Plasmonic Stripe Waveguide Coupler with Integrated Wavelength Division Multiplexer

by

Ongun Arısev

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

KOÇ
UNIVERSITY

May 16, 2017

Plasmonic Stripe Waveguide Coupler with Integrated Wavelength Division Multiplexer

Koç University
Graduate School of Sciences and Engineering
This is to certify that I have examined this copy of a master's thesis by

Ongun Arısev

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Asst. Prof. Şükrü Ekin Kocabaş

Prof. İrşadi Aksun

Asst. Prof. Alexandr Jonas

Date:

To my mother and father...

TABLE OF CONTENTS

ABSTRACT

Plasmonics is the research field concerned with the interaction between free electrons on a metal surface and electromagnetic waves. It is possible to design nanostructures which exploit these interactions to manipulate light at the nanoscale -much beyond the diffraction limit- with the advent of new techniques and devices. Integration with electronic circuits, plasmonic lasers, biosensors, chemical sensors, surface-enhanced Raman spectroscopy and plasmonic lenses are among the ever increasing applications of the field [1].

Coupling from free-space electromagnetic waves to surface plasmon polaritons (SPP) and decoupling from SPPs back to free-space electromagnetic waves is one of the main considerations while designing a nanophotonic chip. Directing and routing the generated SPP beams are also another concern, and usually coupling and direction of SPPs are realized with the same structure. These structures are usually an array of scatterers, but a single scatterer might be sufficient. There is on-going research on various scatterer geometries, designs and fabrication techniques to optimize the efficiency of SPP generation. Furthermore, there are various wavelength demultiplexer designs featuring periodic or aperiodic arrays of scatterers.

In this thesis the aim is to direct optical signals of different wavelengths to three 1 µm wide stripe waveguides whose centers are separated by 4 µm at the end of a gold film. To achieve this aim we first experimented with an SPP beam launcher which excites an SPP beam with a predefined amplitude and phase. However, we were unable to couple into the stripe waveguide mode, because this approach is limited in the sense that it cannot accommodate for a variation in the amplitude and phase that are smaller than the dimensions of the scatterer. Thus we decided to use a different method.

Moving on to an established wavelength demultiplexer design utilizing nanoslits [2] was the new starting point. After mastering the iterative algorithm stripe waveguides were put at the focal positions to study the coupling of SPPs of different wavelengths into corresponding stripe waveguide modes. Simulations were done for two types of scatterers: Δ-antennas and nanoslits. It was observed that owing to the unidirectionality Δ-antennas outperformed their nanoslit counterparts in coupling efficiency by nearly two folds. This design may be generalized for different wavelengths and a larger number of focal points.

ÖZETÇE

Plazmonik metal yüzeylerdeki serbest elektronların ve elektromanyetik dalgaların etkileşimini inceleyen bir araştırma alanıdır. Yeni tekniklerin ve cihazların icadıyla bu etkileşimlerden faydalanarak ışığı saçılım limitinin altındaki boyutlarda kontrol edebilen nano yapılar geliştirmek mümkündür. Elektronik devrelerle entegrasyon, plazmonik lazerler, biyolojik ve kimyasal sensörler, Raman spektroskopisi(SERS) ve plazmonik lensler sayısı sürekli artan uygulama alanlarından bazılarıdır [1].

Boş uzayda yol alan elektromanyetik dalgaları yüzey plazmon polaritonlarına (YPP) eşleştirmek ve YPP'leri boş uzayda yol alan elektromanyetik dalgalara çevirmek nanofotonik bir çip tasarımında temel hususlardandır. YPP ışın demetlerini istenilen şekilde yönlendirmek ise diğer bir husustur. Genellikle eşleştirme ve yönlendirme nanoboyutlarda aynı nano yapılarla sağlanır. Saçıcı geometrileri, YPP oluşumu optimizasyonu için dizayn ve üretim teknikleri üzerine araştırmalar sürmektedir. Ek olarak saçıcı dizileri kullanan dalga boyu çözücüler geliştirilmiştir.

Bu tezdeki amaç farklı dalga boylarına sahip optik sinyalleri altın film üzerinde merkezleri arasındaki uzaklık 4 µm olacak şekilde ayarlanmış 1 µm genişliğe sahip şerit dalga kılavuzlarının kipleri ile eşleştirmektir. Bu amaca ulaşmak için öncelikle belirli bir genlik ve faza sahip YPP demeti uyaran YPP ışınlayıcılar ile deneme yaptık. Ancak bu şekilde şerit dalga kılavuzu kipi ile eşleştiremedik çünkü bu yaklaşımda eşletirilmek istenilen modun genlik ve fazın saçıcının boyutun kullanılan saçıcılar mertebesinde olması gerekiyor. Bunun üzerine farklı bir yöntem kullanmaya karar verdik.

Nano yarıklar kullanan daha önceden çalışılmış bir dalga boyu ayrıştırıcı dizaynının yinelemeli algoritmasını değiştirip uygulamaya karar verdik. Farklı dalga boyuna sahip YPP'leri dalga kılavuzu kiplerine eşleştirmek için odak noktalarına plazmonik şerit dalga kılavuzları konmuştur. Bu işlem Δ-antenler ve nano yarık dizileriyle tekrarlanması sonucunda Δ-antenlerin yaklaşık iki kat daha fazla performans sağladığını gözlemledik.

ACKNOWLEDGMENTS

First of all, I desire to express my earnest gratitude to my family for their never ending support and my supervisor Asst. Prof. Şükrü Ekin Kocabaş for the guidance he provided despite the encumbering circumstances. This thesis is rendered possible thanks to his patience, expertise, encouragement, advice and the ability to provide insight in difficult situations by changing the point of view. He was not only an advisor, but also a humble mentor and a genuine friend. I feel privileged to have worked with such a researcher.

Solmaz Naghizadeh, Adeel Afridi and Aziz Karaşahin deserve special thanks as members of the Nanophotonics Research Group. Solmaz was of great help in particular with the procedures for using the Sun Grid Engine, performing optical measurements, helping with the issues arose during FDTD simulations and I adopted the habit of keeping a logbook from her. In the short period we were together Aziz helped with electron-beam lithography, thermal evaporation, scanning electron microscopy, coursework and taught me how to perform piranha cleaning without hurting myself. While at the topic of fabrication, I owe thanks to Cansu Yıldırım and Dr. Barış Yağcı at Koç University Surface Science and Technology Center (KUYTAM) for their patience in dealing with a curious and ardent person who is a fan of asking too many questions. I would also like to thank Dr. Cenk Yanık from Sabancı University Nanotechnology Research Center (SUNUM) for his assistance in the e-beam lithography, using the facilities at SUNUM and providing e-beam lithography training. Lastly, I acknowledge the cleanroom training I received from Ümmü Koç from Koç University.

I express my sincere gratitude to Hüseyin Özbey from Koç University IT. He helped a great deal with the simulations run on the high performance computing cluster and technical issues were alleviated promptly since he was always available through e-mail and phone. I appreciate his efforts in facilitating my progress.

Our office had the perfect environment for studying; in particular Altynbek Is-

abekov deserves special thanks for providing such a tranquil environment in which one can thrive. Being a computer guru his indispensable skills helped with anything computer related such as LaTeX, Matlab, Linux etc. Berkay Yılmaz was also eager to help with coding issues and I deem him as the master of Matlab. I appreciate the effort Hazan Ezgi İmer put to help me remotely access computer resources and she is also a great friend with wisdom. We started our master's journey together with Adeel Afridi, and he was a great friend, gym-buddy, lab-mate, partner in crime. He managed to motivate me to push through challenges and had a great sense of humor; I wish him success in his future endeavors. Speaking of gym it would be a mistake to forget Sasan Asadi, who is a great friend and a fan of sarcastic humor by the way, as he practically lives there.

I also feel obliged to thank Logo Sailing Team, in particular Onur Bilgen, for providing with the opportunity to be part of a competitive yacht crew. Sailing is a great way to deal with stress, find peace of mind and gather your thoughts.

I would like to acknowledge financial support from TÜBİTAK project with the code 112E247 in the context of 1001 - Scientific and Technological Research Projects Funding Program. Lastly I would like to thank my mother once again for motivating and trusting in me even at times I doubt myself.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

ϵ	Permittivity (F m^{-1})
ϵ_0	Free space permittivity 8.85 × 10^{-12} (F/m)
ϵ_r	Relative permittivity(unitless)
B	Magnetic induction field vector (T)
D	Electric displacement field vector (C/m^2)
E	Electric field vector (V m^{-1})
H	Magnetic field vector (A m^{-1})
J	Current density (A/m^2)
μ	Magnetic permeability (N m^{-2})
μ	Relative permeability (unitless)
μ_0	Free space permeability 4⍰ × 10^{-7} (N/m^2)
ρ_f	Free charge (C)
AOTF	Acousto Optic Tunable Filter
SPP	Surface Plasmon Polariton
TE	Transverse Electric
TM	Transverse Magnetic

Chapter 1

INTRODUCTION

1.1 *Literature Review*

Surface plasmon polaritons (SPPs) were described mathematically as early as the beginning of the 20th century by Sommerfeld and Zenneck, but these works were concerned with radio waves instead of visible light [3, 4]. An unexpected intensity drop was observed when visible light was reflected off a grating by Wood, but a physical meaning could not be associated to it then [5]. This missing information was provided by Fano in 1941 and he explained the lost energy was coupled into surface waves [6]. In 1957 Ritchie experimented with fast electrons and thin films establishing the theoretical description of surface plasmons for the first time [7]. Thereafter, Ritchie attributed the unexpected reflections losses on gratings in the optical domain to surface plasmon resonances [8]. The collective oscillation of electrons on a metal surface was also handled by Economou using the complete set of Maxwell's equations [9]. The term surface plasmon polariton was coined much later in 1974 by Cunningham and co-workers [10]. The main problem with photonics is the inability to confine the light into dimensions less than that of the wavelength of the light. Plasmonics essentially serve as the bridge between semiconductor electronics and dielectric photonics, since photonic devices are bigger and faster compared to semiconductor devices [1, 11]. The major applications of SPPs are biosensors, near-field optics, surface-enhanced Raman spectroscopy [12]. *Plasmonics* is the field concerned with SPPs and it is a very active field as evidenced by exponentially increasing number of publications [13].

Designing SPP launchers is vital in plasmonic applications and several different SPP launchers were designed [14]. Here some SPP launchers will be briefly recalled. The nondiffracting Airy SPP beams are of particular interest as they do not diffract and feature self-healing capabilities. The launcher in this case consists of a metallic phase grating [15]. Another nondiffracting option is the Cosine-Gauss plasmon beam; this beam launcher consisting of intersecting metallic gratings was developed and it can propagate on a straight line for up to 80 µm [16]. Additionally SPP beam collimator [17], Mathieu and Weber SPP beam launchers were also developed. Mathieu beam preserves its shape while propagating along an elliptic trajectory, whereas Weber beam preserves its shape while propagating along a parabolic trajectory; additionally they possess self-healing capabilities just like the Airy beam [18]. One of the latest development in the field is the realization of the possibility to set the amplitude and phase of the SPPs as desired via tuning the orientation of two slits illuminated under circularly polarized laser incident, this is expected to pave the way for the development of plasmonic imaging and lithography devices [19].

An integrated compact plasmonic wavelength demultiplexer proves useful in spectral imaging and sensing applications by redirecting SPPs of different wavelengths to different focii positions and various methods are used to achieve this purpose [2]. Drezet et al. realized demultiplexing of SPP beams using a gold photonic crystal manufactured with standard electron-beam lithography technique [20]. Laux et al. developed a technique enabling the recording of the spectral image cube[1] in a single exposure in contrast with the conventional spectral imaging techniques via utilizing SPPs. The advantage over the conventional method is that long exposure times are not needed [21]. In 2010 Chenglong Zhao and Jiasen Zhang proposed a design using concentric grooves perforated on a gold film; this design resulted in a higher resolution compared to the preceding designs [22]. However, most of these designs inherit the disadvantage of being limited to a particular wavelength and not entirely applicable to integration into photonic circuits due to being periodic and including the coupling element. In 2011 a nanoarray design is proposed that achieves broad band focusing and demultiplexing at the same time [23]. Later in 2014 Pierre Wahl et al. devised a method constructing modal demultiplexer with nanoslits located on the surface of a metal film using their modal source radiator model for speed in the iterative algorithm [24, 25]. There are also holographic approaches such as the one by Daniel Wintz et al. which uses a holographic metalens to focus incident SPP beams on predefined focal

[1] Image of an object is 2D and the wavelength is the third dimension

positions and the polarization of the incident beam determines whether switching will be realized or not [26]. However coupling to stripe waveguides at the end of the device region is not examined in these studies.

Wavelength division multiplexing (WDM) is commonly used in optical communications to achieve increased rate of data transfer. In this approach the optical signals of different wavelengths (or colors) carry different bits of the information. Our objective in this thesis is to propose a device which realizes coupling from free-space to plasmonic stripe waveguides together with WDM functionality. The same structure demultiplexes the incident optical signal, and couples its constituent channels at different wavelengths to plasmonic stripe waveguides.

1.2 Thesis Outline

Chapter 2 is a summary of basic concepts from the electromagnetic theory such as Maxwell's equations, constitutive relations and boundary conditions. Then permittivity models of metals are explored in detail. This chapter finishes with the analysis of SPP propagation on metal-dielectric interfaces.

In Chapter 3 plasmonic scatterers are introduced and their scattering patterns are investigated in detail with finite difference time domain simulations. The scatters of concern in this thesis are 1D gratings, rectangular nanoslits and triangular antennas. The dependence of the scattering pattern on the in-plane angle for a single scatterer is characterized and a fit is developed.

In Chapter 4 aperiodic arrays of triangular scatterers are investigated. The aim of these arrays is to generate the electric field amplitude and phase of a given SPP beam. The case under investigation here is a second-order Hermite-Gauss beam, but this method is applicable for generating an arbitrary beam provided that the features of the beam are of greater size then the dimensions of the individual antennas in the array. We found out that this method is not applicable in our case.

In Chapter 5 antenna theory and *array factor* in particular is discussed before moving on to the algorithm which we use in locating the scatter positions in the aperiodic array. Then the algorithm is explained in a step by step manner.

In Chapter 6 the results of the simulations done for mode coupling are discussed.

Here we have one simulation set for nanoslit arrays and three simulation sets for the Δ-antenna arrays. Each simulation set consists of two simulations: one with the waveguide and one without.

Chapter 7 is the summary and discusses what can be done in the near future to improve the performance of the device.

Chapter 2

BACKGROUND INFORMATION

2.1 Brief review of electromagnetic theory

The following set of first-order partial differential equations are known as classical Maxwell's equations [27]:

$$\nabla \cdot \mathbf{D} = \rho_f \tag{2.1a}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2.1b}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \tag{2.1c}$$

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0 \tag{2.1d}$$

In order to govern the dynamics of interacting electromagnetic fields and charged particles these equations are combined with Lorentz force equation and Newton's second law of motion. Respectively, these equations are as follows [28, 29]:

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{2.2}$$

$$\mathbf{F} = \frac{d\mathbf{P}}{dt} \tag{2.3}$$

We also have the constitutive equations relating **D** with **E** and **H** with **B**. These are as follows:

$$\mathbf{D} = \epsilon \mathbf{E} = \epsilon_0 \epsilon_r \mathbf{E} \tag{2.4}$$

$$\mathbf{B} = \mu \mathbf{H} = \mu_0 \mu_r \mathbf{H} \tag{2.5}$$

Here we made the implicit assumption that the media are linear, yet the media may still be anisotropic and this may be realized by treating ϵ and μ as tensors. These quantities are called electric permittivity and magnetic permeability, respectively. However, in this thesis we will be dealing with linear, isotropic and non-magnetic media. [1]

[1] Non-magnetic means that the permeability of the material is very close to μ_0

It should also be noted that the constitutive relations hold in the frequency and wave-vector domain as the connection in Equation 2.4 and Equation 2.5 can be nonlocal in time and space.

$$\mathbf{D}(\mathbf{x}, t) = \int d^3x^{'} \int dt^{'} \epsilon(\mathbf{x}^{'}, t^{'})\mathbf{E}(\mathbf{x} - \mathbf{x}^{'}, t - t^{'}) \tag{2.6}$$

$$f(\mathbf{k}, \omega) = \int d^3x \int dt f(\mathbf{x}, t) e^{-i\mathbf{k}\cdot x + i\omega t} \tag{2.7}$$

With the introduction of the above Fourier transform Equation 2.4 and Equation 2.5 can be rewritten as follows:

$$\mathbf{D}(\mathbf{k}, \omega) = \epsilon(\mathbf{k}, \omega) E(\mathbf{k}, \omega) \tag{2.8}$$

$$\mathbf{B}(\mathbf{k}, \omega) = \mu(\mathbf{k}, \omega) H(\mathbf{k}, \omega) \tag{2.9}$$

Using Stoke's and Gauss' laws Maxwell's equations can be written in integral form and this is beneficial for analyzing the behavior of electromagnetic fields near boundaries [29].

$$\oiint_S \mathbf{D} \cdot d\mathbf{a} = Q_f \tag{2.10a}$$

$$\oiint_S \mathbf{B} \cdot d\mathbf{a} = 0 \tag{2.10b}$$

$$\oint_C \mathbf{H} \cdot d\mathbf{l} = I_f + \frac{d}{dt} \int_S \mathbf{D} \cdot d\mathbf{a} \tag{2.10c}$$

$$\oint_C \mathbf{E} \cdot d\mathbf{l} = -\frac{d}{dt} \int_S \mathbf{B} \cdot d\mathbf{a} \tag{2.10d}$$

In the first two equations above S is any closed surface, whereas in the remaining two S is a surface bounded by the closed contour C.

Using Equation 2.10 with an appropriate selection of tiny loops and boxes penetrating into two different media at an interface we arrive at the continuity equations for electromagnetic fields.

$$\epsilon_1 E_1^{\perp} - \epsilon_2 E_2^{\perp} = \sigma_f \tag{2.11a}$$

$$B_1^{\perp} - B_2^{\perp} = 0 \tag{2.11b}$$

$$\frac{1}{\mu_1}\mathbf{B}_1^{\parallel} - \frac{1}{\mu_2}\mathbf{B}_2^{\parallel} = \mathbf{K}_f \times \hat{n} \tag{2.11c}$$

$$\mathbf{E}_1^{\parallel} - \mathbf{E}_2^{\parallel} = 0 \tag{2.11d}$$

These equations prove useful when solving for waveguide modes and deriving the electromagnetic fields at interfaces.

The refractive index is defined as the ratio of the speed of the propagation of an electromagnetic wave in free space to the speed of the propagation of an electromagnetic wave in a particular medium.

$$n = c/v = \sqrt{\frac{\mu\epsilon}{\mu_0\epsilon_0}} = \sqrt{\epsilon_r} \tag{2.12}$$

The justification for the second simplification follows from the fact that we are dealing with non-magnetic materials. It is evident from the above equation that when ϵ is complex n is also complex, and vice versa. Employing the notation of putting a tilde over complex quantities the refractive index, attenuation constant and the electric permittivity are [30]:

$$\tilde{n} = n + i\kappa \tag{2.13a}$$

$$\tilde{\epsilon}_r = \epsilon_1 + i\epsilon_2 \tag{2.13b}$$

$$\tilde{n}^2 = \tilde{\epsilon}_r \tag{2.13c}$$

$$\epsilon_1 = n^2 - \kappa^2 \tag{2.13d}$$

$$\epsilon_2 = 2n\kappa \tag{2.13e}$$

$$n^2 = \frac{1}{2}(\epsilon_1 + \sqrt{\epsilon_1^2 + \epsilon_2^2}) \tag{2.13f}$$

$$\kappa^2 = \frac{1}{2}(-\epsilon_1 + \sqrt{\epsilon_1^2 + \epsilon_2^2}) \tag{2.13g}$$

2.2 *Permittivities of metals*

Since surface plasmon polaritons can propagate along metal-dielectric interfaces the permittivity of metals play a huge role in the distribution of the fields. The frequently used mathematical models for the permittivities of materials are Lorentz model, Drude model and Extended Drude model. In this section we will first focus on the Lorentz model and then the Drude model.

2.2.1 Lorentz Model

In Lorentz model the electron is modeled as a damped harmonic oscillator. The damping term accounts for the fact that the electrons can loose energy by collisional processes. Here the motion of the nucleus is disregarded since it is massive compared to electrons. So we can start by writing the equation of motion for the electron [30].

$$m_0 \frac{d^2 x}{dt^2} + m_0 \gamma \frac{dx}{dt} + m_0 \omega_0^2 x = -e\mathrm{E} \tag{2.14}$$

Here m_0 is the mass of the electron, γ is the damping rate, e is the magnitude of the charge of the electron and E is the electric field of the incident electromagnetic wave. Mathematically this is a second-order, inhomogeneous ordinary differential equation. We also assume that the incident light is time harmonic, that is the electric field on the right hand side can be expressed as:

$$\mathrm{E} = \mathrm{E}_o \cos(\omega t + \Phi) = \mathrm{E}_o \Re(\exp(-i\omega t - \Phi)) \tag{2.15}$$

Here E_o is the amplitude and Φ is the phase of the incident light which is the driving term in Equation 2.14. Here the assumption that the driving electric field will impose an oscillation of is own frequency is made, so the solutions sought are of the following form:

$$x(t) = \Re(X_0 \exp(-i\omega t - \phi)) \tag{2.16}$$

Here X_0 is the complex amplitude of the oscillation of electrons. By plugging Equation 2.16 into Equation 2.14 we get the following equation[2]:

$$-m_0 \omega^2 X_0 e^{-i\omega t} - i m_0 \gamma \omega X_0 e^{-i\omega t} + m_0 \omega_0^2 X_0 e^{-i\omega t} = -e\mathrm{E}_o e^{-i\omega t} \tag{2.17}$$

Solving for X_0 yields:

$$X_0 = \frac{-e\mathrm{E}_o / m_0}{\omega_0^2 - \omega^2 - i\gamma\omega} \tag{2.18}$$

[2] The phase information is embedded in the amplitudes making them complex.

This motion of electrons induces a microscopic, time dependent dipole moment p(t) which in turn yields a macroscopic polarization (dipole moment per unit volume) that can be incorporated in constitutive relations. The macroscopic polarization is related to the dipole moment via:

$$\begin{aligned} P_{induced} &= Np \\ &= -Nex \\ &= \frac{Ne^2}{m_0}\frac{1}{(\omega_0^2 - \omega^2 - i\gamma\omega)}\mathrm{E} \end{aligned} \tag{2.19}$$

According to the above equation the magnitude of P is small unless the frequency is close to the resonance frequency ω_0. In order to relate the polarization to the relative permittivity we make use of the relation between the electric displacement field **D**[3], the electric field **E** and the polarization vector **P**. With the assumption that we are dealing with an isotropic material this relation is as follows:

$$\begin{aligned} \mathbf{D} = \epsilon_0\mathbf{E} + \mathbf{P} &= \epsilon_0\mathbf{E} + \mathbf{P_{background}} + \mathbf{P_{induced}} \\ &= \epsilon_0\epsilon_r\mathbf{E} = \epsilon_0\mathbf{E} + \epsilon_0\chi\mathbf{E} + \mathbf{P_{induced}} \end{aligned} \tag{2.20}$$

Comparing Equation 2.19 with Equation 2.20 yields the relative permittivity:

$$\epsilon_r(\omega) = 1 + \chi + \frac{Ne^2}{\epsilon_0 m_0}\frac{1}{(\omega_0^2 - \omega^2 - i\gamma\omega)} \tag{2.21}$$

2.2.2 Drude Model

The Drude model treats the metal as having immovable positive ions with a non-interacting free electron gas. Essentially this model is a special case of the Lorentz model with no restoring force hence the lattice potential and interaction of electrons among themselves are not taken into account [31]. The electrons are assumed to oscillate in response to the driving field. Collisions damped the motion of electrons and the collision frequency is denoted with $\gamma = 1/\tau$ from here on. τ is the relaxation time of the free electron gas, and it is on the order of 10^{-14} s at room temperature. So the equation of motion in the Lorentz model Equation 2.14 is simplified to:

$$m_0\frac{d^2x}{dt^2} + m_0\gamma\frac{dx}{dt}x = -e\mathrm{E} \tag{2.22}$$

[3] Bold face letters are used to denote vectors with three components in euclidean space.

Solving for x(t), the displacement, results in:

$$x(t) = \frac{e}{m_0(\omega^2 + i\gamma\omega)}\mathrm{E(t)} \tag{2.23}$$

Using the same reasoning of deriving the electric permittivity from electric displacement field as before we have the following electric displacement vector [4]:

$$\mathbf{D} = \epsilon_0\mathbf{E} + \mathbf{P} = \epsilon_0\mathbf{E} + \mathbf{P}_{\text{free electrons}} \tag{2.24}$$

The permittivity follows from the above equation,

$$\begin{aligned} \epsilon(\omega) &= 1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} = 1 - \frac{\omega_p^2\tau^2}{\omega^2\tau^2 + i\omega\tau} \\ \epsilon_{\text{real}}(\omega) &= 1 - \frac{\omega_p^2\tau^2}{1 + \omega^2\tau^2} \\ \epsilon_{\text{imag}}(\omega) &= \frac{\omega_p^2\tau}{\omega(1 + \omega^2\tau^2)} \end{aligned} \tag{2.25}$$

Here for the notational convention $\omega_p = Ne^2/(\epsilon_0 m_0)$ is used. The behavior of Equation 2.25 can be examined under different frequency ranges. We start by restricting the discussion to frequencies below ω_p, yet close enough to ω_p such that the product $\omega \gg \gamma$. In this frequency range the complex part of Equation 2.25 can be neglected.

$$\epsilon(\omega) = 1 - \frac{\omega_p^2}{\omega^2} \tag{2.26}$$

When the frequency is very low compared to the electron collision rate γ, that is $\omega \ll \gamma$. This is the frequency range in which metals are absorbing since $\epsilon_{imag} \gg \epsilon_{real}$ in Equation 2.25. In this frequency range the real and complex refractive index calculated using Equation 2.13 are similar in magnitude.

$$n \approx \kappa = \sqrt{\frac{\epsilon_{imag}}{2}} = \sqrt{\frac{\tau\omega_p^2}{2\omega}} \tag{2.27}$$

The power absorption coefficient is given as

$$\alpha = \kappa k_0 = \kappa\frac{2\pi}{\lambda} = \sqrt{\frac{2\omega_p^2\tau\omega}{c^2}} \tag{2.28}$$

By using DC conductivity $\sigma_0 = Ne^2\tau/m_0 = \omega_p^2\tau\epsilon_0$ this equation can be written as follows:

$$\alpha = \sqrt{2\sigma_0\omega\mu_0} \tag{2.29}$$

[4] This time the inherent polarization $\mathbf{P}_{\text{background}}$ is neglected.

It is also useful to define a *skin depth* after which the fields decay by 1/e.

$$\sigma = \frac{2}{\alpha} = \frac{c}{\kappa\omega} = \sqrt{\frac{2}{\sigma_0 \omega \mu_0}} \tag{2.30}$$

2.2.3 *Extended Drude Model*

Unfortunately the simple Drude model cannot cover all aspects of the behavior of metals under illumination especially when the frequency of the incident light is greater than plasma frequency. For noble metals that are ubiquitously used in the field of plasmonics such as Au, Ag, Cu and additional polarization term is needed due to the polarization induced by the d band electrons. The modification is to introduce the term

$$\mathbf{P}_\infty = \epsilon_0(\epsilon_\infty - 1)\mathbf{E} \tag{2.31}$$

and plug this into Equation 2.24 which yields the following after some manipulation.

$$\epsilon(\omega) = \epsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} \tag{2.32}$$

This model still far from perfect and breaks down in the regime of visible light where interband transitions take place.

2.3 *Plasmons*

In this section volume plasmons, surface plasmon polaritons and localized plasmons will be discussed. The differences between these concepts will be highlighted.

2.3.1 *Volume Plasmons*

By combining the two Maxwell curl equations in Equation 2.1, we arrive at the wave equation[5] [31].

$$\nabla \times \nabla \times \mathbf{E} = -\mu_0 \frac{\partial^2 \mathbf{D}}{\partial t^2} \tag{2.33}$$

$$\mathbf{k}(\mathbf{k} \cdot \mathbf{E}) - k^2\mathbf{E} = -\epsilon(\mathbf{k}, \omega)\frac{\omega^2}{c^2}\mathbf{E} \tag{2.34}$$

[5] The fields are assumed to have a time and position dependence of the form $e^{-i\omega t + i\mathbf{k}\cdot\mathbf{r}}$ The substitution of $\mathbf{k}$ for spatial derivatives, $-i\omega$ for temporal derivatives and the use of the vector identity $\nabla \times (\nabla \times \mathbf{A}) = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2\mathbf{A}$ yields the equation in the second row.

For transverse waves ($\mathbf{k} \cdot \mathbf{E} = 0$), Equation 2.34 reduces to the following dispersion relation,

$$k^2 = \epsilon(\mathbf{k}, \omega)\frac{\omega^2}{c^2} \tag{2.35}$$

Whereas for longitudinal waves this implies $\epsilon(\mathbf{k}, \omega) = 0$. Using Equation 2.35 with Equation 2.26 we obtain the explicit dispersion relation for traveling waves,

$$\omega^2 = \omega_p^2 + k^2 c^2 \tag{2.36}$$

From this equation it is obvious that electromagnetic waves with $\omega < \omega_p$ cannot propagate inside the metal and get attenuated. If $\omega > \omega_p$ the free electrons support transverse electromagnetic wave propagation.

In addition to this it should be noted that $\epsilon(\omega)$ from Equation 2.26 is 0 at $\omega = \omega_p$ with $k = 0$ also. It also follows that $\mathbf{D} = 0$ and $\mathbf{E} = \frac{-\mathbf{P}}{⊠_0}$, which is a pure depolarization field.

Assuming free electrons to be moving longitudinally with respect to the fixed positive ions in the metal slab, a collective displacement of these free electrons by x results in a surface charge density of $\sigma = \pm Nex$. From Gauss' law between the positive ions and the negative electrons a homogeneous electric field $\mathbf{E} = \frac{Nex}{\epsilon_0}$ is created.

This electric field acts as a restoring force for the displaced electrons and the equation of motion for them are as follows:

$$\begin{aligned} N m_0 \ddot{x} &= -Ne\mathbf{E} \\ N m_0 \ddot{x} &= -\frac{N^2 e^2 x}{\epsilon_0} \\ \ddot{x} + \omega_p^2 x &= 0 \end{aligned} \tag{2.37}$$

The interpretation of Equation 2.37 is that the free electrons oscillate at ω_p. We assumed that the wavelength is long, so $\mathbf{k} = 0$ and all the electrons move in phase. A quantum of these oscillations are called *bulk plasmons* and since they are of longitudinal nature they do not couple to transverse electromagnetic waves [32].

2.3.2 Surface Plasmon Polaritons

Surface plasmon polaritons are electromagnetic waves propagating along dielectric-metal interfaces, which decay exponentially in the direction perpendicular to the interface. The oscillation of the free electrons (plasma electrons) of the metal are coupled to the electromagnetic fields. We first introduce the necessary concepts, equations and distinctions before delving into the derivations. The insulator (dielectric) is located in the $z > 0$ half space, whereas the conductor (metal) is located in the $z < 0$ half space.

Using the vector identities

$$\nabla \times \nabla \times \mathbf{E} \triangleq \nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E}$$

$$\nabla \cdot (\epsilon \mathbf{E}) \triangleq \mathbf{E} \nabla \epsilon + \epsilon \nabla \cdot \mathbf{E} = \nabla \cdot \mathbf{D} = 0$$

we recast the wave equation Equation 2.33

$$\nabla \left(-\frac{1}{\epsilon} \mathbf{E} \cdot \nabla \epsilon \right) - \nabla^2 \mathbf{E} = -\mu_0 \epsilon_0 \epsilon \frac{\partial^2 \mathbf{E}}{\partial t^2} \tag{2.38}$$

The above equation can be simplified further by assuming the dielectric constant $\epsilon(\mathbf{r})$ has negligible variation over space and the fields are time harmonic with $\mathbf{E}(\mathbf{r}, t) = \mathbf{E}(\mathbf{r})e^{-i\omega t}$. The simplified equation is called *Helmholtz equation* and it reads,

$$\nabla^2 \mathbf{E} + k_0^2 \epsilon \mathbf{E} = 0 \tag{2.39}$$

Throughout this section we assume propagation in the x direction hence the fields should be of the form $\mathbf{E}(x, y, z) = \mathbf{E}(z)e^{i\beta x}$, where β is the propagation constant of the traveling wave[6]. Inserting this into Helmholtz equation yields a simplified wave equation which is the starting point for the analysis of guided electromagnetic modes in waveguides. Please note that an analogous equation exists for

$$\frac{\partial^2 \mathbf{E}(z)}{\partial z^2} + (k_0^2 \epsilon - \beta^2)\mathbf{E} = 0 \tag{2.40}$$

Using the Maxwell curl equations Equation 2.1 and replacing $\frac{\partial}{\partial t}$ with $-i\omega$ we have the

[6] The fields do not depend on y due to the fact that the structure is uniform and extends to infinity in the y direction.

equations coupling electromagnetic fields,

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = i\omega\mu_0 H_x \tag{2.41a}$$

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = i\omega\mu_0 H_y \tag{2.41b}$$

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = i\omega\mu_0 H_z \tag{2.41c}$$

$$\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = -i\omega\epsilon_0\epsilon E_x \tag{2.41d}$$

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = -i\omega\epsilon_0\epsilon E_y \tag{2.41e}$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = -i\omega\epsilon_0\epsilon E_z \tag{2.41f}$$

Equation 2.41 can be simplified further with the replacement of $\frac{\partial}{\partial x} \longrightarrow i\beta$, $\frac{\partial}{\partial y} \longrightarrow 0$, exploiting propagation in the x-directon and the homogeneity in y-direction, respectively.

$$\frac{\partial E_y}{\partial z} = -i\omega\mu_0 H_x \tag{2.42a}$$

$$\frac{\partial E_x}{\partial z} - i\beta E_z = i\omega\mu_0 H_y \tag{2.42b}$$

$$i\beta E_y = i\omega\mu_0 H_z \tag{2.42c}$$

$$\frac{\partial H_y}{\partial z} = i\omega\epsilon_0\epsilon E_x \tag{2.42d}$$

$$\frac{\partial H_x}{\partial z} - i\beta H_z = -i\omega\epsilon_0\epsilon E_y \tag{2.42e}$$

$$i\beta H_y = -i\omega\epsilon_0\epsilon E_z \tag{2.42f}$$

These equations can be further divided into two self-consistent groups with different polarization properties. Combining the first, the third and the fifth equations we only have H_x, H_z and E_y as nonzero and the modes consisting of these fields are called TE modes[7]. Combining the remaining equations we only have E_x, E_z and H_y which constitute the TM modes.

For TE modes Equation 2.42 reduces to the following two equations with E_y as the

[7] They are also called s modes, from German ”senkrecht” meaning perpendicular, by the same token TM modes are also called p modes, again from German ”parallel”.

driving term.

$$H_x = i\frac{1}{\omega\mu_0}\frac{\partial E_y}{\partial z} \tag{2.43a}$$

$$H_z = \frac{\beta}{\omega\mu_0}E_y \tag{2.43b}$$

The governing wave equation is

$$\frac{\partial^2 E_y}{\partial z^2} + (k_0^2\epsilon - \beta^2)E_y = 0 \tag{2.44}$$

Whereas for TM modes Equation 2.42 reduces to the following two equations with H_y as the driving term.

$$E_x = -i\frac{1}{\omega\epsilon_0\epsilon}\frac{\partial H_y}{\partial z} \tag{2.45a}$$

$$E_z = -\frac{\beta}{\omega\epsilon_0\epsilon}H_y \tag{2.45b}$$

The governing TM wave equation is

$$\frac{\partial^2 H_y}{\partial z^2} + (k_0^2\epsilon - \beta^2)H_y = 0 \tag{2.46}$$

In order to investigate the propagation of SPPs on interfaces we focus on the simplest setup possible consisting of two semi-infinite slabs on top of each other as depicted in Figure 2.1, a single interface between a dielectric (insulator) for $z > 0$ and metal (conductor) for $z < 0$. Our aim is to look for propagating waves along the interface that decay exponentially in the direction perpendicular to it.

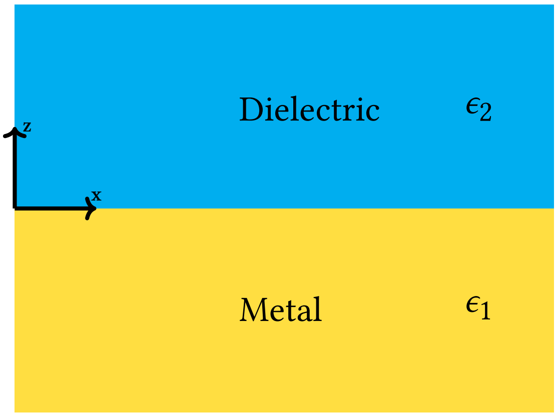


Figure 2.1: Geometry of the the dielectric-metal interface used in the derivations

We focus first on TE modes and analyze their possibility, we employ the notation that $\beta = k_x$, $k_1 \triangleq k_{z,1}$ and $k_2 \triangleq k_{z,2}$. Then the equations for $z > 0$ are

$$E_y(z) = Be^{i\beta x}e^{-k_2 z} \tag{2.47a}$$

$$H_x(z) = -iB\frac{1}{\omega\mu_0}k_2 e^{i\beta x}e^{-k_2 z} \tag{2.47b}$$

$$H_z(z) = B\frac{\beta}{\omega\mu_0}e^{i\beta x}e^{-k_2 z} \tag{2.47c}$$

whereas for $z < 0$ they are

$$E_y(z) = Ae^{i\beta x}e^{k_1 z} \tag{2.48a}$$

$$H_x(z) = iA\frac{1}{\omega\mu_0}k_1 e^{i\beta x}e^{k_1 sz} \tag{2.48b}$$

$$H_z(z) = A\frac{\beta}{\omega\mu_0}e^{i\beta x}e^{k_1 z} \tag{2.48c}$$

Referring back to continuity equations for electromagnetic fields Equation 2.11, E_y and H_x must be continuous at the interface. It immediately follows $A = B$ and hence

$$A(k_1 + k_2) = 0 \tag{2.49}$$

In order for the waves to decay exponentially away from the surface we need to have $\Re(k_1) > 0$ and $\Re(k_2) > 0$, so Equation 2.49 implies $A = B = 0$. This concludes that surface plasmon polaritons do not exist for *TE polarization.*

Lastly we focus on TM modes and utilize the same notation as described before. The equations for the upper half with positive z are

$$H_y(z) = Be^{i\beta x}e^{-k_2 z} \tag{2.50a}$$

$$E_x(z) = iB\frac{1}{\omega\epsilon_0\epsilon_2}k_2 e^{-k_2 z} \tag{2.50b}$$

$$E_z(z) = -B\frac{\beta}{\omega\epsilon_0\epsilon_2}e^{i\beta x}e^{-k_2 z} \tag{2.50c}$$

and for negative z we have

$$H_y(z) = Ae^{i\beta x}e^{k_1 z} \tag{2.51a}$$

$$E_x(z) = iA\frac{1}{\omega\epsilon_0\epsilon_2}k_1 e^{k_1 z} \tag{2.51b}$$

$$E_z(z) = -A\frac{\beta}{\omega\epsilon_0\epsilon_2}e^{i\beta x}e^{k_1 z} \tag{2.51c}$$

Making use of the continuity of H_y and E_x we have the following relations

$$A = B \tag{2.52}$$

$$\frac{k_2}{k_1} = -\frac{\epsilon_2}{\epsilon_1} \tag{2.53}$$

Since both k_1 and k_2 are positive in the convention used, in order to have a electromagnetic wave confined to the surface the real parts of the permittivities ϵ must have opposite signs. That is $\Re(\epsilon_1) < 0$ because of the fact that the dielectric has a positive ϵ_2. On top of these H_y has to satify the TM wave equation Equation 2.46 derived before which imposes

$$k_1^2 = \beta^2 - k_0^2\epsilon_1 \tag{2.54a}$$

$$k_2^2 = \beta^2 - k_0^2\epsilon_2 \tag{2.54b}$$

Combining Equation 2.52 with Equation 2.54 we arrive at the SPP dispersion relation which is plotted in Figure 2.2.

$$\beta = k_0\sqrt{\frac{\epsilon_1\epsilon_2}{\epsilon_1 + \epsilon_2}} \tag{2.55}$$

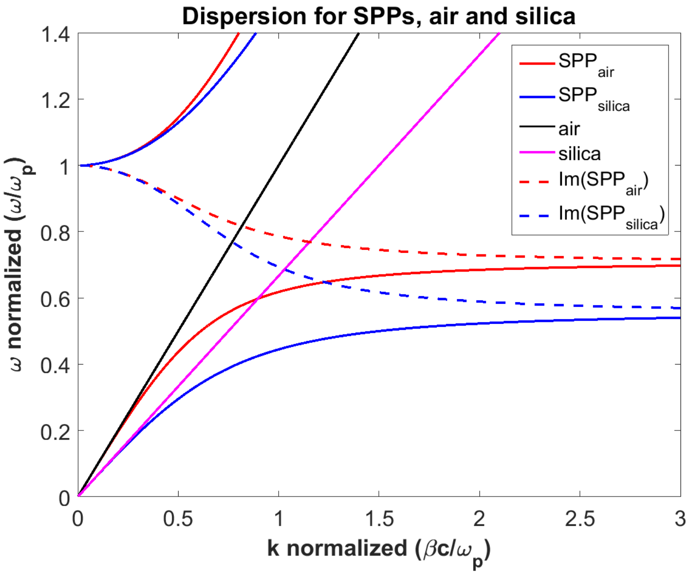


Figure 2.2: SPP dispersion

Chapter 3

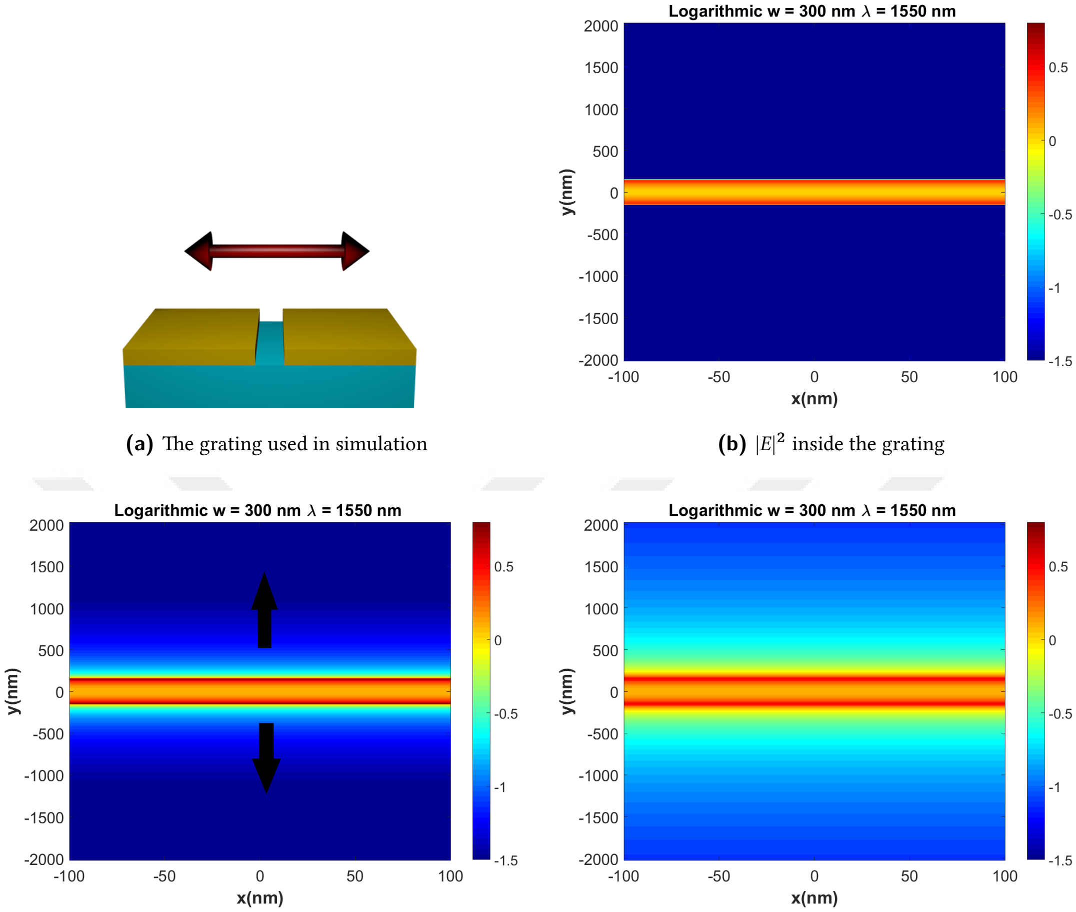
SINGLE SCATTERERS

In this study we are going to focus on three different types of single scatterers; the 1D grating by Iqbal T. and Afsheen S. [33], the rectangular slit by Tanemura T. et al. [2, 24], and the Δ-antenna by Bai B. et al. [14, 34, 35]. In order to make these designs compatible with our optical setup and fabrication procedure some modifications are made, which are discussed in the following sections. The simulations were done in Lumerical FDTD Solutions commercial package [36].

3.1 *Single Grating*

We start with the simplest case that is a single infinite perforation on the gold film surface. There are various grating designs some of which are not compatible with our fabrication process such as the ones by Chen C. and Berini P. [37, 38] due to the different substrate used, namely Si_3N_4, and the presence of the gold layer underneath the grating. It is not possible to manufacture structures with two layers of metal using standard e-beam lithography with a single resist layer. Therefore we employ the design from Iqbal T. [33] which is compatible with our manufacturing processes. We start by simulating a single infinite perforation of a finite width on the metal film to characterize a single scatterer. The width of this infinite perforation is scanned from 125 nm to 350 nm in 25 nm increments while keeping all the other simulation parameters constant.

In this case it is not logical to characterize the single infinite slit with the angular distribution of the electric field norm squared/phase of E_z as the wave vector of the scattered light is always perpendicular to the infinite side of the groove. Therefore the plots of the electric field on the surface, 20 nm above the surface and inside the infinite slit are provided.



(a) The grating used in simulation

(b) $|E|^2$ inside the grating

(c) $|E|^2$ on the metal-air interface

(d) $|E|^2$ 20 nm above the metal-air interface

Figure 3.1: Grating Design, Width = 300 nm λ = 1550 nm

In Figure 3.1a the red arrow indicates the polarization direction of the incident light from the substrate side. We observe that the SPPs excited on both sides of the infinite slit are identical and their field profile do not vary over the x-axis (parallel to the infine side of the slit). The black arrows on Figure 3.1c indicate the direction of SPP propagation on the metal-air interface. The theoretical wavelength of the SPPs defined from Equation 2.55 as $\lambda_{SPP} = 2\pi/\Re(\beta)$ is 1543 nm, whereas the calculated SPP wavelength from the simulation data is 1536 nm. The wavelength is calculated by measuring the distance between the crest-trough of the wave and multiplying by 2. The imaginary part of the propagation constant $\Im(\beta)$ is 1.38×10^4; this is an order of magnitude larger compared to the theoretical value of 1.77×10^3. This discrepancy is caused since SPPs are not the only waves excited on the interface and inspecting

Figure 3.2 with a logarithmic y-axis shows that waves with different power decay laws are excited.

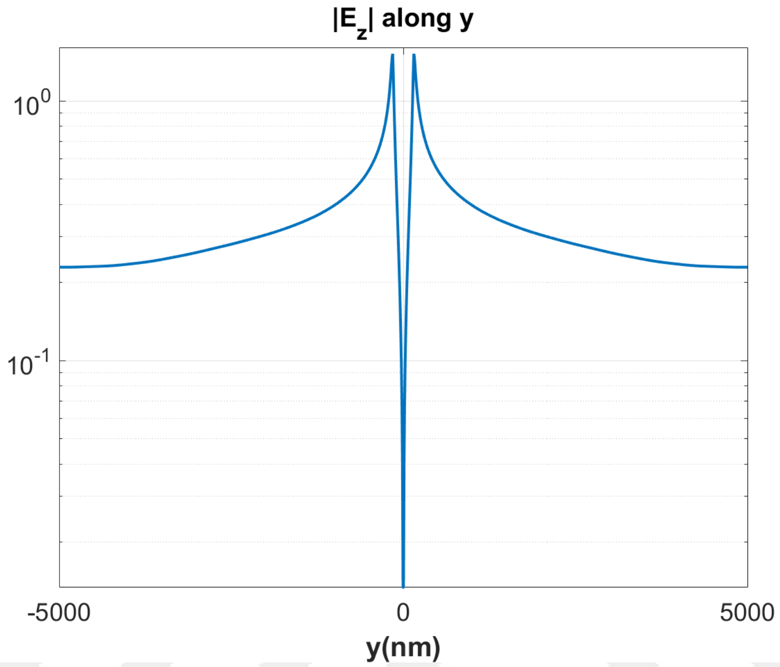


Figure 3.2: $|E_z|$ along propagation direction

3.2 *Rectangular Nanoslit*

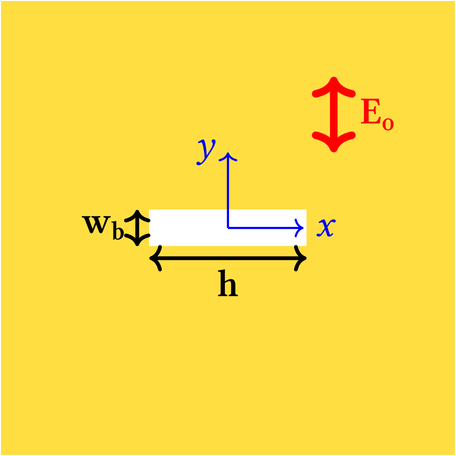
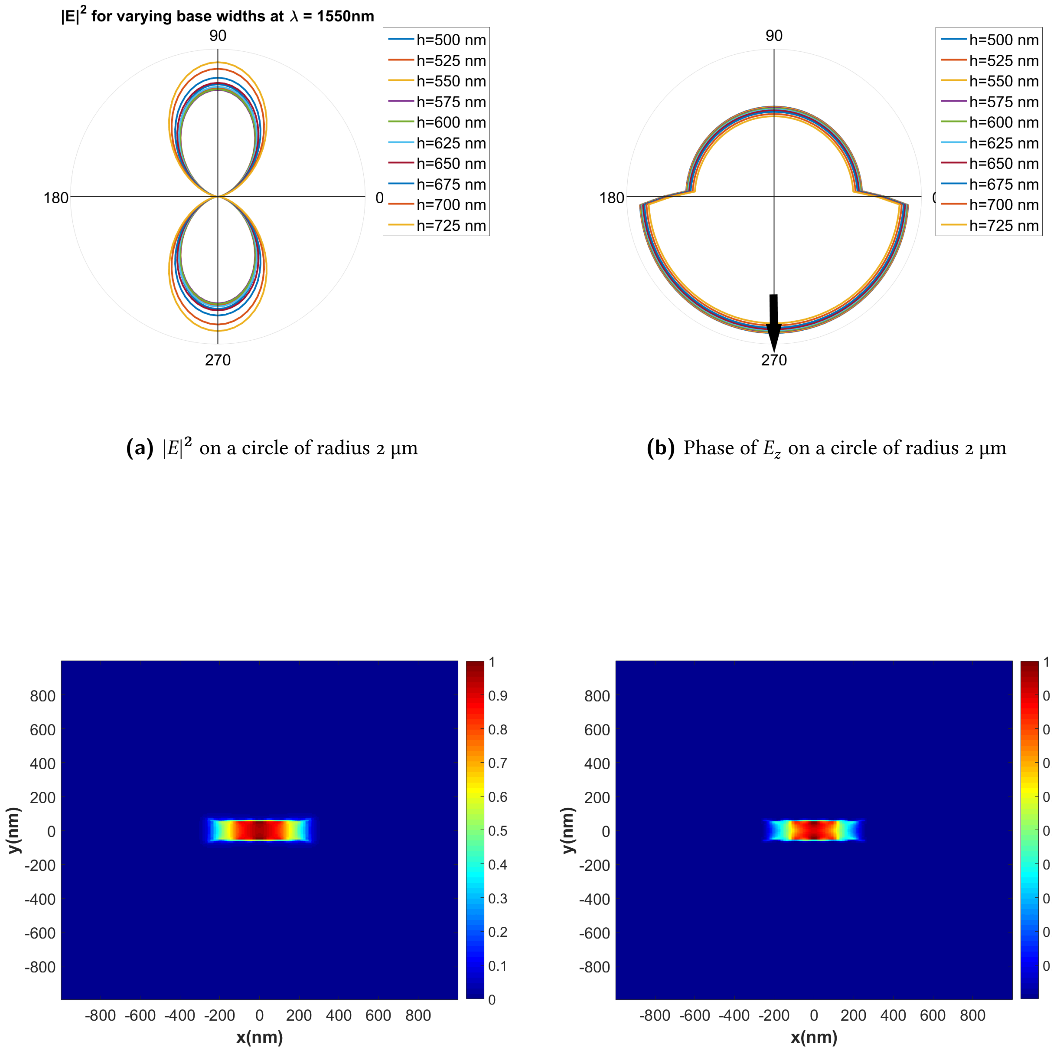


Figure 3.3: Nanoslit top down view

Next we move on to the original rectangular nanoslit design perforated on gold film by Tanemura T. et al. features nanoslits of width 120 nm and height 520 nm [2]. These nanoslits which are perforated on a gold film of 75 nm thickness are illuminated from the air side and the excited SPPs propagate on the metal-dielectric interface. Our design differs from this in that we use a thicker gold layer, 115nm, and the nanoslits are illuminated from the substrate side. Polarization of the normally incident field is in the x-direction as indicated on Figure 3.3 with the red double-sided arrow.The nanoslits are expected to have a symmetric radiation pattern with respect to y-axis. After the field intensity is normalized with the largest value in the dataset it is scaled by C/h in order to isolate the excited field magnitude per unit width along the height of the nanoslit. Here h is the height of the nanoslit and C is some constant. To characterize the angular radiation pattern of the nanoslit the square of the magnitude of the electric field and the phase of the z component of the electric field (E_z) is examined 20 nm above the metal-air interface.

In Figure 3.4b the black arrow indicates the direction of decreasing height for the nanoslits. For Figure 3.4a such a pattern is not recognizable as the trend changes direction at the nanoslit height of 625 nm.



(a) $|E|^2$ on a circle of radius 2 μm

(b) Phase of E_z on a circle of radius 2 μm

(c) Mode from Lumerical FDTD Solutions

(d) $|E|^2$ evaluated at the middle of the metal film

Figure 3.4: Nanoslit Design, Width = 525 nm λ = 1550 nm

We also performed fitting using the built-in *griddedInterpolant* function of Matlab for linear interpolation [39]. These fits are utilized in Chapter 6 to determine the locations of Δ-antennas on the wavelength demultiplexer with the iterative algorithm presented in Chapter 5. The interpolation is done on a circle with the Δ-antenna in the center, and the fit is done by assuming that the SPP field is proportional to the following expression:

$$E \propto \frac{f(\theta)}{\sqrt{|\mathbf{r}^{\text{Field}} - \mathbf{r}^{\text{Source}}|}} \exp[ik^{\text{SPP}}(\mathbf{r}^{\text{Field}} - \mathbf{r}^{\text{Source}})] \tag{3.1}$$

In the above expression the attenuation and oscillation of SPP waves are contained in the exponential term and the denominator serves for normalization such that if there were no attenuation the optical energy flux would be the same for circles of varying radii with their center on the scatterer. The $f(\theta)$ term represents the dependence of the scattered field on the geometry of the scattter and for the nanoslit which is approximated to have a point dipole radiation pattern $f(\theta) = \cos(\theta)$ as in the original expression by Steele et al. [40]. The geometry of the setup used for the fit is illustrated in Figure 3.5; here $R_1 = 2$ µm and $R_2 = 4$ µm.

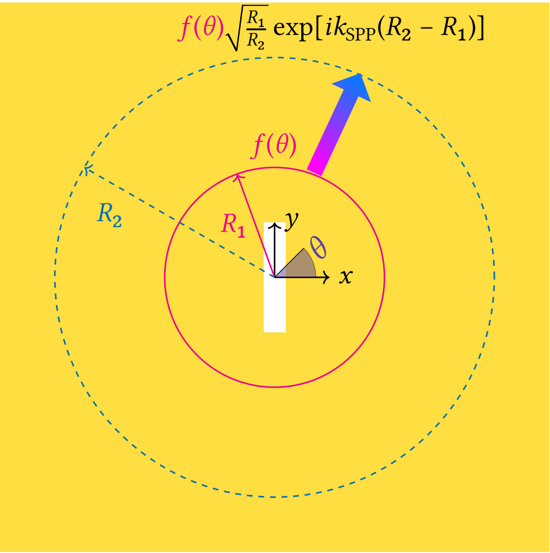
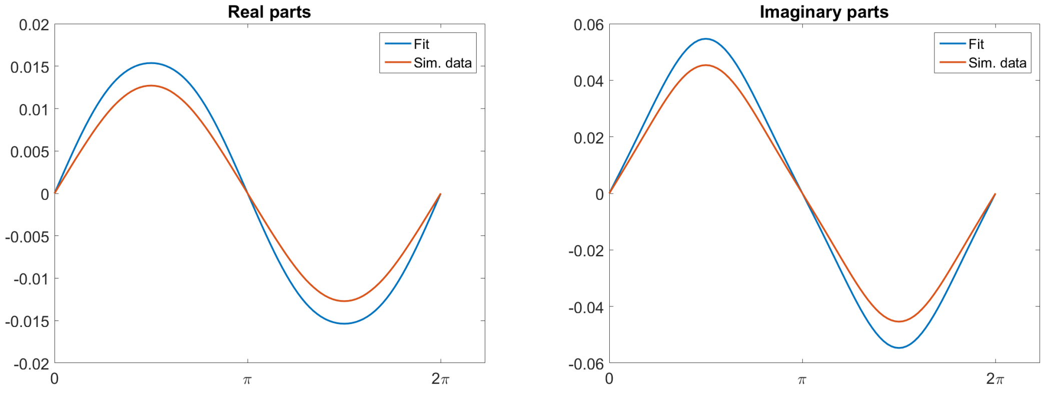

$f(\theta)\sqrt{\frac{R_1}{R_2}}\exp[ik_{\text{SPP}}(R_2 - R_1)]$
$f(\theta)$
R_2
R_1
y
θ
x

Figure 3.5: Illustration of the fitting procedure and coordinates for nanoslit

The fit is performed with the following steps:

1. Simulation is performed on a big domain of 10 µm × 10 µm in xy plane
2. Data is exported to Matlab for post-processing
3. An interpolation is done to have the data on a circle of radius R_1
4. Assuming only SPPs are excited $E_{R_1} \propto \frac{f(\theta)}{\sqrt{R_1}} \exp(ik_{\text{SPP}}R_1)$
5. By the same token $E_{R_2} \propto \frac{f(\theta)}{\sqrt{R_2}} \exp(ik_{\text{SPP}}R_2)$
6. So $E_{R_2} = \sqrt{\frac{R_1}{R_2}} \exp[ik_{\text{SPP}}(R_2 - R_1)]$

The obtained fits are compared with the interpolation from the simulation data in Figure 3.6. The fits are acceptable; however, there are some differences due to the excitation of quasicylindrical waves (quasi-CW) along with SPPs. The excitation of quasi-CW is dominant for $\lambda > 1000$ nm with noble metals [41].



(a) The real part of the fit over the circle

(b) The imaginary part of the fit over the circle

Figure 3.6: Comparison of fit with interpolation for nanoslit

In our algorithm we selected a nanoslit of height 500 nm and width of 100 nm since a nanoslit of this dimension is resonant at a free-space wavelength of 1550 nm. In order to determine the resonant height we scanned the height of the slit while keeping its width constant, and measured the forward scattering cross section by normalizing the transmitted power in the forward direction with the intensity of the source. Later in order to get a dimensionless quantity we further normalized this value with the geometric cross section of the nanoslit. Figure 3.7 illustrates the obtained results.

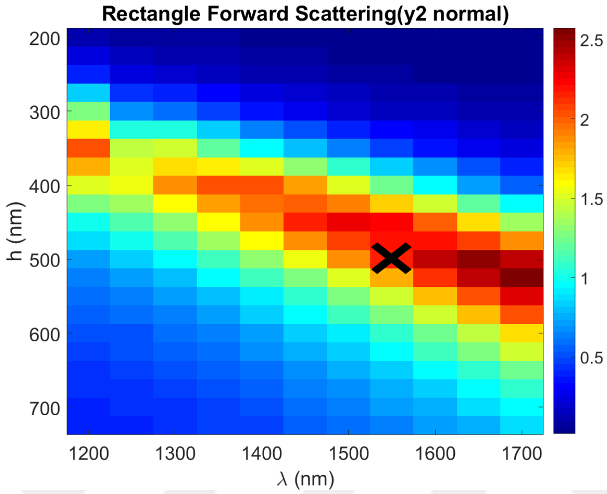


Figure 3.7: Forward scattering of nanoslit

3.3 Δ-*antenna*

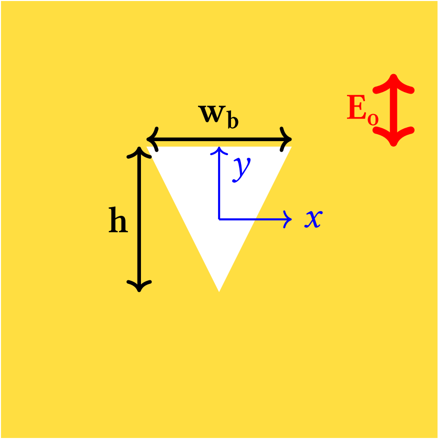
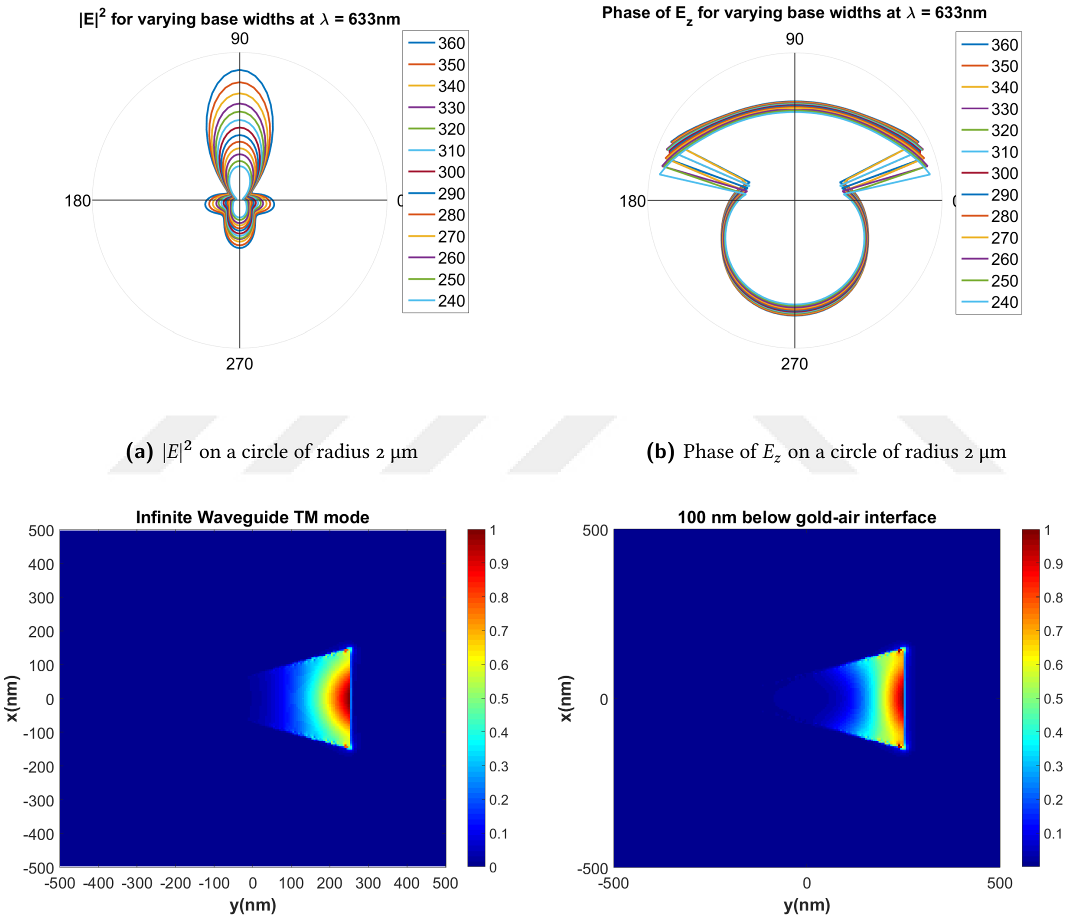


Figure 3.8: Δ-antenna top down view

The original Δ-antenna design [14] is an isosceles triangle hole perforated on an optically thick (200 nm) gold film on fused silica substrate. The excitation beam is normally incident on the Δ-antenna with polarization perpendicular to the base of the triangle (Figure 3.8). The antenna is back illuminated (from the substrate side) and the vacuum wavelength of the illumination light is $\lambda_0 = 633$ nm. The dielectric constant of gold is $\epsilon_m = -11.10 + 1.29i$, and the dielectric constant of air is $\epsilon_d = 1$ at the wavelength of interest.

This design served as a benchmark tool for our simulations and subsequent designs. In order to characterize angular radiation pattern of Δ-antennas we performed the same analysis in the previous section. It should also be noted that the field intensities are normalized with respect to the maximum value and multiplied by 360 nm/w_b[1] in order to isolate the excited field by unit width of the Δ-antenna.

In order to compare our results with those in the original article by You B. [14] we evaluated the electric field 20 nm above the gold-air interface on a circle of radius 2 µm. The antennas were of the same dimension as in the article, namely, 300 nm in width and 505 nm in height.



(a) $|E|^2$ on a circle of radius 2 µm

(b) Phase of E_z on a circle of radius 2 µm

(c) TM mode from Lumerical FDTD Solutions

(d) $|E|^2$ evaluated at the middle of the metal film

Figure 3.9: Δ-antenna original design with w_b = 300 nm and h = 505 nm

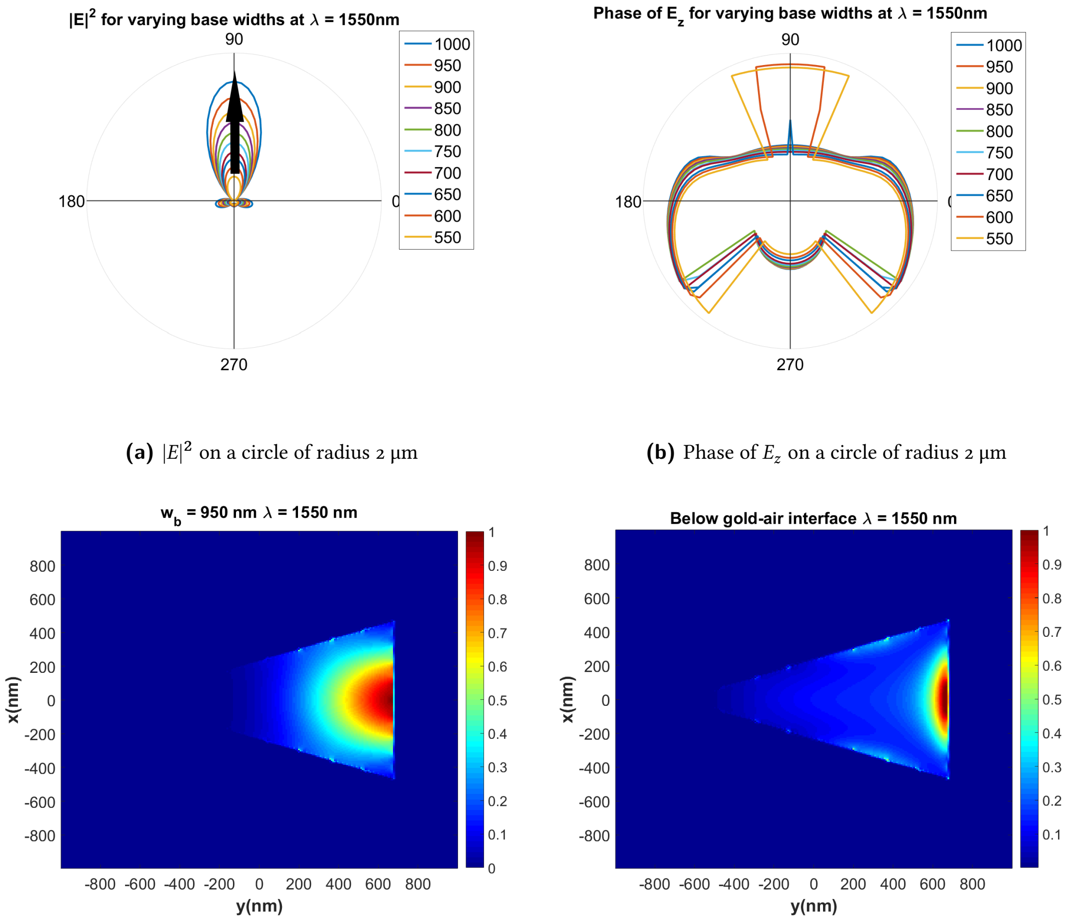
In Figure 3.9a and Figure 3.9b the base width of the Δ-antenna is scanned from 240 nm to 360 nm in 10 nm increments. The black arrows on the figures indicate the direction of increasing base width.

In our laboratory we have an optical setup with a Fianium SC450 supercontinuum laser source and an AOTF with two NIR channels: NIR1 in the range 650-1100 nm and

[1] w_b is the base width of the isosceles triangle.

NIR2 to in the range 1100-2200 nm. Since we work mostly with the NIR2 channel at 1550 nm, we need to employ a Δ-antenna design with large dimensions.

The same Δ-antenna design with different dimensions was utilized before twice. Firstly, in 2009 in a blazing grating working at normal incidence with back illumination at 1530 nm [35]. To accommodate for the increased source wavelength the basewidth and the height of the Δ-antennas were scaled up to 960 nm and 1360 nm, respectively. Secondly, in 2014 a binary area-coded nanohole array is formed with Δ-antennas which have h = 1057 nm and w_b = 570 nm. This time the source laser operated at 1064 nm [34].



(a) $|E|^2$ on a circle of radius 2 μm

(b) Phase of E_z on a circle of radius 2 μm

(c) Mode solution from Lumerical FDTD Solutions

(d) $|E|^2$ evaluated at the middle of the metal film

Figure 3.10: Δ-antenna revised design for NIR

In Figure 3.10a and Figure 3.10b the black arrows indicate the direction of increasing base width. w_b is scanned from 550 nm to 1000 nm in 50 nm increments while keeping the height constant at 1360 nm. In Figure 3.10c and Figure 3.10d the magnitude of the modal electric field and electric field in the middle of the film are illustrated, respectively. Since the largest base width is 1000 nm we multiply with 1000 nm/w_b after the normalization is done. We perform the same analyses of evaluation of the electric field norm squared and the phase of the E_z on a 2 µm radius circle. At 1550 nm the revised design retains its unidirectional scattering capability.

The unidirectionality of the Δ-antenna depends on the base width, height of the antenna and the wavelength of the incident light. Therefore there is a trade-off between the ratio of forward scattered SPPs to backward scattered SPPs and the total scattered power. In Figure 3.11 the depence of this ratio on height is illustrated for a Δ-antenna of basewidth 500 nm.

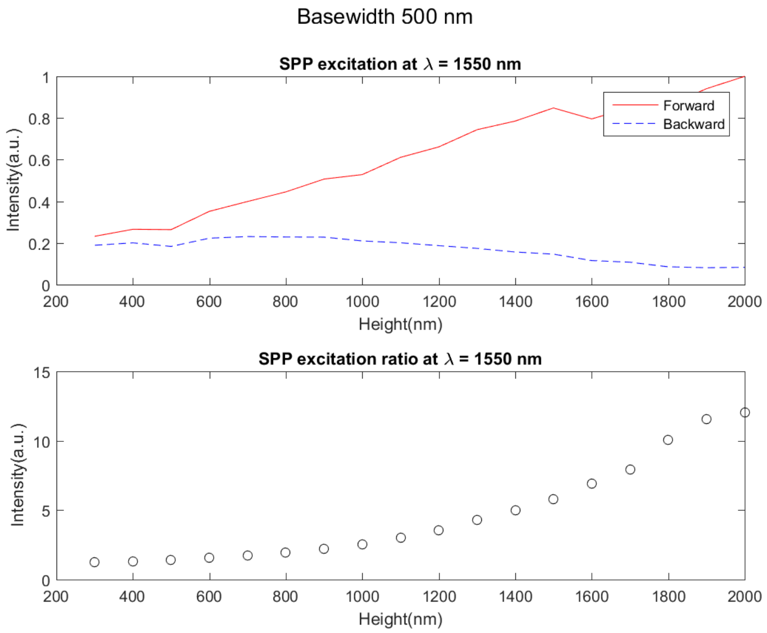


Figure 3.11: SPP amplitude in forward and backward directions (above); and their ratio (below) at 500 nm basewidth

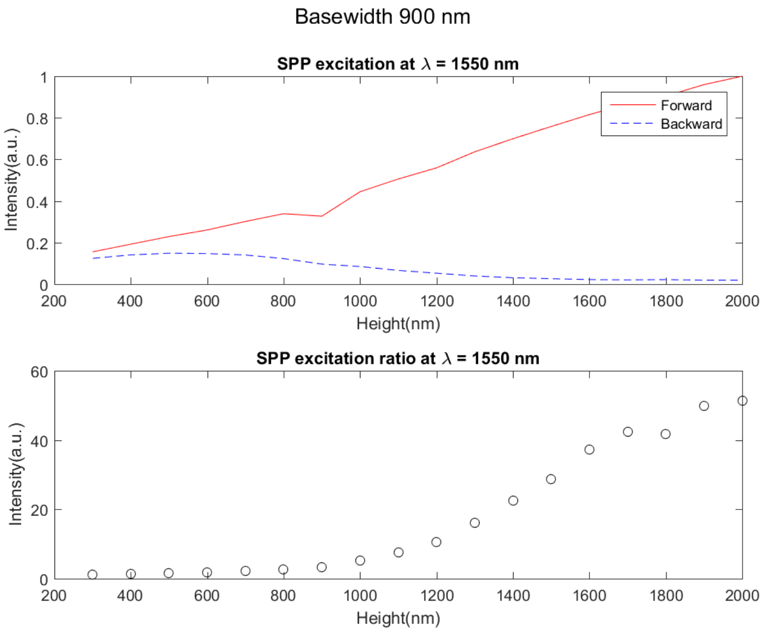


Figure 3.12: SPP amplitude in forward and backward directions (above); and their ratio (below) at 900 nm basewidth

The dependence of the intensity of the forward propagating and backward propagating SPPs is different for a Δ-antenna with a basewidth of 900 nm, yet the trend is similar in the sense that the higher the height is the more the SPPs are excited in the forward direction (Figure 3.12).

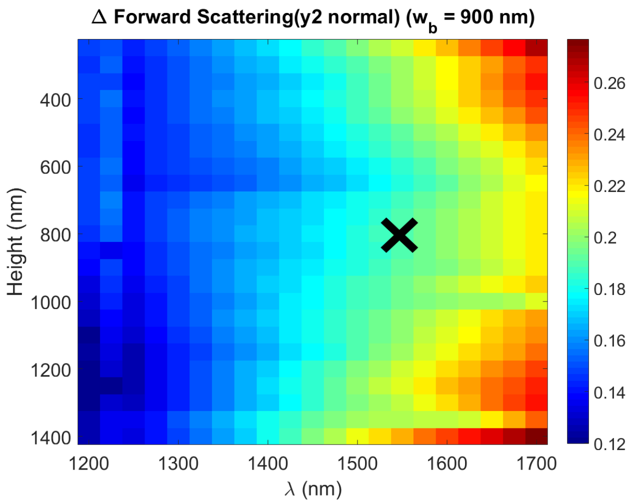


Figure 3.13: Forward scattering of Δ-antenna

Before performing the fits we further perform a sweep over the height of the Δ-antenna to find the height at which the most forward scattering occurs just like we did for the nanoslits. We have chosen a Δ-antenna design with $w_b = 900\,\text{nm}$ and $h = 800\,\text{nm}$ (Figure 3.13). We had to make compromises on the unidirectionality of the Δ-antenna as our primary concern is to concentrate the maximum optical power to the focal point. We were not able to determine the dimension the Δ-antenna is resonant at an incident wavelength of $\lambda = 1550\,\text{nm}$ after doing simulations keeping the h fixed while sweeping over w_b and keeping w_b fixed while sweeping over h.

The fit is performed by interpolating the data from a single Δ-antenna on a circle of radius 2 μm and extracting the angular dependence of the scattering pattern from there. At the point interest this dependence is scaled according to the distance and its phase is calculated taking k_{SPP} into account. The orientation of the Δ-antenna in this fit is given in Figure 3.14.

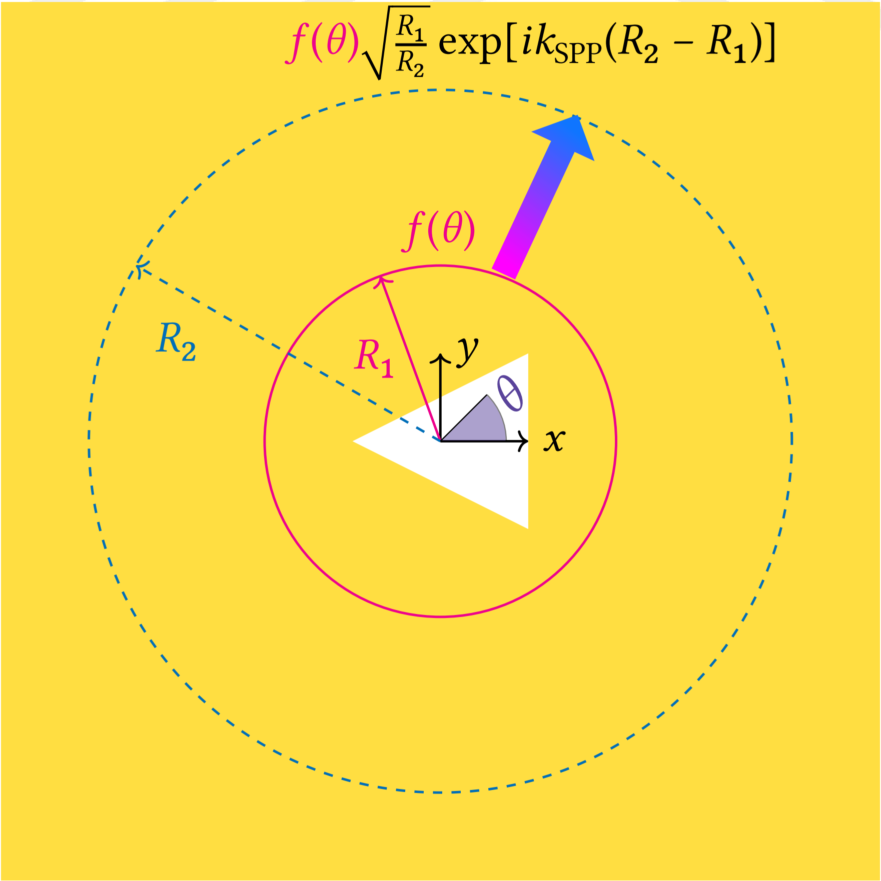
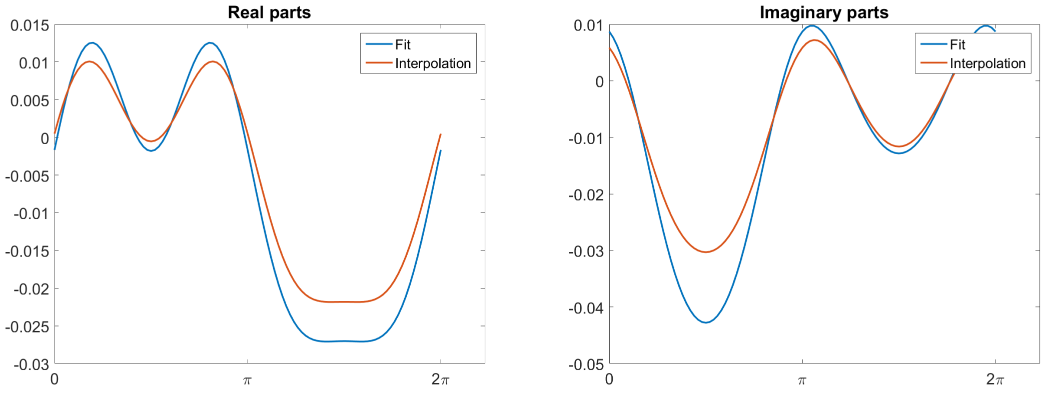

$f(\theta)\sqrt{\frac{R_1}{R_2}}\exp[ik_{\text{SPP}}(R_2 - R_1)]$
$f(\theta)$
R_2
R_1
y
θ
x

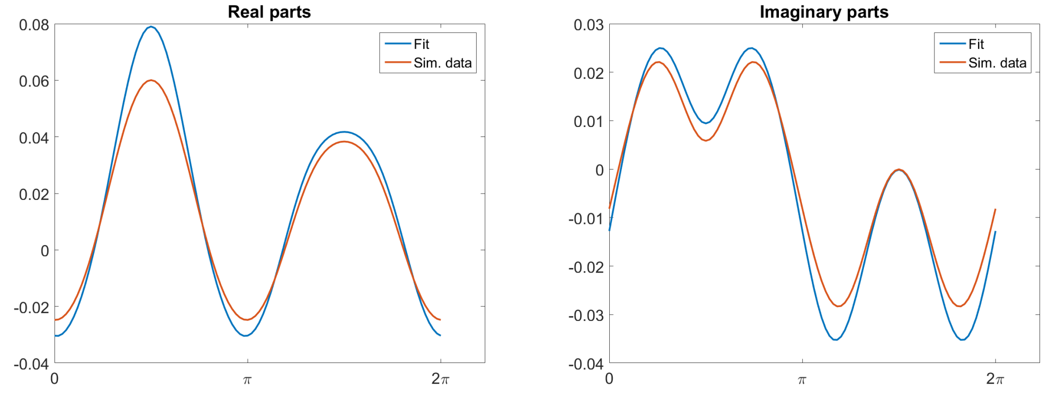
Figure 3.14: Illustration of the fitting procedure and coordinates for Δ-antenna



(a) The real part of the fit over the circle

(b) The imaginary part of the fit over the circle

Figure 3.15: Δ-antenna (w_b = 500 nm, h = 800 nm) comparison of fit with interpolation



(a) The real part of the fit over the circle

(b) The imaginary part of the fit over the circle

Figure 3.16: Δ-antenna (w_b = 900 nm, h = 800 nm) comparison of fit with interpolation

As seen in Figure 3.15 and in Figure 3.16 the fits are acceptable, yet they are not perfect. The fits for both the nanoslits and the Δ-antennas are not exact due to the presence of quasi-cylindrical waves (quasi-CW) which are excited together with SPPs as proposed by Haiato Liu and Philippe Lalanne [41] and Norton waves which are weak in noble metals compared to the preceding two waves [42]. The quasi-CW are attenuated much faster than SPPs over the entire electromagnetic spectrum for noble metals and these waves convert into each other [43].

In the following table the transmitted optical power as a fraction of total input optical power is given. The cross-section is taken to be oriented perpendicular to the direction in which the most scattering occurs and is situated 750 nm away in this direction from the scatterer center in each case.

	Δ-antenna	Rectangular Nanoslit	1D single grating
Transmission %	0.63	0.08	0.61
Width	950 nm	100 nm	300 nm
Height	1360 nm	500 nm	∞

Table 3.1: Transmission percentage as a ratio of input power at λ = 1550 nm

Chapter 4

SPP BEAM LAUNCHER

The main aim of this study is to couple normally incident light to SPPs and then into stripe waveguide modes. In doing so it is of our best interest to couple the maximum amount of energy possible from the incident light into propagating SPPs. In this chapter we try to realize this by the use of a SPP beam launcher. SPP beam launchers excite a given SPP mode with a well defined amplitude and phase.

4.1 Reproduction of Hermite-Gauss SPP Beam

This section is about launching specific SPP modes with a well defined amplitude and phase. We start by testing our implementation of the Hermite-Gauss SPP beam launcher originally proposed by You O. [14]. After verification of the algorithm our objective is to launch an SPP beam having the same amplitude and phase profile with the fundamental mode of a stripe waveguide to couple into this mode.

We analyze the generation of a Hermite-Gauss SPP beam by exploiting the unidirectionality of individual Δ-antennas in the same manner as done by You O. [14]; the objective here is to compare our simulation technique and test its robustness.

The parameters are the same as used in Section 3.3. The analytical expression for the z-component of the electric field of a Hermite-Gauss SPP beam is as follows[1]:

$$\begin{aligned} E_{z,d}(x,y,z) = A_l \sqrt{\frac{W_0}{W(y)}} H_l \left[\frac{\sqrt{2}x}{W(y)}\right] \exp\left[\frac{-x^2}{W^2(y)} - k_{imag}(y-y_0)\right] \\ \exp\left[ik_{real}(y-y_0) + ik_{real}\frac{x^2}{2R(y)} + i\left(\frac{1}{2}+l\right)\phi\right]\exp(-\kappa z) \end{aligned} \tag{4.1}$$

[1] It should be noted that z-direction is the direction perpendicular to the metal-dielectric interface and the subscript d refers to the field inside the dielectric

The definitions of κ, $W(y)$, $R(y)$ and $\phi(y)$ are as follows

$$\kappa = \sqrt{k_{SPP}^2 - \epsilon_d k_0^2} \tag{4.2}$$

$$W^2(y) = W_0^2 \left\{ 1 + \left[\frac{2(y - y_0)}{k_{real} W_0^2} \right]^2 \right\} \tag{4.3}$$

$$R(y) = (y - y_0) \left\{ 1 + \left[\frac{k_{real} W_0^2}{2(y - y_0)} \right]^2 \right\} \tag{4.4}$$

$$\phi(y) = -\arctan \left[\frac{2(y - y_0)}{k_{real} W_0^2} \right] \tag{4.5}$$

Here κ is the field attenuation constant in the direction normal to the metal-dielectric interface, $W(y)$ is the beam width, R(y) is the curvature of the beam, $\phi(y)$ is the additional phase, A_l is the amplitude, y_0 is the position of the beam waist in y, k is the wave vector of the SPP and H_l is the Hermite polynomial of order l. It should be noted that since we deal with second order Hermite-Gauss we are concerned only with the case in which $l = 2$.

Ignoring the constant coefficients the Equation 4.2 can be rewritten as follows:

$$|E_z| \exp(i\Phi) = H_2 \left[\frac{\sqrt{2}x}{W(y)} \right] \exp \left[\frac{-x^2}{W^2(y)} \right] \exp \left[ik_{real} \frac{x^2}{2R(y)} \right] \tag{4.6}$$

Conforming with the procedure in the article we take $y_0 = 15$ µm and perform the following steps to populate the Δ-antenna array.

1. The amplitude and phase of the E_z is calculated at the exit line $y = 0$ and the amplitude is normalized to 1.

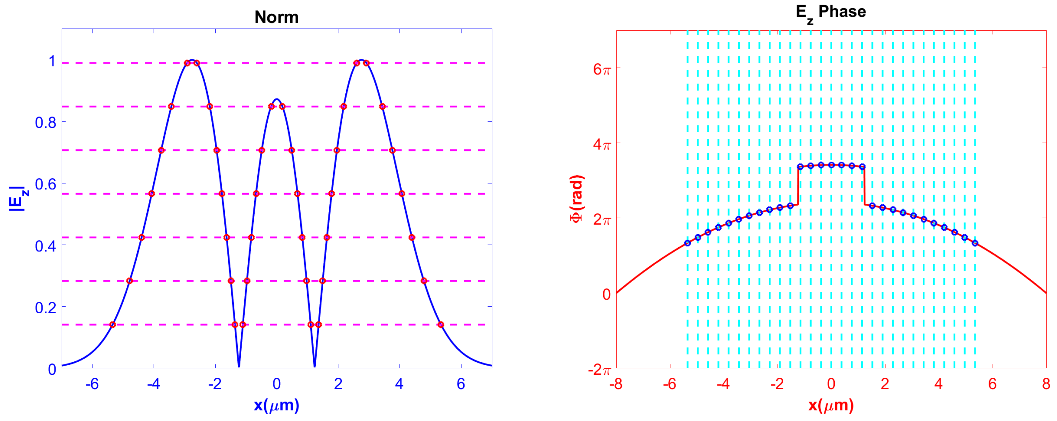
2. Determine how many rows of antennas are needed to reproduce the output beam, which is denoted by N. As in the article we select this to be $N = 7$.

3. The amplitude profile of the desired beam is discretized as $\delta|E_z| = |E_z|_{max}/N$ by drawing horizontal lines intersecting the amplitude profile as in Figure 4.1a.

4. Decide on the number of rows of Δ-antennas needed by looking at the y-intercept of the intersection point. For example if it is 2/7 in our case, it means that 2 rows of Δ-antennas are required.

5. Determine the base widths of the antennas by selecting an appropriate number of delta antennas to fit into the line segment(s) that is(are) truncated by the amplitude profile on both sides.

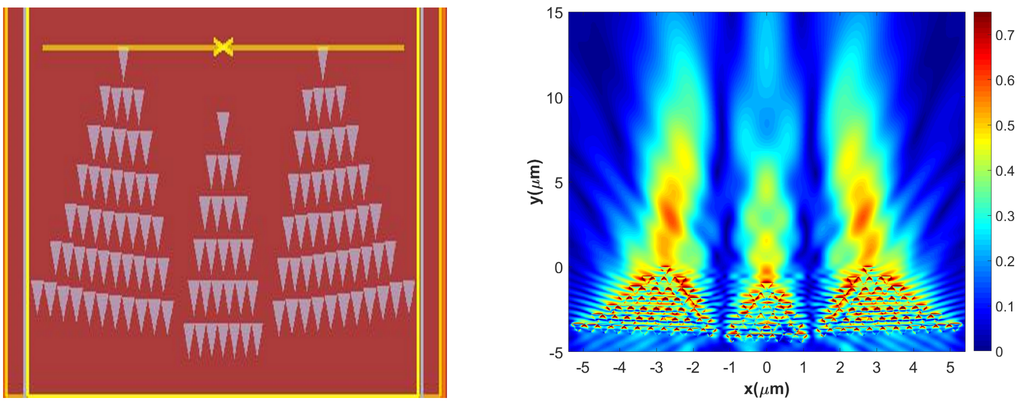
6. Fine tune the y positions of the antennas in order to reproduce the required phase difference after discretization of the phase profile by vertical lines this time as in Figure 4.1b. The individual antennas in columns are moved by $\Delta y = \Phi / k_{SPP,real}$.



(a) Discretization of amplitude profile in Matlab

(b) Discretization of phase profile in Matlab

Figure 4.1: Discretization of the 2^{nd} order Hermite-Gauss SPP beam

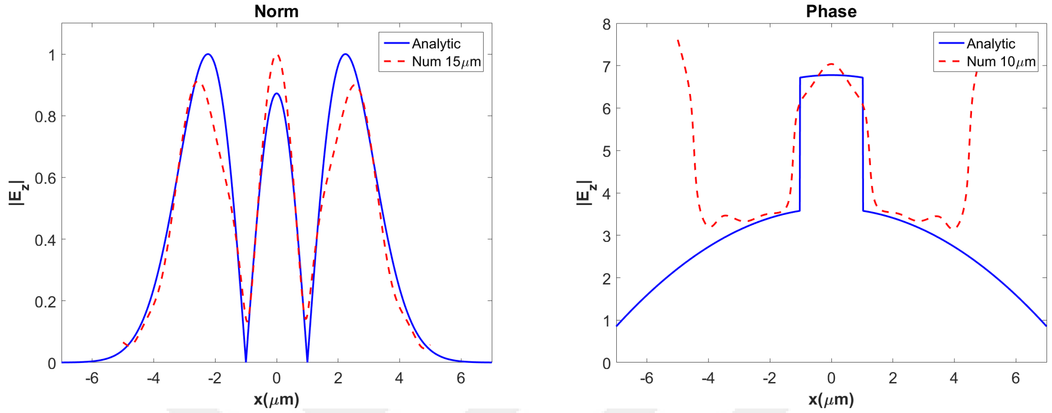


(a) Positions of the antennas in Lumerical

(b) Magnitude of E_z on the surface

Figure 4.2: Lumerical FDTD Solutions setup & results

After these discretization are made the antennas are placed in Lumerical FDTD Solutions as in Figure 4.2a, and a simulation is done. The results conform with those of the article and they are given in the following figures.



(a) $|E_z|$ from FDTD and analytical results at y = 15 µm **(b)** $\arg(E_z)$ from FDTD and analytical results at y = 10 µm

Figure 4.3: Results some distance away

In Figure 4.2b the magnitude of E_z on the surface of the device is shown which is similar to the same plot in the article. The comparison between the analytical expression and the numerical result for the $|E_z|$ and phase of E_z is provided in Figure 4.3a and Figure 4.3b, respectively.

4.2 Stripe Waveguide Coupling

We begin our analysis by finding the amplitude and phase profile of the electric field in the z-direction for the fundamental mode of a stripe waveguide of 1 µm.

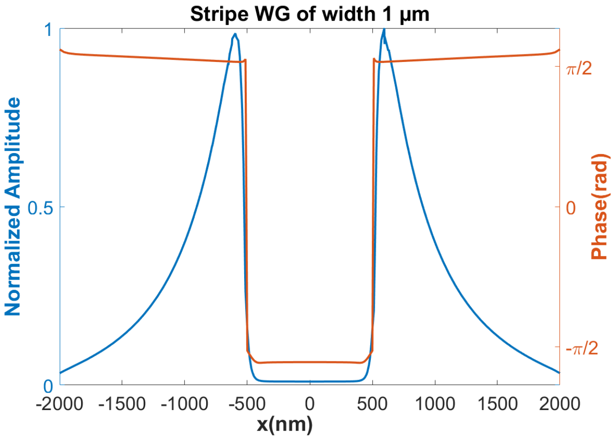
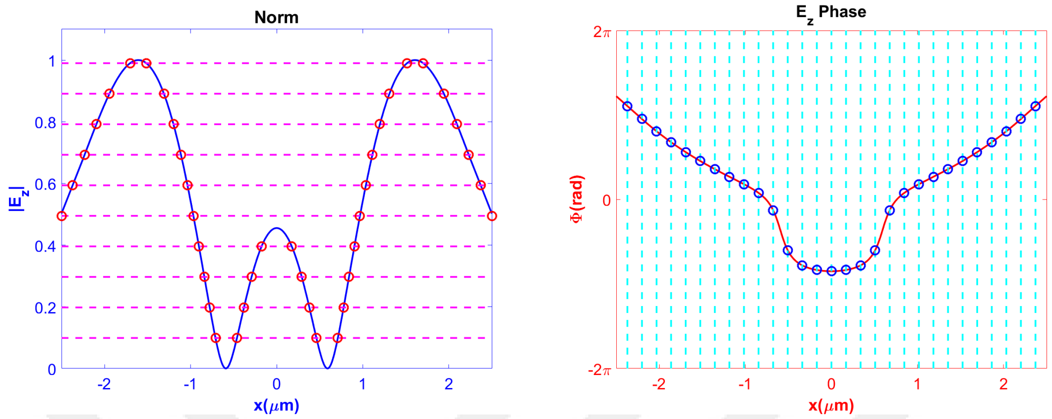


Figure 4.4: $|E_z|$ and $\arg(E_z)$ from Comsol

In Figure 4.4 the amplitude and phase of the normal (z-component) electric field are shown together. From the plot it is evident that the amplitude of the normal electric field changes rapidly, which is a problem for the SPP beam launcher scheme proposed in the previous subsection. This is due to the fact that the finest feature in the amplitude or phase profile of the desired SPP beam it can cope with is a slow variation on the order of the basewidth of the individual Δ-antennas (≈ 300 nm). Note that the original design features an SPP beam whose profile spans 12 µm.

We decided to try it anyways and proceeded with discretizing the amplitude and phase of the z-component of the electric field with the method outlined in the previous subsection, but we were not able to discretize the electric field profile since the basewidth of the antennas were large compared to the fast decay of the amplitude. We could not afford shortening the basewidth much below 300 nm as the performance of the antennas will be severely comprimised.

Then we decided to devise an alternative approach in which the fundamental mode of the stripe waveguide is launched to propagate towards the gold film and the z-component of the electric field is examined on the gold film 1 µm away from the start of the stripe waveguide.

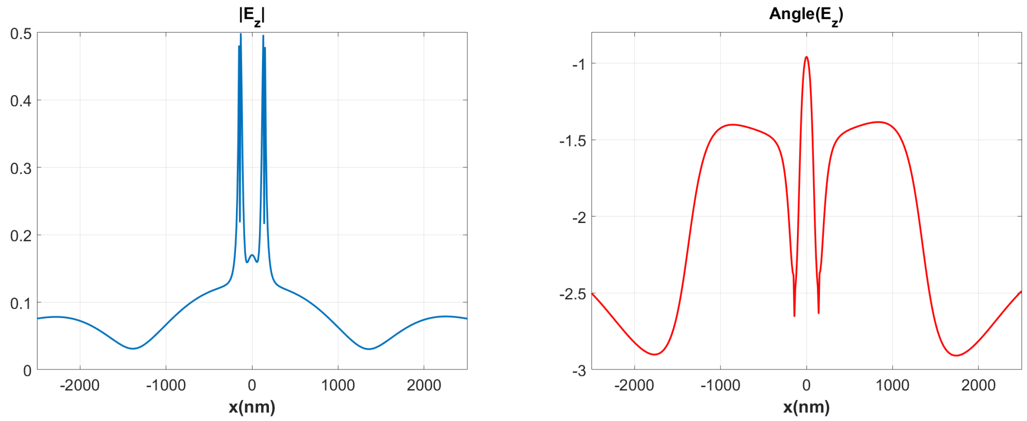


(a) Discretization of amplitude profile in Matlab

(b) Discretization of phase profile in Matlab

Figure 4.5: Discretization of the E_z field after modal excitation

In Figure 4.5a and Figure 4.5b the discretization used for the amplitude and phase of the z-component of the electric field is shown, respectively. Our aim is to obtain a similar amplitude and phase profile on the same line after placing the Δ-antenna array and exciting SPPs with light normally shone from the air side.



(a) $|E_z|$ on the exit line

(b) Phase of E_z on the exit line

Figure 4.6: E_z on the exit line

In Figure 4.6 the amplitude and phase of the z-component of the electric field at the desired line situated 1 μm before the waveguide is shown. Comparing these with

Figure 4.5 it is obvious that this approach has failed. This is due to the fact that we decreased the basewidth of the antennas too much to have enough antennas to characterize the variations in the amplitude and phase profile. Thus the fraction of light scattered in the forward direction was minute and the launched SPP beam also suffered from diffraction before reaching the stripe waveguide.

Since this approach did not yield the coupling into the stripe waveguides at the end of the gold film we decided to utilize a modified version a wavelength demultiplexer design to realize our objective.

Chapter 5

SCALAR WAVE THEORY

5.1 Antenna Arrays

To produce a desired radiation pattern many antennas may be placed together and collectively this structure is referred to as an array antenna. Sometimes it is also desirable to scan the radiation pattern through space and there are two means to achieve this; mechanically changing the positions of individual radiating elements or modulating the current fed to the individual radiating elements. The possibility to electronically scan the beam pattern through space is a huge advantage of an array antenna over a single large antenna. Furthermore, mechanical problems associated with supporting/moving a single large antenna are alleviated with such a design. If the pattern scanning is realized by modulating the currents the array is called a phased array. To illustrate the theory in this section we will focus on **linear arrays** and the discussion follows from the textbook by W.L. Stutzman and G.A. Thiele [44].

5.1.1 Linear Arrays

It is assumed that each scatterer in the array is an isotropic point source and the array antenna is reciprocal under transmission or reception. Mathematically the field of an isotropic source is directly proportional to:

$$E_0 \propto \frac{e^{i\beta r}}{4\pi r} \tag{5.1}$$

With isotropic source assumption in place the resulting radiation pattern is known as the **array factor**. In order to describe the radiation pattern of an array antenna fully we also need the **element patterns** which are the real radiation patterns of the constituent scatterer elements of the array antenna. The total radiation pattern then can be obtained by the multiplication of the element pattern with the array factor.

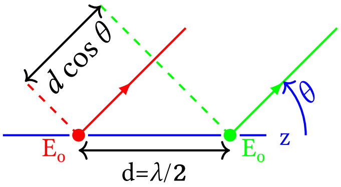


Figure 5.1: Two point sources of the same phase

For a demonstration of the calculation of the array factor a linear array consisting of two point sources will be used as illustrated in Figure 5.1. These sources have identical currents (equal in amplitude and phase), and they are spaced half of a wavelength apart. In order to calculate the radiation pattern a far field point that makes an angle of θ with the lines connecting the sources are selected. The path length difference between the two sources at this point is $d\cos(\theta)$. The array factor then follows:

$$\mathrm{AF} = 1e^{i\beta d\cos(\theta)} + 1 \tag{5.2}$$

Here d is the distance between the sources and equal to $\lambda/2$ and $\beta = 2\pi/\lambda$. Substituting these into Equation 5.2 and normalizing the array factor to have a maximum value of one yields,

$$\begin{aligned}\mathrm{AF} &= e^{i\beta(d/2)\cos(\theta)}(e^{i\beta(d/2)\cos(\theta)} + e^{-i\beta(d/2)\cos(\theta)}) \qquad (5.3)\\ &= 2e^{i\beta(d/2)\cos(\theta)}\cos(\beta(d/2)\cos(\theta)) \qquad (5.4)\\ &= \cos(\frac{\pi\cos(\theta)}{2}) \qquad (5.5)\end{aligned}$$

This array factor agrees well with intuition as it is expected to have zero amplitude on the line connecting the sources ($\theta = 0$), since they are driven by the same current and spaced half of a wavelength apart. The maximum amplitude of the pattern is realized at an angle of $\theta = \pi/2$ which is perpendicular to the line connecting the sources.

If the elemental antennas in an array antenna have the same radiation pattern and oriented in the same direction then the complete radiation pattern can be obtained by multiplication of the elemental pattern with the array factor.

5.1.2 Coupler Design

In this section a wavelength demultiplexer nanoslit array design which focuses different wavelength SPPs to predesignated different spatial positions will be discussed based on the 2011 paper by T. Tanemura et al [2].

In this design the slits have a length of 500 nm and width of 100 nm, and the incident plane wave is incident on the slit array from air side. The incident beam is polarized along the the short direction each slit (the orientation of each slit is the same). The excited SPPs propagate on the bottom metal-silica interface. At the design wavelength of 850 nm each slit act as a point dipole radiator. The reflections and scattering of SPPs by slits are not taken into account, since the reflection from thin slits is found to be negligible experimentally [45]. In this model each slit is regarded as a point dipole source of SPPs and the phase change of SPPs upon travelling through the slits is also minimal since the slit widths are smaller than half of the SPP wavelength [40]. It is possible to devise an iterative algorithm which relocates the slits to focus different wavelengths at different spots on a particular line.

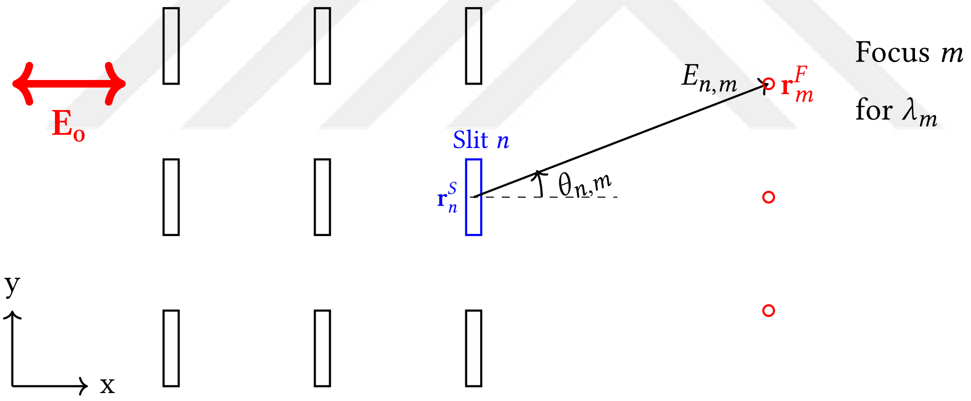


Figure 5.2: Labelling of slits and focal points

Employing the same notation as in the article by Tanemura et al. we have N slits whose positions are given by the position vector $\mathbf{r}^S_n$ where n runs over the slits. The focus points are denoted by $\mathbf{r}^{\mathrm{F}}_m$ where m runs over the wavelengths(λ_{m}) to be focused, for an illustration see Figure 5.2. For points far away from the slit -$|\mathbf{r}^{\mathrm{F}}_m - \mathbf{r}^{\mathrm{S}}_n| \gg \lambda_{SPP}$- the radiation is roughly proportional to $\cos(\theta_{n,m})/|\mathbf{r}^{\mathrm{F}}_m - \mathbf{r}^{\mathrm{S}}_n|$. It follows that the amplitude of the electric field at a focal position is complex and can be evaluated with the following sum:

$$E_m^{total} = \sum_n E_{n,m} \tag{5.6}$$

The individual $E_{n,m}$s represent the effect of n^{th} slit on the m^{th} focus point. These coefficients can be calculated as follows:

$$E_{n,m} = A_{n,m}T_{n,m}\frac{\cos(\theta_{n,m})}{\sqrt{|\mathbf{r}_m^{\mathrm{F}} - \mathbf{r}_n^{\mathrm{S}}|}}\exp[ik_m^{\mathrm{SPP}}(\mathbf{r}_m^{\mathrm{F}} - \mathbf{r}_n^{\mathrm{S}})] \tag{5.7}$$

where k_m^{SPP} is the wave vector of the SPP on the metal-silica interface, $A_{n,m}$ is the amplitude at the slit location and wavelength (hence two indices) and $T_{n,m}$ is the transmission coefficient. As aforementioned the transmission is not prominent so T is taken to be unity for all slits and wavelengths in our work in contrast with the original work which takes it to be $T_{n,m} = t_m^{p_n}$, where the lower case t_m represent the transmission over a single slit at normal angle at a given wavelength and p_n is the number of slit columns between the slit and the line on which the focal points lie. In our work $A_{n,m}$ represent the effect of having an input beam with Gaussian spatial dependence. The dielectric constants are taken from the experimental Johnson and Cristy data for gold [46].

In order to calculate the positions of the slits on the surface of the metal an iterative algorithm really similar to the one utilized by J. Backlund et al. for focusing light into any number of desired spots within a waveguide via incoupling waveguide holograms is used [47]. The algorithm functions as follows:

1. The total electric field E_m is calculated for each focus position $\mathbf{r}_m^{\mathrm{F}}$ at λ_m using Equation 5.7 and Equation 5.6.

2. The electromagnetic field scattered by each slit, that is $E_{n,m}$, is compared with the total electric field E_m^{total} for every focus point $\mathbf{r}_m^{\mathrm{F}}$.

3. The phase mismatch between electric field generated by a single slit and total electric field scattered by the slit array is expressed mathematically as:

$$\arg(E_m^{total}E_{n,m}^{*}/[|E_m^{total}||E_{n,m}|]) \tag{5.8}$$

   The objective is to minimize this difference, however there is the constraint that we focus several wavelengths to different positions at once.

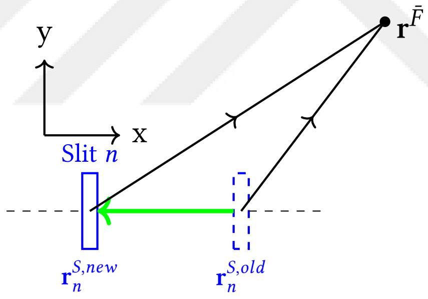
4. Therefore an average phase mismatch factor is calculated which is given by

$$\delta_n \equiv \arg\left[\sum_m W_m \frac{E_m^{total} E_{n,m}^*}{|E_m^{total}||E_{n,m}|}\right] \tag{5.9}$$

5. Then the slit n is relocated **only in the x direction** to decrease $|\delta_n|$ as shown in Figure 5.3. The new position of the slit is mathematically calculated from the following equation:

$$|\mathbf{r}_n^{\mathrm{S,new}} - \mathbf{r}^{\bar{\mathrm{F}}}| - |\mathbf{r}_n^{\mathrm{S,old}} - \mathbf{r}^{\bar{\mathrm{F}}}| = \frac{\delta_n}{\overline{k^{\mathrm{SPP}}}} \tag{5.10}$$

6. These steps are iterated until the desired performance is realized. In most cases 40 iterations are enough.



$$|\mathbf{r}_n^{S,new} - \mathbf{r}^{\bar{F}}| - |\mathbf{r}_n^{S,new} - \mathbf{r}^{\bar{F}}| = \delta_n/\overline{k^{\mathrm{SPP}}}$$

Figure 5.3: Dislocation of slit n

In the steps above $\overline{k^{\mathrm{SPP}}} =< k_m^{\mathrm{SPP}} >$, that is the wave vector of SPP averaged over all the desired operating wavelengths; $\mathbf{r}^{\bar{\mathrm{F}}} =< \mathbf{r}_m^{\mathrm{F}} >$ is the average position of the focal points. Here it is assumed that the change in the overall phase after the relocation of a single slit within an iteration step is negligible, so they are not taken into account. Furthermore, the final slit positions converge to the same value provided that the slits are moved in the correct direction in each iteration step. The W_m is present in order to obtain the correct slit pattern with balanced intensity on all the chosen focal points. Initially W_m is unity for every focal position(that is m) and it is updated in the subsequent iterations in compliance with the following expression:

$$W_m^{\text{new}} = W_m^{\text{old}} \left(\frac{\max(|E_m^{\text{total}}|^2)}{|E_m^{\text{total}}|} \right)^q \tag{5.11}$$

Here q must be small or else the algorithm is unstable [47]. In the article and my work q is selected to be 0.1. The slits are only moved only in the x-direction for simplicity. Since they act as point source dipoles moving them only in the y-direction should give similar results.

In order to cross check my implementation of the algorithm I have used the same design as the article. The input beam has a Gaussian spatial profile with a beam diameter of 17 µm.

	Focal Point 1	Focal Point 2	Focal Point 3
x(nm)	0	0	0
y(nm)	-3000	0	3000
λ(nm)	820	850	880

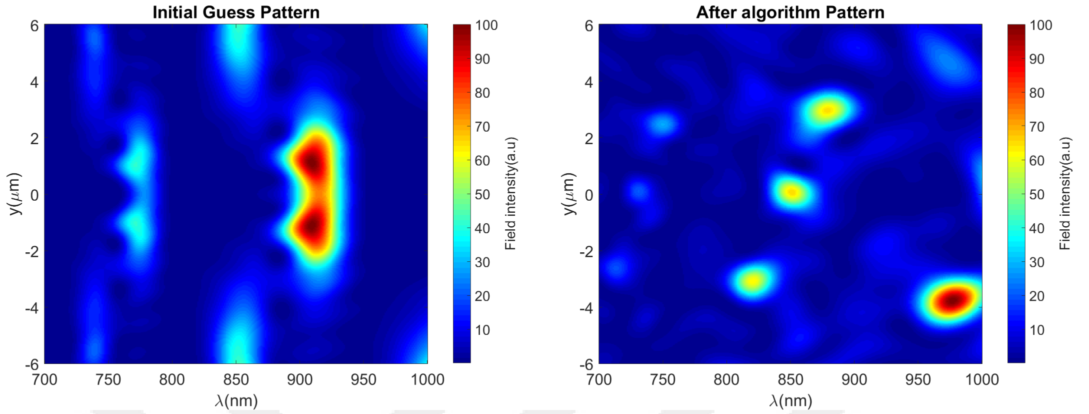
Table 5.1: Original focii parameters in the article

In Table 5.1 the positions of the focii and the corresponding wavelengths to be focused are given.

Beam diameter	17 µm
Au film thickness	75 nm
Periodicity in x	3 µm
Periodicity in y	1.5 µm
Number of columns	6
Number of rows	5
No of iterations	40
Slit width in x	120 nm
Slit length in y	500 nm

Table 5.2: Simulation parameters and initial conditions

In Table 5.2 the excitation source parameters, the slit width/height, the gold film thickness and the initial guess pattern (a periodic array) are given.

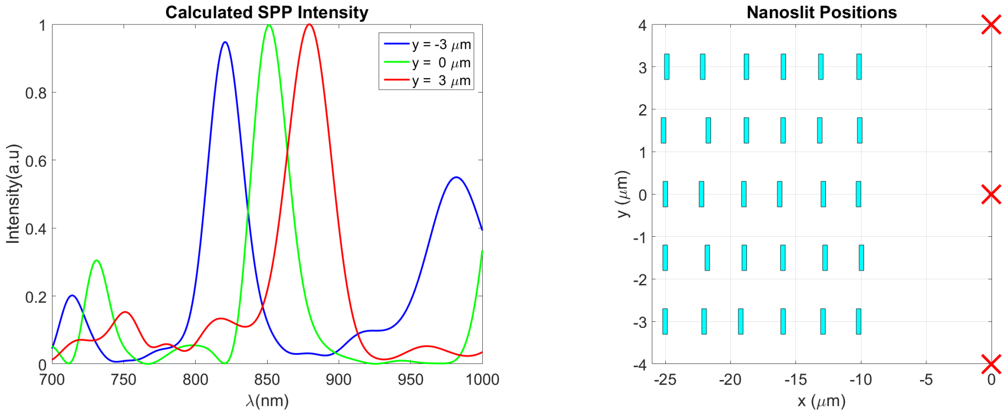


(a) The spectral distribution on the exit line for the initial guess slit pattern

(b) The spectral distribution on the exit line for the optimized slit pattern

Figure 5.4: Spectrum on $x = 0$ line with the initial guess and after optimization

In Figure 5.4a the initial spectral distribution of the output at the exit line located at $x = 0$ is given, whereas in Figure 5.4b the spectral distribution of the output on the same line is given after the optimization.

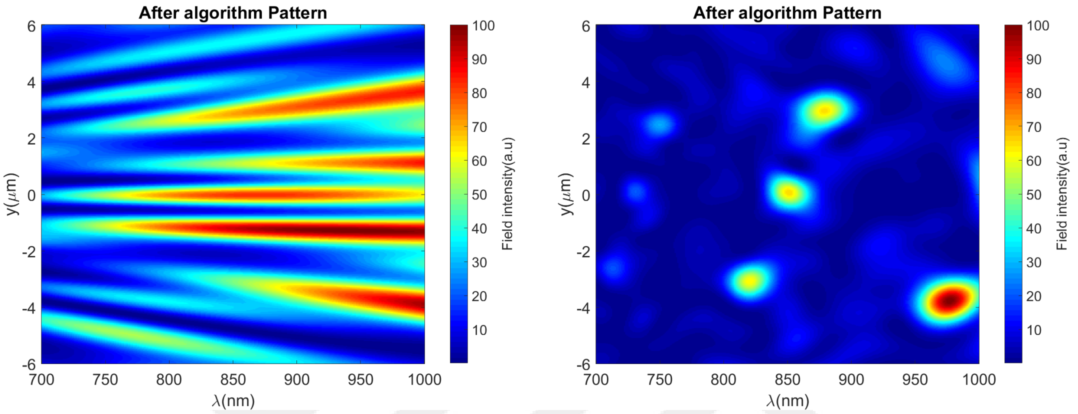


(a) The spectral distribution of the electric field norm on the individual focal points

(b) The slit pattern after 40 iterations of the optimization algorithm

Figure 5.5: Intensity and slit pattern while replicating the work in the article

In Figure 5.5a the spectral intensity pattern of the output light on the invidual focal points are given. In Figure 5.5b the locations of the slits after the optimization of the algoritm are given and the focii are marked with red crosses. The slits are positioned in such a way that the excited SPPs of the selected wavelengths interfere constructively on the focal points and this is essentially the same logic with antenna theory.



(a) Spectra by the last column of slits

(b) Spectra by all slits

Figure 5.6: Comparison of spectra

After optimization we wanted to isolate the effect of the rightest slit column in the final spectra on the $x = 0$ line. As you can see in Figure 5.6 the spectra on the exit line is produced by the all of the slits collectively. The slits in the closest (rightest) column are not entirely responsible for this spectra. In the next chapter we are going to generalize this approach to wavelengths in the near-infrared regime and add stripe waveguides after the focal points to investigate coupling.

Chapter 6

MODE COUPLING

Our aim is to couple incident light of different wavelengths into spatially separated stripe waveguides. In Chapter 4 we investigated coupling into stripe waveguides with SPP beam launchers. However, this approach failed due to the rapid variation of the amplitude and phase of the modal electric field for a stripe waveguide of width 1 µm. Therefore we propose an alternative design.

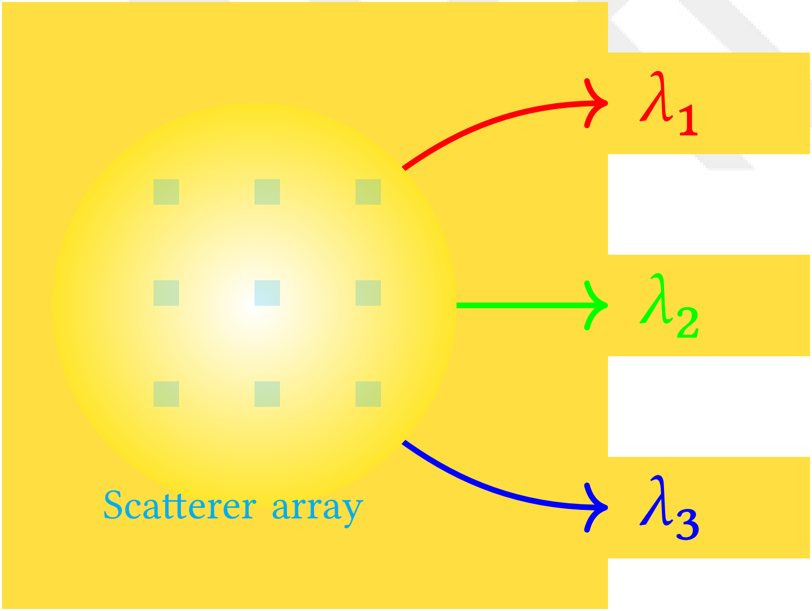
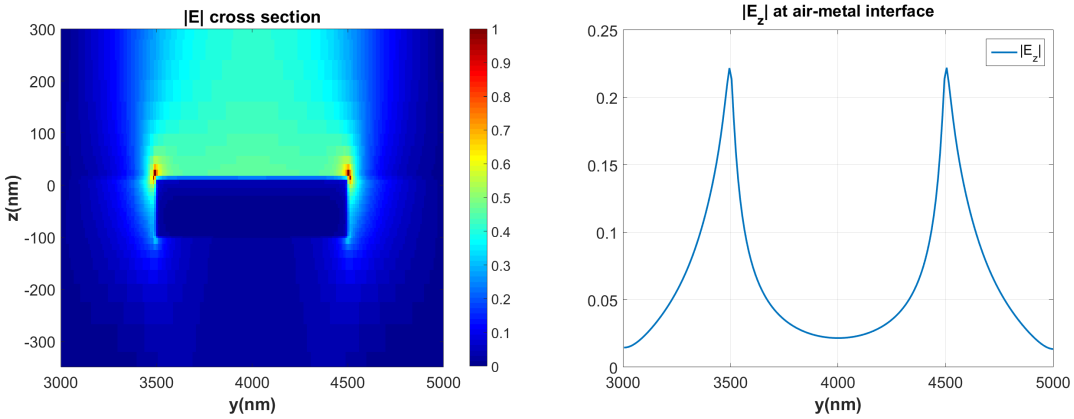


Figure 6.1: Designed coupler

The aim of the integrated device we design is to focus different wavelengths normally incident from the air side on the nanoslit/Δ-antenna pattern to different spatial locations and have them coupled to the stripe waveguides placed on the focal points as illustrated in Figure 6.1. In designing the scatterer-array we will make use of the algorithm presented in Chapter 5 and we will simulate these array designs in Lumerical FDTD Solutions to test their performance.



(a) Fundamental mode $|E|$ with Lumerical

(b) The $|E_z|$ field with Lumerical

Figure 6.2: Mode profiles of gold stripe waveguides at λ = 1550 nm

The stripe waveguides have a thickness of 115 nm and a width of 1000 nm. All of the three stripe waveguides are identical and the mode profiles for these waveguides are given in Figure 6.2.

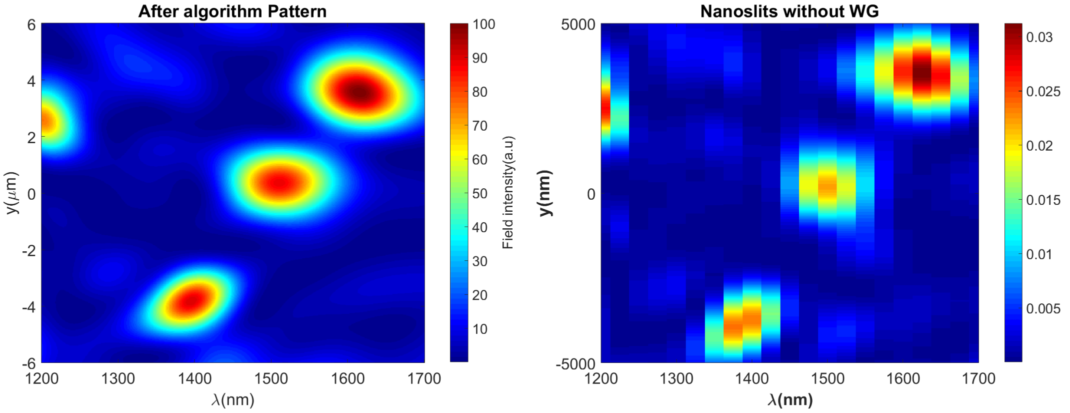
0.1 Nanoslits

First we deal with the nanoslits by selecting focal points in the NIR regime to conform with the NIR2 channel in our optical setup spanning a wavelength range from 1200 nm to 1700 nm.

	Focal Point 1	Focal Point 2	Focal Point 3
x(nm)	0	0	0
y(nm)	-4000	0	4000
λ(nm)	1400	1500	1600

Table 6.1: Focus parameters used for nanoslits

The position of the focal points and the focused wavelengths are given in Table 6.1. Each nanoslit has the dimension of 100 nm x 500 nm as they are resonant at 1550 nm (Figure 3.7). For the design with the waveguides and without the waveguides, the number of iterations is 40 and the used pattern is the same.

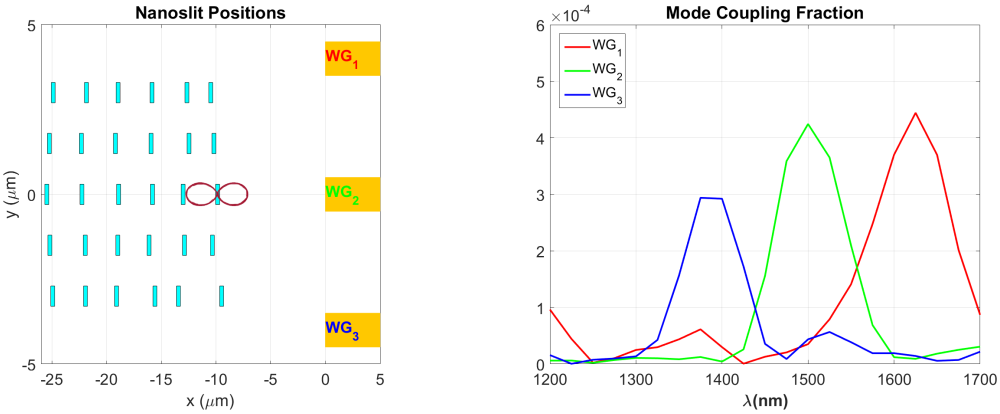


(a) Algorithm prediction

(b) Lumerical FDTD solutions results

Figure 6.3: Intensity on the $x = 0$ line for nanoslits

In Figure 6.3 it can be seen that the results from the algorithm and Lumerical FDTD Solutions agree well with each other for the case without a waveguide. The result from the FDTD solutions software has low spectral resolution in order to limit the file size of the dataset.



(a) Positions of nanoslits

(b) FDTD transmission results

Figure 6.4: Intensity on the $x = 0$ line for nanoslits

Figure 6.4 is associated with the case where three stripe waveguides of 1 μm as aforementioned. The slit positions are given in Figure 6.4a and the transmission is given as the fraction of the total power injected by the source in Figure 6.4b. Note that the same slit pattern is used for both cases (with/without waveguides).

The total power injected by the Gaussian source is calculated as follows:

$$P_{\text{source}}(f) = \frac{1}{2} \iint_{\text{IP}} \frac{\Re(\mathbf{E}(\omega) \times \mathbf{H}^{*}(\omega))}{|s(\omega)|^{2}} \cdot dS \tag{6.1}$$

$$s(\omega) = \int \exp(i\omega t) s(t) dt \tag{6.2}$$

Here $s(t)$ is the time signal of the source and IP is the injection-plane through which the source is injected. This normalization eliminates the dependence of the simulation results on the source spectrum and gives the impulse response of the system at various wavelengths. Transmission is then defined to be the ratio of the transmitted power through a particular cross-section (CS) to the total power injected by the source.

$$T = \frac{P_{\text{CS}}}{P_{\text{source}}} \tag{6.3}$$

$$P_{\text{CS}} = \frac{1}{2} \iint_{\text{CS}} \frac{\Re(\mathbf{E}(\omega) \times \mathbf{H}^{*}(\omega))}{|s(\omega)|^{2}} \cdot dS \tag{6.4}$$

	Focal Point 1	Focal Point 2	Focal Point 3
Transmission %	0.029	0.042	0.037
λ(nm)	1400	1500	1600

Table 6.2: Nanoslit array transmission percentage

These results may seem low since they indicate the percentage of light energy coupled into the waveguide as a fraction of the total light energy injected by the source as aforementioned. It should also be noted that the lowest wavelength peak occurs at 1375 nm instead of 1400 nm and the highest wavelength peak occurs at 1575 nm instead of 1600 nm.

0.2 Δ-antennas

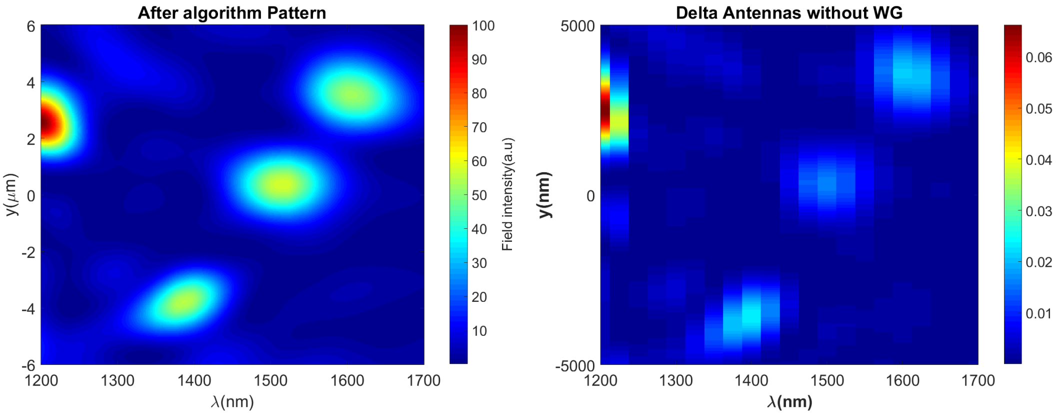
	Focal Point 1	Focal Point 2	Focal Point 3
x(nm)	0	0	0
y(nm)	-4000	0	4000
λ(nm)	1400	1500	1600

Table 6.3: Focus parameters used for Δ-antennas

With the Δ-antennas we select the same focal points as in the design for the nanoslits to be able to directly compare the results, these are given in Table 6.3. The number of iterations of the algorithm is 80 in this case; however later we found that there is not much difference between 40 iterations and 80 iterations. In this section we will first present a non-optimal Δ-antenna design, then proceed to the optimal Δ-antenna design to emphasize the contrast. Finally we present the optimal Δ-antenna design with the light incident from the substrate (silica) side.

6.2.1 Design with w_b = 500 nm, h = 800 nm

We first chose a Δ-antenna with a basewidth of 500 nm and a height of 800 nm which is not optimal according to analysis in Chapter 3.

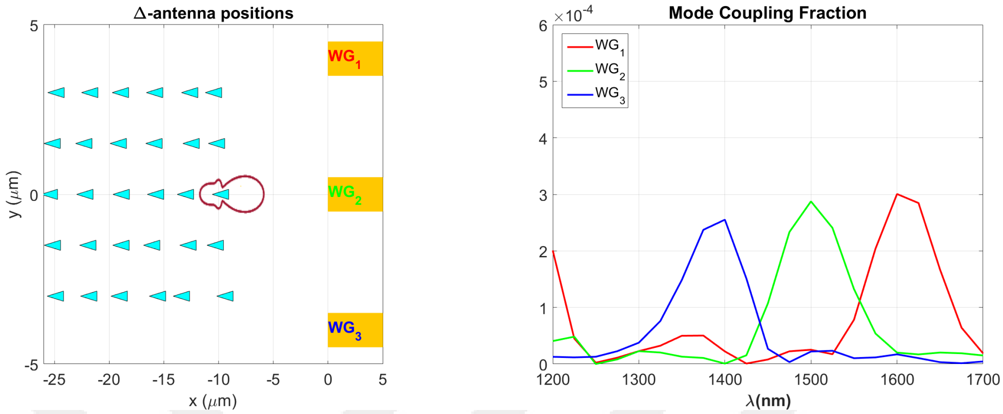


(a) Algorithm prediction

(b) Lumerical FDTD solutions results

Figure 6.5: Δ-antenna intensity pattern on x = 0 line

The resulting spectral pattern on the $x = 0$ line after the FDTD simulation (Figure 6.5b) have good correspondence with the prediction by the algorithm (Figure 6.5a).



(a) Positions of Δ-antennas

(b) FDTD transmission results

Figure 6.6: The Δ-antenna pattern and the intensity profile in the waveguides

The same Δ-antenna pattern is also used for the case in which the gold film is terminated with three 1 μm wide gold stripe waveguides, which is illustrated in Figure 6.1. The Δ-antenna pattern from the algorithm is given in Figure 6.6a and the intensity of the light coupled into the waveguides is given in Figure 6.6b.

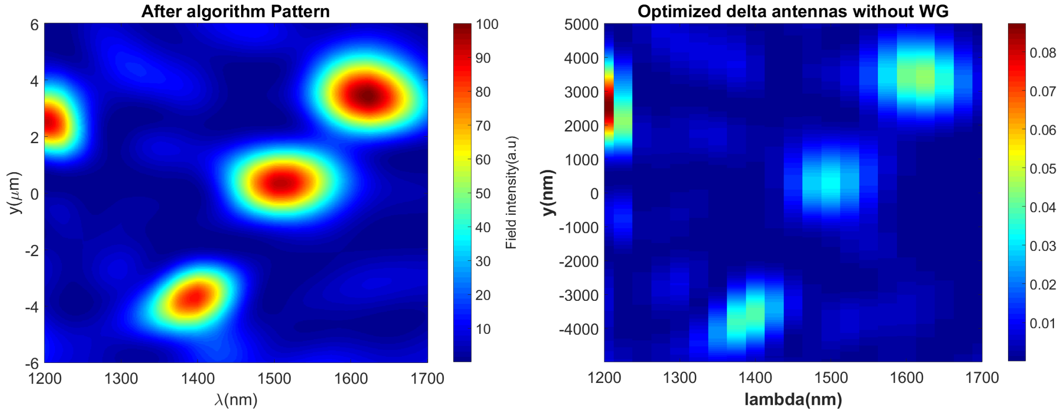
	Focal Point 1	Focal Point 2	Focal Point 3
Transmission %	0.025	0.029	0.030
λ(nm)	1400	1500	1600

Table 6.4: Δ-antenna(w_b = 500 nm, h = 800 nm) array transmission percentage

Table 6.4 summarizes the transmission percentages for the focal points. Comparing this result with those from the nanoslits we arrive at the conclusion that the peaks are in the correct positions, and the intensity distribution is more uniform.

6.2.2 *Design with w_b = 900 nm, h = 800 nm*

This is the optimal Δ-antenna design according to our analysis in Chapter 3, please refer to Figure 3.13. We will see that this design gives the maximum performance.

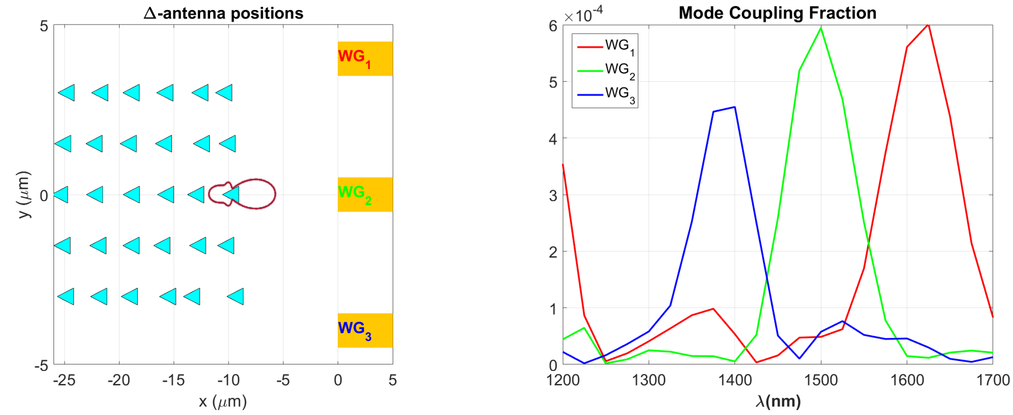


(a) Algorithm prediction

(b) Lumerical FDTD solutions results

Figure 6.7: Δ-antenna intensity pattern on $x = 0$ line

After the FDTD simulation the resulting spectral pattern on the $x = 0$ line have good correspondence with the prediction by the algorithm as shown in Figure 6.7 and comparing these results with Figure 6.5 we arrive at the conclusion that this design has better focusing properties thanks to the enhanced forward scattering cross-section of the Δ-antennas.



(a) Positions of Δ-antennas

(b) FDTD transmission results

Figure 6.8: The Δ-antenna pattern and the intensity profile in the waveguides

In Figure 6.8a the positions of the Δ-antennas are shown. Note that this pattern was used for both structures, with or without waveguides. The transmission percentages to the corresponding stripe waveguide modes are given in Figure 6.8b.

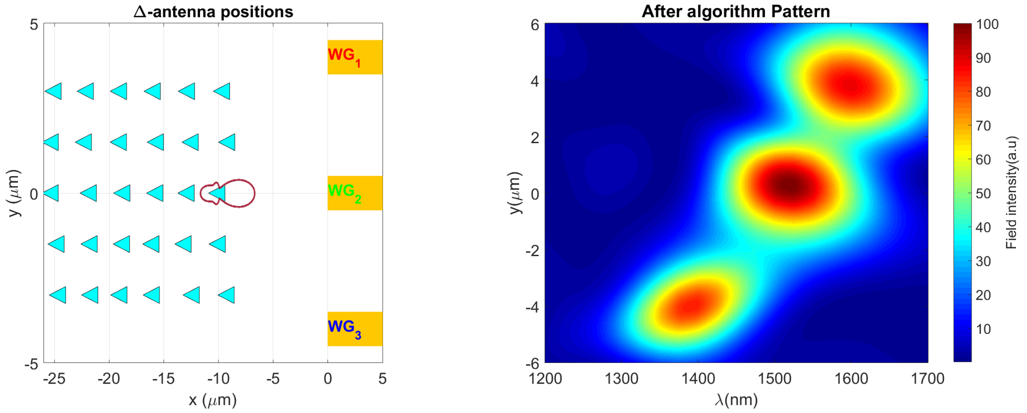
	Focal Point 1	Focal Point 2	Focal Point 3
Transmission %	0.045	0.059	0.056
λ(nm)	1400	1500	1600

Table 6.5: Δ-antenna (w_b = 900 nm, h = 800 nm) array transmission percentage

The distribution is non-uniform as the lowest wavelength peak is smaller compared to the higher wavelength peaks.The highest wavelength peak occurs at 1625 nm instead of 1600 nm. The intensities are almost double of the values observed with the nanoslit and non-optimized Δ-antenna design.

6.2.3 Design with w_b = 900 nm, h = 800 nm with SPP on air-metal

Here we wanted to see how the optimized Δ-antenna design fares when the structure is illuminated from the substrate (silica) side. This time an FDTD simulation without waveguides could not be performed due to time/computing constraints.

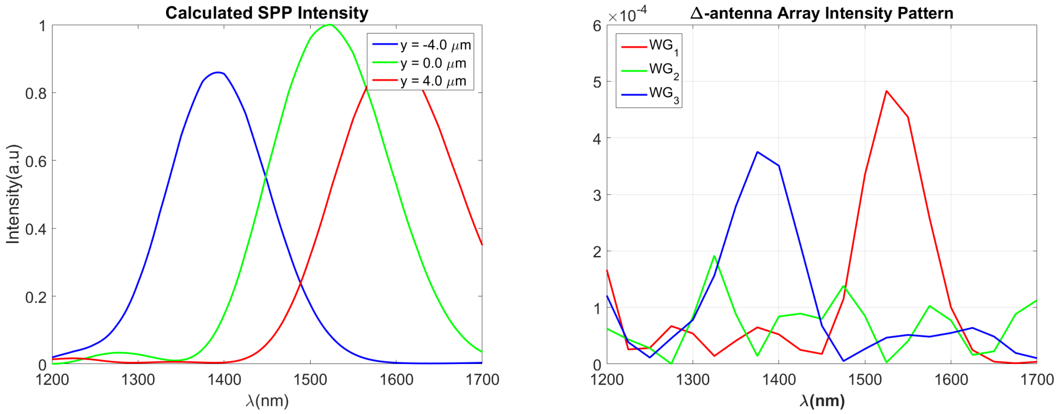


(a) Positions of Δ-antennas

(b) FDTD spectral map on x = 0 line

Figure 6.9: The Δ-antenna pattern and the intensity profile in the waveguides

In Figure 6.9a the positions of the Δ-antennas are show. This pattern is used in the simulation with waveguides as before. Figure 6.9b show the prediction from the algorithm on x = 0 line. Note that this time peaks are more spread over in both spectral and spatial domains.



(a) Algorithm prediction

(b) Lumerical FDTD solutions results with WGs

Figure 6.10: Δ-antenna intensity pattern on $x = 0$ line

The prediction by the algorithm on the $x = 0$ line for the three wavelengths are given in Figure 6.10a. After inspecting Figure 6.10b which corresponds to the result from Lumerical FDTD Solutions for the case with waveguides, we see that the correspondance between simulation results and the prediction by the algorithm was not good this time for the middle wavelength of 1500 nm.

	Focal Point 1	Focal Point 2	Focal Point 3
Transmission %	0.035	0.009	0.009
λ(nm)	1400	1500	1600

Table 6.6: Δ-antenna(w_b = 900 nm, h = 800 nm) array transmission percentage with SPP on air-metal

This time the result was pretty bad for the middle peak and the other peaks were also shifted to lower wavelengths. Furthermore, the peaks were wide compared to the previous designs with Δ-antennas and nanoslits which were illuminated from the air side.

Chapter 7

CONCLUSION AND FUTURE WORK

In this thesis an approach integrating plasmonic wavelength division demultiplexer with plasmonic stripe waveguides was presented. Wavelength division multiplexing (WDM) refers to the process of combining, transmitting and separating of optical signals of different wavelengths. The main purpose in WDM is to enable the transmission of data in several channels; thus increasing the bandwidth available [48]. In optical communication WDM is used for propagating data over transoceanic distances together with the advent of erbium-doped fiber amplifiers (EDFA) [49] and it is also widely used since 1999 for terrestial networks with ever increasing number of channels [50]. Previously various designs for plasmonic demultiplexers were studied, but none of them examined the coupling of the demultiplexed light into plasmonic stripe waveguides.

With this aim in mind we started by characterizing single scatterers. First we selected Δ-antennas of the same geometry as in the article by Oubo You [14] and investigated its radiation pattern on a gold film with the same thickness and dielectric permittivity as in the paper. The results from our FDTD simulations corresponded really well with the results from the original paper, confirming our approach and simulation technique. We used this approach to characterize rectangular nanoslits which are the other type of scatterers we used and Δ-antennas optimized to 1550 nm free-space wavelength. Furthermore after replicating the SPP beam launcher in the paper, we explored the possibility of realizing mode-coupling by designing a beam launcher as described in this paper to produce an SPP beam with the same profile as a 1 μm wide stripe waveguide mode. However, this approach failed due to the fine features in the mode-profile as discussed in Chapter 4. Therefore we decided to use a different approach.

We decided to use the wavelength demultiplexer design proposed by Tanemura [2] consisting of an array of nanoslits whose scattering pattern is adjusted with an iterative algorithm in such a way that SPPs of selected wavelengths interfere constructively on the selected focal points on the exit line. In this algorithm the nanoslits are only moved in the x direction for an easy implementation. These rectangular nanoslits radiated most in the direction perpendicular to their long sides, yet most radiation in Δ antennas were directed perpendicular to the base of the triangle to the opposite site of the vertex. This is valid for antennas having an height of 505 nm and a width of 300 nm turned out to radiate unidirectionally for a free-space wavelength of 633 nm and for the antennas operating at a free-space wavelength of 1550 nm the corresponding height and basewidth value are 1360 nm and 950 nm, respectively. The dimension of the nanoslit and Δ-antenna were selected to have the highest forward scattering cross-section to couple the most light energy possible to the focal spots. In this step we also performed a fit for the radiation pattern to be able to use different scatterers in the same algorithm. The fits were performed on a circle with radius 2 μm centered on the scatterer and checked on a circle with the same center but a radius of 4 μm. The fits were good, yet not in exact agreement with the interpolated simulation data on the circle possibly due to the excitation of waves other than SPPs.

We began our multiplexer design with a triplexer as in the study by Takuo Tanemura et al. to validate our implementation of the algorithm [2]. The location of focal points and the associated wavelengths were the same in both cases. There was good correspondence between the results by our implementation of the algorithm and the results of the original algorithm. After having confirmed our implementation of the algorithm we did an FDTD simulation with the slits in the positions predicted by our algorithm and the results were similar to the prediction of our implementation.

After these verifications we proceed with a design in the infrared and selected to focus SPPs having the wavelength of 1400 nm, 1500 nm and 1600 nm at the y position of -4 μm, 0 μm and 4 μm. We chose these wavelengths to be compatible with our optical setup. First we performed two simulations for an array of nanoslits, one with waveguides and one without waveguides. There were good correspondence between algorithm results and FDTD results. However the peaks were slightly off their intended positions. The transmission percentages were of the order 10^{-4} of the incident light energy. When we performed two other simulation for Δ-antennas (w_b = 500nm, $h = 800nm$) we found out that the transmissions values were similar yet the middle peak were not well defined. Then we performed simulation for optimized Δ-antennas (w_b = 900nm, $h = 800nm$) there were again good correspondence between algorithm predictions and FDTD results. This scatterer yielded the best transmission results which corresponds to 10^{-3} of the incident light energy which is two-folds of the previous result. Lastly we tried illumination from the substrate (silica) side, yet the peaks were spread over too much and the focusing of middle wavelength was pretty poor as shown at the end of Chapter 6.

In the future the wavelength demultiplexer design consisting of Δ-antennas with w_b = 900 nm, h = 800 nm presented in Chapter 6 can be manufactured with conventional lithography techniques along with plasmonic waveguides and integrated photo sensors in order to quantitatively measure the light intensity and compare it with the FDTD simulation results presented in this article. The device performance may be improved by designing better transition regions to waveguides in order to minimize the reflection at the gold film-waveguide boundary. Furthermore, different scatterer geometries and algorithms may also be investigated to see if they yield higher performance.

Appendices

Appendix A

SUN GRID ENGINE SCRIPTS

This appendix includes the job submission scripts for the high performance computing clusters in our facility, namely Yunus and Lufer which runs on Sun Grid Engine 6.2u5. The simulations were performed with Lumerical FDTD Solutions version 8.15.697 and the connections to the HPC server are established via MobaXterm on Windows or ssh via command prompt on Linux.

```
#!/bin/bash
#$ -N Lumerical
#$ -S /bin/bash
#$ -q all.q
#$ -pe mpich 5
#$ -cwd
#$ -o Lumericalstdout.log
#$ -e Lumericalstderr.log
#$ -l mem_free=34G
#$ -M oarisev14@ku.edu.tr
#$ -m bea

export PATH=/share/apps/lumerical/fdtd-8.15.697/bin:$PATH
export LD_LIBRARY_PATH=/usr/lib64:/usr/lib:$LD_LIBRARY_PATH

PATH=/share/apps/libxml2/libxml2-2.9.2/bin:$PATH ; export PATH
LD_LIBRARY_PATH=/share/apps/libxml2/libxml2-2.9.2/lib:$LD_LIBRARY_PATH ; export LD_LIBRARY_PATH

sh /share/apps/lumerical/fdtd-8.15.697/bin/fdtd-run-local.sh -n 5 *.fsp
```

The explanation of the lines above are given in the list on the next page.

1. Name of the submitted job(other users also see it)
2. Path to the shell to be used
3. Which queue the job goes(all.q or short.q in our HPC)
4. Name of the parallel environment(smp or mpich for FDTD Solutions)
5. Have output placed in the current working directory(cwd)
6. Standard output(can use bash variables such as $JOB_ID)
7. Standard error (can use bash variables such as $JOB_ID)
8. Hard resource requirement(34 GB of RAM in this case)
9. Address to mail to
10. Mail under which conditions(**b**-begin, **e**-end, **a**-abort)

On lines 5 and 19 the **number of processors** used is written and they must be the same. Another example job script using the **smp** parallel environment, short queue that is allocated for jobs having a run time lower than 2 hours(short.q) and no hard resource limit is given on the next page. For a tutorial on how to use the Sun Grid Engine one can refer to the following URLs:

- Koç University Tutorial with sample files
- NJIT- Quick SGE
- More in depth guide

```
#!/bin/bash
#$ -N Lumerical
#$ -S /bin/bash
#$ -q short.q
#$ -pe smp 10
#$ -cwd
#$ -o $JOB_ID.out
#$ -e $JOB_ID.err
#$ -M oarisev14@ku.edu.tr
#$ -m bea

export PATH=/share/apps/lumerical/fdtd-8.15.697/bin:$PATH;
export LD_LIBRARY_PATH=/usr/lib64:/usr/lib:$LD_LIBRARY_PATH;

PATH=/share/apps/libxml2/libxml2-2.9.2/bin:$PATH ; export PATH;
LD_LIBRARY_PATH=/share/apps/libxml2/libxml2-2.9.2/lib:$LD_LIBRARY_PATH ; export LD_LIBRARY_PATH;

sh /share/apps/lumerical/fdtd-8.15.697/bin/fdtd-run-local.sh -n 10 *.fsp;
```

Appendix B

MATLAB CODES

Most of the calculations, implementation of the algorithms, plotting and processing figures are done in Matlab. Here we are going to present the most important codes. We start with the codes used for characterization of single scatterers.

B.1 Characterization of single scatterer

```
clear;
close all;
%% Choose the radius of the circle
R = 2000;
dt = 1/100;
theta = (0:dt:1) * 2 * pi;
% Adjust the phase to rotate the image
ROT = -pi / 2;
% Get the x and y coordinates of the circle
XI = R * cos(theta - ROT);
YI = R * sin(theta - ROT);
N = 13; % Number of data sets
wb = 240:10:360; % The changing base width
idstr = 'StandardSubst';
%% Load the files
for i = 1:N
    load(sprintf('../Data/ScatterFINEsubst%d.mat', N+1-i), 'Eabov*');
    ELUM{i} = Eabove;
end
Lsize = length(Eabove.lambda);
% Loop over wavelength
for f_ind = 1:Lsize
    % Loop over all the files with differing base widths
    for i = 1:N
        %     f_ind = 1;
        Eabove = ELUM{i};
        if i == 1
            X = round(Eabove.x * 1e9);
            Y = round(Eabove.y * 1e9);
            [XX,YY] = ndgrid(X,Y);
        end
        % Check the dimension of the electric field
        s = size(Eabove.E); l = length(s);
        if l == 2
            E{i}.Ex = flipud(reshape(Eabove.E(:, 1), length(X), length(Y)));
            E{i}.Ey = flipud(reshape(Eabove.E(:, 2), length(X), length(Y)));
            E{i}.Ez = flipud(reshape(Eabove.E(:, 3), length(X), length(Y)));
            lambda = round(Eabove.lambda * 1e9);
        elseif l == 3
            % f_ind corresponds to the frequency index
            E{i}.Ex = flipud(reshape(Eabove.E(:, 1, f_ind), length(X), length(Y)));
            E{i}.Ey = flipud(reshape(Eabove.E(:, 2, f_ind), length(X), length(Y)));
            E{i}.Ez = flipud(reshape(Eabove.E(:, 3, f_ind), length(X), length(Y)));
            lambda = round(Eabove.lambda(f_ind) * 1e9);
        end
        % Derived variables
        E{i}.Norm = sqrt(E{i}.Ex .* conj(E{i}.Ex) + E{i}.Ey .* conj(E{i}.Ey) + E{i}.Ez .* conj(E{i}.Ez));
        Enorm = E{i}.Norm;
        E{i}.X = round(Eabove.x * 1e9);
        E{i}.Y = round(Eabove.y * 1e9);
        % Add pi since we polar plot does not support negative r values
        E{i}.EzPhase = angle(E{i}.Ez) + 2*pi;
        EzPhase = E{i}.EzPhase;

        %% Form gridded interpolants to interpolate the data on the circle
        EnormInt = griddedInterpolant(XX, YY, Enorm);
        EnormCirc = EnormInt(XI, YI);
        EzPhaseInt = griddedInterpolant(XX, YY, EzPhase);
        EzPhaseCirc= EzPhaseInt(XI, YI);
        E{i}.NormCircle = EnormCirc;
        E{i}.EzPhaseCircle = EzPhaseCirc;
        Norm{i} = EnormCirc;
        Phase{i} = EzPhaseCirc;
        %% For testing purposes
        %     figure;
        %     imagesc(X, Y, Enorm);
        if i == 1;
            f1 = figure;
        end
        p1(i) = polar(theta, EzPhaseCirc);
        p1(i).LineWidth = 2;
        if i == 1
            hold on;
        end
    end
    l = legend('360', '350', '340', '330', '320', '310', '300', '290', '280', '270', '260', '250', '240', 'Location', 'northeastoutside');
    l.FontSize = 18;
    tstring = ['Phase of E_z for varying base widths at \lambda = ' int2str(lambda) 'nm']; title(tstring);
%     saveas(f1, ['./Plot/EPhaseL', idstr, int2str(lambda), '.fig']);
    polarticks(tstring, 4, p1);
    th = findall(gcf,'Type','text');
    for i = 1:length(th),
      set(th(i),'FontSize',18)
    end
    f1.PaperUnits = 'points';
    f1.PaperPosition = [0 0 640 480];
    print(['./Plot/EPhaseL', idstr, int2str(lambda), '.png'], '-dpng', '-r144');

    for i = 1:N
        if i == 1
            f2 = figure;
        end
        p2(i) = polar(theta, 360 / wb(i) * E{i}.NormCircle.^2);
        p2(i).LineWidth = 2;
        if i == 1
            hold on;
        end
    end
    l = legend('360', '350', '340', '330', '320', '310', '300', '290', '280', '270', '260', '250', '240', 'Location', 'northeastoutside');
    l.FontSize = 18;
    tstring = ['|E|^2 for varying base widths at \lambda = ', int2str(lambda) 'nm']; title(tstring);
%     saveas(f2, ['./Plot/E2L', idstr, int2str(lambda), '.fig']);
    polarticks(tstring, 4, p2);
    th = findall(gcf,'Type','text');
    for i = 1:length(th),
        set(th(i),'FontSize',18)
    end
    f2.PaperUnits = 'points';
    f2.PaperPosition = [0 0 640 480];
    print(['./Plot/E2L', idstr, int2str(lambda), '.png'], '-dpng', '-r144');
    if l == 2
        break;
    end
end
```

The code is essentially the same for the rectangular nanoslit scatters and it can easily be extended to accommodate wavelength dependence. The polarticks function by Adam Danz is necessary to obtain nice polar plots since built-in functionality for polar plots in Matlab prior to R2016a is really limited.

B.2 Fitting Script

The fits are done with the following script for the Δ-antennas, the script for the rectangular nanoslits is similar.

```
clear;
close all;
%% The excitation is from silica side
load('../Data/DeltaAntenna2AirTF.mat');
%% Pay attention to  where SPPs should propagate
load('../Data/kSPP1200-1700JCsilica');
ksppNERF = kspp(1:25:end);
id = 'Delta2SingleSPPsilica';
% Get Lumerical data in units of nanometers
X = round(Esurface.x * 1e9);
Y = round(Esurface.y * 1e9);
R1 = 2000; % in nanometers
R2 = 4000; % in nanometers
dt = 1/100; % the increment fraction in the angle
theta = (0:dt:1) * 2 * pi; % Angular sweep
[XX,YY] = ndgrid(X, Y); % X and Y as meshgrid
ROT = 0; % We may need to rotate the coordinates
%% Circle X and Y
XI = R1 * cos(theta + ROT);
YI = R1 * sin(theta + ROT);
XII = R2 * cos(theta + ROT);
YII = R2 * sin(theta + ROT);
Larr = round(Esurface.lambda * 1e9); % wavelength array in nanometers
%% Reshaping of Lumerical datasets
for i = 1:numel(Larr)
    Esurf{i}.Ex = (reshape(Esurface.E(:, 1, i), length(X), length(Y)));
    Esurf{i}.Ey = (reshape(Esurface.E(:, 2, i), length(X), length(Y)));
    Esurf{i}.Ez = (reshape(Esurface.E(:, 3, i), length(X), length(Y)));
    Esurf{i}.Norm = sqrt(Esurf{i}.Ex .* conj(Esurf{i}.Ex) + Esurf{i}.Ey .* conj(Esurf{i}.Ey) + Esurf{i}.Ez .* conj(Esurf{i}.Ez));
    % We take the z component of the electric field
    EzSurfInt = griddedInterpolant(XX, YY, Esurf{i}.Ez); % interpolation
    EzSurfIntArr{i} = EzSurfInt;
    EzSurfCirc1{i} = EzSurfInt(XI, YI);
    EzSurfCirc2{i} = EzSurfInt(XII, YII);
    EzSurfCircInt{i} = griddedInterpolant(theta, EzSurfCirc1{i});
end
f1 = figure(1);
k = 1;
indfor = 1:2:numel(Larr);
indfor = 15;
for i = indfor
```

```
    plt = real(EzSurfCirc1{i});
    p = plot(theta,plt./max(plt));
    if k == 1; hold on; end
    legendinfo{k} = [num2str(Larr(i)) ' nm'];
    k = k + 1;
end
xlabel('Angle(rad)');
ylabel('Intensity normalized(a.u)');
title([id ' Re(E) along a circle of radius R=', num2str(R1), ' nm'])
legend(legendinfo);
f1.PaperUnits = 'points';
f1.PaperPosition = [0 0 640 480];
ax.FontSize = 18;
ax.XTick = [-2*pi -pi 0 pi 2*pi];
ax.XTickLabel = {'-2\pi','-\pi','0','\pi','2\pi'};
fname = [id 'Real' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f1, ['./Plot/' fname]);

f2 = figure(2);
k = 1;
for i = indfor
    plt = imag(EzSurfCirc1{i});
    p = plot(theta,plt./max(plt));
    if k == 1; hold on; end
    legendinfo{k} = [num2str(Larr(i)) ' nm'];
    k = k + 1;
end
xlabel('Angle(rad)');
ylabel('Intensity normalized(a.u)');
title([id ' Im(E) along a circle of radius R=', num2str(R1), ' nm'])
legend(legendinfo);
f2.PaperUnits = 'points';
f2.PaperPosition = [0 0 640 480];
ax.FontSize = 18;
ax.XTick = [-2*pi -pi 0 pi 2*pi];
ax.XTickLabel = {'-2\pi','-\pi','0','\pi','2\pi'};
fname = [id 'Imag' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f2, ['./Plot/' fname]);

f3 = figure(3);
k = 1;
for i = indfor
    plt = abs(EzSurfCirc1{i});
    p = plot(theta,plt./max(plt));
    legendinfo{k} = [num2str(Larr(i)) ' nm'];
    k = k + 1;
```

```
end
xlabel('Angle(rad)');
ylabel('Intensity normalized(a.u)');
title([id ' |E| along a circle of radius R=', num2str(R1), ' nm'])
legend(legendinfo);
f3.PaperUnits = 'points';
f3.PaperPosition = [0 0 640 480];
ax.FontSize = 18;
ax.XTick = [-2*pi -pi 0 pi 2*pi];
ax.XTickLabel = {'-2\pi','-\pi','0','\pi','2\pi'};
fname = [id 'Abs' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f3, ['./Plot/' fname]);

f4 = figure(4);
k = 1;
for i = indfor
    plt = angle(EzSurfCirc1{i});
    plot(theta,plt./max(1));
    if k == 1; hold on; end
    legendinfo{k} = [num2str(Larr(i)) ' nm'];
    k = k + 1;
end
xlabel('Angle(rad)');
ylabel('Phase');
title([id ' phase(E_z)  a circle of radius R=', num2str(R1), ' nm'])
legend(legendinfo);
f4.PaperUnits = 'points';
f4.PaperPosition = [0 0 640 480];
ax.FontSize = 18;
ax.XTick = [-2*pi -pi 0 pi 2*pi];
ax.XTickLabel = {'-2\pi','-\pi','0','\pi','2\pi'};
fname = [id 'Phase' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f4, ['./Plot/' fname]);


for i = 1:numel(Larr)
    fieldR2{i} = EzSurfCirc1{i} * exp(1i * ksppNERF(i) * (R2 - R1) * 1e-9) * sqrt(R1 / R2);
end

p1 = fieldR2{indfor};
p2 = EzSurfCirc2{indfor};

f5 = figure(5);
plot(theta, real(p1), 'linewidth', 2);
hold on;
plot(theta, real(p2), 'linewidth', 2);
```

```
title('Real parts');
f5.PaperUnits = 'points';
f5.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
ax.XTick = [-2*pi -pi 0 pi 2*pi];
ax.XTickLabel = {'-2\pi','-\pi','0','\pi','2\pi'};
legend('Fit', 'Interpolation');
fname = [id 'FitCompareReal' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f5, ['./Plot/' fname]);

f6 = figure(6);
plot(theta, imag(p1), 'linewidth', 2);
hold on;
plot(theta, imag(p2), 'linewidth', 2);
title('Imaginary parts');
f6.PaperUnits = 'points';
f6.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
ax.XTick = [-2*pi -pi 0 pi 2*pi];
ax.XTickLabel = {'-2\pi','-\pi','0','\pi','2\pi'};
legend('Fit', 'Interpolation');
fname = [id 'FitCompareImag' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f6, ['./Plot/' fname]);

f7 = figure(7);
plot(theta, abs(p1), 'linewidth', 2);
hold on;
plot(theta, abs(p2), 'linewidth', 2);
title('Magnitude');
f7.PaperUnits = 'points';
f7.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
ax.XTick = [-2*pi -pi 0 pi 2*pi];
ax.XTickLabel = {'-2\pi','-\pi','0','\pi','2\pi'};
legend('Fit', 'Interpolation');
fname = [id 'FitCompareAbs' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f7, ['./Plot/' fname]);

cst = find( X == min(abs(X)));
f8 = figure(8);
subplot(1,2,1);
imagesc(X,Y, real(Esurf{indfor}.Ez).', [-1e-1 1e-1]);
```

```
xlabel('x(nm)');
ylabel('y(nm)');
set(gca, 'YDir','normal');
colorbar; colormap jet;
subplot(1,2,2);
plot(X,real(Esurf{indfor}.Ez(cst,:))); % .*sqrt(abs(X.'))
xlabel('y(nm)');
xlim([-5000 5000]);
fname = [id 'PatternMiddle' num2str(R1)];
print(['./Plot/' fname], '-dpng', '-r144');
%saveas(f8, ['./Plot/' fname]);

EzSurfCircDelta = EzSurfCircInt;
% save('../Data/CircleFitDeltaAir.mat', 'EzSurfCirc*');
```

B.3 Scattering Analysis Script

The scattering cross-section normalized to the geometric cross-section of the scatterers are calculated to decide what the height, width of the scatterer should be. In the code quoted below Δ-antennas having a base width of 900 nm and various heights are characterized; forward scattering in the y_2 direction is of concern here.

```
clear;
close all;
load('../Data/Delta2HeightSweepData.mat');
Q = 1;
% The minus signs are there to revert the surface normal
% The scattering cross section is given in m^2 please look at Lumerical
% Script files for details
baseSI = 900e-9;
heightSI = height;
areaSI = repmat(baseSI .* heightSI / 2, 1, numel(LambdaArr));
if Q == 1
    y2_2 = y2_2 ./ areaSI;
    y1_2 = y1_2 ./ areaSI;
    x2_2 = x2_2 ./ areaSI;
    x1_2 = x1_2 ./ areaSI;
    z2_2 = z2_2 ./ areaSI;
    z1_2 = z1_2 ./ areaSI;
    Total = Total ./ areaSI;
    Total_2 = Total_2 ./ areaSI;
end
f1 = figure;
imagesc(LambdaArr*1e9, base*1e9,y2_2);
```

```
colormap jet;
colorbar;
title('\Delta Forward Scattering(y2 normal) (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f1.PaperUnits = 'points';
f1.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f1, './Plot/DeltaHForwardScattering.png', '-dpng', '-r144');
% saveas(f1, './Plot/DeltaHForwardScattering.fig');

f2 = figure;
imagesc(LambdaArr*1e9, base*1e9, -y1_2);
colormap jet;
colorbar;
title('\Delta Backward Scattering(y1 normal) (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f2.PaperUnits = 'points';
f2.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f2, './Plot/DeltaHBackwardScattering.png', '-dpng', '-r144');
% saveas(f2, './Plot/DeltaHBackwardScattering.fig');

f3 = figure;
imagesc(LambdaArr*1e9, base*1e9,-x1_2);
colormap jet;
colorbar;
title('\Delta Side scattering(x1 normal) (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f3.PaperUnits = 'points';
f3.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f3, './Plot/DeltaHXpScattering.png', '-dpng', '-r144');
% saveas(f3, './Plot/DeltaHXpScattering.fig');

f4 = figure;
imagesc(LambdaArr*1e9, base*1e9, x2_2);
colormap jet;
colorbar;
title('\Delta Side scattering(x2 normal) (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f4.PaperUnits = 'points';
```

```matlab
f4.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f4, './Plot/DeltaHXmScattering.png', '-dpng', '-r144');
% saveas(f4, './Plot/DeltaHXmScattering.fig');

f5 = figure;
imagesc(LambdaArr*1e9, base*1e9, -z1_2);
colormap jet;
colorbar;
title('\Delta Bottom scattering(z1 normal) (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f5.PaperUnits = 'points';
f5.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f5, './Plot/DeltaHZmScattering.png', '-dpng', '-r144');
% saveas(f5, './Plot/DeltaHZmScattering.fig');

f6 = figure;
imagesc(LambdaArr*1e9, base*1e9,z2_2);
colormap jet;
colorbar;
title('\Delta Top scattering(z2 normal) (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f6.PaperUnits = 'points';
f6.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f6, './Plot/DeltaHZpScattering.png', '-dpng', '-r144');
% saveas(f6, './Plot/DeltaHZpScattering.fig');

t2 = z2_2 - z1_2 + y2_2 - y1_2 + x2_2 - x1_2;

% f7 = figure;
% imagesc(LambdaArr*1e9, base*1e9,t2);
% colormap jet;
% colorbar;
% title('\Delta Total scattering(inside script calc.)');
% xlabel('\lambda (nm)');
% ylabel('Height (nm)');
% f6.PaperUnits = 'points';
% f6.PaperPosition = [0 0 640 480];
% print(f7, './Plot/DeltaHTotalScatteringScript.png', '-dpng', '-r144');
% saveas(f7, './Plot/DeltaHTotalScatteringScript.fig');

```

```
f8 = figure;
imagesc(LambdaArr*1e9, base*1e9,Total_2);
colormap jet;
colorbar;
title('\Delta Total scattering (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f6.PaperUnits = 'points';
f6.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f8, './Plot/DeltaHTotalScattering.png', '-dpng', '-r144');
% saveas(f8, './Plot/DeltaHTotalScattering.fig');

f9 = figure;
imagesc(LambdaArr*1e9, base*1e9,-Total);
colormap jet;
colorbar;
title('\Delta Total absorption (w_b = 900 nm)');
xlabel('\lambda (nm)');
ylabel('Height (nm)');
f6.PaperUnits = 'points';
f6.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
print(f9, './Plot/DeltaHTotalAbsorption.png', '-dpng', '-r144');
% saveas(f9, './Plot/DeltaHTotalAbsorption.fig');
```

B.4 Hermite-Gauss SPP beam reproduction

The following code is used for the discretization of the Hermite-Gauss SPP beam and processing of the FDTD simulation data.

```
% The analytical expression for second order Hermite-Gauss beam
clear;
close all;
load('../Data/AntennaArrayMesh4.mat', 'E0', 'E5', 'E10', 'E15'); % FDTD simulation data
Ez0 = E0.E(:,3);
Ez5 = E5.E(:,3);
Ez10 = E10.E(:,3);
Ez15 = E15.E(:,3);
xLum = round(E0.x*1e9)/1e3;
%% The relevant parameters
kp = 10.40e6; % real k in units of meter inversed
kpp = 0.059e6; % imaginary k
```

```matlab
lspp = 605; % The wavelength of SPP in nanometers
W0 = 2e-6; % as in the article
y0 = 15e-6; % as in the article
y = 0e-6; % the position of the exit line as in the article
x = (-8:0.01:8) * 1e-6; % the x array in micrometers
N = 7; % number of rows as in the article
%% The anonymous function definitions
W2 = @(y) (W0^2)*(1 + (2*(y - y0)./(kp*W0^2))^2);
R = @(y) (y - y0) * ( 1 + ((kp * W0^2)./(2 * (y - y0))).^2);
Phi = @(y) -atan(2*(y - y0)./(kp*W0^2));
% The expression for the z component of the electric field
EzSimp = hermiteH(2, ((sqrt(2).*x)./(sqrt(W2(y))))) .* exp((-x.^2)./(W2(y))) .*
 ↪   exp((1i*kp.*x.^2)./(2*R(y)));
%% Plot the amplitude of Ez
f1 = figure(1);
ax = axes;
NormEz = abs(EzSimp)./max(abs(EzSimp));
plot(ax, x*1e6, NormEz, '-b', 'linewidth', 2);
xlim([-7 7]);
ylim([0 1.1]);
title('Norm', 'fontweight', 'bold');
xlabel('x(\mum)', 'fontweight', 'bold');
ylabel('|E_z|', 'fontweight', 'bold');
ax.XColor = 'blue';
ax.YColor = 'blue';
ax.FontSize = 18;
hold on;
antennaNLR = [1, 4, 5, 7, 8, 10, 11];
antennaNM = [1, 3, 4, 5, 6, 7];
%% Plot the lines diving Ez
distLR = zeros(1,N);
distM = zeros(1,N-1);
Parr = zeros(1,N);
for i = 1:N
    l = ones(size(x))*((8-i)/7)*0.99;
    plot(x*1e6, l, '--m', 'linewidth', 2);
    % This intersection file is from the net
    P = InterX([x*1e6; l], [x*1e6; NormEz]);    lp = length(P(1,:));
    Parr(i) = P(1,1);
    LRpoint(i) = P(1,2);
    distLR(i) = P(1,2) - P(1,1);
    if i > 1
        distM(i-1) = P(1,4) - P(1,3);
    end

    plot(P(1,:), P(2,:), 'ro', 'markersize', 6, 'linewidth', 2);

end
```

```
hold off;
widthLR = round(distLR*1e3 ./ antennaNLR);
widthM = round(distM*1e3 ./ antennaNM);
fprintf('The widths(nm) of antennas to the left or right\n');
disp(widthLR.');
fprintf('The widths(nm) of antennas in the middle\n');
disp(widthM.');
f1.PaperUnits = 'points';
f1.PaperPosition = [0 0 640 480];
print(['./Plot/HermiteGaussMatlab.png'], '-dpng', '-r144');
% Plot the straight lines
% Manipulate them so they look like the axes in the article
% Default jump tolerance is too small
EzPhase = unwrap(angle(EzSimp), 3/2*pi);
EzPhase = EzPhase - min(EzPhase);
%% Plot the phase of Ez
f2 = figure(2);
PlotPhase =  plot(x*1e6,EzPhase, 'r', 'linewidth', 2);
ax = gca;
title('E_z Phase', 'fontweight', 'bold');
xlabel('x(\mum)', 'fontweight', 'bold');
ylabel('\Phi(rad)', 'fontweight', 'bold');
ax.YTick = [-2*pi, 0, 2*pi, 4*pi, 6*pi];
ax.YTickLabel = {'-2\pi', '0', '2\pi', '4\pi', '6\pi'};
ax.XColor = 'red';
ax.YColor = 'red';
ax.FontSize = 18;
ax.YLim = ([-2*pi, 7*pi]);
hold on;
% There are this many scatters adjacent to each other
Nadj = 29;
vertx = linspace(-max(abs(Parr)), max(abs(Parr)), Nadj);
F = griddedInterpolant(x*1e6, EzPhase);
% P is the array for phases, whereas Y is the array for coordinate y
P_ant = zeros(Nadj, 1);
for i = 1:Nadj
    Nvert = 100;
    xline = ones(Nvert,1) * vertx(i);
    y = linspace(-2*pi, 7*pi, Nvert);
    plot(xline,y, '--c', 'linewidth', 2);
    P_ant(i) = F(vertx(i));
    plot(vertx(i), P_ant(i), 'bo', 'markersize', 6, 'linewidth', 2);
end
f2.PaperUnits = 'points';
f2.PaperPosition = [0 0 640 480];
print(['./Plot/HermiteGaussMatlabPhase.png'], '-dpng', '-r144');
% How much the scatterers are going to be displaced relative to middle
% section
```

```
Y_ant = -round((P_ant * lspp)/(2*pi));
Y_ant = round(Y_ant - mean(Y_ant(12:18)));

f3 = figure;
y = 15e-6;
EzSimp = hermiteH(2, ((sqrt(2).*x)./(sqrt(W2(y))))) .* exp((-x.^2)./(W2(y))) .* 1;
NormEz = abs(EzSimp)./max(abs(EzSimp));
plot(x*1e6, NormEz, '-b', 'linewidth', 2);
ax = gca;
ax.FontSize = 18;
xlim([-7 7]);
ylim([0 1.1]);
title('Norm', 'fontweight', 'bold');
xlabel('x(\mum)', 'fontweight', 'bold');
ylabel('|E_z|', 'fontweight', 'bold');
hold on;
Ez15 = abs(Ez15)./ max(abs(Ez15));
plot(xLum, Ez15, 'r--', 'linewidth', 2);
legend('Analytic', 'Num 15\mum');
f3.PaperUnits = 'points';
f3.PaperPosition = [0 0 640 480];
print(['./Plot/HermiteGaussCompare.png'], '-dpng', '-r144');

f4 = figure;
y = 10e-6;
EzSimp = hermiteH(2, ((sqrt(2).*x)./(sqrt(W2(y))))) .* exp((-x.^2)./(W2(y))) .* exp((1i*kp.*x.^2)./(2*R(y)));
NormEz = abs(EzSimp)./max(abs(EzSimp));
EzPhase = unwrap(angle(EzSimp), 3/2*pi);
EzPhase = EzPhase - min(EzPhase);
plot(x*1e6, EzPhase, '-b', 'linewidth', 2);
ax = gca;
xlim([-7 7]);
% ylim([0 1.1]);
title('Phase', 'fontweight', 'bold');
xlabel('x(\mum)', 'fontweight', 'bold');
ylabel('|E_z|', 'fontweight', 'bold');
ax.FontSize = 18;
hold on;
% Ez10 = abs(Ez10)./ max(abs(Ez10));
Ez10Phase = unwrap(angle(Ez10), 3/2*pi);
Ez10Phase = Ez10Phase - min(Ez10Phase) + pi;
plot(xLum, Ez10Phase, 'r--', 'linewidth', 2);
legend('Analytic', 'Num 10\mum');
f4.PaperUnits = 'points';
f4.PaperPosition = [0 0 640 480];
print(['./Plot/HermiteGaussComparePhase.png'], '-dpng', '-r144');
```

B.5 Triplexer algorithm

B.5.1 Original Triplexer Algorithm

For confirmation of our implementation of the algorithm we reproduced the article results with the following code.

```
clear;
close all;
% load('../Data/Takuo2011A_T.mat');
load('../Data/kSPP700-1000.mat'); % kspp at silica-gold interface
nk = numel(kspp);
dy = 20; % in nanometers
y_field = -6000:dy:6000;
x_slit = -25000:3000:-10000;
y_slit = -3000:1500:3000;
y_field = y_field * 1e-9;
x_slit = x_slit * 1e-9;
y_slit = y_slit * 1e-9;
wbeam = (17/2)*1e-6; % radius of Gaussian beam
GBeam = @(r) exp(-2*r^2/(wbeam^2));
% Aint = griddedInterpolant(lambdaLUM(:), A(:));
% A_val = Aint(linspace(700, 1000, nk));
% A_vec = reshape(A_val, nk, 1); %  the wavelength dependence of SPP excitation
A_vec = ones(nk, 1);
T_vec = ones(size(A_vec)); % the wavelength dependence of transmission normal to the slit
r_field = Vector2D; % See the relevant file for class definition
r_field.X = zeros(size(y_field));
r_field.Y = y_field;
r_pattern = Vector2D;
r_pattern.X = -17.5e-6;
r_pattern.Y = 0;
[XXslit, YYslit] = meshgrid(x_slit, y_slit);
% Loop over slits
Esum = zeros(nk, 601);
for i = 1:numel(XXslit) % Careful length does not work here
% for i = 1
    r_slit{i} = Vector2D;
    r_slit{i}.X = XXslit(i);
    r_slit{i}.Y = YYslit(i);
    d{i} = distance(r_slit{i}, r_field);
    cos_vec{i} = (r_field.X - r_slit{i}.X) ./ d{i};
    % Adjust for Gaussian beam
    factor = GBeam(distance(r_slit{i}, r_pattern));
%     factor = 1;
    Econtr{i} = (A_vec .* T_vec * factor * (cos_vec{i} ./ sqrt(d{i}))) .* exp(1i .* kspp * d{i});
    Esum = Esum + Econtr{i};
```

```
end
f1 =figure;
yplot = y_field * 1e6;
lplot = lambda * 1e9;
Eplot = (abs(Esum).').^2;
imagesc(lplot, yplot, 100 * Eplot./(max(max(Eplot))));
colormap jet;
c = colorbar;
ax = gca;
ax.YDir = 'normal';
ax.FontSize = 18;
title('Initial Guess Pattern');
xlabel('\lambda(nm)');
ylabel('y(\mum)');
ylabel(c, 'Field intensity(a.u)');
f1.PaperUnits = 'points';
f1.PaperPosition = [0 0 640 480];
fname = 'Takuo2011_InitialGuessArticle';
% print(['./Plot/' fname], '-depsc', '-r144');
print(['./Plot/' fname], '-dpng', '-r144');
% saveas(f1, ['./Plot/' fname]);x_focus = zeros(1,3);
y_focus = [-3, 0, 3] * 1e-6; % focus points
l_focus = [820, 850, 880] * 1e-9; % desired wavelengths at focii
W = ones(size(y_focus)); % initialize the weight to be 1 as in the article
y_focii_ave = mean(y_focus);
x_focii_ave = mean(x_focus);
q = 0.1;
Nite = 40;
for i = 1:numel(y_focus)
    r_focus{i} = Vector2D;
    r_focus{i}.X = 0;
    r_focus{i}.Y = y_focus(i);
    % Beware of floating point numbers
    y_ind(i) = find(abs(y_field - y_focus(i)) < 0.5e-9);
    l_ind(i) = find(abs(lambda - l_focus(i)) < 0.5e-9);
end
lspp_mean = mean(lspp(l_ind));
r_focus_ave = Vector2D;
r_focus_ave.X = x_focii_ave;
r_focus_ave.Y = y_focii_ave;
% Calculate the linear indices to avoid a nested for loop
lin_ind = sub2ind(size(Esum), l_ind, y_ind); % Verified works well
%% The actual iterations take place here
%% The actual iterations take place here
for i = 1:Nite % total no of iterations defined above
    Emem = Esum;
    Esum = 0;
        % Upgrade W as in the algorithm
```

```matlab
    if i ~= 1 % Adjust this number if need be
        W = W .* (max(abs(Emem(lin_ind)).^2)./abs(Emem(lin_ind)).^2).^q;
    end
    for n = 1:numel(XXslit) % phase mismatch is averaged for each slit
        delta_n(n) = angle(sum(W .* Emem(lin_ind) .* conj(Econtr{n}(lin_ind)) ./ ...
            (abs(Emem(lin_ind) .* abs(Econtr{n}(lin_ind))))));
        new_dis2foc = distance(r_slit{n}, r_focus_ave) + delta_n(n) * (lspp_mean / (2*pi));
        xnew = -sqrt(new_dis2foc(1).^2 - (r_slit{n}.Y(1) - r_focus_ave.Y)^2) + r_focus_ave.X;
        r_slit{n}.X  = double(xnew);
        slit_x(n) = double(xnew);
        slit_y(n) = r_slit{n}.Y;
        % Recalculate the fields
        d{n} = distance(r_slit{n}, r_field);
        cos_vec{n} = (r_field.X - r_slit{n}.X) ./ d{n};
        factor = GBeam(distance(r_slit{n}, r_pattern));
%         factor = 1;
        % A_vec may be adjusted to incorparate Gaussian beam for each slit
       Econtr{n} = (A_vec .* T_vec * factor * (cos_vec{n} ./ sqrt(d{n}))) .* exp(1i .* kspp * d{n});
        Esum = Esum + Econtr{n};
    end
end
%% Lets get the fields again to plot
Econtr(1,:) = {0};
Esum = 0;
for i = 1:numel(XXslit) % Careful length does not work here
    d{i} = distance(r_slit{i}, r_field);
    cos_vec{i} = (r_field.X - r_slit{i}.X) ./ d{i};
    % A_vec may be adjusted to incorparate Gaussian beam for each slit
    factor = GBeam(distance(r_slit{i}, r_pattern));
%     factor = 1;
    Econtr{i} = (A_vec .* T_vec * factor * (cos_vec{i} ./ sqrt(d{i}))) .* exp(1i .* kspp * d{i});
    Esum = Esum + Econtr{i};
end
f2 = figure;
yplot = y_field * 1e6;
lplot = lambda * 1e9;
Eplot = (abs(Esum).').^2;
imagesc(lplot, yplot, 100 * Eplot./(max(max(Eplot))));colormap jet;
c = colorbar;
ax = gca;
ax.FontSize = 18;
ax.YDir = 'normal';
title('After algorithm Pattern');
xlabel('\lambda(nm)');
ylabel('y(\mum)');
ylabel(c, 'Field intensity(a.u)');
f2.PaperUnits = 'points';
f2.PaperPosition = [0 0 640 480];
```

```
fname = 'Takuo2011_OptimizedArticle';
% print(['./Plot/' fname], '-depsc', '-r144');
print(['./Plot/' fname], '-dpng', '-r144');
% saveas(f2, ['./Plot/' fname]);

f3 = figure;
plotstyle = {'b-', 'g-', 'r-'};
maximum = max(max(abs(Esum(:, y_ind)).^2));
for m = 1:numel(y_focus)
    plot(lplot, abs(Esum(:, y_ind(m))).^2 ./maximum, plotstyle{m}, 'linewidth', 2)
    if m == 1
        hold on;
    end
    legendinfo{m} = [sprintf('y = %2d ', y_field(y_ind(m))*1e6) '\mum'];
end
xlabel('\lambda(nm)');
ylabel('Intensity(a.u)');
title('Calculated SPP Intensity');
legend(legendinfo);
ax = gca;
ax.FontSize = 18;
f3.PaperUnits = 'points';
f3.PaperPosition = [0 0 640 480];
fname = 'Takuo2011_IndividualArticle';
% print(['./Plot/' fname], '-depsc', '-r144');
print(['./Plot/' fname], '-dpng', '-r144');
% saveas(f3, ['./Plot/' fname]);

% plot(slit_x * 1e6, slit_y * 1e6, 'rs', 'MarkerSize', 12);
vertX = [-6 -6 6 6];
vertY = [-10 10 10 -10];
[f4,patchHndl,lineHndl] = plotCustMark(slit_x * 1e6, slit_y *1e6, vertX,vertY,0.03);
axis square;
box on;
grid on;
xlabel('x (\mum)');
ylabel('y (\mum)');
title('Nanoslit Positions');
ax = gca;
ax.FontSize = 18;
f4.PaperUnits = 'points';
f4.PaperPosition = [0 0 640 480];
axis([-26 0 -4 4]);
fname = 'Takuo2011_PositionsArticle';
% print(['./Plot/' fname], '-depsc', '-r144');
print(['./Plot/' fname], '-dpng', '-r144');
% saveas(f4, ['./Plot/' fname]);
```

B.5.2 Triplexer for Δ-antennas

This is the code which calculates the positions of the Δ-antennas which yielded the results used in this thesis.

```
clear;
close all;
% Run the script which initializes the slit positions etc.
Takuo2011CalcDelta;
x_focus = zeros(1,3);
y_focus = [-4, 0, 4] * 1e-6; % focus points
l_focus = [1400, 1500, 1600] * 1e-9; % desired wavelengths at focii
W = ones(size(y_focus)); % initialize the weight to be 1 as in the article
y_focii_ave = mean(y_focus);
x_focii_ave = mean(x_focus);
q = 0.1;
Nite = 80;
for i = 1:numel(y_focus)
    r_focus{i} = Vector2D;
    r_focus{i}.X = 0;
    r_focus{i}.Y = y_focus(i);
    % Beware of floating point numbers
    y_ind(i) = find(abs(y_field - y_focus(i)) < 0.5e-9);
    l_ind(i) = find(abs(lambda - l_focus(i)) < 0.5e-9);
end
lspp_mean = mean(lspp(l_ind));
r_focus_ave = Vector2D;
r_focus_ave.X = x_focii_ave;
r_focus_ave.Y = y_focii_ave;
% Calculate the linear indices to avoid a nested for loop
lin_ind = sub2ind(size(Esum), l_ind, y_ind); % Verified works well
%% The actual iterations take place here
for i = 1:Nite % total no of iterations defined above
        Emem = Esum;
        Esum = 0;
    % Upgrade W as in the algorithm
    if i ~= 1 % Adjust this number if need be
        W = W .* (max(abs(Emem(lin_ind)).^2)./abs(Emem(lin_ind)).^2).^q;
    end
    for n = 1:numel(XXslit) % phase mismatch is averaged for each slit
        delta_n(n) = angle(sum(W .* Emem(lin_ind) .* conj(Econtr{n}(lin_ind)) ./ ...
            (abs(Emem(lin_ind) .* abs(Econtr{n}(lin_ind))))));
        new_dis2foc = distance(r_slit{n}, r_focus_ave) + delta_n(n) * (lspp_mean / (2*pi));
        xnew = -sqrt(new_dis2foc(1).^2 - (r_slit{n}.Y(1) - r_focus_ave.Y)^2) + r_focus_ave.X;
        r_slit{n}.X  = double(xnew);
        slit_x(n) = double(xnew);
        slit_y(n) = r_slit{n}.Y;
        % Recalculate the fields
        d{n} = distance(r_slit{n}, r_field);
        cos_vec{n} = (r_field.X - r_slit{n}.X) ./ d{n};
        %% The above line is replaced
        rad = atan2((r_field.Y - r_slit{n}.Y), (r_field.X - r_slit{n}.X)) + pi/2;
        rad(find(rad<0)) = rad(find(rad<0)) + 2 * pi;
        %% We use the fit here
        cos_vec2{n} = EzSurfCircInt{1}(rad);
        aux = [];
        for ii = 1:length(EzSurfCircInt)
            aux = [aux; EzSurfCircInt{ii}(rad)];
        end
        [LLUM, TT] = ndgrid(lambdaLUM, rad);
        GI = griddedInterpolant(LLUM, TT, aux);
        [LL, TT] = ndgrid(lambda, rad);
        cos_vec2{n} = GI(LL, TT);
        % Adjust for Gaussian beam
        Gaussian = GBeam(distance(r_slit{n}, r_pattern));
        d_mat = repmat(d{n}, 501, 1);
        F_mat = repmat(A_vec .* T_vec, 1, 601);
        Econtr{n} = (F_mat .* Gaussian .* (cos_vec2{n} ./ sqrt(d_mat))) .* exp(1i .* kspp * d{n});
        Esum = Esum + Econtr{n};
    end
end
%% Lets get the fields again to plot
Econtr(1,:) = {0};
Esum = 0;
for i = 1:numel(XXslit) % Careful length does not work here
    d{i} = distance(r_slit{i}, r_field);
    cos_vec2{i} = (r_field.X - r_slit{i}.X) ./ d{i};
    %% Fit is used here
    aux = [];
    for ii = 1:length(EzSurfCircInt)
        aux = [aux; EzSurfCircInt{ii}(rad)];
    end
    [LLUM, TT] = ndgrid(lambdaLUM, rad);
    GI = griddedInterpolant(LLUM, TT, aux);
    [LL, TT] = ndgrid(lambda, rad);
    cos_vec2{i} = GI(LL, TT);
    % Adjust for Gaussian beam
    Gaussian = GBeam(distance(r_slit{i}, r_pattern));
    d_mat = repmat(d{i}, 501, 1);
    F_mat = repmat(A_vec .* T_vec, 1, 601);
    Econtr{i} = (F_mat .* Gaussian .* (cos_vec2{i} ./ sqrt(d_mat))) .* exp(1i .* kspp * d{i});
    Esum = Esum + Econtr{i};
end
f2 = figure(2);
yplot = y_field * 1e6;
```

```
lplot = lambda * 1e9;
Eplot = (abs(Esum).').^2;
imagesc(lplot, yplot, 100 * Eplot./(max(max(Eplot))));colormap jet;
c = colorbar;
ax = gca;
ax.FontSize = 18;
ax.YDir = 'normal';
title('After algorithm Pattern');
xlabel('\lambda(nm)');
ylabel('y(\mum)');
ylabel(c, 'Field intensity(a.u)');
f2.PaperUnits = 'points';
f2.PaperPosition = [0 0 640 480];
fname = 'Takuo2011_OptDelta2silica';
% print(['./Plot/' fname], '-depsc', '-r144');
print(['./Plot/' fname], '-dpng', '-r144');
% saveas(f2, ['./Plot/' fname]);

f3 = figure;
plotstyle = {'b-', 'g-', 'r-'};
maximum = max(max(abs(Esum(:, y_ind)).^2));
for m = 1:numel(y_focus)
    plot(lplot, abs(Esum(:, y_ind(m))).^2 ./maximum, plotstyle{m}, 'linewidth', 2)
    if m == 1
        hold on;
    end
    legendinfo{m} = [sprintf('y = %.1f ', y_field(y_ind(m))*1e6) '\mum'];
end
xlabel('\lambda(nm)');
ylabel('Intensity(a.u)');
title('Calculated SPP Intensity');
legend(legendinfo);
ax = gca;
ax.FontSize = 18;
f3.PaperUnits = 'points';
f3.PaperPosition = [0 0 640 480];
fname = 'Takuo2011_IndDelta2silica';
% print(['./Plot/' fname], '-depsc', '-r144');
print(['./Plot/' fname], '-dpng', '-r144');
% saveas(f3, ['./Plot/' fname]);

% plot(slit_x * 1e6, slit_y * 1e6, 'rs', 'MarkerSize', 12);
vertX = [-15 15 15];
vertY = [0 5 -5];
[f4,patchHndl,lineHndl] = plotCustMark(slit_x * 1e6, slit_y *1e6, vertX,vertY,0.05);
axis square;
box on;
grid on;
```

```
xlabel('x (\mum)');
ylabel('y (\mum)');
title('\Delta-antenna positions');
f4.PaperUnits = 'points';
f4.PaperPosition = [0 0 640 480];
ax = gca;
ax.FontSize = 18;
axis([-26 0 -4 4]);
fname = 'Takuo2011_PosDelta2silica';
% print(['./Plot/' fname], '-depsc', '-r144');
print(['./Plot/' fname], '-dpng', '-r144');
% saveas(f4, ['./Plot/' fname]);
save('../Data/LumericalDeltaPositions.mat', 'slit_*');
```

In some of these codes external functions published on file exchange part of the Matlab Central website were used after minor modifications. The links for these functions are given below.

- Custom Marker Plot - `plotCustMark`

- Refined Polar Plot - `polarticks`

Appendix C

LUMERICAL FDTD SOLUTIONS SCRIPTS

The Lumerical FDTD Solutions program features a scripting language to manipulate simulation objects, provide an interface between Matlab and Lumerical, launch simulations, and analyze results. The files have the extension ".lsf" standing for Lumerical Script File.

C.1 Simulation Setup

The following script is used for setting up the simulation and saving them as individual files to be able to run them on high performance computing cluster. This file also sets the various monitors used for calculating scattering cross section.

```
# Create the base width vector for the triangular scatterer
height = (50:50:300)*1e-9;
hmax = 300e-9;
base = 500e-9;
##############################################################
# start a loop over each desired base width
for(i=1:length(height)) {
    save('Delta500small' + num2str(i) + '.fsp');
    # switch to layout mode so that you can edit the objects
    switchtolayout;
    # set the height of the delta-antenna
    select("DeltaAntenna");
    set("Height",height(i));
    set("Width", base);
    # set the inner trans_box parameters
    select("trans_box");
    set("x1", 280e-9);
    set("x2", 280e-9);
    #set("y1", 30e-9 + (height(i)/2));
    #set("y2", 30e-9 + (height(i)/2));
    set("y1", 30e-9 + (hmax/2));
    set("y2", 30e-9 + (hmax/2));
    # set the TFSF source parameters
    select("TFSF");
    set("y", 0e-9);
    #set("y span", height(i) + 120e-9);
    set("y span", hmax + 120e-9);
    set("x", 0e-9);
    set("x span", 620e-9);
    # set the outer trans_box parameters
    select("trans_box2");
    set("x1", 340e-9);
    set("x2", 340e-9);
    #set("y1", 90e-9 + (height(i)/2));
    #set("y2", 90e-9 + (height(i)/2));
    set("y1", 90e-9 + (hmax/2));
    set("y2", 90e-9 + (hmax/2));
    # output which simulation is running
    ?"Setting simulation " + num2str(i) + " of " + num2str(length(height));

    save('Delta500small' + num2str(i) + '.fsp');
} # end of the main loop over the radius
```

C.2 Data Extraction

A sample script used for data extraction in Matlab format is given below. The data from various monitors on the cube surrounding the scatter is saved in different matrices; two dimensions are necessary to carry wavelength and height information at the same time.

```
clear;
# The used basewidth values for delta antennas
height = linspace(250e-9,1400e-9,24);
base = height; # for script purposes
# Loop over base width
for(i=1:length(base)) {
    f_name="DeltaHeight"+num2str(i);
    # Load files
    load(f_name);
    temp =  getresult("trans_box","T");
    temp2 = getresult("trans_box2","T");
    LambdaArr = temp.lambda;
    f = temp.f;
    sp=sourcepower(f);     # get power injected by source (Watts)
    I=sourceintensity(f);   # get source intensity (Watts/m^2)
    area = getdata("TFSF","area"); # get source area
    if (i == 1) {
        # Reserve the matrices
        L = length(temp.lambda);
        x1 = matrix(length(base), L);
        x2 = matrix(length(base), L);
        y1 = matrix(length(base), L);
        y2 = matrix(length(base), L);
        z1 = matrix(length(base), L);
        z2 = matrix(length(base), L);
        Total = matrix(length(base), L);
        x1_2 = matrix(length(base), L);
        x2_2 = matrix(length(base), L);
        y1_2 = matrix(length(base), L);
        y2_2 = matrix(length(base), L);
        z1_2 = matrix(length(base), L);
        z2_2 = matrix(length(base), L);
        Total_2 = matrix(length(base), L);
        Iarr = matrix(length(base), L);
        SParr = matrix(length(base), L);
        Areaarr = matrix(length(base), L);
    }
    # Transmission data from individual monitors
    # Uncommenting the following line cancels the normalization
    #sp = I;
    x1(i, 1:L) = transpose(transmission("trans_box::x1")*sp/I);
    x2(i, 1:L) = transpose(transmission("trans_box::x2")*sp/I);
    y1(i, 1:L) = transpose(transmission("trans_box::y1")*sp/I);
    y2(i, 1:L) = transpose(transmission("trans_box::y2")*sp/I);
    z1(i, 1:L) = transpose(transmission("trans_box::z1")*sp/I);
    z2(i, 1:L) = transpose(transmission("trans_box::z2")*sp/I);
    Iarr(i, 1:L) = I;
    SParr(i, 1:L) = sp;
    Areaarr(i, 1:L) = area;
    x1_2(i, 1:L) = transpose(transmission("trans_box2::x1")*sp/I);
    x2_2(i, 1:L) = transpose(transmission("trans_box2::x2")*sp/I);
    y1_2(i, 1:L) = transpose(transmission("trans_box2::y1")*sp/I);
    y2_2(i, 1:L) = transpose(transmission("trans_box2::y2")*sp/I);
    z1_2(i, 1:L) = transpose(transmission("trans_box2::z1")*sp/I);
    z2_2(i, 1:L) = transpose(transmission("trans_box2::z2")*sp/I);
    # get total from the analysis object
    runanalysis;
    Total(i, 1:L) = transpose(temp.T)*sp/I;
    Total_2(i, 1:L) = transpose(temp2.T)*sp/I;
} # end of the main loop over the radius

matlabsave('DeltaHeightSweepData.mat');
```

C.3 Scatterer positioning for reproduction of SPP beam

The following script positions the Δ-antennas in the correct positions to generate the Hermite-Gauss beam. Note that the structure ”Antenna” must be defined.

```
# Create the base width vector for the triangular scatterer
baseLR = [310, 310, 360, 325, 345, 330, 360] * 1e-9;
baseM = [365, 325, 340, 335, 325, 315] * 1e-9;
# Create the position vectors for antennas
PosXL = [-2604, -2179, -1958, -1784, -1633, -1494, -1362] * 1e-9;
PosXR = [2604, 2179, 1958, 1784, 1633, 1494, 1362] * 1e-9;
PosXM =  [-183, -489, -676, -832, -972, -1104] * 1e-9;
# Create arrays for the number of antennas
NumLR = [1, 4, 5, 7, 8, 10, 11];
NumM = [1, 3, 4, 5, 6, 7];
ydisp = 605e-9; # as in the article
yheight = 505e-9; # height of antennas
# Create the array for the displacement required for the realization
# of the phase difference
Pdisp = [622, 577, 536, 498, 463, 431, 403, 378, 356, 338, 323, 9] * 1e-9;
# switch to layout mode so that you can edit the objects
switchtolayout;
select("::model::ScriptGenerated");
delete;
for(i=1:length(baseLR)) {
    # Put the antennas to the left
    for(j=1:NumLR(i)) {
        select("Antenna");
        copy;
        set("x", PosXL(i) - (2*j - 1)*baseLR(i)/2  );
        set("y", 0 - ydisp*(i-1) - yheight/2 + Pdisp(12-j));
        set("name", "AntennaL" + num2str(i) + num2str(j));
        set("Width",baseLR(i));
        set("enabled", 1);
        addtogroup("ScriptGenerated");
    } # end of for loop for antennas to the left
    if (i > 1) {
        i = i - 1;
        for(j=1:NumM(i)) {
            select("Antenna");
            copy;
            set("x", PosXM(i) + (2*j - 1)*baseM(i)/2  );
            set("y", 0 - ydisp*(i) - yheight/2);
            set("name", "AntennaM" + num2str(i) + num2str(j));
            set("Width",baseM(i));
            set("enabled", 1);
            addtogroup("ScriptGenerated");
        }
        i = i + 1;
    }
    # Put the antennas to the left
    for(j=1:NumLR(i)) {
        select("Antenna");
        copy;
        set("x", PosXR(i) + (2*j - 1)*baseLR(i)/2  );
        set("y", 0 - ydisp*(i-1) - yheight/2 + Pdisp(12-j));
        set("name", "AntennaR" + num2str(i) + num2str(j));
        set("Width",baseLR(i));
        set("enabled", 1);
        addtogroup("ScriptGenerated");
    } # end of for loop for antennas to the left
} # End of for loop for the number of rows
```

C.4 *Scatterer positioning for triplexer*

The following script positions the nanoslits in an aperiodic array to achieve wavelength demultiplexing by focusing three wavelengths at distinct focii.

```
matlabload("../Data/LumericalSlitPositions.mat");
for(i=1:length(slit_x)) {
    select("Nanoslit");
    copy;
    set("x", slit_x(i));
    set("y", slit_y(i));
    set("name", "Scatterer" + num2str(i));
    set("enabled", 1);
    addtogroup("ScriptGenerated");
} # end of for loop
```

BIBLIOGRAPHY

[1] M. L. Brongersma, “Introductory lecture: nanoplasmonics”, Faraday Discuss. **178**, 9–36 (2015) [vii, ix, 1].

[2] T. Tanemura, K. C. Balram, D.-S. Ly-Gagnon, P. Wahl, J. S. White, M. L. Brongersma, and D. A. B. Miller, “Multiple-wavelength focusing of surface plasmons with a non-periodic nanoslit coupler”, Nano Letters **11**, 2693–2698 (2011) [viii, 2, 18, 20, 41, 56, 57].

[3] A. Sommerfeld, “Über die fortpflanzung elektrodynamischer wellen längs eines drahtes”, Annalen der Physik **303**, 233–290 (1899) [1].

[4] J. Zenneck, “Über die fortpflanzung ebener elektromagnetischer wellen längs einer ebenen leiterfläche und ihre beziehung zur drahtlosen telegraphie”, Annalen der Physik **328**, 846–866 (1907) [1].

[5] R. W. Wood, “On a remarkable case of uneven distribution of light in a diffraction grating spectrum”, Proceedings of the Physical Society of London **18**, 269 (1902) [1].

[6] U. Fano, “The theory of anomalous diffraction gratings and of quasi-stationary waves on metallic surfaces (sommerfeld’s waves)”, J. Opt. Soc. Am. **31**, 213–222 (1941) [1].

[7] R. H. Ritchie, “Plasma losses by fast electrons in thin films”, Phys. Rev. **106**, 874–881 (1957) [1].

[8] R. Ritchie, E. Arakawa, J. Cowan, and R. Hamm, “Surface-plasmon resonance effect in grating diffraction”, Physical Review Letters **21**, 1530–1533 (1968) [1].

[9] E. N. Economou, “Surface plasmons in thin films”, Physical Review **182**, 539–554 (1969) [1].

[10] S. L. Cunningham, A. A. Maradudin, and R. F. Wallis, “Effect of a charge layer on the surface-plasmon-polariton dispersion curve”, Phys. Rev. B **10**, 3342–3355 (1974) [1].

[11] M. L. Brongersma, and V. M. Shalaev, “The case for plasmonics”, Science **328**, 440–441 (2010) [1].

[12] J. Zhang, L. Zhang, and W. Xu, "Surface plasmon polaritons: physics and applications", Journal of Physics D: Applied Physics **45**, 113001 (2012) [1].

[13] S. Hayashi, and T. Okamoto, "Plasmonics: visit the past to know the future", Journal of Physics D: Applied Physics **45**, 433001 (2012) [1].

[14] O. You, B. Bai, X. Wu, Z. Zhu, and Q. Wang, "A simple method for generating unidirectional surface plasmon polariton beams with arbitrary profiles", Opt. Lett. **40**, 5486–5489 (2015) [1, 18, 24, 25, 32, 56].

[15] A. Minovich, A. E. Klein, N. Janunts, T. Pertsch, D. N. Neshev, and Y. S. Kivshar, "Generation and near-field imaging of airy surface plasmons", Phys. Rev. Lett. **107**, 116802 (2011) [2].

[16] J. Lin, J. Dellinger, P. Genevet, B. Cluzel, F. de Fornel, and F. Capasso, "Cosine-gauss plasmon beam: a localized long-range nondiffracting surface wave", Phys. Rev. Lett. **109**, 093904 (2012) [2].

[17] L. Li, T. Li, S. M. Wang, and S. N. Zhu, "Collimated plasmon beam: nondiffracting versus linearly focused", Phys. Rev. Lett. **110**, 046807 (2013) [2].

[18] A. Libster-Hershko, I. Epstein, and A. Arie, "Rapidly accelerating mathieu and weber surface plasmon beams", Phys. Rev. Lett. **113**, 123902 (2014) [2].

[19] Q. Xu, X. Zhang, Y. Xu, C. Ouyang, Z. Tian, J. Gu, J. Li, S. Zhang, J. Han, and W. Zhang, "Polarization-controlled surface plasmon holography", Laser & Photonics Reviews **11**, 1600212 (2016) [2].

[20] A. Drezet, D. Koller, A. Hohenau, A. Leitner, F. R. Aussenegg, and J. R. Krenn, "Plasmonic crystal demultiplexer and multiports", Nano Letters **7**, PMID: 17500579, 1697–1700 (2007) [2].

[21] E. Laux, C. Genet, T. Skauli, and T. W. Ebbesen, "Plasmonic photon sorters for spectral and polarimetric imaging", Nature Photonics **2**, 161–164 (2008) [2].

[22] C. Zhao, and J. Zhang, "Plasmonic demultiplexer and guiding", ACS Nano **4**, PMID: 20925325, 6433–6438 (2010) [2].

[23] L. Li, T. Li, S. Wang, S. Zhu, and X. Zhang, “Broad band focusing and demultiplexing of in-plane propagating surface plasmons”, Nano Letters **11**, PMID: 21939224, 4357–4361 (2011) [2].

[24] T. Tanemura, P. Wahl, S. Fan, and D. A. B. Miller, “Modal source radiator model for arbitrary two-dimensional arrays of subwavelength apertures on metal films”, IEEE Journal of Selected Topics in Quantum Electronics **19**, 4601110–4601110 (2013) [2, 18].

[25] P. Wahl, T. Tanemura, N. Vermeulen, J. V. Erps, D. A. B. Miller, and H. Thienpont, “Design of large scale plasmonic nanoslit arrays for arbitrary mode conversion and demultiplexing”, Optics Express **22**, 646 (2014) [2].

[26] D. Wintz, P. Genevet, A. Ambrosio, A. Woolf, and F. Capasso, “Holographic metalens for switchable focusing of surface plasmons”, Nano Letters **15**, PMID: 25915541, 3585–3589 (2015) [3].

[27] J. D. Jackson, *Classical electrodynamics.* (New York : Wiley, 1999) [5].

[28] S. T. Thornton, and J. B. Marion, *Classical dynamics of particles and systems.* (Belmont, CA : Brooks/Cole, 2004) [5].

[29] D. J. Griffiths, *Introduction to electrodynamics.* (Upper Saddle River, N.J. : Prentice Hall, 1999) [5, 6].

[30] M. Fox, *Optical properties of solids.* Oxford master series in condensed matter physics (Oxford [England] ; New York : Oxford University Press, 2001) [7, 8].

[31] S. A. Maier, *Plasmonics : fundamentals and applications.* (New York : Springer, 2007) [9, 11].

[32] C. Kittel, *Introduction to solid state physics.* (Hoboken, NJ : Wiley, 2005) [12].

[33] T. Iqbal, and S. Afsheen, “Coupling efficiency of surface plasmon polaritons for 1D plasmonic gratings: role of under- and over-milling”, Plasmonics **11**, 1247–1256 (2016) [18].

[34] O. You, B. Bai, and X. Li, “Experimental study of unidirectional excitation of spps by binary area-coded nanohole arrays”, Chin. Opt. Lett. **12**, 082401 (2014) [18, 26].

[35] B. Bai, X. Meng, J. Laukkanen, T. Sfez, L. Yu, W. Nakagawa, H. P. Herzig, L. Li, and J. Turunen, "Asymmetrical excitation of surface plasmon polaritons on blazed gratings at normal incidence", Phys. Rev. B **80**, 035407 (2009) [18, 26].

[36] *Lumerical solutions, inc.* `http://www.lumerical.com/tcad-products/fdtd/` [18].

[37] C. Chen, and P. Berini, "Broadside excitation of long-range surface plasmons via grating coupling", IEEE Photonics Technology Letters **21**, 1831–1833 (2009) [18].

[38] C. Chen, and P. Berini, "Grating couplers for broadside input and output coupling of long-range surface plasmons", Optics Express **18**, 8006 (2010) [18].

[39] Matlab R2015b, (The MathWorks Inc., Natick, Massachusetts, 2015) [22].

[40] J. M. Steele, Z. Liu, Y. Wang, and X. Zhang, "Resonant and non-resonant generation and focusing of surface plasmons with circular gratings", Opt. Express **14**, 5664–5670 (2006) [22, 41].

[41] H. Liu, and P. Lalanne, "Light scattering by metallic surfaces with subwavelength patterns", Physical Review B **82**, 115418 (2010) [23, 30].

[42] J. Le Perchec, "Origin of the norton-type wave scattered by a subwavelength metallic slit", Phys. Rev. B **92**, 155404 (2015) [30].

[43] P. Lalanne, J. Hugonin, H. Liu, and B. Wang, "A microscopic view of the electromagnetic properties of sub- metallic surfaces", Surface Science Reports **64**, 453–469 (2009) [31].

[44] W. L. Stutzman, and G. A. Thiele, *Antenna theory and design.* (New York : John Wiley & Sons, Inc., c1998, 1998) [39].

[45] J. Seidel, S. Grafström, L. Eng, and L. Bischoff, "Surface plasmon transmission across narrow grooves in thin silver films", Applied Physics Letters **82**, 1368–1370 (2003) [41].

[46] P. B. Johnson, and R. W. Christy, "Optical constants of the noble metals", Phys. Rev. B **6**, 4370–4379 (1972) [42].

[47] J. Backlund, J. Bengtsson, C.-F. Carlström, and A. Larsson, "Incoupling waveguide holograms for simultaneous focusing into multiple arbitrary positions", Appl. Opt. **38**, 5738–5746 (1999) [42, 44].

[48] R. Paschotta, *Wavelength division multiplexing*, `https://www.rp-photonics.com/wavelength_division_multiplexing.html` (visited on 05/02/2017) [56].

[49] N. S. Bergano, and C. R. Davidson, "Wavelength division multiplexing in long-haul transmission systems", Journal of Lightwave Technology **14**, 1299–1308 (1996) [56].

[50] G. E. Keiser, "A review of WDM technology and applications", Optical Fiber Technology **5**, 3–39 (1999) [56].