112 %, 123 %, and 129 %, respectively.

Analysis of loss simulations revealed these propagating modes to be susceptible to significant lateral and vertical leakage losses due to their low core confinement and their wider evanescent field distributions. The effect was found to be increasingly pronounced for waveguides of low duty cycles where core confinement were reduced. Simulations found that the lateral leakage losses could be mitigated by increasing the width of the grating cladding, but reduction of vertical leakage losses were limited by the insulator thickness of the SOI wafers available for fabrication. Additionally, overlap analysis assuming end-fire coupling between the waveguide facets and a Gaussian beam of beam waist diameter 7.5 µm revealed the presence of large coupling losses of at least 4 dB.

With a basis in preliminary simulation results, several batches of free-standing waveguides were successfully fabricated with measured duty cycles of 15 %, 10 %, and 7 % and grating widths of 4 µm and 5 µm. The structures were observed to be intact, and the grating structures generally appeared homogenous and well-formed. Some challenges were encountered however, such as periodic structural deviations and surface roughness, imposed by varying dose intensities at the boundaries of EBL subwritefields and what is hypothesised to be unknown contaminants, respectively. Free-standing waveguides of lower duty cycles were observed to be slightly susceptible to moderate lateral displacement of the grating tethers and possibly droop of the waveguide core.

Optical characterisation of the fabricated free-standing waveguides were carried out at the University of Tromsø. Originally, spectroscopic measurements were planned, but the signal-to-noise ratios at the waveguide outputs following in-coupling were too small to pursue further characterization of the waveguides. Optical characterization of the fabricated waveguides were therefore limited to measurements of the propagation loss. In general, the propagation loss was measured to be significantly higher than suggested by the simulated loss results, likely caused by a combination of factors pertaining to deviations from the set design paramaters following fabrication, periodic scattering losses at locations corresponding to EBL writefield and subwritefields, roughness-induced scattering losses, and uncertainties in the approximations and assumptions required for two-dimensional FDE simulations.

To address the large propagation and coupling losses, which are believed to be the limiting factors for the practical operation of the waveguides, a number of suggestions for future continuation of the work have been presented. While the operation of lower duty cycle waveguide designs will be limited in practice by the vertical losses of the relatively thin inuslator layer, successfully fabricating free-standing waveguides of above-unity air confinement factor should be readily possible by targeting structures of duty cycles in the 15 % – 20 % range.

References

[1] Arthur N. Cox. *Allen's Astrophysical Quantities*. AIP Press, 2000, pp. 258–259.

[2] Anna Agustí-Panareda et al. "Technical note: The CAMS greenhouse gas reanalysis from 2003 to 2020". In: *Atmospheric Chemistry and Physics* 23 (6 Mar. 2023), pp. 3829–3859. ISSN: 1680-7324. DOI: 10.5194/acp-23-3829-2023.

[3] Alexander Tøsdal Tveit et al. "Metabolic and trophic interactions modulate methane production by Arctic peat microbiota in response to warming". In: *Proceedings of the National Academy of Sciences* 112 (19 May 2015). ISSN: 0027-8424. DOI: 10.1073/pnas.1420797112.

[4] Alexander Tøsdal Tveit et al. "Simultaneous Oxidation of Atmospheric Methane, Carbon Monoxide and Hydrogen for Bacterial Growth". In: *Microorganisms* 9 (1 Jan. 2021), p. 153. ISSN: 2076-2607. DOI: 10.3390/microorganisms9010153.

[5] Achuthanunni Chokkathukalam et al. "Stable isotope-labeling studies in metabolomics: new insights into structure and dynamics of metabolic networks". In: *Bioanalysis* 6 (4 Feb. 2014), pp. 511–524. ISSN: 1757-6180. DOI: 10.4155/bio.13.348.

[6] Peng Wang et al. "A 13C isotope labeling method for the measurement of lignin metabolic flux in Arabidopsis stems". In: *Plant Methods* 14 (1 Dec. 2018), p. 51. ISSN: 1746-4811. DOI: 10.1186/s13007-018-0318-3.

[7] Min Yuan et al. "Ex vivo and in vivo stable isotope labelling of central carbon metabolism and related pathways with analysis by LC–MS/MS". In: *Nature Protocols* 14 (2 Feb. 2019), pp. 313–330. ISSN: 1754-2189. DOI: 10.1038/s41596-018-0102-x.

[8] Jane Hodgkinson and Ralph P Tatam. "Optical gas sensing: a review". In: *Measurement Science and Technology* 24 (1 Jan. 2013), p. 012004. ISSN: 0957-0233. DOI: 10.1088/0957-0233/24/1/012004.

[9] J. Barry McManus. "Application of quantum cascade lasers to high-precision atmospheric trace gas measurements". In: *Optical Engineering* 49 (11 Nov. 2010), pp. 111–124. ISSN: 0091-3286. DOI: 10.1117/1.3498782.

[10] Jared F. Bauters et al. "Planar waveguides with less than 01 dB/m propagation loss fabricated with wafer bonding". In: *Optics Express* 19 (24 Nov. 2011), p. 24090. ISSN: 1094-4087. DOI: 10.1364/OE.19.024090.

[11] Jared F. Bauters et al. "Ultra-low-loss high-aspect-ratio Si3N4 waveguides". In: *Optics Express* 19 (4 Feb. 2011), p. 3163. ISSN: 1094-4087. DOI: 10.1364/OE.19.003163.

[12] L. Tombez et al. "Methane absorption spectroscopy on a silicon photonic chip". In: *Optica* 4 (11 Nov. 2017), p. 1322. ISSN: 2334-2536. DOI: 10.1364/OPTICA.4.001322.

[13] Marek Vlk et al. "Extraordinary evanescent field confinement waveguide sensor for mid-infrared trace gas spectroscopy". In: *Light: Science and Applications* 10 (1 Jan. 2021), p. 26. ISSN: 2047-7538. DOI: 10.1038/s41377-021-00470-4.

[14] Wen Zhou et al. "Fully suspended slot waveguide platform". In: *Journal of Applied Physics* 123 (6 Feb. 2018), p. 063103. ISSN: 0021-8979. DOI: 10.1063/1.5017780.

[15] Derek M. Kita et al. "Are slot and sub-wavelength grating waveguides better than strip waveguides for sensing?" In: *Optica* 5 (9 Sept. 2018), p. 1046. ISSN: 2334-2536. DOI: 10.1364/OPTICA.5.001046.

[16] J. Soler Penades et al. "Suspended silicon mid-infrared waveguide devices with subwavelength grating metamaterial cladding". In: *Optics Express* 24 (20 Oct. 2016), p. 22908. ISSN: 1094-4087. DOI: 10.1364/OE.24.022908.

[17] Weixin Liu et al. "Larger-Than-Unity External Optical Field Confinement Enabled by Metamaterial-Assisted Comb Waveguide for Ultrasensitive Long-Wave Infrared Gas Spectroscopy". In: *Nano Letters* 22 (15 Aug. 2022), pp. 6112–6120. ISSN: 1530-6984. DOI: 10.1021/acs.nanolett.2c01198.

[18] Bahaa E. A. Saleh and Malvin Carl Teich. *Fundamentals of Photonics*. John Wiley and Sons, Inc., Aug. 1991. ISBN: 0471839655. DOI: 10.1002/0471213748.

[19] Marek Vlk. "Optical Waveguides for Infrared Spectroscopic Detection of Molecular Gases". UiT The Arctic University of Norway, June 2021. URL: https://hdl.handle.net/10037/21327.

[20] Jacob T. Robinson et al. "First-principle derivation of gain in high-index-contrast waveguides". In: *Optics Express* 16 (21 Oct. 2008), p. 16659. ISSN: 1094-4087. DOI: 10.1364/OE.16.016659.

[21] T.D. Visser et al. "Confinement factors and gain in optical amplifiers". In: *IEEE Journal of Quantum Electronics* 33 (10 1997), pp. 1763–1766. ISSN: 00189197. DOI: 10.1109/3.631280.

[22] Richard Soref. "Mid-infrared photonics in silicon and germanium". In: *Nature Photonics* 4 (8 Aug. 2010), pp. 495–497. ISSN: 1749-4885. DOI: 10.1038/nphoton.2010.171.

[23] Hongtao Lin et al. "Mid-infrared integrated photonics on silicon: a perspective". In: *Nanophotonics* 7 (2 Dec. 2017), pp. 393–420. ISSN: 2192-8614. DOI: 10.1515/nanoph-2017-0085.

[24] Harry D. Downing and Dudley Williams. "Optical constants of water in the infrared". In: *Journal of Geophysical Research* 80 (12 Apr. 1975), pp. 1656–1661. ISSN: 01480227. DOI: 10.1029/JC080i012p01656.

[25] John D. Joannopoulos et al. *Photonic Crystals: Molding the Flow of Light (Second Edition)*. 2nd ed. Princeton University Press, 2008. ISBN: 0691124566.

[26] Lu Liu, Qingzhong Deng, and Zhiping Zhou. "Subwavelength-grating-assisted broadband polarization-independent directional coupler". In: *Optics Letters* 41 (7 Apr. 2016), p. 1648. ISSN: 0146-9592. DOI: 10.1364/OL.41.001648.

[27] S. M. Rytov. "Electromagnetic Properties of a Finely Stratified Medium". In: *Soviet Journal of Experimental and Theoretical Physics* 2 (3 Nov. 1955), pp. 605–616.

[28] Zhaoming Zhu and Thomas Brown. "Full-vectorial finite-difference analysis of microstructured optical fibers". In: *Optics Express* 10 (17 Aug. 2002), p. 853. ISSN: 1094-4087. DOI: 10.1364/OE.10.000853.

[29] Ansys Optics. *MODE - Finite Difference Eigenmode (FDE) solver introduction*. URL: https://optics.ansys.com/hc/en-us/articles/360034917233-MODE-Finite-Difference-Eigenmode-FDE-solver-introduction#:~:text=The%20Finite%2DDifference%20Eigenmode%20(FDE,%2C%20effective%20index%2C%20and%20loss..

[30] Erwin Kreyszig. *Advanced engineering Mathematics, 10th Edition*. Wiley, 2020.

[31] Guozhong Cao and Ying Wang. *Nanostructures and Nanomaterials*. Vol. 2. WORLD SCIENTIFIC, Jan. 2011. ISBN: 978-981-4322-50-8. DOI: 10.1142/7885.

[32] G. K. Celler and Sorin Cristoloveanu. "Frontiers of silicon-on-insulator". In: *Journal of Applied Physics* 93 (9 May 2003), pp. 4955–4978. ISSN: 0021-8979. DOI: 10.1063/1.1558223.

[33] Katsutoshi Izumi. "Historical overview of SIMOX". In: *Vacuum* 42 (5-6 Jan. 1991), pp. 333–340. ISSN: 0042207X. DOI: 10.1016/0042-207X(91)90050-S.

[34] Wayne M. Moreau. *Semiconductor lithography: principles, practices, and materials*. Plenum, 1988.

[35] Ansys inc. *Ansys Lumerical MODE*. URL: https://www.ansys.com/products/photonics/mode.

[36] Edward D. Palik. *Handbook of Optical Constants of Solids*. 1st ed. Academic Press, June 1985.

[37] Henock Demessie Yallew et al. "Silicon-on-Insulator (SOI) Slot Waveguide for Methane Absorption Sensing in the Mid-Infrared". In: Optica Publishing Group, 2022, EM1D.2. ISBN: 978-1-957171-10-4. DOI: 10.1364/ES.2022.EM1D.2.

[38] Henock Demessie Yallew, Jana Jágerská, and Martin M. Greve. "Long, stitch-free slot waveguide with s-bend tapered couplers for IR-sensing applications using electron beam lithography". In: *Journal of Vacuum Science and Technology B* 41 (1 Jan. 2023), p. 012601. ISSN: 2166-2746. DOI: 10.1116/6.0002187.

[39] BRIGHT Photonics. *Nazca Design – Photonic IC Design Framework*. URL: https://nazca-design.org/.

[40] Raphael Linus Levien. "From Spiral to Spline: Optimal Techniques in Interactive Curve Design". PhD thesis. EECS Department, University of California, Berkeley, 2009. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-162.html.

[41] Tong Chen et al. "A general design algorithm for low optical loss adiabatic connections in waveguides". In: *Optics Express* 20 (20 Sept. 2012), p. 22819. ISSN: 1094-4087. DOI: 10.1364/OE.20.022819.

[42] Hansuek Lee et al. "Ultra-low-loss optical delay line on a silicon chip". In: *Nature Communications* 3 (1 May 2012), p. 867. ISSN: 2041-1723. DOI: 10.1038/ncomms1876.

[43] Marek Vlk et al. "Free-standing tantalum pentoxide waveguides for gas sensing in the mid-infrared". In: *Optical Materials Express* 11 (9 Sept. 2021), p. 3111. ISSN: 2159-3930. DOI: 10.1364/OME.430994.

[44] GenISys GmbH. *BEAMER Electron- and Laser-Beam Lithography Software*. URL: https://www.genisys-gmbh.com/beamer.html.

[45] GenISys GmbH. *TRACER Monte Carlo Simulation Software*. URL: https://www.genisys-gmbh.com/tracer.html.

[46] Allresist GmbH. *E-Beam Resist AR-P 6200 series (CSAR 62)*. URL: https://www.allresist.com/portfolio-item/e-beam-resist-ar-p-6200-series-csar-62/.

[47] Diener Electronic GmbH. *FEMTO Low Pressure Plasma Systems (Plasma cleaner)*. URL: https://www.plasma.com/en/low-pressureplasma-femto/.

[48] Filmetrics. *F20 Series Film Thickness Measurement Instruments*. URL: https://www.filmetrics.com/thickness-measurement/f20.

[49] STS Elionix. *ELS G-100 HIGH PERFORMANCE 100 KV LITHOGRAPHY SYSTEM*. URL: https://sts-elionix.com/product/els-g100/.

[50] Allresist GmbH. *Developer AR 600-456*. URL: https://www.allresist.com/portfolio-item/developer-ar-600-546/.

[51] Oxford Instruments. *Inductively Coupled Plasma Etching (ICP RIE)*. URL: https://plasma.oxinst.com/technology/icp-etching.

[52] Solvay. *Fomblin® PFPE Lubricants*. URL: https://www.solvay.com/en/brands/fomblin-pfpe-lubricants.

[53] Allresist GmbH. *Remover AR 600-71*. URL: https://www.allresist.com/portfolio-item/remover-ar-600-71/.

[54] Sigma-Aldrich. *Buffered oxide etchant (BOE) 10:1*. URL: https://www.sigmaaldrich.com/NO/en/product/aldrich/901621.

[55] Nikon Metrology NV. *Nikon Eclipse LV150 yellow light microscope*. URL: https://industry.nikon.com/en-gb/products/industrial-microscopy/industrial-microscopes/upright-eclipse-lv150na-lv150n/.

[56] NorFab. *FEI APREO SEM*. URL: https://ntnu.norfab.no/WebForms/Equipment/EquipmentView.aspx?toolId=96.

[57] LatticeGear. *LatticeAx 225 Accurate Indent and Cleaving System*. URL: https://www.latticegear.com/store/latticeax-225.html.

[58] ThorLabs Inc. *ITC4001 - Benchtop Laser Diode/TEC Controller*. URL: https://www.thorlabs.com/thorproduct.cfm?partnumber=ITC4001.

[59] ThorLabs Inc. *C028TME-E - f = 5.95 mm, NA = 0.56, Mounted Geltech Aspheric Lens, ARC: 3 - 5 µm*. URL: https://www.thorlabs.com/thorproduct.cfm?partnumber=C028TME-E.

[60] ThorLabs Inc. "PF10-03-P01 - Ø1" Protected Silver Mirror". In: (). URL: https://www.thorlabs.com/thorproduct.cfm?partnumber=PF10-03-P01.

[61] Telops. *M350 Scientific Thermal Infrared Camera*. URL: https://www.telops.com/wp-content/uploads/2021/06/2021-fast-m350-brochure.pdf.

[62] ThorLabs Inc. *3-Axis NanoMax Flexure Stages*. URL: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2386.

A Determining Optimal Solver Parameters Through Convergence Testing

A.1 Convergence in Modal Properties

The accuracy and usefulness of FDE simulations is contingent on using the correct solver parameters. Employing a too coarse mesh, or spanning over too little of a simulation region, may lead to inaccurate and varying results as the solver algorithms struggle to find a stable – or *converging* – solution to the governing set of equations. On the other hand, increasing the solver resolution far past the point of convergence very quickly ramps up simulation time while at best only providing marginal benefits to accuracy. Convergence testing can be described as the process of identifying the simulation resolution and spatial extent that is needed to be able to reduce errors from solver instability to an acceptable level while also keeping simulation time at a minimum.

The Python environment for automated parameter sweeping described in Section 4.2.5 was used to vary the mesh dimensions and span of the FDE solver in both the x and y directions. All simulation results presented in this appendix were calculated at a set 5 % duty cycle and a grating width of 4 µm, as out of all configurations treated in Section 5.1, this was expected to have the greatest sensitivity to loss variations. An iterative approach was used to avoid varying four parameters simultaneously. The x- and y-span were first fixed to rather large values of 20 µm and 12 µm, respectively. Then the x- and y-dimensions of the mesh were gradually varied from very coarse to fine through a two dimensional parameter sweep. Figure A.2 shows the percentage difference between the effective mode index (n_{eff}), propagation loss, and air confinement factor (CF_{air}) at the diagonal ($dx = dy$) and the values of the properties at a mesh size of 15 nm × 15 nm.

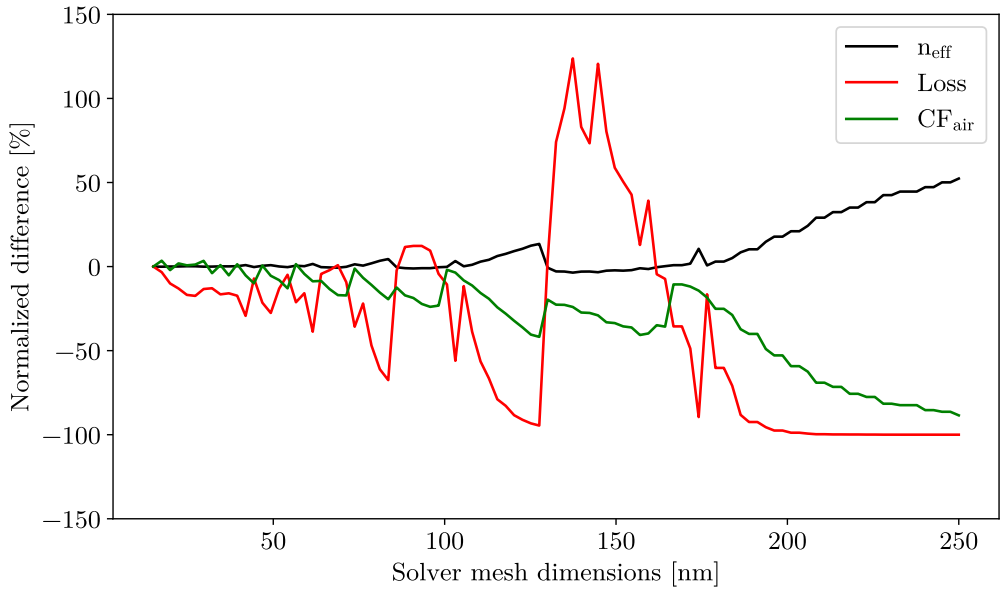


Figure A.1: Differences in effective mode index (n_{eff}), propagation loss, and air confinement factor (CF_{air}) as a function of the solver mesh size normalized to their values at a mesh size of 15 nm × 15 nm.

As is evident, the effective mode index stabilizes at a relatively large mesh size of about 75 nm × 75 nm. The propagation loss and air confinement factor, however, fail to adequately converge at the test limit of 15 nm × 15 nm. As further mesh reductions of the entire simulation region starts becoming unpractical due to fast increasing computational times, a 1.4 µm × 0.55 µm reinforcement mesh rib was added around the waveguide core. As most of the mode intensity is

spatially distributed within the bounds of the reinforcement mesh rib, it was deemed adequate to keep the unaffected solver mesh at a rather lenient 50 nm × 50 nm. Figure A.2 shows the percentage difference between the effective mode index (n_{eff}), propagation loss, and air confinement factor (CF_{air}) at the diagonal ($dx = dy$) and the values of the properties at a reinforcement mesh size of 2 × 2 nm. The loss adequately converges at reinforcement meshes finer than 10 nm × 10 nm, while the air confinement factor remains seemingly unwilling to converge. The latter observation is the result of an interesting effect occuring on the material boundaries and is treated in greater detail in Appendix A.2. Here it is concluded that a reinforcement mesh size of 1 nm × 5 nm should be adequate to achieve convergence also in the confinement factor.

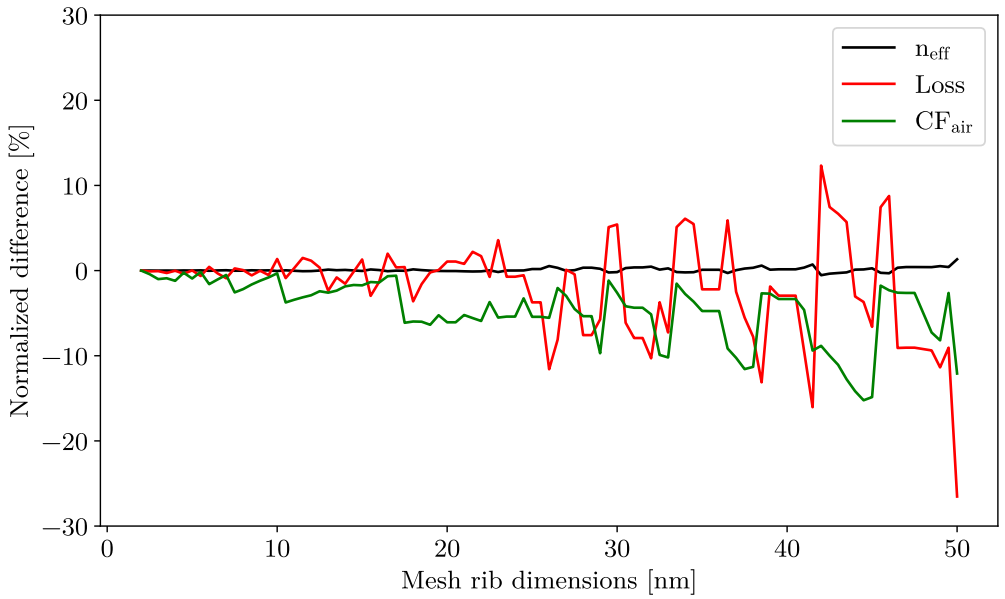


Figure A.2: Differences in effective mode index (n_{eff}), propagation loss, and air confinement factor (CF_{air}) as a function of the mesh rib mesh size normalized to their values at a mesh size of 2 × 2 nm. The reinforcement mesh rib has a x- and y-span of 1.4 µm and 0.55 µm, respectively, while the remaining solver mesh has a fixed mesh size of 50 × 50 nm.

Having determined the minimum dimensions of the solver mesh and reinforcement mesh, a similar process was repeated for the x- and y-span of the FDE solver. First, a sweep varying the x-span was carried out with a fixed y-span of 12 µm. Then, the x-span was kept at µm while the y-span was varied. Figure A.3 shows the percentage difference between the effective mode index (n_{eff}), propagation loss, and air confinement factor (CF_{air}) and the values of the properties at a solver span of 20 µm × 20 µm for the sweeps over the solver x-span and y-span.

It can be seen in the figure that the effective mode index and air confinement factor remain practically unchanged during the sweeps, while the loss is found to converge at solver spans larger than 10 µm × 10 µm. However, for the simulation results detailed in Section 5.1, it was decided to use a nominal span of 20 µm × 12 µm. This was done not only to have a margin of error, but also to comfortably encompass the entire design including the substrate and buried oxide layer. A slight periodic variation of constant amplitude can be seen in the loss also at solver spans greater than 10 µm × 10 µm. Increasing the span further was not found to attenuate the variation significantly. Due to the small size of the variations, an explanation for the root cause of the effect was not pursued further, but it should be noted as a potential contributor to uncertainty in the loss results presented in Section 5.1.2.

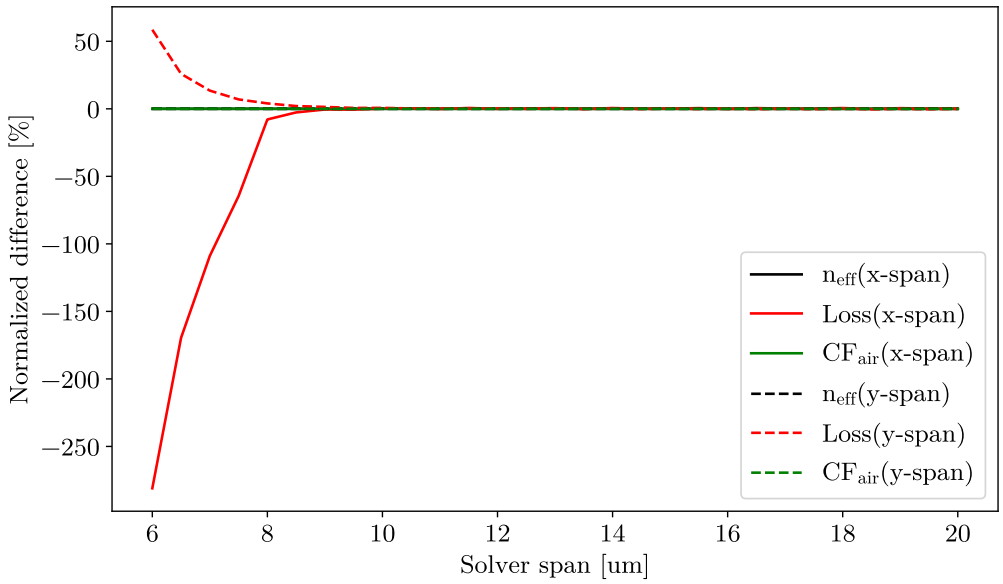


Figure A.3: Differences in effective mode index (n_{eff}), propagation loss, and air confinement factor (CF_{air}) as a function of the FDE solver x-span and y-span normalized to their values at a solver span of 20 × 20 µm. Variations in the x-span were done with a set y-span of 12 µm, while y-span variations were carried out at a set x-span of 20 µm. The reinforcement mesh rib has a mesh size of 1 × 5 nm, while the remaining solver mesh has a fixed mesh size of 50nm × 50 nm.

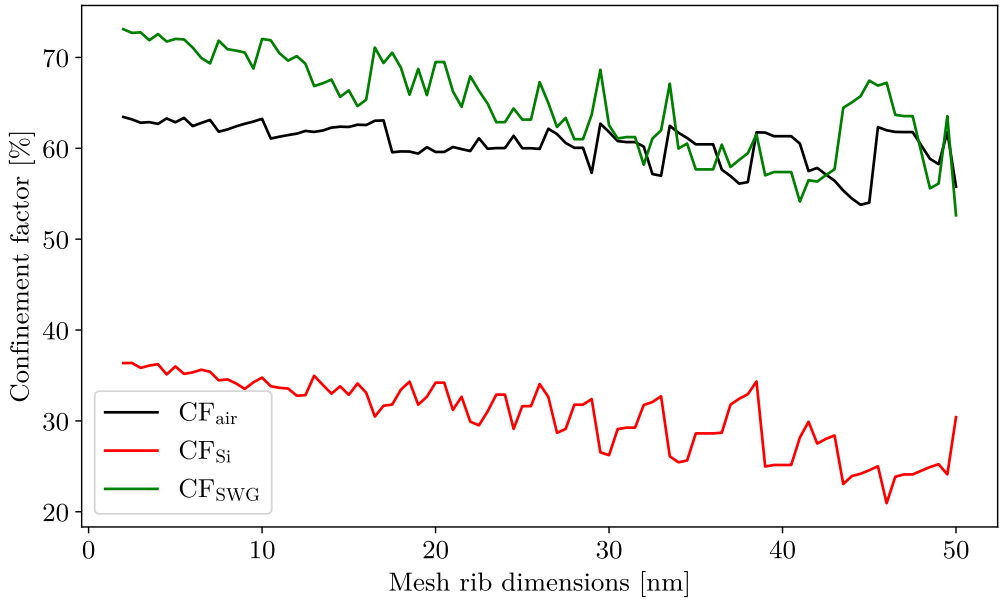


Figure A.4: Curves showing variations in the air confinement factor (CF_{air}), silicon confinement factor (CF_{Si}), and subwavelength grating confinement factor (CF_{SWG}) as a function of decreasing reninforcement rib mesh dimensions. The reinforcement mesh rib has a x- and y-span of 1.4 µm and 0.55 µm, respectively, while the remaining solver mesh has a fixed mesh size of 50 nm × 50 nm.

A.2 The Effect of Material Boundaries on Convergence Behaviour

As detailed in Appendix A.1, the confinement factor was found to resist convergence even at small mesh dimensions of 1 nm $\times$ 1 nm. In fact, as shown in Figure A.4, all confinement factors were seen to steadily increase in value as the mesh dimensions shrunk.

The cause was found to be associated with how material boundaries are meshed by the FDE solver. At material boundaries a mesh cell is often occupied by multiple materials, but as only one set of material properties can be represented in a single cell, the solver will need to determine a value for the refractive index of the cell using one of several available methods. In the simulation environment presented here, the solver employs an approach where cells on boundaries are assigned as refractive index a weighted average of the refractive indices of the neighbouring materials based on their respective representation in the cell. Mesh cells on material borders will thus behave as a sort of metamaterial with a refractive index not corresponding to any physical material found in the structure. Formation of such metamaterials on material borders can be seen in Figure A.5 that shows the effect of reducing mesh dimensions from 50 nm $\times$ 50 nm to 1 nm $\times$ 5 nm.

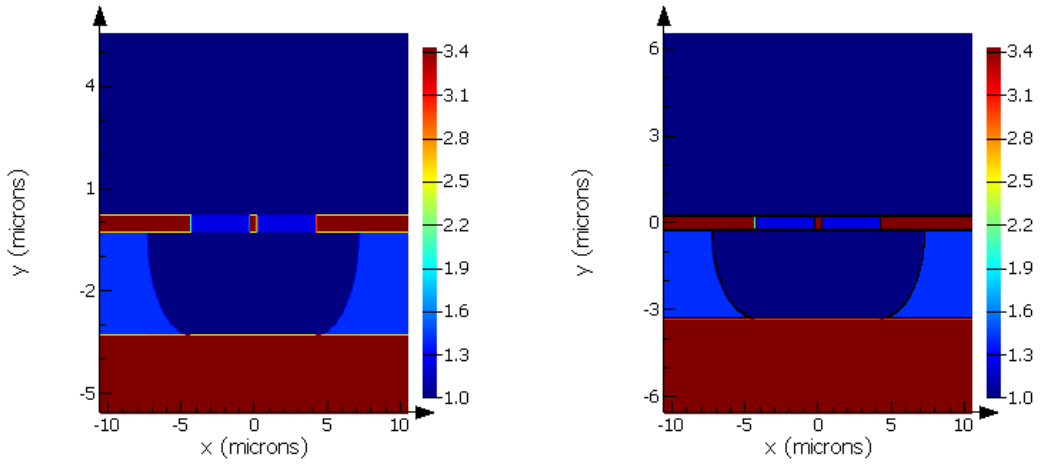


Figure A.5: Mapping of material refractive index to each mesh cell in the FDE solver simulation region showing the effect of reducing the mesh dimensions from 50 nm $\times$ 50 nm (left) to 1 nm $\times$ 5 nm (right).

At inflated mesh dimensions rather large portions of the field energy density can be located in these regions. As the implementation of the confinement factor calculations presented in Appendix D is unable to attribute the field energy density in these mesh cells as belonging to either of the neighbouring materials, it is not counted towards the confinement factor for either. Thus, the confinement factor will be underpredicted at larger mesh dimensions. It is possible to reduce the error in the underprediction by reducing the mesh size. Figure A.6 shows the calculated confinement factor of a waveguide of grating duty cycle 10 % plotted as a function of the refractive index for mesh dimensions of 50 nm $\times$ 50 nm (left) and 1 nm $\times$ 5 nm (right). The confinement factors of the spurious peaks arising from the metamaterials are seen to be almost completely extinguished at finer mesh dimensions. Thus, even though the confinement factors of the waveguide constituents are still increasing with reductions in mesh dimensions, one could make the argument that a point of adequate convergence has been reached as decreasing the spurious metamaterial peaks further will only result in negligble changes to the confinement factors of the constituent materials.

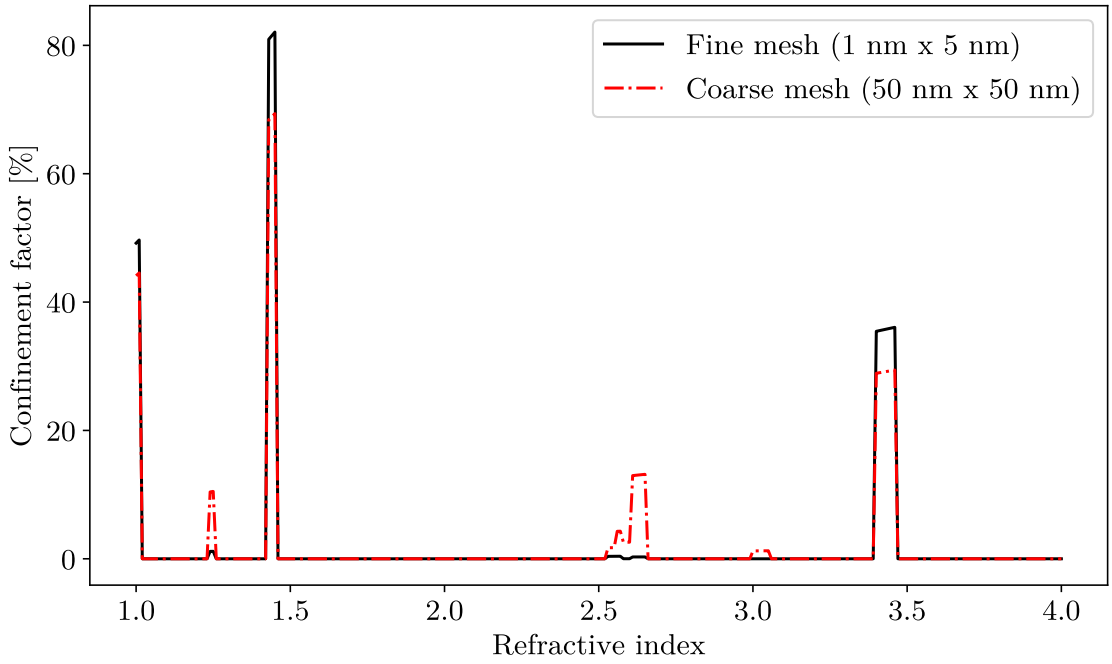


Figure A.6: Calculated confinement factor as a function of refractive index in a waveguide with 10 % grating duty cycle. Confinement factor peaks seen at refractive indices of 1.00, 1.44, and 3.43 correspond to the confinement factors of the air, subwavelength grating, and silicon, respectively. Remaining peaks are spurious and have been found to arise from metamaterials being formed at material boundaries. Decreasing mesh dimensions are found to extinguish the spurious peaks.

B Challenges Encountered during Device Fabrication

B.1 Surface Roughness in the Silicon Device Layer

Upon inspection, micrographs of a portion of the fabricated waveguide batches revealed the presence of extreme surface roughness in the silicon device layer. The micrograph in Figure B.1 shows the effect of the surface roughness on the sidewall profiles of a waveguide with a set duty cycle of 12 % and a grating period of 1000 nm. As discussed in Section 5.2.1, the final waveguide dimensions are expected to deviate by around 50 - 60 nm from the set parameters under normal fabrication circumstances. In the affected samples, however, dimensions were reduced significantly past the nominal, resulting in severely decreased mechanical stability.

In some waveguides, total collapse of the grating thought to be a direct consequence of the loss of structural integrity was observed. Figure B.2 shows a section of waveguide core following detachment from the subwavelength grating. It provides a good perspective into the depth and inhomogenous nature of the roughness. The top of the silicon device layer also shows signs of similar roughness at a less pronounced scale. While the cause of the induced roughness remains unknown, it might be possible to draw some insights from the observations.

There is a notable difference between the extent of the roughness at the top and sides of the device layer. This roughness imbalance might indicate that the sidewall roughness was formed during the reactive ion etching step where the top of the device layer is protected by a resist layer. This would not explain the top surface roughness, however. A more likely explanation is that the surface roughness is a result of oxidation of the silicon device layer by an oxygen providing

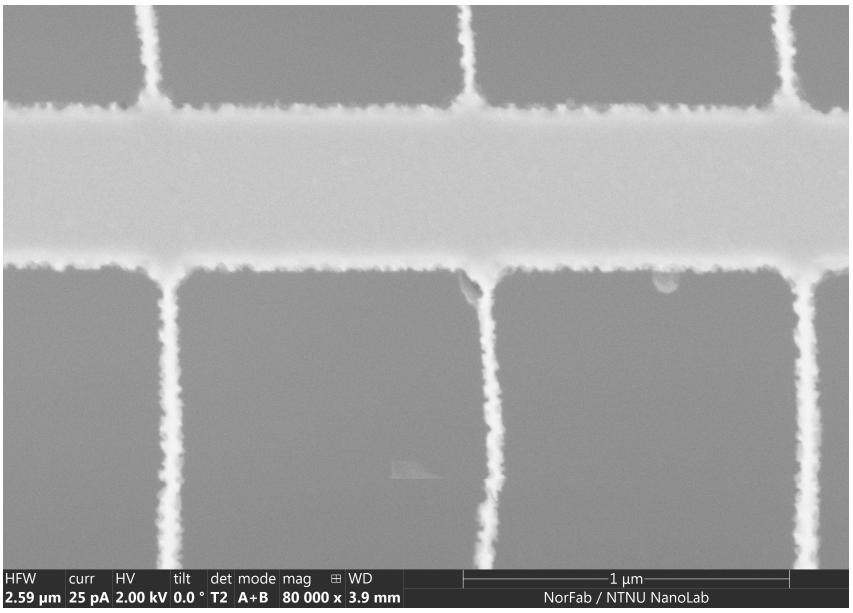


Figure B.1: SEM micrograph of free-standing waveguide core and grating following hydrofluoric acid underetching. The waveguide has a set duty cycle of 12 % and a grating period of 1000 nm. Severe surface roughness can be seen on the grating sidewalls, reducing the width and structural integrity of the grating tethers.

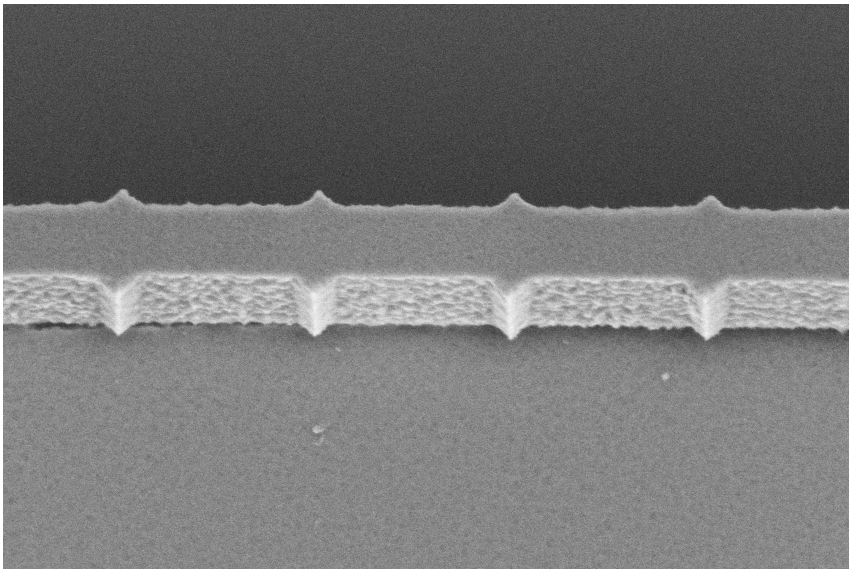


Figure B.2: SEM micrograph showing a section of waveguide core following detachment from the subwavelength grating. The waveguide has a set duty cycle of 12 % and a grating period of 1000 nm. Serious surface roughness can be seen on the sidewalls and top of the device layer, albeit with an imbalance in severity.

contaminant during the hydrofluoric acid (HF) wet etch. As SiO_2 is readily etched by HF, portions of oxidized silicon would be continuously removed during the process, exposing unoxidized silicon to the contaminant so that the oxidation continues. In fact, the surface roughness pattern is highly reminiscent of that caused by aggressive oxygen induced corrosion of other metals such as iron and copper.

The inconsistency in roughness between the surfaces of the device layer might indicate that the chemical process responsible for the etching is anisotropic. This argument might be supported by the very straight angles of the triangular shaped connection point of the grating tethers to the waveguide core. One example of an anisotropic silicon etchant is potassium hydroxide (KOH). The lithographic area of NTNU NanoLab is shared by a large group of users, and as they are available in the same cabinet, it is not unlikely that labware for HF work gets mixed up with labware for KOH etching from time to time. In conjunction with insufficient cleaning, this could cause small amounts of contaminants to remain in beakers or at tweezers and interfere with the nominal wet etching process.

B.2 Effect of EBL Writefield Boundaries on Grating Homogenity

Upon inspection under an optical microscope, a periodic intensity pattern of periodicity 25 µm was observed in the subwavelength gratings. The effect was most pronounced in the lower duty cycle gratings. Figure B.3a shows the intensity variations clearly as dark lines in the grating. The lines are likely the result of an optical effect occuring due to localized width reductions of the associated grating tethers. The width reduction is thought to stem from electorn dose variations at the boundaries of EBL subwritefields. The writefields defined by the user in the EBL software (in this case 500 µm $\times$ 500 µm) is partitioned into several 25 µm $\times$ 25 µm subwritefields during operation. The instrument will try to keep the applied dose constant over the entire area of the subwritefield, but variations might occur, especially at the boundaries between two neighbouring fields. Such dose variations have likely overexposed the resist at the subwritefield boundaries, so that the affected resist area increases. This will result in narrower grating tethers in the affected regions. The width reduction is assumed to be equal across structures of varying duty cycles. This entails that lower duty cycle waveguides will have a larger portion of its affected grating tethers reduced, which in turn might explain the increased pronouncement of the optical effect in these waveguides.

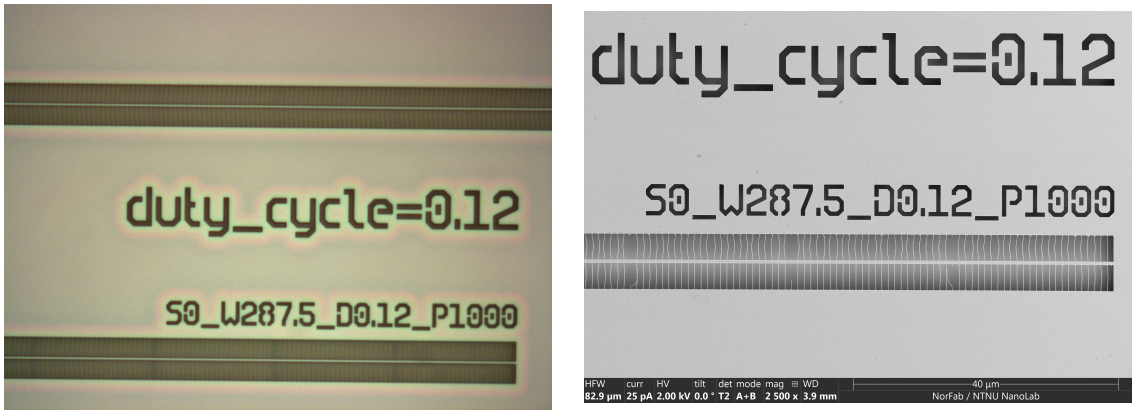


Figure B.3: Optical micrograph acquired at 100x magnification (left) and SEM micrograph (right) showing periodic collapse of grating tethers in waveguides with set duty cycles of 12 %.

In some waveguides, the width reduction was enough to periodically break the tethers, as shown in Figure B.3b. Such collapse of tethers was especially prominent in samples that had also been affected by the surface roughness discussed in Appendix B.1. In addition to showing extreme surface roughness, Figure B.4 also shows the collapse of a pair of tethers on a subwritefield boundary.

To reduce the effect of the EBL subwritefield boundaries, newer batches of waveguides have been

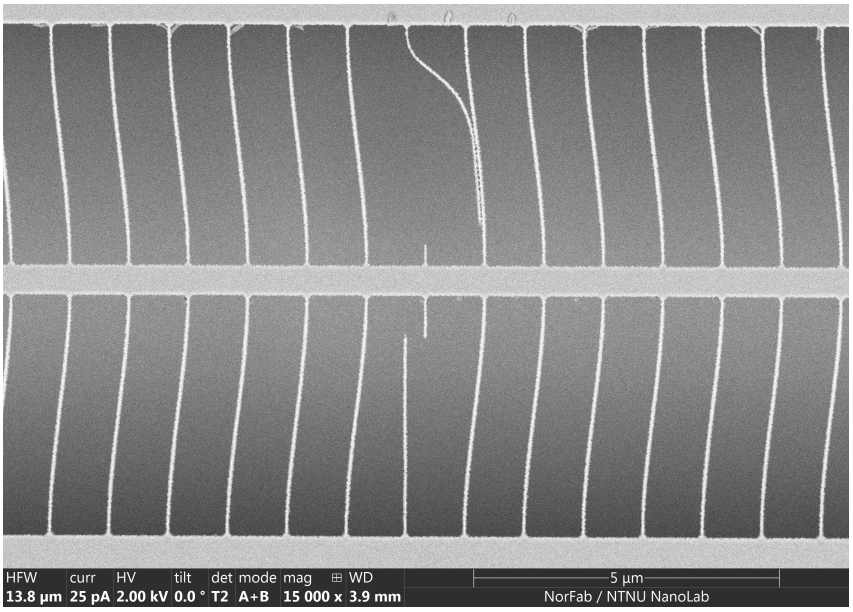


Figure B.4: SEM micrograph showing a section of a waveguide with a set duty cycle of 12 % and a grating period of 1000 nm. Serious surface roughness can be seen on the sidewalls. One pair of grating tethers are seen to have collapsed, likely due to thinning caused by increased electron doses at boundaries of EBL subwritefields during fabrication.

fabricated with a 990 nm grating periodicity to ensure that the periodic dose peaks and tether positions are out of phase. Upon inspection, a spurious intensity pattern was not observed for these samples.

B.3 Effects of Cleaving on Waveguide Facets

In order to couple light into the waveguides, the end facets need to be exposed to indicent light. This has been done by manually scribing and cleaving the wafers perpendicular to the length of the waveguide. Figure B.5 shows the end facet of a free-standing waveguide following cleaving, revealing a multiple of the commonly associated challenges. Firstly, as the waveguide core is mechanically isolated from the rest of the silicon device layer though the rather fragile grating tethers, stress propagation along the cleave stops at the grating. The stress is transferred into the grating upon separation of the two wafer pieces, and the waveguide core breaks as a result. Due to this fracture mechanism, wave cores do not necessarily break flush with the cleaving edge, but may either be found to portrude or retract relative to the cleaving edge. In some samples, facets were found to extend for tens of microns over the cleaving edge, often with a significant droop displacing the facet from its nominal position in the cross-sectional plane.

Furthermore, the imposed stresess on the grating were often found to break and collapse tethers in the vicinity of waveguide facets, reducing the local structural integrity of the waveguide and possibly displacing the facet from its nominal position. In addition to increasing coupling losses, this may introduce reflections that can interfere with spectroscopic measurements.

Optimizing a cleaving procedure is a key point for further work on the fabrication process, which should also be transferable to fabrication of other free-standing waveguide designs. A higher proportion of well cleaved facets was achieved in designs with higher grating duty cycles, assumed to be a consequence of the increased structural integrity of the grating tethers. One

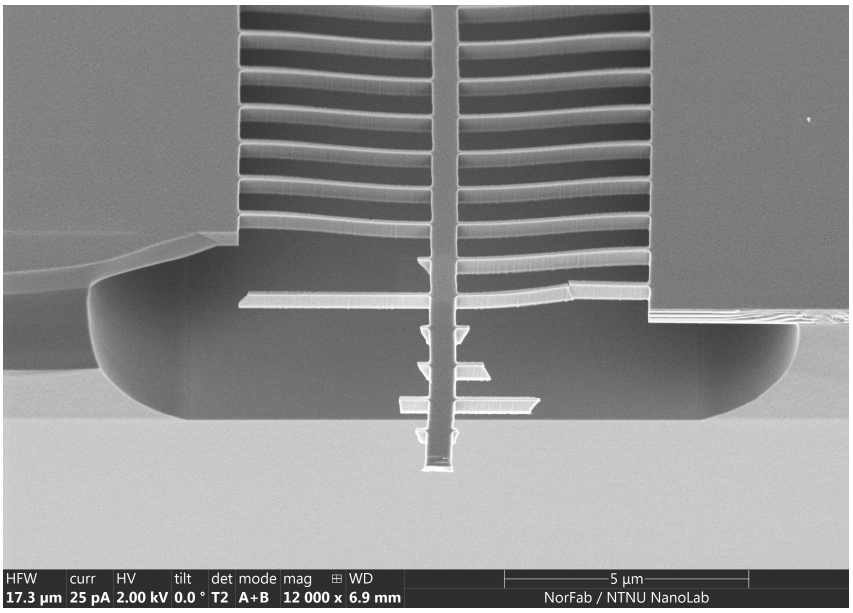


Figure B.5: SEM micrograph showing the end facet of a cleaved waveguide. The figure shows multiple challenges commonly associated with cleaving, such as broken and collapsed grating tethers, waveguide cores extending past the cleaved wafer edge, and asymmetric chipping of the device layer surrounding the waveguide. A slight droop in the waveguide core may also be observed.

way of increasing the mechanical stability of the structures in the vicinity of the cleaving edge may thus be to fabricate strucutres with duty cycles gradually decreasing as a function of the waveguide length. This would allow the cleave to be done in a portion of the waveguide of increased resilience to mechanical stress, while the greater portion of the waveguide could keep the geometric dimensions optimal for sensing. This would require a more in-depth analysis of the associated losses, however. Moreover, cleaving regions could possibly also be defined lithographically, for example by etching scribe lines at the underside of the wafer that can serve to both spatially designate the cleave position and to reduce the forces required for cleaving.

C Python Environment for Ansys Lumerical MODE Batch Simulations

C.1 main.py

```python
"""
Part of parameter sweeping environment for Lumerical MODE using lumapi
Petter Wiker @ 2023
"""

from ParameterSweep import SingleParameterSweep, DoubleParameterSweep
import config
import lumapi as lum


def set_mode_parameters(mode_instance,
                        design: dict[dict[float]] = config.DESIGN,
                        eigensolver: dict[float] = config.EIGENSOLVER_SETTINGS):
    """
    Sets parameters of a lumapi.MODE() instance according to a preset design
    :param design: preset design defined in config.py
    :param mode_instance: an instance of a lumapi.MODE() object
    :param eigensolver: preset ESA settings defined in config.py
    """
    mode_instance.switchtolayout()
    for obj in [config.STRUCTURE_GROUP, config.SOLVER, config.MESH_RIB]:
        mode_instance.select(obj)
        for parameter, value in design[obj].items():
            mode_instance.set(parameter, value)
        mode_instance.runsetup()
    for key, value in eigensolver.items():
        mode_instance.setanalysis(key, value)

    # Set the script variables in the analysis group to True or False depending on if we want to calc and extract them
    mode_instance.select(config.ANALYSIS_GROUP)
    for key, value in config.DESIRED_SCRIPT_PROPERTIES.items():
        mode_instance.set(f"calculate_{key}", 1 if value else 0)
    mode_instance.runsetup()
    pass


if __name__ == "__main__":

    if config.SWEEP_TARGETS:
        for target in config.TARGETS:
            mode = lum.MODE(config.SIMULATION_FILE)
            set_mode_parameters(mode_instance=mode)
            if len(target) == 1:
                sweep = SingleParameterSweep(target=target[0], mode_instance=mode)
                sweep.save_results()
            elif len(target) == 2:
                sweep = DoubleParameterSweep(targets=target, mode_instance=mode)
                sweep.save_results()
            continue
```

C.2 ParameterSweep.py

```python
"""
Part of parameter sweeping environment for Lumerical MODE using lumapi
Petter Wiker @ 2023
"""

import numpy as np
import config
from datetime import datetime
import logging
logging.basicConfig(format="%(asctime)s-%(levelname)s-%(message)s", level=logging.INFO)

```

```python

class SingleParameterSweep:
    def __init__(self, target: dict, mode_instance, design: dict = config.DESIGN):
        """
        Simple class for conducting a n point sweep varying a target parameter from a set min to set max
        :param target: dictionary representing information about a target parameter
        :param mode_instance: a lumapi.MODE() object corresponding to an instance of your simulation design file
        """
        logging.info(f"'{self.__class__.__name__}-{id(self)}' initialized")
        self.sweep_type = self.__class__.__name__
        self.design = design
        self.target = target
        self.structure_group = self.target["scope"]
        self.instance = mode_instance
        self.timestamp = datetime.now().strftime("%d%m%Y_%H%M%S")
        self.name = f"single_sweep_{self.target['name']}_{self.target['min']}_{self.target['max']}_" \
                    f"{self.target['n_points']}_@_{self.timestamp}"
        self.results = self.sweep()
        self.as_dict = {"target": self.target,
                        "name": self.name,
                        "results": self.results,
                        "sweep_type": self.sweep_type,
                        "design": self.design}
        logging.info(f"'{self.__class__.__name__}-{id(self)}' terminated successfully")
        pass

    def sweep(self) -> list[tuple]:
        """
        Performs the 1D sweep across the target parameter and returns the results
        :return: A list of tuples on format [(param_val1, {"mode1": {...}, "mode2": {...}, ...}), (param_val2, ...)]
        """
        results = []
        logging.info(f"Commencing sweep over parameter '{self.target['name']}' from {self.target['min']} to {self.target['max']}")
        for target_instance in np.linspace(self.target["min"], self.target["max"], self.target["n_points"]):
            logging.info(f"Performing mode calculations for '{self.target['name']}' = {target_instance}")
            self.instance.switchtolayout()
            if self.target["type"] == "ESA_sweep":
                for target in self.target["name"]:
                    self.instance.setanalysis(target, target_instance)
            elif self.target["type"] == "DES_sweep":
                self.instance.select(self.structure_group)
                for target in self.target["name"]:
                    self.instance.set(target, target_instance)
            self.instance.runsetup()
            self.instance.select(config.ANALYSIS_GROUP)
            self.instance.runsetup()
            mode = self._get_mode()
            results.append((target_instance, mode))

            if config.DYNAMICALLY_UPDATE_SEARCH_INDEX:
                self.instance.switchtolayout()
                # This is a bit too undynamical for my taste, but just need to have something that works in a hurry..
                try:
                    print(mode["mode1"]["neff"])
                    self.instance.setanalysis("n", mode["mode1"]["neff"])
                except:
                    pass

        return results

    def _get_mode(self) -> dict:
        """
        Extracts the config.DESIRED_PROPERTIES from the different modes that may be present after running the
        Eigensolver Analysis for a selected value of the target parameter
        :return: dict on format {"mode1": {...}, "mode2": {...}, ...} if modes are found, else None
        """
        modes = {}
        n_modes = self.instance.findmodes()
        if n_modes > 0:
            if not config.ALLOW_MULTIPLE_MODES:
                n_modes = 1
            mode_number = 1
            while mode_number <= n_modes:
                self.instance.select(config.SOLVER)
                mode = Mode(name=f"mode{mode_number}",
                            path=f"{config.SOLVER}::data::mode{mode_number}",
                            origin=self.instance)
                modes[f"mode{mode_number}"] = mode.properties
                mode_number += 1
            return modes
```

```
        return None

    def save_results(self):
        """
        Saves the results of the sweep to the folder specified in config.SAVE_DIR
        """
        np.save(f"{config.SAVE_DIR}/{self.name}.npy", self.as_dict)
        logging.info(f"Results of '{self.__class__.__name__}-{id(self)}' saved to '{config.SAVE_DIR}/{self.name}.npy'")
        pass


class DoubleParameterSweep:
    def __init__(self, targets: list[dict], mode_instance, design: dict = config.DESIGN, save_single_sweep: bool = config.SAVE_SINGLE_SWEEP):
        """
        Simple class for conducting a n x m point sweep varying two target parameters from a set min to set max
        :param targets: length 2 list of dictionaries each representing information about a target parameter
        :param mode_instance: a lumapi.MODE() object corresponding to an instance of your simulation design file
        """
        logging.info(f"'{self.__class__.__name__}-{id(self)}' initialized")
        self.sweep_type = self.__class__.__name__
        self.design = design
        self.save_single_sweep = save_single_sweep
        self.targets = targets
        self.structure_group = self.targets[0]["scope"]
        self.instance = mode_instance
        self.timestamp = datetime.now().strftime("%d%m%Y_%H%M%S")
        self.name = f"double_sweep_{self.targets[0]['name']}_{self.targets[0]['min']}_{self.targets[0]['max']}_" \
                    f"{self.targets[0]['n_points']}_@_{self.timestamp}"
        self.results = self.sweep()
        self.as_dict = {"target": self.targets,
                        "name": self.name,
                        "results": self.results,
                        "sweep_type": self.sweep_type,
                        "design": design}
        logging.info(f"'{self.__class__.__name__}-{id(self)}' terminated successfully")
        pass

    def sweep(self) -> list[tuple]:
        """
        Performs the 2D sweep across the target parameters
        :return: A list of tuples on format
                [(param1_val1, [(param2_val1, {"mode1": {...}, "mode2": {...}, ...})), (param2_val2, ...)]),
                 (param1_val2, (...)) ]
        """
        results = []
        x_target = self.targets[0]
        y_target = self.targets[1]
        logging.info(f"Commencing 2D sweep over parameters '{x_target['name']}' from {x_target['min']} to {x_target['max']}, and '{y_target['
        for x_target_instance in np.linspace(x_target["min"], x_target["max"], x_target["n_points"]):
            logging.info(f"Setting '{x_target['name']}' to {x_target_instance}")
            self.instance.switchtolayout()
            if x_target["type"] == "ESA_sweep":
                self.instance.setanalysis(x_target["name"], x_target_instance)
            elif x_target["type"] == "DES_sweep":
                self.instance.select(self.structure_group)
                self.instance.set(x_target["name"], x_target_instance)
            self.instance.runsetup()
            self.instance.select(config.ANALYSIS_GROUP)
            self.instance.runsetup()

            y_sweep = SingleParameterSweep(target=y_target,
                                           mode_instance=self.instance,
                                           design=self.design)
            results.append((x_target_instance, y_sweep.results))
            if self.save_single_sweep:
                y_sweep.save_results()

            if config.DYNAMICALLY_UPDATE_SEARCH_INDEX:
                self.instance.switchtolayout()
                self.instance.setanalysis("n", config.EIGENSOLVER_SETTINGS["n"])
        return results

    def save_results(self):
        """
        Saves the results of the sweep to the folder specified in config.SAVE_DIR
        """
        np.save(f"{config.SAVE_DIR}/{self.name}.npy", self.as_dict)
        logging.info(f"Results of '{self.__class__.__name__}-{id(self)}' saved to '{config.SAVE_DIR}/{self.name}.npy'")
        pass

```

```python

class Mode:
    def __init__(self, name: str, path: str, origin):
        """
        Simple class to represent a found mode and its properties. Allows for running .lsf scripts on the mode's data.
        :param name: the name of the mode in the origin instance. On format "mode1", "mode2", etc.
        :param path: the location of the mode in the origin instance
        :param origin: the lumapi.MODE() object that the mode exists in
        """
        self.name = name
        self.path = path
        self.origin = origin
        self.properties = {}
        params = self.origin.getresult(path)
        keys = params.split("\n")
        for key in keys:
            try:
                if config.DESIRED_MODE_PROPERTIES[key]:
                    self.properties[key] = self.origin.getresult(self.path, key)
            except:
                pass
        if config.RUN_SCRIPTS:
            self._run_lumerical_scripts()
        pass

    def _run_lumerical_scripts(self):
        """
        When supported, this function will run .lsf scripts on the mode in the self.origin instance, allowing for more
        complex data handling, such as calculation and extraction of the Confinement Factor or Modal Intensity Ratio
        """
        self.origin.select(config.ANALYSIS_GROUP)
        self.origin.set("mode", self.name)
        self.origin.runanalysis()
        for key, value in config.DESIRED_SCRIPT_PROPERTIES.items():
            if value:
                self.properties[key] = self.origin.getresult(config.ANALYSIS_GROUP, key)  # Can easily be more efficient
        pass
```

C.3 SweepPlot.py

```python
"""
Part of parameter sweeping environment for Lumerical MODE using lumapi
Petter Wiker @ 2023
"""

import numpy as np
import config
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import cm


class PlotPoint2D:
    def __init__(self, x: float, y: float, mode_type: str) -> None:
        """
        Represents a point in 2D space
        :param x: Generally an instance of the target parameter in a sweep
        :param y: Generally an instance of a mode property for such a target parameter instance
        :param mode_type: Does the simulated waveguide state support NoModes, SingleMode, or MultiModes?
        """
        self.x = x
        self.y = y
        self.pos = (self.x, self.y)
        self.mode_type = mode_type
        pass


class PlotPoint3D(PlotPoint2D):
    def __init__(self, x: float, y: float, z: float, mode_type: str) -> None:
        """
        Represents a point in 3D space.
        :param x: Generally an instance of the first target parameter in a 2D sweep
        :param y: Generally an instance of the second target parameter in a 2D sweep
        :param z: Generally an instance of a mode property for such instances of the target parameters
```

```
        :param mode_type: Does the simulated waveguide state support NoModes, SingleMode, or MultiModes?
        """
        super().__init__(x, y, mode_type)
        self.z = z
        self.pos = (self.x, self.y, self.z)
        pass


class SweepPlot:
    def __init__(self, filename: str):
        """
        Simple class for plotting the results of a Single- or DoubleParameterSweep
        :param filename: the name of a .npy file resulting from a parameter sweep
        """
        self.sweep = np.load(f"{config.SAVE_DIR}/{filename}", allow_pickle=True)
        self.sweep = dict(enumerate(self.sweep.flatten(), 1))[1]  # Nasty...

        self.results = self.sweep["results"]
        self.global_plot_properties = []
        pass

    def plot(self, property: str, mode_number: str = "mode1"):
        """
        Launches either a 2D or 3D plot depending on type of sweep
        :param property: The mode property we want to plot ("neff", "loss", etc.)
        :param mode_number: The name of the mode we want to plot ("mode1", "mode2", etc.)
        """
        if self.sweep["sweep_type"] == "SingleParameterSweep":
            self.plot2D(property, mode_number)
        elif self.sweep["sweep_type"] == "DoubleParameterSweep":
            self.plot3D(property, mode_number)
        pass

    def get_2D_plot_objects(self, property: str, mode_number: str = "mode1", y_offset: float = 0) -> list[PlotPoint2D]:
        plot_objects = []
        for result in self.results:
            if result[1] is not None:
                mode = result[1][mode_number]
            else:
                mode = None
            if mode is not None:
                y = mode[property]
                if type(y) == np.ndarray:
                    y = y[0][0]
                plot_objects.append(PlotPoint2D(x=result[0],
                                                y=np.real(y) + y_offset,
                                                mode_type=self.get_mode_type(len(result[1]))))
            else:
                plot_objects.append(PlotPoint2D(x=result[0],
                                                y=0,
                                                mode_type="NoMode"))
        return plot_objects

    def get_3D_plot_objects(self, property: str, mode_number: str = "mode1", z_offset: float = 0) -> list[PlotPoint3D]:
        plot_objects = []
        for row_element in self.results:
            x = row_element[0]
            for column_element in row_element[1]:
                if column_element[1] is not None:
                    mode = column_element[1][mode_number]
                    z = mode[property]
                    if type(z) == np.ndarray:
                        z = z[0][0]
                    plot_objects.append(PlotPoint3D(x=x,
                                                    y=column_element[0],
                                                    z=np.real(z)+z_offset,
                                                    mode_type=self.get_mode_type(len(column_element[1]))))
                else:
                    plot_objects.append(PlotPoint3D(x=x,
                                                    y=column_element[0],
                                                    z=np.nan,
                                                    mode_type="NoMode"))
        return plot_objects

    def plot2D(self, property: str, mode_number: str = "mode1", plt_line: bool = True):
        """
        Launches a 2D plot corresponding to the results of a SingleParameterSweep
        :param property: The mode property we want to plot ("neff", "loss", etc.)
        :param mode_number: The name of the mode we want to plot ("mode1", "mode2", etc.)
        :param plt_line: Plot a line between the points?
```

```
        """
        plot_objects = self.get_2D_plot_objects(property=property, mode_number=mode_number)

        possible_propagation_modes = ["NoMode", "SingleMode", "MultiMode"]

        data = {}
        for mode_type in possible_propagation_modes:
            data[mode_type] = []
        for point in plot_objects:
            data[point.mode_type].append(point.pos)

        legend = []
        if plt_line:
            legend = ["AllPoints"]
            plt.plot(*zip(*[point.pos for point in plot_objects]), "k-", alpha=0.8)
        for mode_type in possible_propagation_modes:
            if len(data[mode_type]):
                legend.append(mode_type)
                plt.plot(*zip(*data[mode_type]), "o", alpha=1)
        plt.xlabel(self.sweep["target"]["name"])
        plt.ylabel(property)
        plt.legend(legend)
        plt.show()
        pass

    def plot3D(self, property, mode_number="mode1", plt_heat_map=True, plt_propagation_modes=False):
        """
        Launches a 3D plot corresponding to the results of a DoubleParameterSweep
        :param property: The mode property we want to plot ("neff", "loss", etc.)
        :param mode_number: The name of the mode we want to plot ("mode1", "mode2", etc.)
        :param plt_heat_map: Plot the resulting properties in a heat map
        :param plt_propagation_modes: Plot the resulting properties in scatterplot
        """
        plot_objects = self.get_3D_plot_objects(property=property, mode_number=mode_number)

        possible_propagation_modes = ["NoMode", "SingleMode", "MultiMode"]

        data = {}
        for mode_type in possible_propagation_modes:
            data[mode_type] = []
        for point in plot_objects:
            data[point.mode_type].append(point.pos)

        fig = plt.figure(figsize=plt.figaspect(1))
        ax = fig.add_subplot(projection='3d')

        if plt_heat_map:
            n_x = self.sweep["target"][0]["n_points"]
            n_y = self.sweep["target"][1]["n_points"]
            x, y, z = zip(*[point.pos for point in plot_objects])
            z = list(z)
            z = np.array([np.array(z[(i*n_y):(i*n_y+n_y)]) for i in range(n_x)])
            z_tmp = np.ma.masked_where(np.isnan(z), z)
            z = z_tmp
            x, y = np.meshgrid(np.linspace(self.sweep["target"][1]["min"],
                                           self.sweep["target"][1]["max"],
                                           n_y),
                               np.linspace(self.sweep["target"][0]["min"],
                                           self.sweep["target"][0]["max"],
                                           n_x))
            surf = ax.plot_surface(x, y, z,
                                   norm=matplotlib.colors.Normalize(vmin=np.min(z),
                                                                    vmax=np.max(z)),
                                   cmap=cm.RdYlGn_r, alpha=1)
            ax.set_xlabel(self.sweep["target"][1]["name"])
            ax.set_ylabel(self.sweep["target"][0]["name"])
            ax.set_zticklabels([])
            color_bar = fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10)
            color_bar.set_label(property)

        if plt_propagation_modes:
            legend = []
            for mode_type in possible_propagation_modes:
                if len(data[mode_type]):
                    legend.append(mode_type)
                    ax.scatter(*zip(*data[mode_type]), "o", s=1, alpha=1)
            ax.set_xlabel(self.sweep["target"][1]["name"])
            ax.set_ylabel(self.sweep["target"][0]["name"])
            ax.set_zticklabels([])
            ax.legend(legend)
```

```
        plt.show()
        pass

    @staticmethod
    def get_mode_type(n_modes: float) -> str:
        """
        Get the string representation of a simulated waveguide propagation state from number of found modes
        :param n_modes: number of found modes
        :return: string representation of simulated waveguide propagation state
        """
        if n_modes == 0:
            return "NoMode"
        if n_modes == 1:
            return "SingleMode"
        return "MultiMode"
```

C.4 config.py

```
"""
Part of parameter sweeping environment for Lumerical MODE using lumapi
Petter Wiker @ 2023
"""

SIMULATION_FILE      = "Si_fsswg"   # Should contain the relative path to your Lumerical simulation file
SOLVER               = "FDE"        # The name of the solver you use in your Lumerical simulation file
STRUCTURE_GROUP      = "waveguide"  # The name of the structure group containing the variables you sweep over
ANALYSIS_GROUP       = "::model"    # The name of the parent model that contains the analysis scripts
MESH_RIB             = "mesh_rib"   # The name of the reinforcement mesh rib contained in the parent model
SAVE_DIR             = "data"       # Should contain the relative path to your data storage
RUN_SCRIPTS          = True         # After running the solver on a sweep point, should we run any .lsf scripts?
ALLOW_MULTIPLE_MODES = True         # --//--, should we extract multiple modes if possible?
SWEEP_TARGETS        = True         # Should we sweep through the targets in the target list?
SAVE_SINGLE_SWEEP    = True         # In a double param sweep, should we also save results of single sweeps?

DYNAMICALLY_UPDATE_SEARCH_INDEX = True

# Define your sweep targets. Each element will be ran as a Single- or DoubleParameterSweep depending on length (1 or 2)
# May have as many targets as you want, which will be sweeped in order
TARGETS = [{"name": ["grating_width"],
            "min": 2e-6,
            "max": 6e-6,
            "n_points": 31,
            "type": "DES_sweep",
            "scope": STRUCTURE_GROUP}]]

# "DES_sweep" corresponds to sweep over design and solver parameters, while "ESA_sweep"'s are over eigensolver settings

# Select which properties to be extracted from the calculated mode(s) at each node in the sweep
DESIRED_MODE_PROPERTIES = {
    "E": False,
    "H": False,
    "P": False,
    "neff": True,
    "loss": True,
    "TE polarization fraction": False,
    "waveguide TE/TM fraction": False,
    "waveguide TE-TM fraction": False,
    "mode effective area": False,
    "x": False,
    "y": False,
    "z": False,
    "Ex": False,
    "Ey": False,
    "Ez": False,
    "Hx": False,
    "Hy": False,
    "Hz": False,
    "Z0": False
}

# Select which properties to calculate/extract in/from the analysis group.
DESIRED_SCRIPT_PROPERTIES = {
    "CF_air": True,
```

```
    "CF_SiO2": True,
    "CF_Si": True,
    "CF_SWG": True,
    "CF_air_approx": True,
    "MIF_air": True,
    "MIF_SiO2": True,
    "MIF_Si": True,
    "MIF_SWG": True,
    "MIF_air_approx": True,
    "NEFED_air": True,
    "NEFED_SiO2": True,
    "NEFED_Si": True,
    "NEFED_SWG": True,
    "CF_index_sweep_indices": False,
    "CF_index_sweep_cfs": False
}

EIGENSOLVER_SETTINGS = {
    "bent waveguide": 0,
    "bend radius": 1e6,
    "bend orientation": 360,
    "use max index": 0,
    "n": 1.39,
    "number of trial modes": 2,
    "wavelength": 3274e-9
}

DESIGN_PETTER = {
    STRUCTURE_GROUP: {
        "slot_width": 0e-9,
        "wg_width": 230e-9,
        "Si_height": 500e-9,
        "Si_bottom_thickness": 20e-6,
        "SiO2_thickness": 15e-6,
        "grating_width": 3e-6,
        "anchor_Si_width": 40e-6,
        "wavelength": 3274e-9,
        "duty_cycle": 0.15,
        "z_depth": 1e-6,
        "delta_w": 0,
        "use_custom_background": 0
    },
    SOLVER: {
        "x span": 20e-6,
        "y span": 12e-6,
        "dx": 50e-9,
        "dy": 50e-9,
        "x min bc": "Anti-Symmetric",
        "x max bc": "PML",
        "y min bc": "PML",
        "y max bc": "PML"
    },
    MESH_RIB: {
        "x span": 1.4e-6,
        "y span": 0.55e-6,
        "dx": 50e-9,
        "dy": 50e-9
    }
}

DESIGN_ZHOU = {
    STRUCTURE_GROUP: {
        "slot_width": 70e-9,
        "wg_width": 465e-9,
        "Si_height": 340e-9,
        "Si_bottom_thickness": 20e-6,
        "SiO2_thickness": 3e-6,
        "grating_width": 4e-6,
        "anchor_Si_width": 20e-6,
        "wavelength": 2250e-9,
        "duty_cycle": 0.342,
        "z_depth": 1e-6,
        "delta_w": 0e-9
    },
    SOLVER: {
        "x span": 4e-6,
        "y span": 2.5e-6,
        "dx": 50e-9,
        "dy": 50e-9,
        "x min bc": "PML",
```

```
        "x max bc": "PML",
        "y min bc": "PML",
        "y max bc": "PML"
    },
    MESH_RIB: {
        "x span": 1.01e-6,
        "y span": 0.4e-6,
        "dx": 1e-9,
        "dy": 5e-9
    }
}

DESIGN = DESIGN_PETTER
```

D ANSYS Lumerical MODE Scripting Environment for Confinement Factor Computation

D.1 analysis.lsf

```
function compute_normalized_electrical_field_intensity(n, mode) {

    index_x = real(pinch(getdata("FDE::data::material","index_x")));
    x = pinch(getdata("FDE::data::material","x"));
    y = pinch(getdata("FDE::data::material","y"));
    nx = length(x);
    ny = length(y);

    # define the mask matrix
    filter = matrix(nx,ny);

    rel_dif = 0.01; # something like an error for the ref. index
    index_val_mat = ones(nx,ny)*n;
    filter(1:nx,1:ny) = almostequal(pinch(index_x), index_val_mat, rel_dif);

    # clear unnecessary data
    clear(index_x, nx, ny, rel_dif, index_val_mat);

    # retrieve the field data of a given mode
    Ex = pinch(getdata("FDE::data::"+mode,"Ex"));
    Ey = pinch(getdata("FDE::data::"+mode,"Ey"));
    Ez = pinch(getdata("FDE::data::"+mode,"Ez"));
    Hx = pinch(getdata("FDE::data::"+mode,"Hx"));
    Hy = pinch(getdata("FDE::data::"+mode,"Hy"));

    # calculate Pz and |E| squared
    Pz = 0.5 * (Ex*conj(Hy) - Ey*conj(Hx));

    # factor 0.5 is important due to fields described with complex numbers
    Enorm2 = Ex*conj(Ex) + Ey*conj(Ey) + Ez*conj(Ez);
    return Enorm2;

}

function compute_confinement_factor_at_n(n, mode){

    #Enorm2 = compute_normalized_electrical_field_intensity(n, mode);
    index_x = real(pinch(getdata("FDE::data::material","index_x")));
    x = pinch(getdata("FDE::data::material","x"));
    y = pinch(getdata("FDE::data::material","y"));
    nx = length(x);
    ny = length(y);

    # define the mask matrix
    filter = matrix(nx,ny);

    rel_dif = 0.01; # something like an error for the ref. index
    index_val_mat = ones(nx,ny)*n;
    filter(1:nx,1:ny) = almostequal(pinch(index_x), index_val_mat, rel_dif);

    # clear unnecessary data
    clear(index_x, nx, ny, rel_dif, index_val_mat);

    # retrieve the field data of a given mode
    Ex = pinch(getdata("FDE::data::"+mode,"Ex"));
    Ey = pinch(getdata("FDE::data::"+mode,"Ey"));
    Ez = pinch(getdata("FDE::data::"+mode,"Ez"));
    Hx = pinch(getdata("FDE::data::"+mode,"Hx"));
    Hy = pinch(getdata("FDE::data::"+mode,"Hy"));

    # calculate Pz and |E| squared
    Pz = 0.5 * (Ex*conj(Hy) - Ey*conj(Hx));

    # factor 0.5 is important due to fields described with complex numbers
    Enorm2 = Ex*conj(Ex) + Ey*conj(Ey) + Ez*conj(Ez);

    # apply a mask to |E| squared
    Enorm2_f = Enorm2*filter;

```

```
    # confinement factor calculation and acquisition
    intP = integrate(real(Pz), [1,2], x, y);
    intE_f = integrate(real(Enorm2_f), [1,2], x, y);
    CF = (c*eps0/2.0) * n * intE_f / intP;

    # clear unnecessary data
    clear(n, x, y, Ex, Ey, Ez, Hx, Hy, Pz, Enorm2, Enorm2_f, intP, intE_f);
    # ?mode + " confinement factor is " + num2str(CF*100) + "%";
    return CF;

}



function compute_modal_intensity_fraction_at_n(n, mode){

    #Enorm2 = compute_normalized_electrical_field_intensity(n, mode);
    index_x = real(pinch(getdata("FDE::data::material","index_x")));
    x = pinch(getdata("FDE::data::material","x"));
    y = pinch(getdata("FDE::data::material","y"));
    nx = length(x);
    ny = length(y);

    # define the mask matrix
    filter = matrix(nx,ny);

    rel_dif = 0.01; # something like an error for the ref. index
    index_val_mat = ones(nx,ny)*n;
    filter(1:nx,1:ny) = almostequal(pinch(index_x), index_val_mat, rel_dif);

    # clear unnecessary data
    clear(index_x, nx, ny, rel_dif, index_val_mat);

    # retrieve the field data of a given mode
    Ex = pinch(getdata("FDE::data::"+mode,"Ex"));
    Ey = pinch(getdata("FDE::data::"+mode,"Ey"));
    Ez = pinch(getdata("FDE::data::"+mode,"Ez"));
    Hx = pinch(getdata("FDE::data::"+mode,"Hx"));
    Hy = pinch(getdata("FDE::data::"+mode,"Hy"));

    # calculate Pz and |E| squared
    Pz = 0.5 * (Ex*conj(Hy) - Ey*conj(Hx));

    # factor 0.5 is important due to fields described with complex numbers
    Enorm2 = Ex*conj(Ex) + Ey*conj(Ey) + Ez*conj(Ez);

    # apply a mask to |E| squared
    Enorm2_f = Enorm2*filter;

    # modal intensity fraction calculations
    intE_f = integrate(real(Enorm2_f), [1,2], x, y);
    intE = integrate(real(Enorm2), [1,2], x, y);
    MIF = intE_f/intE;

    # clear unnecessary data
    clear(n, x, y, Ex, Ey, Ez, Hx, Hy, Enorm2, Enorm2_f, intE_f, intE);

    return MIF;
}



function compute_electrical_field_energy_density_at_n(n, mode){

    #Enorm2 = compute_normalized_electrical_field_intensity(n, mode);
    index_x = real(pinch(getdata("FDE::data::material","index_x")));
    x = pinch(getdata("FDE::data::material","x"));
    y = pinch(getdata("FDE::data::material","y"));
    nx = length(x);
    ny = length(y);

    # define the mask matrix
    filter = matrix(nx,ny);

    rel_dif = 0.01; # something like an error for the ref. index
    index_val_mat = ones(nx,ny)*n;
    filter(1:nx,1:ny) = almostequal(pinch(index_x), index_val_mat, rel_dif);

    # clear unnecessary data
    clear(index_x, nx, ny, rel_dif, index_val_mat);
```

```
150
151     # retrieve the field data of a given mode
152     Ex = pinch(getdata("FDE::data::"+mode,"Ex"));
153     Ey = pinch(getdata("FDE::data::"+mode,"Ey"));
154     Ez = pinch(getdata("FDE::data::"+mode,"Ez"));
155     Hx = pinch(getdata("FDE::data::"+mode,"Hx"));
156     Hy = pinch(getdata("FDE::data::"+mode,"Hy"));
157
158     # calculate Pz and |E| squared
159     Pz = 0.5 * (Ex*conj(Hy) - Ey*conj(Hx));
160
161     # factor 0.5 is important due to fields described with complex numbers
162     Enorm2 = Ex*conj(Ex) + Ey*conj(Ey) + Ez*conj(Ez);
163
164     # apply a mask to |E| squared
165     Enorm2_f = Enorm2*filter;
166
167     # calculating electrical field energy density
168     intE_f = n^2*integrate(real(Enorm2_f), [1,2], x, y);
169
170     # clear unnecessary data
171     clear(n, x, y, Ex, Ey, Ez, Hx, Hy, Enorm2, Enorm2_f);
172
173     return intE_f;
174 }
175
176 function sweep_confinement_factors_over_n(n_start, n_stop, mode, stepsize){
177
178     #num_indices = (n_stop - n_start)/stepsize;
179     num_indices = 301;
180     indices = [1:num_indices];
181     confinement_factors = [1:num_indices];
182
183     n = n_start;
184     for (i=1; i<=length(indices); i=i+1){
185         indices(i) = n;
186         confinement_factors(i) = compute_confinement_factor_at_n(n, mode);
187         n = n + stepsize;
188     }
189     return [indices, confinement_factors];
190 }
191
192 if (calculate_CF_index_sweep_indices | calculate_CF_index_sweep_cfs){
193
194     idx_cf = sweep_confinement_factors_over_n(index_start=1, index_stop=4, mode=mode, step=0.01);
195
196     if (calculate_CF_index_sweep_indices){
197         CF_index_sweep_indices = idx_cf(:, 1);
198     } else {
199         CF_index_sweep_indices = [-1];
200     }
201
202     if (calculate_CF_index_sweep_cfs){
203         CF_index_sweep_cfs = idx_cf(:, 2);
204     } else {
205         CF_index_sweep_indices = [-1];
206     }
207 }
208
209 # Calculating the electrical field energy density (EFED) for the constituents of the waveguide
210 EFED_air = compute_electrical_field_energy_density_at_n(index_background, mode);
211 EFED_Si = compute_electrical_field_energy_density_at_n(index_Si, mode);
212 EFED_SiO2 = compute_electrical_field_energy_density_at_n(index_SiO2, mode);
213 EFED_SWG = compute_electrical_field_energy_density_at_n(index_SWG, mode);
214 sum_EFED = EFED_air + EFED_Si + EFED_SiO2 + EFED_SWG;
215
216 # calculating the normalized electrical field energy density (NEFED) from the waveguide constituents
217 NEFED_air = EFED_air/sum_EFED;
218 NEFED_Si = EFED_Si/sum_EFED;
219 NEFED_SiO2 = EFED_SiO2/sum_EFED;
220 NEFED_SWG = EFED_SWG/sum_EFED;
221
222 if (calculate_CF_air == 1){
223     CF_air = compute_confinement_factor_at_n(index_background, mode);
224 } else {
225     CF_air = -1;
226 }
227
228 if (calculate_MIF_air == 1){
229     MIF_air = compute_modal_intensity_fraction_at_n(index_background, mode);
```

```
} else {
    MIF_air = -1;
}

if (calculate_CF_Si == 1){
    CF_Si = compute_confinement_factor_at_n(index_Si, mode);
} else {
    CF_Si = -1;
}

if (calculate_MIF_Si == 1){
    MIF_Si = compute_modal_intensity_fraction_at_n(index_Si, mode);
} else {
    MIF_Si = -1;
}

if (calculate_CF_SiO2 == 1){
    CF_SiO2 = compute_confinement_factor_at_n(index_SiO2, mode);
} else {
    CF_SiO2 = -1;
}

if (calculate_MIF_SiO2 == 1){
    MIF_SiO2 = compute_modal_intensity_fraction_at_n(index_SiO2, mode);
} else {
    MIF_SiO2 = -1;
}

if (calculate_CF_SWG == 1){
    CF_SWG = compute_confinement_factor_at_n(index_SWG, mode);
} else {
    CF_SWG = -1;
}

if (calculate_MIF_SWG == 1){
    MIF_SWG = compute_modal_intensity_fraction_at_n(index_SWG, mode);
} else {
    MIF_SWG = -1;
}

if (calculate_MIF_air_approx == 1){
    MIF_air_approx = MIF_SWG*(1-duty_cycle) + MIF_air;
} else {
    MIF_air_approx = -1;
}

if (calculate_CF_air_approx == 1){
    CF_air_approx = CF_SWG*(1-duty_cycle) + CF_air;
} else {
    CF_air_approx = -1;
}
```

E Python Environment for Lithographic Mask Design

E.1 main.py

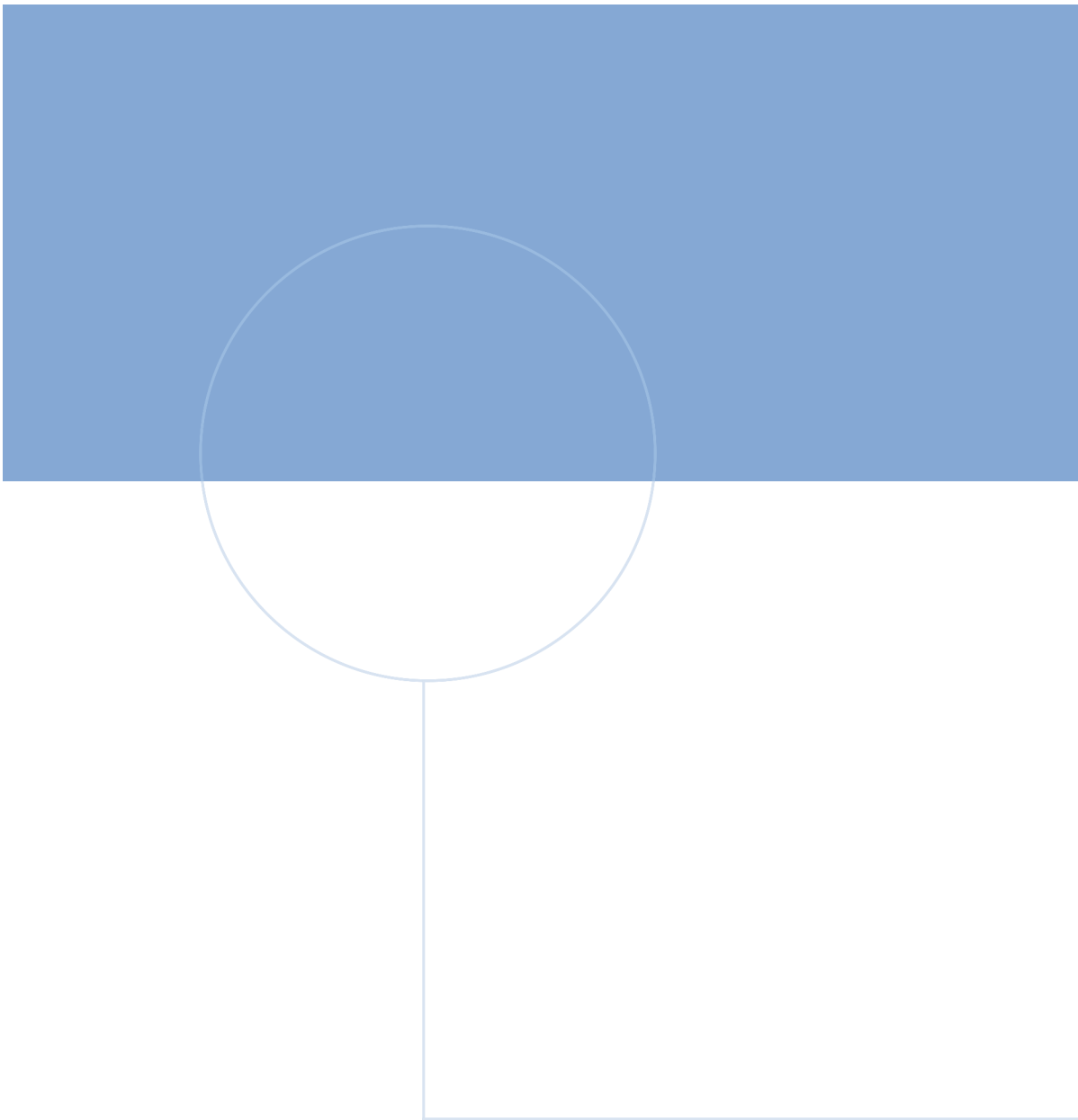
```
import spiral_lib as spi
import numpy as np
import nazca as nd
import copy
from tqdm import tqdm
from scipy.misc import derivative
from scipy.interpolate import interp1d


def tg_angle(x_in, y_in, s):
    interpmode = 'cubic'
    x = interp1d(s, x_in, kind=interpmode, fill_value='extrapolate')
    y = interp1d(s, y_in, kind=interpmode, fill_value='extrapolate')

    dx = derivative(x, s, dx = 0.000001)
    dy = derivative(y, s, dx = 0.000001)
    th = np.arctan(dy / dx)

    for i in range(0, np.size(th)):
        if dx[i] < 0.0:
            th[i] += np.pi
        elif dx[i] == 0.0 and dy[i] < 0.0:
            th[i] = np.arctan(-np.inf)
        elif dx[i] == 0.0 and dy[i] > 0.0:
            th[i] = np.arctan(np.inf)

    return th


def xy_coordinates_from_s_grating(wg_dim, x, y, s, beam_s_resolution: int = 2, straight: bool = False):
    """
    Transforms a parameterized curve s into xy-coordinates and calculates the placement of all elements in
    the free-standing waveguide design. General for use with all parameterized curves s.
    :param wg_dim: The dimensions of the waveguide design, i.e. width, grating period, duty cycle, etc.
    :param x: The x-origin of the waveguide.
    :param y: The y-origin of the waveguide.
    :param s: A parameterized curve.
    :param beam_s_resolution: Resolution of the transformation.
    :param straight: Is the waveguide straight or not?
    :return:
    """
    with nd.Cell(name="my_cell") as cell:
        beam_width = wg_dim["period"]*(1-wg_dim["duty_cycle"])

        s_start = min(s) + (beam_width/2)
        s_grating = [s_start]
        s_end = max(s)
        while s_start <= s_end:
            s_start += wg_dim["period"]
            s_grating.append(s_start)

        partitions = np.linspace(-beam_width/2, beam_width/2, beam_s_resolution)
        if straight:
            tg_angles = [np.pi/2 for i in range(len(s))]
            mysterious_factor = -1
        else:
            tg_angles = tg_angle(x, y, s)
            mysterious_factor = 1
        for s_beam_center in tqdm(s_grating):
            closest_s_index = min(range(len(s)), key=lambda i: abs(s[i]-s_beam_center))

            # make a section for each beam placement
            s_center_line_points = []
            for idx, val in enumerate(partitions):
                s_location = s_beam_center + val
                if mysterious_factor*(s_location - s[closest_s_index]) <= 0:
                    try:
                        s_curr_next_progression = (s_location - s[closest_s_index])/ \
                                                  (s[closest_s_index+1] - s[closest_s_index])
                        s_center_line_points.append((x[closest_s_index] + s_curr_next_progression*(x[closest_s_index+1] -
                                                                                                   x[closest_s_index]),
```

```
                                                y[closest_s_index] + s_curr_next_progression*(y[closest_s_index+1] -
                                                                                              y[closest_s_index])
                                                )
                                               )

                    except:
                        None
                else:
                    try:
                        s_prev_curr_progression = (s[closest_s_index] - s_location)/ \
                                                  (s[closest_s_index] - s[closest_s_index-1])
                        s_center_line_points.append((x[closest_s_index] - s_prev_curr_progression*(x[closest_s_index] -
                                                                                                   x[closest_s_index-1]),
                                                     y[closest_s_index] - s_prev_curr_progression*(y[closest_s_index] -
                                                                                                   y[closest_s_index-1])
                                                     )
                                                    )

                    except:
                        None

            tangent_angle = tg_angles[closest_s_index]
            r_angle = np.pi/2-tangent_angle
            l_angle = tangent_angle
            right_line_points = []
            left_line_points = []
            for point in s_center_line_points:
                right_line_points.append(((point[0] + np.cos(r_angle)*wg_dim["wg_width"]),
                                          (point[1] - np.sin(r_angle)*wg_dim["wg_width"])))

                left_line_points.append(((point[0] - np.sin(l_angle)*wg_dim["wg_width"]),
                                         (point[1] + np.cos(l_angle)*wg_dim["wg_width"])))
            if len(s_center_line_points):
                right_bar_points = copy.copy(right_line_points)
                right_bar_points.append((right_bar_points[-1][0] + np.cos(r_angle)*wg_dim["bridge_width"],
                                         right_bar_points[-1][1] - np.sin(r_angle)*wg_dim["bridge_width"]))
                right_bar_points.append((right_bar_points[0][0] + np.cos(r_angle)*wg_dim["bridge_width"],
                                         right_bar_points[0][1] - np.sin(r_angle)*wg_dim["bridge_width"]))
                right_bar_points = [(point[0]*1000, point[1]*1000) for point in right_bar_points]
                nd.Polygon(points=right_bar_points, layer="layer1").put()

                left_bar_points = copy.copy(left_line_points)
                left_bar_points.append((left_bar_points[-1][0] - np.sin(l_angle)*wg_dim["bridge_width"],
                                        left_bar_points[-1][1] + np.cos(l_angle)*wg_dim["bridge_width"]))
                left_bar_points.append((left_bar_points[0][0] - np.sin(l_angle)*wg_dim["bridge_width"],
                                        left_bar_points[0][1] + np.cos(l_angle)*wg_dim["bridge_width"]))
                left_bar_points = [(point[0]*1000, point[1]*1000) for point in left_bar_points]
                nd.Polygon(points=left_bar_points, layer="layer1").put()
    return cell

nd.clear_layout()
# clear out layers from memory
nd.clear_layers()
# clear out xsections from memory
nd.clear_xsections()

# create a layer and define its accuracy
nd.add_layer(name='layer1', layer=1, accuracy=0.001)


class StraightWaveGuide:

    def __init__(self, length: float, dimensions: dict):
        """
        Simple class for generation of straight free-standing waveguides
        :param length: The length of the waveguide
        :param dimensions: The dimensions of the waveguide design, i.e. width, grating period, duty cycle, etc.
        """
        self.dimensions = dimensions
        self.length = 1000*length
        s = [0, length]
        x = [0, 0]
        y = [0, length]
        self.cell = xy_coordinates_from_s_grating(dimensions, x, y, s, straight=True)
        pass

    def put(self, x: float = 0, y: float = 0, a: float = 0, text_height_sem: float = 5, text_height_om: float = 15):
        """
        Places a labeled waveguide at a certain point in the GDS-file
        :param x: The x-origin of the waveguide placement
```

```
        :param y: The y-origin of the waveguide placement
        :param a: The angle at which the waveguide is placed
        :param text_height_sem: Text height for smallest text, visible in SEM
        :param text_height_om: Text height for largest text, visible in optical microscope
        :return:
        """
        self.cell.put(x, y, a)
        message = f"W_{round(1000*self.dimensions['wg_width'], 3)}_D{self.dimensions['duty_cycle']}_P{1000**2*self.dimensions['period']}"
        nd.text(text=message, height=text_height_sem, align="lb").put(x-15*np.cos(np.deg2rad(a)), y-15*np.sin(np.deg2rad(a)), a+90)
        nd.text(text=message, height=text_height_om, align="lb").put(x-25*np.cos(np.deg2rad(a)), y-25*np.sin(np.deg2rad(a)), a+90)

        nd.text(text=message, height=text_height_sem, align="rb").put(x-15*np.cos(np.deg2rad(a))+self.length*np.cos(np.deg2rad(a+90)),
                                                                      y-15*np.sin(np.deg2rad(a))+self.length*np.sin(np.deg2rad(a+90)),
                                                                      a+90)
        nd.text(text=message, height=text_height_om, align="rb").put(x-25*np.cos(np.deg2rad(a))+self.length*np.cos(np.deg2rad(a+90)),
                                                                     y-25*np.sin(np.deg2rad(a))+self.length*np.sin(np.deg2rad(a+90)),
                                                                     a+90)
        pass


class SpiralWaveGuide:

    def __init__(self, length: float, dimensions: dict, spiral_params: dict, no_waveguides: int = 1, separation: float = 100):
        """
        Class for generation of adiabatic spiral free-standing waveguides
        :param length: The length of the waveguides.
        :param dimensions: The dimensions of the waveguide design, i.e. width, grating period, duty cycle, etc.
        :param spiral_params: Parameters for the spiral, i.e. radius of curvature, pitch, and number of turns
        :param no_waveguides: Number of parallel waveguides in the spiral
        :param separation: Separation between the parallel waveguides in the spiral
        """
        self.length = length
        self.dimensions = dimensions
        self.radius_of_curvature = spiral_params["radius_of_curvature"]
        self.pitch = spiral_params["pitch"]
        self.n_turns = spiral_params["n_turns"]
        self.curve = self.define_curve()
        self.cells = []
        for idx, offset in enumerate(np.linspace(0, (no_waveguides-1)*separation, no_waveguides)):
            x, y, s = spi.clone(self.curve.x, self.curve.y, self.curve.s, offset-(no_waveguides-1)*separation/2)
            self.cells.append(xy_coordinates_from_s_grating(dimensions, x, y, s))
        pass

    def put(self, x: float = 0, y: float = 0, a: float = 0, text_height_sem: float = 10, text_height_om: float = 50):
        """
        Places a labeled waveguide at a certain point in the GDS-file
        :param x: The x-origin of the waveguide placement
        :param y: The y-origin of the waveguide placement
        :param a: The angle at which the waveguide is placed
        :param text_height_sem: Text height for smallest text, visible in SEM
        :param text_height_om: Text height for largest text, visible in optical microscope
        :return:
        """
        for cell in self.cells:
            cell.put(x, y, a)

        offset_sem_x = (self.radius_of_curvature*1000 - text_height_om)/2
        offset_om_x = self.radius_of_curvature/2*1000 + text_height_om
        offset_y = self.radius_of_curvature*1000/3

        message = f"W_{round(1000*self.dimensions['wg_width'], 3)}_D{self.dimensions['duty_cycle']}_P{1000**2*self.dimensions['period']}"
        message_2 = f"R_{round(1000*self.radius_of_curvature, 3)}_Pi{self.pitch}_N{self.n_turns}"
        nd.text(text=message, height=text_height_sem, align="cb").put(x+offset_sem_x*np.cos(np.deg2rad(a))-offset_y*np.sin(np.deg2rad(a)),
                                                                      y+offset_sem_x*np.sin(np.deg2rad(a))+offset_y*np.cos(np.deg2rad(a)),
                                                                      a+90)
        nd.text(text=message, height=text_height_om, align="cb").put(x+offset_om_x*np.cos(np.deg2rad(a))-offset_y*np.sin(np.deg2rad(a)),
                                                                     y+offset_om_x*np.sin(np.deg2rad(a))+offset_y*np.cos(np.deg2rad(a)),
                                                                     a+90)

        nd.text(text=message_2, height=text_height_sem, align="cb").put(
                                        x+(offset_sem_x-text_height_sem*10)*np.cos(np.deg2rad(a))-offset_y*np.sin(np.deg2rad(a)),
                                        y+(offset_sem_x-text_height_sem*10)*np.sin(np.deg2rad(a))+offset_y*np.cos(np.deg2rad(a)),
                                        a+90)
        nd.text(text=message_2, height=text_height_om, align="cb").put(
                                        x+(offset_om_x-text_height_sem*10)*np.cos(np.deg2rad(a))-offset_y*np.sin(np.deg2rad(a)),
                                        y+(offset_om_x-text_height_sem*10)*np.sin(np.deg2rad(a))+offset_y*np.cos(np.deg2rad(a)),
                                        a+90)
        pass

    def define_curve(self):
```

```
        """
        Defines the path of the adiabatic spiral. In part written by Marek Vlk.
        :return:
        """
        ROC = self.radius_of_curvature
        pitch = self.pitch
        n_turns = self.n_turns

        # new instance of the Archimedean spiral
        arch = spi.Archimedes(ROC, pitch)

        # starting point
        (x0_in, y0_in) = (0, 0)

        # connection point
        th_inner_connect = arch.find_th_cond1() + arch.find_b_cond1()
        (x1_in, y1_in) = (arch.x(th_inner_connect), 0)
        k1_in = arch.kappa(th_inner_connect)
        dk1_in = arch.d_kappa(th_inner_connect)
        th1_in = arch.tg_angle(th_inner_connect)

        # new instance of a 3rd-order clothoid for the central S-bend
        cloth3 = spi.Clothoid_order3(x0_in, y0_in, x1_in, y1_in, k1_in, dk1_in, th1_in)

        # insert your initial estimates
        a3_in_est = 2.3
        s1_in_est = 1.8

        # solve for a3 and s1, then calculate the rest and print all...
        cloth3.solve_eq_xy(a3_in_est, s1_in_est)

        th_outer_connect = (th_inner_connect + n_turns * 2 * np.pi - (1/4.0) * np.pi)
        # starting point
        th0_out = (1/2.0) * np.pi
        (x0_out, y0_out) = (arch.x(th_outer_connect + (1/4.0)*np.pi), -4.2)

        # conection point
        (x1_out, y1_out) = (arch.x(th_outer_connect), arch.y(th_outer_connect))
        k1_out = - arch.kappa(th_outer_connect)
        dk1_out = - arch.d_kappa(th_outer_connect)
        th1_out = arch.tg_angle(th_outer_connect) - np.pi

        # new instance of a 4th-order clothoid for an adiabatic input/output connection
        cloth4 = spi.Clothoid_order4(x0_out, y0_out, x1_out, y1_out, k1_out, dk1_out, th1_out, th0_out)

        # insert your initial guesses - very important!
        a4_out_est = -0.0015
        s1_out_est = 7.3

        # solve for a4 and s1, then claculate the rest and print all...
        cloth4.solve_eq_xy(a4_out_est, s1_out_est)

        n = 1200
        s_in = np.linspace(0, cloth3.s1, n)
        s_out = np.linspace(0, cloth4.s1, n)

        th = np.linspace(th_inner_connect, th_inner_connect + n_turns * 2 * np.pi - (1/4.0) * np.pi, n)

        # new instance of curve
        crv = spi.curve(arch.x(th), arch.y(th), s_in[-1] + arch.s(th[0], th))

        # add the central clothoid
        crv.read_next(cloth3.x(s_in), cloth3.y(s_in), s_in)
        crv.match('First')
        crv.splice()

        # add the input/output clothoid
        crv.read_next(cloth4.x(s_out), cloth4.y(s_out), s_in[-1] + arch.s(th[0], th[-1]) + s_out)
        crv.match('Last')
        crv.splice('Last')

        target = self.length

        l_ext = target/2-max(crv.s)
        s_ext = np.linspace(0, l_ext, 2)

        x_ext = s_ext * np.cos(-th0_out) + x0_out
        y_ext = s_ext * np.sin(-th0_out) + y0_out

        crv.read_next(x_ext, y_ext, s_in[-1] + arch.s(th[0], th[-1]) + s_out[-1] + s_ext)
```

```
312
313         crv.splice('Last')
314
315         crv.mirror()
316
317         crv.x = np.flip(crv.x)
318         crv.y = np.flip(crv.y)
319
320         length05t = spi.length(crv.x, crv.y)
321         return crv
322
323
324 class MarkerCross:
325
326     def __init__(self, thickness=40, width=600, nameWg='Cross'):
327         """
328         Creates a simple marker cross that can be used for alignment or as a cleaving mark.
329         :param thickness: Thickness of the arms of the cross
330         :param width: Width of the cross, i.e. its extent
331         :param nameWg: The name of the cross in the GDS-file
332         """
333         with nd.Cell(name=nameWg) as mycross:
334             pointsCross = [[-width/2.0, thickness/2.0],
335                            [-thickness/2.0, thickness/2.0],
336                            [-thickness/2.0, width/2.0],
337                            [thickness/2.0, width/2.0],
338                            [thickness/2.0, thickness/2.0],
339                            [width/2.0, thickness/2.0],
340                            [width/2.0, -thickness/2.0],
341                            [thickness/2.0, -thickness/2.0],
342                            [thickness/2.0, -width/2.0],
343                            [-thickness/2.0, -width/2.0],
344                            [-thickness/2.0, -thickness/2.0],
345                            [-width/2.0, -thickness/2.0]
346                            ]
347
348             nd.Polygon(points=pointsCross, layer='layer1').put()
349         self.cross = mycross
350         pass
351
352     def put(self, x, y, a):
353         """
354         Places a cross at a given position in the GDS-file
355         :param x: The x-origin of the marker cross placement
356         :param y: The y-origin of the marker cross placement
357         :param a: The angle at which the marker cross is placed
358         :return:
359         """
360         self.cross.put(x, y, a)
361         pass
```



NTNU

Kunnskap for en bedre verden