Simulation of high-purity circularly polarized light from a waveguide ring resonator with azimuthal grating

Tom Schatteburg
Supervisor: Dr. Karan Kartik Mehta
Group leader: Prof. Dr. Jonathan Home
ETH Zürich

July 30, 2018

Abstract

An integrated optics ring resonator with an angular emission grating was simulated using the volume current method (VCM) and the finite-difference time-domain (FDTD) method. FDTD simulations of the coupling of the feeding waveguide to a half ring as well as simulations of the intensity decay through emission by a straight grating allow the selection of coupling gap width and grating strength to achieve critical coupling of the resonator. The theory predicts the emitted electrical field to be purely circularly polarized on-axis perpendicular to the waveguide plane if the azimuthal mode index of the guided mode differs from the number of grating elements by 1. Both methods confirm this, already an unoptimized structure reaches a purity of at least 99.95%.

Contents

Chapter 1

Introduction

1.1 Motivation

Great research effort is put into exploiting quantum mechanical effects for simulation and calculation purposes. One approach which is followed by this research group is to take transitions in single ions as qubits, allowing for simple binary operations. A detailed review on quantum computing with trapped ions can be found in reference [1]. In order to manipulate the single ions in an isolated and controlled environment, they are held in an ion trap, which can be achieved for example by a linear Paul trap.[2] Since the atomic transitions are induced by laser light, the delivery of optical power to the exact trapping position of the ion is crucial. Here the techniques of integrated optics come into play. Integrated optics devices to address single ions in a surface electrode trap have recently been demonstrated and promise to allow for better large-scale quantum information processing.[3] This research group is working on setting up a new ion trap where waveguide structures are integrated into the wafer which contains the electrodes that generate the trapping potentials.

For a controlled interaction of light with the trapped atom, the spin and orbital angular momentum of each photon has to be well-defined. The spin angular momentum of a photon is defined by its state of polarization, so each photon carries a spin of $\pm\hbar$ depending on whether it is left- or right-handedly circularly polarized. Since linear polarization is a superposition of left- and right-handed circular polarization, its photons will be in a superposition of both spin states. To obtain a pure spin state, one thus has to use light with a pure state of polarization. Linearly polarized light can be obtained for example by Brewster-angled reflection. Also from integrated optics devices, it is possible to obtain linearly polarized light with more than 99.9% purity at the focus.[4] For some operations however, it is necessary to use light with pure circular polarization. For example during readout, light impurities due to photons with opposite spin than desired may couple the atom to a different state as intended, which then decays into a dark state.[5] Circularly polarized light is usually created by phase-shifting one component of linearly polarized light by $\frac{\pi}{2}$ in a quarter-wave plate. However, Cai et al. demonstrated the emission of optical vortex beams carrying orbital angular momentum from an integrated optics device.[6] Under certain circumstances, the radiation on-axis from such emitters corresponds to pure circular polarization, which we try to exploit. In this project, the radiation emitted from this ring resonator with azimuthal grating is simulated and analyzed with respect to circular polarization at the center.

The ion used for this experiment is the hydrogen-like $^{40}Ca^{+}$, whose upper state has a long lifetime of about 1 s.[7] The $S_{1/2}$ to $D_{5/2}$ transition is induced by 729 nm light, which is readily

available with current laser technology. This is why this simulation focuses on light of this wavelength.

1.2 Propagation of light

From Maxwell's equations in a source-free, non-magnetic medium follows that each of the components of the electric and magnetic field $\vec{E}$ and $\vec{H}$ has to satisfy the wave equation

$$\vec{\nabla}^2 u - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0, \tag{1.1}$$

where u is the scalar wavefunction, $c = \frac{1}{\sqrt{\epsilon\mu_0}} = \frac{c_0}{n}$ is the speed of light in the medium and $n = \sqrt{\frac{\epsilon}{\epsilon_0}}$ is the refractive index of that medium. The wavefunction $u(\vec{r}, t)$ is real, however it is convenient to use a complex wavefunction $U(\vec{r}, t)$, so that

$$u(\vec{r}, t) = \mathrm{Re}\{U(\vec{r}, t)\}. \tag{1.2}$$

With the Ansatz

$$U(\vec{r}, t) = U(\vec{r}) \exp\left(i2\pi\nu t\right) \tag{1.3}$$

follows from the wave equation that the complex amplitude $U(\vec{r})$ satifies the Helmholtz equation:

$$\left(\vec{\nabla}^2 + k^2\right) U(\vec{r}) = 0. \tag{1.4}$$

Here, k is the angular wavenumber

$$k = \frac{2\pi\nu}{c} = \frac{\omega}{c} = \frac{2\pi}{\lambda}, \tag{1.5}$$

which should not be confused with the (linear) wavenumber $\tilde{\nu} = \frac{1}{\lambda}$, which is often used in Chemistry and spectroscopic applications. Note that k, c and λ are depending on the refractive index of the material, whereas ν and ω are not. With the complex amplitude, the optical intensity is given as

$$I(\vec{r}) = |U(\vec{r})|^2. \tag{1.6}$$

Two simple solutions to the Helmholtz equation are the plane wave

$$U(\vec{r}) = A \exp\left(-i\vec{k}\vec{r}\right) \tag{1.7}$$

and the spherical wave

$$U(\vec{r}) = A \exp\left(-ikr\right). \tag{1.8}$$

$r = |\vec{r}|$ is the distance from the origin and $\vec{k}$ is the wavevector so that $k = |\vec{k}|$.

1.3 Gaussian and Laguerre-Gaussian beams

If you consider a geometry where light propagates generally along one direction (usually z), one can make the approximation to express the complex amplitude as

$$U(\vec{r}) = A(\vec{r}) \exp(-ikz). \tag{1.9}$$

This so called paraxial wave is similar to a plane wave travelling along the z direction, but the carrier wave is modified by the complex envelope $A(\vec{r})$, which is a slowly varying function of position. From the Helmholtz equation then follows the paraxial Helmholtz equation

$$\vec{\nabla}_T^2 A - i2k\frac{\partial A}{\partial z} = 0, \tag{1.10}$$

where $\vec{\nabla}_T^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the transverse Laplace operator. One important solution to this equation is the Gaussian beam, which gives for the complex amplitude

$$U(\vec{r}) = \underbrace{A_0 \frac{W_0}{W(z)} \exp\left[-\frac{\rho^2}{W^2(z)}\right] \exp\left[-ik\frac{\rho^2}{2R(z)} + i\zeta(z)\right]}_{A(\vec{r})} \exp(-ikz) \tag{1.11}$$

with the beam width

$$W(z) = W_0\sqrt{1 + \left(\frac{z}{z_R}\right)^2}, \tag{1.12}$$

the radius of curvature

$$R(z) = z\left[1 + \left(\frac{z}{z_R}\right)^2\right], \tag{1.13}$$

the Gouy phase $\zeta(z) = \arctan\left(\frac{z}{z_R}\right)$ and the beam width at the waist $W_0 = \sqrt{\frac{\lambda z_R}{\pi}}$. The parameter z_R is called the Rayleigh range and together with the amplitude A_0 determines all parameters of the beam. z_R and A_0 themselves are fixed by the boundary conditions. ρ is the radius in cylindrical coordinates (ρ,ϕ,z), which are useful for paraxial problems. Further properties of Gaussian beams can be found in [8].

The Gaussian beam is not the only solution to the paraxial Helmholtz equation, but there are whole families of solutions. The most commonly-known family are the Hermite-Gaussian modes, which show the same change in phase as the Gaussian beam. This means the surfaces of constant phase, called the wavefronts, are identical. However, the transverse intensity distribution is different for each mode, given by Hermite-Gaussian functions along the x and y directions. These contain Hermite polynomials, so that each mode is described by a pair of indices (n, m) which stand for the order of the polynomials.

Another set of solutions are the Laguerre-Gaussian modes

$$\begin{aligned} U_{l,p}(\vec{r}) = &\frac{C_{l,p}}{W(z)}\left(\frac{\sqrt{2}\rho}{W(z)}\right)^{|l|} \exp\left[-\frac{\rho^2}{W^2(z)}\right] L_p^{|l|}\left[\frac{2\rho^2}{W^2(z)}\right] \times \\ &\exp\left[-ik\frac{\rho^2}{2R(z)} + i\zeta(z)\right] \exp(-il\phi) \exp(-ikz), \end{aligned} \tag{1.14}$$

where $W(z)$, $R(z)$ and $\zeta(z)$ are defined the same as above, $C_{l,p}$ is a normalization constant depending on the mode indices l and p and $L_p^{|l|}(\cdot)$ are the generalized Laguerre polynomials. There are some important features to note about the Laguerre-Gaussian beams. For $l = p = 0$, equation (1.14) reduces to the normal Gaussian beam as given by equation (1.11). This means that the transverse intensity profile is given by a Gaussian shape, the wavefronts are planar at the waist where $z = 0$ and spherical for larger distances. p gives the number of radial nodes within the beam profile. With increasing l, it becomes clear from the second term that the beam extends to larger radii. Also, the azimuthal phase factor $\exp(-il\phi)$ is introduced. This shapes the formerly planar or bent wavefronts to a helix, with the azimuthal mode index l giving the number of spiral arms. When following one round trip of the azimuthal angle ϕ, the light accumulates an additional phase of $2\pi l$. Figure 1.1 shows a schematic wavefront of such a beam with $l = 2$. The Matlab script to create this figure can be found in listing 6.1.

Figure 1.1: Scheme of the wavefront of a Laguerre-Gaussian beam with azimuthal mode index $l = 2$.

The Laguerre-Gaussian modes as well as the Hermite-Gaussian modes are both complete sets of solutions to the paraxial Helmholtz equation. This means that any solution can be expressed as superposition of modes from one of these sets of solutions. Consequently, also each Laguerre-Gaussian mode can be expressed as a complex-weighted superposition of Hermite-Gaussian modes and vice versa, as has been shown by [9]. One reason to use the Laguerre-Gaussian modes for certain problems is its convenient description of light's orbital angular momentum. The momentum that electromagnetic radiation has is split into linear and angular momentum. Angular momentum itself may have contributions from spin, which is associated with the polarization of the field vectors, and an orbital part, which comes from the field's spatial distribution. In short, a Laguerre-Gaussian mode with azimuthal mode index l carries orbital angular momentum of $L = l\hbar$ per photon.

One has to pay attention to where to apply the description with Laguerre or Hermite-Gaussian modes, though. They are a solution to the paraxial Helmholtz equation, which originates from the scalar wave equation. This means that one single mode alone is not sufficient to describe the vector nature of the electric field. Instead, each component of the field vector is individually described by one mode or a superposition of modes. To treat this, one multiplies the scalar wave with the unit vector of the polarization direction. Since the spatial distribution is totally included in the scalar wave function, the unit vectors may not be position-dependent. This is valid for most polarizations like linearly polarized light or circularly polarized light by superposition. However, for example for a radially or azimuthally polarized beam, the field vector points in different directions depending on position. Thus, it cannot be described by a single Gaussian mode alone. Instead, one has to use a superposition of Gaussian beams with different polarization directions, as will be done in section 2.2.

1.4 Guided-wave optics

For a basic introduction of the concept of waveguides, we begin here in the ray optics picture. This simplification yields useful results for waveguides with dimensions much larger than the wavelength of the guided light, which support multiple modes. For waveguides with smaller transverse dimensions like in this project, the approximations of ray optics are not sufficient anymore. Instead, one has to base the theory of guiding light on solving the wave equation (1.1) with position-dependent refractive index. Also, for the understanding of the principles of waveguides, the polarization of the electric field vector is left out. It is then discussed in section 2.1.

Dielectric waveguides are formed by a dielectric material which is transparent at the wavelength of interest, and a surrounding material with lower refractive index. For the basic principles, we consider a two-dimensional structure where z is the general direction of propagation of light and y is the second dimension, as can be seen in Figure 1.2.

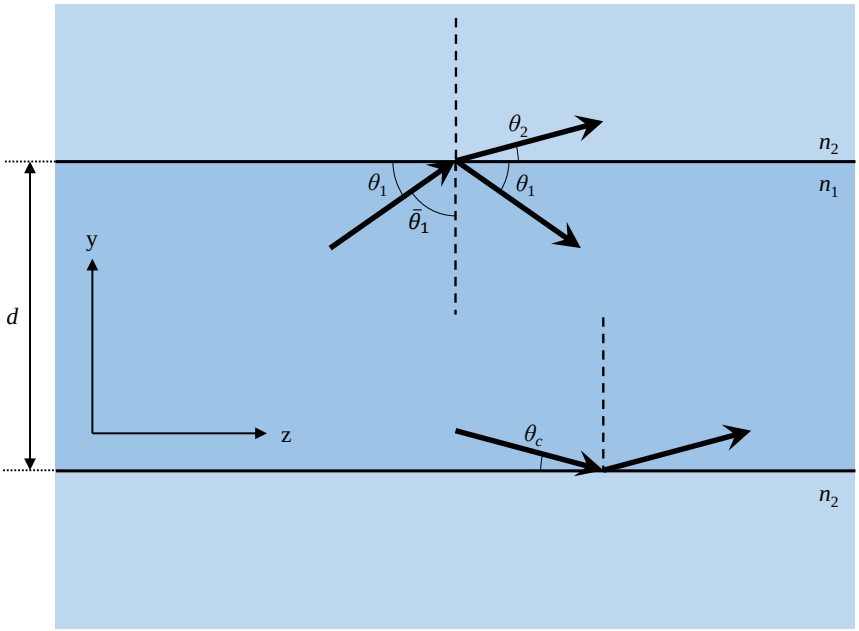


Figure 1.2: Two-dimensional scheme of a waveguide composed of the core material with refractive index n_1 (blue) and the cladding material with refractive index $n_2 < n_1$ (light blue). For incident angles θ_1 to the interface, the light is reflected a the same angle, and refracted at a smaller angle θ_2 governed by Snell's law (equation (1.15)). If the incident angle is small enough, this law cannot be fulfilled anymore and the light is completely reflected. The maximum angle at which this occurs is the critical angle θ_c.

The guiding material is called the "core" of the waveguide and has refractive index n_1, whereas the surrounding material is called the "cladding" of the waveguide and has refractive index $n_2 < n_1$. Darker colors denote higher refractive index. Rays hitting the waveguide interface between core and cladding at an incident angle θ_1 are partially reflected at the same angle. The other part is transmitted but refracted at an angle θ_2 governed by the refractive indices through Snell's law:

$$n_1 \sin\bar{\theta}_1 = n_2 \sin\bar{\theta}_2, \tag{1.15}$$

where $\bar{\theta}_{1/2} = 90° - \theta_{1/2}$ are the complementary angles as drawn in Figure 1.2. Decreasing the incident angle θ_1 with respect to the interface means increasing the angle $\bar{\theta}_1$. This also increases the refracted angle $\bar{\theta}_2$. However, its maximum value is 90°, so that $\sin\bar{\theta}_2 = 1$. Then, the incident angle θ_1 becomes the critical angle θ_c:

$$\begin{aligned} \theta_1 = \theta_c &= 90° - \bar{\theta}_c \\ &= 90° - \arcsin\left(\frac{n_2}{n_1}\right) \\ &= \arccos\left(\frac{n_2}{n_1}\right). \end{aligned} \tag{1.16}$$

For all incident angles $\theta_1 \leq \theta_c$, Snell's law can not be satisfied and there is no refracted beam. Instead, all the light is reflected, which is called total internal reflection. For parallel waveguide interfaces, the reflected beam will hit the opposite interface at the same angle, so that it is totally reflected again. This means that the light can propagate along the waveguide without losing any power, provided that the material losses are negligible.

In this ray optics picture, it is clear why it is favourable to guide light making small angles with respect to the waveguide boundaries. However, there are more limitations introduced by the wave nature of light. The interested reader is referred to [8] for a more detailed discussion and proper visualization of the introduced variables. In brief: For a twice-reflected wave to propagate along the waveguide, it has to be consistent in phase with the original wave. This imposes a condition on the incident angles, so that they can only take specific values θ_m, called bounce angles. For a dielectric waveguide, they are not as straight-forward to obtain as for mirror waveguides because the phase-shift φ_r introduced at the boundary is not π anymore.

A guided wave then consists of two plane waves with angles $\pm\theta_m$ with respect to the z axis which is set parallel to the waveguide boundary. Both waves have the same z dependency, so that any superposition takes the form $E_m \propto e^{-i\beta_m z}$, where

$$\beta_m = n_1 k_0 \cos\theta_m \tag{1.17}$$

is called the propagation constant and $k_0 = \dfrac{2\pi}{\lambda_0}$ is the angular wavenumber of the unguided wave in vacuum. Since over longer distances, the wave effectively travels in the z direction, one also defines the effective refractive index

$$n_{eff} = n_1 \cos\theta_m \tag{1.18}$$

so that $\beta_m = n_{eff} k_0$. The effective refractive index takes the role of the material refractive index if one neglects the details of how the wave is guided in the structure. However, both should not be confused. Additionally, the effective refractive index is not only dependent on both the core and the cladding material, but also on the mode m in which the light propagates.

If several waves with different bounce angles are present, the resulting field is a superposition of these modes:

$$E(y,z) = \sum_m a_m u_m(y) e^{-i\beta_m z}, \tag{1.19}$$

where a_m is the amplitude of the mth mode and $u_m(y)$ is its transverse field profile. These can become more complex depending on the exact shape of the waveguide and are later calculated by a solver for waveguide modes. For this two-dimensional waveguide with parallel interfaces, the transverse field profile inside the waveguide ($|y| \leq \frac{d}{2}$) is given by

$$u_m(y) \propto \begin{cases} \cos\left(\frac{2\pi \sin\beta_m}{\lambda} y\right), & \text{for } m = 0, 2, 4, ... \\ \sin\left(\frac{2\pi \sin\beta_m}{\lambda} y\right), & \text{for } m = 1, 3, 5, ... \end{cases} \tag{1.20}$$

with $\lambda = \frac{\lambda_0}{n_1}$. For a waveguide with mirror interfaces, the field is zero at the interfaces. As becomes clear from equation 1.20, this is not the case for dielectric waveguides. Outside the guiding layer ($|y| \geq \frac{d}{2}$), the field decreases exponentially, so it takes the form

$$u_m(y) \propto e^{-\gamma_m |y|}. \tag{1.21}$$

γ_m is the extinction coefficient, which decreases with m. This means that higher order modes extend deeper into the surrounding material.

This reasoning can be applied analogously to a second transverse direction, thus introducing a second mode index. For our problem, we assume only one contributing mode, which is the fundamental mode with both indices zero.

1.5 Ring resonators

An optical resonator is a system in which light circulates and is fed back to its original state without escaping. A simple setup is the Fabry-Perot etalon which is comprised of two opposing mirrors. Other setups may also form a resonator, for example multiple mirrors arranged so that rays follow a closed geometrical shape, or an optical fiber whose ends meet each other. Although there are many ways to obtain a resonator, the underlying principles are similar. There is the optical path length L which light rays travel until they reach their initial position again, and an attenuation factor α which accounts for losses during one round-trip. Such losses could originate for example from imperfect mirrors, which also transmit part of the light, or objects which scatter some part into a spatial direction where it can escape. Although this occurs only at specific positions in the resonator, one usually 'distributes' the losses mathematically so that they are averaged over one round-trip.

Optical resonators have resonant modes at particular frequencies. Additionally to spatial feedback, light also has to have its original phase when it reaches its initial position again. Otherwise, the light wave interferes destructively with itself so that these components are cancelled out. This means that only electric fields with wavelengths by which the optical path length is divisible without remainder are self-consistent and can exist in the resonator. In a circular resonator, such electric fields are called whispering gallery modes (WGMs).

Figure 1.3 shows a scheme of the coupling region at the gap between the feeding waveguide and the ring resonator. As discussed in section 1.4, the electric field in a dielectric waveguide is not totally confined to the core. It also extends into the surrounding material, where it is unguided and loses amplitude quickly (see equation (1.21)). However, in the coupling region, the gap between ring and feeding waveguide is close enough so that the electric field extends into the ring. There, it is guided again and does not lose amplitude by propagation. The different electric fields at the coupling region are related by equation (1.22)

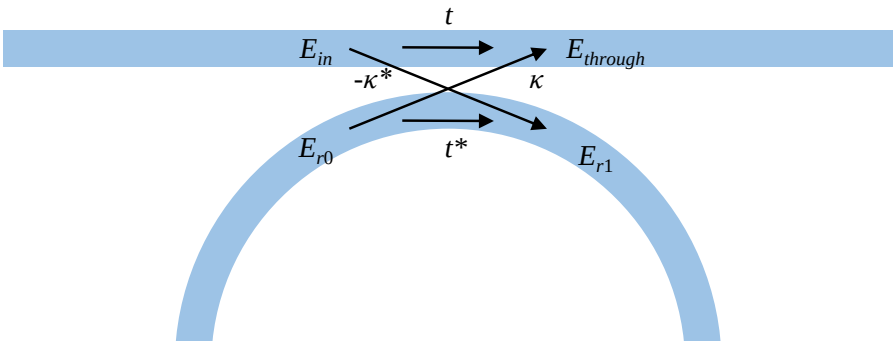


Figure 1.3: Scheme of the coupling region between feeding waveguide and ring resonator. The electric fields before and after the coupling are related to each other via the transmission coefficient t and the coupling coefficient κ, according to equation (1.22).

$$\begin{pmatrix} E_{through} \\ E_{r1} \end{pmatrix} = \begin{pmatrix} t & \kappa \\ -\kappa^* & t^* \end{pmatrix} \begin{pmatrix} E_{in} \\ E_{r0} \end{pmatrix}, \tag{1.22}$$

where E_{in} and $E_{through}$ are the electric fields in the feeding waveguide before and after the coupling, E_{r0} and E_{r1} are the electric fields in the ring before and after the coupling, t is the transmission coefficient and κ is the coupling coefficient.[10] Energy conservation requires that the transmission matrix is unitary, so that $|t|^2 + |\kappa|^2 = 1$. E_{r0} follows only from E_{r1} due to propagation of the electric field in the ring,

$$E_{r0} = E_{r1} e^{-\frac{\alpha}{2} 2\pi R} e^{i\phi} \tag{1.23}$$

where α is the intensity decay rate due to loss in the resonator, R is the radius in the ring and ϕ is the phase shift introduced in one round-trip. The transmitted field $E_{through}$ can thus be expressed in terms of the input field E_{in}:

$$E_{through} = E_{in} \frac{te^{-i\phi} - e^{-\alpha\pi R}}{e^{-i\phi} - t^* e^{-\alpha\pi R}}. \tag{1.24}$$

At resonance, there must be constructive interference between the field $t^* E_{r0}$ which propagates in the ring and the field E_{r1} added by the feeding waveguide. This means $\arg\{t^* E_{r0}\} = \arg\{E_{r1}\}$ and thus $\phi = \varphi_t$ up to a shift of a multiple of 2π, where φ_t is the phase of t. Then, equation (1.24) reduces to

$$E_{through} = E_{in} e^{i\varphi_t} \frac{|t| - e^{-\alpha\pi R}}{1 - |t| e^{-\alpha\pi R}}. \tag{1.25}$$

Looking at the numerator, it follows that the transmitted field becomes zero for

$$|t| = e^{-\alpha\pi R}, \tag{1.26}$$

which is the critical coupling condition. For these pairs of parameters $|t|$ and α, the field in the feeding waveguide interferes destructively with the field coupled back from the ring. This means all the input power is transferred into the ring, where it is lost through whatever mechanism causes the intensity decay rate α.

For this project, critical coupling is the optimal state of operation because the loss mechanism is emission of radiation, which should be maximized. The transmission coefficient t is determined by the coupling strength, which can be influenced by the width of the gap between feeding waveguide and ring resonator. The loss rate depends on the grating strength, where parameters like grating periodicity and shape are to be determined.

Chapter 2

Principle and simulation approach

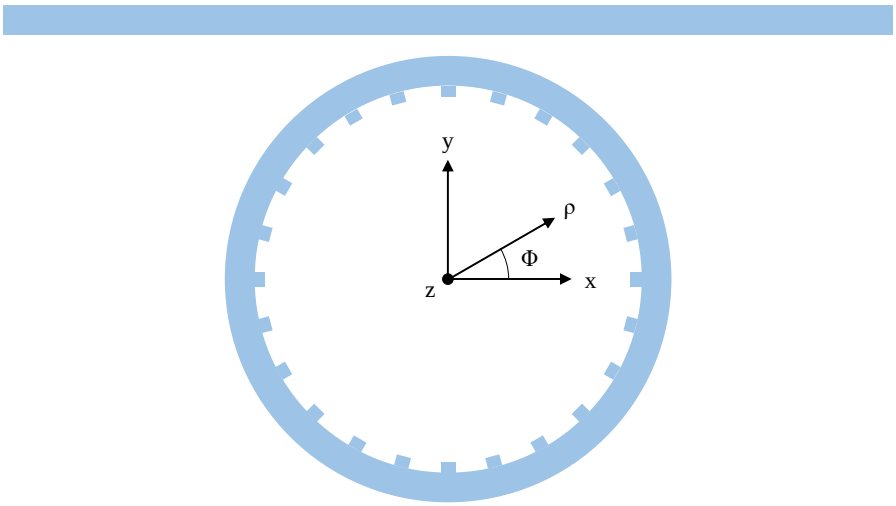


Figure 2.1: Scheme of a ring resonator with a feeding waveguide. On the inner boundary of the ring, grating elements are added to couple radiation out of the ring. Radiation is mainly emitted into the z direction due to the circular symmetry of the emission grating.

A ring resonator with an angular grating as in Figure 2.1 theoretically allows the emission of almost purely circularly polarized light. This chapter is devoted to explaining why this can be achieved by matching the azimuthal mode index m to the number of grating elements N by a difference of 1.

2.1 Propagation of light in the ring resonator and emission by the diffraction grating

Circular resonators support whispering gallery modes (WGMs). One uses angular propagation constants $\nu_{WGM} = \beta R$, which give the phase shift per unit azimuthal angle. R is the effective radius of the WGM and β is the propagation constant of the guided mode (see section 1.4). On resonance, the phase shift for one round trip has to be a multiple of 2π, so that $\nu_{WGM} = \beta_m R =$

m. m is an integer which is called the azimuthal mode number and denotes the number of cycles that the light goes through when propagating one round-trip through the resonator. The electric field in the ring waveguide then takes the form $E_0 \propto e^{im\phi}$.

At the grating elements, a part of the light is reflected and removed from the guided mode. According to Huygen's principle, each point where this occurs is the source of a new spherical wave. Each grating element can be regarded as quasi-point source for such a spherical wave. There are different phase shifts between these waves from neighboring grating elements due to different path lengths up to a given point in space. If the phase shift is an integer multiple of 2π, constructive interference occurs and light is radiated into this spatial direction. Oppositely, for uneven integer multiples of π, destructive interference is observed.

In a linear grating, if the grating elements are uniformly spaced, the phase shift is the same between each neighboring spherical wave. This means that for a certain angle of emission, there is a plane on which the light has the same phase at every point, called wavefront. Note that in principle, there can be more than one wavefront, if the angle is high enough so that the phase difference is a multiple of 2π, which is called the diffraction order. This depends on the effective wavelength in the waveguide and in the propagation medium, as well as the spacing of the grating elements. For the parameters in this project, only first order diffraction is possible.

If the grating is not linear anymore, but formed into a ring, these wavefronts are bent. Due to the circular symmetry of the problem, the wavefront does not show a discontinuity at any point, but rather forms a helical structure. Beams with such wavefronts are described by an azimuthal dependence of $E_{rad} \propto e^{il\phi}$, where l is an integer. Similar to the Laguerre-Gaussian modes described in section 1.3, they carry an orbital angular momentum of $l\hbar$ per photon. The azimuthal dependence manifests itself in a helical wavefront, similar to the scheme for $l = 2$ in Figure 1.1.

To derive the form of the emitted electric field more rigorously, a coupled mode analysis goes as follows: Which radiation modes are excited by a waveguide mode is determined by their coupling coefficient κ. It is given by

$$\kappa \propto \int_V \vec{E}_{rad} \Delta\epsilon \vec{E}_0 dV, \tag{2.1}$$

which is an overlap integral of the unperturbed field in the waveguide $\vec{E}_0$ with the radiated field $\vec{E}_{rad}$, mediated by the dielectric perturbation $\Delta\epsilon$. The angularly dependent parts are evaluated separately, and $\Delta\epsilon$ is expanded in a Fourier series. Considering only the first term, it is given by $\Delta\epsilon \propto \sin(N\phi)$, where N is the number of grating elements. Then, each of the electric field vector components contributes to the coupling coefficient like the following:

$$\kappa \propto \int_0^{2\pi} e^i(m + N - l) + e^i(m - N - l) d\phi. \tag{2.2}$$

For this integral to be non-zero, one of the exponents has to become zero. This gives the condition $l = m \pm N$. Since the radius of the radiation modes increases with l, a small structure like the azimuthal grating only overlaps with modes with small values of l. m and N are both numbers in the order of 10 to 100, so only the difference condition produces significant overlap. Thus, the azimuthal mode index l of the radiation is given by the difference between the azimuthal mode index m and the number of grating elements N:

$$l = m - N. \tag{2.3}$$

Then, the propagation constant of the radiation in z direction $\beta_{rad,z}$ is given by

$$\beta_{rad,z} = \sqrt{(\beta_{rad})^2 - (\beta_{rad,\phi})^2}$$
$$= \sqrt{\left(\frac{2\pi n_{surr}}{\lambda}\right)^2 - (\nu_{rad} R)^2}, \tag{2.4}$$

where n_{surr} is the refractive index off the medium surrounding the waveguide and λ is the free-space wavelength. Because of the circular symmetry, the general propagation is along the z direction.

Now we consider the state of polarization of light inside the waveguide. The geometry of the waveguides in this study is close to the one of a planar dielectric waveguide. The modes in this waveguide are quasi-TE modes, so that the electric field vector lies within the waveguide plane. As have been shown by simulations,[6] in the center of the waveguide where the intensity of the guided mode is highest, the electric field vector is transverse, which corresponds to radial polarization for a ring waveguide. In contrast, at the inner sidewall of the waveguide, which is where the grating is placed, the electric field vector has both transverse and longitudinal components. In fact, due to strong confinement, the longitudinal components at the waveguide boundary can be of comparable strength as the transverse components.[11] From the perspective of a ring, this means a mixture of radial and azimuthal polarization. As will be shown in the following section, the exact emitter state of polarization is not important for the reasoning to obtain purely circularly polarized light at one point. However, it does determine the overall ratio of the amplitudes of left-handed and right-handed circular polarization.

2.2 Circular polarization of the emitted field

There are different conventions about the terminology of circular polarization. Here we use the unit vectors $\hat{\sigma}_+ = \frac{1}{\sqrt{2}}(\hat{x} + i\hat{y})$ for left-handed circular polarization (LHCP) and $\hat{\sigma}_- = \frac{1}{\sqrt{2}}(\hat{x} - i\hat{y})$ for right-handed circular polarization (RHCP).

From the reasoning in the previous section follows that the transverse components of the radiated field are

$$\vec{E}_{rad,transverse} = a_r(r,z)e^{il\phi}\hat{r} + a_\phi(r,z)e^{il\phi}\hat{\phi}, \tag{2.5}$$

where $\hat{r} = \cos\phi\hat{x} + \sin\phi\hat{y}$ and $\hat{\phi} = -\sin\phi\hat{x} + \cos\phi\hat{y}$ are the unit vectors in radial and azimuthal direction, respectively. $a_r(r,z)$ and $a_\phi(r,z)$ are amplitudes determining the ratio of radial to azimuthal polarization. Due to circular symmetry, they are independent of angle ϕ, but in principle they do depend on position. The radiated field in principle also has a longitudinal component along the z direction which will be discussed in section 2.3.

The transverse components of the radiated field $E_{rad,transverse}$ can also be expressed in the basis of circular polarization:

$$\vec{E}_{rad,transverse} = E_+\hat{\sigma}_+ + E_-\hat{\sigma}_-. \tag{2.6}$$

$\hat{\sigma}_+$ and $\hat{\sigma}_-$ are the basis vectors for left-handed and right-handed circularly polarized light, respectively, E_+ and E_- are the corresponding amplitudes. Then, by projection onto the basis vectors of circular polarization, E_+ and E_- are given by equations (2.7) and (2.8).

$$
\begin{aligned}
E_+ &= e^{il\phi}\left(a_r(r,z)\hat{r}\cdot\hat{\sigma}_+ + a_\phi(r,z)\hat{\phi}\cdot\hat{\sigma}_+\right) \\
&= e^{il\phi}\left(\frac{a_r(r,z)}{\sqrt{2}}e^{i\phi} + \frac{a_\phi(r,z)}{\sqrt{2}}ie^{i\phi}\right) \\
&= \tilde{a}_+(r,z)e^{i(l+1)\phi}
\end{aligned} \tag{2.7}
$$

$$
\begin{aligned}
E_- &= e^{il\phi}\left(a_r(r,z)\hat{r}\cdot\hat{\sigma}_- + a_\phi(r,z)\hat{\phi}\cdot\hat{\sigma}_-\right) \\
&= e^{il\phi}\left(\frac{a_r(r,z)}{\sqrt{2}}e^{-i\phi} + \frac{a_\phi(r,z)}{\sqrt{2}}ie^{-i\phi}\right) \\
&= \tilde{a}_-(r,z)e^{i(l-1)\phi}
\end{aligned} \tag{2.8}
$$

where $\tilde{a}_\pm(r,z) = \frac{a_r(r,z)\pm ia_\phi(r,z)}{\sqrt{2}}$. Although $\tilde{a}(r,z)$ is complex, it does not vary with angle ϕ or with time, so its phase stays constant. It is important to note that the amplitudes of left- and right-handed circularly polarized light do not have to be the same, but are determined by the relative amplitudes and phases of the radial and azimuthal components in the waveguide.

We now consider the case where $l = 1$. E_- becomes independent of angle ϕ. For E_+, the angular dependence is given by $e^{i2\phi}$. With non-zero intensity, there would be a discontinuity at $r = 0$. Because this is unphysical, the intensity at the center of the r-ϕ plane must vanish. This applies more generally, so that all Laguerre-Gaussian modes with $l \neq 0$ have zero intensity on axis. In this case, this means that the $\hat{\sigma}_+$ component vanishes at this point. Instead, all contributions which are not in the z direction, must be purely right-handedly polarized. The analogous argument leads to the conclusion that for $l = -1$, the transverse components of the light must be purely left-handedly polarized at the center of the r-ϕ plane.

2.3 Longitudinal component of the emitted field

The different components of the electric field are related to each other via Gauss's law in the absence of electric charges:

$$
\vec{\nabla}\cdot\vec{E} = \frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} + \frac{\partial E_z}{\partial z} = 0. \tag{2.9}
$$

A longitudinal component E_z can then only be present if the electric field varies strongly along the x- and/or y-direction. The point of our interest is at the center of the r-ϕ plane where only one circularly polarized component contributes to the field at this point, as has been discussed in the previous section. This component does not vary with angle ϕ but only with radius r. The trapped ion will be several tens of micrometers above the waveguide plane. For ring dimensions of 10 to 20 micrometer diameter, this is not in the near-field regime anymore but rather in the far field, which indicates a more slowly amplitude. From this follows that we have reason to expect the z component to be small, possibly negligible. To which extent it contributes is calculated in more detail in the 2D and 3D simulations and will be discussed in chapter 3.

2.4 Volume Current Method

The Volume Current Method (VCM) is a method to calculate the radiation loss of dielectric waveguides.[12] Initially it was intended to allow the minimization of radiation from structures

like bends, branches or tapers, but here it is applied to obtain the radiated field from the angular grating at the inner boundary of the ring waveguide, as can be seen in Figure 2.1.

2.4.1 Theoretical background

Radiating structures are modeled by treating them as perturbation $\delta\epsilon\left(\vec{r}\right) = \epsilon_1\left(\vec{r}\right) - \epsilon_2$ to the surrounding dielectric constant ϵ_2. Then, the polarization current density is defined as

$$\vec{J}_P = \frac{\partial \vec{P}}{\partial t} = i\omega\delta\epsilon\left(\vec{r}\right)\vec{E}\left(\vec{r}\right), \tag{2.10}$$

where $\vec{E}\left(\vec{r}\right)$ is the electric field in the waveguide, which is usually unknown. However, it can be approximated by the field of the mode in the waveguide, which is obtained from numerical solutions or from waveguide theory as in section 1.4.

In Maxwell's equations, the electric and magnetic fields can be expressed in terms of the magnetic vector potential $\vec{A}$, so that

$$\vec{B} = \vec{\nabla} \times \vec{A} \tag{2.11}$$

and

$$\vec{E} = -\vec{\nabla}\varphi - \frac{\partial \vec{A}}{\partial t}, \tag{2.12}$$

where φ is the scalar potential. Using the Lorentz gauge $\vec{\nabla} \cdot \vec{A} + \frac{1}{c^2}\frac{\partial \varphi}{\partial t} = 0$ and assuming a time-dependency of $\vec{A} \propto e^{i\omega t}$, the electric field takes the form

$$\vec{E} = -i\omega\vec{A} + \frac{\vec{\nabla}\left(\vec{\nabla} \cdot \vec{A}\right)}{i\omega\mu_0\epsilon_2}, \tag{2.13}$$

Also, then the Helmholtz equation in a dielectric medium assuming no free or magnetic current density is given by

$$\left(\vec{\nabla}^2 + \omega^2\mu_0\epsilon_2\right)\vec{A} = -\mu_0\vec{J}_P. \tag{2.14}$$

The solving vector potential $\vec{A}$ is a superposition of spherical waves caused by the polarization current density:

$$\vec{A}\left(\vec{r}\right) = \frac{\mu_0}{4\pi}\int_V \vec{J}_P\left(\vec{r}'\right)\frac{\exp\left(-ik_2|\vec{r} - \vec{r}'|\right)}{|\vec{r} - \vec{r}'|}dV', \tag{2.15}$$

where k_2 is the angular wavenumber in the surrounding material. Note that although the integral goes over the whole space, it only needs to be evaluated over the waveguide structure because outside, $\delta\epsilon\left(\vec{r}\right)$ and thus $\vec{J}_P\left(\vec{r}\right)$ is zero.

Thus, the procedure is as follows: One models the structure of interest with a perturbance $\delta\epsilon\left(\vec{r}\right)$ to the surrounding refractive index, and together with an assumption for the electric field in the waveguide obtains the polarization current density $\vec{J}_P$ according to equation (2.10). Integration over the waveguide structure according to equation (2.15) gives the vector potential $\vec{A}\left(\vec{r}\right)$, so that equation (2.13) yields the radiated field $\vec{E}\left(\vec{r}\right)$ at any point in space $\vec{r}$.

2.4.2 Modeling of the ring resonator with azimuthal grating

The most important assumption using the volume current method in this project is that the waveguide structure is treated as only two-dimensional. As mentioned earlier in section 2.3, we are interested in the radiated field several tens of micrometers above the waveguide plane. So for a simplified treatment, the extent of the waveguide into this direction, which we denote as the z direction, is neglected. Instead, this allows for less a less detailed description of the waveguide in term of its dielectric constant and additionally simplifies the integral in equation (2.15). Mathematically, the polarization current density $\vec{J}_P(\vec{r})$, also called volume current, is replaced by $\vec{K}(\rho,\phi)\,\delta(z)$, a surface current times the Dirac delta distribution along z. The two-dimensional treatment also means that any cover above the dielectric which is part of the waveguide is neglected here. Instead, the surrounding medium in which the radiation propagates is assumed to be vacuum, which is a good approximation since any cover in the real structure would only be a few micrometers thick.

For the volume current method, only the radiating structures are modeled. This means that the feeding waveguide is left out completely, and of the ring resonator only the grating is included. After setting the parameters of the ring, which are radius and width, the grating is approximated as $\delta\epsilon(r,\phi) \propto \epsilon_0 \sin(N\phi)$ at the radii where the ring is and to zero elsewhere. So there is no grating at the inside of a ring waveguide, but only a grating in which there is still the electric field of a guided mode. This might result in a deviation in the radial distribution of the electric field, but not at the center due to the exactly circular symmetry of the model. Instead of the rectangular shape of the grating elements, they can be described well in this sinusoidal form because in the Fourier decomposition of the dielectric perturbance for the derivation of equation (2.3), it is shown that only the fundamental component can couple to the radiation modes.[6] Although this depends on parameters like the free-space wavelength of interest, the refractive index of the surrounding, the waveguide refractive index and the grating periodicity, the condition is fulfilled for the parameters in this project, too.

The electric field $\vec{E}(r,\phi)$ in the waveguide has an angular dependence of $E \propto e^{im\phi}$, as pointed out in section 2.1. For the radial dependence, one has to implement the transverse field profile inside a dielectric waveguide, as introduced in equation (1.20). For a first simple description, it is not taken into account that there may be multiple modes propagating. Also, instead of the exact profile, the simple sinusoidal form $u(r) = \cos\left(\frac{\pi}{W}(r-R)\right)$ is chosen, where R is the ring radius and W is the ring width. Similar to the argument for $\delta\epsilon(r,\phi)$, this can only have an effect on the radial distribution of the radiated field due to the circular symmetry of the deviation. The polarization of the electric field in the waveguide was modeled to be radial. For a further discussion, see section 2.1.

With these assumptions, first calculations can be made. However, in order to speed up the computation time to obtain the electric field from the vector potential, equation (2.15) is inserted into equation (2.13). Then, the divergence of the vector potential and the gradient of the resulting scalar field are evaluated analytically. For these spherical waves, this is easily possible and saves expensive and inaccurate numerical derivations. In the Appendix in section 6.1.1, a Mathematica notebook can be found which evaluates all necessary derivatives. These analytical formulas were then used for the numerical calculations.

Numerical calculations were performed with MATLAB R2017a, the analytical derivations were evaluated with Wolfram Mathematica 11.2.

2.5 Finite-Difference Time-Domain Method

The finite-difference time-domain method is based on solving Maxwell's equations in the time-domain numerically until a steady state is reached. From the resulting electric and magnetic fields for each time, their frequency components are obtained via Fourier transformation.

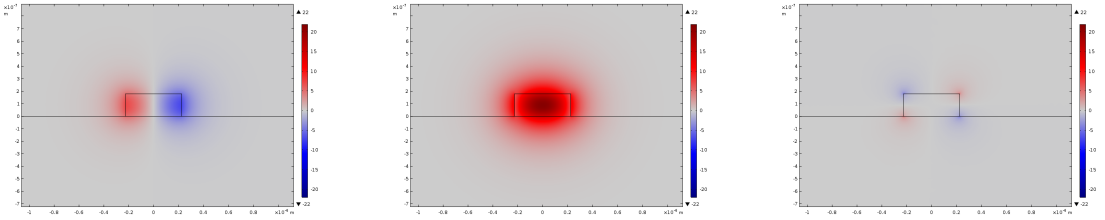
For the numerical solution, Maxwell's equations are replaced by a set of finite-difference equations. This approximates the derivatives in the differential equations. Since both space and time are discretized, one has to chose the spacing between the points on the numerical grid as well as the interval at which time is sampled. Usually the structure of interest determines the grid spacing, also called the mesh size. Then the time interval is derived from this via the speed of light in the medium of interest because it takes a certain minimum time for light to propagate through a mesh cell. Before running the calculations, also the boundary conditions have to be set. This includes the electric and magnetic fields at every point of the mesh grid at the starting time, and how the fields are treated when they reach the limits of the computational domain. Usually the boundaries are set to be absorbing, so the energy present within the domain decreases continuously until it reaches a certain threshold which will terminate the calculations. The finite-difference equations are solved in a leap-frog manner, so at one point in time, the equations are solved for the electric fields at all points in space. At the next point in time, these electric fields are used to calculate the magnetic fields at all points in space. This alternating procedure is repeated and thus the fields propagate in time.

There are commercial solutions which are designed to solve this problem, for this project the program lumerical FDTD Solutions release 2017b was used. The geometry of the ring resonator was inputted and the boundary conditions were all set to absorbing. The initial electric and magnetic fields are distributed according a guided mode in the feeding waveguide, which was calculated with the integrated mode solver. Additionally, there are fields following up, so that at the feeding waveguide, there is a pulse input from the simulation boundary. This pulse is very short, typically in the femtosecond range. This is necessary for a feasible computational effort but also advantageous. Owing to the inverse relation of pulse duration and bandwidth, the fields cover a wide range of frequencies. Thus, the response of the structure to many different frequency components is simulated at one go. This is one large advantage of simulations in the time-domain over the frequency domain. Moreover, this is a necessary feature for this project, because due to the perturbance of the ring waveguide by its bend and by the emission grating, an accurate estimation of its resonance frequencies is difficult.

The structure is modeled as follows: The waveguides are formed by Silicon nitride with a refractive index of 2.007, surrounded by Silicon dioxide with a refractive index of 1.454. Between the waveguide and the air, the layer of Silicon dioxide is 1 μm thick. The feeding as well as the ring waveguide are 450 nm wide and 180 nm in height and an initial ring radius of 10 μm was chosen. This profile yields an effective refractive index for the waveguide of 1.67. Figures 2.2a to 2.2b show the x, y and z components of the mode at the input port, which lies in the y-z plane.

The gap width and the width that each grating element adds to the waveguide, here called grating tooth height, have to be matched to achieve critical coupling (see section 1.4). Rough initial values are 100 nm and 50 nm, respectively. The number of grating teeth is set with respect to the azimuthal mode number m according to equation (2.3). Since the desired wavelength of the emitted radiation is 729 nm, the azimuthal mode number can be estimated via the effective waveguide index. Then the number of grating elements is matched to this number. The grating periodicity is chosen to match the effective wavelength of in the waveguide. The length of each grating element is determined by the duty cycle, which is chosen to be 0.5 in order to maximize the radiation. One can choose between different settings for the mesh accuracy. Here, the setting 3 was selected because it was the highest that could be handled with the installed amount of

memory on the computer which performed the calculations. It corresponds to a mesh grid size of about 25 nm in the waveguide. The mesh is non-uniform and adapted to the refractive index of the corresponding medium, and changes gradually at the interfaces. The electric fields are monitored throughout the simulation in the waveguide plane, and in the air above. From the monitor in the air, the diffracted field will be calculated.



(a) x component (longitudinal) (b) y component (transverse) (c) z component (transverse)

Figure 2.2: Electric field components of the waveguide mode at the input port, plotted is a cut through the y-z plane. At the center of the waveguide, the transverse y component is clearly dominant. At the interfaces, also the other transverse component in z direction and the longitudinal component in x direction are present.

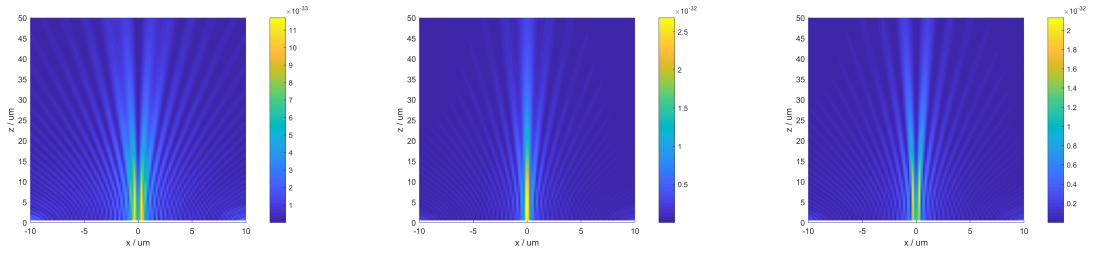
Chapter 3

Results and Discussion

3.1 Volume Current Method

The Matlab scripts used to implement the volume current method can be found in listings 6.2 to 6.6 in section 6.1.2.1 in the Appendix.

3.1.1 Angular intensity distribution



(a) $l = 2$ (b) $l = 1$ (c) $l = 0$

Figure 3.1: Magnitude squared of the vector potential $\vec{A}(\vec{r})$ in the x-z plane at the venter of the ring. The ring has a radius of 10 μm, so the waveguide is located at $x = \pm 10\ \mu m$.

Although the vector potential itself is not a measurable quantity, it is used here to give an intuition about the angular distribution of the electric field. As discussed in section 2.3, the second term of equation (2.13) is expected to be small, so the electric field is approximately proportional to the vector potential. Figures 3.1a to 3.1c show the magnitude squared of the vector potential $\vec{A}(\vec{r})$ in the x-z plane, the ring resonator is placed in the x-y plane. The Figures show $\vec{A}(\vec{r})$ for different values of l, the azimuthal phase variation of the radiated field.

For $l = 1$, there is intensity on-axis, while for $l = 2$ or $l = 0$, there is not. This matches the reasoning based on equations (2.7) and (2.8) in section 2.2 that only for $l = \pm 1$, the field at the center of a plane transverse to the propagation direction is non-vanishing. In addition to a central lobe, there are regularly spaced lobes at higher angles with respect to the propagation axis, which come from the repeated local maxima of the Laguerre polynomials which describe the transverse field distribution of the Laguerre Gaussian beams. The radius of the ring resonator here is 10 μm, which manifests itself in the increase in intensity in the near field at $x = \pm 10 \mu$m.

The maximum of the vector potential is close to the waveguide plane, but still at a certain distance. It is conceivable that this can be influenced by parameters like ring radius and grating width.

3.1.2 Electric field in the observer plane

Table 3.1 shows the components of the simulated electric field in a plane 50 μm above the waveguide plane for different values of the azimuthal mode index l. Yellow and blue stand for positive and negative values, respectively. Green denotes an electric field close to zero.

Of the circularly polarized components, the most striking feature is that for $l = 1$ the right-handed, and for $l = -1$ the left-handed component is circularly symmetric. This follows from the reasoning in section 2.2 that for those components, the dependence of the electric field on the azimuthal angle ϕ vanishes. This also means that these are the only cases where the field on-axis is non-zero. In all the other cases, the phase varies along the angle ϕ, which manifests itself in a change from positive values to negative and back again. One of those cycles corresponds to a phase shift of 2π. The number of such cycles for one round-trip of ϕ is given by $|l - 1|$ for the right-handed, and $|l + 1|$ for the left-handed component, respectively, which also becomes clear from equations (2.7) and (2.8).

The z component has highest symmetry for $l = 0$. As expected, it does not have an angular phase dependency, but is circularly symmetric. This mode is the only one which has intensity on-axis, but it also has pronounced outer lobes. For the other values of l, there is no z component at the center of the r-ϕ plane. This means that for the $l = \pm 1$ modes, there is purely circularly polarized light on-axis according to the volume current method.

3.2 FDTD Simulations for parameter estimation

As mentioned in section 1.5, the transmission coefficient t has to be matched to the intensity decay coefficient α according to equation (1.26) for the critical coupling condition to be met, which leads to full power transfer into the ring. t depends on the coupling coefficient κ, which itself is determined by the width of the gap between feeding waveguide and ring resonator, whereas α depends on the strength of the emission grating. For both t and α, the relations can be estimated by individual simulations.

3.2.1 Coupling of the feeding waveguide to the ring

To estimate the connection between coupling gap width and transmission coefficient t, a structure where the feeding waveguide couples to a half ring with similar radius, width and height as the one in the real structure is simulated. The lumerical script for the structure generation can be found in listing 6.23, the MATLAB script to generate and analyze the following plots can be found in listing 6.7. Figure 3.2 shows the magnitude of the electric field in the waveguide plane for an exemplary coupling gap width of 0.1 μm.

It is important that the ring resonator is not closed, so that no radiation is fed back into the feeding waveguide from the resonator. In the terminology of section 1.5, $E_{r0} = 0$. Then, the transmission coefficient can be calculated from the electric fields before and after the coupling region by

$$|t| = \left| \frac{E_{through}}{E_{in}} \right|. \tag{3.1}$$

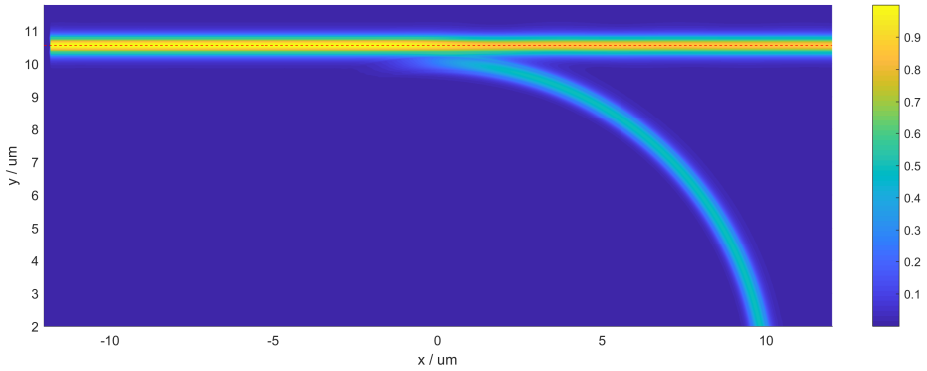


Figure 3.2: Magnitude of the electric field in the waveguide plane for a structure that consists of a straight waveguide coupled to a half ring. The power inside the waveguide is partially transferred to the half ring, which before hosted no power. The red dashed line indicates the middle of the waveguide, where a slice of the field profile is plotted in Figure 3.3.

Figure 3.3 shows a slice through the electric field in the middle of the feeding waveguide, which is marked in Figure 3.2 with a red dashed line. Since apart from the coupling, no electric field can escape the waveguide, it is fitted with horizontal lines before and after the coupling region, respectively. Here, a value of $t = 0.8645$ was obtained. From the transmission coefficient t, one can then get an estimate for the intensity decay coefficient α which would satisfy the critical coupling condition. For this gap width of 0.1 μm, that would be $\alpha = 4633$ m^{-1}. In general, the coupling strength decreases with increasing gap width, which matches the intuition that for a large gap, the waveguides should not be influenced by each other.

A possible source of error here is that the ring waveguide is modeled without the emission grating. One can think that if there are grating elements present in the vicinity of the coupling region, the power transfer might behave slightly differently than for the simulated plain waveguide.

3.2.2 Power loss from a straight grating

In order to obtain values for the intensity decay parameter α, the intensity decay caused by the emission grating has to be simulated and evaluated quantitatively. Since we neglect bending loss, a straight waveguide with a similar emission grating as in the ring serves this purpose. The lumerical script for the structure generation can be found in listing 6.24, the MATLAB script to generate and analyze the following plots can be found in listing 6.8.

Figure 3.4 shows the magnitude squared of the electric field in the waveguide plane of such a straight grating. From this intensity profile, one can take a slice in the middle of the waveguide, indicated by a red dashed line, similar to section 3.2.1. The difference is that here, the intensity does not decrease in a step-wise manner, but as an exponential decay, as can be seen in Figure 3.5.

From fitting the intensity profile as indicated with a red dashed line, one obtains the intensity decay parameter α. However, the profile is not exactly an exponential decay, but exhibits oscillatory features, which one could already observe in Figure 3.4. At the grating elements, the light wave is partially reflected, which also causes counter-propagating WGMs in the waveguide. They locally form standing waves with intensity maxima and minima. The feature of reflection

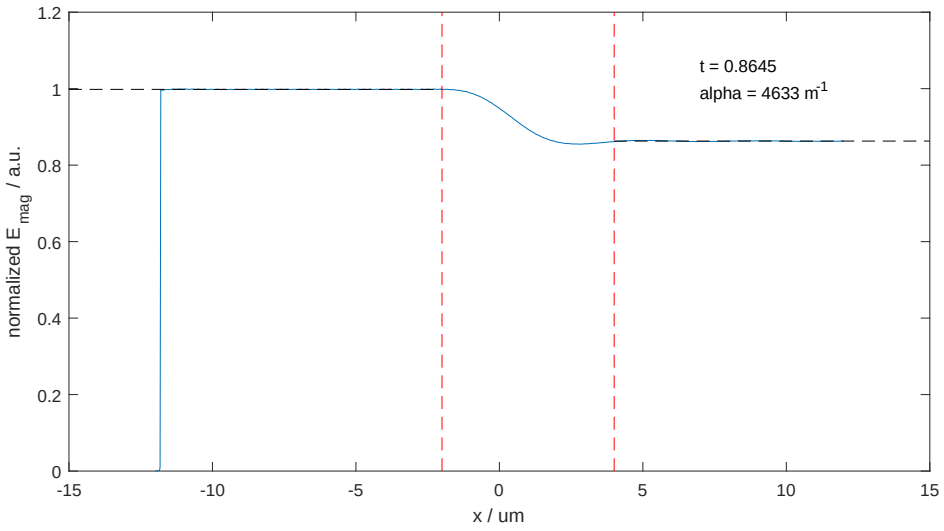


Figure 3.3: Normalized magnitude of the electric field in a straight waveguide which couples to a half ring waveguide at $x = 0$. The electric field is reduced because power is transferred into the half ring. Due to no backcoupling into the feeding waveguide, the transmission coefficient t can be obtained as ratio between the transmitted and the input electric field, according to equation (3.1). The black dashed lines indicate the horizontal fits to obtain the magnitudes before and after the coupling, the red dashed lines indicate the region which was omitted for the fitting.

at the grating elements will be observable at a later point, too. Additionally to this, there is a feature like a beating, but this actually is a result of the repeated shift in intensity maximum along the y axis, as observed before in Figure 3.4. Since the grating tooth height is is not a weak perturbation to the waveguide anymore with 0.21 μm, higher order waveguide modes may start to contribute, which causes the oscillation in intensity maximum tranverse location.

For the Figures presented here, the grating tooth height was 210 nm, which resulted in a large $\alpha = 21363\ \mathrm{m}^{-1}$. The procedure of simulating a straight grating with specific grating strength and following calculation of α was repeated for several other grating tooth heights, the correlation is presented in Figure 3.6. This allows to pick a grating tooth height which matches the coupling gap width so that the resonator is critically coupled and the emitted power is maximized. For example for the simulation of the full structure, a gap width of 130 nm was chosen so that the grating tooth height can be set to a relatively small value of 44 nm, which reduces deviations caused by reflection from the emission grating.

3.3 FDTD Simulations of the full structure

3.3.1 Transmission spectrum

The lumerical script for the structure generation can be found in listing 6.25, the MATLAB script to generate and analyze the following plots can be found in listing 6.9.

Figure 3.7 shows the transmission spectrum of the ring resonator structure. It is obtained in a similar way as the transmission coefficient t in section 3.2.1, only that t is defined as the ratio of the electric field magnitudes, whereas a transmission spectrum typically shows the relative

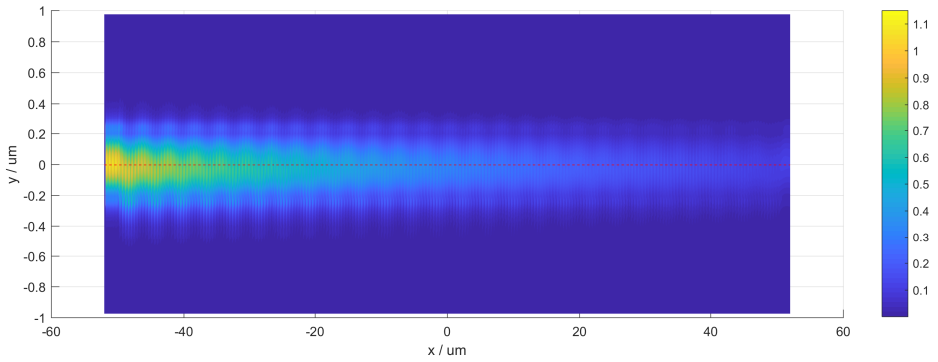


Figure 3.4: Magnitude squared of the electric field in a straight waveguide with an emission grating of 210 nm thickness. The intensity maximum moves along the y axis, which is caused by contribution from higher order waveguide modes due to the large additional grating. Reflections at the grating elements cause counter-propagating waves, which form oscillatory features of standing waves in the intensity profile. The red dashed line indicates the middle of the waveguide, a slice through the profile at this plane is plotted in Figure 3.5.

intensities. Also, the fitting procedure of the electric field before and after the coupling was repeated for a range of wavelengths to cover enough relevant resonator modes. In particular, the ones for which the azimuthal propagation constant m matches the number of grating elements N exactly or by a difference of 1 or 2 are of particular interest.

For a wavelength to be a resonant mode, its multiple must match the effective radius of the whispering gallery mode. If this is the case, the wave is self-consistent within the ring. As explained in sections 3.2.1 and 3.2.2 based on the coupling theory in section 1.5, the ring resonator is designed to be critically coupled. This means that on resonance, the transmission drops (close to) zero. In Figure 3.7, this can be observed for five modes within the wavelength range, which are the modes with $l = 2$ to $l = -2$. Shorter wavelengths have a higher l since the light can undergo more cycles during one round-trip through the resonator, i.e. m is higher. The modes can be identified by analyzing the radiated electric fields, which will be presented in section 3.3.2. When designing the ring resonator structure, the number of grating elements was selected via estimating m by taking the ring radius as the effective radius of the whispering gallery mode, which is not exactly known at first. This explains why the mode with $l = m - N = 0$ is not the one closest to the target wavelength $\lambda = 729$ nm. The transmission does not vanish completely at the resonant modes because due to the spacing of the wavelength sample points, they might not match the exact resonance wavelength. Probably more importantly though, the resonator structure differs from the one used to determine the parameters for critical coupling by the addition of the emission grating. This might affect the coupling coefficient, as has been mentioned in section 3.2.1. Still, the resonator can be regarded critically coupled since the vast majority of the power is transferred into the ring instead of being transmitted through the feeding waveguide. Another factor to keep in mind is that during fabrication, fluctuations in the exact width of the coupling gap or in the strength of the grating elements may be much more significant than the deviations in this simulation.

One feature in this transmission spectrum, which one might not find in usual transmission spectra of other resonators, is the split of the $l = 0$ resonance near 724 nm. This has been observed before,[6] and stems from the reflection of the electric field waves at the grating elements. Then the whispering gallery mode couples to its counter-propagating self, causing a split in

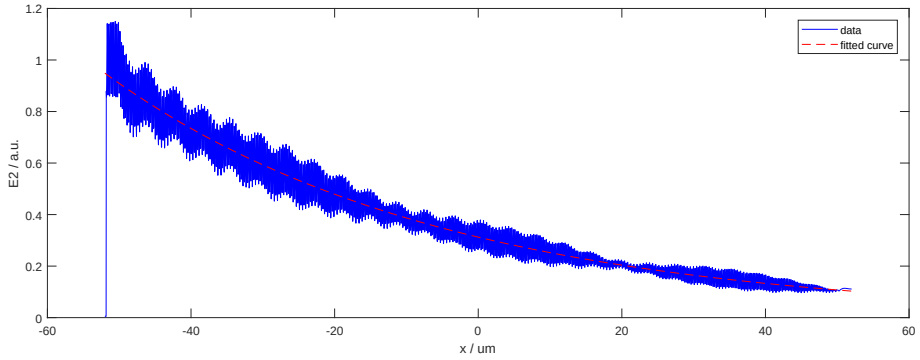


Figure 3.5: Normalized magnitude squared versus position along the straight waveguide (blue line) with a 210 nm thick grating. The intensity decays due to emission of power out of the waveguide by reflection at the grating elements which is overall well described by an exponential decay, as indicated with a red dashed line. However, there are also oscillatory features caused by counter-propagating waves which were reflected are the grating. What could be mistaken as beating is actually the shifting of the intensity maximum about the y axis, as observed in Figure 3.4.

frequencies.[13] Both waves are also coupled back into the feeding waveguide, so this feature is visible in the transmission spectrum. However, here this split is only observable for one mode, which is the $l = 0$ mode. At this wavelength, the wavelength in the waveguide matches the grating periodicity exactly, so that Bragg reflection occurs and reflected waves from neighboring grating elements add up constructively to form a much stronger counter-propagating wave than for other wavelengths. It is important that this effect of reflection is small for the $l = \pm 1$ modes because the addition of strong counter-propagating waves implies that the azimuthal mode index m is not well defined anymore. This would cause the emitted field to be a superposition of modes with different l values, so that the circularly polarized field on-axis would not be pure anymore.

At the resonances, the fraction of the input power which is emitted through the monitor in the air above the waveguide cover is about 16-20%. Half of the power is radiated downwards into the opposite direction, but still this value is unexpectedly low. In Figure 3.8 one can see that the majority of the emission seems to be captured with the air monitor. From the transmission spectrum becomes clear that almost no power is lost by transmission through the feeding waveguide. Possibly a large portion of the power is lost into unmonitored directions at the coupling region.

3.3.2 Electric field in the observer plane

The lumerical script for the structure generation can be found in listing 6.27. Since monitoring the whole frequency range of the pulse requires a lot of memory, the emitted fields were only saved around the resonance wavelengths obtained in section 3.3.1, both in the waveguide plane and in the air above. However, the different monitor geometries resulted in slightly different mesh grid positions, which caused a shift in the resonance wavelengths of about 1 nm. An exemplary intensity profile of the electric field in the air at 720.474 nm corresponding to the $l = 1$ mode in shown in Figure 3.8. From such a field monitor, the diffracted field is calculated. The MATLAB script to caluclate and analyze the radiated electric field can be found in listing 6.10.

Table 3.2 shows the electric field in the observer plane 50 μm above the waveguide plane for

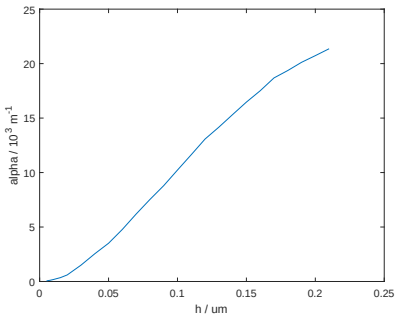


Figure 3.6: Intensity decay rate α versus grating tooth height h. More protruding grating elements increase the grating strength, which in turn lowers the intensity of guided electric fields faster.

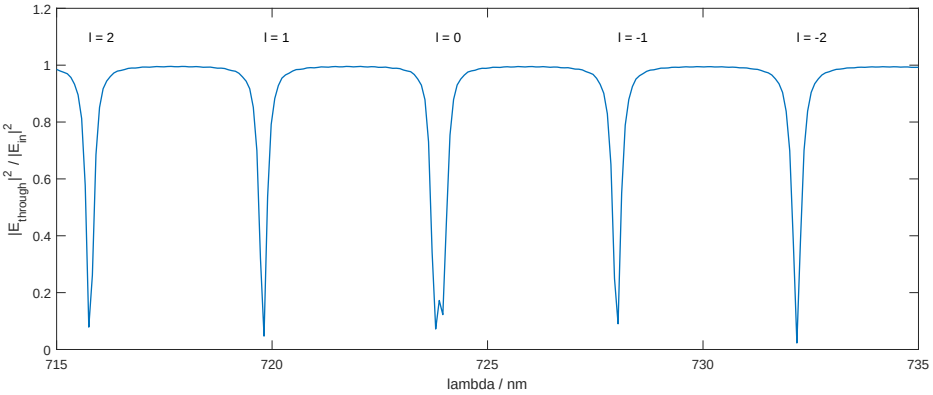


Figure 3.7: Transmission spectrum of the ring resonator obtained by fitting the magnitude of the electric field squared before and after the coupling similar to section 3.2.1, but repeated for a range of wavelengths.

different wavelengths, simulated from the three-dimensional structure. The different wavelengths correspond to different l modes, only the two patterns near 724 nm are from the same mode which is split, as pointed out in section 3.3.1. The field patterns can be compared to the ones obtained using the volume current method, which were presented in Table 3.1. The general features are similar, which is that for the $l = 1$ mode the right-handed, for the $l = -1$ mode the left-handed and for the $l = 0$ mode the z component is the only one with circular symmetry, and thus the only one with intensity on-axis. The respective other modes exhibit an angular phase dependency visible by one or multiple cycles through positive (yellow) and negative (blue) field when following a path around the origin. The two different wavelengths for the $l = 0$ mode only differ in a circular shift, but not in a change of general features.

A major difference to the results of the volume current method can be noticed in these plots already. For the modes with $l \neq 0$, there are irregularities in the shape of the left-handed component which are not present in the right-handed component. These come from the fact that the LH component is about one order of magnitude weaker than the RH component, which can also be seen in the different colorbars. This means that artifacts from the numerical grid or

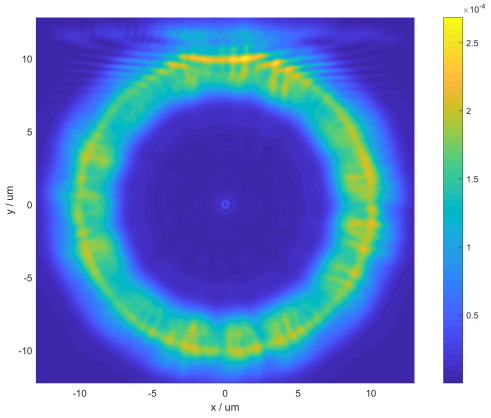


Figure 3.8: Intensity profile of the electric field at 720.474 nm in the air directly above the waveguide cover. The coupling from the feeding waveguide causes a blur at $y = 11$ μm.

noise from the coupling region are more visible than for stronger fields. The contributions of the individual field components to the overall intensity are compared in section 3.3.3.

3.3.3 Polarization contributions to overall intensity

Figure 3.9 shows the average contributions of the different polarization components to the overall intensity for the $l = 1$ mode, plotted versus the radius ρ. The right-handed component dominates the others by far. As pointed out in section 2.3, the z component is expected to be generally small. The difference in left-handed and right-handed circular polarization can be understood from equations (2.7) and (2.8) in section 2.2. The different strengths of the circular components stem from a difference in their respective amplitudes $\tilde{a}_{\pm}(r,z) = \frac{a_r(r,z) \pm i a_\phi(r,z)}{\sqrt{2}}$. They are determined by the relative magnitude and phase of the radial and azimuthal components in the ring waveguide. With the volume current method, a pure radial polarization was assumed, but the behaviour in the real structure is different, as already pointed out in section 2.1. Also in the source mode profiles in Figures 2.2a to 2.2c, one can see that the waveguide profile causes longitudinal fields at the interface, which is where the emission grating is located. The relative phase of the radial and azimuthal field components can be understood by a simplified analysis based on Gauss's law in a dielectric with no free charges:

$$\vec{\nabla} \cdot \vec{D} = 0, \tag{3.2}$$

where $\vec{D} = \epsilon \vec{E}$ is the electric displacement field in a dielectric with permittivity ϵ. In a two-dimensional single-mode waveguide along the z direction, $\vec{D}$ is of the form

$$\vec{D} = a_x(x,y)e^{i\beta z}\hat{x} + a_z(x,y)e^{i\beta z}\hat{z}, \tag{3.3}$$

where β is the propagation constant of the mode, and $a_x(x,y)$ and $a_z(x,y)$ are the transverse amplitude profiles of the transverse and longitudinal components, respectively. Inserting this form into equation (3.2) yields

$$\frac{\partial a_x(x,y)}{\partial x} + i\beta a_z(x,y) = 0$$
$$\Leftrightarrow a_z(x,y) = \frac{i}{\beta}\frac{\partial a_x(x,y)}{\partial x}. \tag{3.4}$$

At the interface going into the waveguide, the transverse component increases with x, so that the derivative in equation (3.4) is positive. This means that the longitudinal field component in a dielectric waveguide is phase-shifted by $\frac{\pi}{2}$ with respect to the transverse component. Identifying the longitudinal component in the ring waveguide as $a_\phi(r,z)$, and the transverse component as $a_r(r,z)$, it follows that $a_\phi(r,z) \propto i a_r(r,z)$ at the inner interface of the ring. Now, going back to the amplitudes of the emitted left- and right-handed circularly components $\tilde{a}_\pm(r,z) = \frac{a_r(r,z) \pm i a_\phi(r,z)}{\sqrt{2}}$, it becomes clear that the behaviour of the electric field in the ring waveguide causes the right-handed ($\hat{\sigma}_-$) component have greater amplitude than the left-handed one. The fact that the observed difference between both components is so large implies that radial and azimuthal components in the ring are of comparable magnitude. Note that an emission grating at the outer interface of the ring would produce the opposite behaviour, with the left-handed circularly polarized component being the overall dominant one. The volume current method results were obtained on the basis of pure radial polarization in the ring, but the observed behavior here could be reproduced using the appropriate superposition of radial and azimuthal polarization.

Going back to the observed field profiles, especially at the center for $\rho = 0$, other components than the RHCP vanish almost completely. More than 99.95% of the overall intensity is pure right-handed circularly polarized light, in accordance with the theory presented in section 2.2. Purity is defined here as the ratio of the intensity of the dominant component to the total intensity at $\rho = 0$. However, it is important to note that the numerical grid does not necessarily have a data point exactly at the origin. In fact, for the volume current method this had to be avoided since radial and azimuthal polarization are not defined at the origin. Instead, one could use a different grid for the far field calculations than for the simulation of the structure. Or like here, the data point for $\rho = 0$ is obtained by interpolation of the surrounding points. This then means that if there is a local minimum at the origin, the data is overestimated by the interpolation. A finer grid would decrease this error, and it would approximate the circular symmetry of the structure even better, thus approaching the analytical result of 100% purity. However, there are also other sources of non-ideality: The coupling waveguide breaks the circular symmetry which is the basis of the argument for pure circular polarization. Also the simulation domain introduces artifacts if the guided electric field is significantly non-zero at the boundary. This can be seen in Figure 3.8 as yellow intensity maxima at the vertical and horizontal edges of the ring. Still in general, to obtain reliable values at this level of accuracy, one should probably perform experimental characterization instead of basing on numerical simulations.

An estimate can be made of the size of the central area in which the radiated field is high-purity circularly polarized. A single trapped ion has a translational ground state wave function with an extend of about 10 nm. Within this range, the dominant polarization still well exceeds the impurities by three orders of magnitude. With increasing distance from the origin, the z component soon becomes the dominant impurity.

Figure 3.10 shows the contributions of the intensity of the $l = -1$ mode. Analogously, on-axis there is only contribution from the left-handed component. But since also for this mode, the right-handed component generally dominates, the purity at the center is lower with 98.94%. Figures 3.11a to 3.11d show the contribution distributions for the other modes. Interestingly, the right- and left-handed components have comparable intensities for the $l = 0$ mode, but for

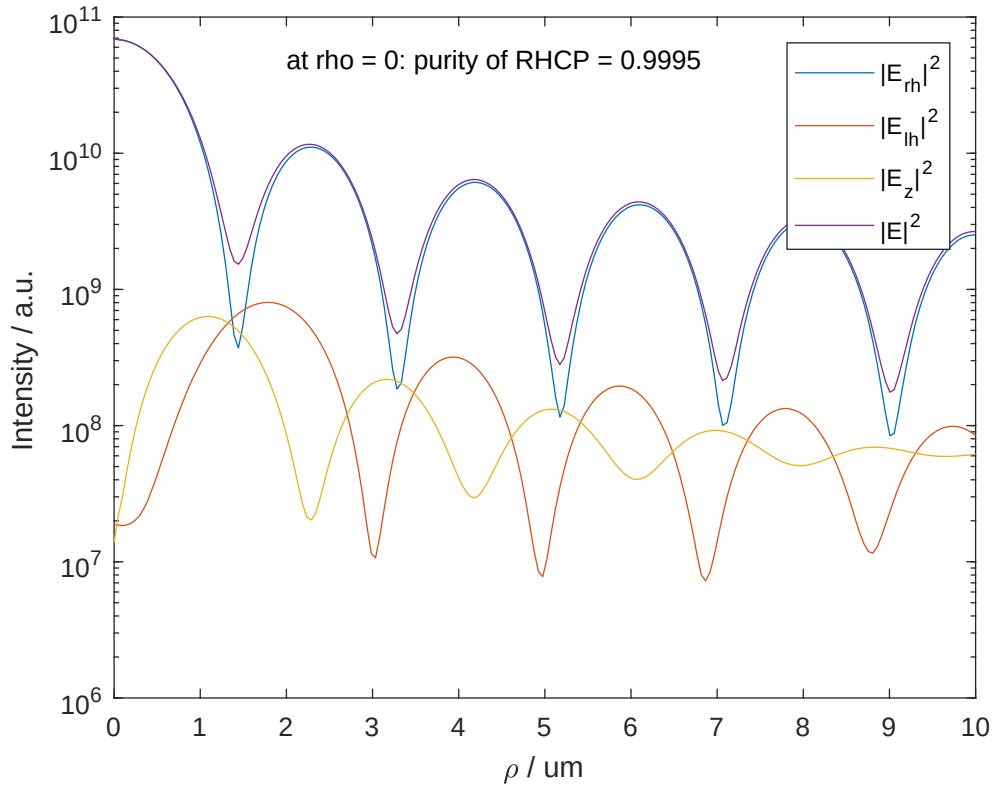


Figure 3.9: Average radial intensity profile for the $l = 1$ mode of each component and of the total intensity, plotted on a logarithmic scale. The right-handed component dominates the left-handed one due to constructive versus destructive superposition of radial and azimuthal polarization components in the ring waveguide. At the center, there is almost only contribution from the right-handed component, in agreement with the theory outlined in section 2.2.

the others, the right-handed component is always dominant. This specialty of the $l = 0$ mode could not be reproduced with the volume current method, so it is thought to be an effect of the Bragg reflection at the grating elements which creates counter-propagating WGMs.

Also, the contributions to the overall intensity are not the same for both resonances of the $l = 0$ mode. The circularly polarized components differ slightly, but the z component is much stronger at for the 724.473 nm resonance, especially at the center.

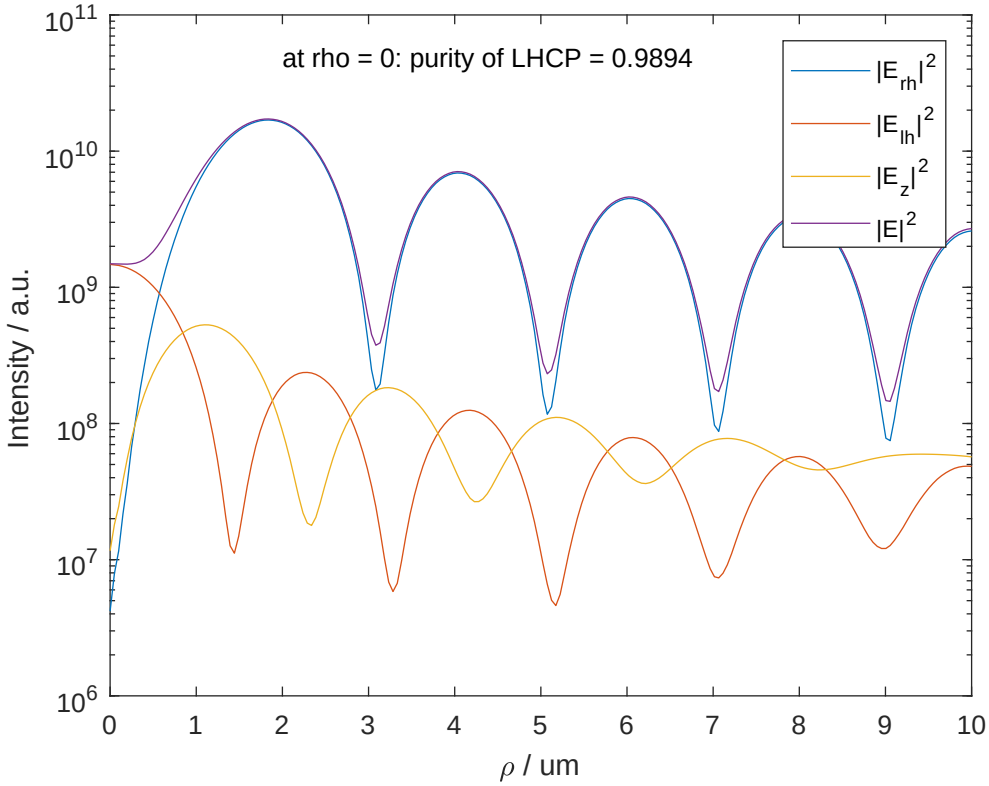
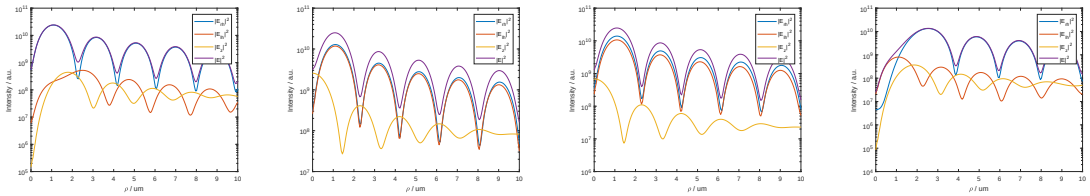


Figure 3.10: Average radial intensity profile for the $l = -1$ mode of each component and of the total intensity, plotted on a logarithmic axis. Analogous to Figure 3.9, there is now almost only the left-handed component contributing to the overall intensity at the center. However, the right-handed component is still much stronger for the outer lobes.



(a) $l = 2$ (b) $l = 0$ (c) $l = 0$ (d) $l = -2$

Figure 3.11: Average radial intensity profiles of each component and of the total intensity for different azimuthal modes l, plotted on a logarithmic axis. Except for the $l = 0$ modes, the right-handed component always dominates the intensity. This exception might be an effect of the Bragg reflection which occurs only for this mode.

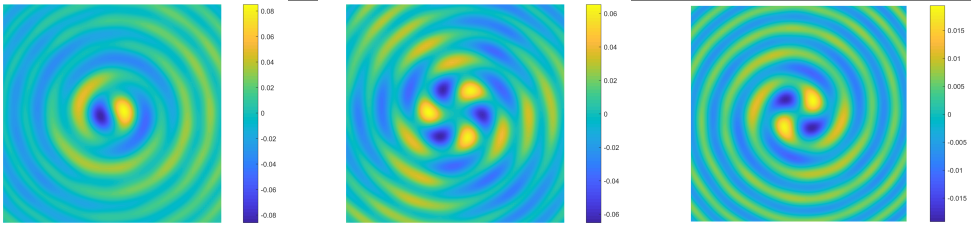
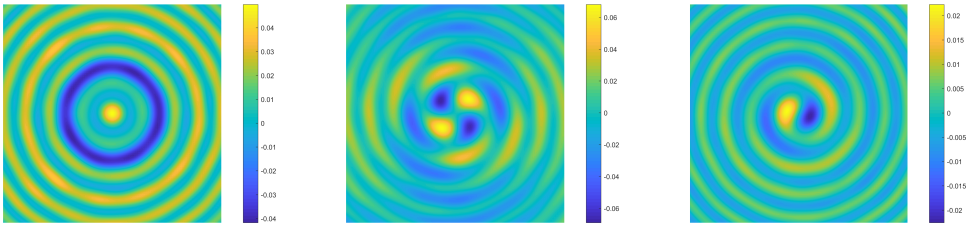
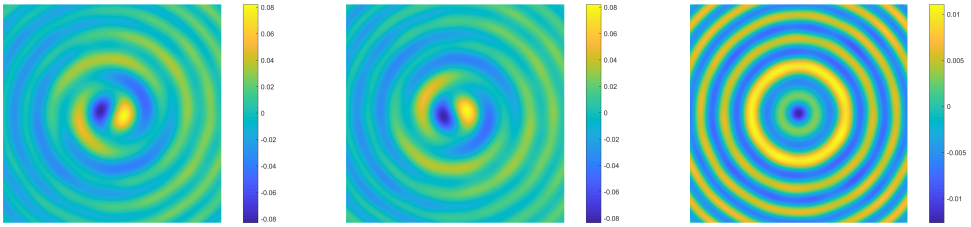
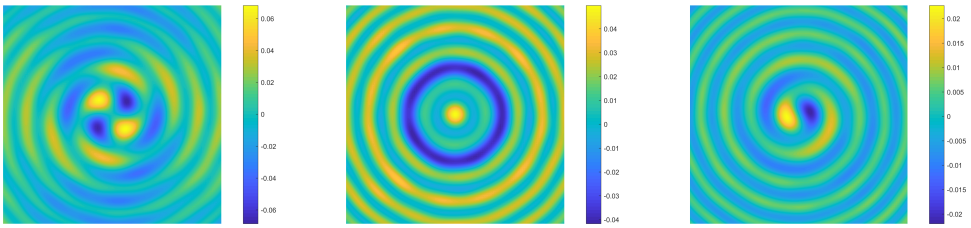
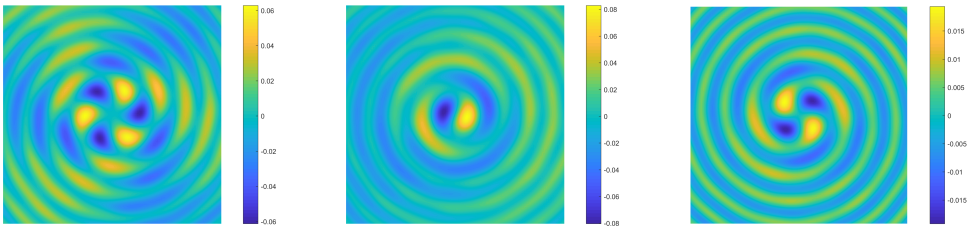
l	**RHCP**	**LHCP**	**z**
2			
1			
0			
-1			
-2			

Table 3.1: Simulated electric field patterns at a distance of 50 μm in the z direction for different azimuthal modes l, obtained with the volume current method. The x and y extend are ± 10 μm.

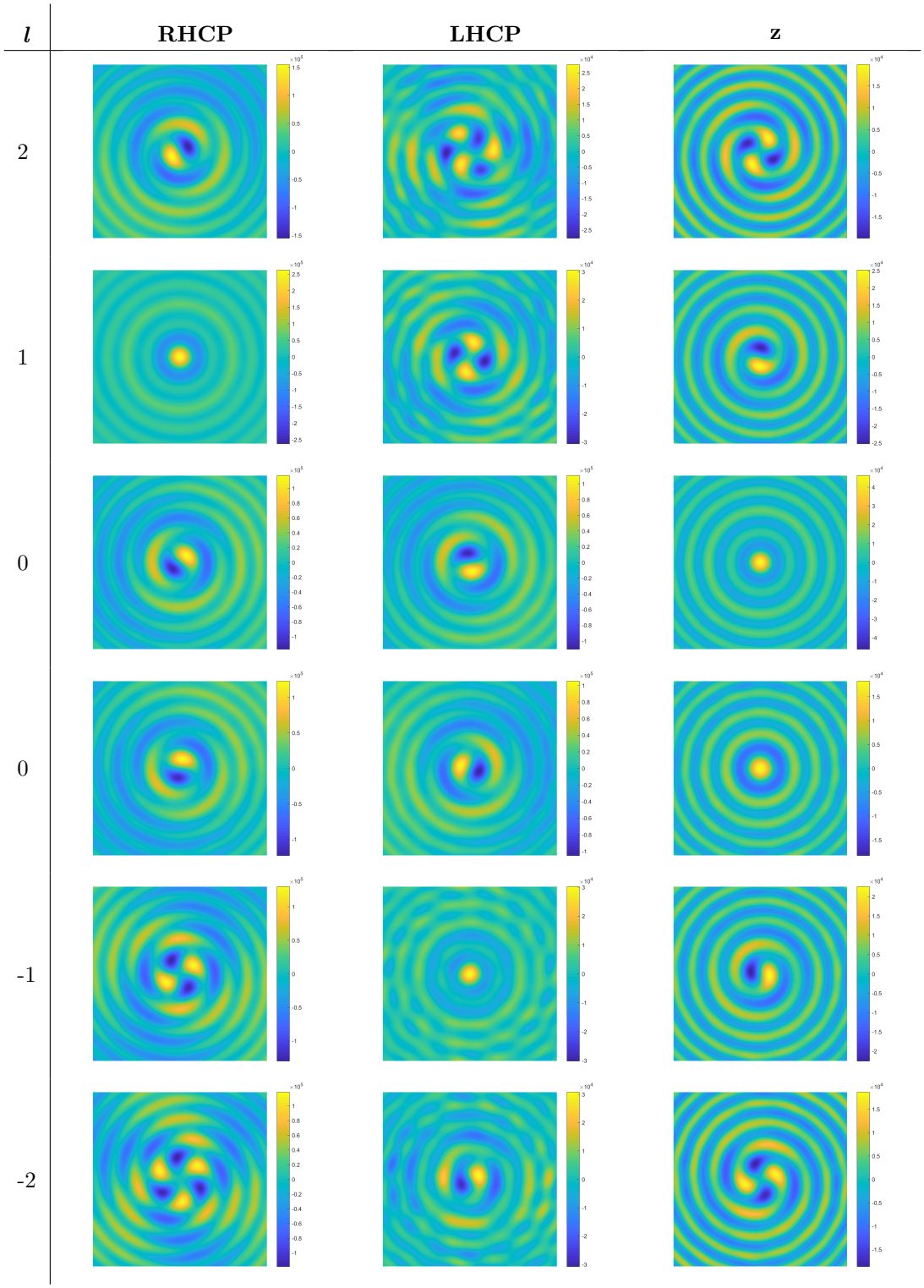


Table 3.2: Simulated electric field patterns at a distance of 50 μm in the z direction for different wavelengths λ, which correspond to different azimuthal modes l, obtained with the FDTD method. The x and y extend are ± 10 μm.

Chapter 4

Conclusions and Outlook

The radiation emitted by a ring resonator with an angular grating as sketched in Figure 2.1 was simulated by two different methods. The volume current method was adapted semi-analytically in a two-dimensional domain where the structure is implemented as perturbation to the surrounding dielectric constant. The finite-difference time-domain method allowed a more detailed study in three dimensions, implementing the real geometry of the ring resonator. For this geometry, structural parameters like the width of the coupling gap and the grating tooth height were optimized to achieve critical coupling. Despite the major differences of both methods, they confirm the key theory that for an azimuthal mode in the resonator with its index differing from the number of grating elements by 1, the resonator emits almost purely circularly polarized light at the symmetry axis of the ring. Depending on the relative strengths of the radial and azimuthal polarization components in the ring - corresponding to transverse and longitudinal components in the waveguide picture - one circular polarization component may be generally dominant over the other, leading to even higher purity.

Using the implementation of these methods, one can simulate structures with varying parameters like ring radius, waveguide dimensions, and differences in the shape of the grating elements. Their effects on the purity of the circularly polarized light as well as the spatial extent of the desired central area can be simulated. Another factor to optimize is the intensity of the central spot compared to radiation emitted in other directions. This also applies to the distribution of intensity along the emission axis z, which attains a maximum at a certain distance. Since the aim is to induce a certain transition in a trapped ion, the device has to be designed to support exactly this wavelength. Ideally, one device would allow to be tuned to support a range of wavelengths, potentially by means of temperature change or by applying a static electric field to an electro-optic waveguide material. This tuning is also necessary to compensate for possible fabrication errors which shift the resonance wavelength. One might also try to design the ring waveguide and its grating in such way that the longitudinal and transverse field components are of equal magnitude and precisely phase-shifted to maximize the radiation of only the one desired circular polarization component.

The next step is to fabricate such a device and experimentally test it. As mentioned in section 3.3.1 however, systematical errors in the fabrication of the waveguide and grating structure can be significant enough to impair critical coupling. Thus, a set of ring resonators with different parameters should provide enough variety to achieve good power transfer for some of them. Then, the emitted radiation can be characterized by microscopical methods, and the results can be compared to similar simulations. Important is not only whether these devices can achieve the desired pure radiation, but also how accurately the simulations predict their properties.

If the ring resonators with azimuthal grating fulfill the purity requirements and deliver enough optical power to the desired location, they may be used in an integrated optics ion trap to induce different atomic transitions than linearly polarized light. The use of integrated optics for this purpose is advantageous as compared to external laser beams due to the ability to easily deliver optical power to a desired point in space accurately.

Bibliography

[1] H. Häffner, C.F. Roos, and R. Blatt. Quantum computing with trapped ions. *Physics Reports*, 469(4):155 – 203, 2008. ISSN 0370-1573. doi: https://doi.org/10.1016/j.physrep.2008.09.003. URL http://www.sciencedirect.com/science/article/pii/S0370157308003463.

[2] M. G. Raizen, J. M. Gilligan, J. C. Bergquist, W. M. Itano, and D. J. Wineland. Ionic crystals in a linear paul trap. *Phys. Rev. A*, 45:6493–6501, May 1992. doi: 10.1103/PhysRevA.45.6493. URL https://link.aps.org/doi/10.1103/PhysRevA.45.6493.

[3] Karan K. Mehta, Colin D. Bruzewicz, Robert McConnell, Rajeev J. Ram, Jeremy M. Sage, and John Chiaverini. Integrated optical addressing of an ion qubit. *Nature Nanotechnology*, 11:1066 EP –, Aug 2016. URL http://dx.doi.org/10.1038/nnano.2016.139.

[4] Karan K. Mehta and Rajeev J. Ram. Precise and diffraction-limited waveguide-to-free-space focusing gratings. *Scientific Reports*, 7(1):2019, 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-02169-2. URL https://doi.org/10.1038/s41598-017-02169-2.

[5] C. E. Langer. *High Fidelity Quantum Information Processing with Trapped Ions*. PhD thesis, Department of Physics, University of Colorado, Boulder, 2006.

[6] Xinlun Cai, Jianwei Wang, Michael J. Strain, Benjamin Johnson-Morris, Jiangbo Zhu, Marc Sorel, Jeremy L. O'Brien, Mark G. Thompson, and Siyuan Yu. Integrated compact optical vortex beam emitters. *Science*, 338(6105):363–366, 2012. ISSN 0036-8075. doi: 10.1126/science.1226528. URL http://science.sciencemag.org/content/338/6105/363.

[7] H. C. Nägerl, Ch. Roos, D. Leibfried, H. Rohde, G. Thalhammer, J. Eschner, F. Schmidt-Kaler, and R. Blatt. Investigating a qubit candidate: Spectroscopy on the $S_{1/2}$ to $D_{5/2}$ transition of a trapped calcium ion in a linear paul trap. *Phys. Rev. A*, 61:023405, Jan 2000. doi: 10.1103/PhysRevA.61.023405. URL https://link.aps.org/doi/10.1103/PhysRevA.61.023405.

[8] Bahaa E. A. Saleh and Malvin Carl Teich. *Fundamentals of Photonics*. John Wiley & Sons, Inc., Dec 2001.

[9] L. Allen, M. W. Beijersbergen, R. J. C. Spreeuw, and J. P. Woerdman. Orbital angular momentum of light and the transformation of laguerre-gaussian laser modes. *Phys. Rev. A*, 45:8185–8189, Jun 1992. doi: 10.1103/PhysRevA.45.8185. URL https://link.aps.org/doi/10.1103/PhysRevA.45.8185.

[10] A. Yariv. Universal relations for coupling of optical power between microresonators and dielectric waveguides. *Electronics Letters*, 36(4):321–322, Feb 2000. ISSN 0013-5194. doi: 10.1049/el:20000340.

[11] Jeffrey B. Driscoll, Xiaoping Liu, Saam Yasseri, Iwei Hsieh, Jerry I. Dadap, and Richard M. Osgood. Large longitudinal electric fields (ez) in silicon nanowire waveguides. *Opt. Express*, 17(4):2797–2804, Feb 2009. doi: 10.1364/OE.17.002797. URL `http://www.opticsexpress.org/abstract.cfm?URI=oe-17-4-2797`.

[12] M. Kuznetsov and H. Haus. Radiation loss in dielectric waveguide structures by the volume current method. *IEEE Journal of Quantum Electronics*, 19(10):1505–1514, Oct 1983. ISSN 0018-9197. doi: 10.1109/JQE.1983.1071758.

[13] D. S. Weiss, V. Sandoghdar, J. Hare, V. Lefèvre-Seguin, J.-M. Raimond, and S. Haroche. Splitting of high-q mie modes induced by light backscattering in silica microspheres. *Opt. Lett.*, 20(18):1835–1837, Sep 1995. doi: 10.1364/OL.20.001835. URL `http://ol.osa.org/abstract.cfm?URI=ol-20-18-1835`.

Chapter 6

Appendix

6.1 Scripts/Code

6.1.1 Mathematica

VCM Simplifation of eq (2.6)

```
Δr[x_, y_, z_] := Sqrt[(x - x1)^2 + (y - y1)^2 + z^2]
```
Quadratwurzel

```
F[a_] := Exp[-i * k * a] / a
```
Exponentialfunktion

```
d2dx2f = D[F[Δr[x, y, z]], {x, 2}]
```
leite ab

$$\frac{2\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i k \left(x-x1\right)^2}{\left(\left(x-x1\right)^2+\left(y-y1\right)^2+z^2\right)^2} +$$

$$e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}} \left(\frac{3\left(x-x1\right)^2}{\left(\left(x-x1\right)^2+\left(y-y1\right)^2+z^2\right)^{5/2}} - \frac{1}{\left(\left(x-x1\right)^2+\left(y-y1\right)^2+z^2\right)^{3/2}}\right) +$$

$$\left(\frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i k \left(x-x1\right)^2}{\left(\left(x-x1\right)^2+\left(y-y1\right)^2+z^2\right)^{3/2}} + \frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i^2 k^2 \left(x-x1\right)^2}{\left(x-x1\right)^2+\left(y-y1\right)^2+z^2} - \right.$$

$$\left.\frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i k}{\sqrt{\left(x-x1\right)^2+\left(y-y1\right)^2+z^2}}\right) \Big/ \left(\sqrt{\left(x-x1\right)^2+\left(y-y1\right)^2+z^2}\right)$$

```
FullSimplify[d2dx2f]
```
vereinfache vollständig

$$\left(e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\right.$$

$$\left(i^2 k^2 x^4 - 4 i^2 k^2 x^3 x1 + i^2 k^2 x1^4 - \left(\left(y-y1\right)^2+z^2\right)\left(1 + i k \sqrt{\left(x-x1\right)^2+\left(y-y1\right)^2+z^2}\right) + \right.$$

$$x1^2 \left(2 + i k \left(i k \left(\left(y-y1\right)^2+z^2\right) + 2\sqrt{\left(x-x1\right)^2+\left(y-y1\right)^2+z^2}\right)\right) -$$

$$2\, x\, x1 \left(2 + i k \left(2\sqrt{\left(x-x1\right)^2+\left(y-y1\right)^2+z^2} + i k \left(2\, x1^2 + \left(y-y1\right)^2+z^2\right)\right)\right) +$$

$$\left.\left.x^2 \left(2 + i k \left(2\sqrt{\left(x-x1\right)^2+\left(y-y1\right)^2+z^2} + i k \left(6\, x1^2 + \left(y-y1\right)^2+z^2\right)\right)\right)\right) \Big/$$

$$\left(\left(x-x1\right)^2+\left(y-y1\right)^2+z^2\right)^{5/2}$$

```
d2dy2f = D[F[Δr[x, y, z]], {y, 2}]
```

leite ab

$$\frac{2\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i\, k\, (y-y1)^2}{\left((x-x1)^2+(y-y1)^2+z^2\right)^2} +$$

$$e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}} \left(\frac{3\,(y-y1)^2}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}} - \frac{1}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{3/2}} \right) +$$

$$\left(\frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i\, k\, (y-y1)^2}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{3/2}} + \frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i^2\, k^2\, (y-y1)^2}{(x-x1)^2+(y-y1)^2+z^2} - \right.$$

$$\left. \frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i\, k}{\sqrt{(x-x1)^2+(y-y1)^2+z^2}} \right) \Big/ \left(\sqrt{(x-x1)^2+(y-y1)^2+z^2}\right)$$

```
FullSimplify[d2dy2f]
```

vereinfache vollständig

$$\left(e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}} \right.$$

$$\left. \left(-(x-x1)^2 + 2\,(y-y1)^2 - z^2 + 2\, i\, k\, (y-y1)^2 \sqrt{(x-x1)^2+(y-y1)^2+z^2} - i\, k \right.\right.$$

$$\left.\left. \sqrt{(x-x1)^2+(y-y1)^2+z^2}\, \left((x-x1)^2 + z^2 - i\, k\, (y-y1)^2 \sqrt{(x-x1)^2+(y-y1)^2+z^2} \right) \right) \right) \Big/$$

$$\left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}$$

```
ddxddyf = D[F[Δr[x, y, z]], x, y]
```

leite ab

$$\frac{3\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, (x-x1)\, (y-y1)}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}} +$$

$$\frac{3\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i\, k\, (x-x1)\, (y-y1)}{\left((x-x1)^2+(y-y1)^2+z^2\right)^2} + \frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i^2\, k^2\, (x-x1)\, (y-y1)}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{3/2}}$$

```
FullSimplify[ddxddyf]
```

vereinfache vollständig

$$\left(e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, (x-x1)\, (y-y1) \right.$$

$$\left. \left(3 + i\, k \left(3 \sqrt{(x-x1)^2+(y-y1)^2+z^2} + i\, k \left((x-x1)^2+(y-y1)^2+z^2 \right) \right) \right) \right) \Big/$$

$$\left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}$$

```
ddzddxf = D[F[Δr[x, y, z]], x, z]
```

leite ab

$$\frac{3\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, (x-x1)\, z}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}} +$$

$$\frac{3\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i\, k\, (x-x1)\, z}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{2}} + \frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i^2\, k^2\, (x-x1)\, z}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{3/2}}$$

```
FullSimplify[ddzddxf]
```

vereinfache vollständig

$$\left(e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, (x-x1)\, z \left(3+i k \left(3\sqrt{(x-x1)^2+(y-y1)^2+z^2}+i k \left((x-x1)^2+(y-y1)^2+z^2\right)\right)\right)\right) \Big/ \left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}$$

```
ddzddyf = D[F[Δr[x, y, z]], x, y]
```

leite ab

$$\frac{3\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, (x-x1)\, (y-y1)}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}} +$$

$$\frac{3\, e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i\, k\, (x-x1)\, (y-y1)}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{2}} + \frac{e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, i^2\, k^2\, (x-x1)\, (y-y1)}{\left((x-x1)^2+(y-y1)^2+z^2\right)^{3/2}}$$

```
FullSimplify[ddzddyf]
```

vereinfache vollständig

$$\left(e^{-i k \sqrt{(x-x1)^2+(y-y1)^2+z^2}}\, (x-x1)\, (y-y1) \left(3+i k \left(3\sqrt{(x-x1)^2+(y-y1)^2+z^2}+i k \left((x-x1)^2+(y-y1)^2+z^2\right)\right)\right)\right) \Big/ \left((x-x1)^2+(y-y1)^2+z^2\right)^{5/2}$$

6.1.2 MATLAB

Listing 6.1: helix.m

```
%{
helix.m
stom ETH Zurich 05.07.2018

Script to create a surface plot of a helix with l = 0.
%}

syms s t

fig = figure();
fsurf(s*sin(t),-s*cos(t),t,[-5 5 0 4*pi],'EdgeColor','none')
axis off

savefig(fig,'helix.fig')
```

6.1.2.1 Volume Current Method

Listing 6.2: ring_resonator_4.m

```
%{
ring_resonator_2.m
stom ETH Zurich 23.05.2018

Based on ring_resonator_2.m. Calculation with different parameters in order
to have best comparability with the lumerical simulations. The other
functions are just copied

Script to run the radiation calculation of the ring resonator specified in
function my_radiation_fct.m for different parameters.

Scalars are usually denoted with small letters, matrices are capitalized.
Matrices describing vectors have a 3rd dimension, whose 1st and 2nd are
the x and y components of the vector, respectively.
%}

clear variables
clear global

%% natural constants
% source: Wikipedia
% in SI base units (except microm)

global c0 eps0 mu0

% vacuum speed of light
c0 = 299792458 * 10^6;          % [microm s^-1]

% vacuum permittivity
eps0 = 8.854187817 * 10^-18;    % [s^4 A^2 kg^-1 microm^-3]
```

```
% vacuum permeability
mu0 = eps0^-1 * c0^-2;          % [s^-2 A^-2 kg microm]


%% parameters
% all length units are micrometer
path = sprintf( ...
    './01 results sm0.1 obss0.1 z50 xmax10'); % ADJUST PATH REGULARLY

wg_spacing_multiplier = 0.1;    % stepsize of coordinates in waveguide
                                % plane compared to lambda0 ( called sm)
obs_z = 50;                  % distance of view
obs_xmax = 10;          % size of viewing plane, independent of waveguide
obs_ymax = obs_xmax;  % plane
obs_spacing = 0.1;     % spacing of points in viewing plane
lambda0 = 0.729;           % wavelength in vacuum
eps1 = 1.67 * eps0;     % "permittivity" of the waveguide, is eps in vcm!
                    % actually this is just the effective index of the
                    % waveguide
eps2 = eps0;             % permittivity of the surrounding, is eps1 in vcm!
mN = -2;     % difference of m and N_grating, this used to be lm
radius = 10;              % waveguide radius
width = 0.45;    % waveguide width
what_to_calculate = 'A_vec';
% string whether to calculate the far field approximation, the X-Y plane
% without approximation, or the X-Z plane without approximation, or just
% J_vec
% options: 'A_vec_ff', 'A_vec', 'A_vec_XZ', 'J_vec'


%% calculation
% looping over the different values of the parameters, do one calculation
% of 'what_to_calculate' each

% cell array to store the results
results = cell(length(mN),length(radius));
for i = 1:length(mN)
    for j = 1:length(radius)
        for k = 1:length(width)
            tic
            [wg_X,wg_Y,obs_X,obs_Y,obs_Z,wg_Rhat,wg_Phihat,...
                obs_Rhat,obs_Phihat,obs_Rhat3D,obs_Phihat3D, ...
                Eps,J_vec,A_vec,A_vec_XZ] = ...
                my_radiation_fct(wg_spacing_multiplier, ...
                obs_xmax,obs_ymax,obs_spacing,lambda0,eps1,eps2, ...
                obs_z,mN(i),radius(j),width(k),what_to_calculate);
            toc
            if strcmp(what_to_calculate, 'A_vec_ff')
                results{i,j,k} = A_vec_ff;
            elseif  strcmp(what_to_calculate, 'A_vec')
                results{i,j,k} = A_vec;
                filename02 = sprintf(...
```

```
                    '%s/A_vec/A_vec_mN%.0f_r%.0f_w%.2f.mat', ...
                    path, mN(i), radius(j), width(k));
                save(filename02, 'A_vec','obs_X','obs_Y')
            elseif strcmp(what_to_calculate, 'A_vec_XZ')
                results{i,j,k} = A_vec_XZ;
                figure()
                surf(obs_X,obs_Z, mag2_fct(A_vec_XZ))
                shading interp
                filename03 = sprintf(...
                    '%s/A_vec_XZ/A_vec_XZ_mN%.0f_r%.0f_w%.2f.mat', ...
                    path, mN(i), radius(j), width(k));
                save(filename03, 'A_vec_XZ','obs_X','obs_Z')
            else
                results{i,j,k} = nan;
            end
        end
    end
end


%% Electric radiation field

if strcmp(what_to_calculate, 'A_vec')
    % with the three-dimensional gradient
    tic
    [E_vec_3D, E1_vec_3D, E2_vec_3D] = ...
        E_vec_3D_fct(lambda0,eps2,wg_X,wg_Y,obs_X,obs_Y,obs_z,J_vec,A_vec);
    toc
    % save the results
    filename1 = sprintf('%s/E_vec_3D/E_vec_3D_mN%.0f_r%.0f_w%.1f.mat', ...
        path, mN(i), radius(j), width(k));
    save(filename1, 'E_vec_3D','obs_X','obs_Y')
end


%% Plots

% % Rhat
% quiver(X,Y,Rhat(:,:,1),Rhat(:,:,2))

% % Phihat
% quiver(X,Y,Phihat(:,:,1),Phihat(:,:,2))

% % Rhat3D
% quiver(X,Y,Rhat3D(:,:,1),Rhat3D(:,:,2))

% % Eps
% surf(X(1,:),Y(:,1),Eps)
% colorbar
% shading interp

% % waveguide_E_vec
% surf(waveguide_E_scal)
```

```
% colorbar
% shading interp

%% waveguide_E_vec
% contour(X,Y,waveguide_E_scal)
% hold on
% quiver(X,Y,waveguide_E_vec(:,:,1),waveguide_E_vec(:,:,2))
% hold off

%% J_vec
% plot_all_components_fct(J_vec, wg_X, wg_Y, wg_Rhat, wg_Phihat)

%% A_vec_ff
% figure(1)
% quiver(X,Y,abs(A_vec_ff(:,:,1)),abs(A_vec_ff(:,:,2)))

% A_vec
% plot_all_components_fct(A_vec,'A\_vec',obs_X,obs_Y,obs_Rhat,obs_Phihat)

%% A_vec_XZ
% figure(1)
% quiver(obs_X,obs_Z,real(A_vec_XZ(:,:,1)),real(A_vec_XZ(:,:,2)))
% axis equal
% figure()
% surf(obs_X,obs_Z,mag2_fct(A_vec_XZ))
% shading interp
% title({'A\_vec\_XZ',sprintf('mN = %.0f',mN)})
% axis equal
%% compare multiple results
% for i=1:length(width)
%     figure(i)
%     quiver(obs_X,obs_Z,real(results{i}(:,:,1)),real(results{i}(:,:,2)))
% end

%% E_vec_3D
plot_circular_components_fct(E_vec_3D,sprintf('mN = %.0f',mN), ...
    obs_X,obs_Y)
% plot_all_components_wo_phases_fct(E_vec_3D,'E\_vec\_3D', ...
%     obs_X,obs_Y,obs_Rhat3D,obs_Phihat3D)
% plot_all_components_fct(E_vec_3D,'E\_vec\_3D', ...
%     obs_X,obs_Y,obs_Rhat3D,obs_Phihat3D)
% plot_all_components_wo_unwrap_fct(E_vec_3D,'E\_vec\_3D', ...
%     obs_X,obs_Y,obs_Rhat3D,obs_Phihat3D)
```

Listing 6.3: my_radiation_fct.m

```
%{
my_radiation_fct.m
stom ETH Zurich 23.05.2018

Based on the function from ring resonator_2, here, lm is now mN, and m is
now depending on N, not the other way round. This is similar to the
lumerical simulaitons, where the modes are chosen according to their
wavelengths.
```

```
Function to run what was initially just a script for different parameters.

First calculation of the radiation out of a periodically modified ring
resonator using the volume current method (vcm, Kuznetsov, Haus 1983 -
Radiation Loss in Dielectric Waveguide Structures by the Volume Current
Method). Also gives back the basis of calculation, that is the matrices
wg_X, wg_Y, Rhat and Phihat because the calculation domain is based on
waveguide dimensions, which are intended to be parameters to be changed at
will. Also gives back the coordinates at which the radiation is defined,
that is obs_X and obs_Y or obs_Z.

Scalars are usually denoted with small letters, matrices are capitalized.
Matrices describing vectors have a 3rd dimension, whose 1st and 2nd are
the x and y components of the vector, respectively.
%}

function [ wg_X,wg_Y,obs_X,obs_Y,obs_Z,wg_Rhat,wg_Phihat, ...
    obs_Rhat,obs_Phihat,obs_Rhat3D,obs_Phihat3D,Eps, ...
    J_vec,A_vec,A_vec_XZ ] = ...
    my_radiation_fct( wg_spacing_multiplier, obs_xmax, obs_ymax, ...
    obs_spacing, lambda0, eps1, eps2, ...
    obs_z, mN, radius, width, what_to_calculate )

%% global variables
% get from running script

global c0 eps0


%% parameters
% all length units are micrometer

omega0 = 2*pi*c0/lambda0;
circumfence_inner = 2*pi*(radius-0.5*width);
circumfence_outer = 2*pi*(radius+0.5*width);

N_lower = ceil(circumfence_inner/lambda_fct(eps1,lambda0));      % just an
N_upper = floor(circumfence_outer/lambda_fct(eps1,lambda0));     % estimate
N = floor((N_upper+N_lower)/2);
% N is the number of cyclic variations of waveguide permittivity, this
% models the grating teeth
m = N + mN;
% m is the number of wavelengths in the resonator for constructive
% interference -> the mode number



%% calculation basis: waveguide domain

wg_xmax = (radius + 0.1);
wg_ymax = wg_xmax;
% npts has to be even, or else there are NaN @ origin
```

```
npts_wg_x = ceil(wg_xmax/wg_spacing_multiplier/lambda0/2)*2;
npts_wg_y = npts_wg_x+2;

% 2D
[wg_X,wg_Y] = meshgrid(linspace(-wg_xmax, wg_xmax, npts_wg_x), ...
    linspace(-wg_ymax, wg_ymax, npts_wg_y));
wg_R = sqrt((wg_X.^2 + wg_Y.^2));
wg_Rhat = cat(3,wg_X./wg_R,wg_Y./wg_R); % unit vector
wg_Phi = atan2(wg_Y,wg_X);
wg_Phihat = cat(3,-sin(wg_Phi), cos(wg_Phi)); % unit vector


%% calculation basis: observer domain

% npts has to be even, or else there are NaN @ origin
npts_obs_x = ceil(obs_xmax/obs_spacing/2)*2;
npts_obs_y = npts_obs_x+2;

% 2D
[obs_X,obs_Y] = meshgrid(linspace(-obs_xmax, obs_xmax, npts_obs_x), ...
    linspace(-obs_ymax, obs_ymax, npts_obs_y));
obs_R = sqrt((obs_X.^2 + obs_Y.^2));
obs_Rhat = cat(3,obs_X./obs_R,obs_Y./obs_R); % unit vector
obs_Phi = atan2(obs_Y,obs_X);
obs_Phihat = cat(3,-sin(obs_Phi), cos(obs_Phi));    % unit vector

% 3D (cylindrical coordinates, so just add zeros for the 3rd component)
obs_Rhat3D = cat(3,obs_Rhat,zeros(size(obs_R))); % unit vector
obs_Phihat3D = cat(3,obs_Phihat,zeros(size(obs_Phi)));    % unit vector


%% setting up waveguide

% set delta_eps everywhere to zero, then set to eps1-eps2+variation where
% the waveguide is
delta_Eps = zeros(size(wg_X)) ...
    + (eps1-eps2 + eps0*0.2*sin(N*wg_Phi)) ...
    .*(wg_R >= (radius-0.5*width)).*(wg_R <= (radius+0.5*width));

% set Eps everywhere to eps2, then add delta_Eps where the waveguide is
Eps = ones(size(wg_X)) * eps2 + ...
    delta_Eps.*(wg_R >= (radius-0.5*width)).*(wg_R <= (radius+0.5*width));


%% Electric field in the waveguide

% scalar amplitude
% set everywhere to zero, then set to half a sine where the waveguide is
wg_E_scal = zeros(size(wg_X)) ...
    + sin(pi/width*(wg_R-(radius-0.5*width))) ...
    .*(wg_R >= (radius-0.5*width)).*(wg_R <= (radius+0.5*width));

% introducing field vector with phase
```

```
wg_E_vec = wg_Rhat .* wg_E_scal .* exp(1i*m*wg_Phi);


%% polarization surface current density
% equation (2.4) in vcm

J_vec = 1i * omega0 .* delta_Eps .* wg_E_vec;


%% Radiation field

if  strcmp(what_to_calculate, 'A_vec')
    % without far field approximation (equation (2.7) in vcm)
    % in the observer X-Y plane for a fixed z value:
    A_vec = A_vec_fct(lambda0,wg_X,wg_Y,obs_X,obs_Y,obs_z,J_vec);
    obs_Z = nan;
    A_vec_XZ = nan;

elseif strcmp(what_to_calculate, 'A_vec_XZ')
    % in the X-Z plane for a fixed y
    obs_y = 0;
    [obs_X, obs_Z] = meshgrid(linspace(-obs_xmax,obs_xmax,npts_obs_x), ...
        linspace(0.5, obs_z, 2*npts_obs_x));
    % start z from 0.5 micrometers to avoid singularities
    % still use npts_obs_x for comparable calculation time

    obs_Y = nan;
    A_vec_XZ = A_vec_XZ_fct(lambda0,wg_X,wg_Y,obs_X,obs_Z,obs_y,J_vec);
    A_vec = nan;
elseif strcmp(what_to_calculate, 'J_vec')
    obs_X = nan;
    obs_Y = nan;
    obs_Z = nan;
    A_vec = nan;
    A_vec_XZ = nan;
else
    obs_X = nan;
    obs_Y = nan;
    obs_Z = nan;
    A_vec = nan;
    A_vec_XZ = nan;
end
end
```

Listing 6.4: A_vec_fct.m

```
%{
A_vec_fct.m
stom ETH Zurich 29.03.2018

Based on the one in ring_resonator_1, but modified to allow for an observer
plane with different size than the waveguide plane.

Calculates the vector potential A_vec at the observer coordinates obs_X,
```

```
obs_Y, obs_z from the propagation wavelength lambda, the waveguide
coordinates wg_X, wg_Y and the polarization current density J_vec,
according to equation (2.7) in vcm. Note: The integral is evaluated for
every combination of coordinates X and Y, but all the time for the one
fixed z.
%}

function [ A_vec ] = A_vec_fct( lambda,wg_X,wg_Y,obs_X,obs_Y,obs_z,J_vec )
    global mu0
    [i_obs_X,i_obs_Y] = meshgrid(1:size(obs_X,2),1:size(obs_Y,1));
    prefactor = mu0/(4*pi);

    integral_X = @(i_obs_x,i_obs_y) trapz(wg_Y(:,1),trapz(wg_X(1,:), ...
        J_vec(:,:,1) .* exp(1i*2*pi/lambda ...
        .* sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_Y(i_obs_y,1)-wg_Y).^2 + obs_z^2)) ...
        ./ sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_Y(i_obs_y,1)-wg_Y).^2 + obs_z^2), ...
        2),1);
    integral_Y = @(i_obs_x,i_obs_y) trapz(wg_Y(:,1),trapz(wg_X(1,:), ...
        J_vec(:,:,2) .* exp(1i*2*pi/lambda ...
        .* sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_Y(i_obs_y,1)-wg_Y).^2 + obs_z^2)) ...
        ./ sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_Y(i_obs_y,1)-wg_Y).^2 + obs_z^2) ...
        ,2),1);

    A_vec = cat(3, ...
        prefactor.*arrayfun(integral_X,i_obs_X,i_obs_Y), ...
        prefactor.*arrayfun(integral_Y,i_obs_X,i_obs_Y));
end
```

Listing 6.5: A_vec_XZ_fct.m

```
%{
A_vec_XZ.m
stom ETH Zurich 29.03.2018

Based on the one in ring_resonator_1, but modified to allow for an observer
plane with different size than the waveguide plane.

Radiation field vector from the popagation wavelength lambda, the
coordinates wg_X, wg_Y of the polarization current density vector J_vec,
the observer coordinates obs_X, obs_Z and obs_y at which the integral shall
be evaluated, according to equation (2.7) in vcm. Note: The integral is
evaluated for every combination of coordinates obs_X and obs_Z, but all
the time for the one fixed obs_y.
%}

function [ A_vec_XZ ] = A_vec_XZ_fct( lambda,wg_X,wg_Y, ...
    obs_X,obs_Z,obs_y,J_vec )
    % note that the components of A_vec_XZ are still in the x- and
    % y-direction
    global mu0
```

```
    % indices of the coordinate grid
    [i_obs_X,i_obs_Z] = meshgrid(1:size(obs_X,2),1:size(obs_Z,1));
    prefactor = mu0/(4*pi);

    % calculate the components of A_vec individually
    integral_X = @(i_obs_x,i_obs_z) trapz(wg_Y(:,1), trapz(wg_X(1,:), ...
        J_vec(:,:,1) .* exp(1i*2*pi/lambda ...
        .* sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_y-wg_Y).^2 + obs_Z(i_obs_z,1)^2)) ...
        ./ sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_y-wg_Y).^2 + obs_Z(i_obs_z,1)^2), ...
        2),1);
    integral_Y = @(i_obs_x,i_obs_z) trapz(wg_Y(:,1), trapz(wg_X(1,:), ...
        J_vec(:,:,2) .* exp(1i*2*pi/lambda ...
        .* sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_y-wg_Y).^2 + obs_Z(i_obs_z,1)^2)) ...
        ./ sqrt((obs_X(1,i_obs_x)-wg_X).^2 ...
        + (obs_y-wg_Y).^2 + obs_Z(i_obs_z,1)^2), ...
        2),1);

    A_vec_XZ = cat(3, ...
        prefactor.*arrayfun(integral_X,i_obs_X,i_obs_Z), ...
        prefactor.*arrayfun(integral_Y,i_obs_X,i_obs_Z));
end
```

Listing 6.6: E_vec_3D_fct.m

```
%{
E_vec_3D_fct.m
stom ETH Zurich 09.04.2018

Calculation of the electric field vector E_vec in the obs_X-Y plane at
distance obs_z from the wavelength lambda, the propagation dielectric
constant eps2 and the vector potential A_vec according to eq. (2.6) in vcm.
However, to try to reduce calculation time, instead of doing "brute force"
numerical divergence and gradient, they are applied analytically to the
vector potential A_vec of eq. (2.7), which also requires the polarization
current density J_vec and its coordinates in teh waveguide plane wg_X-Y.
This is done partially on paper, the second derivatives are calculated in
the Mathematica notebook vcm_simplification.nb.
The two terms of eq. (2.6) are denoted as E1_vec and E2_vec, respectively.
%}

function [ E_vec, E1_vec, E2_vec ] = ...
    E_vec_3D_fct( lambda0,eps2,wg_X,wg_Y,obs_X,obs_Y,obs_z,J_vec,A_vec )
    global c0 eps0
    omega = 2*pi*c0/lambda0;
    n = sqrt(eps2/eps0);
    k = 2*pi*n/lambda0;

    % split equation 2.6 into two terms: E1 and E2
    %% E1
    A_vec(:,:,3) = 0;   % add the z-dimension to A_vec
    E1_vec = -1i * omega .* A_vec;
```

```
%% E2
% instead of using arrayfun, this time use for-loops. This allows to
% reduce unnecessary multiple calculations/function calls (e.g.
% distance).
E2x = zeros(size(A_vec(:,:,1)));
E2y = E2x;
E2z = E2x;
for i_obs_x = 1:size(obs_X,2)
    obs_x = obs_X(1,i_obs_x);
    tic
    fprintf('i_x is %.0f of %.0f.\n',i_obs_x,size(obs_X,2))
    for i_obs_y = 1:size(obs_Y,1)
        obs_y = obs_Y(i_obs_y,1);

        % analytical formulas for the derivatives from Mathematica
        % notebook VCM_simplification.nb
        distx = (obs_x-wg_X);
        disty = (obs_y-wg_Y);
        distx2 = distx.^2;
        disty2 = disty.^2;
        z2 = obs_z^2;
        dist = sqrt(distx2+disty2+z2);
        d2dx2f = ...    % from d2dy2f but distx2 <-> disty2
              (1./(dist.^5)).*exp(-1i*k*dist).* ...
              (-disty2 + 2*distx2 - z2 + 2*1i*k*distx2.*dist ...
              - 1i*k*dist.*(disty2 + z2 - 1i*k*distx2.*dist));
        d2dy2f = ...    % from Mathematica
              (1./(dist.^5)).*exp(-1i*k*dist).* ...
              (-distx2 + 2*disty2 - z2 + 2*1i*k*disty2.*dist ...
              - 1i*k*dist.*(distx2 + z2 - 1i*k*disty2.*dist));
        ddxddyf = (1./(dist.^5)).*exp(-1i*k*dist).*distx.*disty.*...
            (3 + 1i*k*(3*dist + 1i*k*dist.^2));
        ddzddxf = (1./(dist.^5)).*exp(-1i*k*dist).*distx*obs_z.* ...
            (3 + 1i*k*(3*dist + 1i*k*dist.^2));
        ddzddyf = (1./(dist.^5)).*exp(-1i*k*dist).*disty*obs_z.* ...
            (3 + 1i*k*(3*dist + 1i*k*dist.^2));

        integrand_x = J_vec(:,:,1).*d2dx2f + J_vec(:,:,2).*ddxddyf;
        integrand_y = J_vec(:,:,1).*ddxddyf + J_vec(:,:,2).*d2dy2f;
        integrand_z = J_vec(:,:,1).*ddzddxf + J_vec(:,:,2).*ddzddyf;
        E2x(i_obs_y,i_obs_x) = trapz(wg_Y(:,1), trapz(wg_X(1,:), ...
            integrand_x,2),1);
        E2y(i_obs_y,i_obs_x) = trapz(wg_Y(:,1), trapz(wg_X(1,:), ...
            integrand_y,2),1);
        E2z(i_obs_y,i_obs_x) = trapz(wg_Y(:,1), trapz(wg_X(1,:), ...
            integrand_z,2),1);
    end
    toc
end

prefactor = 1/(1i*omega*4*pi*eps2);
E2_vec = prefactor * cat(3, E2x, E2y, E2z);
```

```
    E_vec = E1_vec + E2_vec;
end
```

6.1.2.2 Analysis of FDTD Results

Listing 6.7: transmission_coefficient.m

```
%{
transmission_coefficient.m
stom ETH Zurich 27.04.2018

Script to fit the Electric field magnitude along the x dimension, once
before coupling with a half ring, and once after the coupling. From the
ratio of the two fields, the transmission coefficient t is calculated.

The data come from lumerical simulations and are imported as .mat files.
This creates the variables x, y, Ex, Ey, Ez and waveguide_y.

Scalars are usually denoted with small letters, matrices are capitalized.
Matrices describing vectors have a 3rd dimension, whose elements are
the x, y and z components of the vector, respectively.
%}

clear variables
clear global

path = 'data';
gap = 0.13e-6;
infilename = sprintf('%s/E_field_gap%.2f_ma3.mat',path,gap*1e6);
load(infilename);

[X,Y] = meshgrid(x,y);
x0 = -2e-6; % beginning of coupling
x1 = 4e-6; % end of coupling
y0 = waveguide_y;
clear waveguide_y

% transpose E field matrices to fit the matlab mesh
Ex = Ex.';
Ey = Ey.';
Ez = Ez.';

E_vec = cat(3,Ex,Ey,Ez);
Emag = sqrt(mag2_fct(E_vec));

% find the indices of the x vector where the coupling begins and ends
[~,i_x0] = min(abs(x-x0));
[~,i_x1] = min(abs(x-x1));

% find index where y is in the middle of the waveguide
[~,i_y0] = min(abs(y-y0));

% take slice at y0
```

```
Emag_slice = Emag(i_y0,:);

% fit
myfittype = fittype({'1','0*x'},'coefficients',{'a','b'});
myfit0 = fit(x,Emag_slice.',myfittype, ...
    'Exclude',[1:8,(i_x0+1):length(Emag_slice)]);
myfit1 = fit(x,Emag_slice.',myfittype, ...
    'Exclude',1:(i_x1-1));

% get coefficient results
coefs0 = coeffvalues(myfit0);
coefs1 = coeffvalues(myfit1);
a0 = coefs0(1);
a1 = coefs1(1);

% get transmission coefficient t
t = a1/a0;
alpha_target = -log(t)/pi/10e-6;
results = sprintf('t = %.4f\nalpha = %.0f m^{-1}\n',t,alpha_target);

% plot Emag with surf
my_variable_surf_fct(X,Y,Emag,'Emag','gap',gap,1e-6,'um')
line(x*1e6,y0*1e6*ones(size(x)),Emag(i_y0,:), ...
    'Color','red','LineStyle','--');

% plot slice
figure()
plot(x*1e6,Emag_slice);
xlabel('x / um');
ylabel('normalized E_{mag} / a.u.');
xl = xlim;
yl = [0,1.2];
ylim(yl); % set ylim so that it is not changed by adding the vertical lines
line([x0,x0]*1e6,yl,'Color','red','LineStyle','--');
line([x1,x1]*1e6,yl,'Color','red','LineStyle','--');
line([xl(1),x(i_x0)*1e6],[a0,a0],'Color','black','LineStyle','--');
line([x(i_x1)*1e6,xl(end)],[a1,a1],'Color','black','LineStyle','--');
text(xl(2)-8,1,results);
```

Listing 6.8: grating_strength.m

```
%{
grating_strength.m
stom ETH Zurich 11.05.2018

Script to fit the Electric field intensity along the x dimension, where the
grating is. This is fitted with an exponential decay, and the intensity
decay constant alpha is correlated with grating tooth height/thickness h.

The data come from lumerical simulations and are imported as .mat files.
This creates the variables x, y, Ex, Ey, Ez and waveguide_y.

Scalars are usually denoted with small letters, matrices are capitalized.
Matrices describing vectors have a 3rd dimension, whose elements are
```

```
the x, y and z components of the vector, respectively.
%}

clear variables
clear global

path = 'data/0.005-0.21/';
h = [0.005:0.0025:0.015,0.02:0.01:0.21]*1e-6;


x0 = -50e-6; % beginning of grating
x1 = -x0; % end of grating


myfittype = fittype('a*exp(-b*(x-x0))','problem','x0','independent','x');

alpha = zeros(size(h));

for k = 1:length(h)

    infilename = sprintf('%sE_field_h%.4f_ma3.mat',path,h(k)*1e6);
    load(infilename);

    [X,Y] = meshgrid(x,y);

    % transpose E field matrices to fit the matlab mesh
    Ex = Ex.';
    Ey = Ey.';
    Ez = Ez.';

    E_vec = cat(3,Ex,Ey,Ez);
    E2 = mag2_fct(E_vec);
    % take slice in the middle, where y=0
    [~,i_y0] = min(abs(y-0));
    E2_slice = E2(i_y0,:);

    % find the indices of the x vector where the coupling begins and ends
    [~,i_x0] = min(abs(x-x0));
    [~,i_x1] = min(abs(x-x1));

    % fit
    myfit = fit(x,E2_slice.',myfittype,'problem',x(i_x0), ...
        'Exclude',[1:(i_x0-1),(i_x1+1):length(x)]);

    % get coefficient results
    coefs = coeffvalues(myfit);
    alpha(k) = coefs(2);
end

save('results/alpha_vs_h','h','alpha');

figure()
plot(h*1e6,alpha*1e-3);
```

```
% title('intensity decay rate alpha versus grating tooth height h');
xlabel('h / um');
ylabel('alpha / 10^3 m^{-1}');

my_variable_surf_fct(X,Y,E2,'E2','h',h(k),1e-6,'um')
line(x*1e6,y(i_y0)*1e6*ones(size(x)),E2(i_y0,:), ...
    'Color','red','LineStyle','--');

% plot slice
figure()
plot(myfit,'--r',x,E2_slice,'-b');
ax = gca;
ax.XTickLabel = {'-60','-40','-20','0','20','40','60'};
xlabel('x / um');
ylabel('E2 / a.u.');
```

Listing 6.9: transmission_spectrum.m

```
%{
transmission_spectrum.m
stom ETH Zurich 26.04.2018

Based on transmission_ratio.m, similar to transmission_coefficient.m.

Script to loop over differenct frequencies to create a plot of transmission
ratio SQUARED versus wavelength -> a transmission spectrum.

Script to fit the Electric field magnitude along the x dimension, once
before coupling with a half ring, and once after the coupling. From the
ratio of the two fields, the transmission ratio is calculated.

The data come from lumerical simulations and are imported as .mat files.
This creates the variables x, y, Ex, Ey, Ez.

Scalars are usually denoted with small letters, matrices are capitalized.
Matrices describing vectors have a 3rd dimension, whose elements are
the x, y and z components of the vector, respectively.
%}

clear variables
clear global


gap = 0.13e-6;
h = 0.044e-6;
ma = 3; % mesh accuracy of the simulation


path_data = sprintf('data/gap%.2f_h%.3f_ma%.0f_only_wg/',gap*1e6,h*1e6,ma);
path_E = sprintf('%sE_ring/',path_data);
path_monitor_data = sprintf('%smonitor_data/',path_data);

% load general monitor data
load(strcat(path_monitor_data,'lambda'));
```

```
load(strcat(path_monitor_data,'x'));
load(strcat(path_monitor_data,'y'));
load(strcat(path_monitor_data,'waveguide_y'));

[X,Y] = meshgrid(x,y);
x0 = -3e-6; % beginning of coupling
x1 = 4e-6; % end of coupling
y0 = waveguide_y;
clear waveguide_y

% fit type
myfittype = fittype({'1','0*x'},'coefficients',{'a','b'});

% find the indices of the x vector where the coupling begins and ends
[~,i_x0] = min(abs(x-x0));
[~,i_x1] = min(abs(x-x1));

% find index where y is in the middle of the waveguide
[~,i_y0] = min(abs(y-y0));

transmission = zeros(size(lambda));

for frequency_point = 1:length(lambda)
    infilename = sprintf('%sE_ring_lambda%.3f.mat',...
        path_E,lambda(frequency_point)*1e9);
    load(infilename);

    % transpose E field matrices to fit the matlab mesh
    Ex = Ex.';
    Ey = Ey.';
    Ez = Ez.';

    E_vec = cat(3,Ex,Ey,Ez);
    E2 = mag2_fct(E_vec);

    % take slice at y0
    E2_slice = E2(i_y0,:);

    % fit
    myfit0 = fit(x,E2_slice.',myfittype, ...
        'Exclude',[1:8,(i_x0+1):length(E2_slice)]);
    myfit1 = fit(x,E2_slice.',myfittype, ...
        'Exclude',1:(i_x1-1));

    % get coefficient results
    coefs0 = coeffvalues(myfit0);
    coefs1 = coeffvalues(myfit1);
    a0 = coefs0(1);
    a1 = coefs1(1);

    % get transmission t^2 and add to vector
    transmission(frequency_point) = a1/a0;
```

```
end

% clear variables which were created in the loop so that they do not cause
% any confusion afterwards
clear frequency_point infilename Ex Ey Ez E_vec Emag Emag_slice
clear myfit0 myfit1 coefs0 coefs1 a0 a1

% find the resonances
[~,i_modes] = findpeaks(-transmission,'MinPeakHeight',-0.5, ...
    'Annotate','peaks');
lambda_modes = lambda(i_modes);

% save transmission spectrum data
outfilename = sprintf(...
    'results/v2_transmission_spectrum_gap%.2f_h%.3f_ma%.0f_only_wg.mat',...
    gap*1e6,h*1e6,ma);
save(outfilename,'transmission','lambda');

% save mode information
outfilename =sprintf(...
    'results/v2_modes_gap%.2f_h%.3f_ma%.0f_only_wg.mat',...
    gap*1e6,h*1e6,ma);
save(outfilename,'lambda','i_modes');


% plot transmission spectrum
figure()
plot(lambda*1e9,transmission,'LineWidth',1);
ylim([0,1.2]);
xlabel('lambda / nm');
ylabel('|E_{through}|^2 / |E_{in}|^2');
% line(1e9*[lambda(i_modes),lambda(i_modes)],[0,1.2], ...
%     'LineStyle','--','Color','black','LineWidth',0.3);
mode_numbers = [2,1,0,-1,-2];
mode_labels_xpos = 1e9*lambda(i_modes([1:3,5,6]));
for i = 1:length(mode_numbers)
    text(mode_labels_xpos(i),1.1,sprintf('l = %.0f',mode_numbers(i)));
end

% save figure
outfilename = sprintf(...
    'figures/v2_transmission_spectrum_gap%.2f_h%.3f_ma%.0f_only_wg.fig',...
    gap*1e6,h*1e6,ma);
savefig(outfilename);
```

Listing 6.10: evaluate_E_ff.m

```
%{
evaluate_E_ff.m
stom ETH Zurich 25.06.2018

Script to evaluate the electric field. That is to plot the indivudual
components and to analyze the purity at (x,y) = (0,0).
```

```
Also the data is compared quantitatively to the simulations results from
the volume current method (VCM)

The data come from lumerical simulations and are imported as .mat files.
This creates the variables obs_x, obs_y, Ex_ff, Ey_ff, and Ez_ff.

Scalars are usually denoted with small letters, matrices are capitalized.
Matrices describing vectors have a 3rd dimension, whose elements are
the x, y and z components of the vector, respectively.
%}

clear variables
clear global

% parameters
path_E = 'data/E_ff/v2.2/';
% path_modeinfo = '../ring_resonator_v2/results/';
gap = 0.13e-6;
h = 0.044e-6;
ma = 3;
obs_z = 50e-6;
lambda = 720.474e-9;
mymode = 1; % mode of VCM which is taken

%% load mode information data, variables lambda, and i_modes
%% this does not work for v2.2 since not the whole structure is monitored,
%% i.e. there is no transmission spectrum obtained.
% infilename = sprintf('%sv3_modes_gap%.2f_h%.3f_ma%.0f.mat',...
%     path_modeinfo,gap*1e6,h*1e6,ma);
% load(infilename);
% lambda = lambda(i_modes(mymode));

infilename = sprintf('%sE_ff_gap%.2f_h%.3f_ma%.0f_z%.0f_lambda%.3f.mat',...
    path_E,gap*1e6,h*1e6,ma,obs_z*1e6,lambda*1e9);
load(infilename);

% transpose E field matrices to fit the matlab mesh
% additionally they have to be complex conjugated because lumerical uses
% the different sign convention for the propagation of a wave in space
% the command ' does the complex conjugate transpose, the simple transpose
% would be .'
E.x = Ex_ff';
E.y = Ey_ff';
E.z = Ez_ff';
clear Ex_ff Ey_ff Ez_ff
% concatenate to one variable
E.vec = cat(3,E.x,E.y,E.z);

% calculation basis
[X,Y] = meshgrid(obs_x,obs_y);
R = sqrt((X.^2 + Y.^2));
Rhat3D = cat(3,X./R,Y./R,zeros(size(R)));
Phi = atan2(Y,X);
```

```
Phihat3D = cat(3,-sin(Phi), cos(Phi),zeros(size(Phi)));

% normalize for later comparison to VCM
% so the magnitude squared of the electric field integrated over the whole
% area should be equal to one.
C = get_normalization_constant_fct(X,Y,E.vec);
E.x = C * E.x;
E.y = C * E.y;
E.z = C * E.z;
E.vec = C * E.vec;

% get the circular components
[E.rh, E.lh] = circular_components_fct(E.vec);

% get the intensity for each component
E.rh2 = mag2_fct(E.rh);
E.lh2 = mag2_fct(E.lh);
E.z2 = mag2_fct(E.z);
E.E2 = mag2_fct(E.vec);

% integrate over all angles to get a radial distribution. Do this for all
% individual components of the basis. The returned rho only depends on X
% and Y, so it will be the same for these three calls
[E_phiint.rh2,rho] = integrate_phi_fct(E.rh2,X,Y);
[E_phiint.lh2,~] = integrate_phi_fct(E.lh2,X,Y);
[E_phiint.z2,~] = integrate_phi_fct(E.z2,X,Y);
[E_phiint.E2,~] = integrate_phi_fct(E.E2,X,Y);

% get radial slices using the rho obtained from integrate_phi_fct
E_rslice.rh2 = interp2(X,Y,E.rh2,rho,zeros(size(rho)));
E_rslice.lh2 = interp2(X,Y,E.lh2,rho,zeros(size(rho)));
E_rslice.z2 = interp2(X,Y,E.z2,rho,zeros(size(rho)));
E_rslice.E2 = interp2(X,Y,E.E2,rho,zeros(size(rho)));

% calculate some purity of the circular components at the center
purity = max(E_rslice.rh2(1),E_rslice.lh2(1))/E_rslice.E2(1);

% average radial slices
% will yield the same rho as before, because it only depends on X and Y,
% provided that the number of points chosen within both functions is the
% same
[E_avg_rslice.rh2,~] = average_phi_fct(E.rh2,X,Y);
[E_avg_rslice.lh2,~] = average_phi_fct(E.lh2,X,Y);
[E_avg_rslice.z2,~] = average_phi_fct(E.z2,X,Y);
[E_avg_rslice.E2,~] = average_phi_fct(E.E2,X,Y);

% calculate some average purity of the circular components at the center
avg_purity = max(E_avg_rslice.rh2(1),E_avg_rslice.lh2(1)) ...
    / E_avg_rslice.E2(1);


%% Comparison to VCM
```

```
% load data to compare to
path_VCM = '../../VCM_2D/ring_resonator_4/';
path_VCM = sprintf('%s01 results sm0.1 obss0.1 z50 xmax10/', path_VCM);
path_VCM = sprintf('%sE_vec_3D/', path_VCM);
infilename = sprintf('%sE_vec_3D_mN%.0f_r10_w0.5.mat', path_VCM,mymode);
VCM_data = load(infilename);

% convert coordinates to micrometers
VCM_data.obs_X = 1e-6 * VCM_data.obs_X;
VCM_data.obs_Y = 1e-6 * VCM_data.obs_Y;

% interpolate the data to fit the lumerical mesh points
E_VCM.x = interp2(VCM_data.obs_X,VCM_data.obs_Y, ...
    VCM_data.E_vec_3D(:,:,1),X,Y);
E_VCM.y = interp2(VCM_data.obs_X,VCM_data.obs_Y, ...
    VCM_data.E_vec_3D(:,:,2),X,Y);
E_VCM.z = interp2(VCM_data.obs_X,VCM_data.obs_Y, ...
    VCM_data.E_vec_3D(:,:,3),X,Y);
E_VCM.vec = cat(3,E_VCM.x,E_VCM.y,E_VCM.z);

% normalize the VCM data over the lumerical area
C_VCM = get_normalization_constant_fct(X,Y,E_VCM.vec);
E_VCM.x = C_VCM * E_VCM.x;
E_VCM.y = C_VCM * E_VCM.y;
E_VCM.z = C_VCM * E_VCM.z;
E_VCM.vec = C_VCM * E_VCM.vec;

% get the circular components
[E_VCM.rh, E_VCM.lh] = circular_components_fct(E_VCM.vec);

% get the intensity for each component
E_VCM.rh2 = mag2_fct(E_VCM.rh);
E_VCM.lh2 = mag2_fct(E_VCM.lh);
E_VCM.z2 = mag2_fct(E_VCM.z);
E_VCM.E2 = mag2_fct(E_VCM.vec);

% intensity differences
E_diff.lh2 = E.lh2 - E_VCM.lh2;
E_diff.rh2 = E.rh2 - E_VCM.rh2;
E_diff.z2 = E.z2 - E_VCM.z2;
E_diff.E2 = E.E2 - E_VCM.E2;


%% Plots

%% plot all components
%% intensity
% my_surf_fct(X,Y,E.E2,'E2',lambda)
% my_surf_fct(X,Y,E_VCM.E2,'VCM E2',lambda)
%
%% z component
% my_surf_fct(X,Y,real(E.z),'Ez',lambda)
% my_surf_fct(X,Y,real(E_VCM.z),'VCM Ez',lambda)
```

```
%
%% circular components
%% rh
% my_surf_fct(X,Y,real(E.rh),'Erh',lambda)
% my_surf_fct(X,Y,real(E_VCM.rh),'VCM Erh',lambda)
%% lh
% my_surf_fct(X,Y,real(E.lh),'Elh',lambda)
% my_surf_fct(X,Y,real(E_VCM.lh),'VCM Elh',lambda)
%
%% z component intensity
% my_surf_fct(X,Y,E.z2,'Ez2',lambda)
% my_surf_fct(X,Y,E_VCM.z2,'VCM Ez2',lambda)
%
%% circular component intensities
%% rh
% my_surf_fct(X,Y,E.rh2,'Erh2',lambda)
% my_surf_fct(X,Y,E_VCM.rh2,'VCM Erh2',lambda)
%% lh
% my_surf_fct(X,Y,E.lh2,'Elh2',lambda)
% my_surf_fct(X,Y,E_VCM.lh2,'VCM Elh2',lambda)


%% plot radial slices
% fig = figure();
% plot(rho*1e6,real([E_rslice.rh2;E_rslice.lh2;E_rslice.z2;E_rslice.E2]));
% legend('|E_{rh}|^2','|E_{lh}|^2','|E_z|^2','|E|^2');
% xlabel('x / um');
% ylabel('Intensity / a.u.');
% title({'Radial Intensity slice at y=0', ...
%     sprintf('lambda = %.3f nm',lambda*1e9)});
% text(rho(end)*0.5e6,0.9*max(E_rslice.E2), ...
%     sprintf('purity = %.4f',purity));
% outfilename = sprintf('figures/new/E2_rslice_lambda%.3f.fig',lambda*1e9);
% savefig(fig,outfilename,'compact')

%% plot averaged radial distributions
% fig = figure();
% plot(rho*1e6,real([E_avg_rslice.rh2;E_avg_rslice.lh2; ...
%     E_avg_rslice.z2;E_avg_rslice.E2]));
% legend('|E_{rh}|^2','|E_{lh}|^2','|E_z|^2','|E|^2');
% xlabel('\rho / um');
% ylabel('Intensity / a.u.');
% title({'Averaged radial Intensity', ...
%     sprintf('lambda = %.3f nm',lambda*1e9)});
% text(rho(end)*0.5e6,0.9*max(E_avg_rslice.E2), ...
%     sprintf('purity = %.4f',avg_purity));
% outfilename = sprintf( ...
%     'figures/new/E2_averaged_rslice_lambda%.3f.fig',lambda*1e9);
% savefig(fig,outfilename,'compact')

% plot averaged radial distributions with logarithmic y axis
fig = figure();
semilogy(rho*1e6,real([E_avg_rslice.rh2;E_avg_rslice.lh2; ...
```

```
    E_avg_rslice.z2;E_avg_rslice.E2]));
legend('|E_{rh}|^2','|E_{lh}|^2','|E_z|^2','|E|^2');
xlabel('\rho / um');
ylabel('Intensity / a.u.');
title({'Averaged radial Intensity', ...
    sprintf('lambda = %.3f nm',lambda*1e9)});
text(rho(end)*0.2e6,0.5*max(ylim), ...
    sprintf('at rho = 0: purity of RHCP = %.4f',avg_purity));
outfilename = sprintf( ...
    'figures/new/E2_averaged_rslice_ylog_lambda%.3f.fig',lambda*1e9);
savefig(fig,outfilename,'compact')

%% plot integrated radial distributions
% fig = figure();
% plot(rho*1e6,real([E_phiint.rh2;E_phiint.lh2;E_phiint.z2;E_phiint.E2]));
% legend('|E_{rh}|^2','|E_{lh}|^2','|E_z|^2','|E|^2');
% xlabel('x / um');
% ylabel('Intensity / a.u.');
% title({'Integrated radial Intensity', ...
%     sprintf('lambda = %.3f nm',lambda*1e9)});
% outfilename = sprintf( ...
%     'figures/new/E2_integrated_rslice_lambda%.3f.fig',lambda*1e9);
% savefig(fig,outfilename,'compact')


%% plot differences
% my_surf_fct(X,Y,E_diff.E2,'E2 difference',lambda)
%% z component intensity
% my_surf_fct(X,Y,E_diff.z2,'Ez2 difference',lambda)
%
%% circular components
%% rh
% my_surf_fct(X,Y,E_diff.rh2,'Erh2 difference',lambda)
%% lh
% my_surf_fct(X,Y,E_diff.lh2,'Elh2 difference',lambda)
```

6.1.2.3 Helper functions

These are useful functions called in one or more of the scripts listed in the previous subsubsections.

Listing 6.11: average_phi_fct.m

```
%{
average_phi_fct.m
stom ETH Zurich 26.06.2018

Script to average radial slices through the input field over the azimuthal
angle phi.
The output is a vector along the radial direction.

Scalars are usually denoted with small letters, matrices are capitalized.
%}

function [ F_avg_rslice,rho ] = average_phi_fct( F,X,Y )
```

```
    % set the number of points for the radial and angular dimension.
    npts_rho = length(X)*2; % takes the largest dimension
    npts_phi = 201; % take some value

    % assume that the origin is somewhere in the middle
    % find the maximum radial distance which is defined for all angles.
    rho_max = min(abs([X(1,1),X(1,end),Y(1,1),Y(end,1)]));

    % set the polar coordinates
    rho = linspace(0,rho_max,npts_rho);
    phi = linspace(0,2*pi,npts_phi+1);
    phi = phi(1:end-1); % delete last element because it is equal to first

    % add up all slices
    F_avg_rslice = 0;
    for i = 1:length(phi)
        F_avg_rslice = F_avg_rslice + ...
            interp2(X,Y,F,rho*cos(phi(i)),rho*sin(phi(i)));
    end

    % divide by the number of slices taken
    F_avg_rslice = F_avg_rslice/npts_phi;
end
```

Listing 6.12: integrate_phi_fct.m

```
%{
integrate_phi_fct.m
stom ETH Zurich 14.05.2018

Script to integrate the input field over the azimuthal angle phi. The
output is a vector along the radial direction.

Scalars are usually denoted with small letters, matrices are capitalized.
%}

function [ F_rslice,rho ] = integrate_phi_fct( F,X,Y )
    % set the number of points for the radial and angular dimension.
    npts_rho = length(X)*2; % takes the largest dimension
    npts_phi = 201; % take some value

    % assume that the origin is somewhere in the middle
    % find the maximum radial distance which is defined for all angles.
    rho_max = min(abs([X(1,1),X(1,end),Y(1,1),Y(end,1)]));

    % set the polar coordinates
    rho = linspace(0,rho_max,npts_rho);
    phi = linspace(0,2*pi,npts_phi);
    [Rho,Phi] = meshgrid(rho,phi);

    % get the field at the polar coordinates (has to be interpolated)
    F_polar = interp2(X,Y,F,Rho.*cos(Phi),Rho.*sin(Phi));

    % integrate over the angular coordinate phi
```

```
    F_rslice = trapz(Phi(:,1),Rho.*F_polar,1);
end
```

Listing 6.13: circular_components_fct.m

```
%{
circular_components_fct.m
stom ETH Zurich 14.03.2018

Calculating the circular components F_vec_rh and F_vec_lh of a vector field
F_vec by dot product with the circular unit vectors RHhat and LHhat.
%}

function [ F_vec_rh, F_vec_lh ] = circular_components_fct( F_vec )
% there are different conventions, I use this one:
RHhat = 1/sqrt(2) .* [1, -1i];
LHhat = 1/sqrt(2) .* [1, 1i];
F_vec_rh = F_vec(:,:,1) .* RHhat(1) + F_vec(:,:,2) .* RHhat(2);
F_vec_lh = F_vec(:,:,1) .* LHhat(1) + F_vec(:,:,2) .* LHhat(2);
end
```

Listing 6.14: get_normalization_constant_fct.m

```
%{
get_normalization_constant_fct.m
stom ETH Zurich 20.06.2018

Script yields a normalization constant C, which is the inverse of the
magnitude of the input field vector F_vec integrated over the whole
area given by X and Y.

Scalars are usually denoted with small letters, matrices are capitalized.
Matrices describing vectors have a 3rd dimension, whose elements are
the x, y and z components of the vector, respectively.
%}

function [ C ] = get_normalization_constant_fct( X,Y,F_vec )
    integrand = 0;
    for i = 1:size(F_vec,3)
        integrand = integrand + abs(F_vec(:,:,i)).^2;
    end

    C = 1/sqrt(trapz(Y(:,1), trapz(X(1,:),integrand,2),1));
end
```

Listing 6.15: mag2_fct.m

```
%{
mag2_fct.m
stom ETH Zurich 22.03.2018

Magnitude squared of the complex vector field F_vec.
Matrices describing vectors have a 3rd dimension, whose elements are
```

```
the x, y and z components of the vector, respectively.
%}

function [ mag2 ] = mag2_fct( F_vec )
    mag2 = 0;
    for i = 1:size(F_vec,3)
        mag2 = mag2 + abs(F_vec(:,:,i)).^2;
    end
end
```

Listing 6.16: my_surf_fct.m

```
%{
my_surf_fct.m
stom ETH Zurich 28.05.2018

plots the field F as function of X and Y with the plot title made up of
plot_title and the wavelength lambda. Then saves the figure.
%}
function [  ] = my_surf_fct( X,Y,F,plot_title,lambda )
    fig = figure();
    surf(X*1e6,Y*1e6,F)
    shading interp
    title({plot_title,sprintf('lambda = %.3f nm',lambda*1e9)})
    xlabel('x / um');
    ylabel('y / um');

    outfilename = sprintf('figures/new/%s_lambda%.3f.fig', ...
        plot_title,lambda*1e9);
    savefig(fig,outfilename,'compact')
end
```

Listing 6.17: my_variable_surf_fct.m

```
%{
my_surf_fct.m
stom ETH Zurich 20.06.2018

plots the field F as function of X and Y with the plot title made up of
plot_title a variable descibed by its name, its value, its order of
magniture and its unit.
Then saves the figure.
%}
function [  ] = my_variable_surf_fct( X,Y,F,plot_title, ...
    var_name,var_value,var_order_of_magnitude,var_unit )
    fig = figure();
    surf(X*1e6,Y*1e6,F)
    shading interp
    title({plot_title,sprintf('%s = %.2f %s',var_name, ...
        var_value/var_order_of_magnitude,var_unit)})
    xlabel('x / um');
    ylabel('y / um');

    outfilename = sprintf('figures/new/%s_%s%.2f.fig',plot_title, ...
```

```
        var_name,var_value/var_order_of_magnitude);
    savefig(fig,outfilename,'compact')
end
```

Listing 6.18: prepare_figure_for_report_fct.m

```
%{
prepare_figure_for_report_fct.m
stom ETH Zurich 22.06.2018

Deletes the title of the current figure and adds axis labels, or deletes
whole axis, maybe add colorbar, modify as needed.
Is intended for plots of the usual field profiles in the x-y plane.
%}
function [  ] = prepare_figure_for_report_fct(  )
    ax = gca;

    % to make colors 'symmetric':
    clim_max = max(abs(ax.CLim));
    ax.CLim = [-clim_max,clim_max];

    % modify as needed:
    delete(ax.Title);
    colorbar;
%     cb = colorbar();
%     cb.Ruler.Exponent = 0;
    axis equal
    axis off
    view(2) % sets viewpoint from above
%     xlabel('x / um');
%     ylabel('y / um');

    % this is to have tight figure margins:
    outerpos = ax.OuterPosition;
    ti = ax.TightInset;
    left = outerpos(1) + ti(1);
    bottom = outerpos(2) + ti(2);
    ax_width = outerpos(3) - ti(1) - ti(3);
    ax_height = outerpos(4) - ti(2) - ti(4) - 0.05;
    ax.Position = [left bottom ax_width ax_height];
end
```

Listing 6.19: save_current_figure_to_pdf_fct.m

```
%{
save_current_figure_to_pdf_fct.m
stom ETH Zurich 15.06.2018

Script to save the current figure to a pdf with the filename being chosen
via gui.
!!Note that this script will still save the file to the current folder!!

The alternative would be to use the save button in the figure window itself
and then selecting .pdf as type. But that would use A4 as paper size which
```

```
is not what one usually wants.

A note on the resolution of pdf files: a "normal" 2D plot is saved as
vector graphic anyway, which is the advantage of pdfs. However, 3D plots
like generated by the function surf will be saved as bitmaps (except one
forces to take a different renderer, google if needed). For those, one sets
the resolution with the 'r300' option, where 300 is the resolution in dpi.
%}


function [ ] = save_current_figure_to_pdf_fct( )
    h = gcf;
    set(h,'Units','Inches');
    pos = get(h,'Position');
    set(h,'PaperPositionMode','Auto','PaperUnits','Inches', ...
        'PaperSize',[pos(3), pos(4)])

    filename = uiputfile('*');
    print(h,filename,'-dpdf','-r300')
end
```

Listing 6.20: save_current_figure_to_png_fct.m

```
%{
save_current_figure_to_png_fct.m
stom ETH Zurich 15.06.2018

Script to save the current figure to a png with the filename being chosen
via gui.
!!Note that this script will still save the file to the current folder!!
%}


function [ ] = save_current_figure_to_png_fct( )
    h = gcf;
    set(h,'Units','Inches');
    pos = get(h,'Position');
    set(h,'PaperPositionMode','Auto','PaperUnits','Inches', ...
        'PaperSize',[pos(3), pos(4)])

    filename = uiputfile('*');
    print(h,filename,'-dpng','-r300')
end
```

Listing 6.21: save_figure_to_pdf_fct.m

```
%{
save_figure_to_pdf_fct.m
stom ETH Zurich 03.07.2018

Script to save a figure to pdf. The figure is chosen via gui.

A note on the resolution of pdf files: a "normal" 2D plot is saved as
vector graphic anyway, which is the advantage of pdfs. However, 3D plots
```

```
like generated by the function surf will be saved as bitmaps (except one
forces to take a different renderer, google if needed). For those, one sets
the resolution with the 'r300' option, where 300 is the resolution in dpi.
%}


function [ ] = save_figure_to_pdf_fct(  )
    [filename, pathname] = uigetfile;
    filename = sprintf('%s%s',pathname,filename);
    clear pathname
    h = open(filename);
    set(h,'Units','Inches');
    pos = get(h,'Position');
    set(h,'PaperPositionMode','Auto','PaperUnits','Inches', ...
        'PaperSize',[pos(3), pos(4)])

    filename = strrep(filename,'.fig','.pdf');
    print(h,filename,'-dpdf','-r300')
end
```

Listing 6.22: save_figure_to_png_fct.m

```
%{
save_figure_to_png_fct.m
stom ETH Zurich 29.06.2018

Script to save a figure to png. The figure is chosen via gui.
%}


function [ ] = save_figure_to_png_fct(  )
    [filename, pathname] = uigetfile;
    filename = sprintf('%s%s',pathname,filename);
    clear pathname
    h = open(filename);
    set(h,'Units','Inches');
    pos = get(h,'Position');
    set(h,'PaperPositionMode','Auto','PaperUnits','Inches', ...
        'PaperSize',[pos(3), pos(4)])

    filename = strrep(filename,'.fig','.png');
    print(h,filename,'-dpng','-r300')
end
```

6.1.3 Lumerical FDTD Solutions

Listing 6.23: ring_gap.lsf

```
# ring_gap.lsf
# stom ETH Zurich
# 27.04.2018
#
```

```
# script to make a half a ring resonator with a feeding waveguide

clear;
groupscope("::model");
deleteall;

# parameters
path = "matlab_results/";
lambda0 = 0.729e-6; # wavelength
n_waveguide = 2.007;
n_surrounding = 1.454;
n_eff = 1.67; # approximate effective waveguide index
# from source port neff, Karan told me 1.65
gap = 0.13e-6;

# ring properties
ring_radius = 10e-6;
ring_width = 0.45e-6;
ring_height = 0.18e-6;

# waveguide properties
waveguide_span = 28e-6;
waveguide_width = 0.45e-6;
waveguide_height = 0.18e-6;
waveguide_y = ring_radius+0.5*ring_width+gap+0.5*waveguide_width;

# FDTD properties
fdtd_mesh_accuracy = 3;
fdtd_simulation_time = 10*waveguide_span*n_eff/2.9979e8;
fdtd_span = 2*ring_radius+4e-6;
fdtd_ymin = 2e-6;
fdtd_ymax = ring_radius+0.5*ring_width+gap+waveguide_width+1e-6;
fdtd_height = waveguide_height+2e-6;

# port properties
port_x = -(0.5*fdtd_span-0.2e-6);
port_ymin = ring_radius+0.5*ring_width+gap-1e-6;
port_ymax = ring_radius+0.5*ring_width+gap+waveguide_width+1e-6;
port_height = fdtd_height;

# add waveguide
groupscope("::model");
addrect;
select("rectangle");
set("name","waveguide");
set("x",0);
set("x span",waveguide_span);
set("y",waveguide_y);
set("y span",waveguide_width);
set("z",0);
set("z span",waveguide_height);
set("index",2.007);
```

```
# add half ring
groupscope("::model");
addobject("180_bend_wg");
set("x",0);
set("y",0);
set("z",0);
set("radius",ring_radius);
set("base width",ring_width);
set("base angle",90);
set("base height",ring_height);
set("material","<Object defined dielectric>");
groupscope("::model::bend_180");
selectall;
set("material","<Object defined dielectric>");
set("index",2.007);
unselectall;

# add FDTD region
groupscope("::model");
addfdtd;
set("background index",n_surrounding);
set("mesh accuracy",fdtd_mesh_accuracy);
set("z min bc","Symmetric"); # using the vertical symmetry to reduce
    calculation time
set("simulation time",fdtd_simulation_time);
set("x",0);
set("x span",fdtd_span);
set("y min",fdtd_ymin);
set("y max",fdtd_ymax);
set("z",0);
set("z span",fdtd_height);

# add ports
addport;
groupscope("::model::FDTD");
select("ports");
set("source port","port 1");
set("source mode","mode 1");
set("monitor frequency points",500);
setglobalsource("wavelength start",lambda0);
setglobalsource("wavelength stop",lambda0);
groupscope("::model::FDTD::ports");
select("port 1");
set("x",port_x);
set("y min",port_ymin);
set("y max",port_ymax);
set("z",0);
set("z span",port_height);
set("injection axis","x-axis");
set("direction","Forward");
set("mode selection","fundamental mode");
set("frequency points",1);
```

```
# add monitors
groupscope("::model");
addtime; # field time monitor for 1st port
set("name","t_in");
set("x",port_x);
set("y",(port_ymin+port_ymax)/2);
copy; # for 2nd port
set("name","t_through");
set("x",-port_x);
addprofile; # frequency-domain field monitor
set("name","full_profile");
set("x",0);
set("x span",fdtd_span);
set("y min",fdtd_ymin);
set("y max",fdtd_ymax);
set("z",0);

## run simulation, usually do this by hand because otherwise the program
    crashes
#run;

# get and save results
E = getresult("full_profile","E");
Ex = pinch(E.Ex);
Ey = pinch(E.Ey);
Ez = pinch(E.Ez);
x = pinch(E.x);
y = pinch(E.y);
filename = path+"E_field_gap"+num2str(gap*1e6);
filename = filename+"_ma"+num2str(fdtd_mesh_accuracy)+".mat";
matlabsave(filename,x,y,Ex,Ey,Ez,waveguide_y);
```

Listing 6.24: straight_grating_sweep.lsf

```
# straight_grating_sweep.lsf
# stom ETH Zurich
# 18.04.2018
#
# Based on straight_grating.lsf, which is a
# script to make a straight waveguide with a
# grating on one side. Here it is sweeped over
# different grating strengths, that is tooth
# heights, here called tooth thicknesses

clear;
closeall;
groupscope("::model");
deleteall;

# parameters
path = "matlab_results/";
lambda0 = 0.729e-6; # wavelength
n_waveguide = 2.007;
n_surrounding = 1.454;
```

```
n_eff = 1.67; # approximate effective waveguide index
# from source port neff, Karan told me 1.65

# waveguide properties
waveguide_span = 110e-6;
waveguide_width = 0.45e-6;
waveguide_height = 0.18e-6;

# grating properties
grating_span = 100e-6;
grating_period = lambda0/n_eff;
duty_cycle = 0.5; # to maximize radiation
tooth_length = duty_cycle*grating_period;
tooth_thicknesses = 1e-6*[0.005:0.0025:0.015;0.02;0.03:0.01:0.21];
N_grating = ceil(grating_span/grating_period);

# FDTD properties
fdtd_mesh_accuracy = 3;
fdtd_simulation_time = 10*waveguide_span*n_eff/2.9979e8;
fdtd_span = grating_span+4e-6;
fdtd_width = waveguide_width+1.5e-6;
fdtd_height = waveguide_height+1.5e-6;


# loop over different tooth thicknesses
for (k=1:length(tooth_thicknesses)){
        tooth_thickness = tooth_thicknesses(k); # "h" in filenames

        # switch back to layout mode
        switchtolayout;

        groupscope("::model");
        deleteall;

        # add waveguide
        groupscope("::model");
        addrect;
        select("rectangle");
        set("name","waveguide");
        set("x",0);
        set("x span",waveguide_span);
        set("y",0);
        set("y span",waveguide_width);
        set("z",0);
        set("z span",waveguide_height);
        set("index",2.007);

        # add grating
        groupscope("::model");
        addstructuregroup;
        set("name","grating");
        set("x",0);
        set("y",0);
```

```
        set("z",0);
        groupscope("::model::grating");
        for(i=1:N_grating){
            x_tooth = -0.5*grating_span+(i-0.5)*grating_period;
            addrect;
            select("rectangle");
            set("x",x_tooth);
            set("y min",(-0.5*waveguide_width-tooth_thickness));
            set("y max",(-0.5*waveguide_width));
            set("name","reactangle"+num2str(i));
        }
        selectall;
        set("x span",tooth_length);
        set("z",0);
        set("z span",waveguide_height);
        set("index",2.007);

        unselectall;

        # add FDTD region
        groupscope("::model");
        addfdtd;
        set("background index",n_surrounding);
        set("mesh accuracy",fdtd_mesh_accuracy);
        set("z min bc","Symmetric"); # use the vertical symmetry to reduce
            calculation time
        set("simulation time",fdtd_simulation_time);
        set("x",0);
        set("x span",fdtd_span);
        set("y",0);
        set("y span",fdtd_width);
        set("z",0);
        set("z span",fdtd_height);

        # add ports
        addport;
        groupscope("::model::FDTD");
        select("ports");
        set("source port","port 1");
        set("source mode","mode 1");
        set("monitor frequency points",500);
        setglobalsource("wavelength start",lambda0);
        setglobalsource("wavelength stop",lambda0);
        groupscope("::model::FDTD::ports");
        select("port 1");
        set("x",-(0.5*fdtd_span-0.2e-6));
        set("y",0);
        set("y span",fdtd_width);
        set("z",0);
        set("z span",fdtd_height);
        set("injection axis","x-axis");
        set("direction","Forward");
        set("mode selection","fundamental mode");
```

```
        set("frequency points",1);

        # add monitors
        groupscope("::model");
        addtime; # field time monitor for 1st port
        set("name","t_in");
        set("x",-(0.5*fdtd_span-0.2e-6));
        set("y",0);
        copy; # for 2nd port
        set("name","t_through");
        set("x",(0.5*fdtd_span-0.2e-6));
        addprofile; # frequency-domain field monitor
        set("name","full_profile");
        set("x",0);
        set("x span",fdtd_span);
        set("y",0);
        set("y span",fdtd_width);
        set("z",0);
        setglobalmonitor("frequency points",1);

        # run simulation
        run;

        # get and save results
        E = getresult("full_profile","E");
        Ex = pinch(E.Ex);
        Ey = pinch(E.Ey);
        Ez = pinch(E.Ez);
        E2 = pinch(E.E2);
        x = pinch(E.x);
        y = pinch(E.y);
        #image(x*1e6,y*1e6,E2,"x / um","y / um","E2");
        filename = path+"E_field_h"+num2str(tooth_thickness*1e6);
        filename = filename+"_ma"+num2str(fdtd_mesh_accuracy)+".mat";
        matlabsave(filename,x,y,Ex,Ey,Ez);


}
```

Listing 6.25: ring_resonator_v2.lsf

```
# ring_resonator_v2.lsf
# stom ETH Zurich
# 07.05.2018
#
# based on ring_resonator.lsf
#
# Includes just a layer of oxide above, then going into air. This
# resembles the geometry of the structure (but without substrate).
#
# additionally includes the far field calculations, that is a call
# to ff.lsf.
#
# script to make a simple ring resonator with a feeding
```

```
# waveguide and a grating inside the ring

clear;
groupscope("::model");
deleteall;

# parameters
path = "results/new_results/";
lambda0 = 0.729e-6; # wavelength
n_eff = 1.67; # approximate effective waveguide index
# from source port neff, Karan told me 1.65

# ring properties
ring_radius = 10e-6;
ring_width = 0.45e-6;
ring_height = 0.18e-6;
gap = 0.13e-6;

# waveguide properties
waveguide_material_index = 2.007;
waveguide_span = 28e-6;
waveguide_width = 0.45e-6;
waveguide_height = 0.18e-6;
waveguide_y = ring_radius+0.5*ring_width+gap+0.5*waveguide_width;

# grating properties
grating_period = lambda0/n_eff;
duty_cycle = 0.5; # to maximize radiation
tooth_length = duty_cycle*grating_period;
tooth_thickness = 1e-6*0.044;
N_grating = ceil(2*pi*ring_radius/grating_period)-1;
phi_grating = linspace(0,2*pi,(N_grating+1));
# delete last element because it is the same as the first (0=2pi)
phi_grating = phi_grating(1:N_grating);
radius_grating = ring_radius -0.5*ring_width -0.5*tooth_thickness;

# oxide properties
oxide_index = 1.454;
oxide_thickness = 1e-6;

# monitor properties
ring_monitor_name = "monitor_ring";
ff_monitor_name = "monitor_air";
ff_monitor_z = 0.5*waveguide_height+oxide_thickness+0.3e-6;
monitor_lambda_center = 725e-9;
monitor_lambda_span = 20e-9;
monitor_frequency_points = 242;

# FDTD properties
fdtd_mesh_accuracy = 3;
# simulation time: factor 100 is to ensure that the shutoff condition is
    reached
fdtd_simulation_time = 100*2*pi*ring_radius*n_eff/2.9979e8;
```

```
fdtd_span = 2*ring_radius+6e-6;
fdtd_ymin = -(ring_radius+0.5*ring_width+2e-6);
fdtd_ymax = ring_radius+0.5*ring_width+gap+waveguide_width+2e-6;
fdtd_zmin = -(0.5*waveguide_height+1e-6);
fdtd_zmax = ff_monitor_z+0.3e-6;

# air properties
air_index = 1;
air_span = waveguide_span;
air_ymin = fdtd_ymin - 2e-6;
air_ymax = fdtd_ymax + 2e-6;
air_zmin = 0.5*waveguide_height+oxide_thickness;
air_zmax = fdtd_zmax + 2e-6;

# port properties
port_x = -(0.5*fdtd_span-0.2e-6);
port_ymin = ring_radius+0.5*ring_width+gap-1e-6;
port_ymax = ring_radius+0.5*ring_width+gap+waveguide_width+1e-6;
port_zmin = fdtd_zmin;
port_zmax = fdtd_zmax;


# add waveguide
groupscope("::model");
addrect;
select("rectangle");
set("name","waveguide");
set("x",0);
set("x span",waveguide_span);
set("y",waveguide_y);
set("y span",waveguide_width);
set("z",0);
set("z span",waveguide_height);
set("index",waveguide_material_index);

# add two half rings
groupscope("::model");
addobject("180_bend_wg");
set("name","ring_top");
set("x",0);
set("y",0);
set("z",0);
set("radius",ring_radius);
set("base width",ring_width);
set("base angle",90);
set("base height",ring_height);
set("material","<Object defined dielectric>");
groupscope("::model::ring_top");
selectall;
set("material","<Object defined dielectric>");
set("index",waveguide_material_index);
unselectall;
groupscope("::model");
```

```
select("ring_top");
copy;
set("name","ring_bottom");
set("first axis","z");
set("rotation 1",180);


# add grating
groupscope("::model");
addstructuregroup;
set("name","grating");
set("x",0);
set("y",0);
set("z",0);
groupscope("grating");
for(i=1:N_grating){
        x_grating = radius_grating*cos(phi_grating(i));
        y_grating = radius_grating*sin(phi_grating(i));
        addrect;
        select("rectangle");
        set("x",x_grating);
        set("y",y_grating);
        set("first axis","z");
        set("rotation 1",(180*phi_grating(i)/pi));
        set("name","tooth"+num2str(i));
}
selectall;
set("x span",tooth_thickness);
set("y span",tooth_length);
set("z",0);
set("z span",ring_height);
set("index",waveguide_material_index);

# add air
groupscope("::model");
addrect;
select("rectangle");
set("name","air");
set("x",0);
set("x span",air_span);
set("y min",air_ymin);
set("y max",air_ymax);
set("z min",air_zmin);
set("z max",air_zmax);
set("index",air_index);


# add FDTD region
groupscope("::model");
addfdtd;
set("background index",oxide_index);
set("mesh accuracy",fdtd_mesh_accuracy);
set("simulation time",fdtd_simulation_time);
```

```
set("x",0);
set("x span",fdtd_span);
set("y min",fdtd_ymin);
set("y max",fdtd_ymax);
set("z min",fdtd_zmin);
set("z max",fdtd_zmax);

# add ports
addport;
groupscope("::model::FDTD::ports");
select("port 1");
set("name","port_through");
set("x",-port_x);
set("y min",port_ymin);
set("y max",port_ymax);
set("z min",port_zmin);
set("z max",port_zmax);
set("injection axis","x-axis");
set("direction","Backward");
set("mode selection","fundamental mode");
set("frequency points",1);
copy;
set("name","port_in");
set("x",port_x);
set("direction","Forward");
groupscope("::model::FDTD");
select("ports");
set("source port","port_in");
set("source mode","mode 1");
set("monitor frequency points",500);
setglobalsource("wavelength start",lambda0);
setglobalsource("wavelength stop",lambda0);

# note on ports: somehow the input port is not determined by the "source
    port"
# option, but by which port was added last. At least that is how to explain
# the results of some small tests.

# add monitors
groupscope("::model");
addtime; # field time monitor for 1st port
set("name","t_in");
set("x",port_x);
set("y",(port_ymin+port_ymax)/2);
copy; # for 2nd port
set("name","t_through");
set("x",-port_x);
addprofile; # frequency-domain field monitor in the waveguide plane
set("name",ring_monitor_name);
set("x",0);
set("x span",fdtd_span);
set("y min",2*waveguide_y-fdtd_ymax);
set("y max",fdtd_ymax);
```

```
set("z",0);
#addprofile; # frequency-domain field monitor in the air
## uses too much memory with mesh accuracy 3, that is the reason for v2.2
#set("name",ff_monitor_name);
#set("x",0);
#set("x span",fdtd_span);
#set("y min",fdtd_ymin);
#set("y max",fdtd_ymax);
#set("z",ff_monitor_z);
setglobalmonitor("use source limits",0);
setglobalmonitor("use linear wavelength spacing",1);
setglobalmonitor("wavelength center",monitor_lambda_center);
setglobalmonitor("wavelength span",monitor_lambda_span);
setglobalmonitor("frequency points",monitor_frequency_points);

## run simulation, usually do this by hand because otherwise the program
    crashes
#run;

# get and save results of monitors
E_ring = getresult(ring_monitor_name,"E");
#E_air = getresult(ff_monitor_name,"E");
# x,y and lambda are the same for both monitors
lambda = pinch(E_ring.lambda);
matlabsave(path+"monitor_data/lambda",lambda);
x = pinch(E_ring.x);
matlabsave(path+"monitor_data/x",x);
y = pinch(E_ring.y);
matlabsave(path+"monitor_data/y",y);
matlabsave(path+"monitor_data/waveguide_y",waveguide_y);

# ring monitor
Ex_ring = pinch(E_ring.Ex);
Ey_ring = pinch(E_ring.Ey);
Ez_ring = pinch(E_ring.Ez);
filename = path+"monitor_data/ring_monitor_gap"+num2str(gap*1e6);
filename = filename+"_h"+num2str(tooth_thickness*1e6)+".mat";
filename = filename+"_ma"+num2str(fdtd_mesh_accuracy)+".mat";
matlabsave(filename,x,y,lambda,Ex_ring,Ey_ring,Ez_ring,waveguide_y);

## far field monitor (in air)
#Ex_air = pinch(E_air.Ex);
#Ey_air = pinch(E_air.Ey);
#Ez_air = pinch(E_air.Ez);
#filename = path+"monitor_data/air_monitor_gap"+num2str(gap*1e6);
#filename = filename+"_h"+num2str(tooth_thickness*1e6);
#filename = filename+"_ma"+num2str(fdtd_mesh_accuracy)+".mat";
#matlabsave(filename,x,y,lambda,Ex_air,Ey_air,Ez_air,waveguide_y);

# save fields at individual wavelengths
for (m = 1:length(lambda)){
    monitor_lambda = lambda(m);

```

```
    Ex = pinch(Ex_ring,3,m);
    Ey = pinch(Ey_ring,3,m);
    Ez = pinch(Ez_ring,3,m);
    filename = path+"E_ring/E_ring_lambda"+num2str(monitor_lambda*1e9)+".
        mat";
    matlabsave(filename,Ex,Ey,Ez);

    #Ex = pinch(Ex_air,3,m);
    #Ey = pinch(Ey_air,3,m);
    #Ez = pinch(Ez_air,3,m);
    #filename = path+"E_air/E_air_lambda"+num2str(monitor_lambda*1e9)+".mat
        ";
    #matlabsave(filename,Ex,Ey,Ez);
}

# clear the workspace variables so that they are not confused
clear(Ex,Ey,Ez,filename,m);

## calculate and save the far field
## !! frequency point is chosen in script ff.lsf!!
#ff.lsf;
```

Listing 6.26: ff.lsf

```
# ff.lsf
# stom ETH Zurich
# 30.04.2018
#
# script to calculate the far field radiation from
# a field monitor using the farfieldexact3d function.
# uses the existing workspace variables after a run of ring_resonator_v2.
    lsf
# or ring_resonator_v2.2.lsf


path_E_ff = path+"E_ff/";

# choose the resonance/mode according to its mN value
mN = -2;
monitor = air_monitor_name+num2str(-mN+3); # !adapt this depending of v2 or
    v2.2!

# choose frequency point
frequency_point = 11;
E = getresult(monitor,"E");
obs_lambda = pinch(E.lambda(frequency_point));
clear(E);

# observer plane properties
obs_res_x = 101;
obs_res_y = 100;
obs_x = 1e-6*linspace(-10,10,obs_res_x);
obs_y = 1e-6*linspace(-10,10,obs_res_y);
obs_z = 50e-6;
```

```
# calculate the far field
# Note: By default takes the refractive index at the monitor center
# as the propagation refractive index. This is correct for us since
# the monitor is in air.
E_ff = farfieldexact3d(monitor,obs_x,obs_y,obs_z,frequency_point);

# extract field components
Ex_ff = pinch(E_ff,4,1);
Ey_ff = pinch(E_ff,4,2);
Ez_ff = pinch(E_ff,4,3);
E2_ff = abs(Ex_ff)^2 + abs(Ey_ff)^2 + abs(Ez_ff)^2;

for (l = 1:length(obs_z)){
    image_title = "Intensity at z = "+num2str(obs_z(l)*1e6)+" um";
    image(obs_x*1e6,obs_y*1e6,E2_ff,"x (um)","y (um)",image_title);
    filename_ff = path+"E_ff/E_ff_gap"+num2str(gap*1e6);
    filename_ff = filename_ff+"_h"+num2str(tooth_thickness*1e6);
    filename_ff = filename_ff+"_ma"+num2str(fdtd_mesh_accuracy);
    filename_ff = filename_ff+"_z"+num2str(obs_z(l)*1e6);
    filename_ff = filename_ff+"_lambda"+num2str(obs_lambda*1e9)+".mat";
    matlabsave(filename_ff,obs_x,obs_y,Ex_ff,Ey_ff,Ez_ff);
}
```

Listing 6.27: ring_resonator_v2.2.lsf

```
# ring_resonator_height_sweep_v2.2.lsf
# stom ETH Zurich
# 22.05.2018
#
# based on ring_resonator_v2.lsf
#
# includes multiple monitors to record data only at the resonance
    frequencies.
#
# Includes just a layer of oxide above, then going into air. This
# resembles the true geometry of the structure.
#
# additionally includes the far field calculations, that is a call
# to ff.lsf.
#
# script to make a simple ring resonator with a feeding
# waveguide and a grating inside the ring
#
# usually after loops, clear the workspace variables so that they are not
    confused

clear;
groupscope("::model");
deleteall;

# parameters
path = "results/new_results/";
lambda0 = 0.729e-6; # wavelength
```

```
n_eff = 1.67; # approximate effective waveguide index
# from source port neff, Karan told me 1.65

# ring properties
ring_radius = 10e-6;
ring_width = 0.45e-6;
ring_height = 0.18e-6;
gap = 0.13e-6;

# waveguide properties
waveguide_material_index = 2.007;
waveguide_span = 28e-6;
waveguide_width = 0.45e-6;
waveguide_height = 0.18e-6;
waveguide_y = ring_radius+0.5*ring_width+gap+0.5*waveguide_width;

# grating properties
grating_period = lambda0/n_eff;
duty_cycle = 0.5; # to maximize radiation
tooth_length = duty_cycle*grating_period;
tooth_thickness = 1e-6*0.044;
N_grating = ceil(2*pi*ring_radius/grating_period)-1;
phi_grating = linspace(0,2*pi,(N_grating+1));
# delete last element because it is the same as the first (0=2pi)
phi_grating = phi_grating(1:N_grating);
radius_grating = ring_radius -0.5*ring_width -0.5*tooth_thickness;

# oxide properties
oxide_index = 1.454;
oxide_thickness = 1e-6;

# monitor properties
ring_monitor_name = "monitor_ring";
air_monitor_name = "monitor_air";
air_monitor_z = 0.5*waveguide_height+oxide_thickness+0.3e-6;
# set wavelengths manually, according to the transmission spectrum obtained
    from v2
ring_monitor_lambda_center = 1e-9*[716.5,720.5,724.55,728.7,732.86];
ring_monitor_lambda_span = 2e-9*[0.5,0.5,1,0.5,0.5];
ring_monitor_frequency_points = 2*[10,10,20,10,10];
number_ring_monitors = length(ring_monitor_lambda_center);
air_monitor_lambda_center = ring_monitor_lambda_center;
air_monitor_lambda_span = ring_monitor_lambda_span;
air_monitor_frequency_points = ring_monitor_frequency_points;
number_air_monitors = length(air_monitor_lambda_center);

# FDTD properties
fdtd_mesh_accuracy = 3;
#fdtd_simulation_time = 40e-15;
fdtd_simulation_time = 100*2*pi*ring_radius*n_eff/2.9979e8;
fdtd_span = 2*ring_radius+6e-6;
fdtd_ymin = -(ring_radius+0.5*ring_width+2e-6);
fdtd_ymax = ring_radius+0.5*ring_width+gap+waveguide_width+2e-6;
```

```
fdtd_zmin = -(0.5*waveguide_height+1e-6);
fdtd_zmax = air_monitor_z+0.3e-6;

# air properties
air_index = 1;
air_span = waveguide_span;
air_ymin = fdtd_ymin - 2e-6;
air_ymax = fdtd_ymax + 2e-6;
air_zmin = 0.5*waveguide_height+oxide_thickness;
air_zmax = fdtd_zmax + 2e-6;

# port properties
port_x = -(0.5*fdtd_span-0.2e-6);
port_ymin = ring_radius+0.5*ring_width+gap-1e-6;
port_ymax = ring_radius+0.5*ring_width+gap+waveguide_width+1e-6;
port_zmin = fdtd_zmin;
port_zmax = fdtd_zmax;


# add waveguide
groupscope("::model");
addrect;
select("rectangle");
set("name","waveguide");
set("x",0);
set("x span",waveguide_span);
set("y",waveguide_y);
set("y span",waveguide_width);
set("z",0);
set("z span",waveguide_height);
set("index",waveguide_material_index);

# add two half rings
groupscope("::model");
addobject("180_bend_wg");
set("name","ring_top");
set("x",0);
set("y",0);
set("z",0);
set("radius",ring_radius);
set("base width",ring_width);
set("base angle",90);
set("base height",ring_height);
set("material","<Object defined dielectric>");
groupscope("::model::ring_top");
selectall;
set("material","<Object defined dielectric>");
set("index",waveguide_material_index);
unselectall;
groupscope("::model");
select("ring_top");
copy;
set("name","ring_bottom");
```

```
set("first axis","z");
set("rotation 1",180);


# add grating
groupscope("::model");
addstructuregroup;
set("name","grating");
set("x",0);
set("y",0);
set("z",0);
groupscope("grating");
for(i=1:N_grating){
        x_grating = radius_grating*cos(phi_grating(i));
        y_grating = radius_grating*sin(phi_grating(i));
        addrect;
        select("rectangle");
        set("x",x_grating);
        set("y",y_grating);
        set("first axis","z");
        set("rotation 1",(180*phi_grating(i)/pi));
        set("name","tooth"+num2str(i));
}
selectall;
set("x span",tooth_thickness);
set("y span",tooth_length);
set("z",0);
set("z span",ring_height);
set("index",waveguide_material_index);

# add air
groupscope("::model");
addrect;
select("rectangle");
set("name","air");
set("x",0);
set("x span",air_span);
set("y min",air_ymin);
set("y max",air_ymax);
set("z min",air_zmin);
set("z max",air_zmax);
set("index",air_index);


# add FDTD region
groupscope("::model");
addfdtd;
set("background index",oxide_index);
set("mesh accuracy",fdtd_mesh_accuracy);
set("simulation time",fdtd_simulation_time);
set("x",0);
set("x span",fdtd_span);
set("y min",fdtd_ymin);
```

```
set("y max",fdtd_ymax);
set("z min",fdtd_zmin);
set("z max",fdtd_zmax);

# add ports
addport;
groupscope("::model::FDTD::ports");
select("port 1");
set("name","port_through");
set("x",-port_x);
set("y min",port_ymin);
set("y max",port_ymax);
set("z min",port_zmin);
set("z max",port_zmax);
set("injection axis","x-axis");
set("direction","Backward");
set("mode selection","fundamental mode");
set("frequency points",1);
copy;
set("name","port_in");
set("x",port_x);
set("direction","Forward");
groupscope("::model::FDTD");
select("ports");
set("source port","port_in");
set("source mode","mode 1");
set("monitor frequency points",500);
setglobalsource("wavelength start",715e-9);
setglobalsource("wavelength stop",735e-9);
# setting this global source to a broader range instead  of a single point (as
# in v2) will not change the pulse dramatically, but allows to take a transmission
# spectrum from the ports, as they are apparently recording only at these
# wavelengths.

# note on ports: somehow the input port is not determined by the "source port"
# option, but by which port was added last. At least that is how to explain
# the results of some small tests.

# add monitors
groupscope("::model");
addtime; # field time monitor for 1st port
set("name","t_in");
set("x",port_x);
set("y",(port_ymin+port_ymax)/2);
copy;
set("name","t_through");
set("x",-port_x);

# monitors in the waveguide plane for different wavelengths/resonances
for (k = 1:number_ring_monitors){
```

```
    addprofile;
    set("name",ring_monitor_name+num2str(k));
    set("x",0);
    set("x span",fdtd_span);
    set("y min",fdtd_ymin);
    set("y max",fdtd_ymax);
    set("z",0);
    set("override global monitor settings",1);
    set("use linear wavelength spacing",1);
    set("wavelength center",ring_monitor_lambda_center(k));
    set("wavelength span",ring_monitor_lambda_span(k));
    set("frequency points",ring_monitor_frequency_points(k));
    set("spatial interpolation","nearest mesh cell");
}
clear(k);

# monitors in the air for different wavelengths/resonances
for (k = 1:number_air_monitors){
    addprofile;
    set("name",air_monitor_name+num2str(k));
    set("x",0);
    set("x span",fdtd_span);
    set("y min",fdtd_ymin);
    set("y max",fdtd_ymax);
    set("z",air_monitor_z);
    set("override global monitor settings",1);
    set("use linear wavelength spacing",1);
    set("wavelength center",air_monitor_lambda_center(k));
    set("wavelength span",air_monitor_lambda_span(k));
    set("frequency points",air_monitor_frequency_points(k));
    set("spatial interpolation","nearest mesh cell");
}
clear(k);

## run simulation, usually do this by hand because otherwise the program
    crashes
#run;



## save results

matlabsave(path+"monitor_data/waveguide_y",waveguide_y);

# ring monitors
for (k = 1:number_ring_monitors){
    # monitor data
    E_ring = getresult(ring_monitor_name+num2str(k),"E");
    lambda_ring = pinch(E_ring.lambda);
    matlabsave(path+"monitor_data/lambda_ring"+num2str(k),lambda_ring);
    x_ring = pinch(E_ring.x);
    matlabsave(path+"monitor_data/x_ring",x_ring);
    y_ring = pinch(E_ring.y);
```

```
    matlabsave(path+"monitor_data/y_ring",y_ring);

    # fields
    Ex_ring = pinch(E_ring.Ex);
    Ey_ring = pinch(E_ring.Ey);
    Ez_ring = pinch(E_ring.Ez);
    filename = path+"monitor_data/ring"+num2str(k)+"_monitor";
    filename = filename+"_gap"+num2str(gap*1e6);
    filename = filename+"_h"+num2str(tooth_thickness*1e6);
    filename = filename+"_ma"+num2str(fdtd_mesh_accuracy)+".mat";
    matlabsave(filename,x_ring,y_ring,lambda_ring,Ex_ring,Ey_ring,Ez_ring,
        waveguide_y);

    # save fields at individual wavelengths
    for (m=1:length(lambda_ring)){
        monitor_lambda = lambda_ring(m);

        # frequency/wavelength is 4th dimension
        Ex = pinch(E_ring.Ex,4,m);
        Ey = pinch(E_ring.Ey,4,m);
        Ez = pinch(E_ring.Ez,4,m);
        filename = path+"E_ring/E_ring"+num2str(k);
        filename = filename+"_lambda"+num2str(monitor_lambda*1e9)+".mat";
        matlabsave(filename,Ex,Ey,Ez);
    }
}
clear(lambda_ring,Ex_ring,Ey_ring,Ez_ring,monitor_lambda,Ex,Ey,Ez,filename,
    k);

# air monitors
for (k=1:number_air_monitors){

    # monitor data
    E_air = getresult(air_monitor_name+num2str(k),"E");
    lambda_air = pinch(E_air.lambda);
    matlabsave(path+"monitor_data/lambda_air"+num2str(k),lambda_air);
    x_air = pinch(E_air.x);
    matlabsave(path+"monitor_data/x_air",x_air);
    y_air = pinch(E_air.y);
    matlabsave(path+"monitor_data/y_air",y_air);

    # fields
    Ex_air = pinch(E_air.Ex);
    Ey_air = pinch(E_air.Ey);
    Ez_air = pinch(E_air.Ez);
    filename = path+"monitor_data/air"+num2str(k)+"_monitor";
    filename = filename+"_gap"+num2str(gap*1e6);
    filename = filename+"_h"+num2str(tooth_thickness*1e6);
    filename = filename+"_ma"+num2str(fdtd_mesh_accuracy)+".mat";
    matlabsave(filename,x_air,y_air,lambda_air,Ex_air,Ey_air,Ez_air,
        waveguide_y);

    # save fields at individual wavelengths
```

```
    for (m = 1:length(lambda_air)){
        monitor_lambda = lambda_air(m);

        # frequency/wavelength is 4th dimension
        Ex = pinch(E_air.Ex,4,m);
        Ey = pinch(E_air.Ey,4,m);
        Ez = pinch(E_air.Ez,4,m);
        filename = path+"E_air/E_air"+num2str(k);
        filename = filename+"_lambda"+num2str(monitor_lambda*1e9)+".mat";
        matlabsave(filename,Ex,Ey,Ez);
    }
}
clear(lambda_air,Ex_air,Ey_air,Ez_air,monitor_lambda,Ex,Ey,Ez,filename,k);


## calculate and save the far field
## !! frequency point is chosen in script ff.lsf!!
#ff.lsf;
```

Listing 6.28: coupling_fraction.lsf

```
# coupling_fraction.lsf
# stom ETH Zurich
# 29.06.2018
#
# based on Karan's integrate_power.lsf
#
# script to determine the power radiated through specific monitors
# at specific wavelengths (set by monitor numbers and frequency
# points)

#clear;
#closeall;
monitor_number = [1,2,3,3,4,5];
frequency_points = [10,10,19,23,10,11];
for (i = 1:length(monitor_number)){
    monitor_name = "monitor_air"+num2str(monitor_number(i));
    P_ring = getdata(monitor_name,"power");
    P_ring = P_ring(frequency_points(i));
    f = getdata(monitor_name,"f");
    P_input = sourcepower(f(frequency_points(i)));
    ?"Coupling fraction: "+num2str(P_ring/P_input);
    clear(P_ring,P_input);
}
```