Very sharp bends with a metal aided $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge waveguide

December, 2012

Author:
Teus Dubbink (s0202649)

Supervisor:
dr. Sonia Garcia Blanco

Abstract

Combination of rare-earth doped $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ gain material with plasmonic waveguides could pave the way to very small, low loss bends that could become building blocks for very interesting nanophotonic devices. $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ is a monoclinic crystalline material with three different optical axes. The effect of this anisotropy on the bending losses has not been studied in detail before.
In this work, a metal aided $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge waveguide structure is proposed and propagation losses for very small bends are investigated using two-dimensional finite-differences mode-solving techniques and three-dimensional finite difference time domain software.
It was obtained that the bendlosses for the TE mode at λ=1.55 μm are reduced with respect to the purely photonic counterpart of the proposed structure, for a bending radius smaller than 2.5 μm. It is observed that the anisotropy of the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge does not introduce additional losses. At the end, it is concluded that the mode-solving software can be used to simulate the proposed structure despite the anisotropy of the waveguide.

Contents

Chapter 1

Introduction

As smart TV and social networking advent, there is an increasing demand for devices that process a huge amount of information at very large bandwidths. In data centers and in computers, there is a huge increase in the demand for higher performances with lower energy consumption. As physical limits are nearly reached in sizes of electronic devices and the heat generated by the devices is inevitable, there is need for a different way of thinking. One of the proposed solutions is to switch from electronic to optical devices and process the information optically on the chip. This drives quite stringent demand for the development of very small, low power consumption, high bandwidth optical components.
As a consequence of the above demands, several technologies have been the subject of very high interest in recent years. Silicon photonics is an extensive research field, as the fabrication infrastructure for silicon devices is already there. Due to the high refractive index of silicon, it allows for very small photonic waveguides with relatively low losses. A disadvantage of silicon is that it is passive: active devices like lasers and amplifiers are hard to realize. A second field of interest is plasmonics. Plasmons are quanta of oscillations of electrons, existing in metals or at the interface between metals and dielectrics. A more profound explanation can be found in Section 2.1.3. Plasmonic waveguides can be very small, due to the far sub-wavelength mode confinement, and they allow for high bandwidth, ultra fast communication. A high efficiency in terms of modulation can be achieved due to the presence of the metal amidst the optical field. A major disadvantage of plasmonics is the high losses due to absorption in the metal.
A possible solution to this problems is to add a material with optical gain to the plasmonic waveguide structure. A material of particular interest for this work is a crystalline material called potassium double tungstate, with chemical formula $KT(WO_4)_2$. The T stands for a trivalent metal or rare-earth cation. With this material, crystal structures can be grown which can easily be doped with rare-earth metals. A more extensive discussion of this material can be found in the doctoral thesis of Dimitri Geskus[1]. For this thesis, the $K(WO_4)_2$ host material is doped with Yb_3^+ to achieve optical gain at around 1 micrometer wavelength, and with Gd and Lu to correct the lattice parameters. $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ has a monoclinic crystalline structure and has three optical axes, as shown in Figure 1.1.
$KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ can find very interesting applications in cases where amplifiers with high broadband gain[2], or tunable lasers with relatively high output power and narrow linewidths[3] are needed. $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ has also been proposed as a good candidate to compensate the losses of long-range dielectric loaded plasmonic waveguides[4]. However, bends in $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ have never been reported before. The possibility of realizing low-loss bend structures in $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$

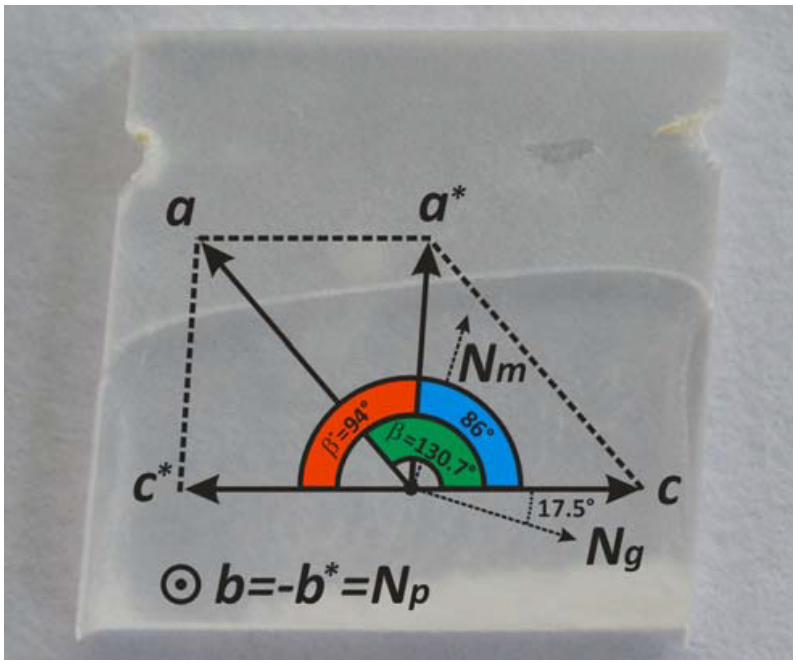


Figure 1.1 – The crystalline and optical axes of $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$. Copied from the thesis of Dimitri Geskus[1].

in combination with plasmonics might open the door to many interesting nano-devices that could be used to address the issues described above.
In this work, the bend losses of long-range dielectric loaded surface plasmon polariton waveguides with a $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge will be investigated. The bend losses obtained for both TE- and TM-modes will be compared to the losses of a similar structure without the metallic layers. Two commercial mode-solvers, FieldDesigner from PhoeniX B.V. and Mode Solutions from Lumerical were used to verify the results. The effect of the 3-axis anisotropy of $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ was also studied by means of 3D-FDTD Solutions from Lumerical.
It was obtained that the bendlosses for the TE mode at λ=1.55 μm are reduced with respect to the purely photonic counterpart of the proposed structure, for a bending radius smaller than 2.5 μm. It is observed that the anisotropy of the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge does not introduce additional losses. At the end, it is concluded that the mode-solving software can be used to simulate the proposed structure despite the anisotropy of the waveguide.

Outline In Chapter 2, the necessary theory about waveguides and plasmonics is described. Also, the simulation techniques used for this thesis are explained. In Chapter 3, the metal aided structure is introduced, as well as a benchmark structure. Also the necessary geometric parameters are defined. In addition, the simulations are prepared in terms of grids, window sizes and other simulation parameters. In Chapter 4, the results of all the simulations are shown and discussed. Conclusions are drawn in Chapter 5. In Chapter 6, recommendations are given for future work on the subject of this thesis.

Chapter 2

Theory

2.1 Waveguides

There are different types of waveguides. Relevant for this thesis are photonic waveguides, discussed in section 2.1.1, and plasmonic waveguides, discussed in section 2.1.3. Also losses in waveguides are discussed, in section 2.1.2.

2.1.1 Photonic waveguides

Photonic waveguides guide electromagnetic waves using total internal reflection. Total internal reflection occurs when the angle of incidence at a the interface between a material of high refractive index and a material of lower refractive index is larger than the critical angle.

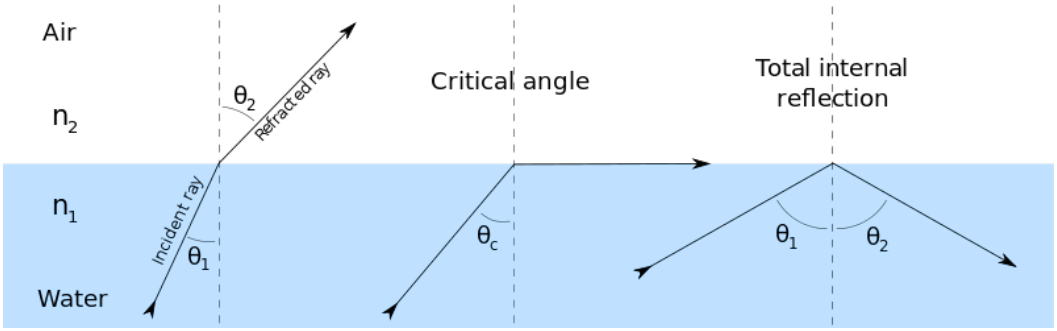


Figure 2.1 – Illustration of angles and refractive indices used for total internal reflection. Source: Wikipedia Commons[5]

As it can be seen in Figure 2.1, the angle of incidence of the light travelling from the high refractive index medium is smaller than the angle of the refracted angle in the lower refractive index medium. At the critical angle, the light in the low refractive index medium will travel with a 90 degree angle. The critical angle can be easily calculated using Snell's Law:

$$n_1 sin\theta_i = n_2 sin\theta_t$$

At the critical angle, $\sin\theta_t = \sin 90° = 1$, so $\sin\theta_c = \frac{n_2}{n_1}$, so the critical angle can be calculated using $\theta_c = \arcsin\frac{n_2}{n_1}$. For a typical air-glass interface this results in a critical angle of

41.8 degree.
For incident angles larger than the critical angle, the light will not cross the interface. In a structure as shown in Figure 2.2, the light will not leave the middle strip if $\alpha > \arcsin \frac{n_2}{n_1}$.

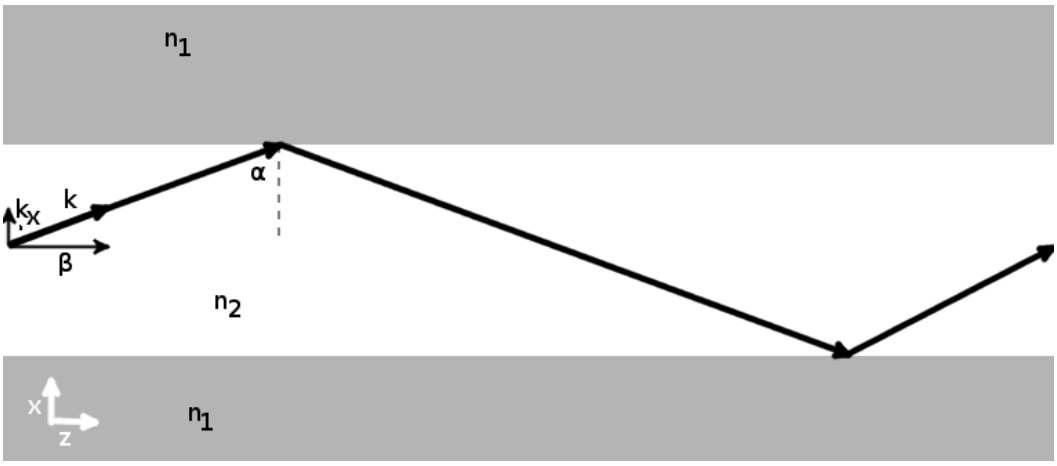


Figure 2.2 – Illustration of the different parameters involved with total internal reflection

The structure is a waveguide if another condition is fullfilled: there needs to be waveguiding. This means the wavefront inserted into the waveguide is maintained during the propagation. There has to be resonance in the transversal direction, in order to get constructive interference from waves reflected at the top interface and waves reflected at the bottom interface. This results in a discrete set of angles at which a mode exists. The ones larger than the critical angle determine the set of guided modes. The modes are often characterized by the effective index n_{eff} defined as

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

in which k_0 equals the propagation constant in free space and β equals the propagation constant of the mode inside the waveguide. The speed at which the mode travels through the waveguide can be calculated using

$$v_{mode} = c / n_{eff}.$$

The mainly used modes are TE (transversal electric) and TM (transversal magnetic) modes. The TE modes have an electric field component in the direction perpendicular to the propagation direction, but parallel to the reflecting surface. It has a magnetic component in the propagation direction and in the direction perpendicular to the surface. For the TM mode, it is the other way around. In some cases, modes are supported that have both electric and magnetic component in the direction of propagation. These modes are called hybrid modes.

Evanescent field

The mode does not stay inside the waveguide completely. At the surface, there will be an exponentially decaying field, which is called the evanescent field. Mathematically, this is represented by the imaginary part of the propagation vector in the direction perpendicular

to the propagation direction. Because it decays exponentially, the evanescent field only exists very close to the surface. This evanescent field causes a phase shift of the reflected fields, which makes analytic calculation of the modes even harder.

2.1.2 Losses

So far, losses have been ignored. Losses can mathematically be descriped by the imaginary part of the effective refractive index. Physically, they are caused by scattering, absorption and radiation[6].

Scattering

The scattering losses can be divided into volume and surface scattering losses. The volume scattering losses are due to defects in the material. The surface scattering losses are due to the roughness of the surface, and depend among other parameters on the decay coefficients of the evanescent field.

Absorption

The absorption losses can be divided into interband and free carrier absorption losses. The interband absorption losses are due to the excitation of electrons from the valence to the conduction band. These losses limit the range of wavelengths which can be used in the waveguide. The losses will increase dramatically when the used energy per photon is higher than the material bandgap. The free carrier absorption losses are due to the excitation of electrons which are already in the conduction band to a higher energy level.

Radiation losses

In a straight waveguide the radiation losses for the well confined modes will be very low. However, a bend in the waveguide introduces significantly more losses. The light travelling at the outer part of the bend needs a higher speed to keep up with the mode than the group velocity. When the necessary speed is higher than the phase velocity of the unguided light, the light will be radiated.

2.1.3 Plasmonics

In metals, the energy of the field can be stored in a different way: plasmons. A plasmon is, a quantum for the oscillation of charged particles. There are two types of plasmons: volume plasmons and surface plasmons.

Volume plasmons

The metal can be considered as a collection of positively charged, heavy atoms in a sea of negatively charged electrons. The motion of the free electrons can be descriped with the free electron model. When an electric field is applied onto the metal, the free electrons will move, but the atoms will stay in their place. This causes the material to be polarized. Because the polarization is linearly correlated to the displacement of the free electrons, the solution for an oscillating electric field can be described using an harmonic oscillator equation. The result is a resonance frequency ω_p, called plasmon frequency. Because of this resonance, the metal behaves like a dielectric material, with the real part of the permittivity larger than zero, for incoming electric fields with $\omega > \omega_p$. As the frequency decreases, the

metal acquires metallic character, with the real part of the permittivity being negative, while the absorption losses increase dramatically.

Surface plasmons

Surface plasmon polaritons, or shorter surface plasmons are propagating plasmons at the interface between a metal and a dielectric. At both sides of the interface, they are evanescently confined in the direction perpendicular to the surface, as shown in Figure 2.3. When the Maxwell equations are solved with the correct boundary conditions imposed, it is found that TM modes are supported, but TE modes are not[7].

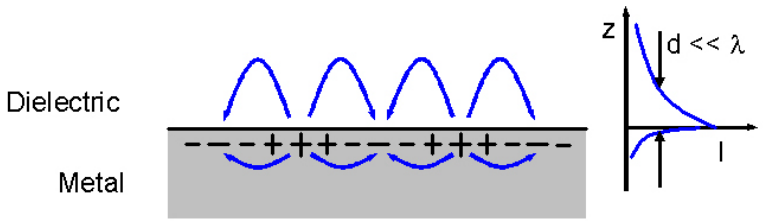


Figure 2.3 – Illustration of a surface plasmon on a metal-dielectric interface. Source: Wikipedia Commons[8]

Plasmonic waveguides

Surface plasmons can be used to make very small waveguides. Because of the evanescent decay of the electric field, the mode is confined to only a few nanometers. But because a relatively large part of the mode penetrates the metal, this comes at the cost of high propagation losses. Another disadvantage is that only TM modes are supported.

2.1.4 Hybrid waveguides

In order to have the advantage of both the low losses of photonic waveguides and the confinement of plasmonic waveguides, many photonic-plasmonic hybrid waveguides have been developed. Besides this hypothethical advantage, the presence of a metal near or inside the waveguide has more applications. It is for example possible to have thermoelectric control of the device by applying a voltage over the metal. This will cause changes in temperature and thus changes of the size and refractive index of the dielectric. Because photonic waveguides are highly sensitive to size and refractive index, this allows control of the optical properties of the waveguide.

2.2 Simulation methods

To calculate the behaviour of the waveguide, the electric and magnetic fields need to be resolved. Exact calculations are often simpler in frequency domain, but if it is necessary to simulate material with arbitrary properties it is more convenient to work in the time domain. A very successful although time consuming time domain method is the finite difference time domain (FDTD) method, which solves a discretized form of the Maxwell equations.

2.2.1 FDTD method

The FDTD method solves the time domain Maxwell equations:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

$$\nabla \cdot \mathbf{D} = \rho$$

$$\nabla \cdot \mathbf{B} = 0$$

The FDTD method uses a leapfrog algorithm to calculate the electric and magnetic field components. The whole space is divided into cells, based on the Yee cell. This is called meshing. This cell was proposed in 1966 by Kane Yee[9], and it defines what electric components are coupled on what magnetic components. Because the time dependency of the electric field is related to the space dependency of the magnetic field and vice versa, it has to be carefully chosen how to calculate the space dependant gradient at a certain point.

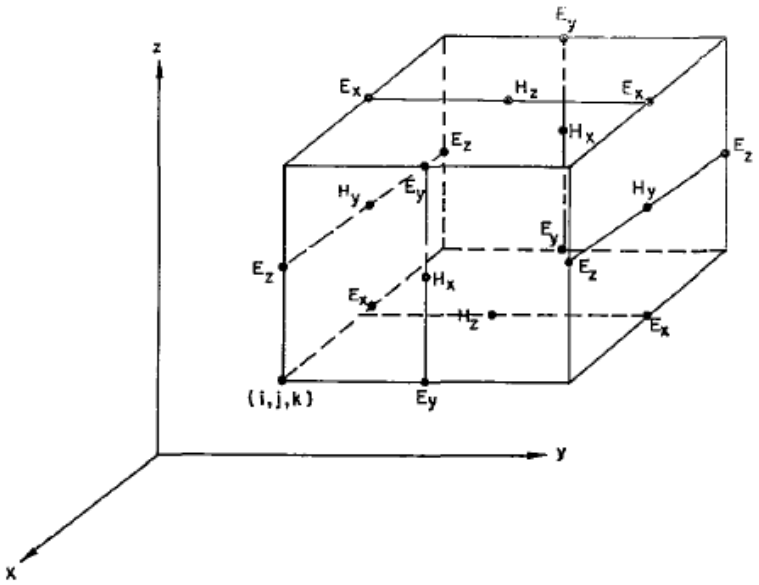


Figure 2.4 – Grid cell as proposed by Yee in his 1966 article.[9]

In Figure 2.4, the Yee cell is pictured. As it can be seen, the magnetic components are displaced half a cell in every direction with respect to the electric components, so that the points where the gradient is calculated are always exactly between the corresponding vectors. This way, the numerical errors are reduced significantly. The electric and magnetic components are calculated alternately throughout the entire space.
If the simulation region would just end without any special boundary layer, the fields radiating out of the simulation region would reflect and interfere with rest of the fields in the simulation window, which does not represent the reality that needs to be simulated. One important boundary layer is the perfectly matched layer (PML). The PML is designed

to cancel the fields coming into the PML using destructive interference, instead of just setting the field to zero outside the simulation region. This way there is very little, none in the ideal case, reflection for fields travelling perpendicular to the PML. For field that are not traveling perpendicular to the PML however, there will be reflection. The accuracy of the PML is also limited by the amount of PML layers used. Most commercial FDTD software packages have built-in PML boundary conditions, the reflectivity of which can be selected by the user depending on the needs of a particular problem.

Limitations of FDTD

The most important limitation of FDTD is the computation cost. Meshing a three dimensional space requires a lot of memory. There are also a lot of computations to be done, depending on the amount of cells used. Due to the fact that both the accuracy of the computations and the accuracy of the way the structure is resolved depend on the size of the grid, the amount of cells needed for an acceptable accuracy can be very high. This is especially relevant when working with rapidly changing fields, like in metals or at large refractive index-steps. A way to work around this problem is to use a non-uniform grid, meaning the cells are smaller near these regions than at regions where the field is varying slowly. Another way to deal with this problem is making use of symmetry in the structure. Using symmetric and/or anti-symmetric boundary conditions, only a part of the structure needs to be meshed and simulated. A sphere, for example, can be simulated with just an eighth of the actual structure.
In the particular case of waveguide simulation, there is a second problem. Simply adding a random field into the waveguide will lead to very high losses. In order to be sure all waveguide modes are found, a lot of different fields should be tried as input. This problem becomes especially important when the waveguide is not a simple photonic waveguide. For this, we need a technique designed to find all the modes of a waveguide.

2.2.2 Modesolving

Modesolving is a method in which the software solves the modes in a two-dimensional cross-section of the waveguide. There are different methods that can be used, including finite difference (FD) and film mode matching (FMM). As the software used for this thesis only implements these two methods, only the working principles of these methods will be discussed here shortly. Also the limitations are shown.

Film Mode Matching

Film Mode Matching is a method which divides the simulated structure into slices and layers with constant permittivity, as shown in Figure 2.5.
The resulting matrices are then solved using a well optimized algorithm, which makes the FMM method an fast and accurate method for step-index waveguides. However, the usage of FMM is limited, because it cannot be used to simulate waveguide bends. As the PhoeniX implementation of FMM is also unable to simulate metals, the FMM method cannot be used for the simulation of the proposed structure.

Finite Differences

The FD mode-solving technique uses a two-dimensional Yee-cell to mesh the structure[11]. Using the Maxwell equations, a coupled set of differential equations is derived, which can be formulated as a matrix eigenvalue problem [12]. This eigenvalue problem can then be solved

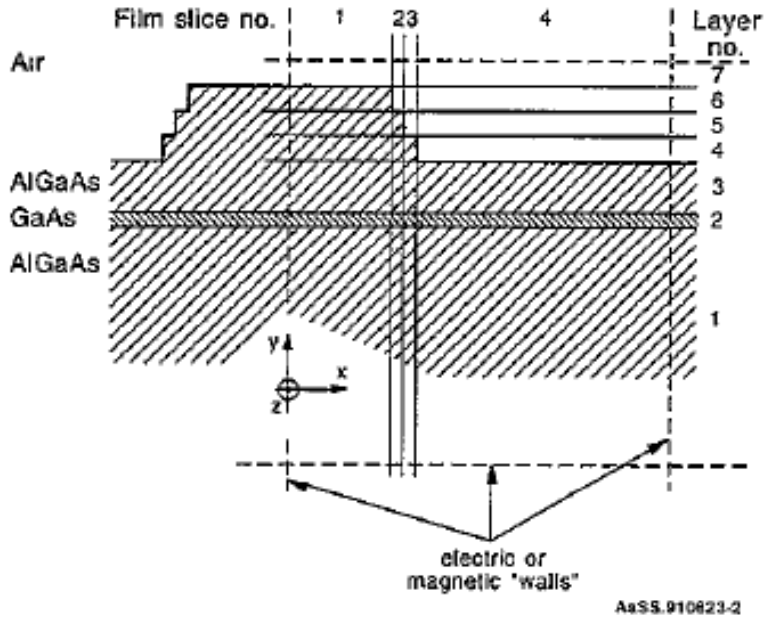


Figure 2.5 – Modelling of a waveguide by FMM.[10]

to obtain the modes of the simulated waveguide. As the FD mode-solving technique can be used to simulate bends of structures with an arbitrary distribution of the permittivity , it is suited well for the simulations of this thesis. A limitation of this technique, is the fact that it uses a two-dimensional mesh. It cannot be said in advance, if this will be a problem for the simulation of an anisotropic ridge waveguide. That is why one of the objectives of this thesis, is to determine whether it is necessary to use 3D FDTD to simulate the anisotropic $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge waveguide.

Chapter 3

Simulations

3.1 Simulated structure

The objective of this thesis is to investigate very sharp bends of a photonic-plasmonic hybrid $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge waveguide. The proposed structure consists of a metal stripe underneath a $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge, as described in this chapter. In order to understand the calculated losses quantitatively, also a benchmark structure is simulated. The benchmark structure is a purely photonic waveguide with the same dimensions as the photonic-plasmonic hybrid waveguide.

3.1.1 Ridge

The structure is a ridge waveguide. A ridge waveguide consists of a ridge on a flat substrate. In a purely photonic waveguide, the field stays inside the ridge due to total internal reflection, as discussed in Section 2.1.1. Because a large part of the field propagates inside the ridge, the ridge is a convenient part of the waveguide to put the active material. This way, a high amplification rate can be achieved. The used gain material is $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$, as introduced in Chapter 1. The material properties at λ=1.55 µm are shown in Table 3.1. The optical axes are defined as in Figure 1.1. The propagation direction (z-axis) is chosen as the Ng axis, with the Np axis pointing vertically upwards. In all simulations, the simulated bend radius is defined as the distance between the center of the bend and the outer surface of the ridge.

Property	Direction	Value
Refractive index	Nm axis	2.01495
	Np axis	1.97954
	Ng axis	2.06182
Yb doping percentage	-	0.475%
Gd doping percentage	-	0.447%
Lu doping percentage	-	0.078%

Table 3.1 – Properties of $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ at λ=1.55 µm.

3.1.2 Metal

As discussed in section 2.1.4, efforts have been made to combine the advantages of photonic and plasmonic waveguides. One way to implement this principle is a stripe of metal underneath the ridge of a standard ridge waveguide. Because the plasmonic character introduces losses, it is expected that for a straight waveguide (or large bends) , the losses are quite high. But for sharp bends, the radiative bending losses are dominant. These losses are expected to decrease because of the presence of the metal. A possible explanation is that the better confinement of the mode will cause smaller difference in speed of the electromagnetic fields in the center and in the outer part of the mode. As discussed in section 2.1.2, this speed difference is the cause of the radiative bending losses. Because the dominant losses decrease, it is expected that the losses for the hybrid structure will be lower than for the pure photonic structure. In this thesis, gold is used as the metal, because it is commonly used in plasmonics. Because any field inside the metal will be absorbed the gold stripe is made as thin as possible, which is 20 nm for fabrication purposes. The width of the stripe is determined in the optimization process.

3.1.3 Buffer layer

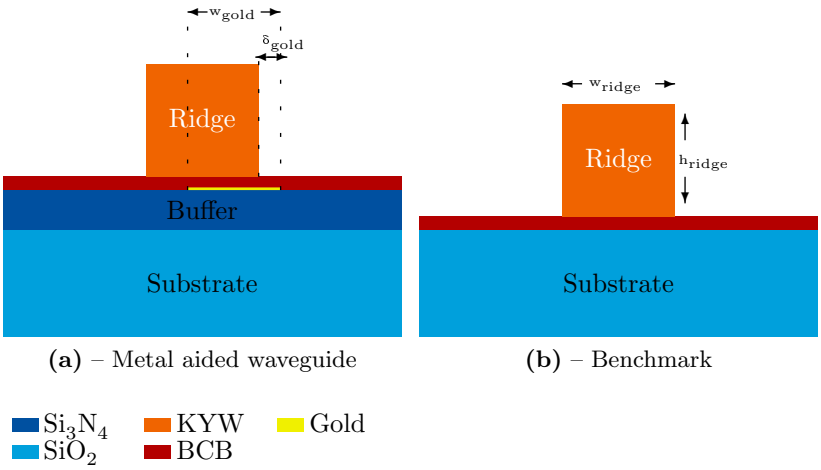
A complication of using a metal stripe underneath the ridge, is the necessity to use a buffer layer between the metal and the substrate. Without a high-index buffer layer below the metal, the electric field would leak into the substrate, leading to high losses. Therefore a buffer layer of silicon nitride (Si_3N_4) is used. This has been previously reported in the literature[13]. One issue arising of this configuration is the bonding of the ridge to the wafer. Because the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ cannot be directly grown onto the buffer layer, an adhesive layer is necessary. Different polymers can be used for this, but in this thesis only benzocyclobutene (BCB) is used.

3.1.4 Structure

In summary, the structure which is the object of this thesis consists of a substrate (SiO_2) with a buffer layer (Si_3N_4) on top of it. On top of the buffer layer is a ridge ($KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$), attached by an adhesive layer (BCB). Below the ridge, on the bottom of the adhesive layer, is a metal stripe. To be able to see if the metal improves the losses for small bends, the results are compared with losses of a benchmark structure. The benchmark structure is chosen as the same structure, without the metal and the buffer layer. The buffer layer is removed because it is not necessary without the metal and it introduces high bend losses. In figure 3.1 the structure of the simulated waveguide is shown, as well as the benchmark, with which the metal aided waveguide is compared.

3.1.5 Optimization

In order to minimize the bendlosses, the structure should be optimized. The ridge is the most obvious part of the waveguide to optimize. Both the height and the width can be varied. Secondly, the width of the gold stripe is object of the optimization process. Some trial simulation showed that giving the metal stripe an offset with respect to the ridge, would result in lower bending losses. Therefore, the offset of the gold stripe will be subject of the optimizations as well.



(a) – Metal aided waveguide **(b)** – Benchmark

Figure 3.1 – Structure of metal aided waveguide and benchmark

3.2 Simulation methods

3.2.1 PhoeniX

The FieldDesigner module of PhoeniX B.V. was utilized to find the different modes of the structure under study. This module is able to calculate the modes in 2D using the FMM method or FD method, the latter using a vectorial or a semi vectorial method. Because the FMM method cannot be used to calculate bend modes and bending losses, it is necessary to use the FD method in the calculations for this thesis. To achieve high accuracy and to be able to calculate the hybrid modes, the full vectorial method is used. The calculation cost is not an issue in this two-dimensional simulation.

Simulation parameters

Before being able to perform the actual simulations, convenient parameters need to be found. The memory requirements limit the amount of cells which can be used. For accuracy, a very fine grid is needed, especially around the metal, so a smallest possible simulation window must be found. For a simulation window with a height of 3.7 μm, a width of 4.0 μm and an 129x129 grid, the software was able to calculate the modes. The buffer layer was divided into 30 sublayers, while the gold and the adhesive layer where divided in 20 sublayers. The best way to check if the calculated results are correct for the used grid size, is to do a convergence test. If the results do not change when making the grid finer, the chosen grid size was good enough. Figure 3.2 shows that for the four found modes, the results are consistent for a large range of grid sizes.
Although there was not manually specified any gradient of the grid, the actual used grid is not uniform. This is, because the PhoeniX software automatically adjusts the size of the cells near the edges of simulated objects and large refractive index-steps. It also enforces at least one grid cell inside every object. This is a convenient tool to ensure a finer grid in the metal and the adhesive and buffer layer, where the field is changing very rapidly. To enforce a finer grid, it is only necessary to build the metal and the adhesive and buffer layer out of a number of sublayers. Because there will be at least one grid cell inside each sublayer, a finer grid is used at the necessary locations automatically. This way, it is not necessary to use a very fine grid throughout the whole simulation region.

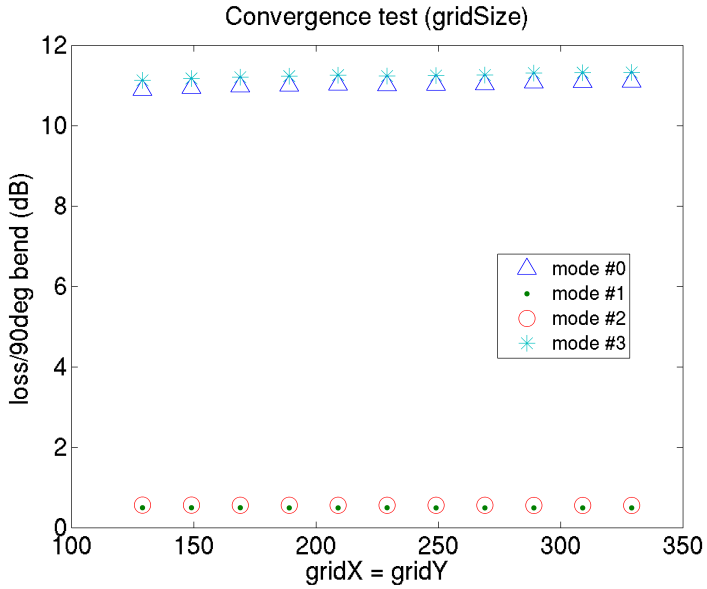


Figure 3.2 – Convergence of the results using different sizes of the grid.

The PhoeniX software is not capable of simulating anisotropic materials. When the refractive index of the material is chosen, it is assumed the mode is purely TE or TM. Then the index is used in the direction where the electric field is directed. The results are expected to differ not too much from the real, anisotropic case. Of course this only holds for the pure TE and TM modes, for hybrid modes differences are expected to be larger.
An advantage of the PhoeniX software is the advanced scripting language, which allows for easy processing of a larger amount of simulations. This means that the optimization of the structure can best be done therewith. The used scripts can be found in Appendix A.

3.2.2 Lumerical MODE

Because the PhoeniX software uses an isotropic model in its simulations, the results should be checked with simulation software which is capable of using materials with different indices in different directions.Therefore, the same simulations are done with the modesolving software of Lumerical, which is capable of using different indices for both directions perpendicular to the propagation. But before the waveguide is simulated using an anisotropic model, the results of the PhoeniX simulations should be confirmed. Otherwise, the two-axis model cannot be compared with the isotropic case. If the PhoeniX results are reproduced, the two-axis model will be simulated and the results will be compared with the isotropic model.

Simulation parameters

The simulation window in the Lumerical MODE simulation is chosen the same as in the PhoeniX simulation. The grid is chosen similar to the PhoeniX grid: the overall grid is chosen uniform, but at the metal and the buffer layer, like the PhoeniX grid, the mesh is overridden to be finer.
In contrast with PhoeniX, it is not necessary to adjust the offset of the radius in order to

find the modes. The Lumerical MODE software is able to sweep over the different radii and find the correct modes by finding the mode which has the largest overlap with the straight mode. This method proved to work well enough. The scripts used for the Lumerical MODE simulations can be found in Appendix B.

3.2.3 Lumerical FDTD

Although it is possible to use two different refractive indices in different directions, the Lumerical MODE solver is still a two dimensional solver. Because the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ is anisotropic in three dimensions, it is necessary to confirm that the simulated results hold for the three dimensional case. This can be done using the 3D FDTD solver of Lumerical. Before the anisotropic simulations are performed, the isotropic results need to be confirmed again.
The set-up for the FDTD simulation is a little different from the MODE simulations. After the structure is built, a mode source is added which injects the mode into the waveguide. Because power monitors can catch a part of the radiating field, the results are averaged over a three-quarter bend. This way, the results should be closer to reality than by just simulating a one-quarter bend. At the end of the three-quarter bend, a straight waveguide is added to ensure the fields propagate out of the simulation region. Figure 3.3 shows the set-up for the Lumerical FDTD simulations.

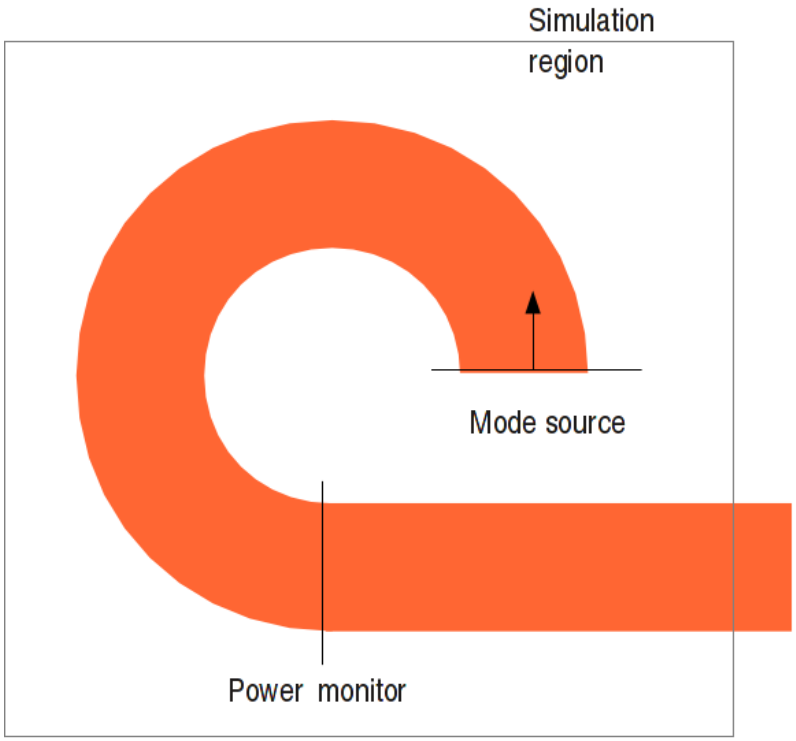


Figure 3.3 – A schematic top view of the set-up for the Lumerical FDTD simulations

Simulation parameters

Because the three-dimensional FDTD simulations are very resource consuming, it is really important that the simulation window is small enough. The size of the cells is more complicated than for the mode solving simulations. In order to achieve stable results, the light should not travel more than half a cell during a time step. Therefore, reducing the size of the cells would implicate reducing the time step. This means that the simulation time is inversely proportional to the fourth power of the cell size, instead of the expected third power. If the cells need to be very small, like around the metal, careful selection of simulation window size is necessary. As the minimal size of the simulation region differs between different simulated structures, this has partly to be done by trial-and-error. As the results converged for a uniform cell size of 0.016 µm in every direction, using a grid size of 0.0016 µm in vertical direction in the gold layer and 0.0032 µm in the adhesive layer, these grid dimensions where used in the simulations.

3.2.4 Overview of simulations

To summarize, Figure 3.4 shows the relation between the different simulations performed in this thesis. The PhoeniX FieldDesigner module uses only one optical axis. The calculated losses are compared with the losses calculated with an isotropic model using Lumerical MODE. After this, the losses calculated with the isotropic model are compared with a two-axial anisotropic model in Lumerical MODE. Then Lumerical FDTD is used to calculate the losses using an isotropic model of the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$. If these results are close to the results for the isotropic model in Lumerical MODE, an anisotropic model of the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ is implemented in the Lumerical FDTD software and used to calculate the losses for bends in anisotropic $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$.

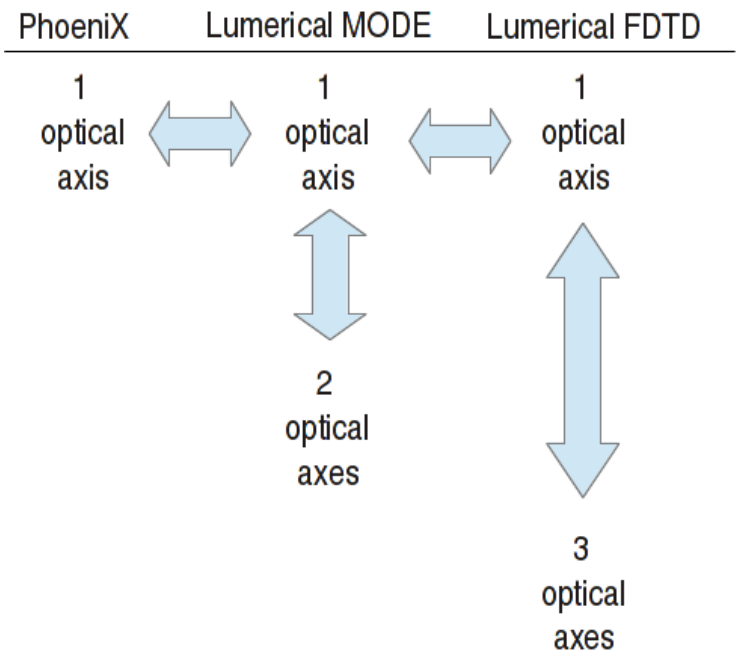


Figure 3.4 – The relation between the different simulations performed in this thesis

Chapter 4

Results

4.1 PhoeniX

As discussed in the previous chapter, FieldDesigner, the mode solver of PhoeniX B.V., permits to calculate the different modes of a given structure for both straight and bend waveguides. As the FMM-solver of PhoeniX FieldDesigner cannot handle bends nor metallic layers, the finite differences method needs to be used. In Section 4.1.1, the modes of both the benchmark structure and the proposed un-optimized structure will be calculated for a large radius of curvature. In Section 4.1.2 the structure will be optimized to minimize bending losses for a radius of 1 micrometer. The propagation loss for the different modes as a function a radius are then calculated

4.1.1 Modes of the proposed structures

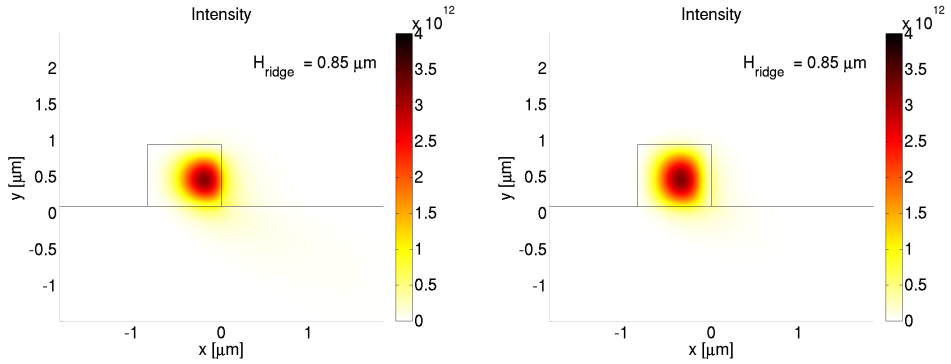
In this section, the modes of the waveguide will be calculated. This was done by sweeping over a range of effective indices. After searching trough all the spurious modes, only a few modes were found to be guided.

Modes of the benchmark structure

In Figure 4.1 the power profiles of the calculated modes are displayed for the benchmark waveguide of Figure 3.1 for a radius of 2 μm. Benchmark mode #1 is the fundamental TM mode of the waveguide. The mode is shifted to the outside of the bend due to the curvature of the waveguide, but a large part of the mode is still in the ridge. Benchmark mode #2 is the fundamental TE mode of the waveguide. The mode is shifted to the outside of the bend, but less than the TM mode. This is probably because the used index for the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge is higher for the TE (2.015) than for the TM (1.980) simulation due to the anisotropy of the crystal.

Modes of the proposed metal-aided structure

In Figure 4.3 the intensity profiles of the calculated modes are displayed for the metal aided waveguides. The modes are numbered from 0 to 3. Mode #0 is a hybrid TE and TM mode. Most of the field is concentrated at de interface between the metal and the adhesive layer, so it can be characterized as a plasmonic mode. This mode is expected to have high losses, for both large and small radius, because of the high adsorption in the metal, as



(a) – Benchmark Mode #1 **(b)** – Benchmark Mode #2

Figure 4.1 – Different modes for benchmark waveguide at a radius of 2 µm, λ=1.55 µm, wridge = hridge = 0.85 µm.

discussed in section 2.1.3. Mode #1 is analogue to the TM mode found for the benchmark waveguide. The mode is mainly photonic, but because the TM field is able to couple into a plasmonic mode at the surface of the metal, a part of the mode is plasmonic. Because of this, a smaller part of the mode is inside the ridge. Since the ridge contains the gain material, this mode is less ideal for amplification of the signal than the purely photonic TM mode found for the benchmark waveguide. Mode #2 is analogue to the TE mode found for the benchmark waveguide. It is very similar, except for a small part of the mode below and at the edges of the metal. The field underneath the metal has a low intensity. The field at the edges however, is very confined. Intuitively, this looks like a plasmonic mode, but as discussed in Section 2.1.3, this is not possible for a TE mode. There is however a possible explanation for this phenomena. As the interface between metal and adhesive layer is turned 90 degrees with respect to the surface at the top, the field is a TM field near the edges of the metal. Therefore it is possible to have a plasmonic character. In Figure 4.2, the electric field vectors perpendicular to the propagation direction are shown for the TE and TM mode near the metal-dielectric interface. Also the electric field of the waveguide TE mode is shown. It is clear that the electric field is a TE field at the top surface of the metal, but a TM field at the right edge of the metal.
Because only a small part of the mode turns into a plasmonic mode, it is expected this does not cause too much extra losses. Mode #3 is a TM mode. The field is almost completely near the metal, having a plasmonic character. This mode is expected to have high losses, comparable with the losses of mode #0, because this mode has the same plasmonic character.

The real part of the effective indices of all the found modes at different radii are plotted in Figure 4.4. It can be observed that the effective indices for the benchmark modes are slightly different from their corresponding metal aided counterparts.

4.1.2 Optimization

As mode #2 has the largest overlap with the active region, this mode is the best suited mode to use in the metal aided structure in order to maximize the available optical gain. Therefore, this mode is used in the optimization process. Because the metal only is expected to improve the losses for very small radius, the waveguide is optimized at a radius of 1 µm.

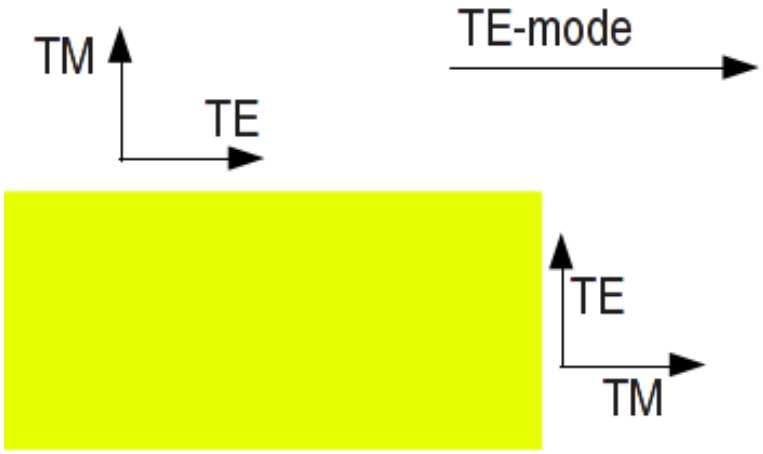
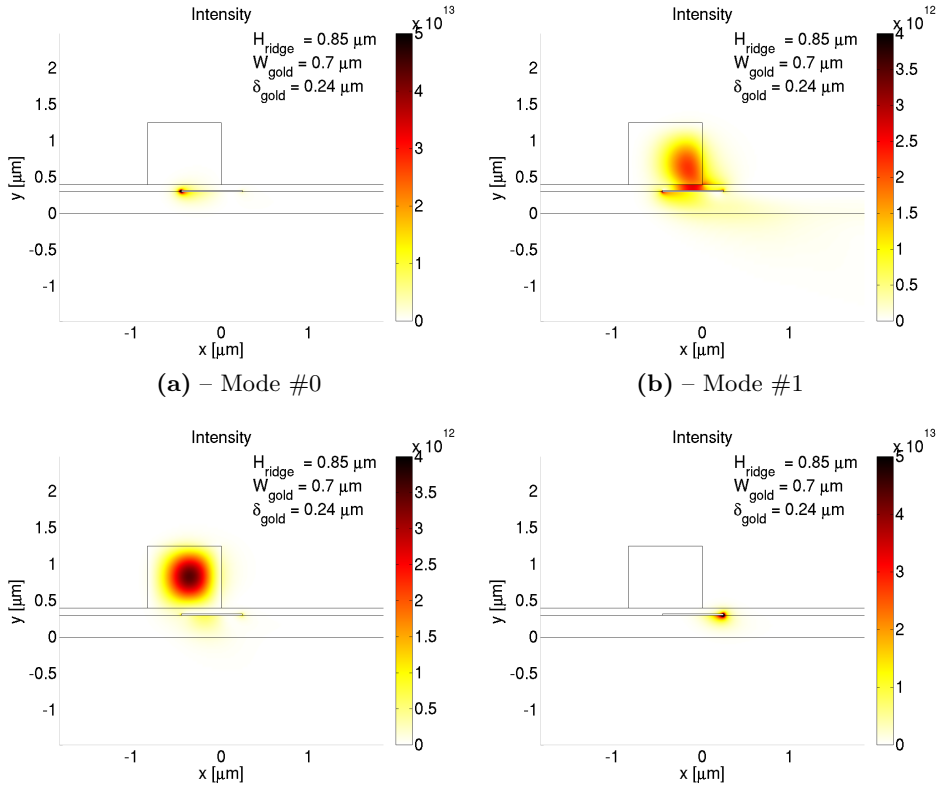


Figure 4.2 – Schematic view of the electric field vectors near the metal-dielectric interface.



(a) – Mode #0

(b) – Mode #1

(c) – Mode #2

(d) – Mode #3

Figure 4.3 – Different modes for the metal aided waveguide at a radius of 2 µm, λ=1.55 µm, wgold=0.7 µm, δgold=0.24 µm, wridge = hridge = 0.85 µm. Note the difference of the scale for the plasmonic and photonic modes

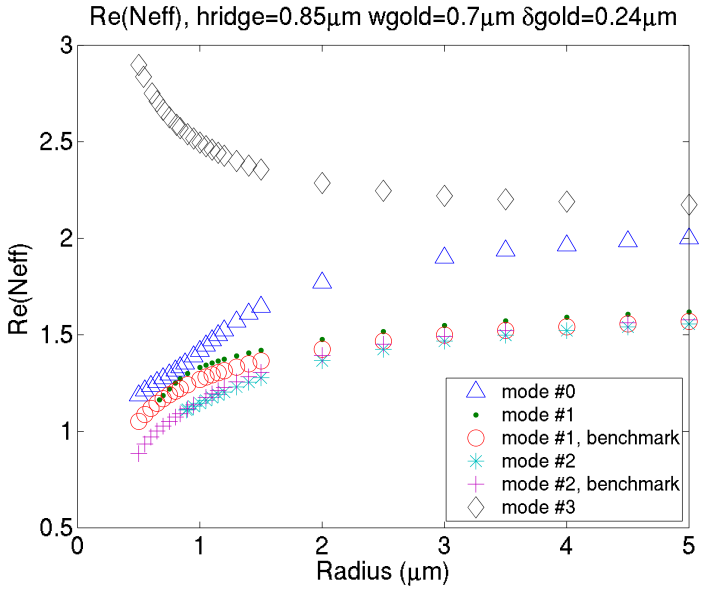


Figure 4.4 – Real part of the effective indices of the modes at different radii

Four parameters are chosen for the optimization: the widht and height of the ridge, the width of the metal stripe and the displacement of the metal stripe. These parameters are defined in section 3.1.4. Optimizing for all these parameters at the same time would increase dramatically the amount of necessary calculations. The effect of the different parameters was analyzed separately.

Effect of the height of the waveguide ridge

The height of the waveguide has significant influence on the losses, except for a radius of about 2 µm, as can be seen in Figure 4.5. For a radius of 1 µm, the losses are lower for a higher waveguide. We do not want to excite higher order modes, so the waveguide cannot be too high. Because the losses increase significantly for a height of 800 nm, we choose a height of 850 nm. For this height, the waveguide performs well for the whole region of interest. In figure 4.6, the power profiles at different radii are shown for three different values of the height of the ridge. It can be observed, that for a lower ridge, the mode is pushed more toward the middle (in the images: the left) of the bend. At small radius, this will lower the losses, because a smaller part of the mode suffers from the radiative bend losses. At higher radius, the mode is pushed more toward the inside of the bend. The interaction with the surface at the inside of the bend causes losses. The higher the ridge, the lesser the mode is pushed toward the center of the bend, so the interaction between the mode and the inner surface is smaller. This causes lower losses, as observed in Figure 4.5. From this point of view it is expected that at small radius the losses are smaller for a lower ridge. However, this is not the case, according to the calculations. A probable explanation for this, is the interaction with the top surface of the ridge. As the ridge is lower, the mode is closer to this surface. This causes additional radiation, increasing the losses for a low ridge. Also, with a lower ridge, the mode has a larger plasmonic part. This can introduce additional adsorption losses.

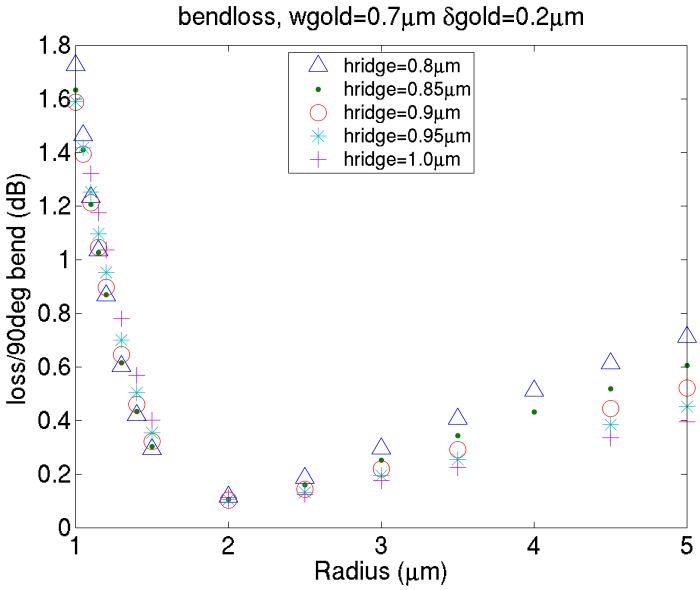


Figure 4.5 – Evolution of the loss per 90 degree bend as a function of bend radius for several ridge heights.

Effect of the width of the waveguide ridge

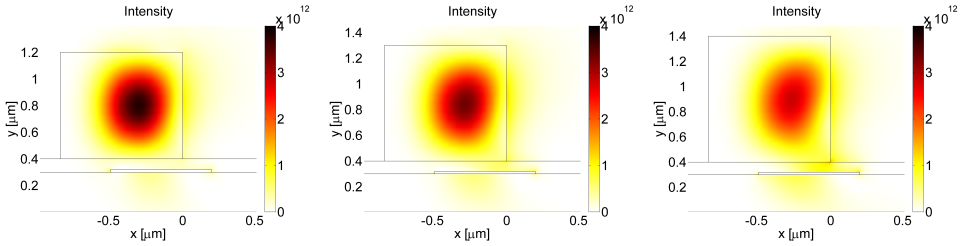
The width of the waveguide has less influence on the losses for very small bends, as can be seen in Figure 4.7. At larger radius the width of the ridge has a larger influence on the losses per 90 degree bend. The smaller the radius, the more the mode is shifted to the outside of the bend. Because the mode has less and less interaction with the interface at the inside of the bend, the difference between a wide and a small waveguide disappears as the radius gets smaller. Since the region of main interest in this work is around 1 micrometer, the width of the ridge is arbitrarily chosen the same as the height of the waveguide, namely 850 nm.

Effect of gold width and gold displacement

After optimizing for the ridge dimension, the gold dimensions can be optimized. The results for a range of values for the gold width and gold displacement are displayed in Table 4.1. The minimum losses were found to be 1.106 dB/90 degree bend, for a width of 0.700 µm and an offset of 0.24 µm.

Because the optimal structure could be for a width of more than 0.7 µm, some more geometries where investigated. The results are shown in Table 4.2. The same minimum is found here, so in the following simulations the used geometries for the gold stripe are a width of 0.700 µm and an offset of 0.24 µm.

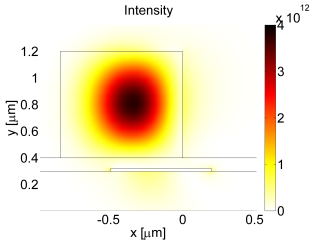
In Figure 4.8, the power profiles for different values of the gold width and displacement are shown. It can be observed, that when both the displacement and the width of the gold are small, there is a high intensity near the gold at the inner side of the bend. This causes losses, due to the adsorbtion by the metal. The difference between the profiles in Figures 4.8f and Figure 4.8i is hard to observe, but the losses for a gold width of 0.8 µm are higher. As the extra gold underneath the ridge does not cause a more advantageous distribution of



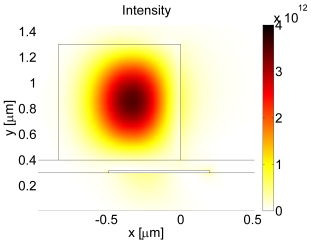
(a) – Radius=1.1 µm, Hridge=0.8 µm

(b) – Radius=1.1 µm, Hridge=0.9 µm

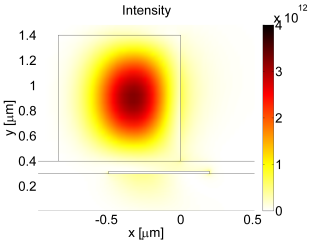
(c) – Radius=1.1 µm, Hridge=1.0 µm



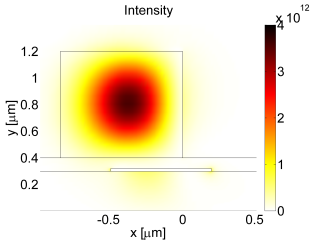
(d) – Radius=1.5 µm, Hridge=0.8 µm



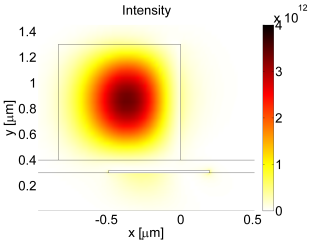
(e) – Radius=1.5 µm, Hridge=0.9 µm



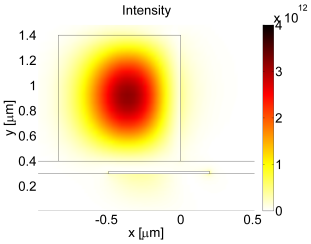
(f) – Radius=1.5 µm, Hridge=1.0 µm



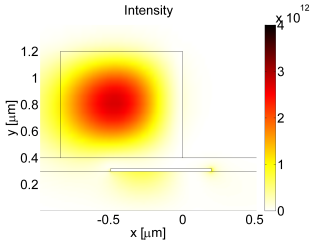
(g) – Radius=2.0 µm, Hridge=0.8 µm



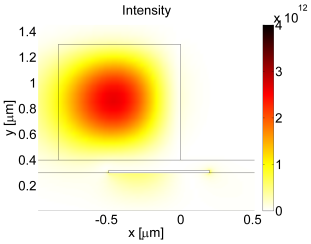
(h) – Radius=2.0 µm, Hridge=0.9 µm



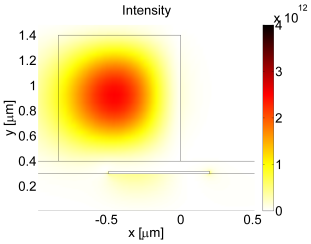
(i) – Radius=2.0 µm, Hridge=1.0 µm



(j) – Radius=4.5 µm, Hridge=0.8 µm



(k) – Radius=4.5 µm, Hridge=0.9 µm



(l) – Radius=4.5 µm, Hridge=1.0 µm

Figure 4.6 – Power profiles at different radii for different values of the height of the ridge for the TE mode. The width of the gold is 0.7µm and the displacement of the gold is 0.2 µm. λ=1.55 µm

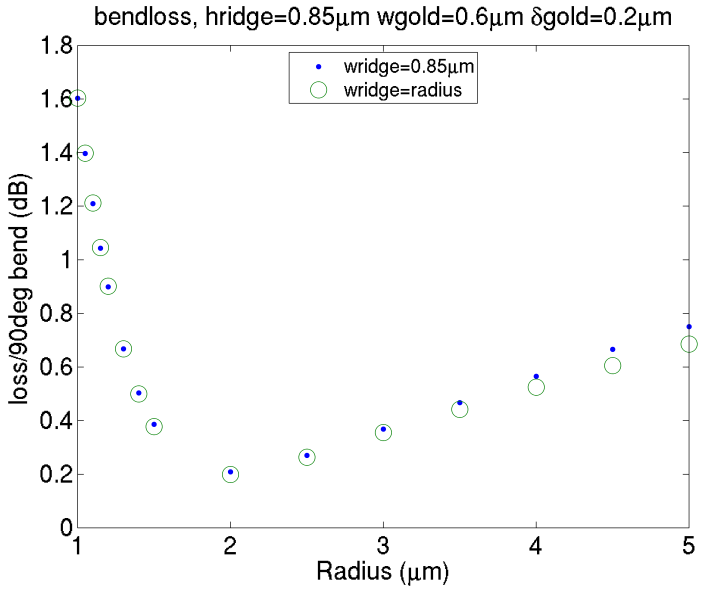


Figure 4.7 – Evolution of the loss per 90 degree bend in dB as a function of bend radius for several ridge widths. The smaller the radius, the lesser the effect of the width of the ridge on the bend loss per 90 degree bend.

wgold (µm) \ δgold (µm)	0.10	0.12	0.14	0.16	0.18	0.20	0.22	0.24	0.26	0.28	0.30
0.50	1.408	1.355	1.306	1.265	1.239	1.235	1.266	1.339	1.465		
0.52	1.396	1.342	1.291	1.246	1.215	1.206	1.230	1.298	1.419		
0.54	1.389	1.334	1.282	1.234	1.198	1.183	1.120	1.261	1.379	1.561	
0.56	1.384	1.330	1.277	1.227	1.186	1.165	1.175	1.230	1.343	1.526	
0.58	1.379	1.327	1.275	1.224	1.179	1.152	1.155	1.203	1.311	1.494	
0.60	1.375	1.325	1.275	1.223	1.176	1.143	1.138	1.179	1.282	1.465	1.736
0.62	1.375	1.322	1.275	1.225	1.177	1.139	1.126	1.158	1.254	1.436	1.715
0.64	1.396	1.319	1.274	1.228	1.180	1.138	1.118	1.139	1.227	1.405	1.698
0.66	1.492	1.323	1.272	1.232	1.186	1.142	1.114	1.124	1.120	1.372	1.677
0.68	1.734	1.355	1.272	1.234	1.194	1.150	1.115	1.112	1.173	1.332	1.632
0.70	1.965	1.431	1.278	1.235	1.203	1.162	1.122	1.106	1.147	1.287	1.571

Table 4.1 – Bendlosses in dB/90 degree bend for different dimensions of the gold for the TE mode at λ=1.55 µm, with hridge = wridge = 0.85 µm. Minima of each row and column are marked with light and dark grey respectively.

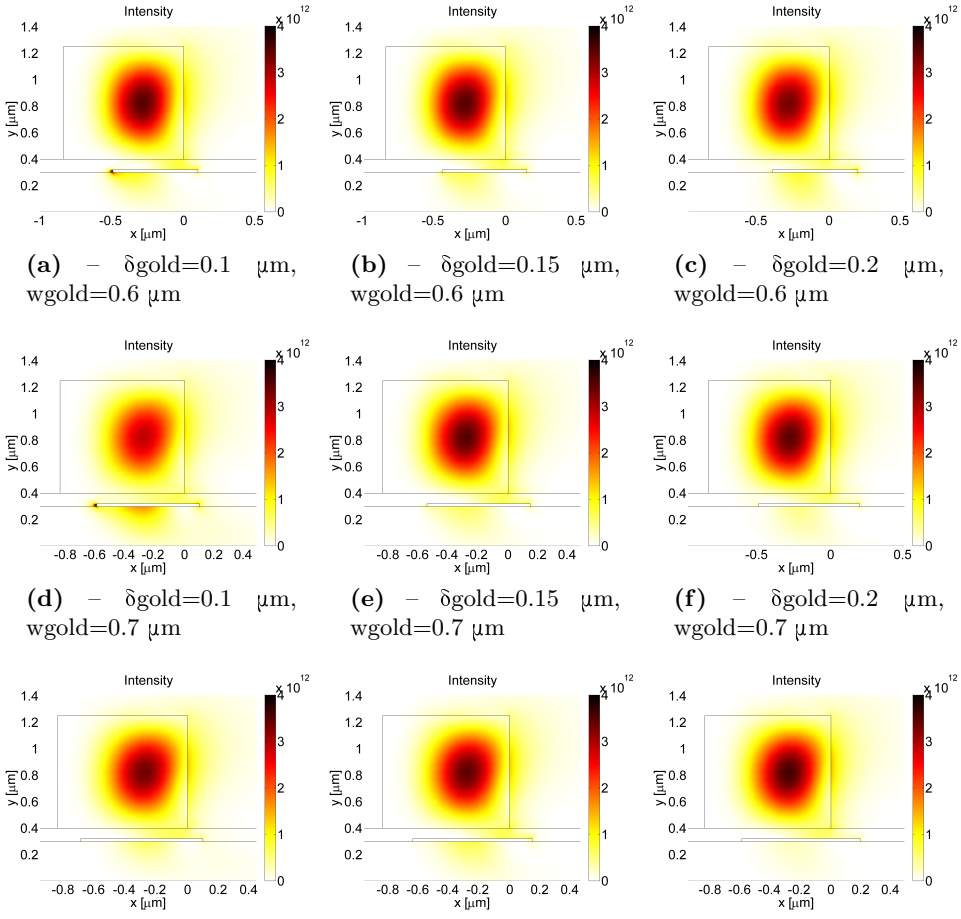
wgold (μm)	δgold (μm) 0.22	0.23	0.24	0.25	0.26
0.70	1.122	1.109	1.106	1.117	1.147
0.72	1.137	1.119	1.107	1.107	1.124
0.74	1.157	1.137	1.120	1.109	1.111
0.76	1.177	1.160	1.142	1.124	1.113
0.78	1.192	1.181	1.167	1.150	1.132
0.80	1.200	1.195	1.187	1.175	1.160

Table 4.2 – bendlosses in dB/90 degree bend for some more geometries of the gold. Minima of each row and column are marked with light and dark grey respectively.

the electric field, there indeed should not be an improvement of the losses. Because there is more metal near the electric field, there could be additional adsorbtion by the metal, causing the propagation losses to increase.

4.1.3 Losses

Using the optimized structures, the bendlosses for all the modes are calculated at different radii. The bendlosses for the different modes are plotted against radius in Figure 4.9. As expected, the plasmonic modes have high losses, even at relatively large radius. This is a disadvantage of the proposed structure. When designing the device, the excitation of these modes should be minimized, as they would add considerably to the losses. It gets more interesting when we compare the fundamental TM and TE mode of both structures. Benchmark mode #1, the TM mode, has low losses at large radius as it can be expected from a mode of a dielectric waveguide dominated by propagation losses. When the radius gets smaller, the losses increase dramatically, as expected due to the contribution of the bend losses. The purpose of the metal aided structure is reducing this effect. As expected, higher losses are observed for the metal aided structure at larger radius. This is because the metal introduces significant absorption losses. At smaller radius the losses are higher than the benchmark losses as well. The cause of this is that a considerable part of the mode has a plasmonic character, as discussed in Section 4.1.1. With respect to the losses, the metal aided structure presents no improvement for the TM mode.
The TE mode shows the same behaviour at large radius. The metal introduces extra losses and the losses for the benchmark structure are very low. However, somewhere between a radii of 2.5 μm and 3.0 μm, the curves of the benchmark and the metal aided structure cross. At smaller radius, the bendlosses for the metal aided TE mode are lower than for the benchmark one. The reason for this behavior could be that thanks to the presence of the metal layer the mode is shielded from the outside of the bend. As can be seen in figures 4.1 and 4.3, the TE mode does not have the same affinity to be around the metal as the TM mode. This is because the plasmonic mode is not supported for TE. Instead of absorbing the electric field via the oscillation of the electrons, the metal reflects the field. This prevents the field from leaking into the substrate and keeps the field more to the middle of the bend. The latter causes the radiation losses to decrease significantly.



(a) – δgold=0.1 µm, wgold=0.6 µm

(b) – δgold=0.15 µm, wgold=0.6 µm

(c) – δgold=0.2 µm, wgold=0.6 µm

(d) – δgold=0.1 µm, wgold=0.7 µm

(e) – δgold=0.15 µm, wgold=0.7 µm

(f) – δgold=0.2 µm, wgold=0.7 µm

(g) – δgold=0.1 µm, wgold=0.8 µm

(h) – δgold=0.15 µm, wgold=0.8 µm

(i) – δgold=0.2 µm, wgold=0.8 µm

Figure 4.8 – Power profiles at a radius of 1 µm for different values of the width and displacement of the gold for the TE mode at λ=1.55 µm. The height of the ridge is 0.85 µm and the width of the ridge is 0.85 µm as well.

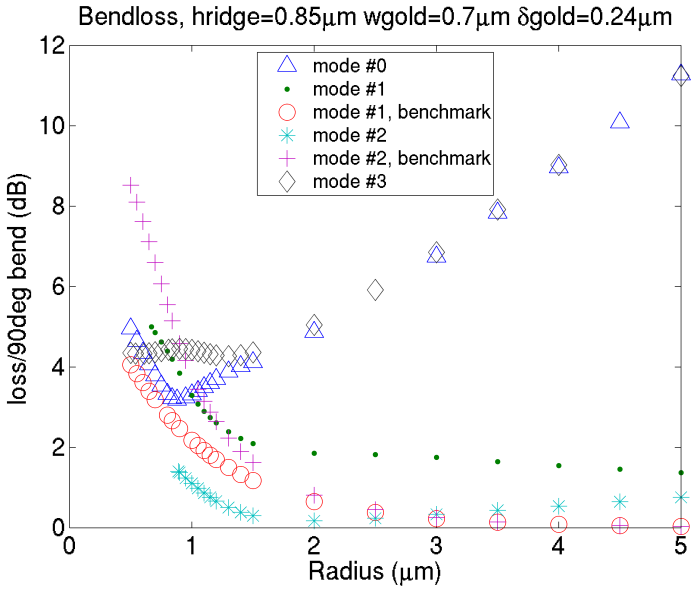


Figure 4.9 – Comparison between the benchmark en the metal aided waveguide for all modes, wridge = hridge = 0.85µm, δgold=0.24 µm, wgold=0.7 µm, λ=1.55 µm

4.2 Lumerical MODE

In order to make sure the results are not an artefact of the PhoeniX simulation software, the results presented in Section 4.1 are reproduced with Lumerical MODE. If successful, the simulations are repeated with the refractive index defined separately in two directions, to verify if the anisotropy of the material poses a problem for the bend loss calculations. Because improvement of the losses is only found for the TE mode and furthermore, this mode is the most interesting from the point of view of the achievable gain, this is the only mode used in this section.

4.2.1 Reproduction of PhoeniX FieldDesigner results

In Figure 4.10 the losses calculated with Lumerical MODE are compared with the losses calculated using PhoeniX FieldDesigner in Section 4.1. The results are very close to each other. The curves for the metal aided structure slightly. Despite this deviation, the curves have the same characteristics. The minima of the curves are both around a radius of 2 µm. The radius at which both waveguides have the same losses, is for both curves more or less the same. It can be concluded that although the results change slightly quantitatively, they are the same qualitatively.

Comparing the calculated effective indices of the mode, the results are even closer. The calculated indices using both Lumerical MODE and PhoeniX are compared in Figure 4.11. For both structures the Lumerical MODE software calculates slightly lower effective indices than the PhoeniX software.

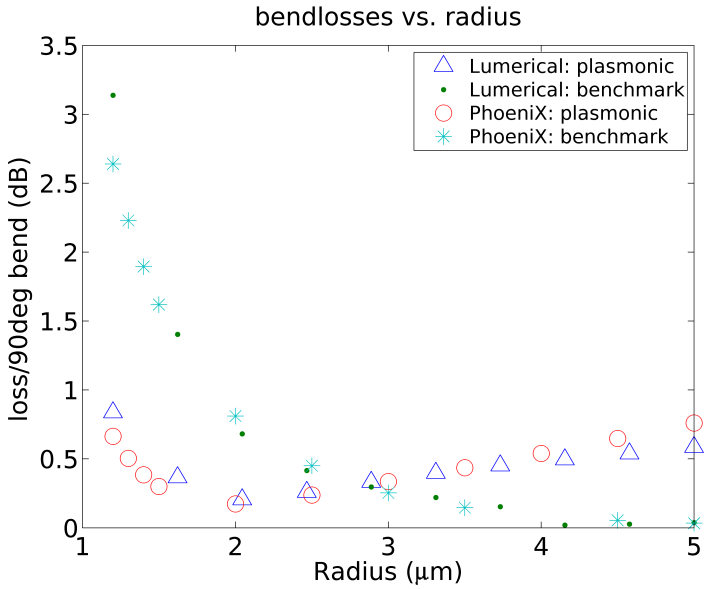


Figure 4.10 – Comparison of bendlosses between Lumerical MODE and PhoeniX for TE mode (mode #2)

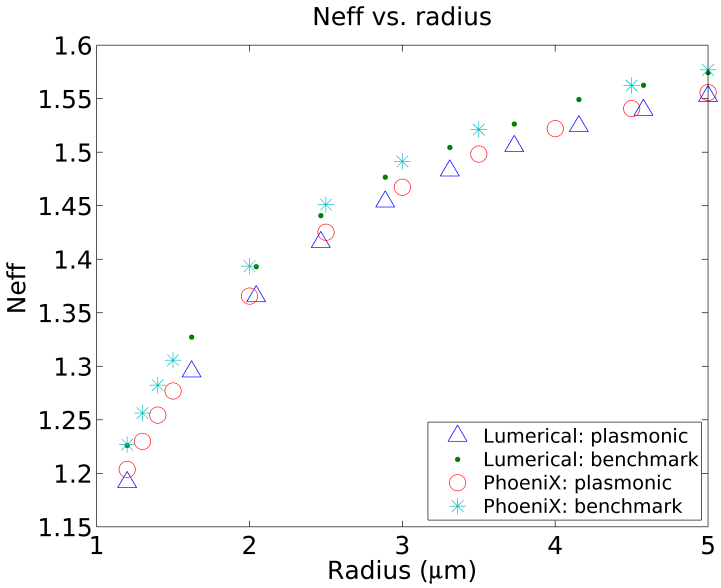


Figure 4.11 – Comparison of the real part of the effective refractive index between Lumerical MODE and PhoeniX for TE mode (mode #2)

4.2.2 Comparison with two-axis model

Now the PhoeniX results are confirmed and reproduced with Lumerical MODE, they can be compared with the two axes model. To be able to really see only the influence of the anisotropy, the results for the two axes model are compared with the Lumerical MODE results. The calculated bendlosses are compared in Figure 4.12. As can be observed, the results for the metal aided waveguide are very close to each other. The results for the benchmark waveguide differ slightly more, but not significant. From this result it can be concluded that the anisotropy, as far as it can be impemented in the two-dimensional mode solver, has no significant effect on the losses at very small radius.

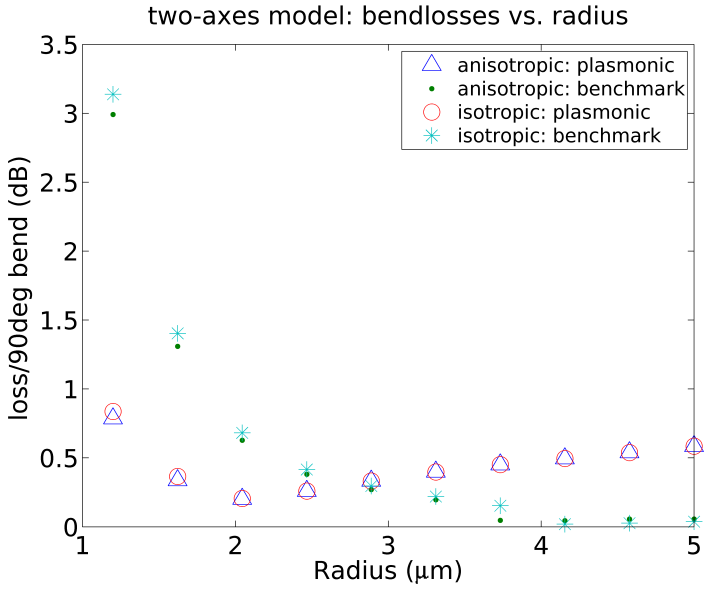


Figure 4.12 – Comparison of bendlosses between isotropic and two-axis model for the TE mode at λ=1.55 µm. wridge = hridge = 0.85 µm, δgold=0.24 µm, wgold=0.7 µm.

Also the effective indices of the mode are compared. The results are shown in Figure 4.13. The effective indices for the anisotropic simulation are higher than for the isotropic case. This is because the index in the added direction, is higher than the index used in the isotropic simulation. Apart from this little shift upward, there is no significant difference between the calculated indices.

In total, the conclusion which can be drawn here, is that the anisotropy has no significant influence on the performance of the waveguide. One has to keep in mind, however, that the mode solver of Lumerical is a two-dimensional solver, while the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ material presents three optical axis. The effect of the three-dimensional anisotropy of the material cannot be predicted with this solver.

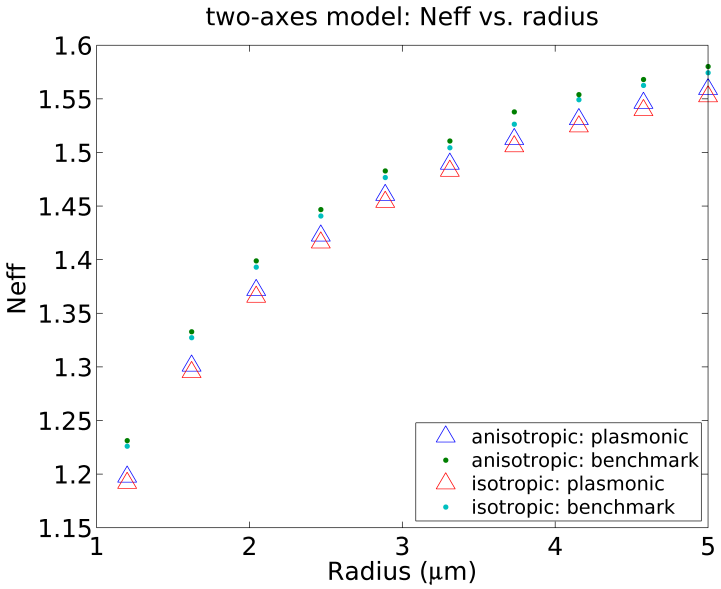


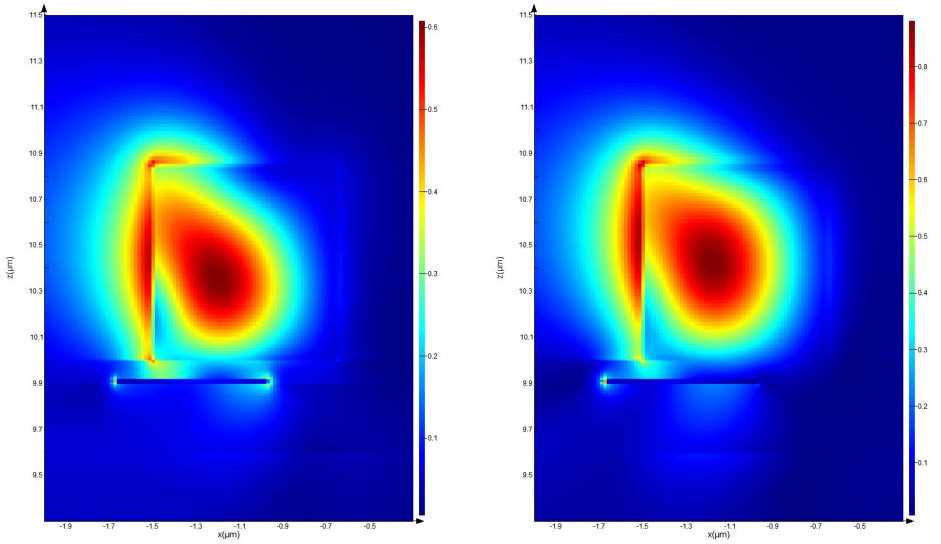
Figure 4.13 – Comparison of the real part of the effective refractive index between isotropic and two-axis model for the TE mode at λ=1.55 μm. wridge = hridge = 0.85 μm, δgold=0.24 μm, wgold=0.7 μm.

4.3 Lumerical FDTD

In order to be able to calculate the influence of the threedimensional anisotropy of the waveguide, a threedimensional solver needs to be used. For this cause, Lumerical FDTD is used. Again, first the isotropic results are reproduced. If these are confirmed, the anisotropy is added and the results are compared.

4.3.1 Bend mode calculation launching the mode of the straight waveguide

Lumerical 3-D FDTD simulation setup requires a calculation of the mode before the simulation starts. The simulation then calculates how the field propagates through the simulated space. During a first run of the simulation, the losses came out much higher than expected. Although this was caused by a small error in the simulation set-up, the results are worth noting. Instead of injecting a bend mode into the waveguide, a straight mode was injected. Normally this would cause some extra losses for the coupling between the straight and bend mode. However, the losses calculated with Lumerical FDTD were much higher than expected from normal overlap losses. Figure 4.14 shows power profiles after a quarter bend when a straight respectively a bend mode was injected. It can be observed, that when a straight mode is injected, a plasmonic mode is excited at the edge of the metal. Due to adsorption in the metal, the losses are increased significantly.



(a) – Power profile when a straight mode is injected

(b) – Power profile when a bend mode is injected

Figure 4.14 – Comparison of the power profile after a 90 degree bend when a straight respectively a bend mode is injected.

4.3.2 Reproduction of previous results with bend mode inserted

In order to prevent exciting the wrong mode, with the corresponding amount of induced propagation losses, a bend mode can be inserted into the 3D FDTD simulations. In this case, both the coupling losses and the losses of the unwanted excited lossy mode are not present. Therefore, the expectation is that the calculated losses are the same as calculated with PhoeniX and Lumerical MODE. In Figure 4.15, the results are plotted for the isotropic model of the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge. The losses are compared with the losses calculated with Lumerical MODE. For the benchmark, the results are very close at small radius. At a larger radius however, the FDTD simulation results in significantly higher losses than the MODE simulation. The calculated losses for the metal aided waveguide differ more, but these differences are still not very large. At larger radius, the results are closer than the benchmark results. The FDTD curves cross at more or less the same radius as for the MODE curves. In summary, the FDTD simulations confirm the previous obtained results when a bend mode is inserted by the source.

4.3.3 Comparison of results with three-axis model

It is known now, that the FDTD simulations lead to comparable results as the mode solving algorithms. The next step is to add the anisotropy of the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge. In Figure 4.16, the losses of the anisotropic model are compared with the previously calculated losses of the isotropic model. As can be seen, the losses for the metal aided waveguide are very similar in both models. The main difference is the minimum of the curve, which is at a smaller radius for the anisotropic model. The losses for the anisotropic benchmark waveguide however, are significantly lower than for the isotropic one. This

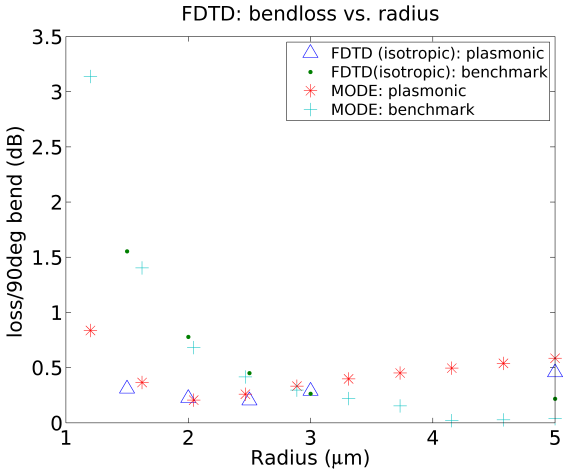


Figure 4.15 – Comparison of calculated bendlosses using Lumerical FDTD with Lumerical MODE results, when TE bend mode is injected at λ=1.55 µm. wridge = hridge = 0.85 µm, δgold=0.24 µm, wgold=0.7 µm.

means that for the anisotropic model the radius at which the metal aided waveguide and the benchmark waveguide have the same losses, is smaller than for the isotropic model. It is still at a radius larger than 2.5 µm and smaller than 3 µm.

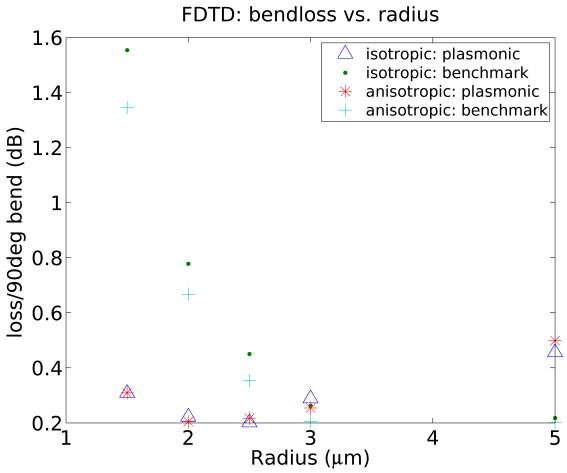


Figure 4.16 – Comparison of calculated bendlosses of the isotropic and the anisotropic model using Lumerical FDTD for the TE mode at λ=1.55 µm. wridge = hridge = 0.85 µm, δgold=0.24 µm, wgold=0.7 µm.

Chapter 5

Conclusion

The possibility of reducing the bend losses of long-range dielectric loaded surface plasmon waveguides with a potassium double tungstate ridge with very sharp bends has been studied in detail and optimized for a radius of curvature around 1 micrometer. Three different integrated optics commercial design packages: FieldDesigner from PhoeniX B.V., Lumerical MODE Solutions and Lumerical 3D FDTD were utilized in order to verify the validity of the results and the influence of the three-axial anisotropy of the potassium double tungstate gain material.

The losses of small bends with a $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge waveguide can be improved for the TE mode using a metal stripe underneath the ridge. In terms of losses, this configuration is worse at larger radius. There also has to be taken into account that a lossy plasmonic mode can easily be excited, causing high losses. It is thus necessary to implement this structure carefully. The bendlosses of the TM mode are worse with the metal, even for very small radius. In total, improvement of the losses can only be achieved for the TE mode at a radius below 2.5 μm.

The anisotropy of the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ ridge does not introduce additional losses. For the purely photonic waveguide the losses are even lower at very small radius. The losses for the metal aided structure are the same as for the isotropic approximation. The results calculated with the PhoeniX and Lumerical MODE software are in a good agreement with the Lumerical FDTD results. Therefore, even for the anisotropic $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ the two-dimensional modesolving software can still be used to simulate and optimize the device.

Chapter 6

Future work

The waveguide will perform better at a free space wavelength of 980 nm. It is unknown if the metal stripe will lower the losses for small bending radius under this condition. As the gain material amplifies at 980 nm, the structure will be much more useful at this wavelength.
Secondly, the influence of the type of metal can be evaluated. Varying the permittivity of the simulated metal could lead to a better understanding of the observed phenomena.
So far, the structure is only tested using simulations. Further work should start with the simulation of a testable device. This device should then be fabricated and tested, in order to confirm the simulations. Secondly, the amplification should be simulated and tested. In the simulations the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ is assumed to be normal dielectric medium. Influence of the dopant concentration on the waveguide performance is unknown.
Besides the structure treated in this thesis, other structures might get improved with the same principle of shielding with a metal strip. Therefore other ridge waveguides or even other types of waveguides should be examined.
Finally, other advantages of the presence of the metal should be investigated. For example modulation of the signal using thermoelectric control of the waveguide dimensions. As the presence of the metal does not lead to a dramatic increase of the losses at small bending radius, this structure would be an excellent approach to implementation of these possibilities.

Acknowledgements

I want to thank dr. Sonia García Blanco for her continuous preparedness to discuss the used methods and the obtained results. Especially in the intensive period of writing the report. The discussions about technical and physical details were a stimulant to think in the right direction and try any possible solution to the encountered problems.
I also want to thank the other people from IOMS, for their willingness to think along about bumps in the process and counter-intuitive results. Especially Mustafa Sefünç for his ideas during subgroup meetings and in the office.
Thirdly, I would like to thank Remco Stoffer for discussing, direct and indirect, my PhoeniX simulation problems and solving some.
Last but not least I want to thank dr. Edwin Carlen from the BIOS group, for letting us use his computer to perform the 3D FDTD simulations. Without this, it would have been quite hard, if not impossible, to obtain accurate results.

Bibliography

[1] D. Geskus, *Channel waveguide lasers and amplifiers in single-crystalline Ytterbium-doped potassium double tungstates*. PhD thesis, Proefschrift Universiteit Twente, Enschede, 2011.

[2] D. Geskus, S. Aravazhi, S. Garcia Blanco, and M. Pollnau, "Giant gain in a rare-earth-ion-doped waveguide," 2011.

[3] D. Geskus, S. Aravazhi, E. Bernhardi, C. Grivas, K. Wörhoff, and M. Pollnau, "Highly efficient gd, lu co-doped KY(WO4)2:Yb3+ planar waveguide laser," 2009.

[4] S. M. Garcia-Blanco, M. Pollnau, and S. I. Bozhevolnyi, "Theoretical study of loss compensation in long-range dielectric loaded surface plasmon polariton waveguides," 2012.

[5] WikipediaCommons. `http://upload.wikimedia.org/wikipedia/commons/5/5d/RefractionReflextion.svg`.

[6] R. Hunsperger, "Losses in optical waveguides," in *Integrated Optics*, pp. 107–128, Springer US, 2009.

[7] "Surface plasmon polaritons at metal / insulator interfaces," in *Plasmonics: Fundamentals and Applications*, pp. 21–37, Springer US, 2007.

[8] WikipediaCommons. `http://upload.wikimedia.org/wikipedia/commons/6/65/Electron_density_wave_-_plasmon_excitations.jpg`.

[9] K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, pp. 302 –307, May 1966.

[10] A. S. Sudbo, "Film mode matching: a versatile numerical method for vector mode field calculations in dielectric waveguides," *Pure and Applied Optics: Journal of the European Optical Society Part A*, vol. 2, no. 3, p. 211, 1993.

[11] Z. Zhu and T. Brown, "Full-vectorial finite-difference analysis of microstructured optical fibers," *Opt. Express*, vol. 10, pp. 853–864, Aug 2002.

[12] Lumerical. `http://docs.lumerical.com/en/mode/ref_eigenmodesolver_physics_main.html`.

[13] T. Holmgaard, J. Gosciniak, and S. I. Bozhevolnyi, "Long-range dielectric-loaded surface plasmon-polariton waveguides," *Optics Express*, vol. 18, p. 23009, Oct. 2010.

List of Figures

Appendix A: PhoeniX FieldDesigner scripts

In this appendix, the scripts written for the PhoeniX FieldDesigner module can be found. Additional libraries which implement the $KYb_{0.475}Gd_{0.447}Lu_{0.078}(WO_4)_2$ and some plotting commands are necessary in order to use this scripts.

file:metal_ aided_ ridge.spt

```
// filename:metal_aided_ridge.spt

//RESET
dsp::clearInfoWin();
res::Clear();
res::plotCloseAll();
sim::clearCS();

// SETTINGS
sys::iniInt("num.fmm.NV",4);
sys::iniInt("num.fmm.DIVFAC",32);
res::SetAccuracy(Double, "%1.15g");
res::SetAccuracy(Complex, "%1.15g+i%1.15g");

// LIBRARY
sys::include("~/lib/"); // add path to libraries here
#include "library_ridge_plasmonic.spt";
#include "library_calculate_bending.spt";

// PATH
string path = "~/results/"; //add path to results here
string fname,logname;

// AUTOMATION
int plotFD = 0; //1: yes, others: no
int checkIndex = 1; //1:yes, others: no

// PARAMETERS SIMULATION
double WL = 1.55;        //um: wavelength
int pol = 0;  //0:TE, 1:TM
double wcalc = 4.75, hcalc = 3.5, xcalc = 0, ycalc = 1;        //um:
    dimensions of the calculation window
```

```
int gridX = 129, gridY = 129; // number of grid points
int NumSampling = 80; //for FMM accuracy
int HybridCriterion = 1; // 1:no Hybrid
int mode_index = 2; // just a number to give the mode a name
int cI_axis = 1; //1:y-direction (x-fixed), others: x-direction
double cI_pos = -0.1; // fixed x or y position for index checking

// BENDING PARAMETERS AND TARGET INDEX TO BE FILED BEFORE EACH SIM
double R_max=1;
double R_min=1;
double offset = 0.0;
complex tindex =1.1996;
int setModeCount=85;
double min_overlap=0.5;
double conf_norm=0.5;

// PARAMETERS STRUCTURE
double wbuffer = 35, wsubstrate = 35;     //um: dimensions of the
    structure
double hgold = 0.02,  hsubstrate = 20;  //um
double hbuffer = 0.3;
double hadhesive = 0.1;                   // adhesive thickness
double hridge = 0.85;
double wridge = 0.85;
double wgold = 0.7;
double shift_gold = 0;
double Y=0, Gd=0.447, Lu=0.078, Yb=0.475;; // Atomic percentages of ions
     in double Tungstate

int numberSubLayersBuffer = 30; // number of sublayers into which the
    buffer is divided into
int numberSubLayersGold = 20; // number of sublayers into which the gold
     layer is divided into
int numberSubLayersAdhesive = 20; //  number of sublayers into which the
     adhesive layer is divided into

double xact=-1.5, yact=0, xact2=1, yact2=1.5; //active region
double xconf=-1.5, yconf=0, xconf2=1, yconf2=1.5; //region where mode
    should be

// FOLLOW ALL MODES
// the following code sweeps the 4 found modes for the metal aided RE:
   KYW waveguide. If only a single mode needs to be investigated, the
    range in the for-loop can simply be adjusted.
var modes[4][3] = {     1, 0, 1.9784,
                        1, 1, 1.62698,
                        0, 1, 1.5996,
                        1, 1, 2.157846};
for (int i=0;i<=3;i++) {
```

```
        pol=modes[i][0]; HybridCriterion=modes[i][1]; tindex=modes[i][2];
            mode_index=i;
        build_ridge_plasmonic(Y,Gd,Lu,Yb,pol,wsubstrate,hsubstrate,
            wbuffer,hbuffer,numberSubLayersBuffer,wridge,hridge,hadhesive
            ,numberSubLayersAdhesive,wgold,hgold,numberSubLayersGold,
            shift_gold);
        logname="result_ridge_plasmonic_mode"+mode_index;
        fname="_mode"+mode_index+"_dgold"+shift_gold+"_wgold"+wgold+"
            _hridge"+hridge+".m";
        calculate_bending(gridX,gridY,wcalc,hcalc,xcalc,ycalc,xact,yact,
            xact2,yact2,xconf,yconf,xconf2,yconf2,NumSampling,R_min,R_max
            ,offset,tindex,WL,pol,HybridCriterion,setModeCount,
            min_overlap,conf_norm,path,fname,logname,plotFD,checkIndex,
            cI_axis,cI_pos);

        build_benchmark(Y,Gd,Lu,Yb,pol,wsubstrate,hsubstrate,wridge,
            hridge,hadhesive,numberSubLayersAdhesive);
        logname="result_ridge_plasmonic_mode"+mode_index;
        fname="_mode"+mode_index+"_hridge"+hridge;
        calculate_benchmark(gridX,gridY,wcalc,hcalc,xcalc,ycalc,xact,
            yact,xact2,yact2,xconf,yconf,xconf2,yconf2,NumSampling,R_min,
            R_max,offset,tindex,WL,pol,HybridCriterion,setModeCount,
            min_overlap,conf_norm,path,fname,logname,plotFD,mode_index,
            checkIndex,cI_axis,cI_pos);
}
```

file:library_ ridge_ plasmonic.spt

```
// filename:library_ridge_plasmonic.spt

// LIBRARIES
#include "library_FieldDesigner_plot_and_polarization.spt";
#include "libKYW.spt";
#include "libKYWdata.spt";
#include "libMatdispersion.spt";
#include "libMatdispersiondata.spt";
#include "library_calculate_bending.spt";

// MATERIALS.
material air { RefractiveIndex() {return 1;} }
sim::AddMaterial(air(),RGB(255,255,255)); //white
sim::AddMaterial(doubleTungstate(), RGB(0,255,255));
sim::AddMaterial(BCB(),RGB(0,192,0)); //green
sim::AddMaterial(gold(),RGB(255,255,0)); //yellow
sim::AddMaterial(SiO2(),RGB(255,0,0)); //red
sim::AddMaterial(Si3N4(),RGB(0,0,255)); //blue

function build_ridge_plasmonic(Y,Gd,Lu,Yb,pol,wsubstrate,hsubstrate,
    wbuffer,hbuffer,numberSubLayersBuffer,wridge,hridge,hadhesive,
    numberSubLayersAdhesive,wgold,hgold,numberSubLayersGold,shift_gold) {
// This function builds a metal aided RE:KYW ridge waveguide
```

```
sim::clearCS();
cs::background(air() );
cs::rectangle(SiO2() MU->[0,0] : wsubstrate,hsubstrate) substrate;

cs::rectangle(Si3N4() ML->substrate.MU+[0,0] : wbuffer,hbuffer/
    numberSubLayersBuffer) buffer_layers;
for(int f=1; f<numberSubLayersBuffer; f++)
        {
        cs::rectangle(Si3N4() ML->buffer_layers.ML+[0,f*hbuffer/
            numberSubLayersBuffer] : wbuffer,hbuffer/
            numberSubLayersBuffer) buffer_layers;
        }

cs::rectangle(doubleTungstate(Y,Gd,Lu,Yb,pol) RL->buffer_layers.ML+[0,
    hbuffer] : wridge,hridge+hadhesive) ridge_gain;

cs::rectangle(BCB() ML->buffer_layers.ML+[0,hbuffer] : wbuffer,hadhesive
    /numberSubLayersAdhesive) adhesive_layers;
for(int f=1; f<numberSubLayersAdhesive; f++){
        cs::rectangle(BCB() ML->adhesive_layers.ML+[0,f*hadhesive/
            numberSubLayersAdhesive] : wbuffer,hadhesive/
            numberSubLayersAdhesive) adhesive_layers;
}
cs::rectangle(gold() RL->buffer_layers.ML+[shift_gold,hbuffer] : wgold,
    hgold/numberSubLayersGold) gold_stripe;
for(int f=1; f<numberSubLayersGold; f++){
        cs::rectangle(gold() ML->gold_stripe.ML+[0,f*hgold/
            numberSubLayersGold] : wgold,hgold/numberSubLayersGold)
            gold_stripe;
}
}

function ridge_plasmonic_single(wgold,shift_gold,hbuffer,hridge,wridge,
    hadhesive,gridX,gridY,wcalc,hcalc,xcalc,ycalc,NumSampling,R,offset,
    tindex,WL,pol,HybridCriterion,setModeCount,plotFD,conf_norm,xconf,
    yconf,xconf2,yconf2) {
  // This function calculates the losses for the metal aided RE:KYW
     ridge waveguide at a single radius.
  // It selects the mode by looking at the confinement in the specified
     region of the waveguide.

//res::Clear();

int countmodes=0, tmval=0,countTM,number_modes;
double bend_loss;
complex Neff;

if (pol == 0) {  //TE
        sim::general(WL,te,useLast);
```

```
}
else {          //TM
        sim::general(WL,tm,useLast);
}
sim::option_mode2d({gridX,gridY},{wcalc,hcalc,xcalc,ycalc},NumSampling,
    PEC,PEC,PEC,PEC,NonUniform,NonUniform,TargetDistance);

printf("Gold width = ", wgold, "\n","gold shift = ",shift_gold,"\n");
printf("hridge = ", hridge, "\n");

var simFD=sim::mode2d(0,setModeCount,FD,Vectorial,false,tindex,R+offset,
    offset,{WL});

getModeInfo(simFD, setModeCount, HybridCriterion) modeinfoFD;
        if (pol == 0)    //TE
                {
                        var temp[] = modeinfoFD.getTE();
                        number_modes = len(temp);
                }
        else if (HybridCriterion!=0)            //TM
                {
                        var temp[] = modeinfoFD.getTM();
                        number_modes = len(temp);
                }
        else
                {
                        var temp[] = modeinfoFD.getHybrid();
                        number_modes = len(temp);
                }

        var tmFD[number_modes];

        if (pol == 0)    //TE
                {
                        tmFD = modeinfoFD.getTE();
                }
        else if (HybridCriterion!=0)              //TM
                {
                        tmFD = modeinfoFD.getTM();
                }
        else
                {
                        tmFD = modeinfoFD.getHybrid();
                }

countTM=len(tmFD);

for (;res::reference(simFD+"|Mode["+countmodes+"]   [,]#0"); countmodes
    ++);
        if (countmodes==0) {
```

```
        printf("No mode has been found!!\n");
        }
        else {
        for (int i=0;i<countmodes;i++){
        if(i==tmFD[tmval]){
                tmval=tmval+1;
                var myfield=res::reference("|Simulation - Mode#0|Mode["+
                    i+"]  [,]#0");

                Neff=res::reference(simFD+"|Neff["+i+"]#0");
                bend_loss=im2bendloss(WL,R,Neff.img);

                var power_actregion = sim::PowerInRegion(myfield,{xconf,
                    yconf},{xconf2,yconf2});
                var confinement_actregion=res::reference(power_actregion
                    +"|Confinement factor#0");

                if (confinement_actregion>conf_norm) {

                printf("bendloss: ",bend_loss,"\n");
                printf("\n",i,"\n");
                // PLOT MODE PROFILES
                if (plotFD==1) {
                if (pol==0) {
                        var plot_myplot=res::plotCreate("R = " + R + ":
                            Neff: "+Neff.re+"Loss/90deg bend:"+bend_loss
                            +" TE",Square) ;
                        var curve_myplot=res::reference(simFD+"|Mode["+i
                            +"]  [,]#0");
                        res::plotDataSquareIntensity(curve_myplot,"",0);
                } else {
                        var plot_myplot=res::plotCreate("R = " + R + ":
                            Neff: "+Neff.re+"Loss/90deg bend:"+bend_loss
                            +" TM",Square) ;
                        var curve_myplot=res::reference(simFD+"|Mode["+i
                            +"]  [,]#0");
                        res::plotDataSquareIntensity(curve_myplot,"",0);
                }

                res::plotGraph(Screen,128,128);
                }
                return bend_loss;
                }
        if(tmval>(countTM-1)) { break; }
        }}
}
}

function build_benchmark(Y,Gd,Lu,Yb,pol,wsubstrate,hsubstrate,wridge,
    hridge,hadhesive,numberSubLayersAdhesive) {
```

```
// This function builds a photonic RE:KYW ridge waveguide

// DEFINITION OF THE STRUCTURE
sim::clearCS();
cs::background(air() );
cs::rectangle(SiO2() MU->[0,0] : wsubstrate,hsubstrate) substrate;

cs::rectangle(doubleTungstate(Y,Gd,Lu,Yb,pol) RL->substrate.MU+[0,0] :
    wridge,hridge+hadhesive) ridge_gain;

cs::rectangle(BCB() ML->substrate.MU+[0,0] : wsubstrate,hadhesive/
    numberSubLayersAdhesive) adhesive_layers;
for(int f=1; f<numberSubLayersAdhesive; f++){
        cs::rectangle(BCB() ML->adhesive_layers.ML+[0,f*hadhesive/
            numberSubLayersAdhesive] : wsubstrate,hadhesive/
            numberSubLayersAdhesive) adhesive_layers;
}
}

function calculate_benchmark(gridX,gridY,wcalc,hcalc,xcalc,ycalc,xact,
    yact,xact2,yact2,xconf,yconf,xconf2,yconf2,NumSampling,R_min,R_max,
    offset,tindex,WL,pol,HybridCriterion,setModeCount,min_overlap,
    conf_norm,path,fname,logname,plotFD,mode_index,checkIndex,cI_axis,
    cI_pos) {
// This function initializes the calculation of the bending losses for
    the benchmark structure, making sure the logfiles can be
    distinguished from the metal aided simulation files.

fname=fname+"_benchmark";
logname=logname+"_benchmark";
calculate_bending(gridX,gridY,wcalc,hcalc,xcalc,ycalc,xact,yact,xact2,
    yact2,xconf,yconf,xconf2,yconf2,NumSampling,R_min,R_max,offset,tindex
    ,WL,pol,HybridCriterion,setModeCount,min_overlap,conf_norm,path,fname
    ,logname,plotFD,checkIndex,cI_axis,cI_pos);
}
```

file:library_ calculate_ bending.spt

```
// filename:library_calculate_bending.spt

#include "library_FieldDesigner_plot_and_polarization.spt";

function CheckIndex(axis,pos,gridX,gridY,xcalc,ycalc,wcalc,hcalc,WL) {
  // This function returns an array of indices, which can be used to
     make sure the correct materials are simulated
  // the results is a list of all different indices along an axis at a
     certain position
        int index=1;
        var rindex[gridX][gridY] = res::reference("|Simulation - Mode#0|
            Index, lambda="+WL+"[,]#0");
        complex temp2, array[max(gridX,gridY)];
```

```
        if (axis==1) { //y-direction (fixed x)
                int posx = round((pos-xcalc+wcalc/2)/wcalc*gridX);
                complex temp=rindex[posx][0];
                array[0]=temp;
                for (int y=0;y<gridY;y++) {
                        temp2=rindex[posx][y];
                        if (temp!=temp2) {
                                array[index]=temp2;
                                temp=temp2;
                                index++;
                        }
                }
        } else { //x-direction (fixed y)
                int posy = round((pos-ycalc+hcalc/2)/hcalc*gridY);
                complex temp=rindex[0][posy];
                array[0]=temp;
                for (int x=0;x<gridX;x++) {
                        temp2=rindex[x][posy];
                        if (temp!=temp2) {
                                array[index]=temp2;
                                temp=temp2;
                                index++;
                        }
                }
        }
        printf("\nindices:");
        for (int i=0;i<index;i++) {
                printf(" "+array[i]);
        }
        printf("\n");
}

function calculate_bending(gridX,gridY,wcalc,hcalc,xcalc,ycalc,xact,yact
    ,xact2,yact2,xconf,yconf,xconf2,yconf2,NumSampling,R_min,R_max,offset
    ,tindex,WL,pol,HybridCriterion,setModeCount,min_overlap,conf_norm,
    path,fname,logname,plotFD,checkIndex,cI_axis,cI_pos) {
// This function calculates the bendlosses for an already built
    structure for a single mode. The mode is followed from high to low
    radius by calculating the overlap between the previous mode and the
    new ones.
// Logfiles are created containing the calculated losses and indices.
    Also the power profiles are saved at every investigated radius.
// This function does NOT depent on the simulated structure

// LOGFILE
string date = sys::truncLeft(sys::DateTime(run),10);
string filename=path+logname+date+".txt";
string filenamelog=path+logname+date+"_log.txt";

int file = fopen(filename,"txt","a");
```

```
int filelog = fopen(filenamelog,"txt","a");

// SIMULATION
int countmodes=0,flag=1,tmval, fileB, index_all=0,number_modes, countTM;
double p_all,loss, bend_loss, confinement_actregion, bend_loss_corr,
    R_step=1, R=R_max, field_overlap, result, loss_corr;
complex Neff,Neff_corr;
string filenameB;

if (pol == 0) {  //TE
        sim::general(WL,te,useLast);
}
else {          //TM
        sim::general(WL,tm,useLast);
}
sim::option_mode2d({gridX,gridY},{wcalc,hcalc,xcalc,ycalc},NumSampling,
    PEC,PEC,PEC,PEC,NonUniform,NonUniform,TargetDistance);

for (;R>=R_min;) {

R_step = 1;
if (R<=5){R_step=0.5;}
if (R<=1.5){R_step=0.1;}
if (R<=1.2){R_step=0.05;}
if (R<=0.9){R_step=0.01;}

var simFD=sim::mode2d(0,setModeCount,FD,Vectorial,false,tindex,R+offset,
    offset,{WL});

getModeInfo(simFD, setModeCount, HybridCriterion) modeinfoFD;

if (pol == 0)    //TE
        {
        var temp[] = modeinfoFD.getTE();
        number_modes = len(temp);
        }
else if (HybridCriterion!=0)            //TM
        {
        var temp[] = modeinfoFD.getTM();
        number_modes = len(temp);
        }
else
        {
        var temp[] = modeinfoFD.getHybrid();
        number_modes = len(temp);
        }

var tmFD[number_modes];

if (pol == 0)    //TE
```

```
        {
        tmFD = modeinfoFD.getTE();
        }
else if (HybridCriterion!=0)              //TM
        {
        tmFD = modeinfoFD.getTM();
}
else
        {
        tmFD = modeinfoFD.getHybrid();
        }

countTM=len(tmFD);

for (;res::reference(simFD+"|Mode["+countmodes+"]  [,]#0"); countmodes
    ++);
        if (countmodes==0) {
        printf("No mode has been found!!\n");
        }
        else {
        tmval=0;
        for (int i=0;i<countmodes;i++){
                if(i==tmFD[tmval]){
                tmval=tmval+1;
                var myfield=res::reference("|Simulation - Mode#0|Mode["+
                    i+"]  [,]#0");

                var myfield_ref;
                if (R == R_max) { //this will be the first mode to
                    calculate and therefore the ref mode
                myfield_ref = myfield;
                res::SaveItem(path+"myfield_ref",myfield_ref);
                } else {
                myfield_ref = res::ReadItem(path+"myfield_ref");
                }

                var mo = sim::Overlap(myfield,myfield_ref,{0.0},{0.0});
                field_overlap = res::reference(mo+"|Overlap#0");
                if (field_overlap>min_overlap)
                        {
                        Neff=res::reference(simFD+"|Neff["+i+"]#0");
                        bend_loss=im2bendloss(WL,R,Neff.img);
                        Neff_corr=Neff*(R+offset)/R;
                        bend_loss_corr = im2bendloss(WL,R,Neff_corr.img)
                            ;

                        loss = im2loss(WL,Neff.img);
                        loss_corr = im2loss(WL,Neff_corr.img);
                        var power_actregion = sim::PowerInRegion(myfield
                            ,{xact,yact},{xact2,yact2});
```

```
confinement_actregion=res::reference(
    power_actregion+"|Confinement factor#0");
if (R==R_max)
{
var power_mode = sim::PowerInRegion(myfield,{
    xconf,yconf},{xconf2,yconf2});
var confinement_mode=res::reference(power_mode
    +"|Confinement factor#0");

if (confinement_mode<conf_norm) {
if(tmval>=(countTM-1)) { break; }
continue;}
} // try next mode if currentmode is not correct
     at R=R_max (overlap checking won't work)

printf("R: "+R+"\n");
printf("Neff_corr: "+Neff_corr+"\nLoss_corr:"+
    loss_corr+"\nBendLoss_corr:"+bend_loss_corr
    +"\n");
printf("Mode number: "+i+"\n");
printf("Confinement to the active region: "+
    confinement_actregion+"\n");
printf("ntarget = ", tindex, "\n\n");
printf ("offset = ", offset, ".\n");

fprintf(filelog,R + "\t"+ offset+"\t"+ Neff_corr
    +"\t"+ bend_loss_corr+ "\t" +
    confinement_actregion + "\n");
fprintf(file,R+"        "+loss_corr+"        "+
    bend_loss_corr+"        "+gridX+"        "+
    gridY+"        "+offset+"\n");

offset = offset+(R+offset)*(Neff.real/tindex
    -1.0);
res::SaveItem(path+"myfield_ref",myfield); //
    save temporary file, so memory can be cleared

// SAVE MODE PROFILE IN A FILE
filenameB = path+"R"+R+fname+".m";
fileB = fopen(filenameB,"txt","w");
res::Export(filenameB,myfield,0,true);
fclose(fileB);

// PLOT MODE PROFILE
if (plotFD ==1 && (R==R_max || R<R_min+R_step))
    {//plot (if asked for) the first and last
    mode
if (pol==0) {
        var plot_myplot=res::plotCreate("R = " +
            R + ": Neff: "+Neff.re+"Loss/90deg
```

```
                                        bend:"+bend_loss_corr+"  TE",Square)
                                        ;
                                var curve_myplot=res::reference(simFD+"|
                                    Mode["+i+"]  [,]#0");
                                res::plotDataSquareIntensity(
                                    curve_myplot,"",0);
                        } else {
                                var plot_myplot=res::plotCreate("R = " +
                                     R + ": Neff: "+Neff.re+"Loss/90deg
                                    bend:"+bend_loss_corr+"  TM",Square)
                                    ;
                                var curve_myplot=res::reference(simFD+"|
                                    Mode["+i+"]  [,]#0");
                                res::plotDataSquareIntensity(
                                    curve_myplot,"",0);
                        }

                        res::plotGraph(Screen,128,128);
                        }

                        // CHECK INDEX
                        if (checkIndex==1) {
                                CheckIndex(cI_axis,cI_pos,gridX,gridY,
                                    xcalc,ycalc,wcalc,hcalc,WL);
                        }

                flag = 0;
                break;
                }
                if(tmval>=(countTM-1)) { break; }
                }
        }
        if (flag==1){printf("No confined mode has been found!!\n");
        break;
        }
        //update variable for the next R calculation
        R=round(R-R_step,3);
        index_all++;
        res::Clear();
}
result=bend_loss_corr;
fclose(file);
fclose(fileB);
fclose(filelog);
res::Clear();
return result;
}
```

Appendix B: Lumerical MODE scripts

In this appendix, the scripts written for Lumerical MODE can be found.
file:ridge_ plasmonic.lsf

```
switchtolayout;
clear;
deleteall;
closeall;

########## VARIABLES ######################
micron=1e-6;

use_anisotropic = 0;
Y=0; Gd=0.447; Lu=0.078; Yb=0.475;

nSiO2 = 1.44402362170326;
nKYW = 2.0149521090137;
nBCB = 1.53475144786238;
nSi3N4 = 1.97929030561321;

WL = 1.55*micron;
mode_num=2;
pol=0;

x_ref_corner = 0;
y_ref_corner = 0;
z_ref_corner = 0 ;
height_substrate = 10*micron;
width_substrate = 35*micron;
depth_substrate = 10*micron;
height_buffer = 0.3*micron;
width_buffer = width_substrate;
depth_buffer = depth_substrate;
height_adhesive = 0.1*micron;
width_adhesive = width_substrate;
depth_adhesive = depth_substrate;
height_metal = 0.02*micron;
width_metal = 0.7*micron;
```

```
depth_metal = depth_substrate;
shift_metal = 0.24*micron;
height_ridge = 0.85*micron;
width_ridge = 0.85*micron;
depth_ridge = depth_substrate;

height_simulation = 2*micron;
width_simulation = 2*micron;
dr = 0*micron; # offset for radius
y_simulation = 0.0*micron; # measured from top of adhesive layer
gridX = 179;
gridY = 179;

tindex = 1.6; # target index
num_modes = 45;
R_max = 5*micron;
R_min = 1.2*micron;
Nsteps = 10; # number of radii

################ MATERIALS ################
materials;

############ BUILD STRUCTURE ##############
build_ridge_plasmonic;

################ SIMULATION ################
calculate_bending;
losses2mfile;

################### PLOT ###################
plot(radius*1e6,loss/100,"radius (microns)","loss (dB/cm)","loss vs
    radius at " + num2str(WL/micron) + " microns");
plot(radius*1e6,loss*pi/2*radius,"radius (microns)","loss/90deg bend","
    bendloss vs radius at " + num2str(WL/micron) + " microns");
plot(radius*1e6,neff,"radius (microns)","Re(Neff)","Re(Neff) vs radius
    at " + num2str(WL/micron) + " microns");
```

materials.lsf

```
if (getmaterial("SiO2 - PhoeniX")=="The material, SiO2 - PhoeniX, is not
     available.") { # check if SiO2 exists
         SiO2=addmaterial("Dielectric");
         setmaterial(SiO2,"name","SiO2 - PhoeniX");
}
setmaterial("SiO2 - PhoeniX","refractive index",nSiO2);

if (getmaterial("KYW")=="The material, KYW, is not available.") { #
    check if KYW exists
         KYW=addmaterial("Dielectric");
         setmaterial(KYW,"name","KYW");
}
```

```
if (use_anisotropic==1) { # formula retrieved from PhoeniX libraries
        lambda = WL/micron;
        amount = Y+Gd+Lu+Yb;
        Y = Y/amount;
        Gd = Gd/amount;
        Lu= Lu/amount;
        Yb = Yb/amount;

        NgY = sqrt(1 + 3.1278346*lambda^2./(lambda^2.-0.02608613));
        NmY = sqrt(1 + 2.9568303*lambda^2./(lambda^2.-0.02534002));
        NpY = sqrt(1 + 2.8134935*lambda^2./(lambda^2.-0.02338012));

        NgGd = 1.3867 + 0.6573*lambda^2./(lambda^2. - 0.028907) -
            0.0002913*lambda^2.;
        NmGd = 1.5437 + 0.4541*lambda^2./(lambda^2. - 0.035687) -
            0.0021567*lambda^2.;
        NpGd = 1.5344 + 0.4360*lambda^2./(lambda^2. - 0.034663) -
            0.0020999*lambda^2.;

        NgLu = sqrt(3.58334 + 0.73512*lambda^2./(lambda^2.-0.071289) -
            0.02953*lambda^.2);
        NmLu = sqrt(3.36989 + 0.74309*lambda^2./(lambda^2.-0.068607) -
            0.04331*lambda^.2);
        NpLu = sqrt(3.21749 + 0.75382*lambda^2./(lambda^2.-0.062830) -
            0.05076*lambda^.2);

        NgYb = sqrt(3.28412 + 0.9921*lambda^2./(lambda^2.-0.064648) -
            0.01936*lambda^.2);
        NmYb = sqrt(3.17884 + 0.91624*lambda^2./(lambda^2.-0.062936) -
            0.00485*lambda^.2);
        NpYb = sqrt(3.06172 + 0.88655*lambda^2./(lambda^2.-0.056920) -
            0.02286*lambda^.2);

        Ng = Y*NgY + Gd*NgGd + Lu*NgLu + Yb*NgYb;
        Nm = Y*NmY + Gd*NmGd + Lu*NmLu + Yb*NmYb;
        Np = Y*NpY + Gd*NpGd + Lu*NpLu + Yb*NpYb;
        setmaterial("KYW","anisotropy",1);
        setmaterial("KYW","refractive index",[Nm, Np, Ng]);
} else {
        setmaterial("KYW","refractive index",nKYW);
}

if (getmaterial("BCB")=="The material, BCB, is not available.") { #
    check if BCB exists
        BCB=addmaterial("Dielectric");
        setmaterial(BCB,"name","BCB");
}
setmaterial("BCB","refractive index",nBCB);
```

```
if (getmaterial("Si3N4")=="The material, Si3N4, is not available.") { #
    check if Si3N4 exists
        Si3N4=addmaterial("Dielectric");
        setmaterial(Si3N4,"name","Si3N4");
}
setmaterial("Si3N4","refractive index",nSi3N4);

# This does not work yet, gold is added using this sampled data:
# 1.55e-6       0.46491666153719        10.5674065100958
#if (getmaterial("Au - PhoeniX")=="The material, Au - PhoeniX, is not
    available.") { # check if Si3N4 exists
#       Au=addmaterial("Sampled data");
#       setmaterial(Au,"name","Au - PhoeniX");
#}
#setmaterial("Au - PhoeniX","sampled data",nAu);
```

build_ridge_plasmonic.lsf

```
addrect; # substrate
set("name","substrate");
set("material", "SiO2 - PhoeniX");
set("x",x_ref_corner );
set("y",y_ref_corner + height_substrate/2 );
set("z",z_ref_corner + depth_substrate/2 );
set("x span", width_substrate);
set("y span", height_substrate);
set("z span", depth_substrate);

addrect; # buffer
set("name","buffer");
set("material", "Si3N4");
set("x",x_ref_corner );
set("y",y_ref_corner + height_substrate + height_buffer/2 );
set("z",z_ref_corner + depth_buffer/2 );
set("x span", width_buffer);
set("y span", height_buffer);
set("z span", depth_buffer);

addrect; # adhesive
set("name","adhesive");
set("material", "BCB");
set("x",x_ref_corner);
set("y",y_ref_corner + height_substrate + height_buffer +
    height_adhesive/2 );
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", width_adhesive);
set("y span", height_adhesive);
set("z span", depth_adhesive);

addrect; # metal
set("name","metal");
```

```
set("material", "Au - PhoeniX");
set("x",x_ref_corner - width_metal/2 + shift_metal );
set("y",y_ref_corner + height_substrate + height_buffer + height_metal/2
    );
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", width_metal);
set("y span", height_metal);
set("z span", depth_metal);

addrect; # ridge
set("name","ridge");
set("material", "KYW");
set("x",x_ref_corner - width_ridge/2);
set("y",y_ref_corner + height_substrate + height_buffer +
   height_adhesive + height_ridge/2 );
set("z",z_ref_corner + depth_ridge/2 );
set("x span", width_ridge);
set("y span", height_ridge);
set("z span", depth_ridge);
```

calculate_ bending.lsf

```
addmode;
set("x",x_ref_corner + dr);
set("y",y_ref_corner + height_substrate + height_buffer +
   height_adhesive + height_ridge/2 + y_simulation);
set("z",z_ref_corner);
set("x span", width_simulation);
set("y span", height_simulation);
set("x min bc", "PML");
set("x max bc", "PML");
set("y min bc", "PML");
set("y max bc", "PML");
set("pml kappa",2);
set("pml sigma",10);
set("pml layers",12);
set("pml polynomial",3);
set("mesh cells x", gridX);
set("mesh cells y", gridY);
set("number of trial modes",num_modes);
set("wavelength",WL);
set("search","near n");
set("use max index",0);
set("n",tindex);
set("mesh refinement","Conformal variant 1");

addmesh;
set("name","metal_mesh");
set("x",x_ref_corner - width_metal/2 + shift_metal );
set("y",y_ref_corner + height_substrate + height_buffer + height_metal/2
    );
```

```
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", width_substrate);
set("y span", height_metal);
set("z span", depth_metal);
set("override x mesh",0);
set("override y mesh",1);
set("dy",height_metal/23);
set("override z mesh",0);

addmesh;
set("name","adhesive_mesh");
set("x",x_ref_corner);
set("y",y_ref_corner + height_substrate + height_buffer +
    height_adhesive/2 + height_metal/2 );
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", width_adhesive);
set("y span", height_adhesive-height_metal);
set("z span", depth_adhesive);
set("override x mesh",0);
set("override y mesh",1);
set("dy",(height_adhesive-height_metal)/22);
set("override z mesh",0);

addmesh;
set("name","ridge_mesh_right");
set("x",x_ref_corner);
set("y",y_ref_corner + height_substrate + height_buffer +
    height_adhesive + height_ridge/2 + y_simulation);
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", 0.2*micron);
set("y span", height_simulation);
set("z span", depth_adhesive);
set("set mesh multiplier",1);
set("override x mesh",1);
set("x mesh multiplier",4);
set("override y mesh",0);
set("override z mesh",0);

addmesh;
set("name","ridge_mesh_left");
set("x",x_ref_corner - width_ridge);
set("y",y_ref_corner + height_substrate + height_buffer +
    height_adhesive + height_ridge/2 + y_simulation);
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", 0.2*micron);
set("y span", height_simulation);
set("z span", depth_adhesive);
set("set mesh multiplier",1);
set("override x mesh",1);
set("x mesh multiplier",4);
```

```
set("override y mesh",0);
set("override z mesh",0);

# find modes in straight waveguide
select("MODE");
set("bent waveguide",0);
set("automatically remove pml modes",1);
set("threshold for pml mode removal",0.1);

N=findmodes;
for (i=1;i<=N;i=i+1) {
  TEfrac=getdata("MODE::data::mode"+num2str(i),"TE polarization fraction
      ");
  if (pol==0) {
    if (TEfrac>0.8) {
      copydcard("MODE::data::mode"+num2str(i),"REF");
      i=N+1;
    }
  } else {
    if (TEfrac<0.2) {
      copydcard("MODE::data::mode"+num2str(i),"REF");
      i=N+1;
    }
  }
}

# calculate bending losses
switchtolayout;
set("bent waveguide",1);
set("bend orientation",0);
set("automatically remove pml modes",0);

radius = linspace(R_max,R_min,Nsteps);
loss = matrix(length(radius));
neff = matrix(length(radius));

for(i=1:length(radius)) {

  bend_radius=radius(i)+dr;
  ?"Calculating modes (plasmonic), step " + num2str(i) + " of " +
      num2str(length(radius));

  set("bend radius",bend_radius);

  findmodes;

  curr_mode = bestoverlap("REF");
  loss(i) = getdata(curr_mode,"loss");
  neff(i) = getdata(curr_mode,"neff");
  modeprofile2mfile;
```

```
    switchtolayout;
    setnamed("MODE","n",neff(i));
}
```

losses2mfile.lsf

```
matlab("str=[datestr(now,7) datestr(now,5) datestr(now,10)];");
matlabget(str);
filename="losses/losses_plasmonic_mode"+num2str(mode_num)+"_"+str+".m";

write(filename,"radius=["+num2str(radius)+"];");
write(filename,"loss=["+num2str(loss/100)+"];");
write(filename,"bendloss=["+num2str(loss*pi/2*radius)+"];");
write(filename,"neff=["+num2str(neff)+"];");
```

ridge_plasmonic_benchmark.lsf

```
switchtolayout;
clear;
deleteall;
#closeall;

########## VARIABLES ######################
micron=1e-6;

use_anisotropic = 0;
Y=0; Gd=0.447; Lu=0.078; Yb=0.475;

nSiO2 = 1.44402362170326;
nKYW = 2.0149521090137;
nBCB = 1.53475144786238;
nSi3N4 = 1.97929030561321;

WL = 1.55*micron;
mode_num=2;
pol=0;

x_ref_corner = 0;
y_ref_corner = 0;
z_ref_corner = 0 ;
height_substrate = 10*micron;
width_substrate = 35*micron;
depth_substrate = 10*micron;
height_adhesive = 0.1*micron;
width_adhesive = width_substrate;
depth_adhesive = depth_substrate;
height_ridge = 0.85*micron;
width_ridge = 0.85*micron;
depth_ridge = depth_substrate;

height_simulation = 2*micron;
```

```
width_simulation = 2*micron;
dr = 0*micron; # offset for radius
y_simulation = 0.0*micron; # measured from top of adhesive layer
gridX = 179;
gridY = 179;

tindex = 1.6; # target index
num_modes = 45;
R_max = 5*micron;
R_min = 1.2*micron;
Nsteps = 10; # number of radii

########## MATERIALS ####################
materials;

########## BUILD STRUCTURE ###############
build_ridge_plasmonic_benchmark;

########## SIMULATION ####################
calculate_bending_benchmark;
losses2mfile_benchmark;

plot(radius*1e6,loss/100,"radius (microns)","loss (dB/cm)","loss vs
    radius at " + num2str(WL/micron) + " microns");
plot(radius*1e6,loss*pi/2*radius,"radius (microns)","loss/90deg bend","
    bendloss vs radius at " + num2str(WL/micron) + " microns");
plot(radius*1e6,neff,"radius (microns)","Re(Neff)","Re(Neff) vs radius
    at " + num2str(WL/micron) + " microns");
```

build_ ridge_ plasmonic_ benchmark.lsf

```
addrect; # substrate
set("name","substrate");
set("material", "SiO2 (Glass) - Palik");
set("x",x_ref_corner );
set("y",y_ref_corner + height_substrate/2 );
set("z",z_ref_corner + depth_substrate/2 );
set("x span", width_substrate);
set("y span", height_substrate);
set("z span", depth_substrate);

addrect; # adhesive
set("name","adhesive");
set("material", "BCB");
set("x",x_ref_corner);
set("y",y_ref_corner + height_substrate + height_adhesive/2 );
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", width_adhesive);
set("y span", height_adhesive);
set("z span", depth_adhesive);
```

```
addrect; # ridge
set("name","ridge");
set("material", "KYW");
set("x",x_ref_corner - width_ridge/2);
set("y",y_ref_corner + height_substrate + height_adhesive + height_ridge
    /2 );
set("z",z_ref_corner + depth_ridge/2 );
set("x span", width_ridge);
set("y span", height_ridge);
set("z span", depth_ridge);
```

calculate_ bending_ benchmark.lsf

```
addmode;
set("x",x_ref_corner + dr);
set("y",y_ref_corner + height_substrate + height_adhesive + height_ridge
    /2 + y_simulation);
set("z",z_ref_corner);
set("x span", width_simulation);
set("y span", height_simulation);
set("x min bc", "PML");
set("x max bc", "PML");
set("y min bc", "PML");
set("y max bc", "PML");
set("pml kappa",2);
set("pml sigma",10);
set("pml layers",12);
set("pml polynomial",3);
set("mesh cells x", gridX);
set("mesh cells y", gridY);
set("number of trial modes",num_modes);
set("wavelength",WL);
set("search","near n");
set("n",tindex);
set("mesh refinement","conformal variant 1");

addmesh;
set("name","adhesive_mesh");
set("x",x_ref_corner);
set("y",y_ref_corner + height_substrate + height_adhesive/2 );
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", width_adhesive);
set("y span", height_adhesive);
set("z span", depth_adhesive);
set("override x mesh",0);
set("override y mesh",1);
set("dy",height_adhesive/22);
set("override z mesh",0);

addmesh;
set("name","ridge_mesh_right");
```

```
set("x",x_ref_corner);
set("y",y_ref_corner + height_substrate + height_adhesive + height_ridge
    /2 + y_simulation);
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", 0.2*micron);
set("y span", height_simulation);
set("z span", depth_adhesive);
set("set mesh multiplier",1);
set("override x mesh",1);
set("x mesh multiplier",4);
set("override y mesh",0);
set("override z mesh",0);

addmesh;
set("name","ridge_mesh_left");
set("x",x_ref_corner - width_ridge);
set("y",y_ref_corner + height_substrate + height_adhesive + height_ridge
    /2 + y_simulation);
set("z",z_ref_corner + depth_adhesive/2 );
set("x span", 0.2*micron);
set("y span", height_simulation);
set("z span", depth_adhesive);
set("set mesh multiplier",1);
set("override x mesh",1);
set("x mesh multiplier",4);
set("override y mesh",0);
set("override z mesh",0);

# find modes in straight waveguide
select("MODE");
set("bent waveguide",0);
set("automatically remove pml modes",1);
set("threshold for pml mode removal",0.1);

N=findmodes;
for (i=1;i<=N;i=i+1) {
  TEfrac=getdata("MODE::data::mode"+num2str(i),"TE polarization fraction
      ");
  if (pol==0) {
    if (TEfrac>0.8) {
      copydcard("MODE::data::mode"+num2str(i),"REF");
      i=N+1;
    }
  } else {
    if (TEfrac<0.2) {
      copydcard("MODE::data::mode"+num2str(i),"REF");
      i=N+1;
    }
  }
}
```

```
# calculate bending losses
switchtolayout;
set("bent waveguide",1);
set("bend orientation",0);
set("automatically remove pml modes",0);

radius = linspace(R_max,R_min,Nsteps);
loss = matrix(length(radius));
neff = matrix(length(radius));

for(i=1:length(radius)) {

  bend_radius=radius(i)+dr;
  ?"Calculating modes (benchmark), step " + num2str(i) + " of " +
      num2str(length(radius));

  set("bend radius",bend_radius);

  findmodes;

  curr_mode = bestoverlap("REF");
  loss(i) = getdata(curr_mode,"loss");
  neff(i) = getdata(curr_mode,"neff");
  modeprofile2mfile_benchmark;

  switchtolayout;
  setnamed("MODE","n",neff(i));
}
```

losses2mfile_ benchmark.lsf

```
matlab("str=[datestr(now,7) datestr(now,5) datestr(now,10)];");
matlabget(str);
filename="losses/losses_benchmark_mode"+num2str(mode_num)+"_"+str+".m";

write(filename,"radius_benchmark=["+num2str(radius)+"];");
write(filename,"loss_benchmark=["+num2str(loss/100)+"];");
write(filename,"bendloss_benchmark=["+num2str(loss*pi/2*radius)+"];");
write(filename,"neff_benchmark=["+num2str(neff)+"];");
```