Instituto Superior de Engenharia de Lisboa

Área Departamental de Engenharia de Eletrónica e Telecomunicações e de Computadores

Silicon waveguides analysis and simulation

by

Paulo Jorge Passos Sério Lourenço

Trabalho Final de Mestrado para a obtenção do Grau de Mestre em Engenharia Eletrónica e Telecomunicações

Orientadores:
Professor Doutor Alessandro Fantoni
Professor Doutor Pedro Pinho

Júri:
Professor Doutor Mário Pereira Véstias (Presidente)
Doutor Manuel João Mendes (Arguente)
Professor Doutor Alessandro Fantoni (Orientador)

Dezembro de 2016

Dedicatória

Ao meu filho e netos.

Agradecimentos

Antes de mais, gostaria de agradecer ao ISEL – Instituto Superior de Engenharia de Lisboa – por ter reacendido em mim o gosto pelo conhecimento e aprendizagem, gosto esse que se encontrava esquecido há muitos anos.

Um reconhecimento especial aos orientadores desta minha fase do percurso académico, nas pessoas do Professor Doutor Pedro Pinho e do Professor Doutor Alessandro Fantoni, que sempre me acompanharam, incentivaram e auxiliaram a ultrapassar os obstáculos entretanto surgidos e que tudo fizeram para que esta dissertação chegasse a bom termo.

A todos os meus professores de Licenciatura e Mestrado, pois sem eles nunca aqui teria chegado.

Aos meus pais, pela compreensão demonstrada pelas longas ausências, especialmente no ano transato.

Ao meu filho, Ricardo Bruno Costa Sério Lourenço, e netos, Ricardo Miguel Sobral Sério Lourenço e Letícia Alexandra Sobral Sério Lourenço, simplesmente por existirem e aos quais dedico este trabalho, com a esperança que um dia os venha a incentivar de alguma forma.

Resumo

Os fenómenos eletromagnéticos têm vindo a ser estudados desde a sua previsão por Maxwell no sec. XIX. A sua propagação quer em espaço livre, quer quando através de estruturas criadas para o efeito, não é exceção e, mais recentemente, o foco parece estar localizado na investigação da fotónica baseada em silício. O seu elevado índice de refração e compatibilidade com a tecnologia CMOS atual, colocam o silício em posição privilegiada como plataforma de integração de alta densidade nos processos de fabrico dos sistemas de processamento digital contemporâneos. É assim intenção desta dissertação, apresentar os primeiros passos de uma alternativa dirigida a um objetivo final, a concretização de sistemas óticos de processamento digital.

Nesse sentido e após uma introdução inicial onde são referidas as principais vantagens de uma substituição das atuais interligações elétricas por pelas suas congéneres óticas, foi efetuado um estudo analítico da formação de modos em guias de ondas dielétricos, seguido finalmente por simulações de algumas destas estruturas com vista à obtenção de dados que permitem caracterizar as perdas de potência ótica ao longo dos dispositivos apresentados. Assim, serão aqui apresentados os resultados obtidos por um conjunto de simulações FDTD (Finite-Difference Time-Domain), a duas e três dimensões, com as características de guias de onda de silício amorfo hidrogenado quando embebido em sílica. A dependência entre a absorção de radiação eletromagnética e as propriedades dos materiais, bem como, a influência do raio de curvatura do guia de ondas, é aqui analisada para comprimentos de onda do espectro infravermelho. A eficiência de transmissão dos referidos comprimentos de onda é determinada pela verificação do decaimento de potência do sinal eletromagnético ao longo dos guias de onda e os resultados obtidos mostram que as perdas por radiação estarão dentro de limites aceitáveis, mesmo quando considerando raios de curvatura tão diminutos quanto 3 μm.

Palavras Chave

Palavras chave:
Diferenças finitas no domínio do tempo (FDTD);
Silício amorfo hidrogenado.
Keywors:
Finite-difference time-domain method (FDTD);
Hidrogenated amorphous silicon.

Abstract

Electromagnetic phenomena have been studied since their prediction by Maxwell in the 19th century. Their propagation, whether in open space or through specifically created conduits, is of no exception and, lately, research interest seems very much focused on Si based photonics. Its high refraction index and compatibility with CMOS actual technology, place this element on a favourable position as a high density integration platform in digital processing systems today's manufacturing processes. Thus, it is the intent of this dissertation to present the first necessary steps towards the ultimate goal, a fully optical computer system.

Hence and after an initial introduction where the main foreseen advantages by the replacement of some of today's electrical interconnects on computer systems by their Si based optical counterparts are mentioned, follows an analytical approach to propagation modes formation and respective main parameters on dielectric waveguides and, finally, by some of these structures simulations in order to obtain data to characterize optical power loss along the simulated devices. For that purpose, it is reported results obtained by a set of FDTD (Finite-Difference Time-Domain) simulations, both 2D and 3D, about the characteristics of hydrogenated amorphous silicon waveguides embedded in a SiO_2 cladding. Light absorption dependence on the material properties and waveguide curvature radius are analysed for wavelengths in the IR (infra-red) spectrum. Wavelength transmission efficiency is determined analysing the decay of the light power along the waveguides and the obtained results show that radiation losses should remain within acceptable limits when considering curvature radius as small as 3 µm at its most.

Table of contents

Dedicatória iii

Agradecimentos iv

Resumo vii

Abstract ix

Table of figures xv

List of tables xvii

Acronyms list xix

Chapter I 1

Introduction 1

1.1. Dissertation framework 2

1.2. Dissertation motivation 2

1.3. Dissertation intent 5

1.4. Dissertation structure 5

1.5. Main contributions 6

Chapter II 7

Dielectric waveguides 7

2.1. Introduction 7

2.2. Planar-mirror dielectric waveguide 8

2.2.1. Self-Consistency (a.k.a. - also known as - Transverse Resonance) condition 10

2.2.2. Waveguide modes 11

Wave equation 12

Propagating modes 14

TM even modes 14

Cut off frequency, TM even modes 19

TM odd modes 20

Cut off frequency, TM odd modes 24

TE even modes 24

Cut off frequency, TE even modes 26

TE odd modes 26

Cut off frequency, TE odd modes 28

2.2.3. Optiwave® simulator - OptiFDTD 29

Mode Solver 29
2.3. Non-planar dielectric waveguides 31
2.3.1. Circular dielectric waveguides - Optical fibre 32
Wave equations for circular dielectric waveguides 33
Cylindrical coordinates wave equation 35
Boundary conditions 38
Normalized parameters 39
Normalized frequency 39
Normalized propagation constant 40
Effective refractive index 40
Meridional modes (TE/TM) 41
Skew/hybrid modes (HE/EH) 42
Cut off conditions 42
Linearly polarized modes (LP) 44
2.3.2. Rectangular dielectric waveguides 47
Marcatili method – Basic equations 47
Dispersion equations for $Epqx$ and $Epqy$ modes 49
Effective Index Method 51
Chapter III 57
Non-ideal photonic waveguides 57
3.1. Absorption and Extinction coefficients α and κ 58
3.2. Complex Effective Refractive Index 59
3.3. Photonic wire simulation 60
Simulation's parameters definition 62
Simulation's parameters definition 65
3.4. Photonic wire loss through radiation as arc radius decreases 66
Simulation's parameters definition 67
3.5. 3D photonic wire simulations 73
Chapter IV 81
Conclusions and future work 81
References 83
Appendix A 1
Guided waves 1

Appendix B 1
Waveguide modes representation code 1
InterX function code 7
Appendix C 1
1st kind Bessel's function 1
2nd kind Bessel's function 3
Modified Bessel function of 1st kind 4
Modified Bessel function of 2nd kind 5
Appendix D 1
LP mode code 1
EM fields function code 2
n_{eff} vs V for HE/EH and TE/TM modes code 3
Appendix E 1
Total internal reflection 1
Guided modes formation 2
Dispersion equations code 5
Appendix F 1
E^x and E^y modes plot code 1
Contourmode function code 2
Postprocess function code 3
Stretchmesh function code 11
Waveguidemeshfull function code 14
Wgmodes function code 15
Appendix G 1
Power read function code 1
Leastsquares function code 1

Table of figures

Figure 1 - Hybrid NoC [1]. 3
Figure 2 - Hybrid NoC unit cell [1]. 4
Figure 3 - Waveguide profile indexes. 8
Figure 4 - Total internal reflection on dielectric waveguides [3]. 9
Figure 5 - Interference pattern created by d/λ = 4.5 and $\boldsymbol{\theta}$ = 65.8° [3]. 9
Figure 6 - Self-Consistency condition [3]. 10
Figure 7 - Device: planar waveguide dimensions. 11
Figure 8 – Core and cladding propagation constants [5]. 12
Figure 9 – TM_{even} found modes @ 1550 nm. 16
Figure 10 - Functions intersection gives characteristic value h_1. 17
Figure 11 - Normalised profiles for the 3 modes. 17
Figure 12 - Fundamental TM_{even} modes for different wavelengths. 18
Figure 13 - Zoom in portion of Figure 10. 19
Figure 14 – TM_{odd} found modes @ 1550 nm. 22
Figure 15 - Fundamental TM_{odd} modes for different wavelengths. 23
Figure 16 - Zoom in portion of Figure 13. 23
Figure 17 – TE_{even} found modes @ 1550 nm. 25
Figure 18 – TE_{odd} found modes @ 1550 nm. 27
Figure 19 - 1.55 $\boldsymbol{\mu}$m wavelength Gaussian Modulated Continuous Wave. 29
Figure 20 - Mode Solver 2D, TE found modes. 30
Figure 21 - Mode Solver 2D, $\lambda = 10\ \mu m$ TE found modes. 30
Figure 22 - Non-planar dielectric waveguides [5]. 31
Figure 23 - Optical fibre diagram. 33
Figure 24 - Effective refractive index behavior as normalized frequency increases. 40
Figure 25 - Meridional modes (TE and TM). 41
Figure 26 - Skew or hybrid modes (EH or HE modes). 42
Figure 27 - Fundamental (LP01 or HE11) mode electrical components. 45
Figure 28 - HE21 mode electrical components. 45
Figure 29 - TM01 mode electrical components. 46
Figure 30 - LP11 mode electrical components. 46
Figure 31 - Photonic wire seen by Marcatili method. 47
Figure 32 - Marcatili method E_x and E_y mode distribution. 50
Figure 33 - Rib waveguide. 52
Figure 34 - Actual refractive index profile n(x,y) variation. 53
Figure 35 - Effective index distribution, n_eff(x). 54
Figure 36 - Designing simulation workspace on OptiFDTD. 62
Figure 37 - DFT of E_y field and Poynting vector ($\boldsymbol{\lambda}$ = 1500 nm). 63
Figure 38 - Observation lines intensity comparison. 64
Figure 39 - DFT of Ey field and Poynting vector ($\boldsymbol{\lambda}$ = 880 nm). 66
Figure 40 - OptiDesigner workspace; inset: - 2D refraction index. 66
Figure 41 - Design conditions (radius = 5 um). 70

Figure 42 - Observed power over script simulation (100 samples). 70
Figure 43 - Arc overall power loss in dB. 71
Figure 44 - ObservationLine3 power loss by path unit. 72
Figure 45 - ObservationLine3 trend (least squares algorithm). 72
Figure 46 - Symmetric structure refractive indexes. 74
Figure 47 - Asymmetric structure refractive indexes. 74
Figure 48 - Symmetric structure results. 75
Figure 49 - Asymmetric structure results. 75
Figure 50 - OptiDesigner workspace (radius = 12.9 µm); inset: - 3D layout model. 76
Figure 51 - Field spot position vs. ObservationArea3's centre (radius=12,3 μm). 76
Figure 52 - Field spot position vs. ObservationArea3's centre (radius=12,4 µm). 77
Figure 53 - Field spot position vs. ObservationArea3's centre (radius=12,5 µm). 77
Figure 54 - Field spot position vs. ObservationArea3's centre (radius=12,6 µm). 77
Figure 55 - Field spot position vs. ObservationArea3's centre (radius=12,7 µm). 78
Figure 56 - Field spot position vs. ObservationArea3's centre (radius=12,8 µm). 78
Figure 57 - Field spot position vs. ObservationArea3's centre (radius=12,9 µm). 78
Figure A1 - Cylindrical waveguide. 1
Figure C1 - First kind Bessel's functions. 1
Figure C2 - Zero crossing for Bessel functions. 2
Figure C3 - Bessel function derivatives and crossing points. 3
Figure C4 - Second kind Bessel's functions. 4
Figure C5 - First kind modified Bessel function. 4
Figure C6 - Second kind modified Bessel function. 5
Figure E1 - Total internal reflection. 1
Figure E2 - Confined light rays and their phase fronts. 2
Figure E3 - Fundamental mode (m=0). 4
Figure E4 - Higher order mode (m = 1). 4
Figure E5 - Normalized frequency Vs. normalized propagation constant. 5

List of tables

Table 1 - Characteristic parameters for all modes [4]..... 28
Table 2 - Modes cut off conditions..... 44
Table 3 - LP modes and their constituents..... 47
Table 4 - Waveguide profile and materials for 2D simulation @ 1500 nm..... 62
Table 5 - Input field parameters for 2D simulation @ 1500 nm..... 62
Table 6 - Observation lines for 2D simulation @ 1500 nm..... 63
Table 7 - Linear waveguide parameters for 2D simulation @ 1500 nm..... 63
Table 8 - Waveguide profile and materials for 2D simulation @ 880 nm..... 65
Table 9 - Input field parameters for 2D simulation @ 880 nm..... 65
Table 10 - Observation lines for 2D simulation @ 880 nm..... 65
Table 11 - Linear waveguide parameters for 2D simulation @ 880 nm..... 65
Table 12 - Profile and materials parameters for 2D simulation..... 67
Table 13 - Input field parameters for 2D simulation..... 67
Table 14 - Observation lines parameters for 2D simulation..... 67
Table 15 - Linear waveguide 1 parameters for 2D simulation..... 67
Table 16 - Arc waveguide parameters for 2D simulation..... 67
Table 17 - Linear waveguide 2 parameters for 2D simulation..... 68
Table 18 - Simulation wafer parameters..... 68
Table 19 - Power loss induced by arc radius decrease..... 69
Table 20 - Normalisation of radiation per path unit..... 69
Table C1 - First kind Bessel functions zero crossings..... 2
Table C2 - First kind Bessel's derivative function zero crossings..... 3

Acronyms list

Acronym	Definition
2D	Bi-dimensional
3D	Tri-dimensional
ACK	Acknowledge
APML	Anisotropic Perfectly Matched Layers
a-Si	Amorphous silicon
a-Si:H	Hydrogenated amorphous silicon
CMOS	Complementary metal–oxide–semiconductor
c-Si	Crystalline silicon
CVD	Chemical Vapour Deposition
EH	Electric/Magnetic hybrid mode
EI	Electrical Interconnects
EM	Electromagnetic
FEM	Finite-Element Method
FDM	Finite-Difference Method
FDTD	Finite-Difference Time-Domain
GMCW	Gaussian Modulated Continuous Wave
GUI	Graphical User Interface
HE	Magnetic/Electric hybrid mode
LP	Linearly Polarized
MAM	Modal Analysis Method
NoC	Network-on-Chip
OFT	Optical Fibre Toolbox
OI	Optical Interconnects
PBC	Periodic Boundary Conditions
PEC	Perfect Electrical Conductor
PECVD	Plasma Enhanced Chemical Vapour Deposition
PMC	Perfect Magnetic Conductor
p-Si	Poly-crystalline silicon
RF	Radio Frequency
SOI	Silicon on Insulator
TE	Transverse Electric
TEM	Transverse Electromagnetic
TIR	Total Internal Reflection
TM	Transverse Magnetic

Chapter I

Introduction

The never ending demand for bandwidth of today's computer systems are being strained by the limits imposed, between other reasons, by their interconnections material physical properties. The widely used material linking components in computer systems today is Cu (Copper). This material is deposited on a substrate forming, at the end of the process, a thin layer conduit that acts as a waveguide for information data and power transportation.

As far as power transport is concerned and considering the energy levels used on actual systems and their decreasing trend, there seems to be no constraints. On the other hand, one cannot expect the same when the transport of information data is involved. As mentioned before, there is an increasing trend on bandwidth demand and future systems will have to find a solution, as processors clock increase, before the compromise between high bandwidth and power dissipation due to skin effect is no more sustainable and, also, the delay factor introduced by metallic waveguides, making it harder to balance the time a processor takes to perform an operation versus data access delay by the same component.

Optical propagation speed is almost the speed of light in vacuum (RC delay non-existent), the large bandwidth associated to the high frequencies of optical signals allows for much larger amounts of data to be transported by time unit and minimal interaction between photons from different signals lead to almost non-existent cross-talk. All these advantages make OI (Optical Interconnects) a serious candidate for EI (Electrical Interconnects) replacement.

Thus, it is the aim of this chapter to propose OI as an alternate solution over EI for on-chip data exchange.

1.1. Dissertation framework

Silicon (Si) is the widely used semiconductor in most of actual electronic devices because of its excellent electronic properties and low cost. Availability of high quality c-Si (crystalline silicon) together with transparent frequency behaviour at telecommunications wavelengths, places Si also as an ideal optical material. Using Si-based waveguides to distribute optical waves, would allow the integration of optical systems into existing integrated circuits as optical interconnects of various kinds (i.e. memory access, interconnects between processors in multiprocessor systems, clock distribution systems…).

In order to have an optical waveguide able to propagate light, it requires a core with higher refractive index than the surroundings. This is where a-Si (amorphous silicon) can play its part as an interesting core candidate for its refractive index at near-infrared wavelengths is higher than that of c-Si. However, pure a-Si has high density of point defects and dangling bonds that lead to an increase of optical absorption. Nevertheless, these defects can be passivated to some extent by the incorporation of hydrogen into the material, originating a-Si:H (Hydrogenated Amorphous Silicon) with excellent electrical quality and lower optical absorption.

There are several processes in use today through which hydrogenated a-Si films can be deposited, namely, plasma enhanced chemical vapour deposition (PECVD), RF (Radio Frequency) sputtering and chemical vapour deposition (CVD). These depositing techniques can also be used to develop photonic crystal structures but this subject is not going to be approached in this dissertation.

1.2. Dissertation motivation

Silicon photonics can be generally defined as the use of silicon based materials to generate, control, transfer and detect light over short distances (centimetres being the furthest) in order to enable communications between processors. The main motivation for the use of OI instead of copper based EI, is due to power and delay limitations on the latter. As device sizes have shrunk according to Moore's law (generally speaking), ultra-short distance communications on chips have been dominated by EI, mainly due to relative design and fabrication ease and, so far met, power and delay requirements at low cost.

The innovations in lithography technology have reached miniaturization beyond $35\,nm$ and transistor delay of $1\,ps$. However, as processors speeds increase it is getting harder to keep balance between on-chip logic operations and read/write data into and out of memory, which is mainly due to RC delay in electrical interconnections and skin effect related power losses. The latter can be reduced by increasing EI width but this worsens bandwidth density.

OI are good alternative candidates for on-chip data exchange. They have many of their characteristics in favour when compared to EI, namely, propagation speed is close to the speed of light in vacuum and large bandwidth that these high frequencies are able to handle. Plus, light short wavelength can propagate through dielectric waveguides with low losses. Also, light is practically immune to transmitted signals cross-talk, which is not the case when EI are involved.

Nevertheless, using OI for communications between processors implies optical/electrical and vice-versa conversions, which add delay to the process. This delay is always the same despite distance

and has been shown that OI are able to achieve lower power and latency than EI, as long as the distances involved are over 1000 times the wavelength [1].

For all the above mentioned reasons, the International Technology Roadmap for Semiconductors in 2007 envisioned the integration of OI for long communications in upper metal layers of CMOS (Complementary metal–oxide–semiconductor) fabrication process. Also, NoCs (Networks-on-Chip) have been proposed to replace traditional electronic busses. Although electronic NoCs have improved in terms of latency and bandwidth, power consumption remains critical because of high switching activity in EI. Introducing photonics on NoCs is predicted to be a more attractive method for it simultaneously achieves low power consumption and latency and high bandwidth.

Studies and tests have been done in this area, i.e., a global on-chip optical bus to get memory access from a multi-core processor and the results showed 50% latency reduction on some applications and close to 30% power reduction over an electrically based bus [1]. Also, it has been presented a hybrid photonic NoC, integrating a photonic circuit-switched network for message transmission and electronic packet-switching for routing path setup, similar to the one depicted on Figure 1.

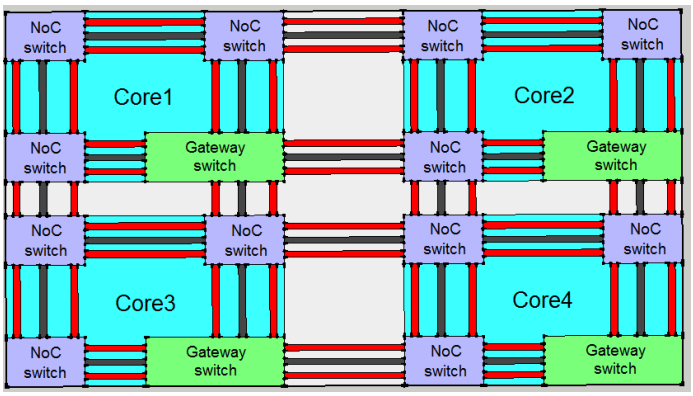

Figure 1 - Hybrid NoC [1].

This NoC can be operated much like known IP based networks and the path taken by a typical message between two cores will be as follows:

- When the core determines the amount of data to be transmitted to the following core, an electronic path setup packet, containing source and destination addresses, photonic message length, error and priority information, is broadcasted over the electronic network. As this electronic path setup propagates along the network, the photonic routers reconfigure themselves in order to establish the best path between source and destination. When destination router is reached, it sends back an ACK (Acknowledge) message to the source through the path created in the meantime by the electronic path setup packet. The source receives the ACK message and knows now there is a path ready to start sending the information data through the photonic network.

In the hybrid NoC architecture, physical location of the photonic/electronic integration on chips fabrication processes, in order to achieve maximum flexibility and minimal complexity and cost, is of utmost importance. Photonic components integration on the upper metal layers of CMOS fabrication process has come forward as a possible and feasible solution.

Compatibility between silicon photonics and CMOS processes used in the development of today's processors, has highly contributed to its fast emergence as a cost-effective solution for the integration of photonic and electronic circuits on a single platform. Some work has been done in the area and integrated modulators and detectors, that form the core of an OI network, have been fabricated and presented [2]. Initially, these devices have been fabricated on single layer c-Si substrate and on SOI (Silicon On Insulator) platform but, with increasing device density, future generations of photonics NoCs will have to grow vertically.

Vertical stacking of c-Si layers is fabrication intensive and involves multiple implantation and epitaxial growth processes for it is not likely to be cost effective. Moreover, c-Si electronic structure is anything but flexible and that adds even more complexity to the process. As so, two alternative solutions have been proposed [1]:

- Poly-crystalline silicon (p-Si).
- Amorphous silicon (a-Si).

Although p-Si is a viable solution, its higher optical losses due to scattering at imperfections and grain boundaries is a downside. Also, backend integration of photonic devices is thermally limited for the interconnect architecture. In order to achieve device quality substrate, p-Si films need to be annealed at over 600 °C which is well above than the 400 °C limit imposed by CMOS technology.

a-Si:H can be deposited on virtually any material using low temperature (250 – 400 °C) PECVD for this is fully compatible with backend thermal budget in CMOS fabrication process, turning a-Si:H the material of choice for integration on a hybrid NoC.

Figure 2 shows a diagram of a proposed photonic/electronic hybrid NoC unit cell using a-Si:H. CMOS transistor is embedded in the bottom layer and is connected to the photonic circuitry through the metal layer. The photonic circuit consists of $460 \times 250\ nm$ cross section a-Si:H photonic wires (waveguides). These wires are functionally identical to metallic wires for they can be of any length and also make sharp bends. There is, however, a fundamental difference: - They require no power to propagate the signal. Once light is coupled into the waveguides, it will propagate at virtually the speed of light and can even cross close by other waveguides with minimal cross-talk interference. This cell is formed by the necessary elements to operate as a data transmission/reception unit: - A laser, a modulator and a photo-detector. The whole cell unit structure is based on SOI with embedded SiGe regions for transistor strain and growing laser and photodetector materials.

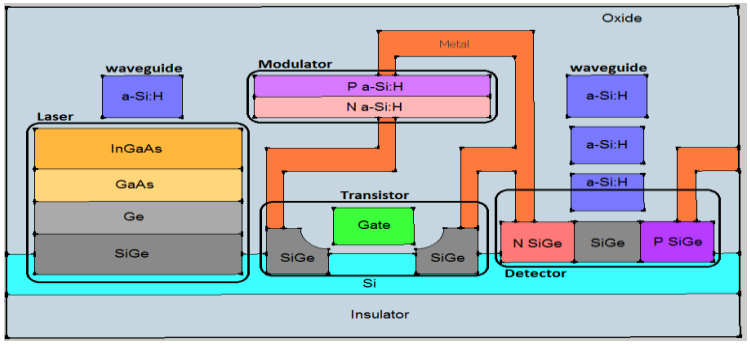

Figure 2 - Hybrid NoC unit cell [1].

1.3. Dissertation intent

The main purpose of this dissertation is to explore the first steps towards a broader ultimate goal, a fully optical computer system. In order to achieve this, several stages were approached, namely:

1. A qualitative approach on a possible implementation of hybrid networks-on-chip.
2. A detailed study on propagation modes formation and fundamental parameters on ideal planar, circular and rectangular dielectric waveguides.
3. Simulations considering dispersion and absorption of a-Si:H as base material for core waveguide, evaluating power losses over distance on a straight waveguide and as bend radius decreases.

1.4. Dissertation structure

This dissertation structure is organized in four main chapters, each of them focusing on one of the different subjects that, together, form alternative first steps towards the ultimate goal: - The fundamentally optical computer system.

Hence, chapter one is introductory and references the main reasons why this study was initiated, driven by the motivation based on advances in lithographic technologies recently available and respective manufacturing process devices physical constraints, as far as access delay and power losses are concerned, towards the first steps exploring EI's alternative and more efficient interconnections. Thus, implementation of OI on actual computer systems is presented, focusing on a possible solution known as a hybrid NoC (Network-on-Chip). It considers a network of interconnected processors where the data path is initially established electronically (EI) and then data is actually transported through dielectric waveguides, thus the name hybrid. Also in this chapter, it is considered a hybrid integration unit cell with a-Si:H as the base material for the core waveguide, responsible for the interface processor/waveguides.

Chapter two provides all the mathematic and theoretical background that supports light propagation phenomenon in a dielectric waveguide, namely modes propagation and their parameters. This study starts with planar-mirror dielectric waveguides, moves on to circular dielectric waveguides (optical fibre) and, finally, rectangular dielectric waveguides are considered. It is also in this chapter that OptiFDTD, from OptiWave®, is used to simulate a planar-mirror waveguide and respective propagating waveguide modes.

On chapter three, using OptiFDTD (OptiWave®), several simulations took place in order to evaluate how losses may affect power when absorption is to be taken into account. Such is the case when using a-Si:H as the elected material for the core waveguide, leading to two main evaluations:

1. Simulate a straight waveguide and determine power loss (due to absorption) over distance. Two wavelengths were considered ($880\,nm$ and $1500\,nm$) in order to verify these devices wavelength dependency.
2. Simulate a bent waveguide and determine power loss (due to absorption and/or radiation) as bend radius decreases.

Chapter four, as this dissertation last chapter, is where main achieved conclusions and found limitations on the developed study are presented and where are also considered future work suggestions.

1.5. Main contributions

In the framework of this MSc dissertation, the following original contribution was submitted and accepted in the CETC'2106 for oral presentation and selected for further full-paper submission and revision to the i-ETC journal.

- Paulo Lourenço, Alessandro Fantoni, Pedro Pinho, "*Simulation of Amorphous Silicon Waveguides*", Conference on Electronics, Telecommunications and Computers, Lisbon, Portugal, 6th – 7th December 2016.

Chapter II

Dielectric waveguides

This chapter presents all the mathematic and theoretical background that supports light propagation phenomenon in a dielectric waveguide, namely modes propagation and their parameters. This study starts with planar-mirror dielectric waveguides, moves on to circular dielectric waveguides (optical fibre) and, finally, rectangular dielectric waveguides are considered. It is also in this chapter that OptiFDTD, from OptiWave®, is used to simulate a planar-mirror waveguide and respective propagating waveguide modes.

2.1. Introduction

All dielectric waveguides are built in a way that a material, with a higher refractive index than the surrounding medium, allows EM (Electromagnetic) wave propagation based on total internal reflection phenomena. Here, the central medium acts as a light "trap" within which optical rays remain confined by multiple internal reflections at its boundaries; a propagating optical wave travels through the wave guide. This behaviour, together with medium's low loss characteristics, are then exploited to make light conduits, enabling light transportation from one location to another (sometimes hundreds or even thousands of kilometres apart from each other) with low propagation losses.

As far as the refractive index profile of a waveguide is concerned, there are mainly two types as graphically depicted on Figure 3:

1. **Step-index waveguide**: - where the refractive index profile varies abruptly between core and surrounding medium;
2. **Graded-index waveguide**: - where the refractive index profile varies gradually between core and the surrounding medium.

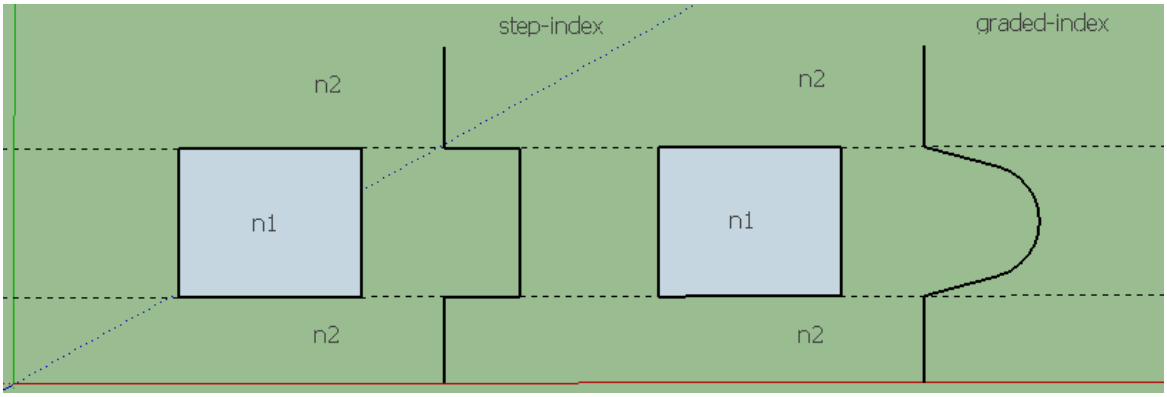

Figure 3 - Waveguide profile indexes.

This dissertation is going to focus only on step-index waveguides, starting with the study of these devices under ideal conditions - 2.2. - and then moving on to real conditions materials - 3. - where dispersion and attenuation will also be taken into account. The first device to be studied will be the simplest one – the planar-mirror dielectric waveguide. Then, it will move on to circular and rectangular waveguides on a substrate still under ideal conditions and, finally, using dispersive material such as hydrogenated amorphous silicon on a substrate in order to characterize power losses on simulated devices.

The perfect dielectric waveguide is assumed to be as if it were made of optically isotropic medium and its boundaries were perfect mirrors, as long as light is concerned and the critical angle ($\boldsymbol{\theta}_c$) of incidence is not overcome. Thus, a ray of light making an angle $\boldsymbol{\theta} > \boldsymbol{\theta}_c$ with the mirrors' normal at the point of incidence, reflects and bounces between the mirrors without loss of energy. These waveguides have no practical application mainly because of cost and difficulty of fabricating loss and dispersion free media but provide an excellent pedagogical approach to light´s behaviour when propagating through waveguides.

2.2. Planar-mirror dielectric waveguide

A planar-mirror dielectric waveguide, as defined in [3], has optical confinement only in one transverse direction and the central medium, which is designated by core, is surrounded by the outer media which is formed by the upper (cladding) and lower (substrate) layers. Often, on planar-mirror dielectric waveguides, it is assumed that cladding and substrate are of the same material and they are referred simply as the surrounding medium. This will be the convention used in this dissertation unless specifically stated.

Under the conditions mentioned before in this chapter on 2.1., light propagates longitudinally (along z-axis) in the waveguide through multiple reflections on its boundaries, as shown on Figure 4. These multiple reflections that result in light guidance and known as TIR (Total Internal

Reflection), do not explain by itself, a number of important effects. In order to describe these, some electromagnetic theory is required. Usually, to carry out an EM analysis, each optical ray is considered as a TEM (Transverse Electromagnetic) plane wave. Then, the total EM field is the sum of these plane waves.

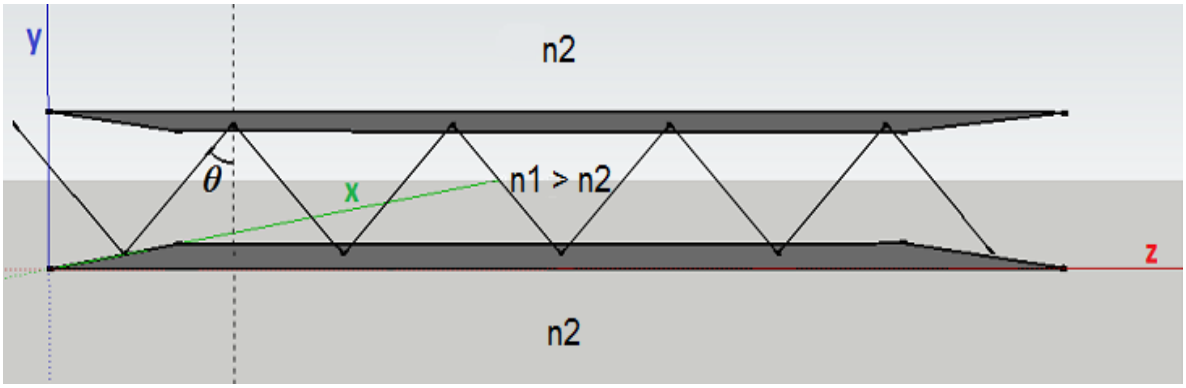

Figure 4 - Total internal reflection on dielectric waveguides [3].

Let us consider a monochromatic TEM plane wave, where ***n*** is the refractive index of the medium between the mirrors. The electric field is parallel to the reflecting surfaces (wave polarized in the *x* direction) and its wave vector lies in the *y*-*z* plane at an angle ***θ*** with the *z* axis. Like the optical ray, the TEM planar wave reflects from the upper mirror, travels at an angle ***θ***, reflects from the lower mirror, progresses at an angle ***θ***, bounces from upper mirror, journeys once more at an angle ***θ*** and so on. Each reflection introduces a wave's phase shift of ***π*** but polarization and amplitude remain the same. The ***π*** phase shift ensures that the total field at the mirrors is zero for the sum of each wave with its own reflection vanishes at this point. So, for each point within the waveguide, there are TEM waves travelling up and downwards at ***θ*** angles. Considering an upward and a downward travelling waves with the same ***θ*** angle, they interfere with each other, as shown on Figure 5, and if 2.2.1. is verified, they will create a transverse field pattern whose amplitude and polarization profiles remain constant along the longitudinal coordinate (*z* axis), thus propagating through the waveguide. This transverse field pattern is designated a waveguide mode and is a characteristic of a particular waveguide structure.

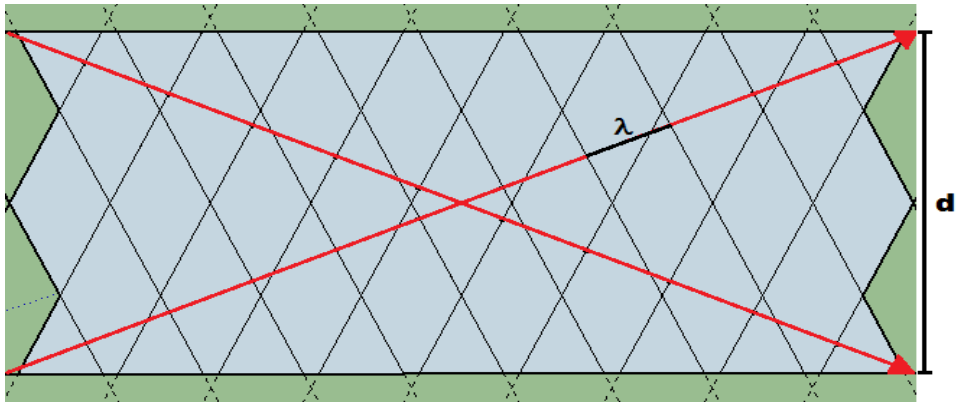

Figure 5 - Interference pattern created by $d/\lambda = 4.5$ and $\theta = 65.8°$ [3].

2.2.1. Self-Consistency (a.k.a. - also known as - Transverse Resonance) condition

Based on [3] and referring to Figure 6, the phase shift suffered by the original wave when travelling from A to B, must be the same, or affected by an integer multiple of $2\boldsymbol{\pi}$, from that encountered when the wave progresses from A to C and reflects once again, if constructive interference is to be verified. Knowing that, for each reflection, a $\boldsymbol{\pi}$ phase shift is introduced on the wave and this can be mathematically explained by the equation:

$$\frac{2\pi\overline{AC}}{\lambda} - 2\pi = \frac{2\pi\overline{AB}}{\lambda} + 2\pi q \qquad \boldsymbol{q} = 0,1,2,\ldots \tag{1}$$

Since,

$$AC - AB = 2d\sin\boldsymbol{\theta}\,, with\ d\ being\ the\ waveguide\ width$$

Equation 1 will become,

$$\frac{2\pi}{\lambda}2d\sin\boldsymbol{\theta} = 2\pi(q+1)$$

$$\frac{2\pi}{\lambda}2d\sin\boldsymbol{\theta} = 2\pi m \qquad \boldsymbol{m} = 1,2,3,\ldots \tag{2}$$

Where,

$$m = q + 1.$$

The Self-Consistency condition is, therefore, only satisfied for selected bouncing angles $\boldsymbol{\theta} = \boldsymbol{\theta}_m$ that verify equation 3,

$$\boxed{\sin\theta_m = m\frac{\lambda}{2d} \qquad \boldsymbol{m} = 1,2,3,\ldots} \tag{3}$$

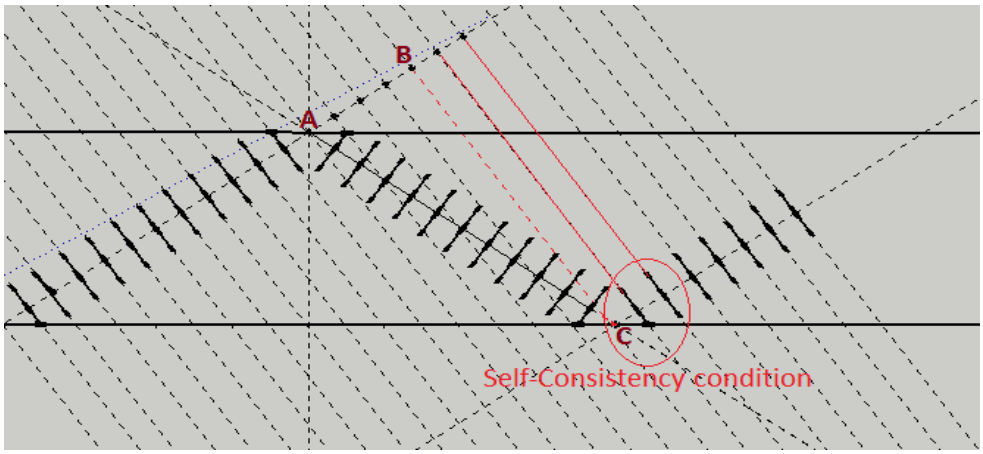

Figure 6 - Self-Consistency condition [3].

After all the previous considerations, one is able to realize that, on a planar-mirror dielectric waveguide, there might be several transverse field patterns whose amplitude and polarization profiles remain constant along the longitudinal coordinate (*z* axis). These field patterns are designated waveguide modes and their number depends on the dielectric waveguide characteristics (index of refraction of both, waveguide and surrounding medium and waveguide's width), the wave's angle of incidence $\boldsymbol{\theta}$ and its wavelength $\boldsymbol{\lambda}$ (for a detailed description of TIR and formation of propagating modes in a waveguide, please see Appendix E).

2.2.2. Waveguide modes

To provide this subject a better understanding, it will be described with the help of the diagram on Figure 7. This procedure is based on the developed work on [4] and will start by finding the fields' longitudinal components through solving the proper wave's equations and, then, obtaining the fields' transversal components. In this case, there are also the boundary conditions to be taken into account.

At the media's interfaces, because there are no superficial electrical charges nor current densities, the following must be accomplished:

Boundary conditions

- E_{tan} continuous
- B_{normal} continuous
- D_{normal} continuous
- H_{tan} continuous

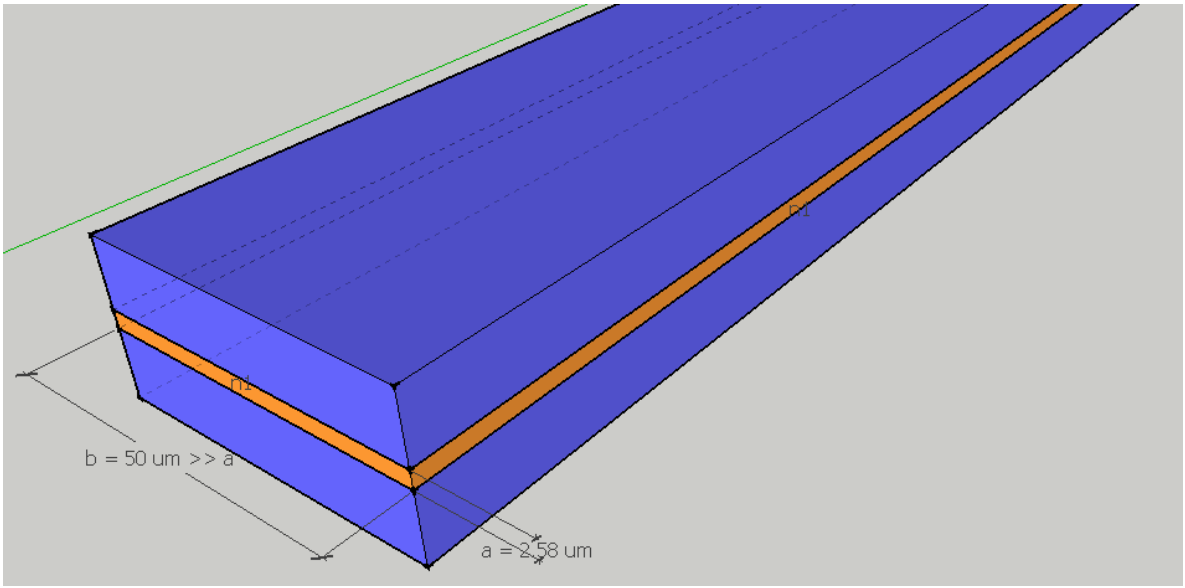

Figure 7 - Device: planar waveguide dimensions.

Wave equation

The general wave equation that explains longitudinal components behaviour is going to be analysed and the following parameters refer to Figure 8. Being *Ψ(y)* a function that verifies a linear 2nd order differential equation with constant coefficients:

$$\frac{\partial^2 \Psi}{\partial y^2} + h^2 \Psi = 0 \qquad where\ h^2 = \begin{cases} h_1^2, & |y| \leq {}^{a}/_{2} \\ h_2^2, & |y| > a \end{cases}$$

Considering Figure 8, being a the height of the core and depending on h, this equation will have two solutions:

1. $h^2 > 0$ → h is real and the solution varies periodically with *y*,

$$\Psi = A \sin(hy) + B \cos(hy)$$

2. $h^2 < 0$ → h is imaginary and the solution varies exponentially with *y*,

$$\Psi = Ce^{-vy} + De^{vy}, \qquad where\ h = jv$$

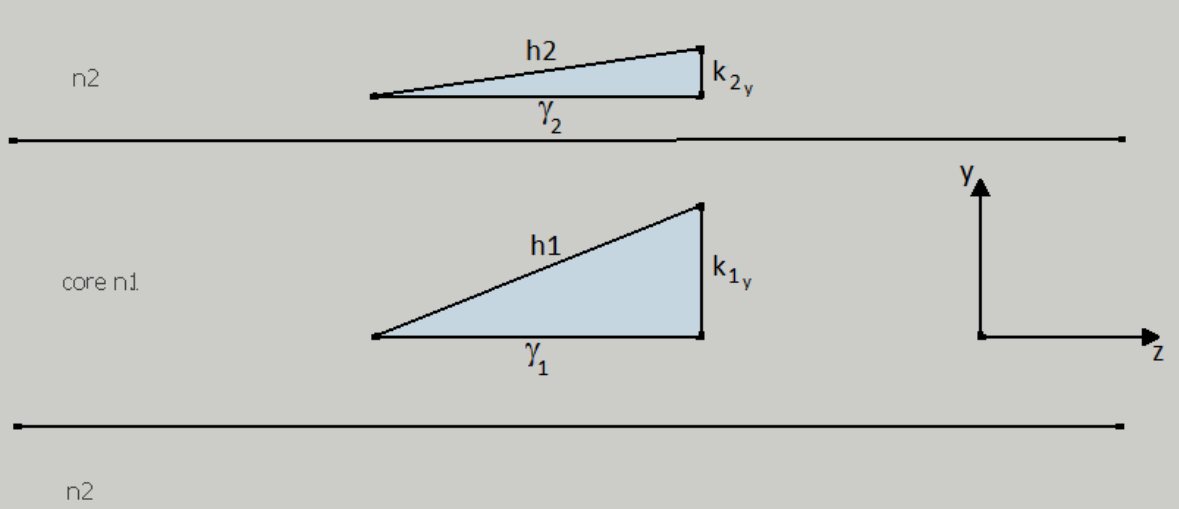

Figure 8 – Core and cladding propagation constants [5].

Thus, for a dielectric device to be able to guide an EM wave, the above mentioned assumptions must be verified; namely, fields must be periodic in the core (h is a real number) and exponential and decreasing intensity as distance to core increases, in the surrounding media (h is a complex number).

Like so, with h_1 real and $h_2 = jv$,

$$h = \begin{cases} h_1^2 = \gamma^2 + \left(\frac{\omega}{c} n_1\right)^2 \\ v^2 = -\gamma^2 - \left(\frac{\omega}{c} n_2\right)^2 \end{cases}$$

h_1 and v (exponential decay coefficient) can now be related to obtain:

$$v = \sqrt{\left(\frac{\omega}{c}\right)^2 (n_1^2 - n_2^2) - h_1^2} \tag{4}$$

Moreover, on a propagating wave $\gamma = j\beta \rightarrow \gamma^2 = -\beta^2$, which leads to,

$$\beta = \sqrt{\left(\frac{\omega}{c} n_1\right)^2 - h_1^2} = \sqrt{\left(\frac{\omega}{c} n_2\right)^2 + v^2}$$

It can now be concluded that, on a propagating mode, the phase constant β will be within the limits:

$$\frac{\omega}{c} n_2 < \beta < \frac{\omega}{c} n_1$$

The above mentioned limits are the phase constants of a plane wave, propagating on limitless media with refraction indexes n_2 and n_1, respectively. One can also infer from previous inequalities that a propagating mode requires $n_1 > n_2$, which is according to the TIR phenomenon, previously mentioned in this dissertation. Using the obtained values, a new $\Psi(y)$ definition can be written,

$$\Psi(y) = \begin{cases} De^{vy} & y < -a/2 \\ A\sin(h_1 y) + B\cos(h_1 y) & |y| \leq a/2 \\ Ce^{-vy} & y > a/2 \end{cases} \tag{5}$$

Because this function represents E_z^0 for TM (Transverse Magnetic) modes and H_z^0 for TE (Transverse Electric) modes, it has to be made sure that it cannot reach infinity and $\Psi(y)$ is a continuous function where $y = \pm a/2$. Meaning,

$$\begin{cases} A\sin\left(\frac{h_1 a}{2}\right) + B\cos\left(\frac{h_1 a}{2}\right) = Ce^{-v\frac{a}{2}} \\ -A\sin\left(\frac{h_1 a}{2}\right) + B\cos\left(\frac{h_1 a}{2}\right) = De^{-v\frac{a}{2}} \end{cases}$$

or,

$$\begin{cases} C = \left[A\sin\left(\frac{h_1 a}{2}\right) + B\cos\left(\frac{h_1 a}{2}\right)\right] e^{v\frac{a}{2}} \\ D = \left[-A\sin\left(\frac{h_1 a}{2}\right) + B\cos\left(\frac{h_1 a}{2}\right)\right] e^{v\frac{a}{2}} \end{cases}$$

Replacing previous results on equation 5,

$$\Psi(y) = \begin{cases} \left[-A\sin\left(\frac{h_1 a}{2}\right) + B\cos\left(\frac{h_1 a}{2}\right)\right] e^{v\left(y+\frac{a}{2}\right)} & y < -a/2 \\ A\sin(h_1 y) + B\cos(h_1 y) & |y| \le a/2 \\ \left[A\sin\left(\frac{h_1 a}{2}\right) + B\cos\left(\frac{h_1 a}{2}\right)\right] e^{-v\left(y-\frac{a}{2}\right)} & y > a/2 \end{cases} \quad (6)$$

Equation 6 explains the propagation modes in the waveguide and it can be further analysed through a parity approach of their longitudinal components. Thus, even modes will be the ones where *Ψ(y)* is an even function or, which is the same, equation 6 with amplitude A set to zero, resulting in the following expression:

$$\Psi_{even}(y) = \begin{cases} B\cos\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} & y < -a/2 \\ B\cos(h_1 y) & |y| \le a/2 \\ B\cos\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} & y > a/2 \end{cases} \quad (7)$$

Also, odd modes will be the ones where *Ψ(y)* is an odd function, or amplitude B set to zero in equation 6:

$$\Psi_{odd}(y) = \begin{cases} A\sin\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} & y < -a/2 \\ A\sin(h_1 y) & |y| \le a/2 \\ A\sin\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} & y > a/2 \end{cases} \quad (8)$$

Propagating modes

Next step is to use the obtained equations 7 and 8 on the wave equations *Ψ(y)* (see Appendix A for a detailed description of guided waves), considering TE and TM modes for each case.

TM even modes

In this case, $H_z^0 = 0$ and $E_z^0 = \Psi_{even}$. The other constituents are calculated using:

$$H_x^0 = \frac{j\omega\varepsilon}{h^2}\frac{\partial E_z^0}{\partial y}$$

$$E_y^0 = -\frac{\gamma}{h^2}\frac{\partial E_z^0}{\partial y}$$

$$E_x^0 = H_y^0 = 0$$

or, in the considered dielectric structure,

$|y| \leq \frac{a}{2}$:

$$\begin{aligned} E_z^0 &= B\cos(h_1 y) \\ H_x^0 &= -\frac{j\omega\varepsilon_1}{h_1} B\sin(h_1 y) \\ E_y^0 &= \frac{j\beta}{h_1} B\sin(h_1 y) \end{aligned}$$

$y > \frac{a}{2}$:

$$\begin{aligned} E_z^0 &= B\cos\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ H_x^0 &= \frac{j\omega\varepsilon_2}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ E_y^0 &= -\frac{j\beta}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \end{aligned}$$

$y < -\frac{a}{2}$:

$$\begin{aligned} E_z^0 &= B\cos\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ H_x^0 &= -\frac{j\omega\varepsilon_2}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ E_y^0 &= \frac{j\beta}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \end{aligned}$$

To come up with these expressions, $\gamma = j\beta$ and $h_2^2 = -v^2$ have been used. Although this solution complies with the longitudinal component of the electrical field continuity, it has not been made sure that the other components comply with the other boundary conditions. Namely, H_x^0 which is a magnetic field component parallel to the interfaces and there is not any superficial current, continuity must be also assured. This means,

$$-\frac{j\omega\varepsilon_1}{h_1} B\sin\left(\frac{h_1 a}{2}\right) = \frac{j\omega\varepsilon_2}{v} B\cos\left(\frac{h_1 a}{2}\right)$$

or equivalently,

$$v = -h_1 \left(\frac{n_2}{n_1}\right)^2 \cot\left(\frac{h_1 a}{2}\right) \tag{9}$$

Combining equations 9 and 4, the characteristic equation can be obtained,

$$\left(\frac{n_1}{n_2}\right)^2 \sqrt{\left(\frac{\omega}{c} n_2\right)^2 \left[\left(\frac{n_1}{n_2}\right)^2 - 1\right] - h_1^2} = -h_1 \cot\left(\frac{h_1 a}{2}\right) \tag{10}$$

This equation can be used to determine h_1 values for a specific waveguide and frequency. Unfortunately, this is a non-linear equation and its solution can only be found through numerical methods. Figure 9 shows functions $\left(\frac{n_1}{n_2}\right)^2 \sqrt{\left(\frac{\omega}{c} n_2\right)^2 \left[\left(\frac{n_1}{n_2}\right)^2 - 1\right] - h_1^2}$ and $-h_1 \cot\left(\frac{h_1 a}{2}\right)$ that have been plotted with Matlab help (see Appendix B). As can be observed, the solutions of equation 10

are finite and depend of the operating frequency (ω more precisely). Once h_1 values are known, v can be calculated through equation 9 and, finally, the EM fields' expressions can be written.

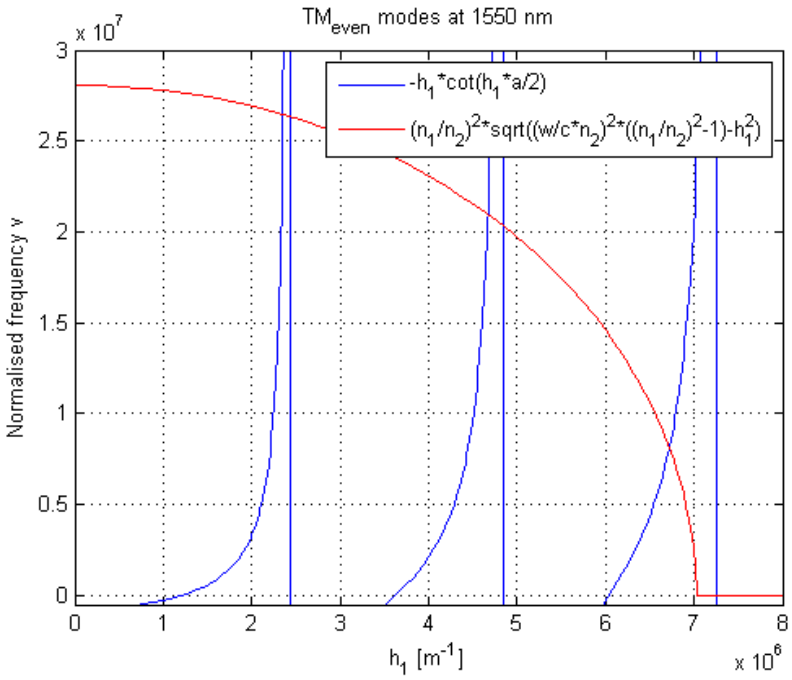

Figure 9 – TM_{even} found modes @ 1550 nm.

Let us assume a planar dielectric waveguide as the one on Figure 7 and the following data:

$n_1 = 2$

$n_2 = 1$

$\lambda = 1.55\ \mu m$

$a = 2.6\ \mu m$

The characteristic values are obtained by solving the characteristic equation:

$$4 * \sqrt{\left(\frac{2\pi}{1.55 * 10^{-6}}\right)^2 * 3 - h_1^2} = -h_1 \cot(1.3 * 10^{-6} h_1)$$

As can be observed on Figure 10, three modes have been found and to each of them corresponds a characteristic value h_1.

So, the characteristic values found mean that this dielectric structure supports three TM_{even} modes:

$$h_1(1) = 2.3483 \times 10^6$$
$$h_1(2) = 4.6648 \times 10^6$$
$$h_1(3) = 6.7193 \times 10^6$$

The exponential decay coefficients are, respectively:

$$v_1 = 6.6168 \times 10^6$$
$$v_2 = 5.2475 \times 10^6$$
$$v_3 = 2.0366 \times 10^6$$

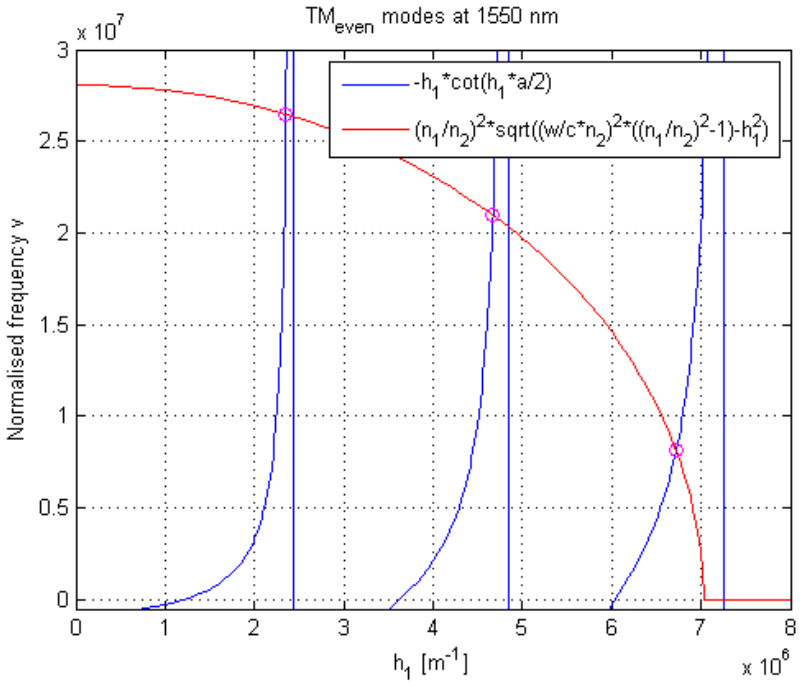

Figure 10 - Functions intersection gives characteristic value h_1.

Looking at these values it can be observed that as h_1 increases (wave's frequency increases), the exponential decay coefficient v decreases, meaning for higher modes there is less decay on surrounding media n_2 and, consequently, they are less confined to the core n_1. On Figure 11 the normalised profile $\frac{E_z^0}{\frac{a}{2}}$ for the three modes is plotted. With h_1 and v values, the EM fields' expressions for TM_{even} modes can now be written.

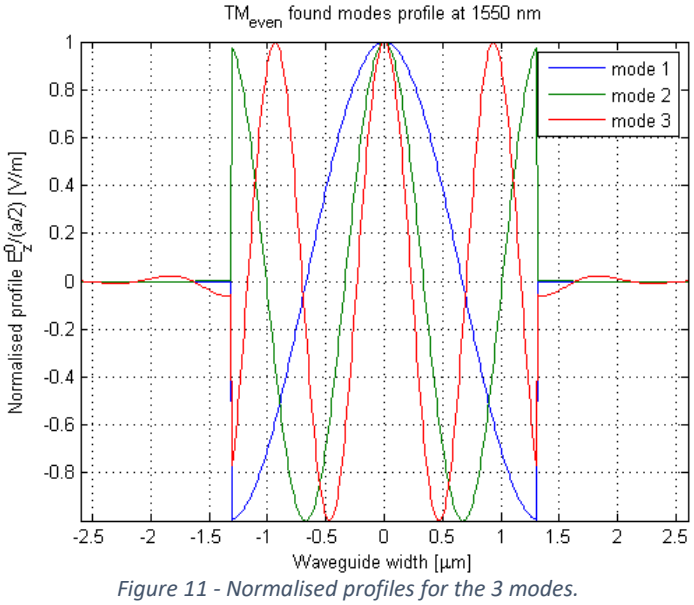

Figure 11 - Normalised profiles for the 3 modes.

One other analysis is required before continuing to next definition. This time and still with the same dielectric structure, the first mode's exponential decay coefficient will be under observation as the wave's frequency decreases or, which is the same, wavelength increases. Wavelengths 820 nm, 1300 nm and 1550 nm will be used and respective v parameters will be compared (see Appendix B for figures plot and calculations). Figure 12 shows the first modes for the previously mentioned wavelengths. Due to the high values involved and their proximity between each other, visual inspection does not reveal much. That is why a zoomed in portion of this figure is shown on Figure 13.

As can be observed, although slightly, there is a deviation towards the waveguide's edges as frequency decreases, meaning the wave's confinement in the core decreases as first mode's wavelength increases (or frequency decreases). The same decreasing behaviour is experienced by the exponential decay coefficient v, as data shows:

$$TM_{even}\ \ n=1: \quad \begin{vmatrix} v_{820\,nm} = 1.3056 \times 10^{7} \\ v_{1300\,nm} = 8.0318 \times 10^{6} \\ v_{1550\,nm} = 6.6168 \times 10^{6} \end{vmatrix}$$

As such, there will be a frequency for which $v = 0$ and the wave is not guided any more. This frequency is designated Cut off frequency.

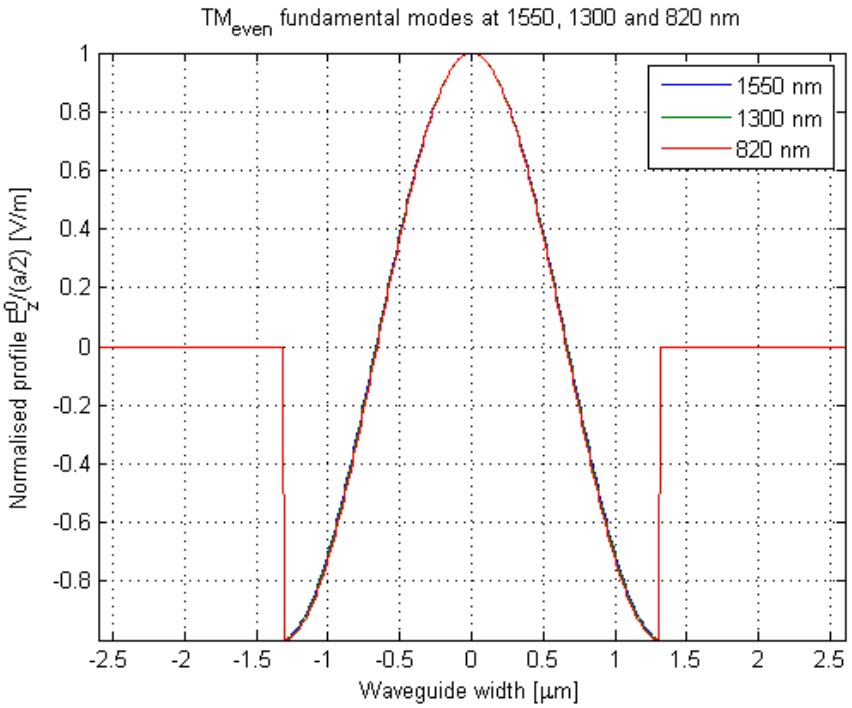

Figure 12 - Fundamental TM_{even} modes for different wavelengths.

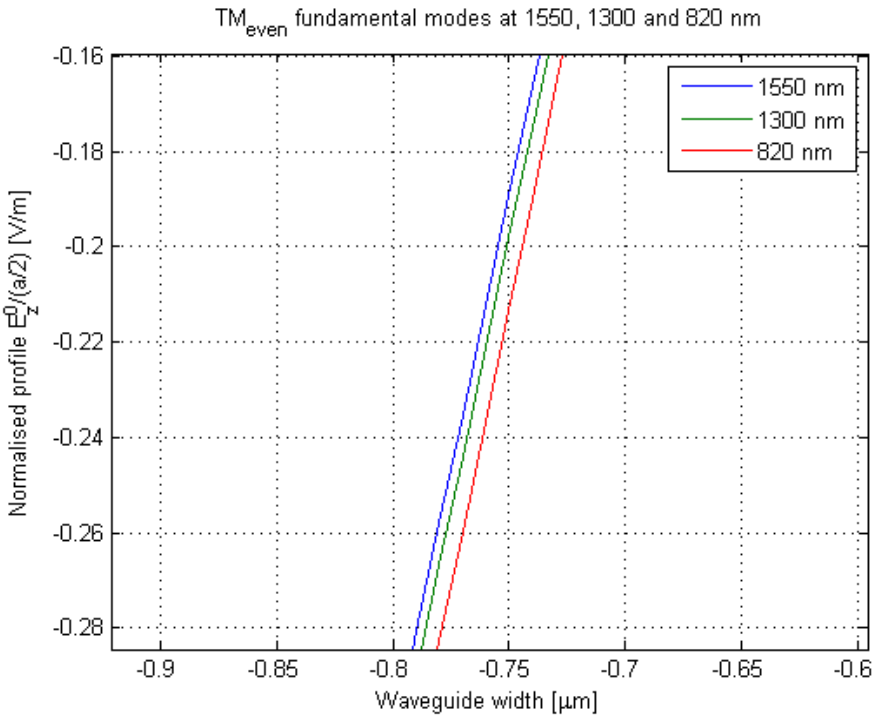

Figure 13 - Zoom in portion of Figure 10.

Cut off frequency, TM even modes

So, as stated before, the cut condition will be:

$$v = 0$$

Using both v and h_1 definitions and assuming the above stated, it can be written,

$$v^2 = -\gamma^2 - \left(\frac{\omega}{c} n_2\right)^2 = 0 \leftrightarrow \gamma^2 = -\left(\frac{\omega}{c} n_2\right)^2$$

$$h_1^2 = \gamma^2 + \left(\frac{\omega}{c} n_1\right)^2 = \left(\frac{\omega}{c}\right)^2 (n_1^2 - n_2^2) \leftrightarrow h_1 = \frac{\omega}{c}\sqrt{n_1^2 - n_2^2}$$

On the other hand, when $v = 0$,

$$v = -h_1 \left(\frac{n_2}{n_1}\right)^2 \cot\left(\frac{h_1 a}{2}\right) = 0 \rightarrow -\cot\left(\frac{h_1 a}{2}\right) = 0$$

which means,

$$\frac{h_1 a}{2} = \frac{\omega a}{2c}\sqrt{n_1^2 - n_2^2} = \left(n - \frac{1}{2}\right)\pi, \qquad n = 1,2,\ldots$$

Using $\omega = 2\pi f$, the cut off frequency expression for TM_{even} modes can be achieved,

$$f_c[TM_{even}] = \frac{\left(n - \frac{1}{2}\right)c}{a\sqrt{n_1^2 - n_2^2}}, \qquad n = 1,2,\ldots$$

One interesting and true fact stated by this equation is that the cut off frequency is inversely proportional to the width of the waveguide. Meaning, a narrower waveguide can support less propagating modes. Also, the lowest cut off frequency is achieved when $n = 1$,

$$f_c[TM_{even}] = \frac{c}{2a\sqrt{n_1^2 - n_2^2}}, \qquad n = 1$$

An easy way to find out how many modes a particular dielectric structure and wave frequency can support, is to use the cut off frequency equation with increasing n until the result equals or overcomes the working frequency. Again, considering the data of previous example,

TM_{even} $\quad n = 1 \quad f_c = 33.31\ THz$

$\quad n = 2 \quad f_c = 99.93\ THz$

$\quad n = 3 \quad f_c = 166\ THz$

$\quad n = 4 \quad f_c = 233\ THz$

and that,

$$\lambda = 1.55\ \mu m \rightarrow f = 193.55\ THz$$

It can confirm the number of TM_{even} modes this device supports (modes = 3).

TM odd modes

The procedure is the same utilized before on TM even modes. There is still $H_z^0 = 0$ and $E_z^0 = \Psi_{odd}$ and the other constituents are calculated using:

$$H_x^0 = \frac{j\omega\varepsilon}{h^2}\frac{\partial E_z^0}{\partial y}$$

$$E_y^0 = -\frac{\gamma}{h^2}\frac{\partial E_z^0}{\partial y}$$

$$E_x^0 = H_y^0 = 0$$

or, in the considered dielectric structure,

$$|y| \le \frac{a}{2} \qquad \left| \begin{aligned} E_z^0 &= A\sin(h_1 y) \\ H_x^0 &= \frac{j\omega\varepsilon_1}{h_1} A\cos(h_1 y) \\ E_y^0 &= -\frac{j\beta}{h_1} A\cos(h_1 y) \end{aligned} \right.$$

$$y > \frac{a}{2} \qquad \left| \begin{aligned} E_z^0 &= A\sin\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ H_x^0 &= \frac{j\omega\varepsilon_2}{v} A\sin\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ E_y^0 &= -\frac{j\beta}{v} A\sin\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \end{aligned} \right.$$

$$y < -\frac{a}{2} \qquad \left| \begin{aligned} E_z^0 &= -A\sin\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ H_x^0 &= \frac{j\omega\varepsilon_2}{v} A\sin\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ E_y^0 &= -\frac{j\beta}{v} A\sin\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \end{aligned} \right.$$

where $\gamma = j\beta$, $h_2^2 = -v^2$ and $\beta = \sqrt{\left(\frac{\omega}{c}n_1\right)^2 - h_1^2} = \sqrt{\left(\frac{\omega}{c}n_2\right)^2 + v^2}$.

Although this solution complies with the longitudinal component of the electrical field continuity, it has not been made sure that the other components comply with the other boundary conditions. Namely, H_x^0 that is a magnetic field component parallel to the interfaces and there is not any superficial current, continuity must be also assured. This means,

$$\frac{j\omega\varepsilon_1}{h_1} A\cos\left(\frac{h_1 a}{2}\right) = \frac{j\omega\varepsilon_2}{v} A\sin\left(\frac{h_1 a}{2}\right)$$

or, equivalently,

$$v = \frac{\varepsilon_2}{\varepsilon_1} h_1 \tan\left(\frac{h_1 a}{2}\right)$$

or, still,

$$\sqrt{\left(\frac{\omega}{c}\right)^2 (n_1^2 - n_2^2) - h_1^2} = \frac{\varepsilon_2}{\varepsilon_1} h_1 \tan\left(\frac{h_1 a}{2}\right)$$

which defines the characteristic equation for TM_{odd} modes, as happened before with TM_{even} modes. Again, numerical methods are required to solve this equation and it can be used to calculate the TM_{odd} modes cut off frequencies and the values h_1, h_2 and γ for a given operating frequency. Also here, there are finite modes and the characteristic parameters are frequency dependent.

In order to illustrate what has been said, let us go back to the previous example on Figure 7 and with the same data. The characteristic equation is, this time,

$$\sqrt{\left(\frac{2\pi}{1.55 * 10^{-6}}\right)^2 * 3 - h_1^2} = \frac{1}{4} h_1 \tan(1.3 * 10^{-6} h_1)$$

Again, with Matlab help, characteristic parameters h_1 and v have been calculated and, also, Figure 14 has been plotted. As can be seen, three modes have been found and the respective characteristic parameters are:

$$TM_{odd} \left| \begin{array}{ll} h_1(1) = 1.1749 \times 10^6 & v_1 = 6.9222 \times 10^6 \\ h_1(2) = 3.5143 \times 10^6 & v_2 = 6.0784 \times 10^6 \\ h_1(3) = 5.7746 \times 10^6 & v_3 = 3.9938 \times 10^6 \end{array} \right.$$

Once again, as before, the first mode's exponential decay coefficient will be under observation as the wave's frequency decreases or, which is the same, wavelength increases. Wavelengths 820 nm, 1300 nm and 1550 nm will be used again and respective v parameters will be compared. Figure 15 shows the first modes for mentioned wavelengths. Also this time, due to the high values involved and their proximity between each other, visual inspection does not reveal much. That is why a zoomed in portion of this figure is shown on Figure 16 (see Appendix B for Matlab script).

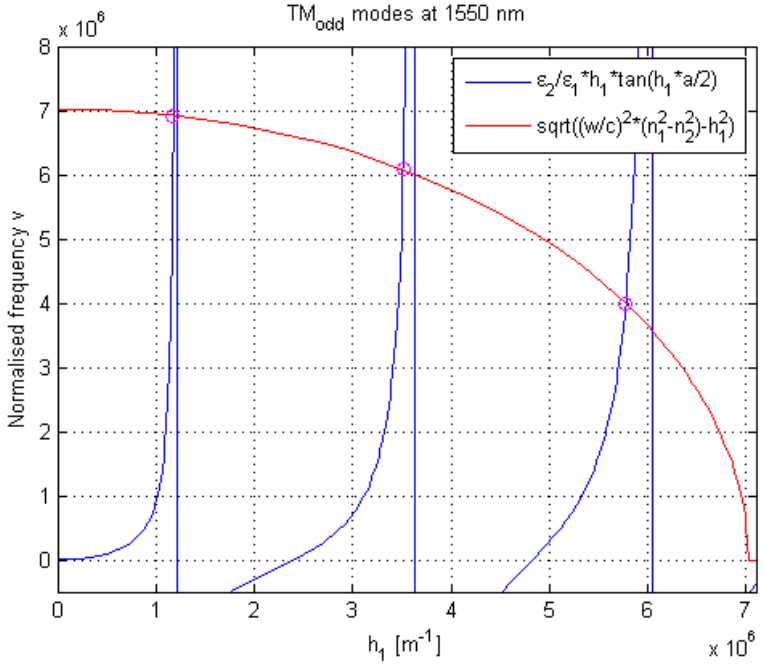

Figure 14 – TM_{odd} found modes @ 1550 nm.

As before, the exponential decay coefficient v has the same behaviour as frequency decreases, that is to say, there is less wave's confinement in the core, as shown by data:

TM_{odd} $n = 1$:

$$v_{820\,nm} = 1.3551 \times 10^7$$
$$v_{1300\,nm} = 8.2877 \times 10^6$$
$$v_{1550\,nm} = 6.9222 \times 10^6$$

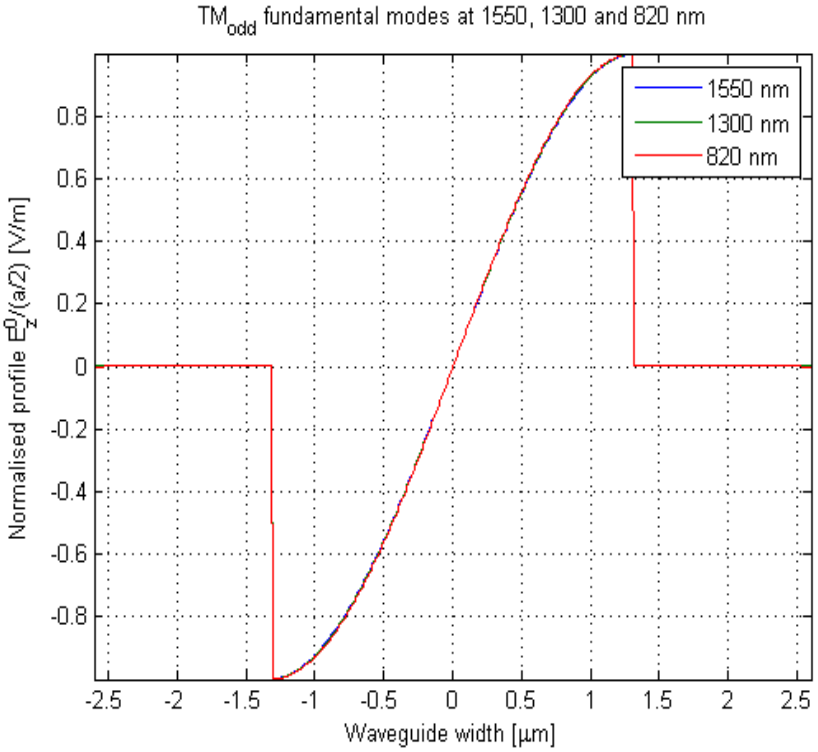

Figure 15 - Fundamental TM_{odd} modes for different wavelengths.

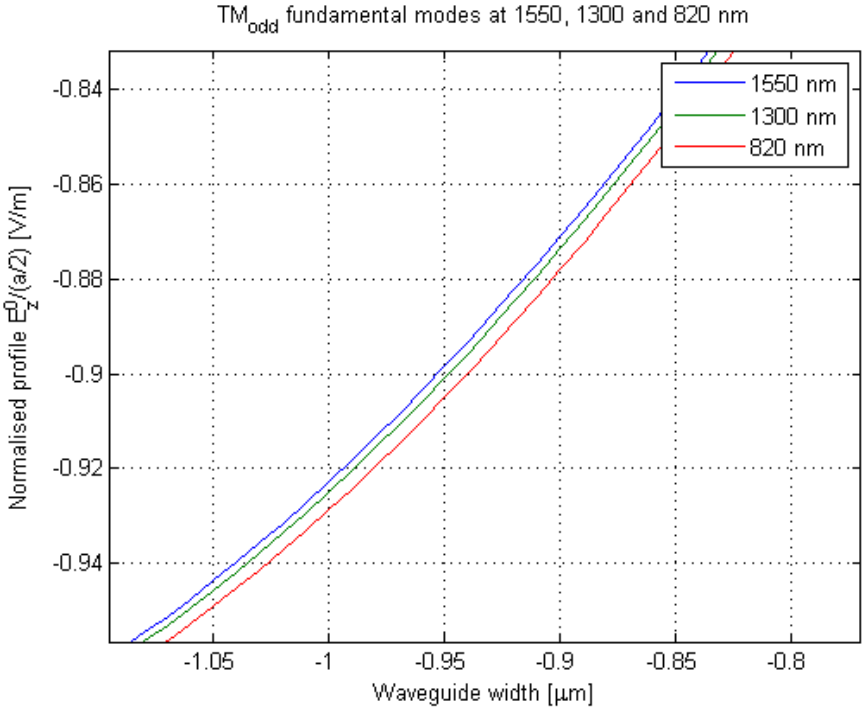

Figure 16 - Zoom in portion of Figure 13.

Cut off frequency, TM odd modes

For these modes, the cut off condition $v = 0$ leads to,

$$\frac{\varepsilon_2}{\varepsilon_1} h_1 \tan\left(\frac{h_1 a}{2}\right) = 0$$

or,

$$\frac{h_1 a}{2} = (n-1)\pi \leftrightarrow h_1 = \frac{(n-1)2\pi}{a}, \qquad n = 1,2,\dots$$

Since, $h_1 = \frac{\omega}{c}\sqrt{n_1^2 - n_2^2}$, the cut off frequency expression for TM_{odd} modes will be:

$$\boxed{f_c[TM_{odd}] = \frac{(n-1)c}{a\sqrt{n_1^2 - n_2^2}}, \qquad n = 1,2,\dots}$$

The lowest cut off frequency is obtained when $n = 1$, which means $f_c[TM_{odd}] = 0$. This means that, independently of frequency or waveguide's width, the 1st TM_{odd} mode is always present.

TE even modes

In this case, $E_z^0 = 0$ and $H_z^0 = \Psi_{even}$, being the not null constituents of EM fields given by:

$$|y| \leq \frac{a}{2} \quad \left| \begin{aligned} H_z^0 &= B\cos(h_1 y) \\ H_y^0 &= \frac{j\beta}{h_1} B\sin(h_1 y) \\ E_x^0 &= \frac{j\omega\mu_0}{h_1} B\sin(h_1 y) \end{aligned} \right.$$

$$y > \frac{a}{2} \quad \left| \begin{aligned} H_z^0 &= B\cos\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ H_y^0 &= -\frac{j\beta}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ E_x^0 &= \frac{j\omega\mu_0}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \end{aligned} \right.$$

$$y < -\frac{a}{2} \quad \left| \begin{aligned} H_z^0 &= B\cos\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ H_y^0 &= \frac{j\beta}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ E_x^0 &= \frac{j\omega\mu_0}{v} B\cos\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \end{aligned} \right.$$

where $\gamma = j\beta$, $h_2^2 = -v^2$ and $\beta = \sqrt{\left(\frac{\omega}{c}n_1\right)^2 - h_1^2} = \sqrt{\left(\frac{\omega}{c}n_2\right)^2 + v^2}$

These expressions fulfil the need for continuity of the longitudinal component of the magnetic field. Nevertheless, it is also essential that the tangential component of the electric field (E_x^0) be also continuous at $y = \pm {}^{a}/_{2}$. This means,

$$\frac{j\omega\mu_0}{h_1} B \sin\left(\frac{h_1 a}{2}\right) = \frac{j\omega\mu_0}{v} B \cos\left(\frac{h_1 a}{2}\right)$$

or still,

$$v = -h_1 \cot\left(\frac{h_1 a}{2}\right)$$

Using v definition, the characteristic equation for TE_{even} modes can be written,

$$\sqrt{\left(\frac{\omega}{c}\right)^2 (n_1^2 - n_2^2) - h_1^2} = -h_1 \cot\left(\frac{h_1 a}{2}\right) \tag{11}$$

Once more, Matlab (see Appendix B) was used to plot Figure 17 and to calculate characteristic parameters h_1 and v:

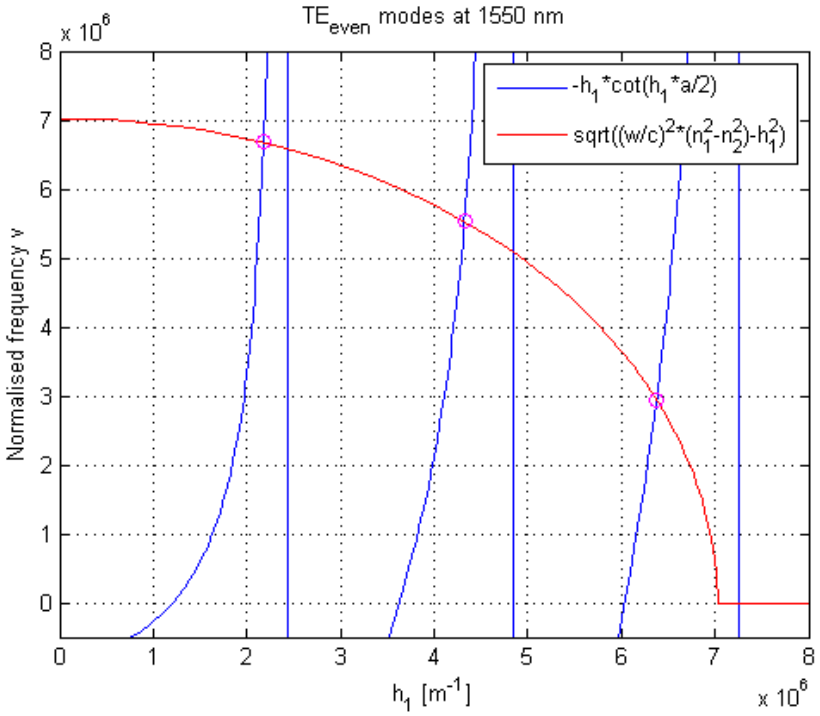

Figure 17 – TE_{even} found modes @ 1550 nm.

$$TE_{even}\left|\begin{array}{ll} h_1(1) = 2.1743 \times 10^6 & v_1 = 6.6760 \times 10^6 \\ h_1(2) = 4.3230 \times 10^6 & v_2 = 5.5325 \times 10^6 \\ h_1(3) = 6.3743 \times 10^6 & v_3 = 2.9437 \times 10^6 \end{array}\right.$$

Cut off frequency, TE even modes

From characteristic equation 11 and cut off condition $v = 0$, it is possible to write,

$$v = -h_1 \cot\left(\frac{h_1 a}{2}\right) = 0 \rightarrow \cot\left(\frac{h_1 a}{2}\right) = 0 \leftrightarrow \frac{h_1 a}{2} = \left(n - \frac{1}{2}\right)\pi, \qquad n = 1,2,\ldots$$

or still,

$$\boxed{f_c[TE_{even}] = \frac{\left(n - \frac{1}{2}\right)c}{a\sqrt{n_1^2 - n_2^2}}, \qquad n = 1,2,\ldots}$$

This equation is identical to the one obtained for TM_{even} modes. So, whatever has been said about TM_{even} modes cut off frequency can also be applied on TE_{even} modes, namely the lowest cut off frequency.

TE odd modes

This time is $E_z^0 = 0$ and $H_z^0 = \Psi_{odd}$, being the not null constituents of EM fields given by:

$$|y| \leq \frac{a}{2} \qquad \left|\begin{array}{l} E_z^0 = A\sin(h_1 y) \\ H_x^0 = -\frac{j\beta}{h_1} A\cos(h_1 y) \\ E_y^0 = -\frac{j\omega\mu_0}{h_1} A\cos(h_1 y) \end{array}\right.$$

$$y > \frac{a}{2} \qquad \left|\begin{array}{l} E_z^0 = A\sin\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ H_x^0 = -\frac{j\beta}{v} A\sin\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \\ E_y^0 = -\frac{j\omega\mu_0}{v} A\sin\left(\frac{h_1 a}{2}\right) e^{-v\left(y-\frac{a}{2}\right)} \end{array}\right.$$

$$y < -\frac{a}{2} \qquad \left| \begin{array}{l} E_z^0 = -A \sin\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ H_x^0 = -\frac{j\beta}{v} A \sin\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \\ E_y^0 = -\frac{j\omega\mu_0}{v} A \sin\left(\frac{h_1 a}{2}\right) e^{v\left(y+\frac{a}{2}\right)} \end{array} \right.$$

where $\gamma = j\beta$, $h_2^2 = -v^2$ and $\beta = \sqrt{\left(\frac{\omega}{c} n_1\right)^2 - h_1^2} = \sqrt{\left(\frac{\omega}{c} n_2\right)^2 + v^2}$

As before, with TE_{even} modes, to fulfil all boundary conditions is necessary to assure continuity at $y = \pm {}^{a}/_{2}$ of the parallel component of the electric field (E_x^0):

$$-\frac{j\omega\mu_0}{h_1} A \cos\left(\frac{h_1 a}{2}\right) = -\frac{j\omega\mu_0}{v} A \sin\left(\frac{h_1 a}{2}\right)$$

meaning,

$$v = h_1 \tan\left(\frac{h_1 a}{2}\right)$$

The characteristic equation for TE_{odd} modes can now be written:

$$\sqrt{\left(\frac{\omega}{c}\right)^2 (n_1^2 - n_2^2) - h_1^2} = h_1 \tan\left(\frac{h_1 a}{2}\right) \qquad (12)$$

Once again, Matlab (see Appendix B) was used to plot Figure 18 for it eases interpretation through visual confirmation.

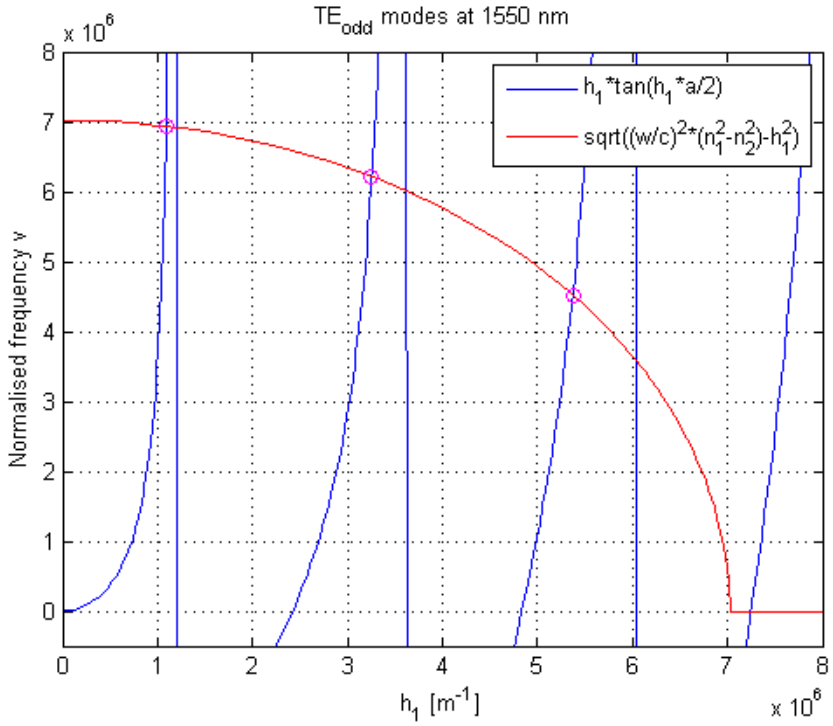

Figure 18 – TE_{odd} found modes @ 1550 nm.

Next, the characteristic parameters h_1 and v are shown for TE_{odd} modes, considering the same practical example that has been used all the way through this chapter ($a = 2.6\ \mu m\ and\ \lambda = 1.55\ \mu m$).

$$TE_{odd} \begin{cases} h_1(1) = 1.0885e^6 & v_1 = 6.9363e^6 \\ h_1(2) = 3.2541e^6 & v_2 = 6.2215e^6 \\ h_1(3) = 5.3714e^6 & v_3 = 4.5215e^6 \end{cases}$$

Cut off frequency, TE odd modes

Using equation 12 and cut off condition $v = 0$,

$$v = h_1 \tan\left(\frac{h_1 a}{2}\right) = 0 \rightarrow \tan\left(\frac{h_1 a}{2}\right) = 0 \rightarrow \frac{h_1 a}{2} = (n-1)\pi, \qquad n = 1,2,\dots$$

or,

$$f_c[TE_{odd}] = \frac{(n-1)c}{a\sqrt{n_1^2 - n_2^2}}, \qquad n = 1,2,\dots$$

This definition is identical to the one found for $f_c[TM_{odd}]$, which means that the lowest cut off frequency for TE_{odd} modes is also zero ($n = 1$) and, for this reason, the first mode is always present, independently of waveguide width or operating frequency.

Table 1 shows the relevant expressions to calculate cut off frequency and exponential decay component for all the modes mentioned so far when on planar dielectric waveguides.

Table 1 - Characteristic parameters for all modes [4].

MODES		EXPONENTIAL DECAY	CUT OFF FREQUENCY
EVEN	TM	$v = -h_1 \left(\frac{n_2}{n_1}\right)^2 \cot\left(\frac{h_1 a}{2}\right)$	$f_c = \frac{\left(n - \frac{1}{2}\right)c}{a\sqrt{n_1^2 - n_2^2}}, \qquad n = 1,2,\dots$
	TE	$v = -h_1 \cot\left(\frac{h_1 a}{2}\right)$	
ODD	TM	$v = \frac{\varepsilon_2}{\varepsilon_1} h_1 \tan\left(\frac{h_1 a}{2}\right)$	$f_c = \frac{(n-1)c}{a\sqrt{n_1^2 - n_2^2}}, \qquad n = 1,2,\dots$
	TE	$v = h_1 \tan\left(\frac{h_1 a}{2}\right)$	

Throughout this last sub-chapter (Propagating modes), in the given example where $\lambda = 1.55\ \mu m$, either on TE or TM modes, the number of found characteristic parameters was three for each polarisation (even or odd). This means that, in this particular case, the dielectric structure was able to support three even and three odd modes, which in total makes six (6) found modes. Now, it is time to confirm this in same working conditions with the Optiwave simulator (OptiFDTD).

2.2.3. Optiwave® simulator - OptiFDTD

Mode Solver

As before, the example used throughout this chapter has been utilized again. Like so, the results obtained in both methodologies can be compared. The device simulated is thus a $2.6\ \mu m$ wide and $50\ \mu m$ long linear waveguide, embedded on an $8\ \mu m$ wide air wafer (surrounding media). The refraction indexes chosen for both waveguide and wafer were, respectively, 2 and 1. A vertical input plane of $1\ V/m$ amplitude was inserted at $1\ \mu m$ position, configured as GMCW (Gaussian Modulated Continuous Wave) and $\lambda = 1.55\ \mu m$, as shown on Figure 19, input field transverse modal and Initial Phase = 0°. The supported modes were, as expected, the ones showed on Figure 20, which completely agrees with what has been determined before by theory. The TE mode solver was able to find six modes. It should not be overlooked that the number of TM found modes would be also six (again, three even and three odd modes).

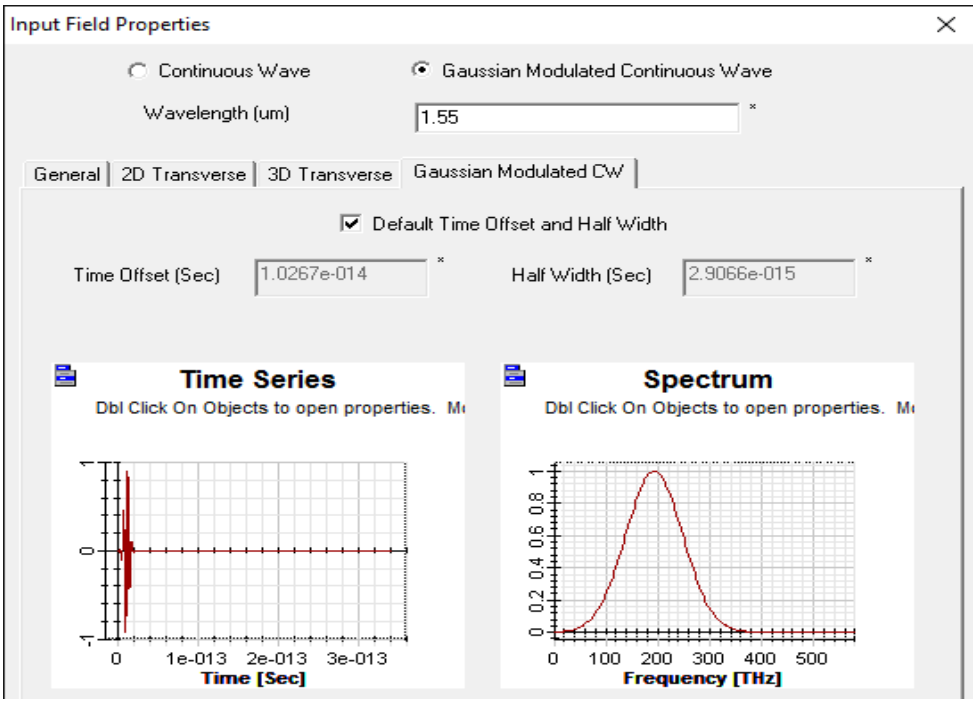

Figure 19 - 1.55 μm wavelength Gaussian Modulated Continuous Wave.

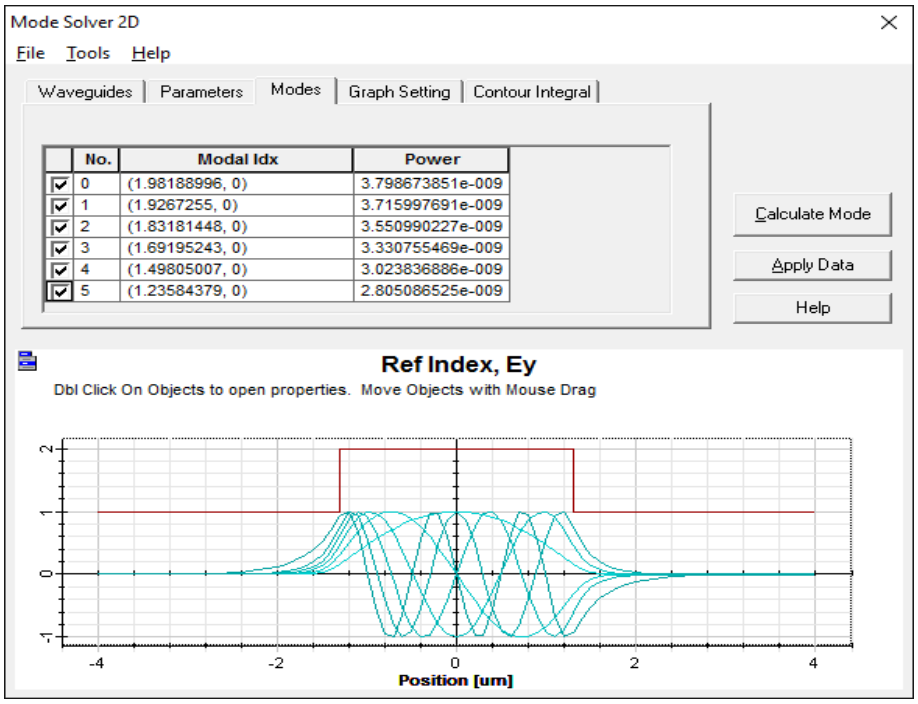

Figure 20 - Mode Solver 2D, TE found modes.

Nevertheless, some other simulations took place in order to confirm whether or not OptiFDTD's results were according to previous calculations:

- Altering only the wavelength to $10\,\mu m$ and keeping unchanged the other data, the number of modes obtained by 2D TE Mode Solver can be observed on Figure 21.

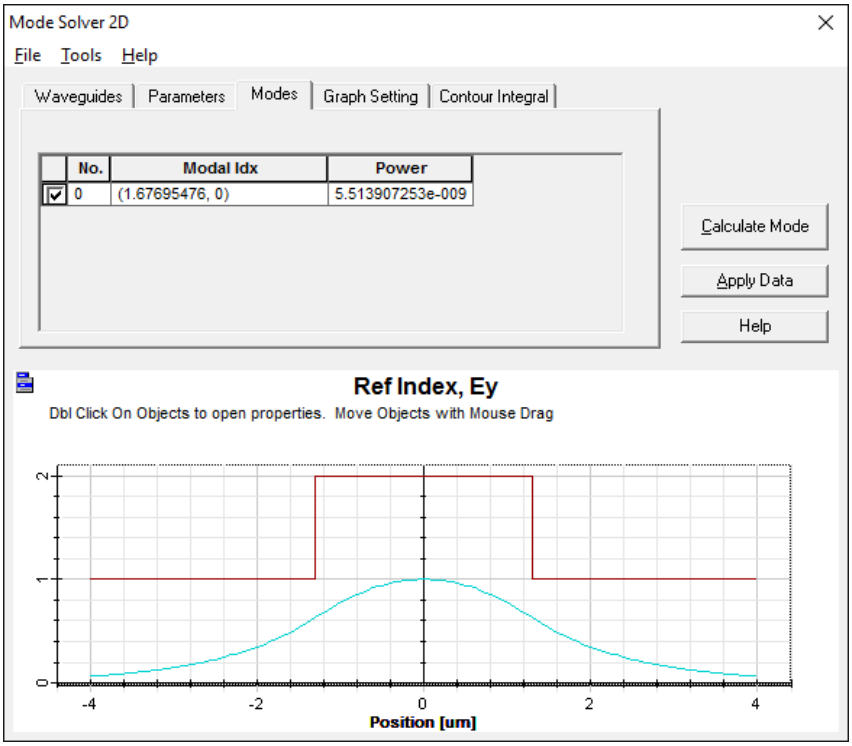

Figure 21 - Mode Solver 2D, $\lambda = 10\ \mu m$ TE found modes.

Theory confirmation:

$$\lambda = 10\ \mu m \leftrightarrow f = 30\ THz$$

$$f_c[TE_{even}] = \frac{c}{2a\sqrt{n_1^2 - n_2^2}} = 33.309\ THz$$

Thus, in the above situation, there can only be one propagating mode, the fundamental TE_{odd} mode for it is always present (as previously mentioned), because $f < f_c$ and, consequently, there are no even propagating modes that can be supported by this waveguide at this wavelength.

2.3. Non-planar dielectric waveguides

It is now time to mention some more realistic dielectric devices, although still ideal. The non-planar dielectric waveguide is characterized by the index profile $n(x, y)$ being a function of both transverse coordinates, x and y, instead of only one of these as were the case of planar dielectric waveguides. There are mainly two groups of non-planar dielectric waveguides:

1. One that is formed by a single particular geometry: - cylindrical, which are designated *optical fibres*.
2. One known as *channel waveguides* and is formed by variations of the general structure, which is usually a square/rectangular dielectric waveguide either embedded or deposited on top of or in between substrate and cladding. There are several combinations that can be achieved with a waveguide, substrate and cladding and Figure 22 shows some of them and also the previously mentioned optical fibre:
    a. Buried channel waveguide.
    b. Strip-loaded waveguide.
    c. Ridge waveguide.
    d. Rib waveguide.
    e. Diffused waveguide.
    f. Optical fibre waveguide.

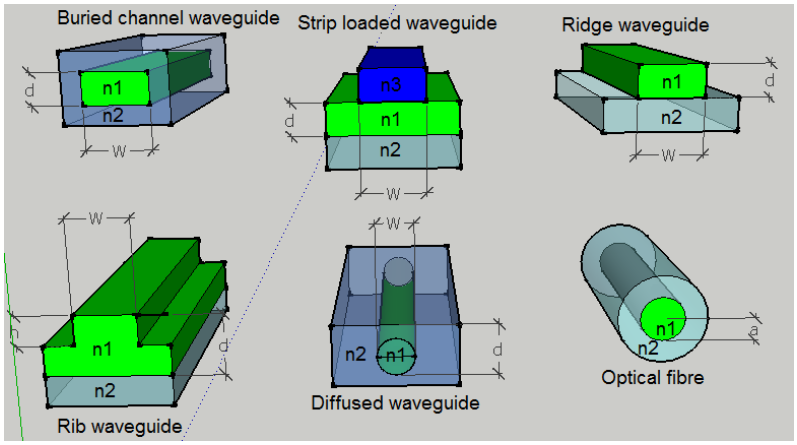

Figure 22 - Non-planar dielectric waveguides [5].

Buried channel waveguide is formed with a high index of refraction core buried in a low index of refraction surrounding media. The cross-section geometry of the core can be any but it is often a rectangular shape.

Strip-loaded waveguide is formed by loading a planar waveguide, which already provides optical confinement in the x direction, with a dielectric strip of index $n_3 < n_1$, or a metal strip, to strengthen optical confinement in the y direction. The guiding core of a strip waveguide is the n_1 region under the loading strip, with its thickness d determined by n_1 layer thickness and its width w defined by the width of the loading strip.

Ridge waveguide has a structure that looks like a strip waveguide but is the strip, or the ridge, on top of a planar dielectric structure that has a high index of refraction and that is, actually, the guiding core. The ridge waveguide has strong optical confinement because it is surrounded over three of its sides by low index of refraction (air or cladding) material.

Rib waveguide has a structure similar to that of a strip or ridge waveguide, but the strip and the planar layer beneath have the same index of refraction and, together, form the guiding core.

These four types of waveguides are usually designated *rectangular waveguides* with a thickness d in the x direction and width w in the y direction, even though their shapes are normally not exactly rectangular.

Diffused waveguide is formed with a high index of refraction region in a substrate through diffusion of dopants, such as $LiNbO_3$ waveguide with a core formed by Ti diffusion. Because of the diffusion process, the core boundaries in the substrate are not sharply defined. A diffused waveguide also has a thickness d defined by the diffusion depth of the dopant in the x direction and a width w defined by the distribution of the dopant in the y direction.

2.3.1. Circular dielectric waveguides - Optical fibre

An optical fibre is formed by a core with a higher refractive index than the surrounding medium, the cladding. Figure 23 is a diagram of such a device where n_1 and n_2 represent core and cladding indexes of refraction, respectively, and a the core's radius. In order to simplify analysis, it will be considered that both core and cladding are of infinite length and that the dielectric materials are ideal (that is to say, there are no losses in these materials). Because this waveguide's geometry is cylindrical, the best suited coordinates system to develop the analysis is the cylindrical one (r, ϕ, z). Thus, the equations are as follows below:

$$r \leq a: \left| \begin{array}{l} \nabla^2_{r\phi} E_z^0 + h_1^2 E_z^0 = 0 \\ \nabla^2_{r\phi} H_z^0 + h_1^2 H_z^0 = 0 \end{array} \right. \quad , \; where \; h_1^2 = \gamma^2 + \left(\frac{\omega}{c} n_1\right)^2$$

$$r > a: \left| \begin{array}{l} \nabla^2_{r\phi} E_z^0 + h_2^2 E_z^0 = 0 \\ \nabla^2_{r\phi} H_z^0 + h_2^2 H_z^0 = 0 \end{array} \right. \quad , \; where \; h_2^2 = \gamma^2 + \left(\frac{\omega}{c} n_2\right)^2$$

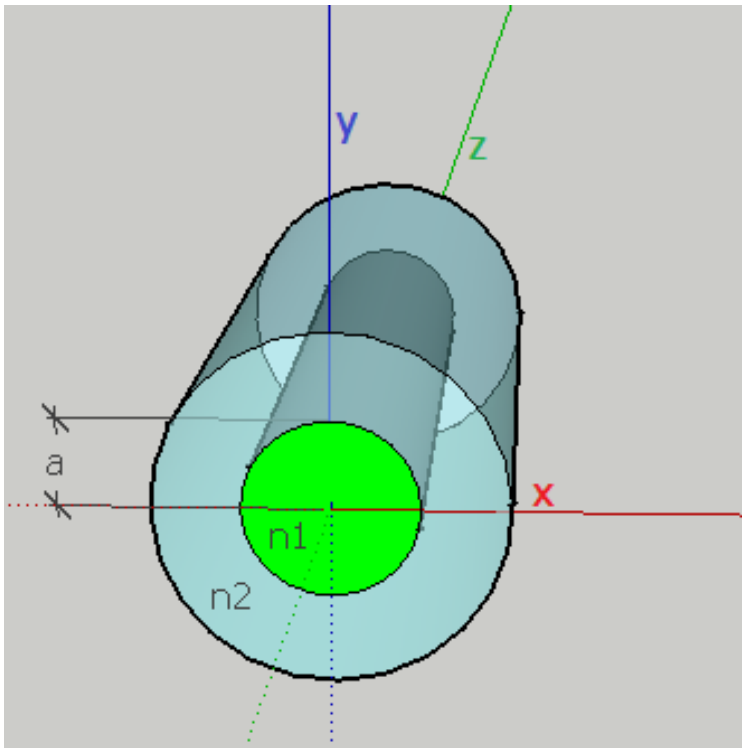

Figure 23 - Optical fibre diagram.

Wave equations for circular dielectric waveguides

Also in this case, one has to analyse the wave equation general solution that explains the longitudinal components behaviour. The fields,

$$\vec{E} = E_r\vec{r} + E_\phi\vec{\phi} + E_z\vec{z} \text{ and } \vec{H} = H_r\vec{r} + H_\phi\vec{\phi} + H_z\vec{z}$$

are obtained from exponential decay coefficient equation 4 and wave vector equations 6:

$$\nabla^2\vec{E} + \omega^2\mu\epsilon\vec{E} = 0$$

$$\nabla^2\vec{H} + \omega^2\mu\epsilon\vec{H} = 0$$

In this coordinates system, the vector field Laplacian (∇^2) shown on previous equations is a rather complex expression for:

$$\nabla^2\vec{V} = \nabla(\nabla\cdot\vec{V}) - \nabla\times(\nabla\times\vec{V})$$

Nevertheless, according to [4] and [6], the procedure for the analysis is similar to the one used before on the planar dielectric waveguides. So, in order to obtain the fields' longitudinal components, scalar wave equations are utilized and the other constituents are encountered

through Maxwell's equations. This really simplifies calculations for vector and scalar fields' Laplacians, as long as the wanted longitudinal components are concerned, use the same below expression:

$$\nabla^2 = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \phi^2} + \frac{\partial^2}{\partial z^2} \tag{13}$$

Moreover, assuming that waves inside the waveguide have a propagation constant γ, meaning:

$$\vec{E} = \overrightarrow{E^0}(r,\phi)e^{-\gamma z}$$

$$\vec{H} = \overrightarrow{H^0}(r,\phi)e^{-\gamma z}$$

Thus, it can be written,

$$\nabla^2_{r\phi} E_z^0 + h^2 E_z^0 = 0$$

$$\nabla^2_{r\phi} H_z^0 + h^2 H_z^0 = 0$$

where,

$$h^2 = \gamma^2 + \omega^2 \mu\epsilon$$

Using equation 13 (Laplacian) on the EM fields, transverse components for this coordinates system can be obtained through $\nabla \times \vec{E} = -j\omega\mu\vec{H}$ and $\nabla \times \vec{H} = j\omega\varepsilon\vec{E}$:

$$\frac{1}{r}\frac{\partial E_z^0}{\partial \phi} + \gamma E_\phi^0 = -j\omega\mu H_r^0 \qquad \frac{1}{r}\frac{\partial H_z^0}{\partial \phi} + \gamma H_\phi^0 = j\omega\varepsilon E_r^0$$

$$-\frac{\partial E_z^0}{\partial r} - \gamma E_r^0 = -j\omega\mu H_\phi^0 \qquad -\frac{\partial H_z^0}{\partial r} - \gamma H_r^0 = j\omega\varepsilon E_z^0$$

$$\frac{E_\phi^0}{r} + \frac{\partial E_\phi^0}{\partial r} - \frac{1}{r}\frac{\partial E_r^0}{\partial \phi} = -j\omega\mu H_z^0 \qquad \frac{H_\phi^0}{r} + \frac{\partial H_\phi^0}{\partial r} - \frac{1}{r}\frac{\partial H_r^0}{\partial \phi} = j\omega\varepsilon H_z^0$$

Combining these equations in pairs, the other constituents can be obtained:

$$\begin{aligned}
H_r^0 &= -\frac{1}{h^2}\left(\gamma\frac{\partial H_z^0}{\partial r} - \frac{j\omega\varepsilon}{r}\frac{\partial E_z^0}{\partial \phi}\right) \\
H_\phi^0 &= -\frac{1}{h^2}\left(\frac{\gamma}{r}\frac{\partial H_z^0}{\partial \phi} + j\omega\varepsilon\frac{\partial E_z^0}{\partial r}\right) \\
E_r^0 &= -\frac{1}{h^2}\left(\gamma\frac{\partial E_z^0}{\partial r} + \frac{j\omega\mu}{r}\frac{\partial H_z^0}{\partial \phi}\right) \\
E_\phi^0 &= -\frac{1}{h^2}\left(\frac{\gamma}{r}\frac{\partial E_z^0}{\partial \phi} + j\omega\mu\frac{\partial H_z^0}{\partial r}\right)
\end{aligned} \tag{14}$$

Cylindrical coordinates wave equation

Before going into the modes analysis in circular dielectric waveguides, one should first look into the general solutions of the wave's equation that explains longitudinal components behaviour.

Considering $\psi(r,\phi)$ a function that verifies the differential equation $\nabla_{r\phi}^2\psi + h^2\psi = 0$, that is:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\psi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\psi}{\partial\phi^2} + h^2\psi = 0 \tag{15}$$

This is a 2nd order partial differential equation and its solution can be obtained by the separation of variables method. That is to say, finding $\psi(r,\phi)$ such as $\psi(r,\phi) = R(r)\Phi(\phi)$. Replacing the obtained expression for $\psi(r,\phi)$ on equation 15 and dividing everything by $\psi(r,\phi)/r^2$, it is obtained after some mathematical manipulation:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\psi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\psi}{\partial\phi^2} + h^2\psi = 0 \qquad , \psi(r,\phi) = R(r)\Phi(\phi)$$

$$\frac{r^2}{R(r)}\frac{\partial^2[R(r)]}{\partial r^2} + \frac{r}{R(r)}\frac{\partial[R(r)]}{\partial r} + h^2r^2 = -\frac{1}{\Phi(\phi)}\frac{\partial^2[\Phi(\phi)]}{\partial\phi^2} \tag{16}$$

As shown on equation 16, equation 15 turned into an equation where each member depends only on one variable, either r or ϕ. The separation of variables method also states that, for a non-trivial solution and because variables r and ϕ are independent from one another, the expressions on both sides of the equation must yield the same constant. Considering this constant k_ϕ^2, for the right hand side of equation 16, one can write the equivalent equation:

$$\frac{\partial^2[\Phi(\phi)]}{\partial\phi^2} + k_\phi^2\Phi(\phi) = 0$$

The general solution for this equation is, as demonstrated before in 2.2.2.:

$$\Phi(\phi) = A\sin(k_\phi\phi) + B\cos(k_\phi\phi)$$

Moreover, because of waveguide's cylindrical symmetry, $\Phi(\phi)$ is periodic with a 2π period. This means that $\Phi(\phi) = \Phi(\phi + 2\pi)$ and, consequently,

$$\sin(k_\phi\phi) = \sin(k_\phi\phi + k_\phi 2\pi)$$

$$\cos(k_\phi\phi) = \cos(k_\phi\phi + k_\phi 2\pi)$$

which leads to conclude that k_ϕ must be an integer. Assuming $k_\phi = n, n \in \mathbb{N}$, then:

$$\Phi(\phi) = A\sin(n\phi) + B\cos(n\phi)$$

Also, by carefully choosing ϕ's origin, according to [6] and [4], $\Phi(\phi)$ simplifies to:

$$\Phi(\phi) = B\cos(n\phi)$$

where B is a constant. For the following analysis, it will be used the equivalent expression for $\Phi(\phi)$:

$$\Phi(\phi) = Be^{jn\phi}$$

for it simplifies the whole procedure.

On the other hand, $R(r)$ function can be obtained from the differential equation:

$$\frac{r^2}{R(r)}\frac{\partial^2[R(r)]}{\partial r^2} + \frac{r}{R(r)}\frac{\partial[R(r)]}{\partial r} + h^2r^2 = n^2 \leftrightarrow$$

$$\leftrightarrow r^2\frac{\partial^2[R(r)]}{\partial r^2} + r\frac{\partial[R(r)]}{\partial r} + (h^2r^2 - n^2)R(r) = 0 \qquad (17)$$

Equation 17 is known as Bessel's differential equation and its general solution is:

$$R(r) = CJ_n(hr) + DN_n(hr)$$

where J_n and N_n are Bessel's functions for first and second kind, respectively. Being $N_n(hr)$ a Bessel function of the 2nd kind (see Appendix C), it tends to infinity as the argument approaches zero. Thus, it cannot be considered as a solution of equation 17, resulting in:

$$R(r) = CJ_n(hr)$$

where C is a constant and J_n is a Bessel function of the 1st kind and order n.

On the other hand, if $h^2 < 0 \rightarrow h = jv$ and equation 17 (Bessel's differential equation) is:

$$r^2\frac{\partial^2[R(r)]}{\partial r^2} + r\frac{\partial[R(r)]}{\partial r} - (v^2r^2 + n^2)R(r) = 0$$

This equation is known as the modified Bessel's differential equation and its general solution involves the modified Bessel function of first and second kind, I_n and K_n, respectively. Thus, $R(r)$ becomes:

$$R(r) = EI_n(vr) + FK_n(vr)$$

where E and F are constants. Again, according to Appendix C, I_n must not be used when the region of interest includes infinity and K_n must not be considered when the region of interest includes the origin, resulting in:

$$R(r) = FK_n(vr)$$

As happened before with planar dielectric waveguides, it will be considered a guided EM wave when the EM fields in the core are related to sine variation and decreasing exponentially in the cladding. This implies that, for these circular waveguides, h_1 is real and $h_2 = jv$. Thus,

$$v = \sqrt{\left(\frac{\omega}{c}\right)^2(n_1^2 - n_2^2) - h_1^2}$$

and because on propagating waves $\gamma = j\beta$,

$$\beta = \sqrt{\left(\frac{\omega}{c}n_1\right)^2 - h_1^2} = \sqrt{\left(\frac{\omega}{c}n_2\right)^2 + v^2}$$

that is,

$$\frac{\omega}{c}n_2 < \beta < \frac{\omega}{c}n_1$$

As before, on planar dielectric waveguides, the expressions $\frac{\omega}{c}n_1$ and $\frac{\omega}{c}n_2$ are the phase constants of a propagating plane wave in limitless media with indexes of refraction n_1 and n_2, respectively. Moreover, using previously obtained results and taking into consideration that $\psi(r,\phi)$ cannot be infinity within the regions of interest, $r \leq a$ in the core and $r > a$ for cladding, it can be written:

$$\psi(r,\phi) \quad \begin{cases} AJ_n(h_1 r)e^{jn\phi} & ,r \leq a \\ BK_n(vr)e^{jn\phi} & ,r > a \end{cases}$$

Once more, $\psi(r,\phi)$ will represent the longitudinal component E_z^0 for TM modes and H_z^0 for TE modes. However, unlike previously mentioned waveguides, other modes must be considered in this case. They are designated hybrid modes for both their longitudinal components (E_z^0 and H_z^0) are not zero. These modes are due to the geometry of circular waveguides and their associated boundary conditions and are classified either as HE (Magnetic/Electric) or EH (Electric/Magnetic), depending on which component, either magnetic or electric, has the strongest contribution for the transverse fields. For this reason, the following analysis will consider the more general approach which is when both longitudinal components are present on the fields:

$$core: \quad \begin{aligned} E_z^0 &= AJ_n(h_1 r)e^{jn\phi} \\ H_z^0 &= BJ_n(h_1 r)e^{jn\phi} \end{aligned}$$

$$cladding: \quad \begin{aligned} E_z^0 &= CK_n(vr)e^{jn\phi} \\ H_z^0 &= DK_n(vr)e^{jn\phi} \end{aligned}$$

where A, B, C and D are constants to be determined. As a side note, if TM modes were to be considered ($H_z^0 = 0$), they could be obtained through previous equations with B=D=0 and if TE modes, then it should be A=C=0.

The transverse components are obtained directly from Maxwell's equations:

$$core: \quad \begin{aligned} H_r^0 &= -\frac{1}{h^2}\left[j\beta h_1 BJ_n'(h_1 r) + \frac{\omega\varepsilon_1 n}{r}AJ_n(h_1 r)\right]e^{jn\phi} \\ H_\phi^0 &= -\frac{1}{h^2}\left[-\frac{\beta n}{r}BJ_n(h_1 r) + j\omega\varepsilon_1 h_1 AJ_n'(h_1 r)\right]e^{jn\phi} \\ E_r^0 &= -\frac{1}{h^2}\left[j\beta h_1 AJ_n'(h_1 r) + \frac{\omega\mu_0 n}{r}BJ_n(h_1 r)\right]e^{jn\phi} \\ E_\phi^0 &= -\frac{1}{h^2}\left[-\frac{\beta n}{r}AJ_n(h_1 r) + j\omega\mu_0 h_1 BJ_n'(h_1 r)\right]e^{jn\phi} \end{aligned}$$

$$cladding: \quad H_r^0 = \frac{1}{v^2}\left[j\beta v D K_n'(vr) + \frac{\omega\varepsilon_2 n}{r} C K_n(vr)\right] e^{jn\phi}$$
$$H_\phi^0 = \frac{1}{v^2}\left[-\frac{\beta n}{r} D K_n(vr) + j\omega\varepsilon_2 v C K_n'(vr)\right] e^{jn\phi}$$
$$E_r^0 = \frac{1}{v^2}\left[j\beta v C K_n'(vr) - \frac{\omega\mu_0 n}{r} D K_n(vr)\right] e^{jn\phi}$$
$$E_\phi^0 = \frac{1}{v^2}\left[-\frac{\beta n}{r} C K_n(vr) - j\omega\mu_0 v D K_n'(vr)\right] e^{jn\phi}$$

Boundary conditions

In this case, there is only one interface to be taken into account, the core/cladding one at $r = a$. Because this is an interface between two dielectric media, there are no superficial electrical charge nor current densities. This means that the electric and magnetic fields of a propagating EM wave must comply with the following:

- E_z^0 continuous at $r = a$
- E_ϕ^0 continuous at $r = a$
- H_z^0 continuous at $r = a$
- H_ϕ^0 continuous at $r = a$

Using previously obtained equations for core and cladding, these conditions turn into:

$$AJ_n(h_1 a) = CK_n(va) \leftrightarrow AJ_n(h_1 a) - CK_n(va) = 0$$

$$BJ_n(h_1 a) = DK_n(va) \leftrightarrow BJ_n(h_1 a) - DK_n(va) = 0$$

$$B\frac{\beta n}{h_1^2 a}J_n(h_1 a) - A\frac{j\omega\varepsilon_1}{h_1}J_n'(h_1 a) + D\frac{\beta n}{v^2 a}K_n(va) - C\frac{j\omega\varepsilon_2}{v}K_n'(va) = 0 \qquad (18)$$

$$A\frac{\beta n}{h_1^2 a}J_n(h_1 a) + B\frac{j\omega\mu_0}{h_1}J_n'(h_1 a) + C\frac{\beta n}{v^2 a}K_n(va) + D\frac{j\omega\mu_0}{v}K_n'(va) = 0$$

These four equations form a four variables system of equations and for its solution to be non-trivial, its determinant should yield zero:

$$\begin{vmatrix} J_n(h_1 a) & 0 & -K_n(va) & 0 \\ 0 & J_n(h_1 a) & 0 & -K_n(va) \\ -\frac{j\omega\varepsilon_1}{h_1}J_n'(h_1 a) & \frac{\beta n}{h_1^2 a}J_n(h_1 a) & -\frac{j\omega\varepsilon_2}{v}K_n'(va) & \frac{\beta n}{v^2 a}K_n(va) \\ \frac{\beta n}{h_1^2 a}J_n(h_1 a) & \frac{j\omega\mu_0}{h_1}J_n'(h_1 a) & \frac{\beta n}{v^2 a}K_n(va) & \frac{j\omega\mu_0}{v}K_n'(va) \end{vmatrix} = 0$$

Solving the above determinant and after some mathematical manipulation, one can find an equation for these conditions:

$$\left(\frac{\omega}{c}\right)^2\left[\frac{J_n'(h_1a)}{h_1J_n(h_1a)}+\frac{K_n'(va)}{vK_n(va)}\right]\left[n_1^2\frac{J_n'(h_1a)}{h_1J_n(h_1a)}+n_2^2\frac{K_n'(va)}{vK_n(va)}\right]=\left(\frac{\beta n}{a}\right)^2\left(\frac{1}{h_1^2}+\frac{1}{v^2}\right)^2\leftrightarrow$$

$$\leftrightarrow\left[\frac{J_n'(h_1a)}{h_1J_n(h_1a)}+\frac{K_n'(va)}{vK_n(va)}\right]\left[n_1^2\frac{J_n'(h_1a)}{h_1J_n(h_1a)}+n_2^2\frac{K_n'(va)}{vK_n(va)}\right]=\left(\frac{\beta n}{k_0a}\right)^2\left(\frac{1}{h_1^2}+\frac{1}{v^2}\right)^2 \quad (19)$$

where $k_0 = {}^{\omega}/_{c} = {}^{2\pi}/_{\lambda_0}$, is the vacuum wavenumber.

Equation 19 is the general characteristic equation for circular dielectric waveguides and can be used for all considered modes, that is, TE, TM, HE and EH modes.

Normalized parameters

The following normalized parameters analysis, effective refractive index included, is based on [7].

Let us consider,

$$H = h_1a = a\sqrt{(k_0n_1)^2-\beta^2}$$

and

$$W = va = a\sqrt{\beta^2-(k_0n_2)^2}$$

H and W are positive real constants, where H describes the wave's phase progression along the core and W its exponential decay at the cladding.

Normalized frequency

When optical fibres are involved, another parameter is considered and is known as the normalized frequency parameter V. It establishes single mode regime conditions and sets the number of propagating modes in a circular dielectric waveguide. Its definition is:

$$V^2 = H^2+W^2 \leftrightarrow V^2 = a^2(k_0^2n_1^2-\beta^2+\beta^2-k_0^2n_2^2)$$

$$\boxed{V = ak_0\sqrt{n_1^2-n_2^2} = ak_0NA}$$

where $NA = \sqrt{n_1^2-n_2^2}$ is the numerical aperture.

Normalized propagation constant

Parameter used for phase constant calculus on a waveguide. Its definition being:

$$b = 1 - \frac{H^2}{V^2} = \frac{W^2}{V^2} = \frac{\left(\frac{\beta}{k_0}\right)^2 - n_2^2}{n_1^2 - n_2^2}$$

For low Δ (contrast) devices, also known as weakly guided ones, which is what usually happens on commercial optical fibres, the following approximation can be utilized:

$$n_1 \cong n_2 \rightarrow \Delta \cong \frac{n_1 - n_2}{n_1}$$

$$b \cong \frac{\left(\frac{\beta}{k_0}\right) - n_2}{n_1 - n_2}$$

Effective refractive index

The effective refractive index n_{eff}, must be such that verifies the identity:

$$\beta = n_{eff} k_0$$

where β is the longitudinal propagation constant and k_0 is the vacuum wavenumber, both for given operating device and wavelength λ.

Figure 24 shows n_{eff} (which is directly related to b through β) dependency of V, for skew (hybrid – HE/EH) and meridional (TE/TM) propagation modes. In this case, a $5\ \mu m$ core radius circular dielectric waveguide with its refractive index yielding 1.448 ($n_1 = 1.448$) and the cladding's refractive index is 1.444 ($n_2 = 1.444$).

As usual, Figure 24 has been plotted with Matlab help, using the OFT (Optical Fiber Toolbox). This toolbox consists of a set of functions created to model three layers (core, cladding and surrounding medium) tapered optical fibers but, through some careful modification, is also perfectly able to model two layers (core and cladding) optical fiber (for a more detailed Matlab functions description, please refer to Appendix D).

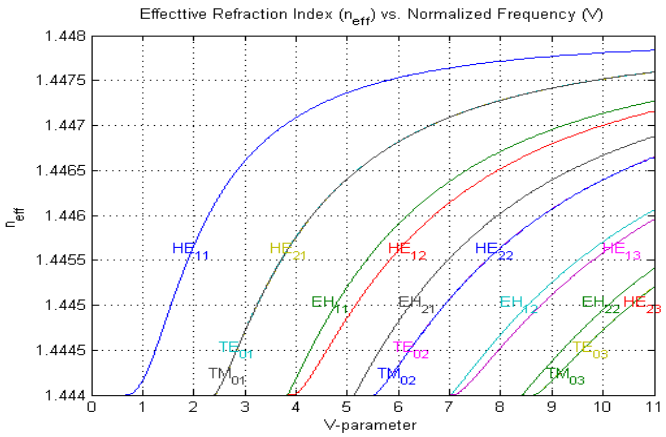

Figure 24 - Effective refractive index behavior as normalized frequency increases.

Meridional modes (TE/TM)

When $n = 0$, the system of equations 18 assumes the form, as in [4]:

$$\begin{bmatrix} J_0(h_1 a) & 0 & -K_0(va) & 0 \\ 0 & J_0(h_1 a) & 0 & -K_0(va) \\ -\frac{j\omega\varepsilon_1}{h_1} J_0'(h_1 a) & 0 & -\frac{j\omega\varepsilon_2}{v} K_0'(va) & 0 \\ 0 & \frac{j\omega\mu_0}{h_1} J_0'(h_1 a) & 0 & \frac{j\omega\mu_0}{v} K_0'(va) \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = 0$$

Also, as stated earlier on, whether on TE modes $\rightarrow \mathrm{A} = \mathrm{C} = 0$ and on TM modes $\rightarrow \mathrm{B} = \mathrm{D} = 0$. These are also called the meridional modes for they cross the axis of the waveguide as the diagram on Figure 25 tries to show:

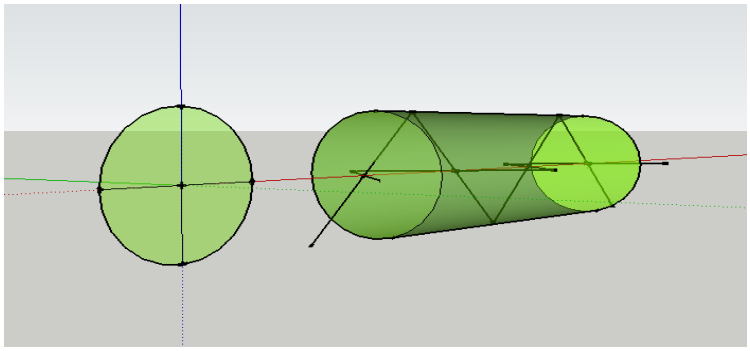

Figure 25 - Meridional modes (TE and TM).

Closely observing the previous system of equations, it is perceivable that is possible to break it in two independent system of equations, one for variables A and C and the other for variables B and D. The resulting systems of equations, associated to each group of constants, are:

$$\begin{bmatrix} J_0(h_1 a) & -K_0(va) \\ -\frac{\varepsilon_1}{h_1} J_0'(h_1 a) & -\frac{\varepsilon_2}{v} K_0'(va) \end{bmatrix} \begin{bmatrix} A \\ C \end{bmatrix} = 0$$

$$\begin{bmatrix} J_0(h_1 a) & -K_0(va) \\ \frac{1}{h_1} J_0'(h_1 a) & \frac{1}{v} K_0'(va) \end{bmatrix} \begin{bmatrix} B \\ D \end{bmatrix} = 0$$

Now, for a non-trivial solution of the first system of equations, it requires its determinant yielding zero, resulting in the equation for TM modes ($\mathrm{B} = \mathrm{D} = 0 \rightarrow H_z^0 = 0$):

$$n_1^2 \frac{J_0'(h_1 a)}{h_1 J_0(h_1 a)} + n_2^2 \frac{K_0'(va)}{vK_0(va)} = 0$$

Using the equivalencies $J_0'(x) = -J_1(x)$ and $K_0'(x) = -K_1(x)$, the characteristic equation for TM modes on circular dielectric waveguides is obtained:

$$n_1^2 \frac{J_1(h_1 a)}{h_1 J_0(h_1 a)} + n_2^2 \frac{K_1(va)}{vK_0(va)} = 0$$

Going through the same procedure but for the second system of equations, the characteristic equation for TE modes ($\mathrm{A} = \mathrm{C} = 0 \rightarrow E_z^0 = 0$) on circular dielectric waveguides is obtained:

$$\frac{J_1(h_1 a)}{h_1 J_0(h_1 a)} + \frac{K_1(va)}{vK_0(va)} = 0$$

Skew/hybrid modes (HE/EH)

For $n \neq 0$ ($E_z^0 \neq 0$ and $H_z^0 \neq 0$), equations 19 (general characteristic equation) becomes more complex and requires numerical methods in order to find its solution. The obtained modes, in this case, are the hybrid modes HE and EH and Figure 26 is an attempt to explain them schematically. Mainly, these modes propagate by TIR along the dielectric waveguide but they do not cross its axis ($x \neq 0, y \neq 0, z$).

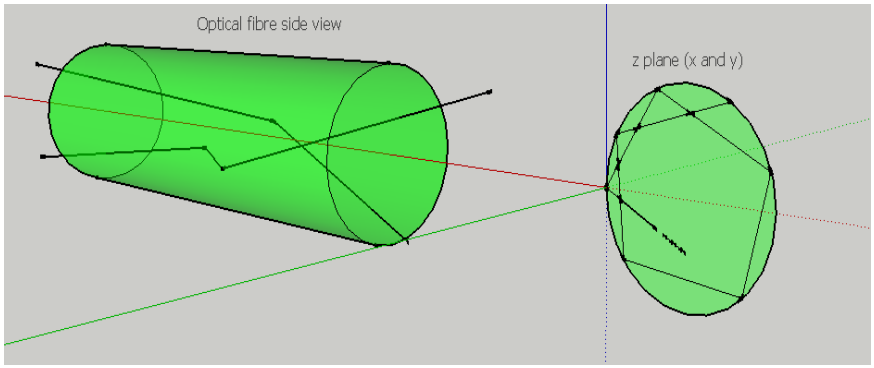

Figure 26 - Skew or hybrid modes (EH or HE modes).

The characteristic equation for EH/HE modes is somewhat different from the one found for meridional modes (TE/TM) and is obtained from equation 19. Here is its definition:

$$\left[\frac{J_n'(ha)}{hJ_n(ha)} + \frac{K_n'(va)}{vK_n(va)}\right]\left[\frac{J_n'(ha)}{hJ_n(ha)} + \left(\frac{n_2}{n_1}\right)^2 \frac{K_n'(va)}{vK_n(va)}\right] = n^2\left[\frac{1}{(ha)^2} + \frac{1}{(va)^2}\right]\left[\frac{1}{(ha)^2} + \left(\frac{n_2}{n_1}\right)^2 \frac{1}{(va)^2}\right]$$

Cut off conditions

As before on planar dielectric waveguides, the modes cut off frequency can be obtained through cut off condition $v = 0$. So, there are two situations to account for, as in [6] and [7]:

- When v parameter is real and finite, which means,

$$v^2 = \beta^2 - k_0^2 n_2^2$$

  - The EM field decays exponentially in the cladding, as distance to the centre increases, and the energy is confined in the core of the dielectric waveguide.

- When $v = 0$,

$$\beta^2 = k_0^2 n_2^2$$

  - The phase velocity becomes equal to the wave's progression in the cladding.
  - The cladding's EM field does not decay, as distance to the core increases, and the energy is not confined in the core.
  - In this situation,

$$V = a k_0 n_1 \sqrt{2\Delta}$$

and

$$\omega_c = \frac{cV}{n_1 a \sqrt{2\Delta}} \qquad \omega_c \rightarrow wave's\ progression\ in\ the\ cladding$$

Modes TE_{0n} and TM_{0n}

$$\boxed{\frac{J_1(h_1 a)}{h_1 J_0(h_1 a)} = -\frac{K_1(va)}{vK_0(va)}} \quad TE_{0n} \qquad \boxed{\frac{J_1(h_1 a)}{h_1 J_0(h_1 a)} = -\left(\frac{n_2}{n_1}\right)^2 \frac{K_1(va)}{vK_0(va)}} \quad TM_{0n}$$

Cut off condition requires that:

$$\frac{J_1(h_1 a)}{h_1 J_0(h_1 a)} \rightarrow \infty \quad \longrightarrow \quad J_0(h_1 a) = 0$$

Skew/hybrid modes EH/HE

$$\boxed{\frac{J_{n+1}(ha)}{hJ_n(ha)} = -\frac{K_{n+1}(va)}{vK_n(va)}} \quad EH_{np} \qquad \boxed{\frac{J_{n-1}(ha)}{hJ_n(ha)} = -\frac{K_{n-1}(va)}{vK_n(va)}} \quad HE_{np}$$

Let us assume that TE_{0n} and TM_{0n} modes have their cut off frequencies defined by x and this is the n^{th} solution of the equation $J_0(x) = 0$. The index n represents the order of each Bessel function roots. Table 2 shows the obtained results:

Table 2 - Modes cut off conditions.

n	mode	cut off condition
0	TE_{0p} TM_{0p}	$J_0(h_1 a) = 0$
1	HE_{1p} EH_{1p}	$J_1(h_1 a) = 0$
≥ 2	EH_{np}	$J_n(h_1 a) = 0$
	HE_{np}	$\left(\frac{n_1^2}{n_2^2} + 1\right) J_{n-1}(h_1 a) = \frac{h_1 a}{n-1} J_n(h_1 a)$

HE_{11} mode's cut off frequency is zero (similarly to odd modes on planar dielectric waveguides), meaning it is always present for any operating frequency or waveguide diameter. As for the TE_{01} and TM_{01} modes, cut off frequency is associated to J_0 first zero, which occurs at, approximately, 2.4048. So, if $V < 2.4048$ the optical fibre operates in single-mode regime and if $V > 2.4048$, then it is the multi-mode regime.

As an example, consider TE and TM cut off modes. According to Table 2, $h_1 a = 2.4048$ and $v = 0$, which means $V = 2.4048$. Thus, if $V > 2.4048$, TE_{01} and TM_{01} will be able to propagate. On the other hand, if $V \leq 2.4048$, only HE_{11} will be able to propagate, meaning that the waveguide is operating in single-mode regime.

One way of finding, approximately, the number of propagating modes in a circular dielectric waveguide is:

$$M \cong \frac{V^2}{2}$$

Linearly polarized modes (LP)

Almost all manufactured optical fibres are classified as weakly guiding devices. This means that the core's refractive index is just slightly higher than that of the cladding, which leads to the weak-guidance approximation ($n_1 \approx n_2$). This gives rise to a new set of modes called LP (Linearly Polarized) modes. These modes have a similar spatial intensity pattern in the transverse plane as TE, TM and hybrid modes but their EM fields each occur in single Cartesian directions in the transverse plane (linearly polarized), meaning, their electrical/magnetic components are orthogonal, simplifying their characteristic equations.

This happens because hybrid modes, which are able to degenerate (assume the same propagation constant (β) of the next mode) can be summed up with either another hybrid mode or a TE or TM mode, originating the LP modes. Being linearly polarized, the polarized light from a commercial laser is able to excite this superposition of true fibre modes.

Let us consider, as an example, an optical fibre core diameter of 5.6 μm and refraction index of 1.448 and silica (refraction index of 1.444) for cladding, on an operating wavelength $\lambda = 900\ nm$. Once again, Matlab and its optical fibre toolbox was of great help to calculate and plot the following set of figures (see Appendix D). Figure 27 shows HE_{11} or LP_{01} mode's electrical field

components. As can be observed, this mode is already linearly polarized for its y and z components being orthogonal, null each other, leaving the transverse one as the mode's contributing component.

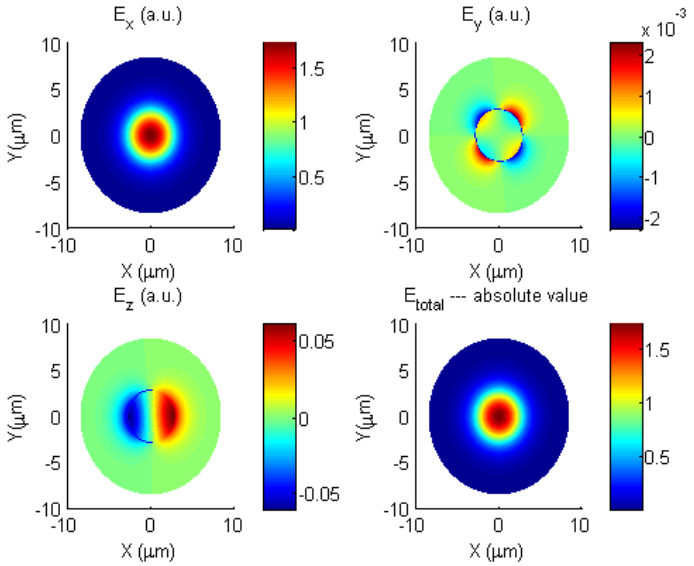

Figure 27 - Fundamental (LP01 or HE11) mode electrical components.

Next figure, Figure 28, shows HE_{21} degenerate mode electrical field components for it is one of the two LP_{11} contributing modes. One can notice the y and z components influence on the overall result of this mode electrical field for it is not only the transverse component that is present on the outcome.

Figure 29 shows TM_{01} mode electrical field components and, as expected, $E_z \neq 0$ ($H_z = 0$) and Figure 30 shows the summed up result, meaning, LP_{11} mode. As with LP_{01}, LP_{11} mode is linearly polarized and its components have influence on the final result of the mode's transverse electrical field. This mode, LP_{11}, can also be the result of the sum of HE_{21} and TE_{01} modes, as mentioned before. On Table 3 there are examples of some LP modes and their constituents.

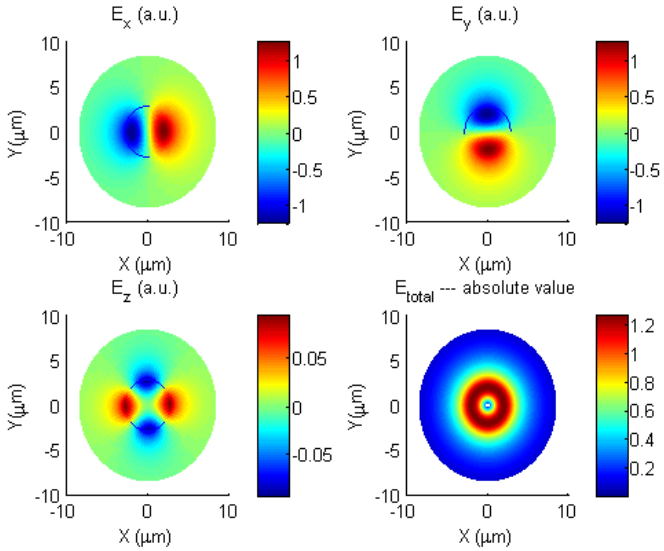

Figure 28 - HE21 mode electrical components.

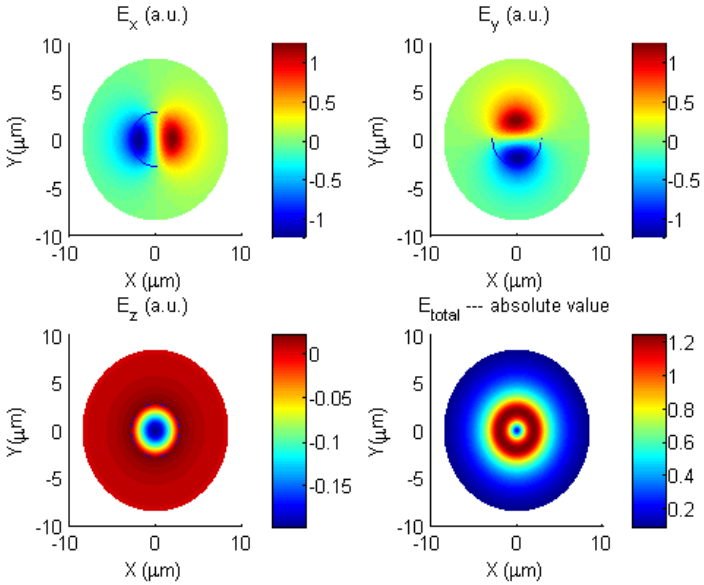

Figure 29 - TM01 mode electrical components.

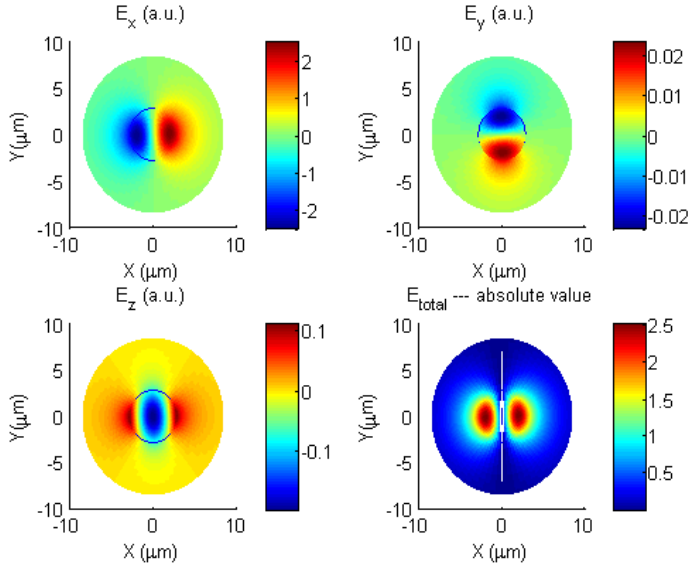

Figure 30 - LP11 mode electrical components.

The construction and labelling rules for LP modes are the following:

- $LP_{0\mu} = HE_{1\mu}$
- $LP_{1\mu} = HE_{2\mu} + TE_{0\mu}$ or $HE_{2\mu} + TM_{0\mu}$
- $LP_{m\mu} = HE_{m+1,\mu} + EH_{m-1,\mu}$

Table 3 - LP modes and their constituents.

$(m\mu)$	(01)	(11)	(21)	(02)
$LP_{m\mu}$ designation	LP_{01}	LP_{11}	LP_{21}	LP_{02}
Hybrid mode designation	HE_{11}	HE_{21}	HE_{31}	HE_{12}
	TE_{01}	EH_{11}		
	TM_{01}			

2.3.2. Rectangular dielectric waveguides

In this sub-chapter it is going to be analysed two main categories of these structures, namely, photonic wires and rib waveguides. For an accurate treatment of these structures, numerical methods are required but a good enough understanding can be obtained through analytical approximation methods. As such, for the former one the analytical method proposed by Marcatili is going to be followed (there is also the Kumar's method, which is more accurate by taking into consideration the perturbation originated by the evanescent fields) and, for the latter, is the Effective Index Method the taken approach because rib waveguides structure is difficult to analyse through either Marcatili or Kumar methods.

Marcatili method – Basic equations

The main assumption in this method is that the EM fields represented by the shaded areas in Figure 31 are not taken into account for the guided mode decays very fast within the cladding area and, thus, there will be no boundary conditions imposed on the EM fields on these shaded zones [8].

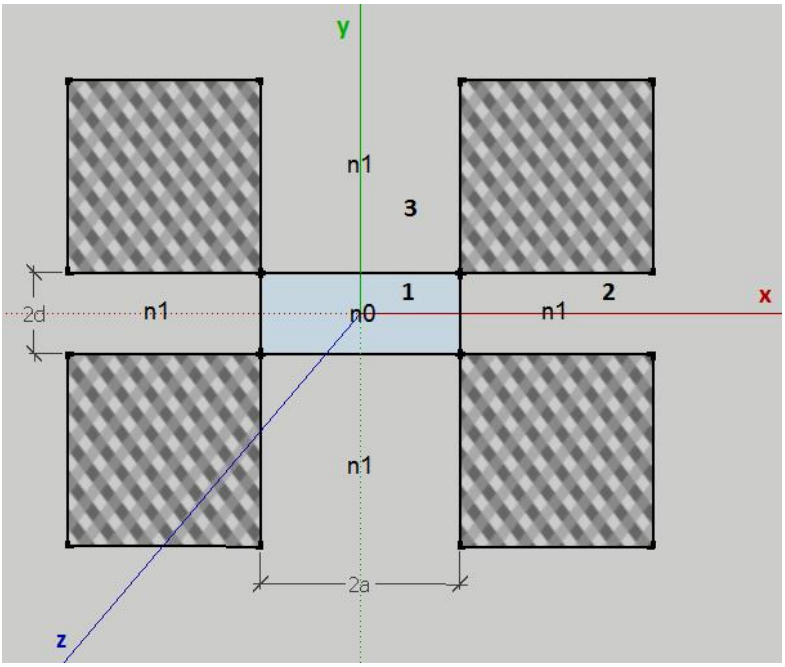

Figure 31 - Photonic wire seen by Marcatili method.

This method starts by considering the analysis for the mode where E_x and H_y are predominant. According to Marcatili, setting $H_x = 0$ and considering that $n^2 = \varepsilon_r$ and $\gamma = j\beta$ (propagating mode) on [8], equations 2.3 and 2.4, the wave equation and the components for the EM fields are as follows:

$$\frac{\partial^2 H_y}{\partial x^2} + \frac{\partial^2 H_y}{\partial y^2} + (h^2 n^2 - \beta^2) H_y = 0 \tag{20}$$

$H_x = 0$

$$E_x = \frac{-j\omega\mu_0 H_y + \frac{\partial E_z}{\partial x}}{-j\beta} \leftrightarrow E_x = \frac{\omega\mu_0}{\beta} H_y - \frac{1}{j\beta}\frac{\partial E_z}{\partial x}$$

Because,

$$E_z = \frac{-j}{\omega\varepsilon_0 n^2}\frac{\partial H_y}{\partial x} \rightarrow \frac{\partial y}{\partial x} = \frac{-j}{\omega\varepsilon_0 n^2}\frac{\partial^2 H_y}{\partial x^2}$$

thus,

$$\boxed{E_x = \frac{\omega\mu_0}{\beta} H_y + \frac{1}{\omega\varepsilon_0 n^2 \beta}\frac{\partial^2 H_y}{\partial x^2}} \tag{21}$$

Using similar algebra's simplification, the rest of the EM fields equations are:

$$\boxed{E_y = \frac{1}{\omega\varepsilon_0 n^2 \beta}\frac{\partial^2 H_y}{\partial x \partial y}}$$

$$\boxed{E_z = \frac{-j}{\omega\varepsilon_0 n^2}\frac{\partial H_y}{\partial x}}$$

$$\boxed{H_z = \frac{-j}{\beta}\frac{\partial H_y}{\partial y}}$$

Then again, considering analysis for the mode where E_y and H_x are predominant, assuming $H_y = 0$ and the rest of previous considerations, the wave equation and the components for the EM fields are as follows:

$$\frac{\partial^2 H_x}{\partial x^2} + \frac{\partial^2 H_x}{\partial y^2} + (h^2 n^2 - \beta^2) H_x = 0 \tag{22}$$

$H_y = 0$

$$\boxed{E_x = -\frac{1}{\omega\varepsilon_0 n^2 \beta}\frac{\partial^2 H_y}{\partial x \partial y}} \tag{23}$$

$$E_y = -\frac{\omega\mu_0}{\beta}H_x - \frac{1}{\omega\varepsilon_0 n^2\beta}\frac{\partial^2 H_x}{\partial y^2}$$

$$E_z = \frac{j}{\omega\varepsilon_0 n^2}\frac{\partial H_x}{\partial y}$$

$$H_z = \frac{-j}{\beta}\frac{\partial H_x}{\partial x}$$

Modes described by equations 21 are known as E_{pq}^x, where p and q are integers, for E_x and H_y are the dominant EM fields. On the other hand, modes described by equations 23 are referred as E_{pq}^y, where p and q are integers, since E_y and H_x are the dominant EM fields. Next sub-chapter will describe the solution method of the dispersion equation for E_{pq}^x modes and only the E_{pq}^y final results will be presented because the procedure is identical.

Dispersion equations for E_{pq}^x and E_{pq}^y modes

The rectangular dielectric waveguide on Figure 31 is symmetrical with respect to x and y axes and, for this reason, it suffices to analyse numbered regions **1** to **3**. The solution fields that satisfy wave equation 20 are:

$$H_y = \begin{cases} A\cos(h_x x - \theta)\cos(h_y y - \varphi) & zone\ \mathbf{1} \\ A\cos(h_x a - \theta)e^{-\gamma_x(x-a)}\cos(h_y y - \varphi) & zone\ \mathbf{2} \\ A\cos(h_x x - \theta)e^{-\gamma_y(y-d)}\cos(h_y d - \varphi) & zone\ \mathbf{3} \end{cases}$$

where transverse wavenumbers h_x, h_y, γ_x and γ_y and the optical phases θ and φ are given by:

$$\begin{cases} -h_x^2 - h_y^2 + h^2 n_1^2 - \beta^2 = 0 & zone\ \mathbf{1} \\ \gamma_x^2 - h_y^2 + h^2 n_0^2 - \beta^2 = 0 & zone\ \mathbf{2} \\ -h_x^2 - \gamma_y^2 + h^2 n_0^2 - \beta^2 = 0 & zone\ \mathbf{3} \end{cases} \quad (24)$$

and

$$\begin{cases} \theta = (p-1)\dfrac{\pi}{2} & p = 1,2,3,\ldots \\ \varphi = (q-1)\dfrac{\pi}{2} & q = 1,2,3,\ldots \end{cases}$$

Integers p and q go from 1 onwards, similarly to the optical fibre mode indexes and they reflect the number of electrical peaks along x and y directions, respectively. This means, mode E_{11}^x or E_{11}^y only have one peak on x or y axis; on the other hand, mode E_{12}^x or E_{12}^y has one electric peak on x axis and two electric peaks at the y axis. Figure 32 represents the electrical field distribution in x and y directions, when E_{pq}^x and E_{pq}^y on modes E_{11}^x, E_{11}^y, E_{12}^x, E_{12}^y, E_{21}^x and E_{21}^y, are involved.

Matlab and its OFT, with some functions modifications (for a detailed understanding of how this was accomplished, please see Appendix F), were used to plot the associated x and y axis fields depicted on Figure 32, considering a square waveguide where:

- $2d = 2a = 0.5\ \mu m$;
- $\lambda = 1000\ nm$;
- $n_1 = 1.55$;
- $n_2^x = 2.156$;
- $n_2^y = 2.232$;
- $n_2^z = 2.232$.

Applying boundary conditions that the electrical field $E_z \propto \frac{1}{n^2}\frac{\partial H_y}{\partial x}$ should be continuous at $x = a$ and the magnetic field $H_z \propto \frac{\partial H_y}{\partial y}$ should be continuous at $y = d$, the following dispersion equations can be obtained:

$$h_x a = (p-1)\frac{\pi}{2} + \tan^{-1}\left(\frac{n_1^2 \gamma_x}{n_0^2 h_x}\right) \tag{25}$$

$$h_y d = (q-1)\frac{\pi}{2} + \tan^{-1}\left(\frac{\gamma_y}{h_y}\right) \tag{26}$$

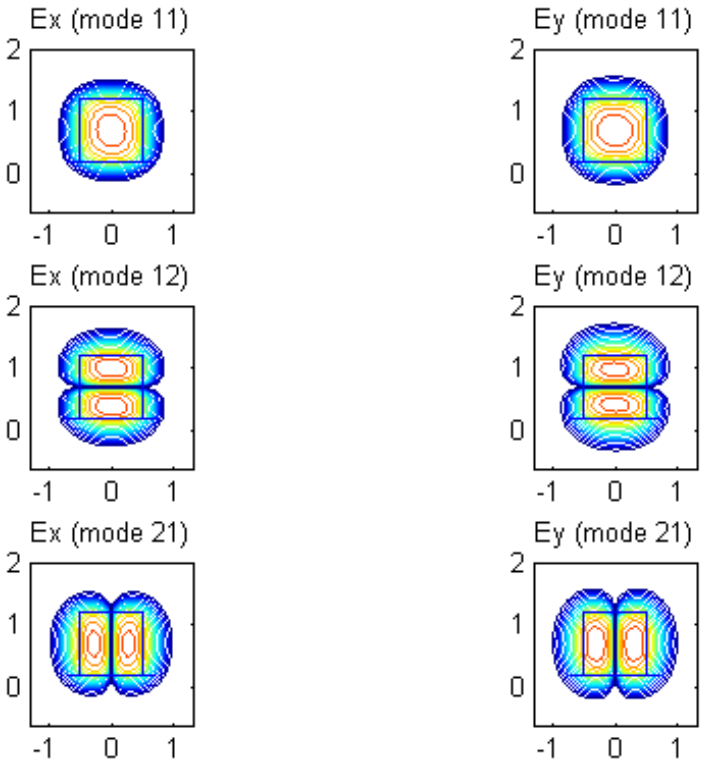

Figure 32 - Marcatili method E_x and E_y mode distribution.

Transversal wavenumbers h_x, h_y, γ_x and γ_y can be related through equation 24 such as:

$$\begin{aligned}\gamma_x^2 &= h_y^2 - h^2 n_0^2 - h_x^2 - h_y^2 + h^2 n_1^2 \leftrightarrow \gamma_x^2 = h^2(n_1^2 - n_0^2) - h_x^2 \\ \gamma_y^2 &= h_x^2 - h_x^2 - h_y^2 + h^2 n_1^2 - h^2 n_0^2 \leftrightarrow \gamma_y^2 = h^2(n_1^2 - n_0^2) - h_y^2\end{aligned} \tag{27}$$

Again, through some algebraic manipulation, one is able to find h_x and h_y expressions through equation 25 and the first expression of equations 27 for the former and through equation 26 and the second expression of equations 27 for the latter. Propagation constant β is then obtained from:

$$\beta^2 = -h_x^2 - h_y^2 + h^2 n_1^2 \leftrightarrow \beta^2 = h^2 n_1^2 - \left(h_x^2 + h_y^2\right)$$

For E_{pq}^y mode, magnetic field H_x is identical to H_y expression on E_{pq}^x mode, only this time with different boundary conditions and can be expressed as:

$$H_x = \begin{cases} A\cos(h_x x - \theta)\cos\left(h_y y - \varphi\right) & zone\ \mathbf{1} \\ A\cos(h_x a - \theta)e^{-\gamma_x(x-a)}\cos\left(h_y y - \varphi\right) & zone\ \mathbf{2} \\ A\cos(h_x x - \theta)e^{-\gamma_y(y-d)}\cos\left(h_y d - \varphi\right) & zone\ \mathbf{3} \end{cases}$$

This time, boundary conditions are the magnetic field $H_z \propto \frac{\partial H_x}{\partial x}$ continuous at $x = a$ and the electrical field $E_z \propto \frac{1}{n^2}\frac{\partial H_x}{\partial y}$ also continuous at $y = d$, resulting on the following dispersion equations:

$$h_x a = (p-1)\frac{\pi}{2} + \tan^{-1}\left(\frac{\gamma_x}{h_x}\right)$$

$$h_y d = (q-1)\frac{\pi}{2} + \tan^{-1}\left(\frac{n_1^2\gamma_y}{n_0^2 h_y}\right)$$

Effective Index Method

Rib waveguides as the one depicted on Figure 33 are difficult to analyse through Marcatili's method for the waveguide structure is too complex to be dealt with by its Cartesian division. In this case, the effective index method is the approximation best suited to deal with these complicated structures [8]. The following assumptions refer only to E_{pq}^x mode on the rib waveguide, since for the E_{pq}^y mode the analysis is identical.

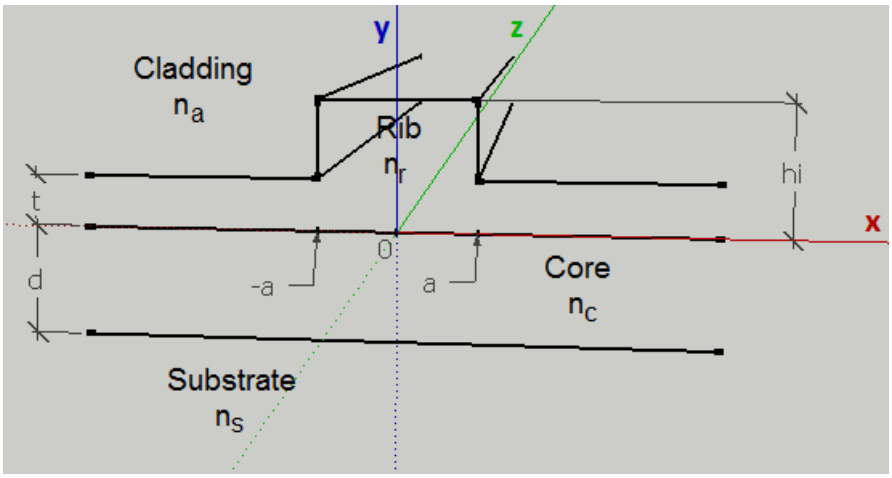

Figure 33 - Rib waveguide.

Wave equation for this mode is given by the following expression:

$$\frac{\partial^2 H_y}{\partial x^2} + \frac{\partial^2 H_y}{\partial y^2} + [h^2 n^2(x,y) - \beta^2] H_y = 0 \tag{28}$$

Much like before on previous analysis, it is assumed that the magnetic field can be expressed, through the separation of variables method, as:

$$H_y(x,y) = X(x)Y(y) \tag{29}$$

One has to be careful to check the validity of the variables separation method when applied on waveguide structures or wavelengths where this method is not valid. Assuming everything is accordingly, substituting equation 29 into equation 28 and dividing everything by XY leads to:

$$\frac{1}{X}\frac{\partial^2 X}{\partial x^2} + \frac{1}{Y}\frac{\partial^2 Y}{\partial y^2} + [h^2 n^2(x,y) - \beta^2] H_y = 0 \tag{30}$$

Now, is time to separate the first member in equation 30 into two independent equations and add/subtract the y independent value $h^2 n_{eff}^2(x)$ to obtain,

$$\frac{1}{Y}\frac{\partial^2 Y}{\partial y^2} + \left[h^2 n^2(x,y) - h^2 n_{eff}^2(x)\right] = 0 \tag{31}$$

$$\frac{1}{X}\frac{\partial^2 X}{\partial x^2} + \left[h^2 n_{eff}^2(x) - \beta^2\right] = 0 \tag{32}$$

where $n_{eff}(x)$ is the *effective index distribution* and $n(x,y)$ is the *actual refractive-index profile*.

The procedure to solve this problem is to determine $n_{eff}(x)$ by solving equations 31 and 32. The variation of the actual refractive index $n(x,y)$ is schematically represented on Figure 34, where

$n_r = n_s$ and s is the height of the rib. The value of s depends of x position in a way that is described by equation 33:

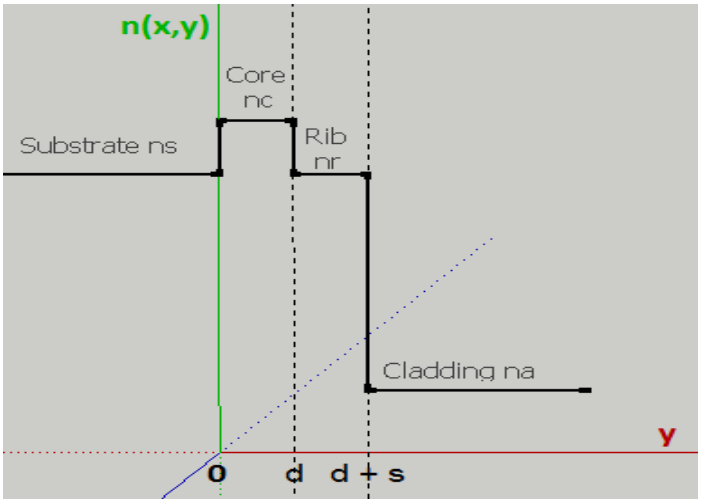

Figure 34 - Actual refractive index profile n(x,y) variation.

$$s = \begin{cases} hi & 0 \le |x| \le a \\ t & |x| > a \end{cases} \tag{33}$$

From boundary condition that $H_z \propto \frac{\partial H_y}{\partial y}$ should be continuous at $y = 0$, d and $d + s$, the continuity condition for $\frac{\partial Y}{\partial y}$ at the afore mentioned boundaries is obtained. Thus, the dispersion equation for the four-layer slab waveguide shown in Figure 34 is given by:

$$\sin(\kappa d - 2\theta) = \sin(\kappa d) e^{-2(\sigma s + \varphi)} \tag{34}$$

where the above parameters are,

$$\theta = \tan^{-1}\left(\frac{\sigma}{\kappa}\right)$$

$$\varphi = \tan^{-1}\left(\frac{\sigma}{\gamma}\right)$$

$$\kappa = h\sqrt{n_c^2 - n_{eff}^2}$$

$$\sigma = h\sqrt{n_{eff}^2 - n_s^2}$$

$$\gamma = h\sqrt{n_{eff}^2 - n_a^2}$$

Solution of equation 34 when $s = hi (0 \le |x| \le a)$ gives the effective index $n_{eff}(hi)$ for $0 \le |x| \le a$ and the solution when $s = t (|x| > a)$ gives the effective index $n_{eff}(t)$ for $|x| > a$. Then the effective index distribution $n_{eff}(x)$ will result as depicted on Figure 35.

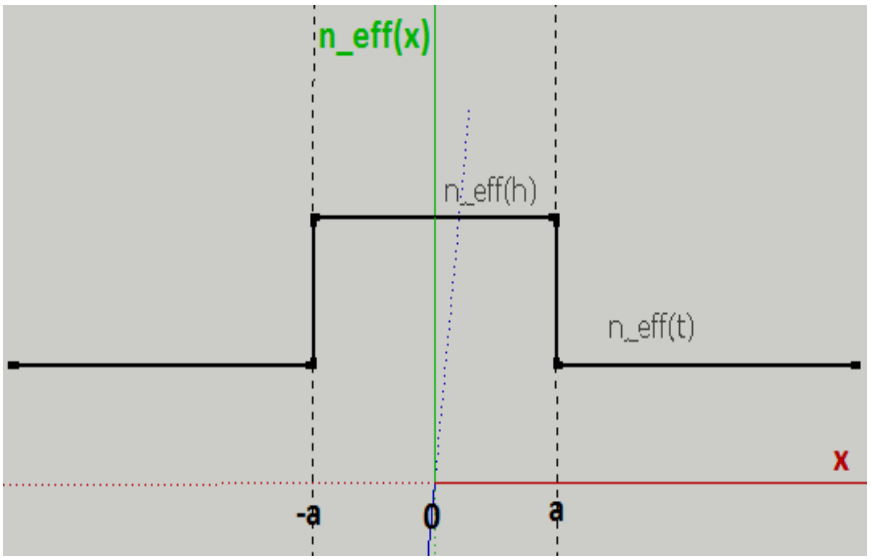

Figure 35 - Effective index distribution, n_eff(x).

The solution for wave equation 32 is obtained by solving the three symmetrical layer slab waveguide with the boundary condition $E_z \propto \frac{1}{n^2}\frac{\partial H_y}{\partial x}$ that should be continuous at $x = \pm a$ and, as though, resulting on the following dispersion equation,

$$u\tan(u) = \frac{n_{eff}^2(hi)}{n_{eff}^2(t)}w$$

where,

$$u = ha\sqrt{n_{eff}^2(hi) - \left(\frac{\beta}{h}\right)^2}$$

$$w = ha\sqrt{\left(\frac{\beta}{h}\right)^2 - n_{eff}^2(t)}$$

For the E_{pq}^y mode, the analysis is much like the same of what has been done previously for the E_{pq}^x mode. For this reason, there will not be a detailed analysis of this mode and only its dispersion equation will be presented here,

$$\sin(\kappa d - 2\theta) = \sin(\kappa d)e^{-2(\sigma s+\varphi)}$$

$$u\tan(u) = w$$

where,

$$\theta = \tan^{-1}\left(\frac{\sigma n_c^2}{\kappa n_s^2}\right)$$

$$\varphi = \tan^{-1}\left(\frac{\sigma n_a^2}{\gamma n_s^2}\right)$$

the unmentioned parameters keep the same expressions and, as though, this concludes the analysis for the Effective Index Method.

Chapter III

Non-ideal photonic waveguides

In this chapter, it is going to be presented some simulations of realistic (non-ideal) materials on Optiwave® OptiFDTD 32bit version, a free simulation tool that implements the FDTD method to solve Maxwell's equations associated to EM field propagation in dielectric waveguides. It will be utilized a-Si:H for the waveguide core material that is going to propagate the EM fields, embedded on an SiO_2 (silicon dioxide, a.k.a. silica or quartz) substrate and, when on 3D simulations, air for the surrounding upper layer (a.k.a. cladding). The reasons justifying the choice of a-Si:H as the core material [1] will be summarized here:

- Excellent electrical quality and low optical absorption.
- Low power consumption and latency and high bandwidth.
- Compatibility between a-Si:H photonics and CMOS processes.
- Cost-effective fabrication process solution.
- a-Si:H can be deposited on virtually any material.
- Fully compatible with backend thermal budget in CMOS fabrication process.

Recently, a-Si:H has been reported by many authors as a possible candidate for mass production of photonic circuits [9, 10, 11, 12, 13]. State of the art good quality a-Si:H can be deposited by PECVD on almost any material. This technique has reached maturity mostly due to the large investment directed to mass production of a-Si:H solar cells [14] and thin film transistor for active matrix flat panel displays [15].

Nowadays, it is possible to fabricate good quality a-Si:H at low temperatures (200 - 400 °C), making amorphous silicon deposition compatible with back-end CMOS processing. The amorphous state of silicon has been intensively studied in the last decade of the XX century and it is well known that

the electronic and optical properties of the films are strongly influenced by the deposition technique and conditions. The presence of defects and dangling bonds in the lattice might result in high density of localized states at energies below the energy gap, producing photon absorption in the near IR range at telecommunication wavelengths. Optimal refractive index of a-Si:H, measured by reflectometry [16] was found to be 3.65 at 1550 nm, which is higher than crystalline silicon, thus leaving the latter not as well positioned as an alternative. However, hydrogen concentration is strongly dependent on deposition temperature and determines sub-gap absorption coefficient, producing small variations of the amorphous silicon optical functions [17].

On the following simulations, low loss ($2.5\ dBcm^{-1}$) non-planar a-Si:H waveguides, as in [1], are going to be considered. It is assumed that these waveguides have been fabricated through a PECVD process, assuring passivation of a-Si dangling bonds and thus capable of achieving such low losses.

3.1. Absorption and Extinction coefficients α and κ

Before starting to describe the developed simulations in order to obtain the necessary results, some parameter definitions and their interdependency must be presented. These parameters are:

1. Absorption coefficient, here designated by the α symbol and expresses the attenuation/gain experienced by an EM field when propagating through a structure of a pre-determined length. Thus, this coefficient represents a field's amplitude multiplying factor by path unit ($\alpha > 1 \rightarrow gain$ and $\alpha < 1 \rightarrow loss$). Usual units are cm^{-1}, which reflect the attenuation/gain that is experienced by an EM field when it propagates over a length of $1\ cm$.
2. Extinction coefficient, here designated by the κ symbol and closely related to the absorption coefficient α through the expression,

$$\alpha = \frac{4\pi\kappa}{\lambda} \qquad \lambda \rightarrow corresponding\ vacuum\ wavelength \tag{35}$$

Let's consider a light blocking structure of a particular geometry and an incident beam of light, as in [18]. This structure casts a shadow that characterizes absorption. The size of the absorption shadow is defined as the effective cross-section ($\sigma_\alpha\ [cm^2]$) and it can be smaller or larger than the geometrical size of the light blocking structure, depending on its absorption efficiency ($Q_\alpha\ dimensionless$), as described by the equation:

$$\sigma_\alpha = Q_\alpha A \qquad A \rightarrow structure\ geometrical\ size\ [cm^2]$$

The absorption coefficient ($\alpha\ [m^{-1}]$) describes a medium containing many light blocking structures at a concentration described as a volume density ($\rho_\alpha\ [cm^3]$). The absorption coefficient is essentially the cross-sectional area per unit volume of medium.

$$\alpha\ [cm^{-1}] = \rho_\alpha\ [cm^3]\sigma_\alpha\ [cm^2]$$

Absorption is the loss in the power of an electromagnetic radiation that is traveling in a medium. The loss is due to the conversion of light energy to other forms of energy, e.g. lattice vibrations

(heat) during the polarization of the molecules of the medium, local vibrations of impurity ions, excitation of electrons from the valence band to the conduction band [19].

Usually, loss is expressed in units of cm^{-1} that reflects the amount of attenuation/gain experienced by an EM field when it propagates over $1\ cm$ of dielectric material. Sometimes, loss comes in units of $dBcm^{-1}$ or even $dB\mu m^{-1}$. Hence, a conversion rule must be established. Conversion is as follows:

- Being next expression the equation modelling the exponential decay of monochromatic light as it travels through a dielectric material,

$S = S_0 e^{-\alpha L}$ $S \rightarrow$ signal intensity after propagating through a medium of length L and loss rate α. $S_0 \rightarrow$ initial signal intensity.

$$loss_{[dB]} = 10 \log_{10}(e^{\alpha L})$$

$$loss_{[dB]} = 10 \log_{10}(e) \alpha L$$

$$\boxed{loss_{[dB]} = 4.3429\, \alpha L}$$

If loss rate is in cm^{-1} and the needed conversion is for $dBcm^{-1}$ then,

$$\boxed{loss_{[dBcm^{-1}]} = 4.3429\, \alpha \times 10^{-2}}$$

One has to convert identically if $dB\mu m^{-1}$ are the desired units and loss rate is in μm^{-1}...

3.2. Complex Effective Refractive Index

The refractive index of a transparent optical medium, a.k.a. index of refraction, is the factor by which the phase velocity v_{ph} is decreased/increased relative to the speed of light in vacuum, as defined by the expression:

$$v_{ph} = \frac{c}{n}$$

It is always referring to plane waves with low optical intensities (linear propagation) propagating with no transverse restrictions and can be calculated through the equation:

$$n^2 = \varepsilon_r \mu_r, because\ \mu_r = 1\ in\ most\ cases \rightarrow n^2 = \varepsilon_r$$

$$\boxed{n = \sqrt{\varepsilon_r}}$$

When an EM field propagates through a dielectric waveguide, its geometry poses some sort of transverse constraint and one or more modes are created. Each mode has a unique phase constant β associated, as referred in 2.2.2., that explains its phase progression along the waveguide and relates to the vacuum wavenumber through the equation:

$$\beta = n_{eff}\frac{2\pi}{\lambda_0}$$

Here, n_{eff} is the effective refractive index and its magnitude is nearer the refractive index of the core or the cladding, depending whether the mode travels closer to the centre or to the periphery of the waveguide. The meaning of the effective refractive index is much the same as the refractive index, a number quantifying the phase delay per unit length relative to phase delay in vacuum, only this time in a waveguide and associated to a particular mode.

The effective refractive index is a complex value and its imaginary part describes either gain or loss, so in the simulation tool Optiwave® OptiFDTD, the variable that represents the complex effective refractive index is designated modal index (as sometimes in literature) and the convention used to deal with complex numbers is the usual:

$$z = a + jb$$

meaning that, if one has to represent a modal index with loss, the following must be taken into consideration,

$$modal\ index = n_{eff} + j(-\kappa_{eff})$$

with n_{eff} being the effective refractive index and κ_{eff} the effective extinction coefficient that relates to the already mentioned absorption coefficient α through equation 35.

3.3. Photonic wire simulation

This sub-chapter is dedicated to a series of simulations on Optiwave® OptiFDTD in order to evaluate power decay over length and as arc radius decreases. These simulations will help to characterize power losses by path unit, a power budget constraint for the optical systems designer. For that matter and as far as linear waveguides are concerned, these simulations will verify how EM field intensity is affected over length and on different wavelengths, for EM field propagation in these structures is highly dependent of frequency. As for the arc waveguides, besides the afore mentioned constraints, there is one other power decay contribution which is loss by radiation. In order to characterize it, one has to distinguish losses due to radiation from total losses by normalizing the latter with loss by path unit or, as it is referred in this dissertation, intrinsic losses.

This simulation tool has the possibility of either one of two options when it comes to control the simulation environment programmatically:

- Variable sweep.
- Visual Basic script (VB script).

For the former, a range of values is defined and the simulation is iterated for each and every value within the variable range. On the latter, almost all of the elements in the workspace are defined programmatically in VB script, allowing their control and configuration according to programmer's intents.

For the purpose of this dissertation, two structures with a-Si:H as the core waveguide were created both in 2D and 3D, allowing the respective simulations and obtained results:

1. photonic wire $15\ \mu m$ long to evaluate power loss on a linear waveguide.
2. photonic wire $14\ \mu m$ long followed by a decreasing radius arc waveguide and another photonic wire of variable length so it compensates the decrease on arc radius, keeping the dimensions of the simulation window unaltered.

As far as 2D simulations are concerned two main approaches were followed. The first one analysed power decay on a photonic wire with frequency. Two wavelengths from the commercial communications range were selected ($880\ nm$ and $1500\ nm$) and data collected. For the $880\ nm$ the results showed that propagation is not possible under these conditions for the EM field vanished completely after few μms. On the other hand, the results for the $1500\ nm$ were satisfactory for propagation with low attenuation of the EM field intensity ($2.6\ dBcm^{-1}$) was possible.

The second main approach evaluated power decay on an arc waveguide as its radius decreases. The results provided by simulations showed that power loss is mainly due to radiation losses rather than to a-Si:H intrinsic losses, however decay was kept within acceptable limits as far as the lowest simulated radius ($3\ \mu m$) was considered.

3D simulations on OptiFDTD 32bit version are highly time, memory and CPU consuming. These simulations easily use up all addressable memory space available in a 32bit system and are not able to benefit from a multiprocessor system. Each of the following 3D photonic wire simulations run for several days on a dedicated machine. Nevertheless, for example, when a 3D structure such as an asymmetric photonic wire is under analysis or one wishes to evaluate a symmetric 3D structure where the refractive index is not linearly distributed (i.e. defects), 3D simulation on these structures has to be performed for accurate results.

3D simulations on this dissertation followed a similar approach to the one conducted on 2D. As far as the photonic wire is concerned two strategies were followed; symmetric (a-Si:H core embedded on surrounding SiO_2) and asymmetric (a-Si:H core embedded on SiO_2 substrate and air cladding) devices were created in order to evaluate power decay as core waveguide width increases ($0.5 \rightarrow 2.5\ \mu m$). Both simulation results were similar although the latter showed slightly better field confinement at the $12\ \mu m$ measuring position, probably explained by the higher contrast of the air/a-Si:H interface. The two set of results showed a decreasing field intensity at both sensor marks, as core width varies, from initial value until it reaches the working wavelength value, levelling out from then onwards. This might be explained by a higher number of reflections when the waveguide's width is lower than the working wavelength, thus increasing the probability of constructive interference, hence a higher intensity EM field. As for the arc waveguide device and because the iterated simulation window is much larger, the time/memory/CPU requirements were very intensive and, for that reason, it was not possible to obtain all intended results. Nevertheless, some results have been collected and their analysis show an identical behaviour as in the 2D device simulations but, this time, with better accuracy.

So, next follows the first 2D simulation. This time, a photonic wire is going to be simulated on Optiwave® OptiFDTD. The a-Si:H core waveguide is $236\ nm$ wide and $15\ \mu m$ long. It is embedded

on a SiO_2 substrate $1\ \mu m$ wide and of the same length. In this simulation it has been included an input field of $1\ V/m$ amplitude (symbolized by the red vertical line with an arrow pointing rightwards – positive direction) that is responsible to insert the TE fundamental mode into the waveguide and two observation lines (symbolized by the vertical green lines), as shown on Figure 36. The inset shows the real part of refractive index of both waveguide's core and substrate ($Re\{n_x\}$, $with\ x \to c\ (core)\ or\ x \to s\ (substrate)$).

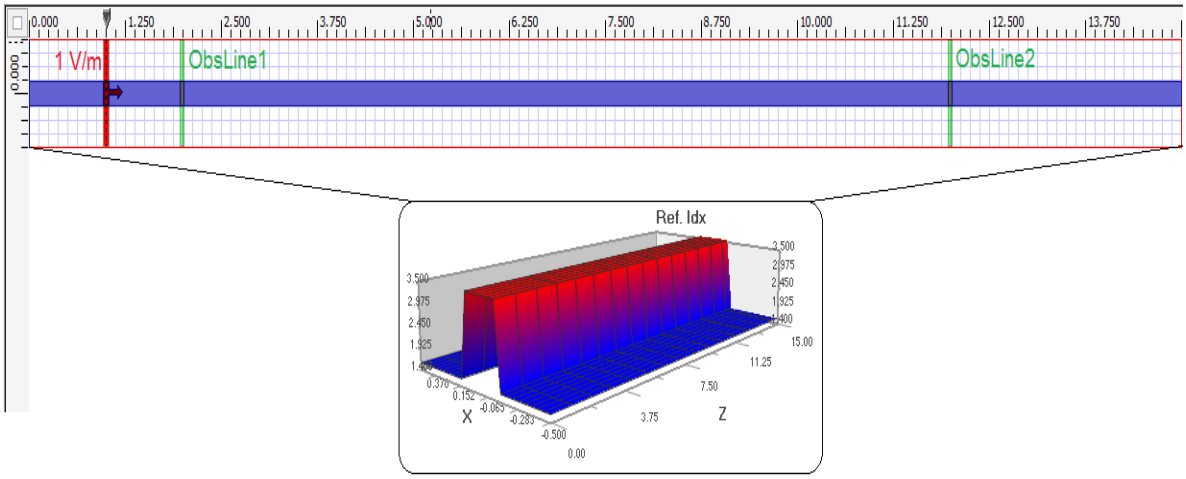

Figure 36 - Designing simulation workspace on OptiFDTD.

Simulation's parameters definition

Table 4 - Waveguide profile and materials for 2D simulation @ 1500 nm.

Profile and materials		
Dielectric		**Channel**
a-Si:H@1.5um	SiO_2	
Isotropic	Isotropic	Name: wg
Refractive index: $Re = 3.4858$ [20] $Im = -6.8713 \times 10^{-6}$ $(*)$	Refractive index: $Re = 1.4446$ [20]	2D profile definition: a-Si:H@1.5um

Table 5 - Input field parameters for 2D simulation @ 1500 nm.

Input field	
GMCW	**Found mode**
Wavelength = $1.5\ \mu m$	Modal index
TE modal	$Re = 2.93011337$
Amplitude = $1\ V/m$	
Z position = $1\ \mu m$	$Im = -6.89308739 \times 10^{-6}$
Default time offset and half width	

Table 6 - Observation lines for 2D simulation @ 1500 nm.

Observation lines	
Observation line 1	**Observation line 2**
Centre	Centre
Horizontal → $2\ \mu m$	Horizontal → $12\ \mu m$
Vertical → $0\ \mu m$	Vertical → $0\ \mu m$
Length → $1\ \mu m$	Length → $1\ \mu m$

Table 7 - Linear waveguide parameters for 2D simulation @ 1500 nm.

Linear waveguide 1		
Start	**End**	Width → $0.236\ \mu m$
Offset	Offset	
Horizontal → $0\ \mu m$	Horizontal → $14\ \mu m$	Profile → wg
Vertical → $2\ \mu m$	Vertical → $2\ \mu m$	

$(*)$ the value $Im = -6.8713 \times 10^{-6}$ represents the extinction coefficient κ and was calculated through equation 35 and considering $\alpha = 2.5\ dBcm^{-1}$ as in [1].

2D TE 32bit simulation was undertaken with Optiwave® OptiFDTD default settings, except the number of Time Steps parameter that had to be increased to 2000 in order to conclude the process in its whole length, and the results are shown in Figure 37. Next, the propagated power and respective attenuation along the length of the photonic wire was determined.

In order to calculate the attenuation suffered by the intensity of the TE fundamental mode, one has to obtain two comparing measurements. Thus, two observation lines were placed $10\ \mu m$ away from each other, at $2\ \mu m$ and at $12\ \mu m$, and each of them collects the EM field intensity going through (see Figure 36).

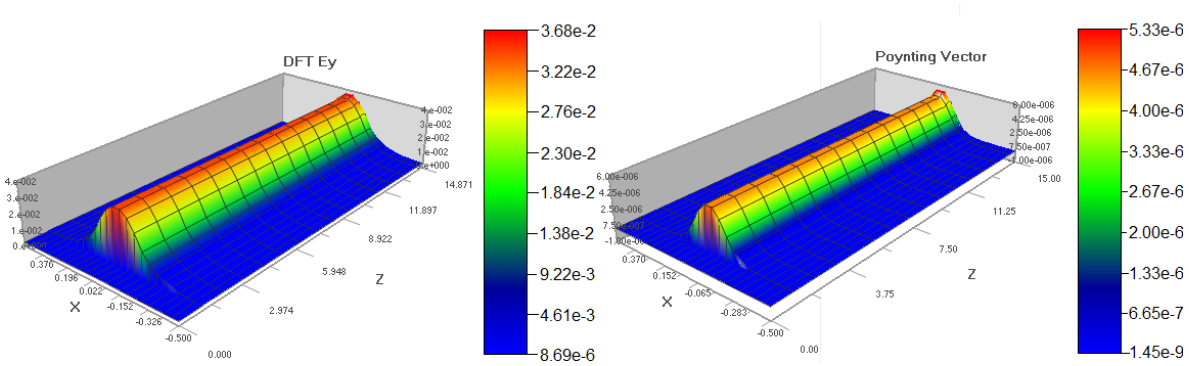

Figure 37 - DFT of E_y field and Poynting vector (λ = 1500 nm).

From the input parameter used, it can be expected that this device would suffer a $2.5\ dBcm^{-1}$ intensity decay. The distance separating the observation lines ($10\ \mu m$) introduces a power decay of $2.5\ dB \times 10^{-3} = 0.0025\ dB$. On Figure 38, although the graphic refers to E_y field's amplitudes,

it is also shown power intensities (Wm^{-1}) for both observation lines 1 and 2. Hence, it is possible to calculate the decay undergone by field power over 10 μm length.

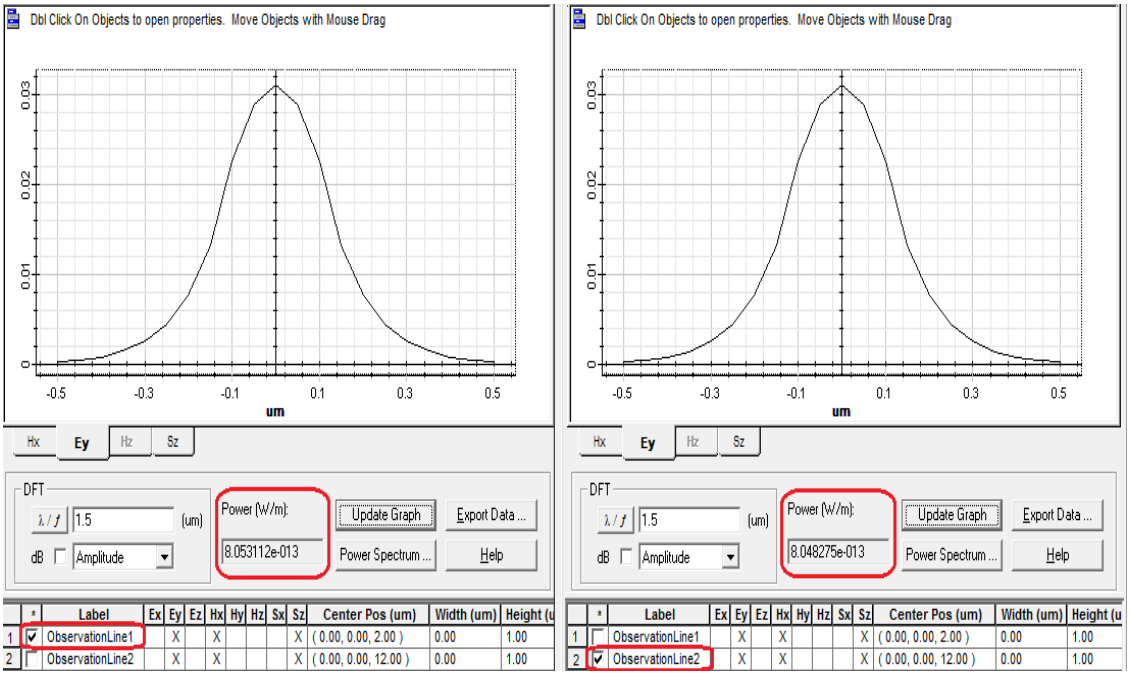

Figure 38 - Observation lines intensity comparison.

$$loss_{[dB10\mu m^{-1}]} = 10\log_{10}\left(\frac{8.048275 \times 10^{-13}}{8.053112 \times 10^{-13}}\right)$$

$$loss_{[dB10\mu m^{-1}]} = 0.0026\, dB10\mu m^{-1}$$

The calculated value is slightly higher than expected but it is due to the difference between the inserted ($Im = -6.8713 \times 10^{-6}$) and the obtained by the simulation tool ($Im = -6.89308739 \times 10^{-6}$) imaginary parts of the complex refractive index (extinction coefficient κ), which is higher in the latter, hence a higher loss. The reason for this is that OptiFDTD needs to find the mode in order to proceed with simulation and the higher value is this mode's refractive effective index (or modal index).

Also, in order to verify wavelength dependency of such devices, a second simulation was carried out. This time, with a different working wavelength ($\lambda = 880\ nm$) and its respective modal index. In this case, the imaginary part of the modal index assumes the value -0.044001, a higher value than before, and the real part of the same parameter is also affected, as shown on Table 8. All the other parameters remained the same as can be observed on the following tables.

Simulation's parameters definition

Table 8 - Waveguide profile and materials for 2D simulation @ 880 nm.

Profile and materials		
Dielectric		**Channel**
a-Si:H@0.88um	SiO_2	
Isotropic	Isotropic	Name: wg
Refractive index: $Re = 3.7785$ [20] $Im = -0.044001$ [20]	Refractive index: $Re = 1.4520$ [20]	2D profile definition: a-Si:H@1.5um

Table 9 - Input field parameters for 2D simulation @ 880 nm.

Input field	
GMCW	**Found mode**
Wavelength = $0.88\ \mu m$	Modal index
TE modal	$Re = 3.76251367$
Amplitude = $1\ V/m$	
Z position = $1\ \mu m$	$Im = -0.0432131774$
Default time offset and half width	

Table 10 - Observation lines for 2D simulation @ 880 nm.

Observation lines	
Observation line 1	**Observation line 2**
Centre	Centre
Horizontal → $2\ \mu m$	Horizontal → $12\ \mu m$
Vertical → $0\ \mu m$	Vertical → $0\ \mu m$
Length → $1\ \mu m$	Length → $1\ \mu m$

Table 11 - Linear waveguide parameters for 2D simulation @ 880 nm.

Linear waveguide 1		
Start	**End**	Width → $0.236\ \mu m$
Offset	Offset	
Horizontal → $0\ \mu m$	Horizontal → $14\ \mu m$	Profile → wg
Vertical → $2\ \mu m$	Vertical → $2\ \mu m$	

The same procedure as previous simulation was followed and the results are shown on Figure 39. As can be observed, at this wavelength the higher extinction coefficient κ results in much higher attenuation and after some μms the mode vanishes completely, confirming that propagation on dielectric waveguides is wavelength/frequency highly dependent.

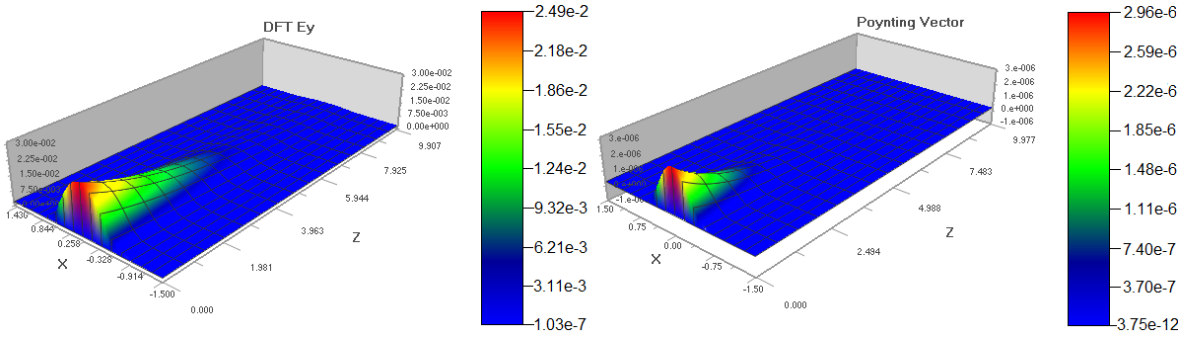

Figure 39 - DFT of Ey field and Poynting vector (λ = 880 nm).

3.4. Photonic wire loss through radiation as arc radius decreases

When developing any device, several design constraints arise quite easily. One of utmost importance is the power budget. Newly developed device characteristics must meet the given specifications. Designing constraints often lead to changes of direction in order to get the wave field from one point to the next one. There are some available techniques to accomplish this and an analysis of power loss over a range of different curve's radius, keeping all other parameters constant, has been considered quite pertinent. Thus, this subchapter is focusing on this subject although many other analysis possibilities may be within reach for Optiwave® OptiFDTD. Through the FDTD numerical method structures like ring resonators and their wavelength filtering capabilities [21], directional couplers and their characteristics [21] and photonic crystals with their many applications [22], to mention some, are possible to simulate, collect and evaluate relevant data.

The chosen application was again Optiwave® OptiFDTD for its good power measuring capabilities, while working with one of the most accurate numerical analysis method available as far as EM waves are involved – FDTD (Finite-difference time-domain) numerical method.

The developed device has been depicted on Figure 40 and the inset shows respective refraction indexes for the core waveguide (a-Si:H) and the wafer substrate (SiO_2). It is also shown in this figure an input field and three observation lines.

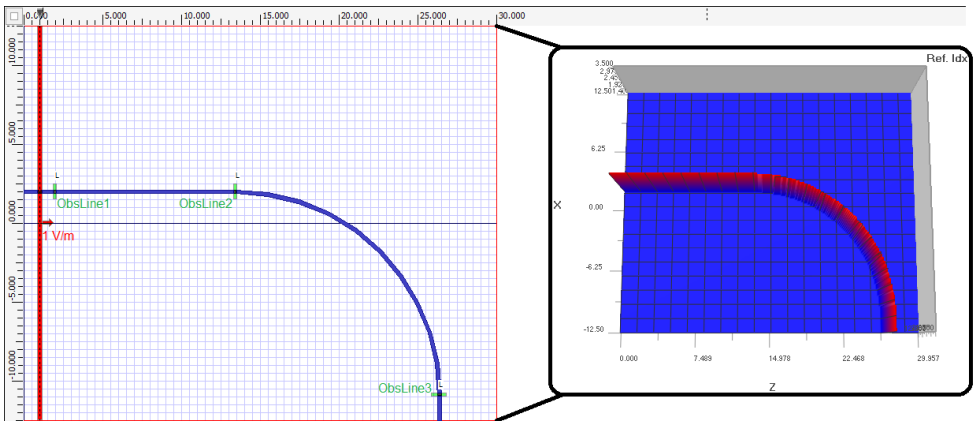

Figure 40 - OptiDesigner workspace; inset: - 2D refraction index.

Simulation's parameters definition

Table 12 - Profile and materials parameters for 2D simulation.

Profile and materials		
Dielectric		**Channel**
a-Si:H@1.5um	SiO_2	
Isotropic	Isotropic	wg
Refractive index: $Re = 3.4858$ $Im = -6.8713 \times 10^{-6}$	Refractive index: $Re = 1.4446$	2D profile definition: a-Si:H@1.5um

Table 13 - Input field parameters for 2D simulation.

Input field	
GMCW	**Found mode**
Wavelength = $1.5\ \mu m$	Modal index
TE modal	$Re = 2.93011337$
Amplitude = $1\ V/m$	
Z position = $1\ \mu m$	$Im = -6.89308739 \times 10^{-6}$
Default time offset and half width	

Table 14 - Observation lines parameters for 2D simulation.

Observation lines		
Observation line 1	**Observation line 2**	**Observation line 3**
Centre	Centre	Centre
Horizontal → $2\ \mu m$	Horizontal → $12\ \mu m$	Horizontal → $14 + radius\ [\mu m]$
Vertical → $0\ \mu m$	Vertical → $0\ \mu m$	Vertical → $1 - radius\ [\mu m]$
Length → $1\ \mu m$	Length → $1\ \mu m$	Length → $1\ \mu m$

Table 15 - Linear waveguide 1 parameters for 2D simulation.

Linear waveguide 1		
Start	**End**	Width → $0.236\ \mu m$
Offset	Offset	
Horizontal → $0\ \mu m$	Horizontal → $14\ \mu m$	Profile → wg
Vertical → $2\ \mu m$	Vertical → $2\ \mu m$	

Table 16 - Arc waveguide parameters for 2D simulation.

Arc waveguide			
Start	**End**	Waveguide radius: $radius\ [\mu m]$	Width → $0.236\ \mu m$
Offset	Offset		
Horizontal: $14\ \mu m$	Horizontal: $14 + radius\ [\mu m]$	Fixed radius	
Vertical: $2\ \mu m$	Vertical: $2\ - radius\ [\mu m]$	Positive radius	Profile → wg

Table 17 - Linear waveguide 2 parameters for 2D simulation.

Linear waveguide 2		
Start	**End**	Width → $0.236\ \mu m$
Offset	Offset	
Horizontal → $14 + radius\ [\mu m]$	Horizontal → $14 + radius\ [\mu m]$	Profile → wg
Vertical → $2 - radius\ [\mu m]$	Vertical → $-radius\ [\mu m]$	

Table 18 - Simulation wafer parameters.

Wafer properties	
Wafer dimensions	**2D wafer properties**
Length → $14.5 + radius\ [\mu m]$	Wafer refractive index → SiO_2 (Silica)
Width → r $[\mu m]$	

In order to develop a simulation to test all the intended arc's radii, there had to be created two auxiliary variables: - $radius\ and\ r$. The expression that defines their relationship is:

$$r = 2 \times radius$$

and the range swept by the variable $radius$ is:

$$radius = 5 : 3\ step\ 0.5 \quad (**)$$

$(**)$ for some unknown reason, this simulator requires arc's radius larger than $2.5015\ldots\ \mu m$.

The main constraint of the FDTD method is the step size for both time and space. These are intimately related to the stability and accuracy of the method. When defining the step size, a rule of thumb is usually considered for the mesh size: - minimum 10 cells per shortest wavelength to simulate [23].

Also, it is not rare to end a simulation and, when the results are being checked, some discrepancies are observed. As such, there are mainly three parameters to be verified when one faces such a situation:

- Mesh parameters; mesh size must be small enough to guarantee FDTD algorithm's accuracy (sometimes even smaller than one tenth ($^1/_{10}$) of the minimum simulated wavelength…).
- Boundary conditions; the choice of the appropriate parameter - APML (Anisotropic Perfectly Matched Layers), PEC (Perfect Electrical Conductor), PMC (Perfect Magnetic Conductor) or PBC (Periodic Boundary Condition) – is of key importance to get meaningful results.
- Time parameters; Time steps value is sometimes not enough to conclude simulation thus, leading to useless results.

The simulation to run the different arc waveguide radii and, at the same time, optimizing the dimensions and positioning of both linear waveguide 2 and observation lines 2 and 3, has been conducted using OptiFDTD variable sweep capability. The observed results are shown on Table 19:

Table 19 - Power loss induced by arc radius decrease.

$\boldsymbol{radius}$ $[\boldsymbol{\mu m}]$	$\boldsymbol{ObservLine1}$ $[\boldsymbol{Wm^{-1}}]$	$\boldsymbol{ObservLine2}$ $[\boldsymbol{Wm^{-1}}]$	$\boldsymbol{ObservLine3}$ $[\boldsymbol{Wm^{-1}}]$	$\boldsymbol{Total_{loss}}$ $[\boldsymbol{dB}]$	$\boldsymbol{Radiation_{loss}}$ $[\boldsymbol{dB}]$
5.0	4.939656×10^{-13}	4.930868×10^{-13}	$-4.830275 \times 10^{-13}$	$8{,}951520 \times 10^{-2}$	$8{,}951519 \times 10^{-2}$
4.5	4.940157×10^{-13}	4.933620×10^{-13}	$-4.827760 \times 10^{-13}$	$9{,}420024 \times 10^{-2}$	$9{,}420024 \times 10^{-2}$
4.0	4.941166×10^{-13}	4.929527×10^{-13}	$-4.836440 \times 10^{-13}$	$8{,}279445 \times 10^{-2}$	$8{,}279445 \times 10^{-2}$
3.5	4.935443×10^{-13}	4.930622×10^{-13}	$-4.833453 \times 10^{-13}$	$8{,}644209 \times 10^{-2}$	$8{,}644209 \times 10^{-2}$
3.0	4.939589×10^{-13}	4.936150×10^{-13}	$-4.858121 \times 10^{-13}$	$6{,}920021 \times 10^{-2}$	$6{,}920021 \times 10^{-2}$

The sign on ObservationLine3 values is due to the direction change from previous observation line (vertical line) to this one (horizontal line). Nevertheless, loss calculations have taken into account the absolute value of ObservationLine3 and the observed decay in the arc waveguide, as follows:

$$Total_{loss}[dB] = 10\log_{10}\left[\frac{abs(ObservLine3)}{ObservLine2}\right] - \left(loss_{2\,\mu m} + loss_{1\,\mu m}\right)$$

Also, one must keep in mind that, on Table 19, column $Radiation_{loss}\ [dB]$ refers to the amount of power decay observed over a distance $\pi/2 \times radius\ [\mu m]$ and that it includes both waveguide intrinsic loss ($\sim 2.6\ dBcm^{-1}$) and radiation loss due to bend radius. Hence, the need for another column that reflects the amount of loss due to radiation and that is given by the expression:

$$Radiation_{loss}[dB] = Total_{loss}[dB] - loss_{\pi/2\times radius[\mu m]}[dB] \qquad (36)$$

As observed, most of the losses observed on Table 19 are due to radiation. One would expect an increase of losses as the radius decreases but, as far as this sampling is concerned, it is not the case. Even trying to normalise radiation loss by path unit (radiation loss/radius), the tendency shown by data is not clear:

Table 20 - Normalisation of radiation per path unit.

$\boldsymbol{radius}$ $[\boldsymbol{\mu m}]$	$\boldsymbol{Radiation_{loss}}$ $[\boldsymbol{dB}]$	$\boldsymbol{Normalisation}$
5.0	$8{,}951519 \times 10^{-2}$	$1{,}790304 \times 10^{-2}$
4.5	$9{,}420024 \times 10^{-2}$	$2{,}093339 \times 10^{-2}$
4.0	$8{,}279445 \times 10^{-2}$	$2{,}069861 \times 10^{-2}$
3.5	$8{,}644209 \times 10^{-2}$	$2{,}469774 \times 10^{-2}$
3.0	$6{,}920021 \times 10^{-2}$	$2{,}306674 \times 10^{-2}$

In order to better understand how radiation loss varies with decreasing radius, another simulation has been conducted using OptiFDTD script simulation capabilities instead of, as previously, parameter sweep. This time, with a wider range of samples (100) and radii ($12.9\ \mu m \rightarrow 3\ \mu m$), finer step of decreasing radius ($100\ nm$) and different configuration parameters, such as:

- Wafer dimensions are kept constant as arc radius decreases (to maintain mesh size and time steps parameters the same all along each and every simulation); in order to achieve this, the vertical linear waveguide (LinearWaveguide2) increases its length accordingly.

- Input field is now a continuous wave with $1.5\ \mu m$ wavelength; in order to find out bend losses, this is a more appropriate wave profile (there is no need to analyse frequency response in this case).
- Repositioning of observation lines (Observation Lines 2 and 3) at the beginning and at the end of arc waveguide; this way is not needed anymore to subtract the losses introduced by the difference between observation lines positions and arc waveguide's start and end.

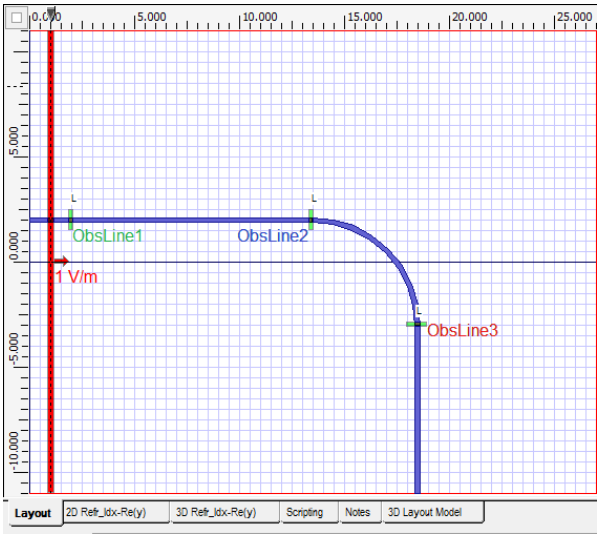

Figure 41 - Design conditions (radius = 5 um).

Figure 41 depicts design conditions at approximately half way simulation range with $radius = 5\ \mu m$. After simulation was concluded, data obtained was fed into Matlab (scripts powerFile.m and leastSquares.m were created; see Appendix G) for further processing. Figure 42 shows power ($[Wm^{-1}]$) obtained from the simulation script for the three observation lines.

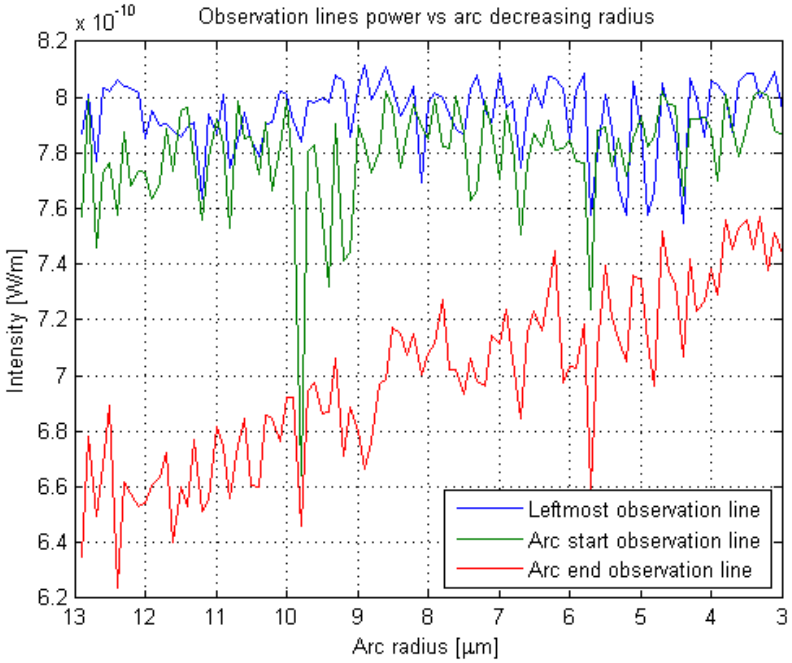

Figure 42 - Observed power over script simulation (100 samples).

Again, the absolute value of ObservationLine3 has been taken into account for the plot and one can notice that ObservationLine2 follows approximately ObservationLine1 (within some tolerance probably due to discretization), despite propagation loss decay. Next, an identical approach to the one previously followed with parameter sweep, has been conducted. Figure 43 shows power loss ($[dB]$) over arc's length and calculated accordingly to expression:

$$Total_{loss}[dB] = 10\log_{10}\left[\frac{abs(ObservLine3)}{ObservLine2}\right]$$

This time, there was no need to include losses on linear waveguides for observation lines 2 and 3 are kept always at the start and end of the arc waveguide. Finally, an attempt to normalise radiation loss by path unit has been conducted, following an identical procedure as before. Figure 44 shows the results obtained and, although slightly and despite some oscillations around the mean value, there is an increase of power loss with decreasing radius trend followed by data, asserting it with expected results.

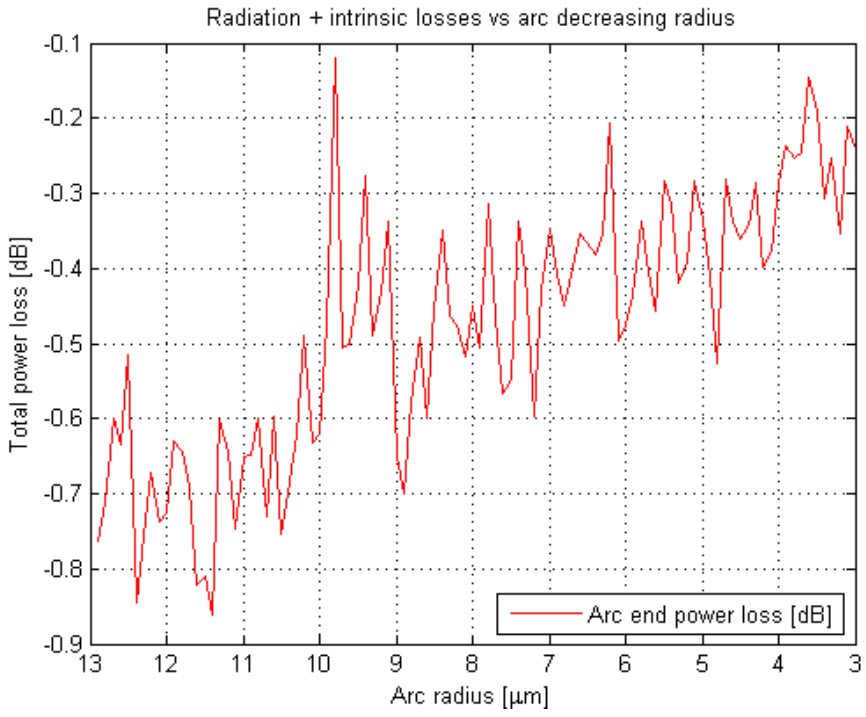

Figure 43 - Arc overall power loss in dB.

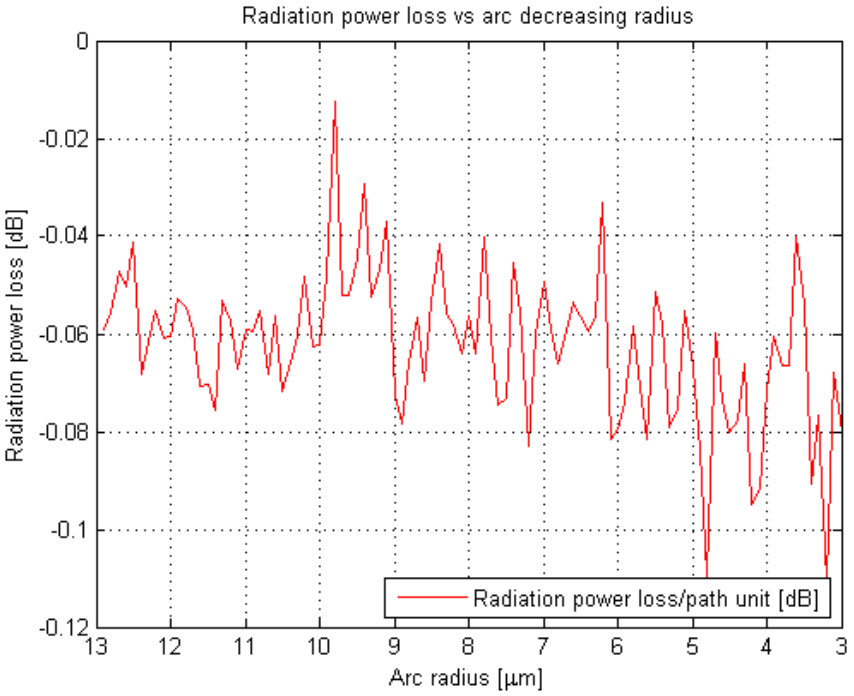

Figure 44 - ObservationLine3 power loss by path unit.

For a better understanding of power decay as arc radius increases, further analysis of data has been performed using the least squares algorithm, clearly showing on Figure 45 an increasing power decay trend due to radiation as arc waveguide radius decreases, initiating at around $9\ \mu m$ and becoming more evident after the $6\ \mu m$ mark, as it approaches operation wavelength value.

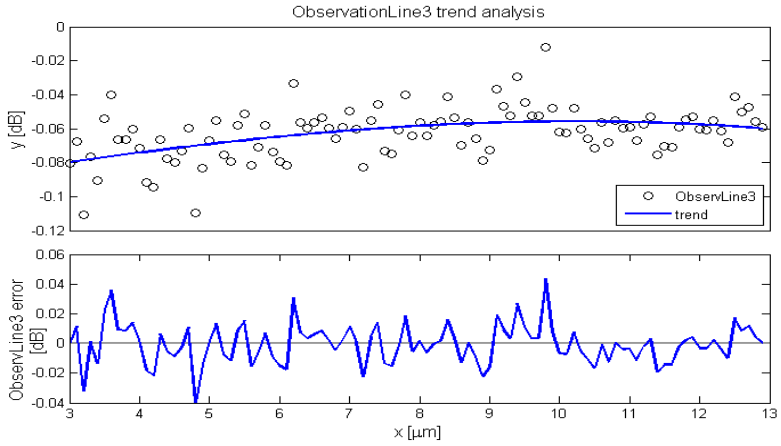

Figure 45 - ObservationLine3 trend (least squares algorithm).

As a final note, variability of plot data on Figure 42, Figure 43, Figure 44 and Figure 45 would probably be explained by the sample discretization introduced by the simulation script and the EM field spot size oscillation resulting from the interference of higher order modes created while propagating in the curved waveguide. This script iterates through arc waveguide radius range in steps of $100\ nm$ and because simulation window dimensions and input field are kept unaltered, the positioning of observation lines is being adjusted at each and every iteration, this might also contribute to the observed fluctuations.

3.5. 3D photonic wire simulations

The Overlap Integral is an important analysis tool, as far as optical field power is concerned, for it calculates power overlap of fields. In OptiFDTD, 3D power overlap integral is given by the following expression:

$$POI = \frac{\left|\int_{\Omega} \overrightarrow{E_1}(x,y) \cdot \overrightarrow{E_2^*}(x,y)\, dx\, dy\right|^2}{\int_{E_1}\left|\overrightarrow{E_1}(x,y)\right|^2 dx\, dy \int_{E_2}\left|\overrightarrow{E_2}(x,y)\right|^2 dx\, dy}$$

where E_1 and E_2 integration limits are the domain of respective functions and Ω is the user defined range of integration within the intersection of the two fields. Also, the square magnitudes are interpreted as:

$$\left|\vec{E}\right|^2 = \vec{E} \cdot \vec{E}^* = E_x E_x^* + E_y E_y^* = |E_x|^2 + \left|E_y\right|^2$$

Power is normalised so calculation of any field overlapped with itself will result in unity if the region of integration Ω is the whole function extent. Because a field overlapped with itself but integrated over a region smaller than the domain results in the square of the confinement factor, Overlap Integral can be used to obtain such ratio.

There are several available methods to calculate bending loss, being MAM (Modal Analysis Method) either with FDM (Finite-Difference Method) or FEM (Finite-Element Method) one of the most popular. In [24], both MAM options and 3D-FDTD Overlap Integral have been studied and results compared against experimental values. The latter method was the chosen one to calculate power decay over path, as far as bending loss is concerned, for it presented the best accuracy against experimental values.

When on 3D bent structures, using the Overlap Integral can provide the total bending loss of a structure such as an arc waveguide. In this case, the input field $\overrightarrow{E_{in}}(x,y)$ must be unit power normalised and the power fraction P which is coupled to the output waveguide is obtained through the Overlap Integral between the output optical field $\overrightarrow{E_{out}}(x,y)$ and the normalised fundamental modal field $\overrightarrow{E_0}(x,y)$ of the output waveguide:

$$P = \left|\iint \overrightarrow{E_{out}}(x,y) \cdot \overrightarrow{E_0^*}(x,y)\, dx\, dy\right|^2$$

Total loss is given, according to [24], through the expression:

$$Loss_{total} = -10 \log_{10}(P)$$

In this sub-chapter it is going to be reported the simulation results of two main 3D structures:

1. Straight photonic wire based on a-Si:H.
2. Bent photonic wire based on a-Si:H and formed in order by a straight, an arc and another straight waveguide.

The following set of simulations consider an a-Si:H photonic wire on two different devices. For the first device, the created structure is symmetric, meaning that the a-Si:H core is completely surrounded by SiO_2, as depicted on Figure 46:

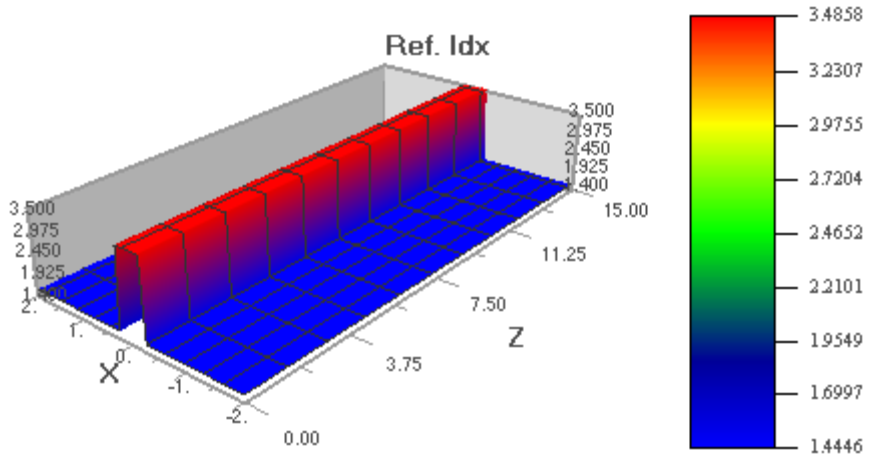

Figure 46 - Symmetric structure refractive indexes.

On the second device, an asymmetric structure was created, that is to say, an SiO_2 substrate supports the a-Si:H core that is covered by an air cladding, as shown in Figure 47:

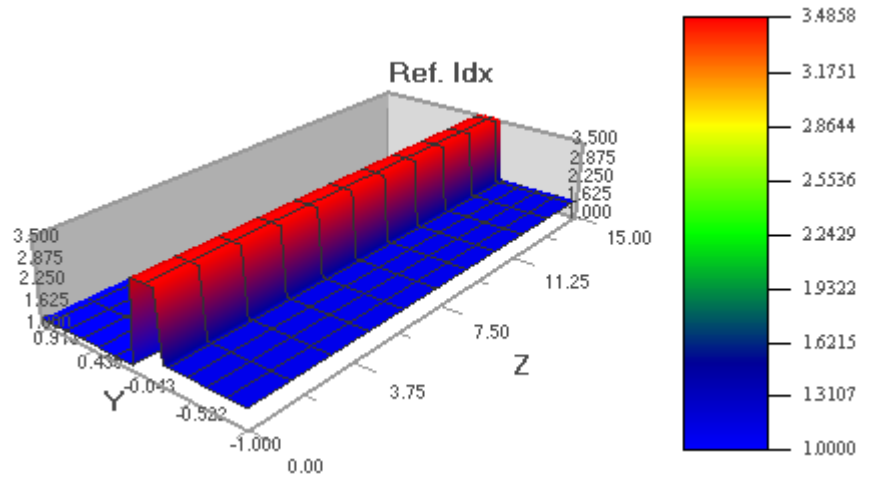

Figure 47 - Asymmetric structure refractive indexes.

The simulation conditions in both devices were the same as the ones undertaken in the first simulation of 3.3., with the exception of the 3D layer that had to be created on Optiwave® OptiFDTD software and, in order to comply with the width variation, a variable sweep had to be implemented.

The collected data is similar in both devices, showing a decreasing field intensity at both sensor positions as core width varies from initial $0.5\ \mu m$ until it reaches the working wavelength $1.5\ \mu m$, levelling out from then onwards up to the widest simulated device, as depicted on Figure 48 and Figure 49. The reason behind this behaviour might be the higher number of reflections when the

waveguide's width is lower than the working wavelength, thus increasing the probability of constructive interference, contributing to a higher intensity EM field. Also, the difference between ObservArea1 and ObservArea2 readings is ploted in both figures and shows an identical behaviour as the one verified on observation areas 1 and 2 but, on the asymmetric device the error data is more erratic although with lower amplitude.

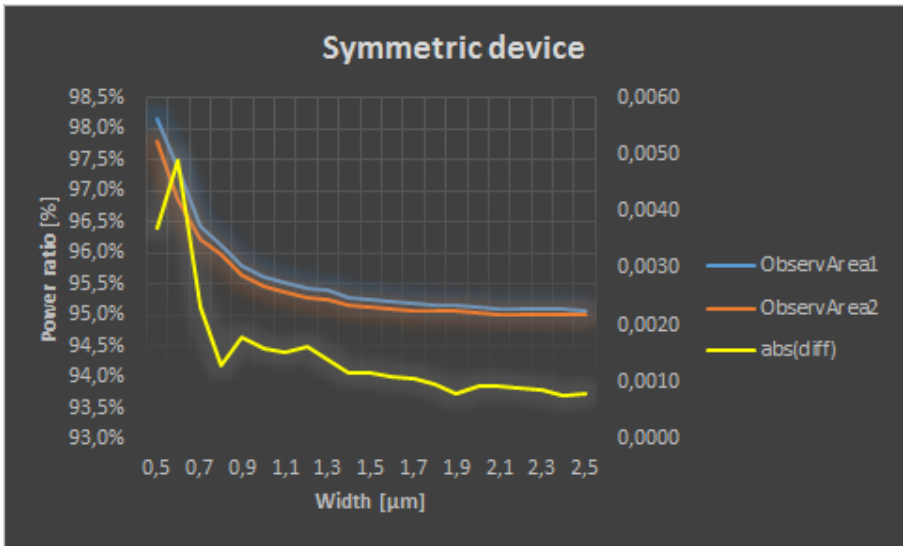

Figure 48 - Symmetric structure results.

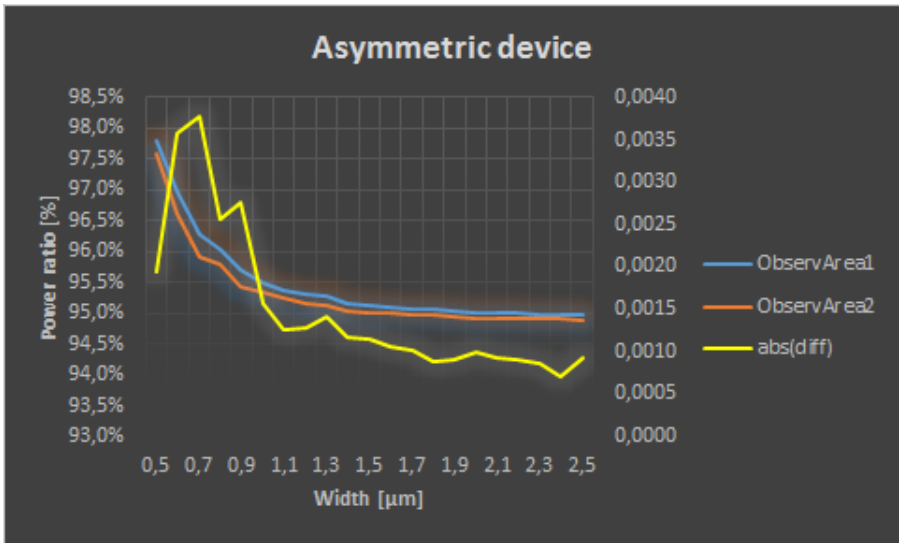

Figure 49 - Asymmetric structure results.

From the analysis of and using the Overlap Integral on both previous simulations, one can conclude that the asymmetric device showed a slightly better EM field overlap for, in the same longitudinal position for both devices, the Mode Overlap Integral calculation was -0.0025 dB as compared to -0.0031 dB for the symmetric case. This might be explained by the higher contrast associated to the air/a-Si:H core interface when compared to the one associated with the SiO_2/a-Si:H core boundary, allowing for a better confinement of the EM field along the waveguide length.

As far as the bent photonic wire a-Si:H based is concerned, the design workspace and simulation conditions are identical to the ones created on 3.4., with the exception of the 3D layers that had to be created on Optiwave® OptiFDTD software for both arc and linear waveguides and a shorter linear waveguide 1 (only $5\ \mu m$ long this time, in order to keep a small simulation window), as depicted on Figure 50. Also, in this layout and because it is a 3D workspace, the simulation window considers a $1\ \mu m$ SiO_2 substrate layer, on top of which the photonic wire is positioned, and a $1\ \mu m$ SiO_2 cladding layer, thus forming a symmetric device.

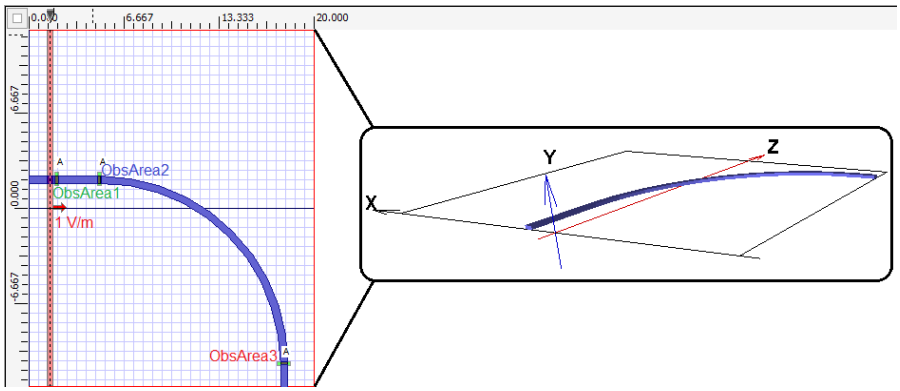

Figure 50 - OptiDesigner workspace (radius = 12.9 µm); inset: - 3D layout model.

The obtained simulation results, although few, show identical variability as the ones presented before on 3.4. last simulation (i.e. Figure 42) and by analysing the field spot positioning one can notice a variable deviation, when taking into account ObservationArea3's centres on Figure 51, Figure 52, Figure 53, Figure 54, Figure 55, Figure 56 and Figure 57, as $radius = 12.3 \rightarrow 12.9\ \mu m$. These observations seem to reinforce what has been said previously on the final paragraph of 3.4., that data variability would probably be explained by the sample discretization introduced by the simulation script and the EM field spot size oscillation resulting from the interference of higher order modes created while propagating from the straight to the curved waveguides.

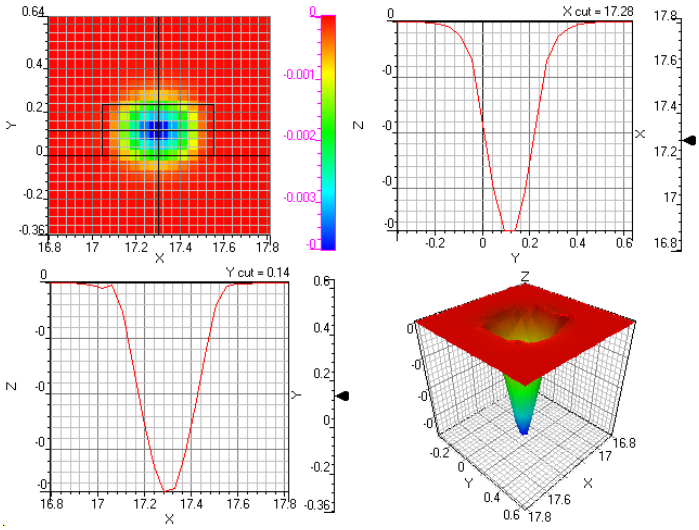

Figure 51 - Field spot position vs. ObservationArea3's centre (radius=12,3 µm).

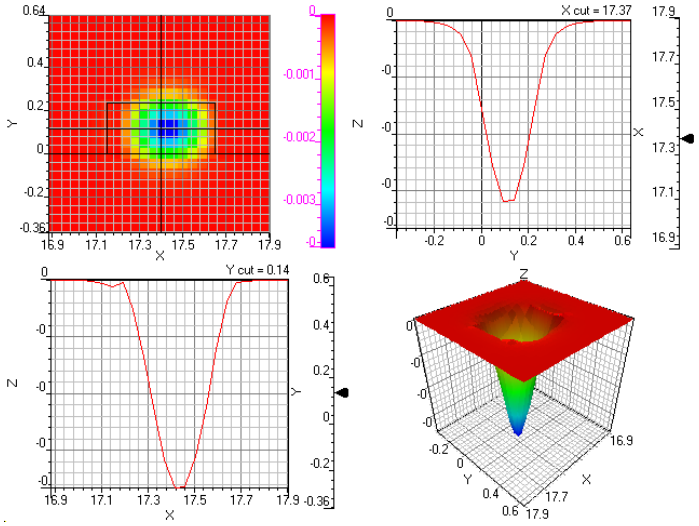

Figure 52 - Field spot position vs. ObservationArea3's centre (radius=12,4 µm).

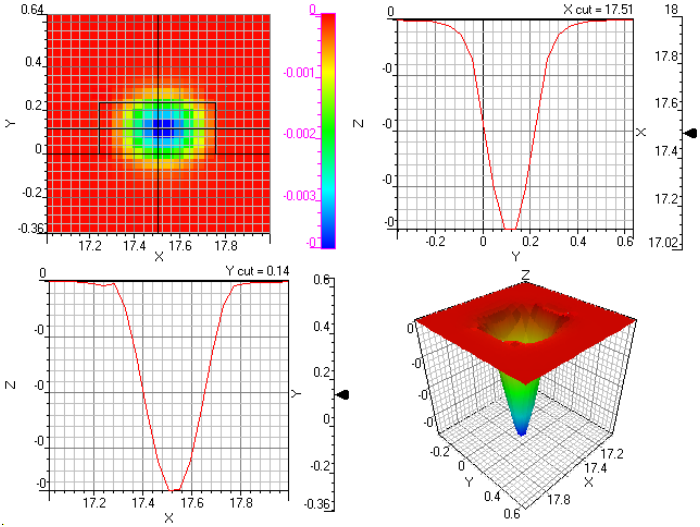

Figure 53 - Field spot position vs. ObservationArea3's centre (radius=12,5 µm).

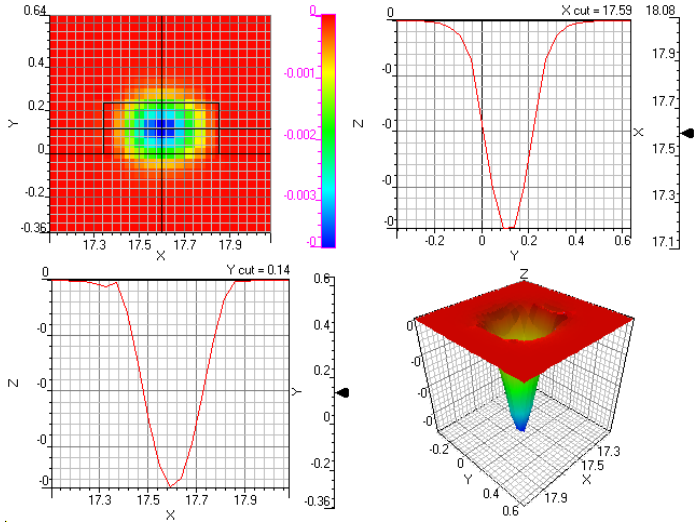

Figure 54 - Field spot position vs. ObservationArea3's centre (radius=12,6 µm).

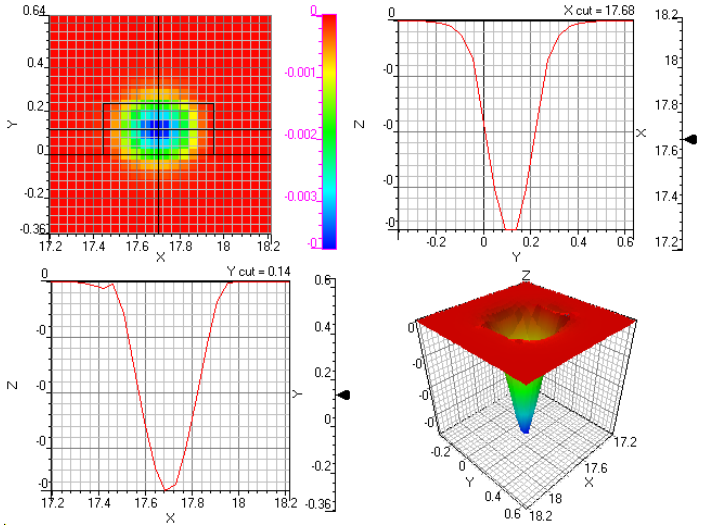

Figure 55 - Field spot position vs. ObservationArea3's centre (radius=12,7 µm).

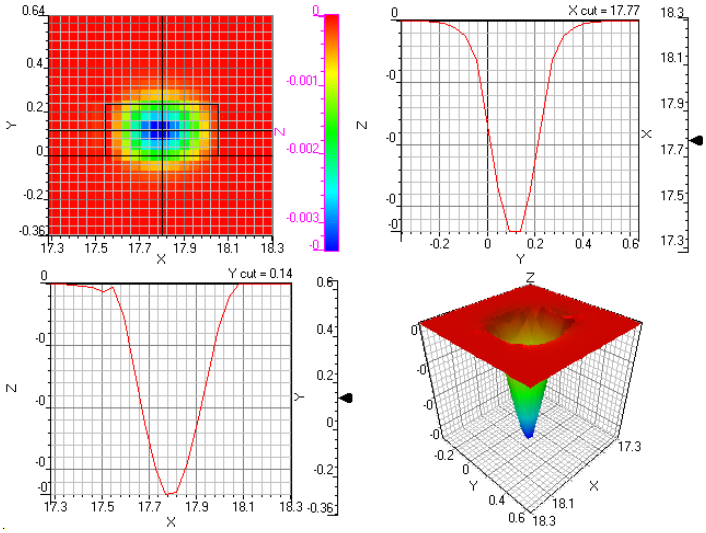

Figure 56 - Field spot position vs. ObservationArea3's centre (radius=12,8 µm).

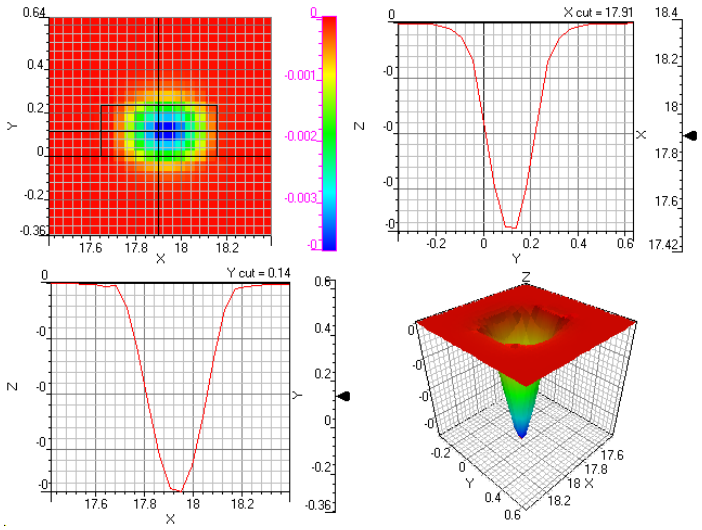

Figure 57 - Field spot position vs. ObservationArea3's centre (radius=12,9 µm).

To better characterize losses on a-Si:H 3D curved waveguides, it would require more data samples, which presents some difficulties for simulating each iteration of the created script is highly time/memory/CPU resources consuming on the 32bit software version.

Nevertheless, OptiFDTD software and its capabilities would produce accurate results when analysing power decay on devices such as photonic wires in most possible configurations. Through the selection of the adequate Overlap Integral option, one should be able to characterize either transition loss (associated to the photonic wire straight/bend interface reflections → higher order modes creation), pure radiation loss (associated to the imaginary part of the mode's propagation constant propagating in the curved section of the photonic wire device) or the total bending loss which is the sum of both previous losses.

Chapter IV

Conclusions and future work

Latest developments on a-Si:H waveguides fabrication techniques, namely PECVD, enable the fabrication of high quality, broadband and low loss devices where light propagates. This high performance material could be used as a platform to enable on-chip optical interconnects in multi-core processors and other applications, mainly due to its compatibility with CMOS fabrication processes, hence providing an alternative platform to existing electrical interconnects with less power constraints, broader bandwidth and less cross-talk between adjacent interconnects.

In this dissertation, after a theoretical approach to propagating modes on ideal planar-mirror dielectric waveguides followed by a detailed description of mode creation on each of the main categories of non-planar structures such as circular and rectangular waveguides, the latter devices have been simulated on Optiwave® OptiFDTD software and the results of power decay over length has been analysed.

Characterization of power decay over length and on curved waveguides will influence the dimensional limits establishment of photonic integrated circuits, hopefully enabling higher density of optical components into the same footprint, increasing functionality and reliability of sub-systems. For this purpose, 2D and 3D simulations were conducted on straight and arc waveguides in order to find out the estimated power loss over length versus frequency for the former devices and, for the latter, the expected power loss versus radius decrease. The results obtained show that attenuation highly increases as frequency moves from infra-red into the visible range of the spectrum and that radiation losses due to bends on arc waveguides are within acceptable limits, as far as the simulated devices and simulation conditions are concerned ($radius \geq 3\ \mu m$).

As suggestions to future work, many other configurations are to be considered and studied such as ring resonators and their unique filtering capabilities, linear coupling devices and photonic crystals which find applications in many fields. Further analysis of a-Si:H based 3D structures, namely the bent photonic wire on high contrast and sharp curvature radius ($< 3\ \mu m$) devices, is required through the use of 3D-FDTD tools for, although extremely resources demanding, their accuracy is of enhanced importance to fully characterize the involved phenomena.

References

[1] K. Narayanan, “Hydrogenated amorphous silicon photonics,” Nova Iorque, 2011.

[2] T. Baher-Jones, M. Hochberg, G. Wang, R. Lawson, Y. Liao, P. A. Sullivan, L. Dalton, A. K.-Y. Jen and A. Scherer, “Optical modulation and detection in slotted Silicon waveguides,” OSA - Optical Society of America, 2005.

[3] B. E. A. Saleh and M. Carl Teich, Fundamentals of Photonics, John Wiley & Sons, Inc., 1991.

[4] M. I. Barbosa de Carvalho, “GO dielectricos,” Porto, 2005.

[5] K. Nakajima, “Lect4-Optical waveguides,” University of Maryland, Baltimore, 2010.

[6] P. Pinho, A. Rocha and J. Rocha Pereira, Propagação Guiada de Ondas Eletromagnéticas, Grupo Editorial Nacional, 2014.

[7] P. Pinho, *Propagação e Radiação,* Lisboa, Lisboa: ISEL - Instituto Superior de Engenharia de Lisboa.

[8] K. Okamoto, Fundamentals of Optical Waveguides, vol. Second Edition, Elsevier Inc., 2006.

[9] G. Cocorullo, F. G. Della Corte, R. De Rosa, I. Rendina, A. Rubino and E. Terzini, “Amorphous silicon-based guided-wave passive and active devices for silicon integrated optoelectronics,” *IEEE Journal of selected topics in quantum electronics,* vol. 4(6), pp. 997-1002, 1998.

[10] A. Harke, M. Krause and J. Mueller, “Low-loss singlemode amorphous silicon waveguides,” *Electronics Letters,* vol. 41(25), p. 1, 2005.

[11] R. Sun, K. McComber, J. Cheng, D. K. Sparacin, M. Beals, J. Michel and L. C. Kimerling, “Transparent amorphous silicon channel waveguides with silicon nitride intercladding layer,” *Applied Physics Letters,* vol. 94 (14), 2009.

[12] F. G. Della Corte and S. Rao, “Use of amorphous silicon for active photonic devices,” *IEEE Transactions on Electron Devices,* vol. 60(5), pp. 1495-1505, 2013.

[13] R. Takei, “Amorphous Silicon Photonics,” P. Mandracci, Ed., InTech, 2016.

[14] R. E. Schropp, R. Carius and G. Beaucarne, “MRS bulletin,” Cambridge Core, 2007. [Online]. Available: https://www.cambridge.org/core/journals/mrs-bulletin/article/amorphous-silicon-microcrystalline-silicon-and-thin-film-polycrystalline-silicon-solar-cells/B3E8DE62A376FF07514B139CFF840155#. [Accessed September 2016].

[15] J. Jang, “Preparation and Properties of Hydrogenated Amorphous Silicon Thin-Film Transistors,” P. A. C.Kagan, Ed., New York, Marcel Dekker, Inc., 2003.

[16] A. Khanna, M. Mulot, S. Arpiainen, A. Säynätjoki, J. Ahopelto, S. Honkanen and H. Lipsanen, “Amorphous silicon optical waveguides and Bragg mirrors,” 2008.

[17] A. Sadao, Optical Constants of Crystalline and Amorphous Semiconductors, Springer, 1999.

[18] S. Steven L. Jacques and A. Prahl, “Definition and units of absorption coefficient,” ECE532 Biomedical Optics, 1998. [Online]. Available: http://omlc.org/classroom/ece532/class3/muadefinition.html. [Accessed 5 11 2016].

[19] S. O. Kasap, Optoelectronics & Photonics, Principles & Practices, International Edition, Pearson Higher Ed., 2013.

[20] refractiveindex.info database: public domain via CC0 1.0, “Refractive index database,” © 2008-2016 Mikhail Polyanskiy, 2008. [Online]. Available: http://refractiveindex.info/. [Accessed September 2016].

[21] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes and R. Baets, “Silicon microring resonators. Laser & Photonics Reviews, 6(1), 47-73.,” Wiley Online Library, 2012. [Online]. Available: http://photonics.intec.ugent.be/download/pub_3105.pdf. [Accessed September 2016].

[22] J. D. Joannopolous, S. G. Johnson, J. N. Winn and R. D. Meade, Photonic Crystals, Molding the Flow of Light, 2nd ed., Princeton: Princeton University Press, 2007.

[23] A. Taflove and S. C. Hagness, Computational Electrodynamics: the FDTD method, London: Artech House Boston, 2000.

[24] Z. Sheng, D. Dai and S. He, “Comparative Study of Losses in Ultrasharp Silicon-on-Insulator Nanowire Bends,” *IEEE JOURNAL OF SELECTED TOPICS IN QUANTUM ELECTRONICS,* vol. 15, pp. 1406-1412, 2009.

[25] P. C. Engineering, “Lecture_17_Optical_Fiber_Modes”.

Appendix A

Guided waves

Propagation of a harmonic electromagnetic wave of angular frequency ω in a linear medium with no sources ($\rho = 0 \; and \; \vec{J} = \vec{0}$) and parameters $\boldsymbol{\varepsilon}$ and $\boldsymbol{\mu}$, is regulated by Helmoltz's wave equations:

$$\begin{aligned} \nabla^2\vec{E} + \omega^2\mu\varepsilon\vec{E} = 0 \\ \nabla^2\vec{H} + \omega^2\mu\varepsilon\vec{H} = 0 \end{aligned} \tag{37}$$

where $\vec{E}$ and $\vec{H}$ are the wave's electric and magnetic fields phasors.

Let us consider the propagation of these waves in a cylindrical guide where its transversal section remains constant along *z-axis*, as is shown on Figure A58,

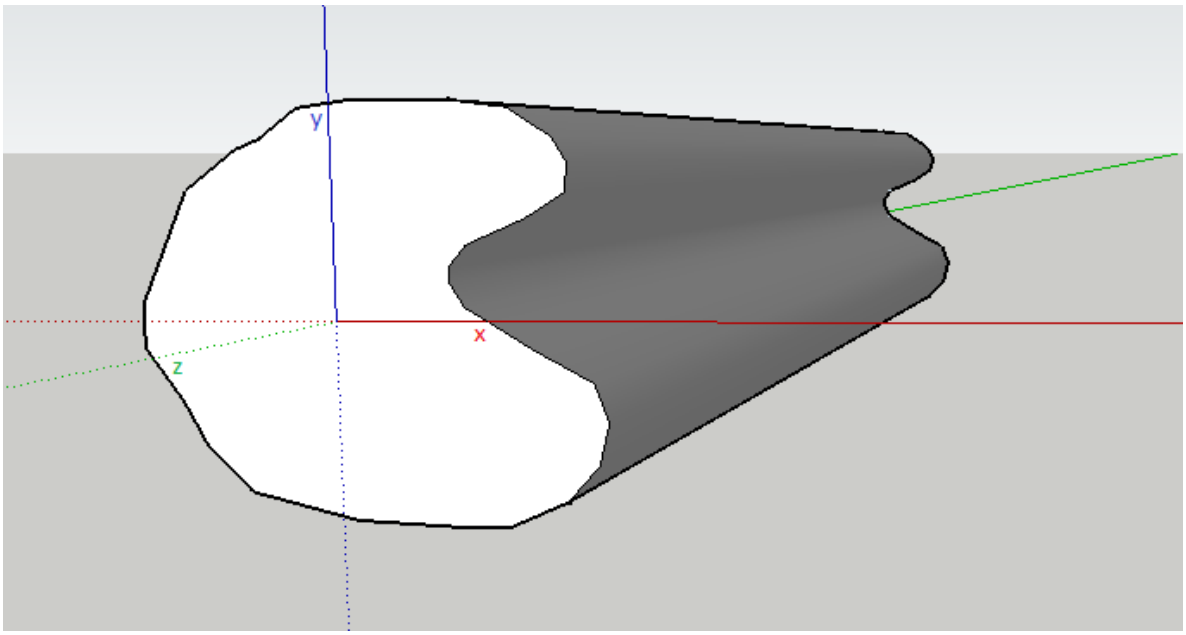

Figure A58 - Cylindrical waveguide.

Assuming this guide length is infinite, one only needs to take into account waves that propagate on +z direction, allowing the following fields equations to be written:

$$\begin{aligned} \vec{E}(x,y,z) = \overrightarrow{E^0}(x,y)e^{-\gamma z} \\ \vec{H}(x,y,z) = \overrightarrow{H^0}(x,y)e^{-\gamma z} \end{aligned}$$

where the propagation constant $\gamma = \alpha + j\beta$. Replacing this expression on equations 37, it will originate,

$$\begin{aligned} \nabla_{xy}^2\overrightarrow{E^0} + h^2\overrightarrow{E^0} = 0 \\ \nabla_{xy}^2\overrightarrow{H^0} + h^2\overrightarrow{H^0} = 0 \end{aligned}$$

where $\nabla_{xy}^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the transverse laplacian, $\overrightarrow{E^0} = E_x^0\hat{x} + E_y^0\hat{y} + E_z^0\hat{z}$ and $\overrightarrow{H^0} = H_x^0\hat{x} + H_y^0\hat{y} + H_z^0\hat{z}$ and $h^2 = \gamma^2 + \omega^2\mu\varepsilon$.

These two wave equations originate six scalar wave equations, one for each field component:

$$\frac{\partial^2 E_x^0}{\partial x^2} + \frac{\partial^2 E_x^0}{\partial y^2} + h^2 E_x^0 = 0 \quad \frac{\partial^2 H_x^0}{\partial x^2} + \frac{\partial^2 H_x^0}{\partial y^2} + h^2 H_x^0 = 0$$
$$\frac{\partial^2 E_y^0}{\partial x^2} + \frac{\partial^2 E_y^0}{\partial y^2} + h^2 E_y^0 = 0 \quad \frac{\partial^2 H_y^0}{\partial x^2} + \frac{\partial^2 H_y^0}{\partial y^2} + h^2 H_y^0 = 0$$
$$\frac{\partial^2 E_z^0}{\partial x^2} + \frac{\partial^2 E_z^0}{\partial y^2} + h^2 E_z^0 = 0 \quad \frac{\partial^2 H_z^0}{\partial x^2} + \frac{\partial^2 H_z^0}{\partial y^2} + h^2 H_z^0 = 0$$

These scalar wave equations are not independent from each other so, it is not necessary to solve all of them to obtain a complete description of $\vec{E}$ and $\vec{H}$. Actually, one needs only to find the longitudinal components because the transversal ones can be obtained from these.

Considering associated boundary conditions constraints for TM mode,

$H_z^0 = 0 \; and \; \frac{\partial}{\partial x} = 0.$

Wave equation is,

$$\frac{\partial^2 E_z^0}{\partial y^2} + h^2 E_z^0 = 0$$

This is a linear 2nd order differential equation with constant coefficients, which characteristic equation is:

$$r^2 + h^2 = 0 \leftrightarrow r^2 = -h^2 \leftrightarrow r = \pm jh$$

Depending on *h* value, this equation will behave in one out of two ways. If $h^2 > 0$ (*h* is real), solution is of the type:

$$y = c_1 e^{\alpha x}\cos(\beta x) + c_2 e^{\alpha x}\sin(\beta x)$$

where $\alpha = \frac{-b}{2a} \; and \; \beta = \frac{\sqrt{4ac-b}}{2a} \leftrightarrow \alpha = 0(b = 0) \; and \; \beta = h(a = 1 \; and \; c = h^2)$.

If $h^2 < 0$ (*h* is imaginary), solution is of the type:

$$y = Ce^{-vy} + De^{+vy} \qquad where \; h = jv$$

So, solution is:

$$E_z^0(y) = A\cos(hy) + B\sin(hy) \tag{38}$$

or,

$$E_z^0(y) = Ce^{-vy} + De^{+vy} \qquad where \; h = jv \tag{39}$$

Obtaining transversal components this way is only valid when $h \neq 0$. If $h = 0$, $\vec{E} \; and \; \vec{H}$ fields must be obtained through Maxwell's equation as in equations 40.

Also, the wave's electric and magnetic fields are related by Maxwell's equations:

$$\nabla \times \vec{E} = -j\omega\mu\vec{H}$$
$$\nabla \times \vec{H} = j\omega\varepsilon\vec{E}$$

or, in scalar form,

$$\begin{aligned} \frac{\partial E_z^0}{\partial y} + \gamma E_y^0 &= -j\omega\mu H_x^0 & \frac{\partial H_z^0}{\partial y} + \gamma H_y^0 &= j\omega\varepsilon E_x^0 \\ -\frac{\partial E_z^0}{\partial x} - \gamma E_x^0 &= -j\omega\mu H_y^0 & -\frac{\partial H_z^0}{\partial x} - \gamma H_x^0 &= j\omega\varepsilon E_y^0 \\ \frac{\partial E_y^0}{\partial x} - \frac{\partial E_x^0}{\partial y} &= -j\omega\mu H_z^0 & \frac{\partial H_y^0}{\partial x} - \frac{\partial H_x^0}{\partial y} &= j\omega\varepsilon E_z^0 \end{aligned} \quad (40)$$

There are some guided waves general properties that still have to be pointed out, such as:

- Cut-off frequency
- Phase constant
- Phase velocity
- Group velocity

The wave's **propagation constant** can be obtained from *h* using:

$$\gamma = \sqrt{h^2 - \omega^2\mu\varepsilon} \leftrightarrow$$
$$\leftrightarrow \gamma = \omega\sqrt{\mu\varepsilon}\sqrt{\frac{h^2}{\omega^2\mu\epsilon} - 1} \quad (41)$$

A guided wave going through a dielectric waveguide needs an imaginary part on its propagation constant γ. Assuming f_c as the **cut-off frequency** and its definition:

$$f_c = \frac{h}{2\pi\sqrt{\mu\varepsilon}}$$

Replacing this expression on equation 41 and after some manipulation,

$$\gamma = \omega\sqrt{\mu\varepsilon}\sqrt{\left(\frac{f_c}{f}\right)^2 - 1}$$

Looking at previous equation:

- If $f < f_c \rightarrow \gamma = \alpha$ (γ is real) and electric and magnetic fields are given by,

  $$\vec{E}(x,y,z) = \overrightarrow{E^0}(x,y)e^{-\alpha z}$$
  $$\vec{H}(x,y,z) = \overrightarrow{H^0}(x,y)e^{-\alpha z}$$

  these fields intensity decreases exponentially with *z*, thus an **evanescent mode**.

- If $f > f_c \rightarrow \gamma = j\beta$ (γ is imaginary) and electric and magnetic fields are given by,

$$\vec{E}(x,y,z) = \overrightarrow{E^0}(x,y)e^{-j\beta}$$
$$\vec{H}(x,y,z) = \overrightarrow{H^0}(x,y)e^{-j\beta}$$

representing thus a **propagating mode**.

Only frequencies above f_c ($f > f_c$) are able to propagate through the waveguide, which justifies designating it "cutoff frequency". When a mode is propagating, its phase constant β is given by,

$$\beta = \beta_m \sqrt{1 - \left(\frac{f_c}{f}\right)^2}$$

where $\beta_m = \omega\sqrt{\mu\varepsilon} = \frac{\omega}{v_m}$ is the phase constant of a propagating plane wave on a medium (μ, ε), with phase velocity $v_m = {}^{1}/_{\sqrt{\mu\varepsilon}}$.

Wavelength can be calculated as,

$$\lambda = \frac{\lambda_m}{\sqrt{1 - \left(\frac{f_c}{f}\right)^2}}$$

where $\lambda_m = \frac{2\pi}{\beta_m} = \frac{1}{f\sqrt{\mu\varepsilon}} = \frac{v_m}{f}$ is the wavelength of an infinite length medium (μ, ε). On this equation, being $f_c \neq 0$ implies that the wavelength of a propagating mode in a waveguide is larger than the wavelength of a wave propagating in an infinite length medium, considering the same dielectric characteristics for both waveguide and infinite length medium; in other words, $\lambda \geq \lambda_m$.

Phase velocity can be obtained through the following equation:

$$v_f = \frac{\omega}{\beta} = \frac{\omega}{\beta_m \sqrt{1 - \left(\frac{f_c}{f}\right)^2}} = \frac{v_m}{\sqrt{1 - \left(\frac{f_c}{f}\right)^2}}$$

The analysis of this equation leads to two main conclusions. First, on an ideal (non-dispersive, where μ and ε do not depend on frequency) medium, phase velocity is frequency dependent, as long as $f_c \neq 0$. On the other hand, $v_f \geq v_m$; in a medium with parameters (μ_0, ε_0), wave's phase velocity can be higher than the speed of light in vacuum. In fact, this is of no great physical importance for the wave's energy does not propagate at phase velocity but group velocity…

Group velocity definition is:

$$v_g = \frac{d\omega}{d\beta} = \frac{1}{\frac{d}{d\omega}\left(\beta_m \sqrt{1 - \left(\frac{f_c}{f}\right)^2}\right)} = v_m \sqrt{1 - \left(\frac{f_c}{f}\right)^2}$$

Looking at previous expression, it is not difficult to notice that $v_g \leq v_m$ which guarantees that group velocity is never greater than the speed of light in vacuum. Moreover, $v_g v_f = v_m^2$.

Appendix B

Waveguide modes representation code

```
%% Exercício constante no .pdf da FEUP (autoria Maria Inês Barbosa)
% Considere-se um guia dieléctrico planar constituído por um material de
% índice de refracção n1=2 e altura 2 cm, colocado no ar (n2=1).
% Este guia opera a 25 GHz. Neste caso, os valores característicos (h1)
são
% obtidos resolvendo independentemente cada um dos membros da equação:
% (n1/n2)^2*sqrt((w/c*n2)^2*((n1/n2)^2-1)-h1^2)=-h1*cot(h1*a/2)
% fplot('-h*cot(0.01*h)',[0 1000 -1000 5000])
% grid on
% hold on
% fplot('4*sqrt(((2e2/1.2*pi)^2)*3-h^2)',[0 1000 -1000 5000],'r')
%% Tese de Mestrado
% Exercício idêntico ao anterior mas com valores de acordo com as
dimensões
% apropriadas à utilização de comprimentos de onda de telecomunicações em
% guias dielétricos planares:
% lambda = 1.55 um (comprimento de onda)
% a = 2.6 um (altura do guia de onda planar)
% n1 = 2 (indice de refração do guia de onda planar)
% n2 = 1 (indice de refração do ar)
%% Modos TM par
clear all
% Representação gráfica das funções envolvidas
fplot('-h*cot(1.3e-6*h)',[0 8e6 -5e5 3e7])
grid on
hold on
% é efetuado o plot dos valores reais apenas para que o Matlab não emita
o
% warning de omissão de valores imaginários
fplot('real(4*sqrt(((2e6/1.55*pi)^2)*3-h^2))',[0 8e6 -5e5 3e7],'r')
title('TM_{even} modes at 1550 nm')
xlabel('h_1 [m^{-1}]')
ylabel('Normalised frequency v')
legend('-h_1*cot(h_1*a/2)',...
'(n_1/n_2)^2*sqrt((w/c*n_2)^2*((n_1/n_2)^2-1)-h_1^2)')
% Constantes
% a=2e-2; % altura do guia para 25 GHz
% lambda=1.2e-2; % comprimento de onda para 25 GHz
a=2.6e-6;
lambda1550=1.55e-6;
n1=2;
n2=1;
c=3e8;
f=c/lambda1550;
w=2*pi*f;
% Visualização e obtenção dos valores característicos h1, nos modos TM
% pares:
% step=1; % step para 25GHz
step=1e4; % step elevado para minimizar uso de memória
h=0:step:8e6-step;
f1=-h.*cot(a/2*h);
% apenas são considerados os valores reais de f2, até ao 1º zero desta,
% para que não seja efetuado o plot dos valores imaginários e estar de
```

```
% acordo com a representação gráfica obtida pela função fplot.m
f2=real((n1/n2)^2*sqrt((w/c*n2)^2*((n1/n2)^2-1)-h.^2));
P=InterX([h;f1],[h;f2]); % função que retorna pontos de interseção
h1=P(:,1:2:length(P)); % pontos de interseção de interesse
plot(h1(1,:),h1(2,:),'mo')
% O vetor h1 contem os valores característicos deste guia de onda planar,
% à frequência de trabalho a que corresponde o comprimento de onda 1.55
um.
% Foram obtidos os 3 valores característicos, correspondendo cada um
deles
% a um modo passível de propagação no guia de onda considerado, obtendo
% assim 3 modos pares TM passíveis de propagação.
% Em seguida há que calcular os coeficientes de decaimento no ar (v),
para
% que seja possível caracterizar completamente a equação de onda de cada
% um destes modos:
% v = sqrt((w/c)^2*(n1^2-n2^2)-h1^2)
v1550=sqrt(((w/c)^2)*(n1^2-n2^2)-h1.^2);
% Representação dos 3 modos encontrados para lambda=1550 nm
st=1e-8;
n=-2.6e-6:st:-1.3e-6-st;
E0_z1550_mode1=cos(h1(1,1)*n).*exp(v1550(1,1)*(n));
E0_z1550_mode2=cos(h1(1,2)*n).*exp(v1550(1,2)*(n));
E0_z1550_mode3=cos(h1(1,3)*n).*exp(v1550(1,3)*(n));
n=-1.3e-6:st:1.3e-6;
E0_z1550_mode1=[E0_z1550_mode1 cos(h1(1,1)*n)];
E0_z1550_mode2=[E0_z1550_mode2 cos(h1(1,2)*n)];
E0_z1550_mode3=[E0_z1550_mode3 cos(h1(1,3)*n)];
n=1.3e-6+st:st:2.6e-6;
E0_z1550_mode1=[E0_z1550_mode1 cos(h1(1,1)*n).*exp(-v1550(1,1)*(n))];
E0_z1550_mode2=[E0_z1550_mode2 cos(h1(1,2)*n).*exp(-v1550(1,2)*(n))];
E0_z1550_mode3=[E0_z1550_mode3 cos(h1(1,3)*n).*exp(-v1550(1,3)*(n))];
figure(2)
plot(-2.6:0.01:2.6,E0_z1550_mode1,...
-2.6:0.01:2.6,E0_z1550_mode2,...
-2.6:0.01:2.6,E0_z1550_mode3)
title('TM_{even} found modes profile at 1550 nm')
legend('mode 1','mode 2','mode 3')
axis tight
xlabel('Waveguide width [\mum]')
ylabel('Normalised profile E_z^0/(a/2) [V/m]')
grid on
% Considerando agora uma análise do valor característico (h1) do 1º modo
e
% correspondente coeficiente exponencial de decaimento (v) com o aumento
da
% frequência (1550 nm, 1300 nm e 820 nm):
lambda=[1.3e-6 0.82e-6];
f=c./lambda;
w=2*pi*f;
% Representação gráfica das funções envolvidas
figure(3)
% lambda = 1300 nm
fplot('-h*cot(1.3e-6*h)',[0 8.45e6 -5e5 4e7])
grid on
hold on
fplot('real(4*sqrt(((2e6/1.3*pi)^2)*3-h^2))',[0 8.45e6 -5e5 4e7],'r')
```

```
title('TM_{even} found modes at 1300 nm')
xlabel('h_1 [m^{-1}]')
ylabel('Normalised frequency v')
legend('-h_1*cot(h_1*a/2)',...
'(n_1/n_2)^2*sqrt((w/c*n_2)^2*((n_1/n_2)^2-1)-h_1^2)')
h1300=0:step:8.45e6-step;
f1=-h1300.*cot(a/2*h1300);
f2_1300=real((n1/n2)^2*sqrt((w(1)/c*n2)^2*((n1/n2)^2-1)-h1300.^2));
P_1300=InterX([h1300;f1],[h1300;f2_1300]); % função que retorna pontos de
interseção
h1_1300=P_1300(:,1:2:length(P_1300)); % pontos de interseção de interesse
plot(h1_1300(1,:),h1_1300(2,:),'mo')
v1300=sqrt(((w(1,1)/c)^2)*(n1^2-n2^2)-h1_1300.^2);
figure(4)
% lambda = 820 nm
fplot('-h*cot(1.3e-6*h)',[0 1.33e7 -5e5 6e7])
grid on
hold on
fplot('real(4*sqrt(((2e6/0.82*pi)^2)*3-h^2))',[0 1.33e7 -5e5 6e7],'r')
title('TM_{even} found modes at 820 nm')
xlabel('h_1 [m^{-1}]')
ylabel('Normalised frequency v')
legend('-h_1*cot(h_1*a/2)',...
'(n_1/n_2)^2*sqrt((w/c*n_2)^2*((n_1/n_2)^2-1)-h_1^2)')
h820=0:step:1.33e7-step;
f1=-h820.*cot(a/2*h820);
f2_820=real((n1/n2)^2*sqrt((w(2)/c*n2)^2*((n1/n2)^2-1)-h820.^2));
P_820=InterX([h820;f1],[h820;f2_820]); % função que retorna pontos de
interseção
h1_820=P_820(:,1:2:length(P_820)); % pontos de interseção de interesse
plot(h1_820(1,:),h1_820(2,:),'mo')
v820=sqrt(((w(1,2)/c)^2)*(n1^2-n2^2)-h1_820.^2);
% Valores característicos (h1) e coeficientes exponenciais de decaimento
% (v), do 1° modo par, nos comprimentos de onda considerados:
h=[h1(1,1) h1_1300(1,1) h1_820(1,1)];
v=[v1550(1,1) v1300(1,1) v820(1,1)];
st=1e-8;
n=-2.6e-6:st:-1.3e-6-st;
E0_z1550=cos(h(1,1)*n).*exp(v(1,1)*(n));
E0_z1300=cos(h(1,2)*n).*exp(v(1,2)*(n));
E0_z820=cos(h(1,3)*n).*exp(v(1,3)*(n));
n=-1.3e-6:st:1.3e-6;
E0_z1550=[E0_z1550 cos(h(1,1)*n)];
E0_z1300=[E0_z1300 cos(h(1,2)*n)];
E0_z820=[E0_z820 cos(h(1,3)*n)];
n=1.3e-6+st:st:2.6e-6;
E0_z1550=[E0_z1550 cos(h(1,1)*n).*exp(-v(1,1)*(n))];
E0_z1300=[E0_z1300 cos(h(1,2)*n).*exp(-v(1,2)*(n))];
E0_z820=[E0_z820 cos(h(1,3)*n).*exp(-v(1,3)*(n))];
figure(5)
plot(-2.6:0.01:2.6,E0_z1550,...
-2.6:0.01:2.6,E0_z1300,...
-2.6:0.01:2.6,E0_z820)
title('TM_{even} fundamental modes at 1550, 1300 and 820 nm')
legend('1550 nm','1300 nm','820 nm')
axis tight
xlabel('Waveguide width [\mum]')
```

```
ylabel('Normalised profile E_z^0/(a/2) [V/m]')
grid on
% É possível verificar que o aumento de frequência conduz a um aumento do
% valor do coeficiente exponencial de decaimento, ou seja, quanto mais
alta
% for a frequência mais a onda fica confinada ao interior do guia e menor
é
% a componente evanescente desta. O que faz todo o sentido mesmo segundo
% uma análise superficial pois, ao aumentar a frequência, diminui o
% comprimento de onda e, mantendo a dimensão do guia, esta terá mais
% "espaço" para progredir ao longo do meio de propagação.
% Considerando o exposto anteriormente, se a frequência for diminuindo,
% haverá um valor para o qual "v" se poderá anular e, consequentemente,
as
% ondas deixarão de estar confinadas ao guia de onda. Este valor, v = 0,
no
% que respeita à frequência, corresponde à frequência de corte:
% fc_TM_par = (n-1/2)*c/(a*sqrt(n1^2-n2^2)), com n = 1,2,3,... (índice do
% modo a considerar)
% A frequência de corte é, regra geral, referente ao 1º dos modos, ou
seja,
% n = 1:
fc_TM_par=c/(2*a*sqrt(n1^2-n2^2));
% Para encontrar o número de modos TM pares em propagação num dado guia e
% para determinada frequência, poderá ser usada a expressão:
% fc_TM_par = (n-1/2)*c/(a*sqrt(n1^2-n2^2)), com n = 1,2,3,... (índice do
% modo a considerar) e incrementar n até que a frequência de corte obtida
% seja superior à frequência de trabalho desejada.
%% Modos TM ímpar
% Para os modos TM ímpares o procedimento é análogo ao tido para os modos
% TM pares. Nos modos ímpares, contudo, a equação característica é algo
% diferente:
% sqrt((w/c)^2*(n1^2-n2^2)-h1^2)=epson2/epson1*h1*tan(h1*a/2)
% Considerando os valores do exercício inicial:
clear all
% Representação gráfica das funções envolvidas
figure(6)
fplot('1/4*h*tan(1.3e-6*h)',[0 7.1e6 -5e5 8e6])
grid on
hold on
% é efetuado o plot dos valores reais apenas para que o Matlab não emita
o
% warning de omissão de valores imaginários
fplot('real(sqrt(((2e6/1.55*pi)^2)*3-h^2))',[0 7.1e6 -5e5 8e6],'r')
title('TM_{odd} modes at 1550 nm')
xlabel('h_1 [m^{-1}]')
ylabel('Normalised frequency v')
legend('\epsilon_2/\epsilon_1*h_1*tan(h_1*a/2)',...
'sqrt((w/c)^2*(n_1^2-n_2^2)-h_1^2)')
% Constantes
a=2.6e-6;
lambda=[1.55e-6 1.3e-6 0.8e-6];
n1=2;
n2=1;
epson1=n1^2;
epson2=n2^2;
c=3e8;
```

```
f=c./lambda;
w=2*pi.*f;
% Visualização e obtenção dos valores característicos h1, nos modos TM
% ímpares:
step=1e4; % step elevado para minimizar uso de memória
h=0:step:7.1e6-step;
f1=epson2/epson1*h.*tan(a/2*h);
% apenas são considerados os valores reais de f2, até ao 1º zero desta,
% para que não seja efetuado o plot dos valores imaginários e estar de
% acordo com a representação gráfica obtida pela função fplot.m
f2=real(sqrt((w(1,1)/c)^2*(n1^2-n2^2)-h.^2));
P=InterX([h;f1],[h;f2]); % função que retorna pontos de interseção
h1_1550=P(:,1:2:length(P)); % pontos de interseção de interesse
plot(h1_1550(1,:),h1_1550(2,:),'mo')
% Calculando os coeficientes de decaimento exponencial para os modos
% obtidos:
% v = sqrt((w/c)^2*(n1^2-n2^2)-h1^2)
v1550=sqrt(((w(1,1)/c)^2)*(n1^2-n2^2)-h1_1550.^2);
h1300=0:step:8.45e6-step;
f1_1300=epson2/epson1*h1300.*tan(a/2*h1300);
f2_1300=real(sqrt((w(1,2)/c*n2)^2*(n1^2-n2^2)-h1300.^2));
P_1300=InterX([h1300;f1_1300],[h1300;f2_1300]); % função que retorna
pontos de interseção
h1_1300=P_1300(:,1:2:length(P_1300)); % pontos de interseção de interesse
v1300=sqrt(((w(1,2)/c)^2)*(n1^2-n2^2)-h1_1300.^2);
h820=0:step:1.33e7-step;
f1_820=epson2/epson1*h820.*tan(a/2*h820);
f2_820=real(sqrt((w(1,3)/c*n2)^2*(n1^2-n2^2)-h820.^2));
P_820=InterX([h820;f1_820],[h820;f2_820]); % função que retorna pontos de
interseção
h1_820=P_820(:,1:2:length(P_820)); % pontos de interseção de interesse
v820=sqrt(((w(1,3)/c)^2)*(n1^2-n2^2)-h1_820.^2);
st=1e-8;
n=-2.6e-6:st:-1.3e-6-st;
E0_z1550=-sin(h1_1550(1,1)*n).*exp(v1550(1,1)*n);
E0_z1300=-sin(h1_1300(1,1)*n).*exp(v1300(1,1)*n);
E0_z820=-sin(h1_820(1,1)*n).*exp(v820(1,1)*n);
n=-1.3e-6:st:1.3e-6;
E0_z1550=[E0_z1550 sin(h1_1550(1,1)*n)];
E0_z1300=[E0_z1300 sin(h1_1300(1,1)*n)];
E0_z820=[E0_z820 sin(h1_820(1,1)*n)];
n=1.3e-6+st:st:2.6e-6;
E0_z1550=[E0_z1550 sin(h1_1550(1,1)*n).*exp(-v1550(1,1)*n)];
E0_z1300=[E0_z1300 sin(h1_1300(1,1)*n).*exp(-v1300(1,1)*n)];
E0_z820=[E0_z820 sin(h1_820(1,1)*n).*exp(-v820(1,1)*n)];
figure(7)
plot(-2.6:0.01:2.6,E0_z1550,...
-2.6:0.01:2.6,E0_z1300,...
-2.6:0.01:2.6,E0_z820)
title('TM_{odd} fundamental modes at 1550, 1300 and 820 nm')
legend('1550 nm','1300 nm','820 nm')
axis tight
xlabel('Waveguide width [\mum]')
ylabel('Normalised profile E_z^0/(a/2) [V/m]')
grid on
% Frequência de corte dos modos ímpares:
% fc_TM_impar=(n-1)*c/(a*sqrt(n1^2-n2^2)), com n=1,2,3,...
```

```
% Caso n=1, ou seja o 1° modo TM ímpar, fc_TM_impar=0 => o 1° dos modos
TM
% ímpares está sempre presente!!!
%% Modos TE par
clear all
% Representação gráfica das funções envolvidas
figure(8)
fplot('-h*cot(1.3e-6*h)',[0 8e6 -5e5 8e6])
grid on
hold on
% é efetuado o plot dos valores reais apenas para que o Matlab não emita
o
% warning de omissão de valores imaginários
fplot('real(sqrt(((2e6/1.55*pi)^2)*3-h^2))',[0 8e6 -5e5 8e6],'r')
title('TE_{even} modes at 1550 nm')
xlabel('h_1 [m^{-1}]')
ylabel('Normalised frequency v')
legend('-h_1*cot(h_1*a/2)',...
'sqrt((w/c)^2*(n_1^2-n_2^2)-h_1^2)')
% Constantes
a=2.6e-6;
lambda1550=1.55e-6;
n1=2;
n2=1;
c=3e8;
f=c/lambda1550;
w=2*pi*f;
% Equação característica dos modos TE pares:
% sqrt((w/c)^2*(n1^2-n2^2)-h1^2)=-h*cot(a/2*h)
% Visualização e obtenção dos valores característicos h1, nos modos TE
% pares:
step=1e4; % step elevado para minimizar uso de memória
h=0:step:8e6-step;
f1=-h.*cot(a/2*h);
% apenas são considerados os valores reais de f2, até ao 1° zero desta,
% para que não seja efetuado o plot dos valores imaginários e estar de
% acordo com a representação gráfica obtida pela função fplot.m
f2=real(sqrt((w/c)^2*(n1^2-n2^2)-h.^2));
P=InterX([h;f1],[h;f2]); % função que retorna pontos de interseção
h1=P(:,1:2:length(P)); % pontos de interseção de interesse
plot(h1(1,:),h1(2,:),'mo')
% De realçar que a expressão dos coeficientes exponenciais de decaimento
% para os modos TE pares é igual à obtida para os modos TM pares:
% v = sqrt((w/c)^2*(n1^2-n2^2)-h1^2)
v1550=sqrt(((w/c)^2)*(n1^2-n2^2)-h1.^2);
% O mesmo se passa em relação à expressão para obtenção da frequência de
% corte para os modos TE pares, ou seja, é igual à expressão obtida
% anteriormente para os modos TM pares:
% fc_TM_par=c/(2*a*sqrt(n1^2-n2^2))
fc_TE_par=c/(2*a*sqrt(n1^2-n2^2));
%% Modos TE ímpares
% Representação gráfica das funções envolvidas
figure(9)
fplot('h*tan(1.3e-6*h)',[0 8e6 -5e5 8e6])
grid on
hold on
```

```
% é efetuado o plot dos valores reais apenas para que o Matlab não emita
o
% warning de omissão de valores imaginários
fplot('real(sqrt(((2e6/1.55*pi)^2)*3-h^2))',[0 8e6 -5e5 8e6],'r')
title('TE_{odd} modes at 1550 nm')
xlabel('h_1 [m^{-1}]')
ylabel('Normalised frequency v')
legend('h_1*tan(h_1*a/2)',...
'sqrt((w/c)^2*(n_1^2-n_2^2)-h_1^2)')
% Equação característica dos modos TE ímpares:
% sqrt((w/c)^2*(n1^2-n2^2)-h1^2)=h*tan(a/2*h)
% Visualização e obtenção dos valores característicos h1, nos modos TE
% ímpares:
step=1e4; % step elevado para minimizar uso de memória
h=0:step:7.1e6-step;
f1=h.*tan(a/2*h);
% apenas são considerados os valores reais de f2, até ao 1º zero desta,
% para que não seja efetuado o plot dos valores imaginários e estar de
% acordo com a representação gráfica obtida pela função fplot.m
f2=real(sqrt((w/c)^2*(n1^2-n2^2)-h.^2));
P=InterX([h;f1],[h;f2]); % função que retorna pontos de interseção
h1=P(:,1:2:length(P)); % pontos de interseção de interesse
plot(h1(1,:),h1(2,:),'mo')
% Como a expressão dos coeficientes exponenciais de decaimento para os
% modos TE pares se mantem para os modos TE ímpares, os valores de 'v'
% utilizados nos cálculos serão os já obtidos anteriormente:
% v = sqrt((w/c)^2*(n1^2-n2^2)-h1^2)
% No que respeita à expressão da frequência de corte para os modos TE
% ímpares, esta é igual à obtida para os modos TM ímpares:
% fc_TE_impar=(n-1)*c/(a*sqrt(n1^2-n2^2)), com n=1,2,3,...
% Caso n=1, ou seja o 1º modo TE ímpar, fc_TE_impar=0 => o 1º dos modos
TE
% ímpares está sempre presente, tal como já acontecia com os modos TM
% ímpares!!!
```

InterX function code

```
function P = InterX(L1,varargin)
%INTERX Intersection of curves
% P = INTERX(L1,L2) returns the intersection points of two curves L1
% and L2. The curves L1,L2 can be either closed or open and are described
% by two-row-matrices, where each row contains its x- and y- coordinates.
% The intersection of groups of curves (e.g. contour lines, multiply
% connected regions etc) can also be computed by separating them with a
% column of NaNs as for example
%
% L = [x11 x12 x13 ... NaN x21 x22 x23 ...;
% y11 y12 y13 ... NaN y21 y22 y23 ...]
%
% P has the same structure as L1 and L2, and its rows correspond to the
% x- and y- coordinates of the intersection points of L1 and L2. If no
% intersections are found, the returned P is empty.
%
% P = INTERX(L1) returns the self-intersection points of L1. To keep
% the code simple, the points at which the curve is tangent to itself are
% not included. P = INTERX(L1,L1) returns all the points of the curve
```

```
% together with any self-intersection points.
%
% Example:
% t = linspace(0,2*pi);
% r1 = sin(4*t)+2; x1 = r1.*cos(t); y1 = r1.*sin(t);
% r2 = sin(8*t)+2; x2 = r2.*cos(t); y2 = r2.*sin(t);
% P = InterX([x1;y1],[x2;y2]);
% plot(x1,y1,x2,y2,P(1,:),P(2,:),'ro')
% Author : NS
% Version: 3.0, 21 Sept. 2010
% Two words about the algorithm: Most of the code is self-explanatory.
% The only trick lies in the calculation of C1 and C2. To be brief, this
% is essentially the two-dimensional analog of the condition that needs
% to be satisfied by a function F(x) that has a zero in the interval
% [a,b], namely
% F(a)*F(b) <= 0
% C1 and C2 exactly do this for each segment of curves 1 and 2
% respectively. If this condition is satisfied simultaneously for two
% segments then we know that they will cross at some point.
% Each factor of the 'C' arrays is essentially a matrix containing
% the numerators of the signed distances between points of one curve
% and line segments of the other.
%...Argument checks and assignment of L2
error(nargchk(1,2,nargin));
if nargin == 1,
L2 = L1; hF = @lt; %...Avoid the inclusion of common points
else
L2 = varargin{1}; hF = @le;
end
%...Preliminary stuff
x1 = L1(1,:)'; x2 = L2(1,:);
y1 = L1(2,:)'; y2 = L2(2,:);
dx1 = diff(x1); dy1 = diff(y1);
dx2 = diff(x2); dy2 = diff(y2);
%...Determine 'signed distances'
S1 = dx1.*y1(1:end-1) - dy1.*x1(1:end-1);
S2 = dx2.*y2(1:end-1) - dy2.*x2(1:end-1);
C1 = feval(hF,D(bsxfun(@times,dx1,y2)-bsxfun(@times,dy1,x2),S1),0);
C2 = feval(hF,D((bsxfun(@times,y1,dx2)-bsxfun(@times,x1,dy2))',S2'),0)';
%...Obtain the segments where an intersection is expected
[i,j] = find(C1 & C2);
if isempty(i),P = zeros(2,0);return; end;
%...Transpose and prepare for output
i=i'; dx2=dx2'; dy2=dy2'; S2 = S2';
L = dy2(j).*dx1(i) - dy1(i).*dx2(j);
i = i(L~=0); j=j(L~=0); L=L(L~=0); %...Avoid divisions by 0
%...Solve system of eqs to get the common points
P = unique([dx2(j).*S1(i) - dx1(i).*S2(j), ...
dy2(j).*S1(i) - dy1(i).*S2(j)]./[L L],'rows')';

function u = D(x,y)
u = bsxfun(@minus,x(:,1:end-1),y).*bsxfun(@minus,x(:,2:end),y);
end

end
```

Appendix C

Bessel's functions

1st kind Bessel's function

For $n \in \mathbb{N}$,

$$J_n(x) = \sum_{m=0}^{\infty} \frac{(-1)^m x^{n+2m}}{m!\,\Gamma(m+n+1)2^{n+2m}}$$

where Γ is the *gamma* function that verifies $\Gamma(n+1) = n\Gamma(n)$. Because n is an integer, it can be written $\Gamma(n+1) = n! \rightarrow \Gamma(1) = 0! = 1$ (if $n < 0$, $\Gamma(n+1) = \infty$).

Again, Matlab has been used to calculate the necessary values in order to create the following tables and figures (see Appendix D). Figure C59 shows J_n behaviour as x increases,

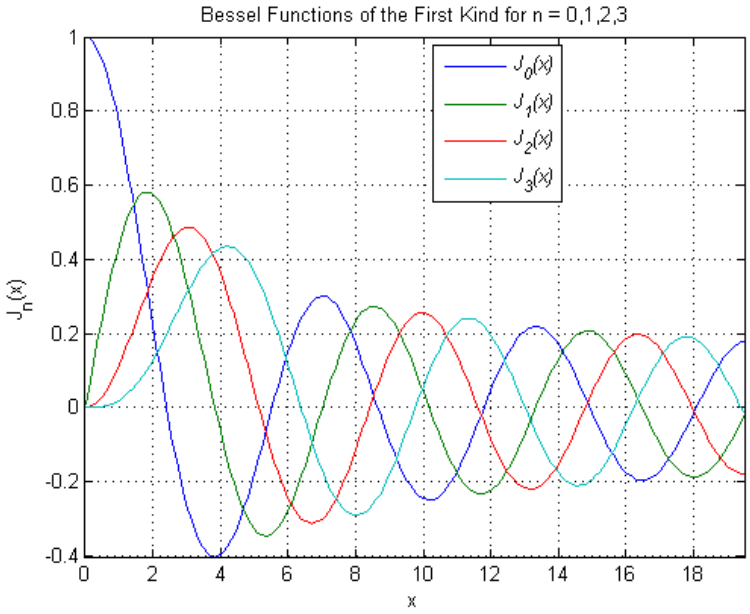

Figure C59 - First kind Bessel's functions.

Some properties of this function:

- For $x = 0$ (see series' expression),

$$n \neq 0 \Longrightarrow J_n(0) = 0$$
$$n = 0 \Longrightarrow J_n(0) = 1$$

- It can be shown that,

$$J_{-n}(x) = (-1)^n J_n(x)$$

- Figure C60 plots these functions and the points where they cross $y = 0$. One can observe their amplitude decreases as x increases and it is also noticeable that the zero crossing points get closer, in pairs and for alternating functions, as x increases. Table C21 has been created with the x coordinate value for each zero crossing point found on Figure C60.

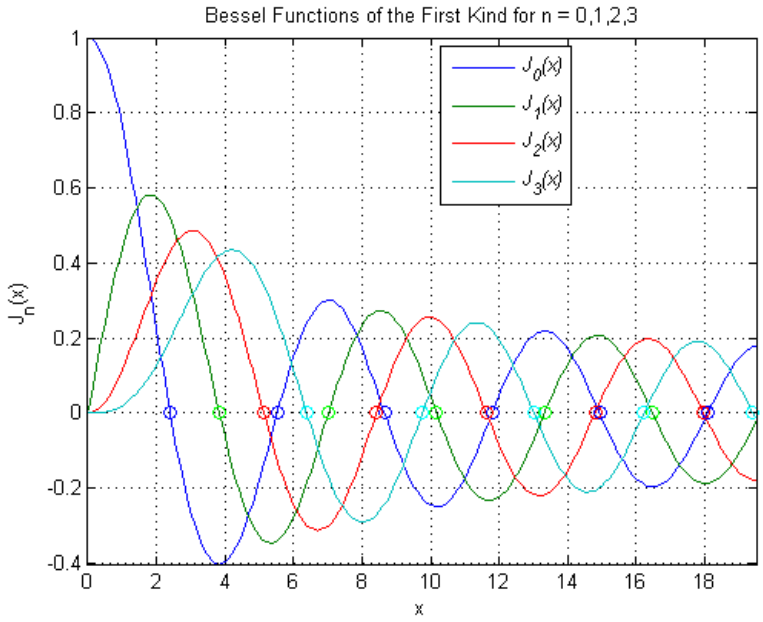

Figure C60 - Zero crossing for Bessel functions.

Table C21 - First kind Bessel functions zero crossings.

zero	$J_0(x)$	$J_1(x)$	$J_2(x)$	$J_3(x)$
1	2.404928	0	0	0
2	5.520239	3.832000	5.135855	6.380274
3	8.653868	7.015695	8.417344	9.761137
4	11.791557	10.173549	11.619925	13.015264
5	14.931003	13.323776	14.795959	16.223537
6	18.071106	16.470679	17.959879	19.409448

In order to obtain the transverse components of $\vec{E}$ and $\vec{H}$ in a circular waveguide, it will be necessary to calculate $J_n'(x)$, J_n's derivative. It is possible to find J_n's derivative through J_{n-1} and J_{n+1} expressions using next equation:

$$J_n'(x) = \frac{1}{2}[J_{n-1}(x) - J_{n+1}(x)]$$

As before, Figure C61 shows $J_n'(x)$ as x varies for some values of n and the zero crossing points. Again, when it comes to the behaviour of the zero crossing points as x increases, the same conclusions as with Bessel's functions can be drawn:

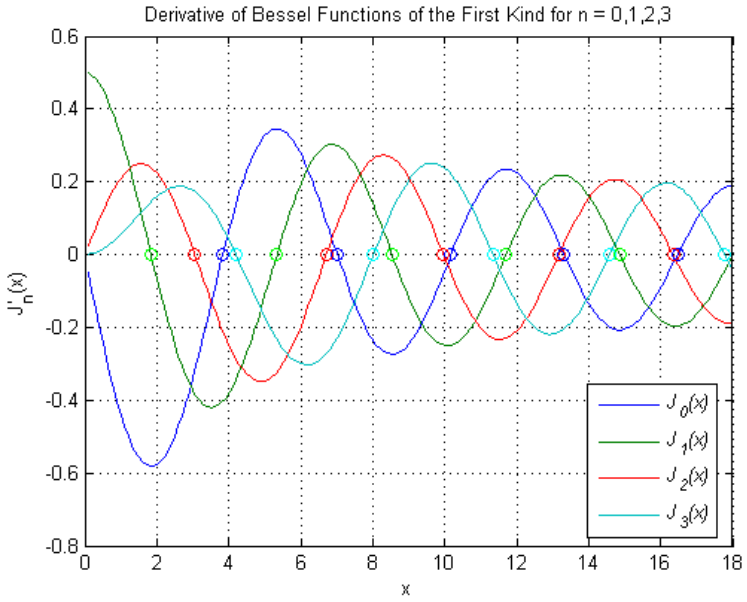

Figure C61 - Bessel function derivatives and crossing points.

Table C22 has been created with the x coordinate value for each zero crossing point found on Figure C61.

Table C22 - First kind Bessel's derivative function zero crossings.

zero	$J_0'(x)$	$J_1'(x)$	$J_2'(x)$	$J_3'(x)$
1	3.832000	1.841297	3.054030	4.201176
2	7.015695	5.331642	6.706175	8.015304
3	10.173549	8.536458	9.969552	11.346020
4	13.323776	11.706037	13.170433	14.585872
5	16.470679	14.863655	16.347598	17.788762

2nd kind Bessel's function

For $n \in \mathbb{N}$,

$$N_n(x) = \lim_{p \to n} \frac{J_p(x)\cos(p\pi) - J_{-p}(x)}{\sin(p\pi)}$$

Since,

$$\frac{J_n(x)\cos(n\pi) - J_{-n}(x)}{\sin(n\pi)} = \frac{J_p(x)\,(-1)^n - (-1)^n J_n(x)}{\sin(n\pi)} = \frac{0}{0}$$

l'Hopital's rule must be applied to circumvent the indeterminate form but Matlab handles this easily with its Bessel function of the second kind. Figure C62 depicts graphically Bessel function of the second kind.

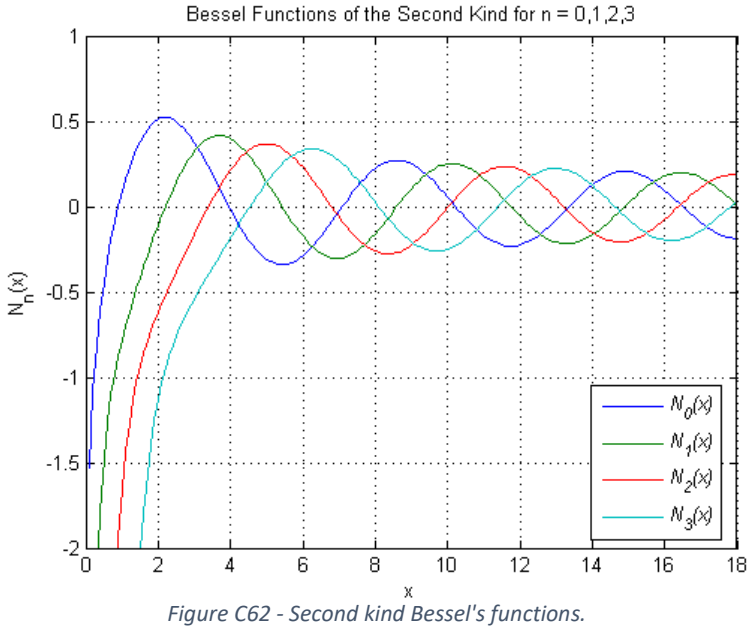

Figure C62 - Second kind Bessel's functions.

It should be noticed that, when the origin is to be considered, this function is infinite for any order.

Modified Bessel function of 1st kind

For $n \in \mathbb{N}$,

$$I_n(x) = j^{-n} J_n(jx) = \sum_{k=0}^{\infty} \frac{(x/2)^{n+2k}}{k!\,(n+k)!}$$

Figure C63 shows $I_n(x)$ dependency of x for some values of n. As can be observed, when x assumes higher values, this function's behaviour can be approximately explained by:

$$I_n(x) \cong e^x/\sqrt{2\pi x}$$

that is, it follows the argument's tendency towards infinity. For this reason, the modified Bessel function of first kind should not be considered when the interest region includes infinity.

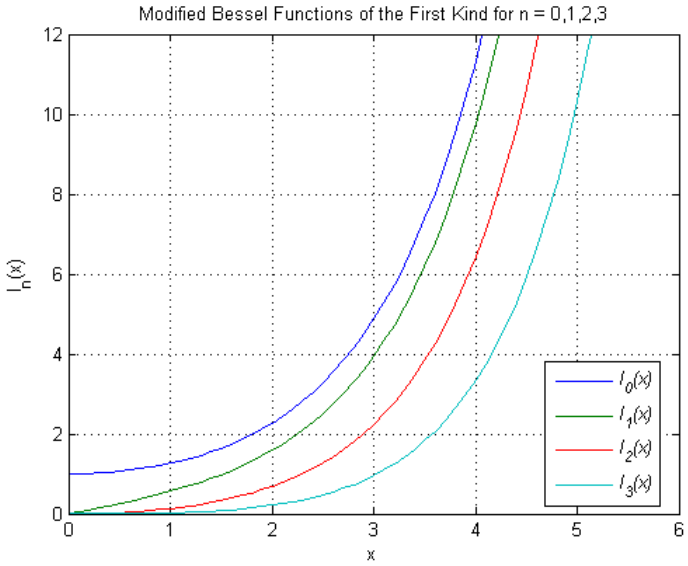

Figure C63 - First kind modified Bessel function.

Modified Bessel function of 2nd kind

For $n \in \mathbb{N}$,

$$K_n(x) = \lim_{p \to n} \frac{\pi}{2\sin(p\pi)} \left[I_{-p}(x) - I_p(x) \right]$$

Assuming n as an integer leads to $I_n(x) = I_n(x)$ and an indeterminate form is encountered when solving the limit. Again, l'Hopital's rule must be applied and Matlab handles it as before. Variation of $K_n(x)$ with x is shown on Figure C64. As can be observed, when the region of interest includes the origin and for any n, $K_n(x)$ is infinity.

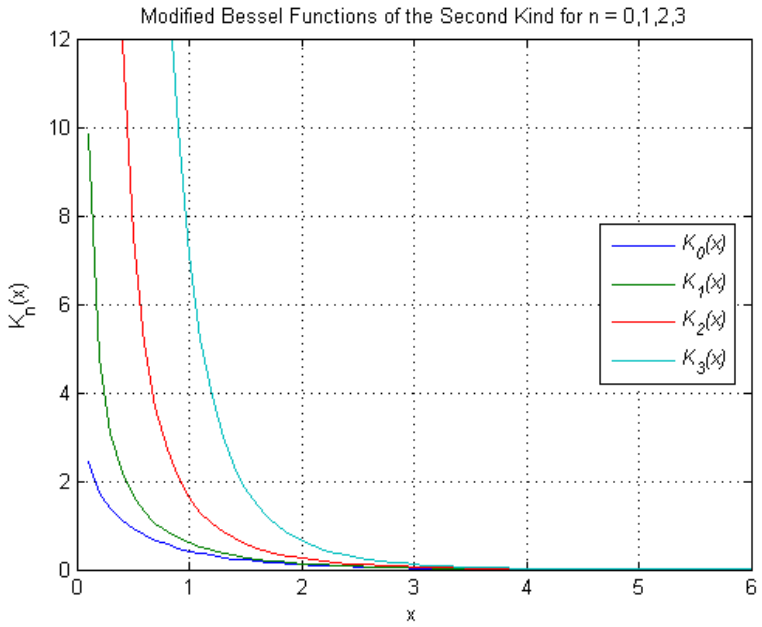

Figure C64 - Second kind modified Bessel function.

Appendix D

LP mode code

```
%% This script has been adapted from:
%
% Optical Fibre Toolbox two-layer demo
% This script demonstrates the use of Optical Fibre Toolbox functions
for
% calculation of two-layer fibres modes.
%
% (cc-by) K. Karapetyan et al., AG Meschede, Uni Bonn, 2008--2011
%
% kotya.karapetyan@gmail.com, http://agmeschede.iap.uni-bonn.de
%
% This script tries to show how the linearly polarized mode LP11 can
be
% achieved through weak guiding approximation, for it is the sum of
two
% other modes: - HE_21 and either TM_01 or TE_01.
% Because commercially available dielectric materials have been used,
all
% calculations and plots have taken into account their dispersion
% coefficients (they are not ideal materials, this time...).
% In order to work properly, this script requires fieldsEM.m function
which
% which is provided and should be in the same path.
%
% Paulo Lourenço, ISEL, Lisbon-Portugal, Summer 2016
%
% a37361@alunos.isel.pt, http://www.isel.pt
%%
clc
clear
close all
tStart = tic;
addpath('..')
fprintf('%s\n', 'Calculating...');
%% Specify the fibre parameters
% Note: diameter is specified in micrometers, wavelength in nanometers
% Fibre materials (core, cladding)
materials = {'sm800core'; 'silica'};
% materials = {1.448;1.444};
% Fibre structure description
fibre = struct(...
'materials', {materials});
%% Create the task for dispersion curves calculation
% Argument for dispersion curves calculation
argument = struct(...
'type', 'wvl',... % calculate vs. wavelength
'harmonic', 1,... % required
'min', 600,... % calculate from
'max', 1000); % calculate to
% Specify which modes to search for
modeTask = struct(...
'nu', [0 1 2],... % first modal index
'type', {{'hybrid', 'te', 'tm'}},... % mode types
'maxmode', 3,... % how many modes of each type and NU to calculate
'diameter', 5.6);%,... % parameter, structure diameter, if argument is
wavelength
%'region', 'cladding');
```

```
%% Find guided modes and calculate the dispersion curves
infomode = false;
modes = buildModes(argument, fibre, modeTask, infomode);
% Display calculated dispersion curves
showModes(modes, 'Modal dispersion in a fibre');
% Show which modes have been calculated
fprintf('Modes found:\n');
for i = 1:numel(modes)
fprintf('%s\n', modeDescription(modes(i), infomode));
end
% This fibre has a very low refractive index step between cladding and
% core (0.005) so is, therefore, weakly guiding. In a weakly guided
fibre,
% modes HE21, TE01 and TM01 lie very close to each other, as can be
seen
% in the picture. In the weakly guidance approximation, the mode HE11
is
% sometimes denoted as LP01 mode and HE21, TM01, TE01 modes triplet as
LP21
% mode.
%% Field distribution
poiType = 'wvl';
poi = 900;
par = 5.6;
infomode = false;
getWholeMode = false;
% Creation of EM fields for modes HE_11, HE_21, TE_01 and TM_01
for c=1:numel(modes)
task = struct('modetype', modes(c,1).modetype, 'modeindex',
modes(c,1).modeindex);
n = neff(poiType, poi, par, fibre, task, infomode, getWholeMode);
F(c,1)=fieldsEM(poi,par,fibre,task,n);
end
% HE_11 is also known as LP_01 or fundamental mode; F(1) this case.
% LP_11 results from the sum of HE_21 with TE_01 or with TM_01; F(2) +
F(3)
% or F(2) + F(4), respectively.
LPte=F(2); % HE_21 mode that is going to be summed to TE/TM mode
LPte.E1=LPte.E1+F(3).E1;
LPte.E2=LPte.E2+F(3).E2;
LPte.H1=LPte.H1+F(3).H1;
LPte.H2=LPte.H2+F(3).H2;
LPtm=F(2); % HE_21 mode that is going to be summed to TE/TM mode
LPtm.E1=LPtm.E1+F(4).E1;
LPtm.E2=LPtm.E2+F(4).E2;
LPtm.H1=LPtm.H1+F(4).H1;
LPtm.H2=LPtm.H2+F(4).H2;
displayField2(F(1), par) % HE_11 or LP_01
displayField2(F(2), par) % HE_21
displayField2(F(3), par) % TE_01
displayField2(F(4), par) % TM_01
displayField2(LPte, par) % LP_11 (HE_21 + TE_01)
displayField2(LPtm, par) % LP_11 (HE_21 + TM_01)
```

EM fields function code

```
function F = fieldsEM(poi,par,fibre,task,n)
d = par;
window = 3 * d;
lambda = poi;
Nr = [500, 1000]; % inn, out
Nphi = [64, 64]; % inn, out
```

```
dr_inn = d/2 / Nr(1);
dr_out = (window/2 - d/2) / Nr(2);
dphi_inn = 2 * pi / Nphi(1);
dphi_out = 2 * pi / Nphi(2);
F = struct('dr', [dr_inn dr_out], 'dphi', [dphi_inn dphi_out], ...
'diam', [d window], 'E1', [], 'H1', [], 'E2', [], 'H2', []);
FG = fieldGrid(F);
[F.E1, F.H1] = modeField(lambda, d, n, fibre, task, FG.R1, FG.PHI1);
[F.E2, F.H2] = modeField(lambda, d, n, fibre, task, FG.R2, FG.PHI2);
F.FG = FG;
```

n_{eff} vs V for HE/EH and TE/TM modes code

```
%% This script requires Matlab OFT (Optical Fibre Toolbox) installed
and
% plots n_eff vs. V for skew/hybrid and TE/TM modes.
clc
clear
close all
% Note: diameter is specified in micrometers, wavelength in nanometers
%% Specify the fibre parameters
% Fibre materials (core, cladding)
% materials = {'sm800core'; 'silica'};
materials = {1.448; 1.444};
% Fibre structure description
fibre = struct(...
'materials', {materials});
%% Create the task for dispersion curves calculation
% Argument for dispersion curves calculation
argument = struct(...
'type', 'wvl',... % calculate vs. wavelength
'harmonic', 1,... % required
'min', 300,... % calculate from
'max', 5000); % calculate to
% 'max', 2000); % calculate to
% Specify which modes to search for
task = struct(...
'nu', [0 1 2],... % first modal index
'type', {{'hybrid', 'te', 'tm'}},... % mode types
'maxmode', 5,... % how many modes of each type and NU to calculate
'diameter', 10);%,... % parameter, structure diameter, if argument is
wavelength
%'region', 'cladding');
%% Find guided modes and calculate the dispersion curves
addpath('..')
fprintf('%s', 'Calculating... ');
modes = buildModes(argument, fibre, task, false);
fprintf('%s\n', 'Done');
%% Print which modes have been calculated
fprintf('Modes found:\n');
for i = 1:numel(modes)
fprintf('%s\n', modeDescription(modes(i), false));
end
%% Convert modes to V-parameter vs. n_eff and plot them
modesV = modeVsV(modes);
figure()
plot(modesV(1).ARG,modesV(1).NEFF,modesV(2).ARG,modesV(2).NEFF,...
modesV(3).ARG,modesV(3).NEFF,modesV(4).ARG,modesV(4).NEFF,...
modesV(5).ARG,modesV(5).NEFF,modesV(6).ARG,modesV(6).NEFF,...
modesV(7).ARG,modesV(7).NEFF,modesV(8).ARG,modesV(8).NEFF,...
modesV(9).ARG,modesV(9).NEFF,modesV(10).ARG,modesV(10).NEFF,...
modesV(11).ARG,modesV(11).NEFF,modesV(12).ARG,modesV(12).NEFF,...
```

```
modesV(13).ARG,modesV(13).NEFF,modesV(14).ARG,modesV(14).NEFF,...
modesV(15).ARG,modesV(15).NEFF,modesV(16).ARG,modesV(16).NEFF)
axis([0 11 1.444 1.448])
grid on
text(1.6,1.4456,'HE_{11}','color','b')
text(4.3,1.445,'EH_{11}','color',[0 0.498 0])
text(5.8,1.4456,'HE_{12}','color','r')
text(8,1.445,'EH_{12}','color',[0 0.75 0.75])
text(10,1.4456,'HE_{13}','color','m')
text(3.5,1.4456,'HE_{21}','color',[0.749 0.749 0])
text(6,1.445,'EH_{21}','color',[0.247 0.247 0.247])
text(7.5,1.4456,'HE_{22}','color','b')
text(9.6,1.445,'EH_{22}','color',[0 0.5 0])
text(10.4,1.445,'HE_{23}','color','r')
text(2.5,1.4445,'TE_{01}','color',[0 0.75 0.75])
text(5.9,1.4445,'TE_{02}','color','m')
text(9.4,1.4445,'TE_{03}','color',[0.75 0.75 0])
text(2.3,1.4442,'TM_{01}','color',[0.25 0.25 0.25])
text(5.6,1.4442,'TM_{02}','color','b')
text(8.9,1.4442,'TM_{03}','color',[0 0.5 0])
xlabel('V-parameter')
ylabel('n_{eff}')
title('Effecttive Refraction Index (n_{eff}) vs. Normalized Frequency
(V)')
```

Appendix E

Qualitative analysis of waveguide modes formation

Total internal reflection

Referring to Figure E65, the condition for total internal reflection at the core-cladding interface is given by:

$$n_1 \sin\left({}^{\pi}/_{2} - \phi\right) \geq n_0$$

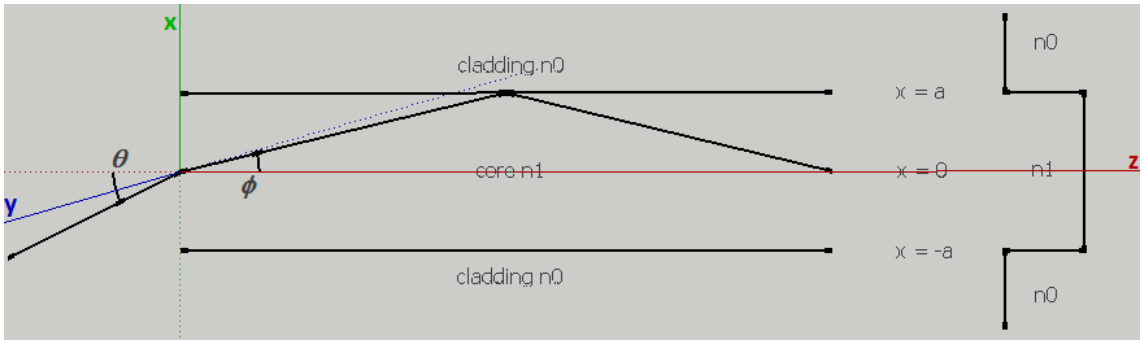

Figure E65 - Total internal reflection.

Since the angle ϕ relates to incident angle θ by:

$$\sin\theta = n_1 \sin\phi \leq \sqrt{n_1^2 - n_0^2}$$

the critical condition for total internal reflection is:

$$\boxed{\theta \leq sin^{-1}\sqrt{n_1^2 - n_0^2} \equiv \theta_{max}}$$

Considering the refractive index difference between core and cladding $n_1 - n_0 \leq 0.01$, one can approximately say that:

$$\theta_{max} \cong \sqrt{n_1^2 - n_0^2}$$

meaning that, θ_{max} is the waveguide's maximum acceptance angle, also known as NA (Numerical Aperture). Another important definition is the relative refractive index between n_1 and n_0 or contrast:

$$\Delta = \frac{n_1^2 - n_0^2}{2n_1^2} \cong \frac{n_1 - n_0}{n_1} \tag{42}$$

Contrast (Δ) is usually expressed as a percentage and it is related to numerical aperture NA by:

$$NA = \theta_{max} \cong n_1\sqrt{2\Delta} \tag{43}$$

The maximum angle for propagating light within the core is:

$$\phi_{max} \cong {}^{\theta_{max}}/_{n_1} \cong \sqrt{2\Delta}$$

Guided modes formation

In order to have a propagating mode confined to the core, its angle ϕ must be smaller than the critical angle ϕ_{max}. This does not mean that all modes with an angle $\phi < \phi_{max}$ will be able to propagate. Each mode is associated with light rays at a discrete angle of propagation, as given by electromagnetic wave analysis.

Let us consider Figure E66, where wavelength and the wavenumber of light in the core are ${}^{\lambda}/_{n_1}$and kn_1 $\left(k = {}^{2\pi}/_{\lambda}\right)$, respectively, and λ is the wavelength of light in vacuum. Propagation constants along axis z and x are:

$$\beta = kn_1 \cos\phi \tag{44}$$

$$\kappa = kn_1 \sin\phi$$

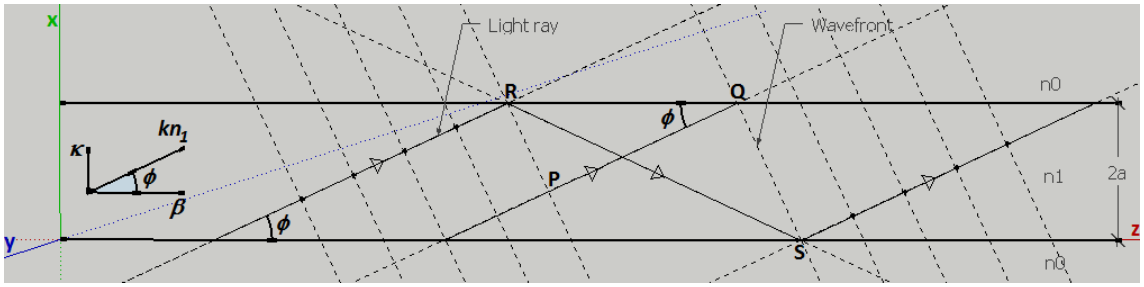

Figure E66 - Confined light rays and their phase fronts.

Before going into details about how modes are possible, it must be explained first the phase shift suffered by a light ray when on total internal reflection. The reflection coefficient of total reflected light (polarized perpendicularly to the plane of incidence, meaning by plane of incidence the plane formed by the incident and reflected rays), naming the reflected and incident rays A_r and A_i, respectively, is given by:

$$r = \frac{A_r}{A_i} = \frac{n_1 \sin\phi + j\sqrt{n_1^2 \cos^2\phi - n_0^2}}{n_1 \sin\phi - j\sqrt{n_1^2 \cos^2\phi - n_0^2}}$$

Expressing the complex reflection coefficient as $r = e^{-j\varphi}$, the phase shift φ is (utilizing equation 42):

$$\varphi = -2\tan^{-1}\left(\frac{\sqrt{n_1^2 \cos^2\phi - n_0^2}}{n_1 \sin\phi}\right) = -2\tan^{-1}\left(\sqrt{\frac{2\Delta}{\sin^2\phi} - 1}\right) \tag{45}$$

this phase shift is known as the Goos-Hänchen shift.

Now, going back to Figure E66, let us consider the phase difference between light rays that travel from P to Q and from R to S. Light ray PQ has not suffered any reflection influence yet,

while light ray RS has been reflected twice before (first on lower then on upper core-cladding interfaces). Since points P and R or Q and S are in the same phase front, the optical paths PQ and RS (including the Goos-Hänchen shifts introduced by the two total reflections) should be equal or their difference must be an integral multiple of 2π.

The distance between points Q and R is,

$$QR = \frac{2a}{\tan\phi} - 2a\tan\phi$$

the distance between points P and Q can be expressed by,

$$PQ = \ell_1 = QR\cos\phi = \left(\frac{2a}{\tan\phi} - 2a\tan\phi\right)\cos\phi = 2a\left(\frac{1}{\sin\phi} - \sin\phi\right) \qquad (46)$$

the distance between points R and S is,

$$RS = \ell_2 = \frac{2a}{\sin\phi} \qquad (47)$$

So, phase matching condition for optical paths PQ and RS is:

$$(kn_1\ell_2 + 2\varphi) - kn_1\ell_1 = 2m\pi \qquad m = 0, 1, 2, \ldots \qquad (48)$$

Substituting equations 45, 46 and 47 into equation 48, it is obtained the condition for propagating angle ϕ,

$$\tan\left(kn_1 a\sin\phi - \frac{m\pi}{2}\right) = \sqrt{\frac{2\Delta}{\sin^2\phi} - 1} \qquad (49)$$

Equation 49 shows that the propagation angle is discrete and is determined by the waveguide structure (core half width a, refractive index n_1 and contrast Δ) and the light source's wavelength $\left(k = {}^{2\pi}/_{\lambda}\right)$. The optical field that satisfies equation 49 is designated the *mode*. The value of propagation constant β given by equation 44 is also discrete and is denoted as an eigenvalue. The mode that has the smallest angle ϕ in equation 49 $(m = 0)$ is the fundamental mode. All other larger angle modes are higher order modes $(m \geq 1)$.

On Figure E67 and Figure E68, one can find diagrams that represent the formation of standing waves for the first and higher order modes caused by phase fronts interference. On these diagrams, the solid black line represents a positive phase front while a dotted black line represents a negative phase front. The electric field is maximum or minimum where two positive or negative phase fronts interfere, respectively. On the other hand, the electric field tends to zero at the core-cladding interface or wherever positive and negative phase fronts tend to cancel each other. Thus, the electric field distribution on transverse direction (x-axis) is a standing wave that varies periodically along the z-axis with period,

$$\lambda_p = {\left(\lambda/n_1\right)}/{\cos\phi} = {2\pi}/{\beta}$$

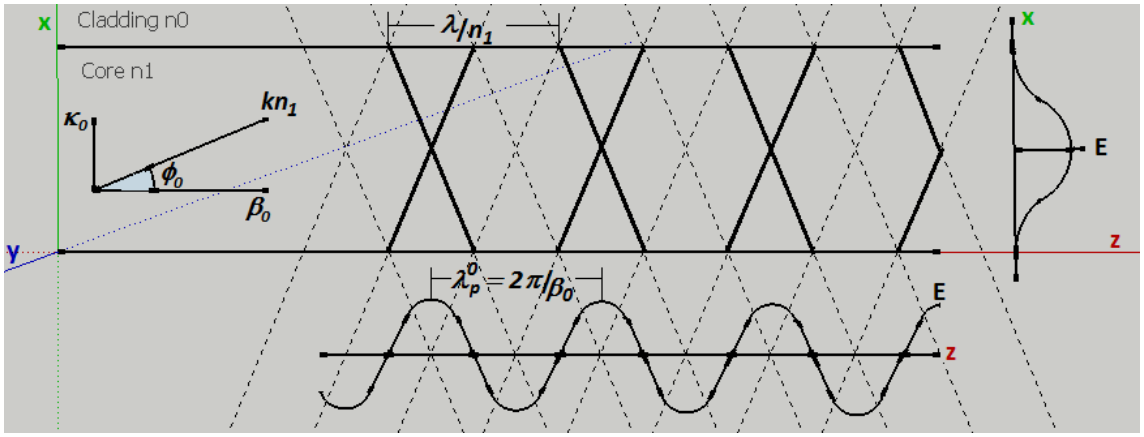

Figure E67 - Fundamental mode (m=0).

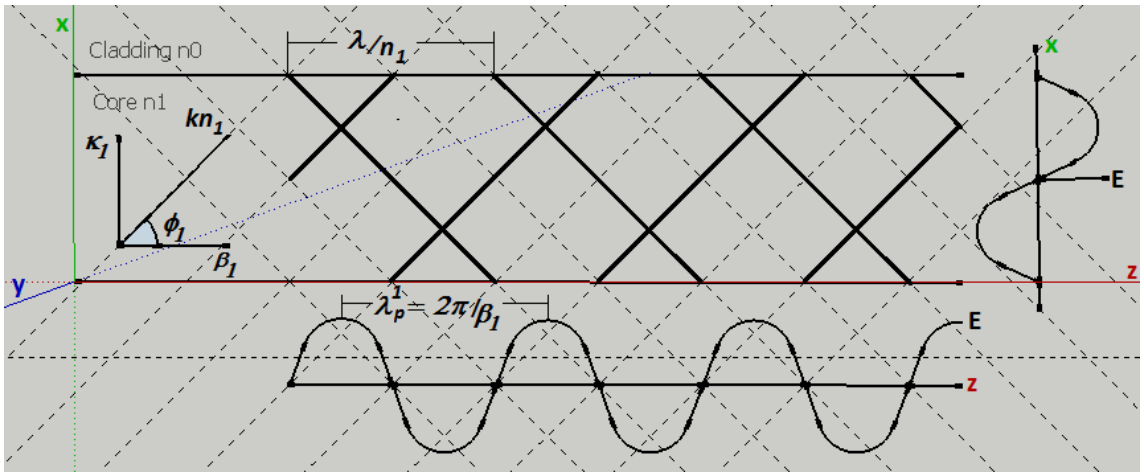

Figure E68 - Higher order mode (m = 1).

Referring to Figure E65 and knowing that,

$$\sin\theta = n_1 \sin\phi \leq \sqrt{n_1^2 - n_0^2}$$

it can be written, using equation 43,

$$n_1 \sin\phi \leq n_1\sqrt{2\Delta} \leftrightarrow \sin\phi \leq \sqrt{2\Delta}$$

Like so, one can introduce the normalized to unity parameter:

$$\xi = \frac{\sin\phi}{\sqrt{2\Delta}}$$

and rewrite equation 49 (propagating angle condition equation) as,

$$kn_1\sqrt{2\Delta} = \frac{\cos^{-1}\xi + {m\pi}/{2}}{\xi}$$

The left hand side term of previous equation is known as the normalized frequency v and it allows the treatment of any waveguide independently of its structure. The normalized frequency designation is thus:

$$v = kn_1\sqrt{2\Delta}$$

The relationship between normalized frequency v and ξ (which is closely related to β) is designated the *dispersion equation*. Figure E69 shows this relationship for modes $m = 0\ to\ 5$ and, as can be observed, every intersection between the value of normalized frequency and the dispersion curve corresponds to a possible propagating mode. One can also notice that when $v \leq {}^{\pi}/_{2}$, only $m = 0$ mode is allowed to propagate. Thus, $v = {}^{\pi}/_{2}$ is designated v_c (normalized frequency cut off) for only one mode is able to propagate.

Very often, it is necessary to find out which is the cut off wavelength λ_c for a certain waveguide. This is accomplished by the following expression:

$$\lambda_c = \frac{2\pi}{v_c} an_1\sqrt{2\Delta} \quad , \qquad where\ v_c = \frac{\pi}{2}$$

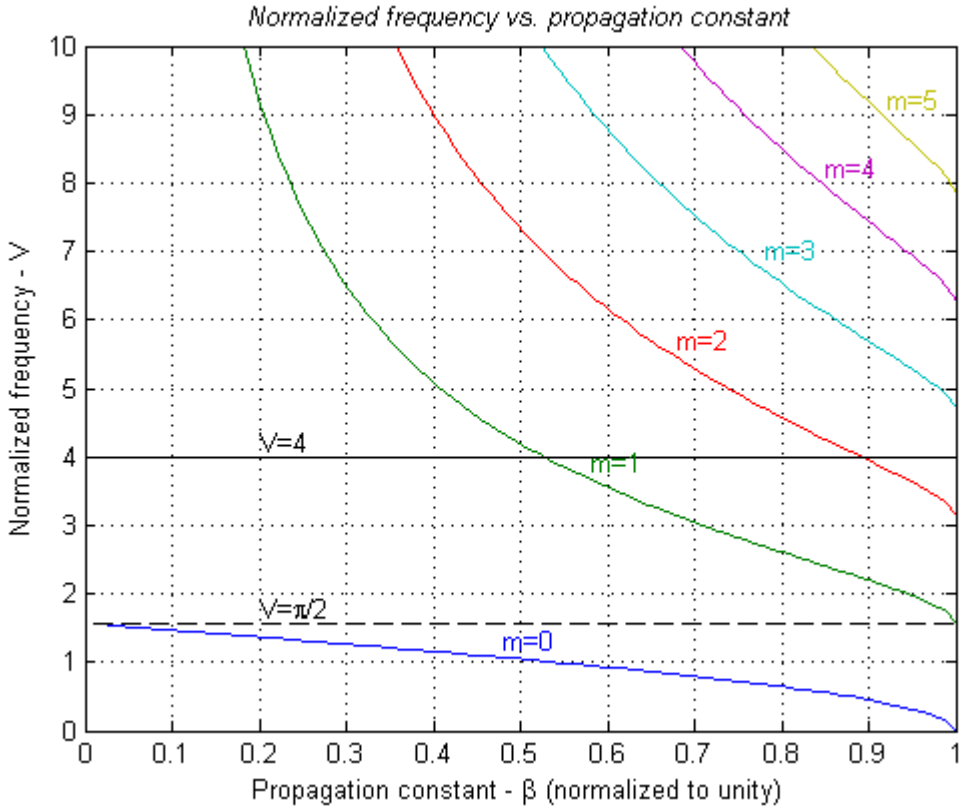

Figure E69 - Normalized frequency Vs. normalized propagation constant.

Dispersion equations code

```
%% Script para plot da equação de dispersão (V vs. beta)
beta=linspace(0,1);
n=5;
V=zeros(n+1,numel(beta));
for m=0:n
V(m+1,:)=acos(beta)+m*pi/2./beta;
```

```
end
figure()
plot(beta,V)
grid on
axis([0 1 0 10])
line([0 1],[pi/2 pi/2],'LineStyle','--','Color','k')
line([0 1],[4 4],'Color','k')
title('Normalized frequency vs. propagation
constant','FontAngle','italic')
text(0.2,1.8,'V=\pi/2','color',[0 0 0])
text(0.2,4.2,'V=4','color',[0 0 0])
text(0.48,1.3,'m=0','color',[0 0 1])
text(0.58,3.9,'m=1','color',[0 0.5 0])
text(0.68,5.7,'m=2','color',[1 0 0])
text(0.78,7,'m=3','color',[0 0.75 0.75])
text(0.85,8.2,'m=4','color',[0.75 0 0.75])
text(0.92,9.2,'m=5','color',[0.75 0.75 0])
xlabel('Propagation constant - \beta (normalized to unity)')
ylabel('Normalized frequency - V')
```

Appendix F

E^x and E^y modes plot code

```
% This example shows how to find the modes of a birefringent
% (uniaxial) waveguide. This waveguide is the same as the one
% considered in 'uniaxial_channel.m', except that the c-axis
% is now rotated by an angle of pi/2 relative to the x axis.
n1 = 1.55;
n2x = 2.156;
n2y = 2.232;
n2z = 2.232;
theta = pi/2;
e2xx = n2x^2*cos(theta)^2 + n2y^2*sin(theta)^2;
e2yy = n2y^2*cos(theta)^2 + n2x^2*sin(theta)^2;
e2xy = cos(theta)*sin(theta)*(n2x^2-n2y^2);
e2yx = e2xy;
Rx = 0.50;
Ry = 0.50;
side = 0.2;
dx = 0.0025; % grid size (x)
dy = dx; % grid size (y)
lambda = 1.00; % wavelength
nmodes = 6; % number of modes to compute
[x,y,xc,yc,nx,ny,epsxx,edges] = ...
waveguidemeshfull([n1,sqrt(e2xx),n1],[side,2*Ry,side],2*Ry,Rx, ...
side,dx,dy);
[x,y,xc,yc,nx,ny,epsxy,edges] = ...
waveguidemeshfull([0,sqrt(e2xy),0],[side,2*Ry,side],2*Ry,Rx, ...
side,dx,dy);
[x,y,xc,yc,nx,ny,epsyx,edges] = ...
waveguidemeshfull([0,sqrt(e2yx),0],[side,2*Ry,side],2*Ry,Rx, ...
side,dx,dy);
[x,y,xc,yc,nx,ny,epsyy,edges] = ...
waveguidemeshfull([n1,sqrt(e2yy),n1],[side,2*Ry,side],2*Ry,Rx, ...
side,dx,dy);
[x,y,xc,yc,nx,ny,epszz,edges] = ...
waveguidemeshfull([n1,n2z,n1],[side,2*Ry,side],2*Ry,Rx, ...
side,dx,dy);
% Now we stretch out the mesh at the boundaries:
[x,y,xc,yc,dx,dy] = stretchmesh(x,y,[80,80,80,80],[4,4,4,4]);
[Hx,Hy,neff] = wgmodes (lambda, n2y, nmodes, dx, dy, epsxx, epsxy,
epsyx, epsyy, epszz,
'0000');
fprintf(1,'neff = %7.5f\n',neff);
for n=1:nmodes
[Hz(:,:,n),Ex(:,:,n),Ey(:,:,n),Ez(:,:,n)] = ...
postprocess (lambda, neff(n), Hx(:,:,n), Hy(:,:,n), dx, dy, epsxx,
epsxy, epsyx,
epsyy, epszz, '0000');
end
subplot(3,2,1);
contourmode(x,y,Ex(:,:,1));
title(sprintf('Ex (mode %d)',11));
for v = edges, line(v{:}); end
subplot(3,2,2);
contourmode(x,y,Ey(:,:,2));
title(sprintf('Ey (mode %d)',11));
for v = edges, line(v{:}); end
subplot(3,2,3);
contourmode(x,y,Ex(:,:,3));
```

```
title(sprintf('Ex (mode %d)',12));
for v = edges, line(v{:}); end
subplot(3,2,4);
contourmode(x,y,Ey(:,:,6));
title(sprintf('Ey (mode %d)',12));
for v = edges, line(v{:}); end
subplot(3,2,5);
contourmode(x,y,Ex(:,:,4));
title(sprintf('Ex (mode %d)',21));
for v = edges, line(v{:}); end
subplot(3,2,6);
contourmode(x,y,Ey(:,:,5));
title(sprintf('Ey (mode %d)',21));
for v = edges, line(v{:}); end
```

Contourmode function code

```
function contourmode(x,y,mode,dB,xyrange);
% Produces a contour plot (in dB) of one field component of the
% mode of an optical waveguide.
%
% USAGE:
%
% contourmode(x,y,mode);
% contourmode(x,y,mode,dBrange);
% contourmode(x,y,mode,dBrange,xyrange);
%
% INPUT:
%
% x,y - vectors describing horizontal and vertical grid points
% mode - the mode or field component to be plotted
% dBrange - contour levels to plot (in dB), with 0 dB corresponding
% to the level |mode| = 1. default = (0:-3:-45)
% xyrange - axis range to use (optional)
%
% EXAMPLE: Make a contour plot of the magnetic field component Hx,
% with contours from 0 dB down to -50 dB, relative to the maximum
% value, in 5 dB increments.
%
% contourmode(x,y,Hx/max(abs(Hx(:))),(0:-5:-50));
%
% NOTES:
%
% (1) This function uses the current color map to determine the
% colors of each contour, with 0 dB corresponding to the
% maximum color and -dbmax corresponding to the minimum color.
% You can use the 'colormap' command to change the current
% color map.
% (2) The aspect ratio of the plot box is automatically adjusted so
% that the horizontal and vertical scales are equal.
% (3) The mode is not normalized or scaled in any way.
x = real(x);
y = real(y);
if (nargin < 5)
xyrange = [min(x),max(x),min(y),max(y)];
end
if (size(mode) == [length(x)-1,length(y)-1])
x = (x(1:end-1) + x(2:end))/2;
y = (y(1:end-1) + y(2:end))/2;
end
if (nargin < 4) || isempty(dB)
dB = (0:-3:-45);
```

```
end
% Compute and plot contours
c = contourc(x,y,20*log10(abs(transpose(mode))),dB);
cmap = colormap;
ii = 1;
cla;
while (ii < length(c)),
level = c(1,ii);
n = c(2,ii);
jj = 1+round((length(cmap)-1)*(level - min(dB))/(max(dB)-min(dB)));
color = cmap(jj,:);
line(c(1,ii+1:ii+n),c(2,ii+1:ii+n),'Color',color);
ii = ii+n+1;
end
axis(xyrange);
set(gca,'PlotBoxAspectRatio',[xyrange(2)-xyrange(1) xyrange(4)-
xyrange(3) 1],...
'Box','on');
```

Postprocess function code

```
function [Hz,Ex,Ey,Ez] = postprocess
(lambda,neff,Hx,Hy,dx,dy,varargin);
% This function takes the two computed transverse magnetic
% fields (Hx and Hy) of an optical waveguide structure and
% solves for the remaining 4 vield components: Hz, Ex, Ey,
% and Ez.
%
% USAGE:
%
% [Hz,Ex,Ey,Ez] = postprocess(lambda, neff, Hx, Hy, dx, dy, ...
% eps, boundary);
% [Hz,Ex,Ey,Ez] = postprocess(lambda, neff, Hx, Hy, dx, dy, ...
% epsxx, epsyy, epszz, boundary);
% [Hz,Ex,Ey,Ez] = postprocess(lambda, neff, Hx, Hy, dx, dy, ...
% epsxx, epsxy, epsyx, epsyy, epszz, boundary);
%
% INPUT:
%
% lambda - optical wavelength at which mode was calculated
% neff - the calculated effective index of the optial mode
% Hx, Hy - the calculated transverse magnetic fields of the mode
% dx - horizontal grid spacing (vector or scalar)
% dy - vertical grid spacing (vector or scalar)
% eps - index mesh (isotropic materials) OR:
% epsxx, epsxy, epsyx, epsyy, epszz - index mesh (anisotropic)
% boundary - 4 letter string specifying boundary conditions to be
% applied at the edges of the computation window.
% boundary(1) = North boundary condition
% boundary(2) = South boundary condition
% boundary(3) = East boundary condition
% boundary(4) = West boundary condition
% The following boundary conditions are supported:
% 'A' - Hx is antisymmetric, Hy is symmetric.
% 'S' - Hx is symmetric and, Hy is antisymmetric.
% '0' - Hx and Hy are zero immediately outside of the
% boundary.
%
% OUTPUT:
%
% Hz - calculated longitudinal magnetic field. This output will
% have the same dimensions as Hx and Hy.
```

```
% Ex, Ey, Ez - calculated electric field. These field components
% are computed at the center of each element instead of on the
% edges or vertices.
%
% NOTES:
%
% 1) This routine is meant to be used in conjunction with
% wgmodes.m, the vector eigenmode solver. Please consult the
% help file for wgmodes.m for more information.
%
% 2) The boundary conditions and waveguide specifications
% (given in dx, dy, eps, and boundary) should be the same as
% what was used in wgmodes.m to compute the mode.
%
% 3) The magnetic field components (Hx, Hy, and Hz) are
% calculated at the edges of each cell, whereas the electric
% field components are computed at the center of each cell.
% Therefore if size(eps) = [n,m], then the magnetic fields
% will have a size of [n+1,m+1] while the computed electric
% fields will have a size of [n,m].
%
% 4) Even though wgmodes.m will optionally calculate more than
% one mode at a time, this postprocessing routine must be
% invoked separately for each computed mode.
%
% AUTHORS: Thomas E. Murphy (tem@umd.edu)
if (nargin == 12)
epsxx = varargin{1};
epsxy = varargin{2};
epsyx = varargin{3};
epsyy = varargin{4};
epszz = varargin{5};
boundary = varargin{6};
elseif (nargin == 10)
epsxx = varargin{1};
epsxy = zeros(size(epsxx));
epsyx = zeros(size(epsxx));
epsyy = varargin{2};
epszz = varargin{3};
boundary = varargin{4};
elseif (nargin == 8)
epsxx = varargin{1};
epsxy = zeros(size(epsxx));
epsyx = zeros(size(epsxx));
epsyy = epsxx;
epszz = epsxx;
boundary = varargin{2};
else
error('Incorrect number of input arguments.\n');
end
[nx,ny] = size(epsxx);
nx = nx + 1;
ny = ny + 1;
% now we pad eps on all sides by one grid point
epsxx = [epsxx(:,1),epsxx,epsxx(:,ny-1)];
epsxx = [epsxx(1,1:ny+1);epsxx;epsxx(nx-1,1:ny+1)];
epsyy = [epsyy(:,1),epsyy,epsyy(:,ny-1)];
epsyy = [epsyy(1,1:ny+1);epsyy;epsyy(nx-1,1:ny+1)];
epsxy = [epsxy(:,1),epsxy,epsxy(:,ny-1)];
epsxy = [epsxy(1,1:ny+1);epsxy;epsxy(nx-1,1:ny+1)];
epsyx = [epsyx(:,1),epsyx,epsyx(:,ny-1)];
```

```
epsyx = [epsyx(1,1:ny+1);epsyx;epsyx(nx-1,1:ny+1)];
epszz = [epszz(:,1),epszz,epszz(:,ny-1)];
epszz = [epszz(1,1:ny+1);epszz;epszz(nx-1,1:ny+1)];
k = 2*pi/lambda; % free-space wavevector
b = neff*k; % propagation constant (eigenvalue)
if isscalar(dx)
dx = dx*ones(nx+1,1); % uniform grid
else
dx = dx(:); % convert to column vector
dx = [dx(1);dx;dx(length(dx))]; % pad dx on top and bottom
end
if isscalar(dy)
dy = dy*ones(1,ny+1); % uniform grid
else
dy = dy(:); % convert to column vector
dy = [dy(1);dy;dy(length(dy))]'; % pad dy on top and bottom
end
% distance to neighboring points to north south east and west,
% relative to point under consideration (P), as shown below.
n = ones(1,nx*ny); n(:) = ones(nx,1)*dy(2:ny+1);
s = ones(1,nx*ny); s(:) = ones(nx,1)*dy(1:ny);
e = ones(1,nx*ny); e(:) = dx(2:nx+1)*ones(1,ny);
w = ones(1,nx*ny); w(:) = dx(1:nx)*ones(1,ny);
% epsilon tensor elements in regions 1,2,3,4, relative to the
% mesh point under consideration (P), as shown below.
%
% NW------N------NE
% | | |
% | 1 n 4 |
% | | |
% W---w---P---e---E
% | | |
% | 2 s 3 |
% | | |
% SW------S------SE
exx1 = ones(1,nx*ny); exx1(:) = epsxx(1:nx,2:ny+1);
exx2 = ones(1,nx*ny); exx2(:) = epsxx(1:nx,1:ny);
exx3 = ones(1,nx*ny); exx3(:) = epsxx(2:nx+1,1:ny);
exx4 = ones(1,nx*ny); exx4(:) = epsxx(2:nx+1,2:ny+1);
eyy1 = ones(1,nx*ny); eyy1(:) = epsyy(1:nx,2:ny+1);
eyy2 = ones(1,nx*ny); eyy2(:) = epsyy(1:nx,1:ny);
eyy3 = ones(1,nx*ny); eyy3(:) = epsyy(2:nx+1,1:ny);
eyy4 = ones(1,nx*ny); eyy4(:) = epsyy(2:nx+1,2:ny+1);
exy1 = ones(1,nx*ny); exy1(:) = epsxy(1:nx,2:ny+1);
exy2 = ones(1,nx*ny); exy2(:) = epsxy(1:nx,1:ny);
exy3 = ones(1,nx*ny); exy3(:) = epsxy(2:nx+1,1:ny);
exy4 = ones(1,nx*ny); exy4(:) = epsxy(2:nx+1,2:ny+1);
eyx1 = ones(1,nx*ny); eyx1(:) = epsyx(1:nx,2:ny+1);
eyx2 = ones(1,nx*ny); eyx2(:) = epsyx(1:nx,1:ny);
eyx3 = ones(1,nx*ny); eyx3(:) = epsyx(2:nx+1,1:ny);
eyx4 = ones(1,nx*ny); eyx4(:) = epsyx(2:nx+1,2:ny+1);
ezz1 = ones(1,nx*ny); ezz1(:) = epszz(1:nx,2:ny+1);
ezz2 = ones(1,nx*ny); ezz2(:) = epszz(1:nx,1:ny);
ezz3 = ones(1,nx*ny); ezz3(:) = epszz(2:nx+1,1:ny);
ezz4 = ones(1,nx*ny); ezz4(:) = epszz(2:nx+1,2:ny+1);
bzxne =
(1./2.*(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*eyx4./ezz4./(n.*eyy3
+s.*eyy4)./ezz2./ezz1./(
n.*eyy2+s.*eyy1)./(e+w).*eyy3.*eyy1.*w.*eyy2+1./2.*(ezz3./exx2.*ezz2.*
w+ezz2./exx3.*ezz3.*e).*
```

```
(1-
exx4./ezz4)./ezz3./ezz2./(w.*exx3+e.*exx2)./(w.*exx4+e.*exx1)./(n+s).*
exx2.*exx3.*exx1.*s).
/b;
bzxse =
(-
1./2.*(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*eyx3./ezz3./(n.*eyy3+
s.*eyy4)./ezz2./ezz1./
(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy1.*w.*eyy2+1./2.*(ezz4./exx1.*ezz1.
*w+ezz1./exx4.*ezz4.*e).
*(1-
exx3./ezz3)./(w.*exx3+e.*exx2)./ezz4./ezz1./(w.*exx4+e.*exx1)./(n+s).*
exx2.*n.*exx1.*exx4)
./b;
bzxnw =
(-1./2.*(-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*eyx1./ezz4./ezz3./(n.*eyy3+s.*eyy4)./ezz1.
/(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*eyy2.*e-
1./2.*(ezz3./exx2.*ezz2.*w+ezz2./exx3.*ezz3.*e)
.*(1-
exx1./ezz1)./ezz3./ezz2./(w.*exx3+e.*exx2)./(w.*exx4+e.*exx1)./(n+s).*
exx2.*exx3.*exx4.*s
)./b;
bzxsw =
(1./2.*(-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*eyx2./ezz4./ezz3./(n.*eyy3+s.*eyy4)./ezz2./
(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*eyy1.*e-
1./2.*(ezz4./exx1.*ezz1.*w+ezz1./exx4.*ezz4.*e).
*(1-
exx2./ezz2)./(w.*exx3+e.*exx2)./ezz4./ezz1./(w.*exx4+e.*exx1)./(n+s).*
exx3.*n.*exx1.*exx4)
./b;
bzxn =
((1./2.*(-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*n.*ezz1.*ezz2./eyy1.*(2.*eyy1./ezz1./n.^2+
eyx1./ezz1./n./w)+1./2.*(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*n.*
ezz4.*ezz3./eyy4.*(2.*ey
y4./ezz4./n.^2-
eyx4./ezz4./n./e))./ezz4./ezz3./(n.*eyy3+s.*eyy4)./ezz2./ezz1./(n.*eyy
2+s.*eyy1
)./(e+w).*eyy4.*eyy3.*eyy1.*w.*eyy2.*e+((ezz3./exx2.*ezz2.*w+ezz2./exx
3.*ezz3.*e).*(1./2.*ezz4
.*((1-exx1./ezz1)./n./w-exy1./ezz1.*(2./n.^2-
2./n.^2.*s./(n+s)))./exx1.*ezz1.*w+(ezz4-ezz1).*s
./n./(n+s)+1./2.*ezz1.*(-(1-exx4./ezz4)./n./e-exy4./ezz4.*(2./n.^2-
2./n.^2.*s./(n+s)))./exx4.*
ezz4.*e)-(ezz4./exx1.*ezz1.*w+ezz1./exx4.*ezz4.*e).*(-
ezz3.*exy2./n./(n+s)./exx2.*w+(ezz3-ezz2
).*s./n./(n+s)-
ezz2.*exy3./n./(n+s)./exx3.*e))./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz4./
ezz1./(w.
*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzxs
=((1./2.*(-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*s.*ezz2.*ezz1./eyy2.*(2.*eyy2./ezz2./s.^2
-
eyx2./ezz2./s./w)+1./2.*(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*s.*
ezz3.*ezz4./eyy3.*(2.*e
yy3./ezz3./s.^2+eyx3./ezz3./s./e))./ezz4./ezz3./(n.*eyy3+s.*eyy4)./ezz
2./ezz1./(n.*eyy2+s.*eyy
```

```
1)./(e+w).*eyy4.*eyy3.*eyy1.*w.*eyy2.*e+((ezz3./exx2.*ezz2.*w+ezz2./ex
x3.*ezz3.*e).*(-ezz4.*ex
y1./s./(n+s)./exx1.*w-(ezz4-ezz1).*n./s./(n+s)-
ezz1.*exy4./s./(n+s)./exx4.*e)-(ezz4./exx1.*ezz
1.*w+ezz1./exx4.*ezz4.*e).*(1./2.*ezz3.*(-(1-exx2./ezz2)./s./w-
exy2./ezz2.*(2./s.^2-2./s.^2.*n
./(n+s))).*exx2.*ezz2.*w-(ezz3-ezz2).*n./s./(n+s)+1./2.*ezz2.*((1-
exx3./ezz3)./s./e-exy3./ezz3
.*(2./s.^2-
2./s.^2.*n./(n+s))).*exx3.*ezz3.*e)).*ezz3./ezz2./(w.*exx3+e.*exx2).*e
zz4./ezz1./(w
.*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzxe =
((n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*(1./2.*n.*ezz4.*ezz3./eyy4
.*(2./e.^2-eyx4./ezz4./n
./e)+1./2.*s.*ezz3.*ezz4./eyy3.*(2./e.^2+eyx3./ezz3./s./e)).*ezz4./ezz
3./(n.*eyy3+s.*eyy4)./ez
z2./ezz1./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*eyy1.*w.*eyy2.*e+(-
1./2.*(ezz3./exx2.*ezz2.*w+
ezz2./exx3.*ezz3.*e).*ezz1.*(1-exx4./ezz4)./n./exx4.*ezz4-
1./2.*(ezz4./exx1.*ezz1.*w+ezz1./exx
4.*ezz4.*e).*ezz2.*(1-
exx3./ezz3)./s./exx3.*ezz3)./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz4./ezz1
./
 (w.*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzxw =
((-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*(1./2.*n.*ezz1.*ezz2./eyy1.*(2./w.^2+eyx1./ezz1.
/
n./w)+1./2.*s.*ezz2.*ezz1./eyy2.*(2./w.^2-
eyx2./ezz2./s./w)).*ezz4./ezz3./(n.*eyy3+s.*eyy4)./e
zz2./ezz1./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*eyy1.*w.*eyy2.*e+(1./
2.*(ezz3./exx2.*ezz2.*w+
ezz2./exx3.*ezz3.*e).*ezz4.*(1-
exx1./ezz1)./n./exx1.*ezz1+1./2.*(ezz4./exx1.*ezz1.*w+ezz1./exx
4.*ezz4.*e).*ezz3.*(1-
exx2./ezz2)./s./exx2.*ezz2)./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz4./ezz1
./
(w.*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzxp =
(((-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*(1./2.*n.*ezz1.*ezz2./eyy1.*(-2./w.^2-2.*eyy1./e
zz1./n.^2+k.^2.*eyy1-eyx1./ezz1./n./w)+1./2.*s.*ezz2.*ezz1./eyy2.*(-
2./w.^2-2.*eyy2./ezz2./s.^
2+k.^2.*eyy2+eyx2./ezz2./s./w))+(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./ey
y2).*(1./2.*n.*ezz4.*ezz
3./eyy4.*(-2./e.^2-
2.*eyy4./ezz4./n.^2+k.^2.*eyy4+eyx4./ezz4./n./e)+1./2.*s.*ezz3.*ezz4./
eyy3.
*(-2./e.^2-2.*eyy3./ezz3./s.^2+k.^2.*eyy3-
eyx3./ezz3./s./e))./ezz4./ezz3./(n.*eyy3+s.*eyy4)./
ezz2./ezz1./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*eyy1.*w.*eyy2.*e+((e
zz3./exx2.*ezz2.*w+ezz2.
/exx3.*ezz3.*e).*(1./2.*ezz4.*(-k.^2.*exy1-(1-exx1./ezz1)./n./w-
exy1./ezz1.*(-2./n.^2-2./n.^2.
*(n-s)./s))./exx1.*ezz1.*w+(ezz4-ezz1).*(n-s)./n./s+1./2.*ezz1.*(-
k.^2.*exy4+(1-exx4./ezz4)./n
./e-exy4./ezz4.*(-2./n.^2-2./n.^2.*(n-s)./s))./exx4.*ezz4.*e)-
(ezz4./exx1.*ezz1.*w+ezz1./exx4.
*ezz4.*e).*(1./2.*ezz3.*(-k.^2.*exy2+(1-exx2./ezz2)./s./w-
exy2./ezz2.*(-2./s.^2+2./s.^2.*(n-s)
```

```
./n))./exx2.*ezz2.*w+(ezz3-ezz2).*(n-s)./n./s+1./2.*ezz2.*(-
k.^2.*exy3-(1-exx3./ezz3)./s./e-ex
y3./ezz3.*(-2./s.^2+2./s.^2.*(n-
s)./n))./exx3.*ezz3.*e))./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz4.
/ezz1./(w.*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzyne =
(1./2.*(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*(1-
eyy4./ezz4)./(n.*eyy3+s.*eyy4)./ezz2./ezz
1./(n.*eyy2+s.*eyy1)./(e+w).*eyy3.*eyy1.*w.*eyy2+1./2.*(ezz3./exx2.*ez
z2.*w+ezz2./exx3.*ezz3.*
e).*exy4./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz4./(w.*exx4+e.*exx1)./(n+s
).*exx2.*exx3.*exx1.*s).
/b;
bzyse =
(-1./2.*(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*(1-
eyy3./ezz3)./(n.*eyy3+s.*eyy4)./ezz2./ez
z1./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy1.*w.*eyy2+1./2.*(ezz4./exx1.*e
zz1.*w+ezz1./exx4.*ezz4.
*e).*exy3./ezz3./(w.*exx3+e.*exx2)./ezz4./ezz1./(w.*exx4+e.*exx1)./(n+
s).*exx2.*n.*exx1.*exx4)
./b;
bzynw =
(-1./2.*(-n.*ezz4.*ezz3./eyy4-s.*ezz3.*ezz4./eyy3).*(1-
eyy1./ezz1)./ezz4./ezz3./(n.*eyy3+s.*ey
y4)./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*eyy2.*e-
1./2.*(ezz3./exx2.*ezz2.*w+ezz2./exx3.*ezz3
.*e).*exy1./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz1./(w.*exx4+e.*exx1)./(n
+s).*exx2.*exx3.*exx4.*s
)./b;
bzysw =
(1./2.*(-n.*ezz4.*ezz3./eyy4-s.*ezz3.*ezz4./eyy3).*(1-
eyy2./ezz2)./ezz4./ezz3./(n.*eyy3+s.*eyy
4)./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*eyy1.*e-
1./2.*(ezz4./exx1.*ezz1.*w+ezz1./exx4.*ezz4.
*e).*exy2./ezz2./(w.*exx3+e.*exx2)./ezz4./ezz1./(w.*exx4+e.*exx1)./(n+
s).*exx3.*n.*exx1.*exx4)
./b;
bzyn =
((1./2.*(-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*ezz1.*ezz2./eyy1.*(1-eyy1./ezz1)./w-1./2.*
(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*ezz4.*ezz3./eyy4.*(1-
eyy4./ezz4)./e)./ezz4./ezz3./(
n.*eyy3+s.*eyy4)./ezz2./ezz1./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*ey
y1.*w.*eyy2.*e+(ezz3./ex
x2.*ezz2.*w+ezz2./exx3.*ezz3.*e).*(1./2.*ezz4.*(2./n.^2+exy1./ezz1./n.
/w)./exx1.*ezz1.*w+1./2.
*ezz1.*(2./n.^2-
exy4./ezz4./n./e)./exx4.*ezz4.*e)./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz4
./ezz1./
(w.*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzys =
((-1./2.*(-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*ezz2.*ezz1./eyy2.*(1-eyy2./ezz2)./w+1./2.
*(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*ezz3.*ezz4./eyy3.*(1-
eyy3./ezz3)./e)./ezz4./ezz3./
(n.*eyy3+s.*eyy4)./ezz2./ezz1./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*eyy3.*e
yy1.*w.*eyy2.*e-(ezz4./e
xx1.*ezz1.*w+ezz1./exx4.*ezz4.*e).*(1./2.*ezz3.*(2./s.^2-
exy2./ezz2./s./w)./exx2.*ezz2.*w+1./2
.*ezz2.*(2./s.^2+exy3./ezz3./s./e)./exx3.*ezz3.*e)./ezz3./ezz2./(w.*ex
x3+e.*exx2)./ezz4./ezz1.
```

```
/(w.*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzye =
(((-n.*ezz4.*ezz3./eyy4-s.*ezz3.*ezz4./eyy3).*(-
n.*ezz2./eyy1.*eyx1./e./(e+w)+(ezz1-ezz2).*w./
e./(e+w)-
s.*ezz1./eyy2.*eyx2./e./(e+w))+(n.*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy
2).*(1./2.*n.*e
zz4.*ezz3./eyy4.*(-(1-eyy4./ezz4)./n./e-eyx4./ezz4.*(2./e.^2-
2./e.^2.*w./(e+w)))+1./2.*s.*ezz3
.*ezz4./eyy3.*((1-eyy3./ezz3)./s./e-eyx3./ezz3.*(2./e.^2-
2./e.^2.*w./(e+w)))+(ezz4-ezz3).*w./e
./(e+w))).ezz4./ezz3./(n.*eyy3+s.*eyy4)./ezz2./ezz1./(n.*eyy2+s.*eyy1
)./(e+w).*eyy4.*eyy3.*ey
y1.*w.*eyy2.*e+(1./2.*(ezz3./exx2.*ezz2.*w+ezz2./exx3.*ezz3.*e).*ezz1.
*(2.*exx4./ezz4./e.^2-ex
y4./ezz4./n./e)./exx4.*ezz4.*e-
1./2.*(ezz4./exx1.*ezz1.*w+ezz1./exx4.*ezz4.*e).*ezz2.*(2.*exx3
./ezz3./e.^2+exy3./ezz3./s./e)./exx3.*ezz3.*e)./ezz3./ezz2./(w.*exx3+e
.*exx2)./ezz4./ezz1./(w.
*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzyw =
(((-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*(1./2.*n.*ezz1.*ezz2./eyy1.*((1-eyy1./ezz1)./n./
w-eyx1./ezz1.*(2./w.^2-2./w.^2.*e./(e+w)))-(ezz1-
ezz2).*e./w./(e+w)+1./2.*s.*ezz2.*ezz1./eyy2.
*(-(1-eyy2./ezz2)./s./w-eyx2./ezz2.*(2./w.^2-
2./w.^2.*e./(e+w))))+(n.*ezz1.*ezz2./eyy1+s.*ezz2
.*ezz1./eyy2).*(-n.*ezz3./eyy4.*eyx4./w./(e+w)-
s.*ezz4./eyy3.*eyx3./w./(e+w)-(ezz4-ezz3).*e./w
./(e+w))).ezz4./ezz3./(n.*eyy3+s.*eyy4)./ezz2./ezz1./(n.*eyy2+s.*eyy1
)./(e+w).*eyy4.*eyy3.*ey
y1.*w.*eyy2.*e+(1./2.*(ezz3./exx2.*ezz2.*w+ezz2./exx3.*ezz3.*e).*ezz4.
*(2.*exx1./ezz1./w.^2+ex
y1./ezz1./n./w)./exx1.*ezz1.*w-
1./2.*(ezz4./exx1.*ezz1.*w+ezz1./exx4.*ezz4.*e).*ezz3.*(2.*exx2
./ezz2./w.^2-
exy2./ezz2./s./w)./exx2.*ezz2.*w)./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz4
./ezz1./(w.
*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
bzyp =
(((-n.*ezz4.*ezz3./eyy4-
s.*ezz3.*ezz4./eyy3).*(1./2.*n.*ezz1.*ezz2./eyy1.*(-k.^2.*eyx1-(1-eyy1
./ezz1)./n./w-eyx1./ezz1.*(-2./w.^2+2./w.^2.*(e-w)./e))+(ezz1-
ezz2).*(e-w)./e./w+1./2.*s.*ezz2
.*ezz1./eyy2.*(-k.^2.*eyx2+(1-eyy2./ezz2)./s./w-eyx2./ezz2.*(-
2./w.^2+2./w.^2.*(e-w)./e)))+(n.
*ezz1.*ezz2./eyy1+s.*ezz2.*ezz1./eyy2).*(1./2.*n.*ezz4.*ezz3./eyy4.*(-
k.^2.*eyx4+(1-eyy4./ezz4
)./n./e-eyx4./ezz4.*(-2./e.^2-2./e.^2.*(e-
w)./w))+1./2.*s.*ezz3.*ezz4./eyy3.*(-k.^2.*eyx3-(1-e
yy3./ezz3)./s./e-eyx3./ezz3.*(-2./e.^2-2./e.^2.*(e-w)./w))+(ezz4-
ezz3).*(e-w)./e./w)).ezz4./e
zz3./(n.*eyy3+s.*eyy4)./ezz2./ezz1./(n.*eyy2+s.*eyy1)./(e+w).*eyy4.*ey
y3.*eyy1.*w.*eyy2.*e+((e
zz3./exx2.*ezz2.*w+ezz2./exx3.*ezz3.*e).*(1./2.*ezz4.*(-2./n.^2-
2.*exx1./ezz1./w.^2+k.^2.*exx1
-exy1./ezz1./n./w)./exx1.*ezz1.*w+1./2.*ezz1.*(-2./n.^2-
2.*exx4./ezz4./e.^2+k.^2.*exx4+exy4./e
zz4./n./e)./exx4.*ezz4.*e)-
(ezz4./exx1.*ezz1.*w+ezz1./exx4.*ezz4.*e).*(1./2.*ezz3.*(-2./s.^2-2
```

```
.*exx2./ezz2./w.^2+k.^2.*exx2+exy2./ezz2./s./w)./exx2.*ezz2.*w+1./2.*e
zz2.*(-2./s.^2-2.*exx3./
ezz3./e.^2+k.^2.*exx3-
exy3./ezz3./s./e)./exx3.*ezz3.*e))./ezz3./ezz2./(w.*exx3+e.*exx2)./ezz
4.
/ezz1./(w.*exx4+e.*exx1)./(n+s).*exx2.*exx3.*n.*exx1.*exx4.*s)./b;
ii = zeros(nx,ny);
ii(:) = (1:nx*ny);
% NORTH boundary
ib = zeros(nx,1); ib(:) = ii(1:nx,ny);
switch (boundary(1))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized north boundary condition: %s.\n', boundary(1));
end
bzxs(ib) = bzxs(ib) + sign*bzxn(ib);
bzxse(ib) = bzxse(ib) + sign*bzxne(ib);
bzxsw(ib) = bzxsw(ib) + sign*bzxnw(ib);
bzys(ib) = bzys(ib) - sign*bzyn(ib);
bzyse(ib) = bzyse(ib) - sign*bzyne(ib);
bzysw(ib) = bzysw(ib) - sign*bzynw(ib);
% SOUTH boundary
ib = zeros(nx,1); ib(:) = ii(1:nx,1);
switch (boundary(2))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized south boundary condition: %s.\n', boundary(2));
end
bzxn(ib) = bzxn(ib) + sign*bzxs(ib);
bzxne(ib) = bzxne(ib) + sign*bzxse(ib);
bzxnw(ib) = bzxnw(ib) + sign*bzxsw(ib);
bzyn(ib) = bzyn(ib) - sign*bzys(ib);
bzyne(ib) = bzyne(ib) - sign*bzyse(ib);
bzynw(ib) = bzynw(ib) - sign*bzysw(ib);
% EAST boundary
ib = zeros(1,ny); ib(:) = ii(nx,1:ny);
switch (boundary(3))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized east boundary condition: %s.\n', boundary(3));
end
bzxw(ib) = bzxw(ib) + sign*bzxe(ib);
bzxnw(ib) = bzxnw(ib) + sign*bzxne(ib);
bzxsw(ib) = bzxsw(ib) + sign*bzxse(ib);
bzyw(ib) = bzyw(ib) - sign*bzye(ib);
bzynw(ib) = bzynw(ib) - sign*bzyne(ib);
bzysw(ib) = bzysw(ib) - sign*bzyse(ib);
% WEST boundary
ib = zeros(1,ny); ib(:) = ii(1,1:ny);
switch (boundary(4))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized west boundary condition: %s.\n', boundary(4));
```

```
end
bzxe(ib) = bzxe(ib) + sign*bzxw(ib);
bzxne(ib) = bzxne(ib) + sign*bzxnw(ib);
bzxse(ib) = bzxse(ib) + sign*bzxsw(ib);
bzye(ib) = bzye(ib) - sign*bzyw(ib);
bzyne(ib) = bzyne(ib) - sign*bzynw(ib);
bzyse(ib) = bzyse(ib) - sign*bzysw(ib);
% Assemble sparse matrix
iall = zeros(1,nx*ny); iall(:) = ii;
is = zeros(1,nx*(ny-1)); is(:) = ii(1:nx,1:(ny-1));
in = zeros(1,nx*(ny-1)); in(:) = ii(1:nx,2:ny);
ie = zeros(1,(nx-1)*ny); ie(:) = ii(2:nx,1:ny);
iw = zeros(1,(nx-1)*ny); iw(:) = ii(1:(nx-1),1:ny);
ine = zeros(1,(nx-1)*(ny-1)); ine(:) = ii(2:nx, 2:ny);
ise = zeros(1,(nx-1)*(ny-1)); ise(:) = ii(2:nx, 1:(ny-1));
isw = zeros(1,(nx-1)*(ny-1)); isw(:) = ii(1:(nx-1), 1:(ny-1));
inw = zeros(1,(nx-1)*(ny-1)); inw(:) = ii(1:(nx-1), 2:ny);
Bzx = sparse ([iall,iw,ie,is,in,ine,ise,isw,inw], ...
[iall,ie,iw,in,is,isw,inw,ine,ise], ...
[bzxp(iall),bzxe(iw),bzxw(ie),bzxn(is),bzxs(in), ...
bzxsw(ine),bzxnw(ise),bzxne(isw),bzxse(inw)]);
Bzy = sparse ([iall,iw,ie,is,in,ine,ise,isw,inw], ...
[iall,ie,iw,in,is,isw,inw,ine,ise], ...
[bzyp(iall),bzye(iw),bzyw(ie),bzyn(is),bzys(in), ...
bzysw(ine),bzynw(ise),bzyne(isw),bzyse(inw)]);
B = [Bzx Bzy];
Hz = zeros(size(Hx));
Hz(:) = B*reshape([Hx,Hy],2*nx*ny,1)/j;
nx = nx-1;
ny = ny-1;
exx = epsxx(2:nx+1,2:ny+1);
exy = epsxy(2:nx+1,2:ny+1);
eyx = epsyx(2:nx+1,2:ny+1);
eyy = epsyy(2:nx+1,2:ny+1);
ezz = epszz(2:nx+1,2:ny+1);
edet = (exx.*eyy - exy.*eyx);
h = dx(2:nx+1)*ones(1,ny);
v = ones(nx,1)*dy(2:ny+1);
i1 = ii(1:nx,2:ny+1);
i2 = ii(1:nx,1:ny);
i3 = ii(2:nx+1,1:ny);
i4 = ii(2:nx+1,2:ny+1);
Dx = +neff*(Hy(i1) + Hy(i2) + Hy(i3) + Hy(i4))/4 + ...
(Hz(i1) + Hz(i4) - Hz(i2) - Hz(i3))./(j*2*k*v);
Dy = -neff*(Hx(i1) + Hx(i2) + Hx(i3) + Hx(i4))/4 - ...
(Hz(i3) + Hz(i4) - Hz(i1) - Hz(i2))./(j*2*k*h);
Dz = ((Hy(i3) + Hy(i4) - Hy(i1) - Hy(i2))./(2*h) - ...
(Hx(i1) + Hx(i4) - Hx(i2) - Hx(i3))./(2*v))/(j*k);
Ex = (eyy.*Dx - exy.*Dy)./edet;
Ey = (exx.*Dy - eyx.*Dx)./edet;
Ez = Dz./ezz;
```

Stretchmesh function code

```
function [x,y,xc,yc,dx,dy] = stretchmesh(x,y,nlayers,factor,method)
% This function can be used to continuously stretch the grid
% spacing at the edges of the computation window for
% finite-difference calculations. This is useful when you would
% like to increase the size of the computation window without
% increasing the total number of points in the computational
% domain. The program implements four different expansion
% methods: uniform, linear, parabolic (the default) and
```

```
% geometric. The first three methods also allow for complex
% coordinate stretching, which is useful for creating
% perfectly-matched non-reflective boundaries.
%
% USAGE:
%
% [x,y] = stretchmesh(x,y,nlayers,factor);
% [x,y] = stretchmesh(x,y,nlayers,factor,method);
% [x,y,xc,yc] = stretchmesh(x,y,nlayers,factor);
% [x,y,xc,yc] = stretchmesh(x,y,nlayers,factor,method);
% [x,y,xc,yc,dx,dy] = stretchmesh(x,y,nlayers,factor);
% [x,y,xc,yc,dx,dy] = stretchmesh(x,y,nlayers,factor,method);
%
% INPUT:
%
% x,y - vectors that specify the vertices of the original
% grid, which are usually linearly spaced.
% nlayers - vector that specifies how many layers of the grid
% you would like to expand:
% nlayers(1) = # of layers on the north boundary to stretch
% nlayers(2) = # of layers on the south boundary to stretch
% nlayers(3) = # of layers on the east boundary to stretch
% nlayers(4) = # of layers on the west boundary to stretch
% factor - cumulative factor by which the layers are to be
% expanded. As with nlayers, this can be a 4-vector.
% method - 4-letter string specifying the method of
% stretching for each of the four boundaries. Four different
% methods are supported: uniform, linear, parabolic (default)
% and geometric. For example, method = 'LLLG' will use linear
% expansion for the north, south and east boundaries and
% geometric expansion for the west boundary.
%
% OUTPUT:
%
% x,y - the vertices of the new stretched grid
% xc,yc (optional) - the center cell coordinates of the
% stretched grid
% dx,dy (optional) - the grid spacing (dx = diff(x))
%
% AUTHOR: Thomas E. Murphy (tem@umd.edu)
if (nargin < 5)
method = 'PPPP';
end
if isscalar(factor)
factor = factor*ones(1,4);
end
% Stretch out north boundary
n = nlayers(1);
f = factor(1);
if ((n > 0) & (f ~= 1));
kv = (length(y)-n:length(y));
q1 = y(length(y)-n);
q2 = y(length(y));
switch upper(method(1))
case 'U' % Uniform expansion
c = polyfit([q1,q2],[q1,q1 + f*(q2-q1)],1);
y(kv) = polyval(c,y(kv));
case 'L' % Linear expansion
c = (f-1)/(q2-q1);
b = 1 - 2*c*q1;
a = q1 - b*q1 - c*q1^2;
```

```
y(kv) = a + b*y(kv) + c*y(kv).^2;
case 'P' % Parabolic expansion
y(kv) = y(kv) + (f-1)*(y(kv)-q1).^3/(q2-q1).^2;
case 'G' % Geometric expansion
b = fzero(@(z) exp(z)-1-real(f)*z,real(f));
a = (q2-q1)/b;
y(kv) = q1 + a*(exp((y(kv)-q1)/a) - 1);
end
end
% Stretch out south boundary
n = nlayers(2);
f = factor(2);
if ((n > 0) & (f ~= 1));
kv = (1:1+n);
q1 = y(1+n);
q2 = y(1);
switch upper(method(2))
case 'U' % Uniform expansion
c = polyfit([q1,q2],[q1,q1 + f*(q2-q1)],1);
y(kv) = polyval(c,y(kv));
case 'L' % Linear expansion
c = (f-1)/(q2-q1);
b = 1 - 2*c*q1;
a = q1 - b*q1 - c*q1^2;
y(kv) = a + b*y(kv) + c*y(kv).^2;
case 'P' % Parabolic expansion
y(kv) = y(kv) + (f-1)*(y(kv)-q1).^3/(q2-q1).^2;
case 'G' % Geometric expansion
b = fzero(@(z) exp(z)-1-f*z,f);
a = (q2-q1)/b;
y(kv) = q1 + a*(exp((y(kv)-q1)/a) - 1);
end
end
% Stretch out east boundary
n = nlayers(3);
f = factor(3);
if ((n > 0) & (f ~= 1));
kv = (length(x)-n:length(x));
q1 = x(length(x)-n);
q2 = x(length(x));
switch upper(method(3))
case 'U' % Uniform expansion
c = polyfit([q1,q2],[q1,q1 + f*(q2-q1)],1);
x(kv) = polyval(c,x(kv));
case 'L' % Linear expansion
c = (f-1)/(q2-q1);
b = 1 - 2*c*q1;
a = q1 - b*q1 - c*q1^2;
x(kv) = a + b*x(kv) + c*x(kv).^2;
case 'P' % Parabolic expansion
x(kv) = x(kv) + (f-1)*(x(kv)-q1).^3/(q2-q1).^2;
case 'G' % Geometric expansion
b = fzero(@(z) exp(z)-1-f*z,f);
a = (q2-q1)/b;
x(kv) = q1 + a*(exp((x(kv)-q1)/a) - 1);
end
end
% Stretch out west boundary
n = nlayers(4);
f = factor(4);
if ((n > 0) & (f ~= 1));
```

```
kv = (1:1+n);
q1 = x(1+n);
q2 = x(1);
switch upper(method(4))
case 'U' % Uniform expansion
c = polyfit([q1,q2],[q1,q1 + f*(q2-q1)],1);
x(kv) = polyval(c,x(kv));
case 'L' % Linear expansion
c = (f-1)/(q2-q1);
b = 1 - 2*c*q1;
a = q1 - b*q1 - c*q1^2;
x(kv) = a + b*x(kv) + c*x(kv).^2;
case 'P' % Parabolic expansion
x(kv) = x(kv) + (f-1)*(x(kv)-q1).^3/(q2-q1).^2;
case 'G' % Geometric expansion
b = fzero(@(z) exp(z)-1-f*z,f);
a = (q2-q1)/b;
x(kv) = q1 + a*(exp((x(kv)-q1)/a) - 1);
end
end
if (nargout > 2)
kv = 1:length(x)-1;
xc = (x(kv) + x(kv+1))/2;
kv = 1:length(y)-1;
yc = (y(kv) + y(kv+1))/2;
end
if (nargout > 4)
dx = diff(x);
dy = diff(y);
end
```

Waveguidemeshfull function code

```
function [x,y,xc,yc,nx,ny,eps,varargout] =
waveguidemeshfull(n,h,rh,rw,side,dx,dy);
% This function creates an index mesh for the finite-difference
% mode solver. The function will accommodate a generalized three
% layer rib waveguide structure. (Note: channel waveguides can
% also be treated by selecting the parameters appropriately.)
%
% USAGE:
%
% [x,y,xc,yc,nx,ny,eps] = waveguidemeshfull(n,h,rh,rw,side,dx,dy)
% [x,y,xc,yc,nx,ny,edges] = waveguidemeshfull(n,h,rh,rw,side,dx,dy)
%
% INPUT
%
% n - indices of refraction for layers in waveguide
% h - height of each layer in waveguide
% rh - height of waveguide feature
% rw - half-width of waveguide
% side - excess space to the right of waveguide
% dx - horizontal grid spacing
% dy - vertical grid spacing
%
% OUTPUT
%
% x,y - vectors specifying mesh coordinates
% xc,yc - vectors specifying grid-center coordinates
% nx,ny - size of index mesh
% eps - index mesh (n^2)
% edges - (optional) list of edge coordinates, to be used later
```

```
% with the line() command to plot the waveguide edges
%
% AUTHOR: Thomas E. Murphy (tem@umd.edu)
if isscalar(side)
side1 = side;
side2 = side;
else
side1 = side(1);
side2 = side(2);
end
ih = round(h/dy);
irh = round (rh/dy);
irw = round (2*rw/dx);
iside1 = round (side1/dx);
iside2 = round (side2/dx);
nlayers = length(h);
nx = irw+iside1+iside2+1;
ny = sum(ih)+1;
x = dx*(-(irw/2+iside1):1:(irw/2+iside2))';
xc = (x(1:nx-1) + x(2:nx))/2;
y = (0:(ny-1))*dy;
yc = (1:(ny-1))*dy - dy/2;
eps = zeros(nx-1,ny-1);
iy = 1;
for jj = 1:nlayers,
for i = 1:ih(jj),
eps(:,iy) = n(jj)^2*ones(nx-1,1);
iy = iy+1;
end
end
iy = sum(ih)-ih(nlayers);
for i = 1:irh,
eps(1:iside1,iy) = n(nlayers)^2*ones(iside1,1);
eps(irw+iside1+1:irw+iside1+iside2,iy) = n(nlayers)^2*ones(iside2,1);
iy = iy-1;
end
nx = length(xc);
ny = length(yc);
if (nargout == 8)
iyp = cumsum(ih);
for jj = 1:nlayers-2,
if (iyp(jj) >= (iyp(nlayers-1)-irh))
edges{1,jj} = dx*[-irw/2,irw/2];
else
edges{1,jj} = dx*[-(irw/2+iside1),(irw/2+iside2)];
end
edges{2,jj} = dy*[1,1]*iyp(jj);
end
jj = nlayers-1;
edges{1,jj} = dx*[-(irw/2+iside1),-irw/2,-irw/2,+irw/2, ...
+irw/2,(irw/2+iside2)];
edges{2,jj} = dy*[iyp(jj)-irh,iyp(jj)-irh,iyp(jj),iyp(jj), ...
iyp(jj)-irh,iyp(jj)-irh];
varargout(1) = {edges};
end
```

Wgmodes function code

```
function [phix,phiy,neff] = wgmodes (lambda, guess, nmodes, dx, dy,
varargin);
% This function computes the two transverse magnetic field
% components of a dielectric waveguide, using the finite
```

```
% difference method. For details about the method, please
% consult:
%
% A. B. Fallahkhair, K. S. Li and T. E. Murphy, "Vector Finite
% Difference Modesolver for Anisotropic Dielectric
% Waveguides", J. Lightwave Technol. 26(11), 1423-1431,
% (2008).
%
% USAGE:
%
% [hx,hy,neff] = wgmodes(lambda, guess, nmodes, dx, dy, ...
% eps,boundary);
% [hx,hy,neff] = wgmodes(lambda, guess, nmodes, dx, dy, ...
% epsxx, epsyy, epszz, boundary);
% [hx,hy,neff] = wgmodes(lambda, guess, nmodes, dx, dy, ...
% epsxx, epsxy, epsyx, epsyy, epszz, boundary);
%
% INPUT:
%
% lambda - optical wavelength
% guess - scalar shift to apply when calculating the eigenvalues.
% This routine will return the eigenpairs which have an
% effective index closest to this guess
% nmodes - the number of modes to calculate
% dx - horizontal grid spacing (vector or scalar)
% dy - vertical grid spacing (vector or scalar)
% eps - index mesh (isotropic materials) OR:
% epsxx, epsxy, epsyx, epsyy, epszz - index mesh (anisotropic)
% boundary - 4 letter string specifying boundary conditions to be
% applied at the edges of the computation window.
% boundary(1) = North boundary condition
% boundary(2) = South boundary condition
% boundary(3) = East boundary condition
% boundary(4) = West boundary condition
% The following boundary conditions are supported:
% 'A' - Hx is antisymmetric, Hy is symmetric.
% 'S' - Hx is symmetric and, Hy is antisymmetric.
% '0' - Hx and Hy are zero immediately outside of the
% boundary.
%
% OUTPUT:
%
% hx - three-dimensional vector containing Hx for each
% calculated mode
% hy - three-dimensional vector containing Hy for each
% calculated mode (e.g.: hy(:,k) = two dimensional Hy
% matrix for the k-th mode
% neff - vector of modal effective indices
%
% NOTES:
%
% 1) The units are arbitrary, but they must be self-consistent
% (e.g., if lambda is in um, then dx and dy should also be in
% um.
%
% 2) Unlike the E-field modesolvers, this method calculates
% the transverse MAGNETIC field components Hx and Hy. Also,
% it calculates the components at the edges (vertices) of
% each cell, rather than in the center of each cell. As a
% result, if size(eps) = [n,m], then the output eigenvectors
% will be have a size of [n+1,m+1].
```

```
%
% 3) This version of the modesolver can optionally support
% non-uniform grid sizes. To use this feature, you may let dx
% and/or dy be vectors instead of scalars.
%
% 4) The modesolver can consider anisotropic materials, provided
% the permittivity of all constituent materials can be
% expressed in one of the following forms:
%
% [eps 0 0 ] [epsxx 0 0 ] [epsxx epsxy 0 ]
% [ 0 eps 0 ] [ 0 epsyy 0 ] [epsyx epsyy 0 ]
% [ 0 0 eps] [ 0 0 epszz] [ 0 0 epszz]
%
% The program will decide which form is appropriate based upon
% the number of input arguments supplied.
%
% 5) Perfectly matched boundary layers can be accomodated by
% using the complex coordinate stretching technique at the
% edges of the computation window. (stretchmesh.m can be used
% for complex or real-coordinate stretching.)
%
% AUTHORS: Thomas E. Murphy (tem@umd.edu)
% Arman B. Fallahkhair (a.b.fallah@gmail.com)
% Kai Sum Li (ksl3@njit.edu)
if (nargin == 11)
epsxx = varargin{1};
epsxy = varargin{2};
epsyx = varargin{3};
epsyy = varargin{4};
epszz = varargin{5};
boundary = varargin{6};
elseif (nargin == 9)
epsxx = varargin{1};
epsxy = zeros(size(epsxx));
epsyx = zeros(size(epsxx));
epsyy = varargin{2};
epszz = varargin{3};
boundary = varargin{4};
elseif (nargin == 7)
epsxx = varargin{1};
epsxy = zeros(size(epsxx));
epsyx = zeros(size(epsxx));
epsyy = epsxx;
epszz = epsxx;
boundary = varargin{2};
else
error('Incorrect number of input arguments.\n');
end
 [nx,ny] = size(epsxx);
nx = nx + 1;
ny = ny + 1;
% now we pad eps on all sides by one grid point
epsxx = [epsxx(:,1),epsxx,epsxx(:,ny-1)];
epsxx = [epsxx(1,1:ny+1);epsxx;epsxx(nx-1,1:ny+1)];
epsyy = [epsyy(:,1),epsyy,epsyy(:,ny-1)];
epsyy = [epsyy(1,1:ny+1);epsyy;epsyy(nx-1,1:ny+1)];
epsxy = [epsxy(:,1),epsxy,epsxy(:,ny-1)];
epsxy = [epsxy(1,1:ny+1);epsxy;epsxy(nx-1,1:ny+1)];
epsyx = [epsyx(:,1),epsyx,epsyx(:,ny-1)];
epsyx = [epsyx(1,1:ny+1);epsyx;epsyx(nx-1,1:ny+1)];
epszz = [epszz(:,1),epszz,epszz(:,ny-1)];
```

```
epszz = [epszz(1,1:ny+1);epszz;epszz(nx-1,1:ny+1)];
k = 2*pi/lambda; % free-space wavevector
if isscalar(dx)
dx = dx*ones(nx+1,1); % uniform grid
else
dx = dx(:); % convert to column vector
dx = [dx(1);dx;dx(length(dx))]; % pad dx on top and bottom
end
if isscalar(dy)
dy = dy*ones(1,ny+1); % uniform grid
else
dy = dy(:); % convert to column vector
dy = [dy(1);dy;dy(length(dy))].'; % pad dy on top and bottom
end
% distance to neighboring points to north south east and west,
% relative to point under consideration (P), as shown below.
n = ones(1,nx*ny); n(:) = ones(nx,1)*dy(2:ny+1);
s = ones(1,nx*ny); s(:) = ones(nx,1)*dy(1:ny);
e = ones(1,nx*ny); e(:) = dx(2:nx+1)*ones(1,ny);
w = ones(1,nx*ny); w(:) = dx(1:nx)*ones(1,ny);
% epsilon tensor elements in regions 1,2,3,4, relative to the
% mesh point under consideration (P), as shown below.
%
% NW------N------NE
% | | |
% | 1 n 4 |
% | | |
% W---w---P---e---E
% | | |
% | 2 s 3 |
% | | |
% SW------S------SE
exx1 = ones(1,nx*ny); exx1(:) = epsxx(1:nx,2:ny+1);
exx2 = ones(1,nx*ny); exx2(:) = epsxx(1:nx,1:ny);
exx3 = ones(1,nx*ny); exx3(:) = epsxx(2:nx+1,1:ny);
exx4 = ones(1,nx*ny); exx4(:) = epsxx(2:nx+1,2:ny+1);
eyy1 = ones(1,nx*ny); eyy1(:) = epsyy(1:nx,2:ny+1);
eyy2 = ones(1,nx*ny); eyy2(:) = epsyy(1:nx,1:ny);
eyy3 = ones(1,nx*ny); eyy3(:) = epsyy(2:nx+1,1:ny);
eyy4 = ones(1,nx*ny); eyy4(:) = epsyy(2:nx+1,2:ny+1);
exy1 = ones(1,nx*ny); exy1(:) = epsxy(1:nx,2:ny+1);
exy2 = ones(1,nx*ny); exy2(:) = epsxy(1:nx,1:ny);
exy3 = ones(1,nx*ny); exy3(:) = epsxy(2:nx+1,1:ny);
exy4 = ones(1,nx*ny); exy4(:) = epsxy(2:nx+1,2:ny+1);
eyx1 = ones(1,nx*ny); eyx1(:) = epsyx(1:nx,2:ny+1);
eyx2 = ones(1,nx*ny); eyx2(:) = epsyx(1:nx,1:ny);
eyx3 = ones(1,nx*ny); eyx3(:) = epsyx(2:nx+1,1:ny);
eyx4 = ones(1,nx*ny); eyx4(:) = epsyx(2:nx+1,2:ny+1);
ezz1 = ones(1,nx*ny); ezz1(:) = epszz(1:nx,2:ny+1);
ezz2 = ones(1,nx*ny); ezz2(:) = epszz(1:nx,1:ny);
ezz3 = ones(1,nx*ny); ezz3(:) = epszz(2:nx+1,1:ny);
ezz4 = ones(1,nx*ny); ezz4(:) = epszz(2:nx+1,2:ny+1);
ns21 = n.*eyy2+s.*eyy1;
ns34 = n.*eyy3+s.*eyy4;
ew14 = e.*exx1+w.*exx4;
ew23 = e.*exx2+w.*exx3;
axxn = ((2*eyy4.*e-eyx4.*n).*(eyy3./ezz4)./ns34 + ...
(2*eyy1.*w+eyx1.*n).*(eyy2./ezz1)./ns21)./(n.*(e+w));
axxs = ((2*eyy3.*e+eyx3.*s).*(eyy4./ezz3)./ns34 + ...
(2*eyy2.*w-eyx2.*s).*(eyy1./ezz2)./ns21)./(s.*(e+w));
ayye = (2.*n.*exx4 - e.*exy4).*exx1./ezz4./e./ew14./(n+s) + ...
```

```
(2.*s.*exx3 + e.*exy3).*exx2./ezz3./e./ew23./(n+s);
ayyw = (2.*exx1.*n + exy1.*w).*exx4./ezz1./w./ew14./(n+s) + ...
(2.*exx2.*s - exy2.*w).*exx3./ezz2./w./ew23./(n+s);
axxe = 2./(e.*(e+w)) + ...
(eyy4.*eyx3./ezz3 - eyy3.*eyx4./ezz4)./(e+w)./ns34;
axxw = 2./(w.*(e+w)) + ...
(eyy2.*eyx1./ezz1 - eyy1.*eyx2./ezz2)./(e+w)./ns21;
ayyn = 2./(n.*(n+s)) + ...
(exx4.*exy1./ezz1 - exx1.*exy4./ezz4)./(n+s)./ew14;
ayys = 2./(s.*(n+s)) + ...
(exx2.*exy3./ezz3 - exx3.*exy2./ezz2)./(n+s)./ew23;
axxne = +eyx4.*eyy3./ezz4./(e+w)./ns34;
axxse = -eyx3.*eyy4./ezz3./(e+w)./ns34;
axxnw = -eyx1.*eyy2./ezz1./(e+w)./ns21;
axxsw = +eyx2.*eyy1./ezz2./(e+w)./ns21;
ayyne = +exy4.*exx1./ezz4./(n+s)./ew14;
ayyse = -exy3.*exx2./ezz3./(n+s)./ew23;
ayynw = -exy1.*exx4./ezz1./(n+s)./ew14;
ayysw = +exy2.*exx3./ezz2./(n+s)./ew23;
axxp = - axxn - axxs - axxe - axxw - axxne - axxse - axxnw - axxsw ...
+ k^2*(n+s).*(eyy4.*eyy3.*e./ns34 + eyy1.*eyy2.*w./ns21)./(e+w);
ayyp = - ayyn - ayys - ayye - ayyw - ayyne - ayyse - ayynw - ayysw ...
+ k^2*(e+w).*(exx1.*exx4.*n./ew14 + exx2.*exx3.*s./ew23)./(n+s);
axyn = (eyy3.*eyy4./ezz4./ns34 - ...
eyy2.*eyy1./ezz1./ns21 + ...
s.*(eyy2.*eyy4 - eyy1.*eyy3)./ns21./ns34)./(e+w);
axys = (eyy1.*eyy2./ezz2./ns21 - ...
eyy4.*eyy3./ezz3./ns34 + ...
n.*(eyy2.*eyy4 - eyy1.*eyy3)./ns21./ns34)./(e+w);
ayxe = (exx1.*exx4./ezz4./ew14 - ...
exx2.*exx3./ezz3./ew23 + ...
w.*(exx2.*exx4 - exx1.*exx3)./ew23./ew14)./(n+s);
ayxw = (exx3.*exx2./ezz2./ew23 - ...
exx4.*exx1./ezz1./ew14 + ...
e.*(exx4.*exx2 - exx1.*exx3)./ew23./ew14)./(n+s);
axye = (eyy4.*(1-eyy3./ezz3) - eyy3.*(1-eyy4./ezz4))./ns34./(e+w) -
...
2*(eyx1.*eyy2./ezz1.*n.*w./ns21 + ...
eyx2.*eyy1./ezz2.*s.*w./ns21 + ...
eyx4.*eyy3./ezz4.*n.*e./ns34 + ...
eyx3.*eyy4./ezz3.*s.*e./ns34 + ...
eyy1.*eyy2.*(1./ezz1-1./ezz2).*w.^2./ns21 + ...
eyy3.*eyy4.*(1./ezz4-1./ezz3).*e.*w./ns34)./e./(e+w).^2;
axyw = (eyy2.*(1-eyy1./ezz1) - eyy1.*(1-eyy2./ezz2))./ns21./(e+w) -
...
2*(eyx4.*eyy3./ezz4.*n.*e./ns34 + ...
eyx3.*eyy4./ezz3.*s.*e./ns34 + ...
eyx1.*eyy2./ezz1.*n.*w./ns21 + ...
eyx2.*eyy1./ezz2.*s.*w./ns21 + ...
eyy4.*eyy3.*(1./ezz3-1./ezz4).*e.^2./ns34 + ...
eyy2.*eyy1.*(1./ezz2-1./ezz1).*w.*e./ns21)./w./(e+w).^2;
ayxn = (exx4.*(1-exx1./ezz1) - exx1.*(1-exx4./ezz4))./ew14./(n+s) -
...
2*(exy3.*exx2./ezz3.*e.*s./ew23 + ...
exy2.*exx3./ezz2.*w.*s./ew23 + ...
exy4.*exx1./ezz4.*e.*n./ew14 + ...
exy1.*exx4./ezz1.*w.*n./ew14 + ...
exx3.*exx2.*(1./ezz3-1./ezz2).*s.^2./ew23 + ...
exx1.*exx4.*(1./ezz4-1./ezz1).*n.*s./ew14)./n./(n+s).^2;
ayxs = (exx2.*(1-exx3./ezz3) - exx3.*(1-exx2./ezz2))./ew23./(n+s) -
...
```

```
2*(exy4.*exx1./ezz4.*e.*n./ew14 + ...
exy1.*exx4./ezz1.*w.*n./ew14 + ...
exy3.*exx2./ezz3.*e.*s./ew23 + ...
exy2.*exx3./ezz2.*w.*s./ew23 + ...
exx4.*exx1.*(1./ezz1-1./ezz4).*n.^2./ew14 + ...
exx2.*exx3.*(1./ezz2-1./ezz3).*s.*n./ew23)./s./(n+s).^2;
axyne = +eyy3.*(1-eyy4./ezz4)./(e+w)./ns34;
axyse = -eyy4.*(1-eyy3./ezz3)./(e+w)./ns34;
axynw = -eyy2.*(1-eyy1./ezz1)./(e+w)./ns21;
axysw = +eyy1.*(1-eyy2./ezz2)./(e+w)./ns21;
ayxne = +exx1.*(1-exx4./ezz4)./(n+s)./ew14;
ayxse = -exx2.*(1-exx3./ezz3)./(n+s)./ew23;
ayxnw = -exx4.*(1-exx1./ezz1)./(n+s)./ew14;
ayxsw = +exx3.*(1-exx2./ezz2)./(n+s)./ew23;
axyp = -(axyn + axys + axye + axyw + axyne + axyse + axynw + axysw)
...
- k^2.*(w.*(n.*eyx1.*eyy2 + s.*eyx2.*eyy1)./ns21 + ...
e.*(s.*eyx3.*eyy4 + n.*eyx4.*eyy3)./ns34)./(e+w);
ayxp = -(ayxn + ayxs + ayxe + ayxw + ayxne + ayxse + ayxnw + ayxsw)
...
- k^2.*(n.*(w.*exy1.*exx4 + e.*exy4.*exx1)./ew14 + ...
s.*(w.*exy2.*exx3 + e.*exy3.*exx2)./ew23)./(n+s);
ii = zeros(nx,ny);
ii(:) = (1:nx*ny);
% NORTH boundary
ib = zeros(nx,1); ib(:) = ii(1:nx,ny);
switch (boundary(1))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized north boundary condition: %s.\n', boundary(1));
end
axxs(ib) = axxs(ib) + sign*axxn(ib);
axxse(ib) = axxse(ib) + sign*axxne(ib);
axxsw(ib) = axxsw(ib) + sign*axxnw(ib);
ayxs(ib) = ayxs(ib) + sign*ayxn(ib);
ayxse(ib) = ayxse(ib) + sign*ayxne(ib);
ayxsw(ib) = ayxsw(ib) + sign*ayxnw(ib);
ayys(ib) = ayys(ib) - sign*ayyn(ib);
ayyse(ib) = ayyse(ib) - sign*ayyne(ib);
ayysw(ib) = ayysw(ib) - sign*ayynw(ib);
axys(ib) = axys(ib) - sign*axyn(ib);
axyse(ib) = axyse(ib) - sign*axyne(ib);
axysw(ib) = axysw(ib) - sign*axynw(ib);
% SOUTH boundary
ib = zeros(nx,1); ib(:) = ii(1:nx,1);
switch (boundary(2))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized south boundary condition: %s.\n', boundary(2));
end
axxn(ib) = axxn(ib) + sign*axxs(ib);
axxne(ib) = axxne(ib) + sign*axxse(ib);
axxnw(ib) = axxnw(ib) + sign*axxsw(ib);
ayxn(ib) = ayxn(ib) + sign*ayxs(ib);
ayxne(ib) = ayxne(ib) + sign*ayxse(ib);
ayxnw(ib) = ayxnw(ib) + sign*ayxsw(ib);
ayyn(ib) = ayyn(ib) - sign*ayys(ib);
```

```
ayyne(ib) = ayyne(ib) - sign*ayyse(ib);
ayynw(ib) = ayynw(ib) - sign*ayysw(ib);
axyn(ib) = axyn(ib) - sign*axys(ib);
axyne(ib) = axyne(ib) - sign*axyse(ib);
axynw(ib) = axynw(ib) - sign*axysw(ib);
% EAST boundary
ib = zeros(1,ny); ib(:) = ii(nx,1:ny);
switch (boundary(3))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized east boundary condition: %s.\n', boundary(3));
end
axxw(ib) = axxw(ib) + sign*axxe(ib);
axxnw(ib) = axxnw(ib) + sign*axxne(ib);
axxsw(ib) = axxsw(ib) + sign*axxse(ib);
ayxw(ib) = ayxw(ib) + sign*ayxe(ib);
ayxnw(ib) = ayxnw(ib) + sign*ayxne(ib);
ayxsw(ib) = ayxsw(ib) + sign*ayxse(ib);
ayyw(ib) = ayyw(ib) - sign*ayye(ib);
ayynw(ib) = ayynw(ib) - sign*ayyne(ib);
ayysw(ib) = ayysw(ib) - sign*ayyse(ib);
axyw(ib) = axyw(ib) - sign*axye(ib);
axynw(ib) = axynw(ib) - sign*axyne(ib);
axysw(ib) = axysw(ib) - sign*axyse(ib);
% WEST boundary
ib = zeros(1,ny); ib(:) = ii(1,1:ny);
switch (boundary(4))
case 'S', sign = +1;
case 'A', sign = -1;
case '0', sign = 0;
otherwise,
error('Unrecognized west boundary condition: %s.\n', boundary(4));
end
axxe(ib) = axxe(ib) + sign*axxw(ib);
axxne(ib) = axxne(ib) + sign*axxnw(ib);
axxse(ib) = axxse(ib) + sign*axxsw(ib);
ayxe(ib) = ayxe(ib) + sign*ayxw(ib);
ayxne(ib) = ayxne(ib) + sign*ayxnw(ib);
ayxse(ib) = ayxse(ib) + sign*ayxsw(ib);
ayye(ib) = ayye(ib) - sign*ayyw(ib);
ayyne(ib) = ayyne(ib) - sign*ayynw(ib);
ayyse(ib) = ayyse(ib) - sign*ayysw(ib);
axye(ib) = axye(ib) - sign*axyw(ib);
axyne(ib) = axyne(ib) - sign*axynw(ib);
axyse(ib) = axyse(ib) - sign*axysw(ib);
% Assemble sparse matrix
iall = zeros(1,nx*ny); iall(:) = ii;
is = zeros(1,nx*(ny-1)); is(:) = ii(1:nx,1:(ny-1));
in = zeros(1,nx*(ny-1)); in(:) = ii(1:nx,2:ny);
ie = zeros(1,(nx-1)*ny); ie(:) = ii(2:nx,1:ny);
iw = zeros(1,(nx-1)*ny); iw(:) = ii(1:(nx-1),1:ny);
ine = zeros(1,(nx-1)*(ny-1)); ine(:) = ii(2:nx, 2:ny);
ise = zeros(1,(nx-1)*(ny-1)); ise(:) = ii(2:nx, 1:(ny-1));
isw = zeros(1,(nx-1)*(ny-1)); isw(:) = ii(1:(nx-1), 1:(ny-1));
inw = zeros(1,(nx-1)*(ny-1)); inw(:) = ii(1:(nx-1), 2:ny);
Axx = sparse ([iall,iw,ie,is,in,ine,ise,isw,inw], ...
[iall,ie,iw,in,is,isw,inw,ine,ise], ...
[axxp(iall),axxe(iw),axxw(ie),axxn(is),axxs(in), ...
axxsw(ine),axxnw(ise),axxne(isw),axxse(inw)]);
```

```
Axy = sparse ([iall,iw,ie,is,in,ine,ise,isw,inw], ...
[iall,ie,iw,in,is,isw,inw,ine,ise], ...
[axyp(iall),axye(iw),axyw(ie),axyn(is),axys(in), ...
axysw(ine),axynw(ise),axyne(isw),axyse(inw)]);
Ayx = sparse ([iall,iw,ie,is,in,ine,ise,isw,inw], ...
[iall,ie,iw,in,is,isw,inw,ine,ise], ...
[ayxp(iall),ayxe(iw),ayxw(ie),ayxn(is),ayxs(in), ...
ayxsw(ine),ayxnw(ise),ayxne(isw),ayxse(inw)]);
Ayy = sparse ([iall,iw,ie,is,in,ine,ise,isw,inw], ...
[iall,ie,iw,in,is,isw,inw,ine,ise], ...
[ayyp(iall),ayye(iw),ayyw(ie),ayyn(is),ayys(in), ...
ayysw(ine),ayynw(ise),ayyne(isw),ayyse(inw)]);
A = [[Axx Axy];[Ayx Ayy]];
% fprintf(1,'nnz(A) = %d\n',nnz(A));
shift = (guess*k)^2;
options.tol = 1e-8;
options.disp = 0; % suppress output
clear Axx Axy Ayx Ayy ...
axxnw axxne axxne ...
axxw axxp axxe ...
axxsw axxse axxse ...
axynw axyne axyne ...
axyw axyp axye ...
axysw axyse axyse ...
ayynw ayyne ayyne ...
ayyw ayyp ayye ...
ayysw ayyse ayyse ...
ayxnw ayxne ayxne ...
ayxw ayxp ayxe ...
ayxsw ayxse ayxse ...
iall ie iw in iw ...
isw inw ine ise ...
exx1 exx2 exx3 exx4 ...
exy1 exy2 exy3 exy4 ...
eyx1 eyx2 eyx3 eyx4 ...
eyy1 eyy2 eyy3 eyy4 ...
ezz1 ezz2 ezz3 ezz4 ...
ns21 ns34 ew14 ew23;
[v,d] = eigs(A,speye(size(A)),nmodes,shift,options);
neff = lambda*sqrt(diag(d))/(2*pi);
phix = zeros(nx,ny,nmodes);
phiy = zeros(nx,ny,nmodes);
temp = zeros(nx,2*ny);
% Normalize modes
temp = zeros(nx*ny,2);
for kk = 1:nmodes;
temp(:) = v(:,kk);
[mag,ii] = max(sqrt(sum(abs(temp).^2,2)));
if abs(temp(ii,1)) > abs(temp(ii,2)),
jj = 1;
else
jj = 2;
end
mag = mag*temp(ii,jj)/abs(temp(ii,jj));
temp = temp/mag;
phix(:,:,kk) = reshape(temp(:,1),nx,ny);
phiy(:,:,kk) = reshape(temp(:,2),nx,ny);
end;
return;
```

Appendix G

Power read function code

```
%% Reads power file created by OptiFDTD and builds table with values
read,
% considering 3 observation lines. Plots power results afterwards.
Using
% least squares method, plots data trend.
clear all
clc
f=fileread('wire_w=236nm_PowerAtWavelength.txt');
positions=strfind( f, 'ObservationLine1' );
data=textscan(f,'%s%s%s%s%s');
for m=1:3
i=1;
for n=m:3:3*numel(positions)
ObvLine{m,i}=str2double(strrep(data{1,5}(n),',','.'));
i=i+1;
end
end
radius=12.9:-0.1:3;
ob=cell2mat(ObvLine);
obT=ob';
% ob=(cell2mat(ObvLine))';
figure(1)
plot(radius,abs(ob))
grid on
legend('Leftmost observation line','Arc start observation line',...
'Arc end observation line','Location','SouthEast')
set(gca,'xdir','reverse')
title('Observation lines power vs arc decreasing radius')
xlabel('Arc radius [\mum]')
ylabel('Intensity [W/m]')
ob3=ob(:,3);
% radius=12.9:-0.1:3;
loss_dB=10*log10(abs(obT(:,3))./obT(:,2));
loss_pathUnit=loss_dB./radius';
figure(2)
plot(radius,loss_dB','r')
grid on
legend('Arc end power loss [dB]','Location','SouthEast')
set(gca,'xdir','reverse')
title('Radiation + intrinsic losses vs arc decreasing radius')
xlabel('Arc radius [\mum]')
ylabel('Total power loss [dB]')
figure(3)
plot(radius,loss_pathUnit','r')
grid on
legend('Radiation power loss/path unit [dB]','Location','SouthEast')
set(gca,'xdir','reverse')
title('Radiation power loss vs arc decreasing radius')
xlabel('Arc radius [\mum]')
ylabel('Radiation power loss [dB]')
leastSquares(radius',loss_pathUnit')
```

Leastsquares function code

```
function leastSquares(x,y)
%LEASTSQUARES Reconstruct figure in SPLINETOOL.
%
```

```
% LEASTSQUARES(X,Y) creates a plot, similar to the plot in SPLINETOOL,
% using the data that you provide as input.
% You can apply this function to the same data you used with
% SPLINETOOL or with different data. You may want to edit the
% function to customize the code or even this help message.
%
% Because of data-dependent changes, this may not work for data
% sites other than the ones used in SPLINETOOL when this file
% was written.
% Make sure the data are in rows ...
x = x(:).'; y = y(:).';
% ... and start by plotting the data specific to the highlighted
spline fit.
figure()
firstbox = [0.1300 0.4900 0.7750 0.4850];
subplot('Position',firstbox)
plot(x,y,'ok'), hold on
names={'data'};
ylabel('y')
xtick = get(gca,'Xtick');
set(gca,'xtick',[])
% Now generate and plot the fit.
% we are starting off with the least squares polynomial approximation
of
% order 4:
spline1 = spap2(1,4,x,y);
% extract knots from current approximation:
knots = fnbrk(spline1,'knots');
% you changed the order:
knots = augknt(knots,3);
% you changed the order:
knots = augknt(knots,2);
% least-squares approximation:
spline1 = spap2(knots,2,x,y);
names{end+1} = 'spline1'; fnplt(spline1,'-',2)
% Plot the second graph from SPLINETOOL
subplot('Position',[ 0.1300 0.1300 0.7750 0.3100])
plot(xtick([1 end]),zeros(1,2),'Linewidth',2,'Color',repmat(.6,1,3))
hold on
plot(x,y-fnval(spline1,x),'Linewidth',2)
hold off
ylabel('Error in spline1')
xlabel('x')
% Return to plotting the first graph
subplot('Position', firstbox)
legend(names{:},'Location','Southwest')
hold off
set(gcf,'NextPlot','replace')
```