NTNU
Norwegian University of
Science and Technology

Photonic crystal light emitting diode

Erlend Leirset

Master of Science in Electronics

Submission date:	March Astrid Aksnes, IET
Supervisor:	Helge Engan, IET
Co-supervisor:	

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

Light extraction from light emitting diodes (LED) is very much limited by total internal reflection because semiconductors usually have very high refractive index. At our department Gallium Antimonide emitting LEDs have been grown using molecular beam epitaxy.

This thesis explores the use of photonic crystals to reduce the effect of total internal reflection from Gallium Antimonide LED. Numerical simulations of a LED with photonic crystal top layer will therefore be conducted.

To perform the simulations an electromagnetic model of the grown LED will be developed based on physical dimensions and material parameters estimated from previous research. The modeling of the photonic crystal is to be performed with finite difference time domain (FDTD) methods. The thesis will treat photonic crystal theory necessary for utilizing FDTD for this problem. To evaluate the physical performance of the simulated structures, the results are to be in the form of light extraction efficiency and radiation pattern. The simulation methods including post processing will be documented. As the simulations are computationally demanding, a method for distributing the load over several nodes must be used.

Assignment given: 09. March 2010
Supervisor: Astrid Aksnes, IET

Abstract

This master's thesis describe electromagnetic simulations of a gallium antimonide (GaSb) light emitting diode, LED. A problem for such devices is that most of the generated light is reflected from the surface due to total internal reflection, and is therefore prevented from coupling out of the semiconductor material. Etching out a 2D photonic crystal grating on the LED surface would put aside the absolute rule of total internal reflection, and could therefore be used to increase the total transmission. The simulation method which was developed was supposed to find geometry parameters for the photonic crystal to optimize the light extraction. A set of plane waves were therefore simulated using FDTD to build an equivalent to the Fresnel equations for the photonic crystal surface. From that the total transmittance and radiation patterns for the simulated geometries were calculated. The results indicated an increase in the transmission properties of up to 70% using a square grating of holes where the holes have a radius of 0.5μm, the hole depth is 0.4μm, and the grating constant is 1μm. A hexagonal grating of holes and a square grating of isotropically etched holes were also simulated, and featured improvements on the same scale, but with different dimensions for the holes. The simulations were computationally very demanding, and the simulation structure therefore had to be highly trimmed to limit the calculation time to reasonable values. This might have reduced the accuracy of the results. Especially the optimum grating constant, and the value of the optimum improvement itself is believed to be somewhat inaccurate.

Preface

This is a master's thesis which is the final product from the master program at NTNU. The work has been conducted at the Electro-Optics group at the Department of Electronics and Telecommunication. The topic is simulation of photonic crystals for improving the performance of infrared light emitting diodes. Working with such a popular research field as photonic crystals has been very interesting. Although working in a lab is what I like best, building up a working simulation method was also very satisfactory. Also the experience gained when doing this work will most probably be very useful for many kinds of electromagnetic problems. It has been challenging to fully understand and utilize the underlying theory, and it has been very demanding to get and maintain an overview over all the different aspects of the simulation method. Although demanding it has also, mostly for the same reasons, been very interesting to use general electromagnetic and wave theory to implement a method for solving a concrete problem. A practical issue that caused much concern was the enormous amounts of data generated by the simulations. Handling and compressing the thousands of gigabytes generated in an efficient manner was a significant part of the exercise.

The exercise was specified by PhD student Tron Arne Nielsen, whom was working with photonic crystals in the materials in question. I would like to thank him for helping me get started, and especially for helping me with the needed semiconductor physics. Since this was something I knew little about, it was highly beneficial to get help understanding it. The simulations required the computational power of several computers. I would therefore like to thank Senior Engineer Karl Andrew Iversen and Head Engineer Dag Nummedal for letting me use the best computers in their student computer rooms. I would also like to thank co adviser Professor Helge E. Engan for helping me construct a model describing the light generated inside the light emitting diode. Finally I would like to thank my adviser Associate Professor Astrid Aksnes. She has helped me very much with everything from strategic decisions on what software to use to the actual writing of this thesis.

Glossary

FDTD Finite-difference time-domain. Simulation method which steps through time and uses a discretized space grid.

GaInSb Gallium indium antimonide. Semiconductor material.

GaSb Gallium antimonide. Semiconductor material.

LED Light Emitting Diode. Diode where recombination of charge carriers is radiative.

MEEP MIT Electromagnetic Equation Propagation. Simulation software that uses the FDTD-method.

PBC Periodic boundary conditions. Boundary conditions for a simulation cell which effectively simulates a photonic crystal where the unit cell is given by the simulation cell. The $\boldsymbol{k_B}$-vector must be specified for such simulations.

PML Perfectly matched layer. A layer in a simulation cell which insulates the interior from the boundary conditions of the edges.

Contents

Chapter 1

Introduction

This thesis describes numerical simulations of the light extraction properties of a Gallium Antimonide (GaSb) light emitting diode, LED, with a 2D photonic crystal grating on its surface. LEDs are usually made of semiconductor material which has a very high refractive index. This causes a very small critical angle, and consequently most of the light experiences total internal reflection at the surface. As will be explained a photonic crystal can refract light, that otherwise would be completely reflected, to an angle that is smaller than the critical angle. Thereby can the total transmission through the surface be increased. The common way of reducing this problem is to place an epoxy dome, with higher refractive index than air, over the LED. The method described here could be used in conjunction with such a dome by increasing the coupling from the semiconductor to the epoxy material. The simulations are however performed for a semiconductor to air interface because this is the easiest to fabricate. They are therefore a step towards a viable design where the preliminary goal is to develop a working simulation model which relate well to measurements on actual samples. At the time of startup on the work described in this thesis, the LED structure had been grown by PhD student Tron Arne Nielsen, using molecular beam epitaxy, MBE. The surface was however not patterned, and simulations were needed to find a photonic crystal which would significantly increase the light extraction. The objective of the simulations were therefore to find an optimal photonic crystal for fabrication. Also gaining experience and knowledge about photonic crystals and methods for simulating their electromagnetic properties were important motivational factors.

The second chapter in this thesis describes some general theory regarding photonic crystals that is needed in order to understand the simulations. The third chapter treats the necessary theory behind LEDs, and directly analyses the actual LED assumed in these simulations. The purpose of this is mainly to establish the properties of the light incident on the photonic crystal surface. As will be explained these properties affect the extraction efficiency of the surface, and are therefore needed. The light extraction of a LED without a photonic crystal is also found in this chapter. This is the reference which the extraction efficiencies of the photonic crystals is later measured against. The fourth chapter presents the simulation method which has been developed in this thesis. First the necessary theory which describes the light extraction process is explained. Later sections describe, on basis of the physical theory, how numerical simulations are used to find the desired parameters describing the extraction efficiency. Also the actual simulations that are performed are presented in this chapter. That is the exact geometries and the parameters which describe them. In the fourth chapter, the simulation software which has been used is presented. The usage of the simulation applied tool and the scripts made for running large series of simulations on several computers are described. Also much of the post processing code that incorporates the developed simulation method is presented here. Following is a chapter that shows an essential subset of the simulation results. Finally a discussion of the results, their credibility, and suggestions for further work is stated before a conclusion with regards to the results and the simulation method is drawn.

Chapter 2

Photonic crystal theory

2.1 Introduction

A photonic crystal is a structure where the refractive index varies periodically in one, two, or three dimensions as illustrated in Figure 2.1.1. The realization of such structures could for instance be etched out structures in semiconductor or dielectric solids. In this case the refractive index will vary between the refractive index of air and of the employed solid. If desirable, the etched out matter can in turn be replaced by another material using various techniques.

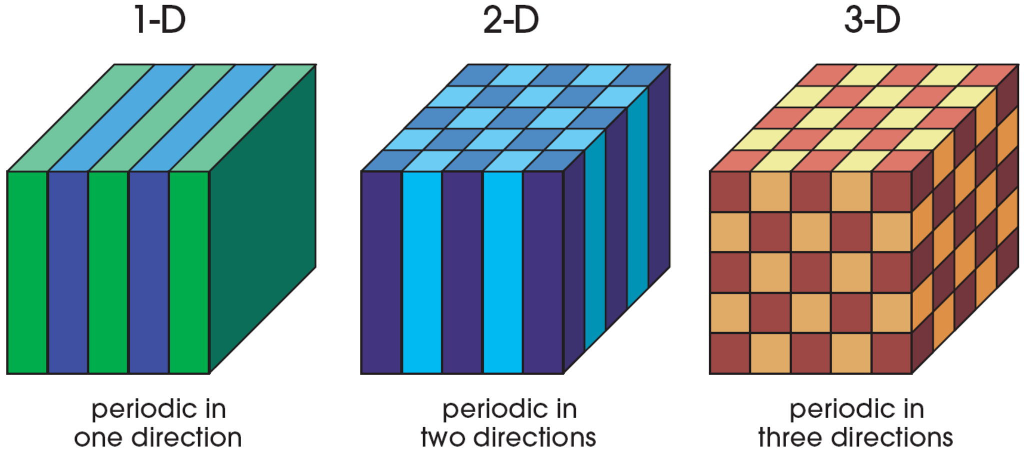


Figure 2.1.1: Illustration of photonic crystals structures [1].

Photonic crystals can be used for many purposes where the most known is to create a photonic band gap. That is a periodic structure where light at a certain

[1] Figure 2.1.1 is from ref. [12]

wavelength range is unable to propagate. One example of this is fiber Bragg gratings which reflect light within a narrow band. This is an example of a finite 1D photonic crystal where the light within a certain band is unable to propagate in the grating and is instead reflected. Photonic crystals in two and three dimensions can have similar properties, but they are much more complicated. The principle of photonic band gap is not really relevant for this thesis and will therefore not be discussed. Instead the light diffraction in 2D photonic crystals is primarily the property which is utilized. In order to clarify this phenomenon, the following sections will describe the relevant theory for light propagation in photonic crystals. The most important concepts are the scaling properties of the Maxwell equations and the effect of Bloch symmetry. The theory in chapter 2 will mainly be based on the first three chapters of [12].

2.2 Eigenvalue equation

The basis of light propagation in photonic crystals starts, as for all electromagnetic problems, with the Maxwell equations which are stated in equations (2.2.1) through (2.2.4). Although not stated explicitly, the parameters in these equations are dependent on both time and position.

$$\nabla\times\boldsymbol{E_t} + \frac{\partial \boldsymbol{B_t}}{\partial t} = 0 \tag{2.2.1}$$

$$\nabla\times\boldsymbol{H_t} - \frac{\partial \boldsymbol{D_t}}{\partial t} = \boldsymbol{J_t} \tag{2.2.2}$$

$$\nabla\boldsymbol{D_t} = \rho \tag{2.2.3}$$

$$\nabla\boldsymbol{B_t} = 0 \tag{2.2.4}$$

$\boldsymbol{E_t}$	Electric field vector
$\boldsymbol{H_t}$	Magnetic field vector
$\boldsymbol{B_t}$	Magnetic flux density
$\boldsymbol{D_t}$	Electric flux density
$\boldsymbol{J_t}$	Electric current density
ρ	Electric charge per unit volume (space charge)

The following derivation will assume a linear, isotropic, non absorbing, non magnetic, source free, but not necessarily homogeneous medium. This will give the relations stated in equations (2.2.5) to (2.2.8).

$$\boldsymbol{D_t} = \varepsilon_0 \varepsilon(\boldsymbol{r}) \boldsymbol{E_t} \tag{2.2.5}$$
$$\boldsymbol{B_t} = \mu_0 \boldsymbol{E_t} \tag{2.2.6}$$
$$\boldsymbol{J_t} = 0 \tag{2.2.7}$$
$$\rho = 0 \tag{2.2.8}$$

ε_0	Electrical vacuum permittivity ($\approx 8.85 \cdot 10^{-12}$ F/m)
ε	Relative electrical permittivity. The value is dependent on position
μ_0	Magnetic vacuum permeability ($= 4\pi \cdot 10^{-7}$ H/m)
$\boldsymbol{r}$	Position vector

Only stationary solutions will be considered, which means solutions where the only time dependence is a complex harmonic oscillation at a certain frequency. As explained in [12], this does not lead to loss of generality because any solution can be constructed from a linear combination of harmonic solutions using Fourier transform techniques. This assumption is stated in mathematical terms for $\boldsymbol{E_t}$ and $\boldsymbol{H_t}$ in equations (2.2.9) and (2.2.10) respectively.

$$\boldsymbol{E_t}(\boldsymbol{r}, t) = \boldsymbol{E}(\boldsymbol{r}) \mathrm{e}^{-j\omega t} \tag{2.2.9}$$
$$\boldsymbol{H_t}(\boldsymbol{r}, t) = \boldsymbol{H}(\boldsymbol{r}) \mathrm{e}^{-j\omega t} \tag{2.2.10}$$

ω	Angular frequency.
t	Time in seconds.
$\boldsymbol{E}(\boldsymbol{r})$	Electric field distribution. Includes both amplitude and phase information
$\boldsymbol{H}(\boldsymbol{r})$	Magnetic field distribution. Includes both amplitude and phase information

Solutions to the Maxwell equations which are of the form of equations (2.2.9) and (2.2.10) are called eigenmodes, modes, or sometimes states. The field distribution in the medium is determined by the vector functions $\boldsymbol{E}(\boldsymbol{r})$ and $\boldsymbol{H}(\boldsymbol{r})$, and the time dependence is determined by the ω-factor in the complex exponential function. Inserting the assumed solutions in equations (2.2.9) and (2.2.10) into the Maxwell equations and using the assumptions in equation (2.2.5) to (2.2.8), transforms the Maxwell equations into equations (2.2.11) to (2.2.14). The $\boldsymbol{r}$-dependence is here not stated explicitly for the fields.

$$\nabla \times \boldsymbol{E} = j\omega\mu_0 \boldsymbol{H} \tag{2.2.11}$$
$$\nabla \times \boldsymbol{H} = -j\omega\varepsilon_0 \varepsilon(\boldsymbol{r}) \boldsymbol{E} \tag{2.2.12}$$
$$\nabla (\varepsilon(\boldsymbol{r})\boldsymbol{E}) = 0 \tag{2.2.13}$$
$$\nabla \boldsymbol{H} = 0 \tag{2.2.14}$$

As stated in [12], the $\boldsymbol{H}$-field can be decoupled from the $\boldsymbol{E}$-field using the following procedure. Divide equation (2.2.12) by $\varepsilon(\boldsymbol{r})$, and take the curl. Equation (2.2.11) can now be used to eliminate $\boldsymbol{E}$. The result is here stated in equation (2.2.15). Note that the factor $1/(\varepsilon_0\mu_0)$ has been replaced with c^2.

$$\nabla\times\left(\frac{1}{\varepsilon(\boldsymbol{r})}\nabla\times\boldsymbol{H}\right)=\left(\frac{\omega}{c}\right)^2\boldsymbol{H} \tag{2.2.15}$$

c Speed of light in vacuum.

Equation (2.2.15) together with equation (2.2.14), expresses all the requirements for the $\boldsymbol{H}$-field for a mode of the structure defined by $\varepsilon(\boldsymbol{r})$. The $\boldsymbol{E}$-field can subsequently be found using equation (2.2.12). The left side of equation (2.2.15) defines an operation on $\boldsymbol{H}$, and the equation as a whole states that the result of this operation must be equal to a scaled version of the original $\boldsymbol{H}$-field. Such an equation is called an *eigenvalue equation* because its solutions $\boldsymbol{H}$ are *eigenfunctions* or *eigenmodes* of the operation on the left side. This can be stated more precisely by defining an *operator* $\hat{\Theta}$, which is defined by equation (2.2.16).

$$\hat{\Theta}\boldsymbol{V}(\boldsymbol{r})=\nabla\times\left(\frac{1}{\varepsilon(\boldsymbol{r})}\nabla\times\boldsymbol{V}(\boldsymbol{r})\right) \tag{2.2.16}$$

$V(\boldsymbol{r})$ Arbitrary vector function.

Using this operator definition, equation (2.2.15) can be expressed as equation (2.2.17).

$$\hat{\Theta}\boldsymbol{H}=\left(\frac{\omega}{c}\right)^2\boldsymbol{H} \tag{2.2.17}$$

The solution to equation (2.2.17) are the eigenfunctions of the operator $\hat{\Theta}$, and the corresponding *eigenvalues* are the required value of the factor $\left(\frac{\omega}{c}\right)^2$.

2.3 Uniform media

This section will use the theory described in section 2.2 to analyze the electromagnetic properties of uniform media. Since these are well known the purpose is mainly to give a familiar example of the nomenclature and the general theory in section 2.2. The aim is to better illustrate the differences and similarities between the analyses of this known structure and more advanced photonic crystal structures.

Equation (2.2.15) is valid for all possible structures defined by a dielectric constant $\boldsymbol{\varepsilon}(\boldsymbol{r})$. It is not limited to photonic crystals or any other structure. In particular it applies to a uniform medium where $\boldsymbol{\varepsilon}(\boldsymbol{r})$ is constant. For such a medium the $\hat{\Theta}$-operator in equation (2.2.16) simplifies to equation (2.3.1), and equation (2.2.15) simplifies to equation (2.3.2).

$$\hat{\Theta}\boldsymbol{V}(\boldsymbol{r}) = \frac{1}{\varepsilon_r}\nabla\times\nabla\times\boldsymbol{V}(\boldsymbol{r}) \tag{2.3.1}$$

ε_r Dielectric constant in uniform medium.

$$\frac{1}{\varepsilon_r}\nabla\times\nabla\times\boldsymbol{H} = \left(\frac{\omega}{c}\right)^2\boldsymbol{H} \tag{2.3.2}$$

Using appropriate vector identities equation (2.3.2) can be further simplified to the known Helmholtz equation here stated in equation (2.3.3).

$$\frac{1}{\varepsilon_r}\boldsymbol{\nabla}^2\boldsymbol{H} = \left(\frac{\omega}{c}\right)^2\boldsymbol{H} \tag{2.3.3}$$

Solutions which satisfy equation (2.3.3) and (2.2.14) are plane waves which can be expressed by equation (2.3.4) provided that $\boldsymbol{H}_0\perp\boldsymbol{k}$.

$$\boldsymbol{H} = \boldsymbol{H}_0\,\mathrm{e}^{j\boldsymbol{kr}} \tag{2.3.4}$$

$\boldsymbol{H_0}$ Vector constant
$\boldsymbol{k}$ Wave vector

Inserting the assumed solution in equation (2.3.4) into the eigenvalue equation (2.2.15) verifies the solutions, and gives a relation between the possible eigenfunctions $\boldsymbol{H}$, and their corresponding eigenvalues $\left(\frac{\omega}{c}\right)^2$. This relation is the known dispersion relation for uniform media here stated in equation (2.3.5).

$$\frac{|\boldsymbol{k}|^2}{\varepsilon_r} = \left(\frac{\omega}{c}\right)^2 \tag{2.3.5}$$

A more physical way of describing this result is that the *modes of a uniform medium* are plane waves, and have frequencies ω equal to $\frac{|\boldsymbol{k}|c}{\sqrt{\varepsilon_r}}$.

2.4 Scaling properties

The $\hat{\Theta}$-operator defined in equation (2.2.16) has many useful properties whereof most are thoroughly described in [12]. This section will describe the scaling properties because these are of significant importance when interpreting the analysis which will be described in later chapters. These properties are not related to a particular structure, and therefore apply for an arbitrary $\varepsilon(\boldsymbol{r})$. As the name suggests, the property describes the relation between the modes of two structures which are scaled copies of each other. To find this relation two structures $\varepsilon_1(\boldsymbol{r})$ and $\varepsilon_2(\boldsymbol{r})$ are defined such that $\varepsilon_2(\boldsymbol{r}) = \varepsilon_1(\boldsymbol{r}/s)$, or equivalently $\varepsilon_2(s\boldsymbol{r}) = \varepsilon_1(\boldsymbol{r})$. This means that the structure described by $\varepsilon_2(\boldsymbol{r})$ is the structure described by $\varepsilon_1(\boldsymbol{r})$ magnified by a factor s. The eigenvalue equation for $\varepsilon_2(\boldsymbol{r})$ is stated in equation (2.4.1). The spatial dependence of the field is here specified because it is relevant for the coming derivation.

$$\nabla\times\left(\frac{1}{\varepsilon_2(\boldsymbol{r})}\nabla\times\boldsymbol{H}_2(\boldsymbol{r})\right) = \left(\frac{\omega_2}{c}\right)^2 \boldsymbol{H}_2(\boldsymbol{r}) \tag{2.4.1}$$

The position vector $\boldsymbol{r}$ will be substituted by a primed vector $\boldsymbol{r}' = \frac{\boldsymbol{r}}{s}$. To specify that the differentiation implied by the curl operators is with respect to $\boldsymbol{r}'$, they will also be primed. Using regular substitution rules, the primed curl operator can loosely be said to satisfy the equality $\nabla\times = \frac{1}{s}\nabla'\times$. This changes equation (2.4.1) into equation (2.4.2).

$$\frac{1}{s}\nabla'\times\left(\frac{1}{\varepsilon_2(s\boldsymbol{r}')}\frac{1}{s}\nabla'\times\boldsymbol{H}_2(s\boldsymbol{r}')\right) = \left(\frac{\omega_2}{c}\right)^2 \boldsymbol{H}_2(s\boldsymbol{r}') \tag{2.4.2}$$

Recognizing that $\varepsilon_2(s\boldsymbol{r}')$ in equation (2.4.2) can be replaced by $\varepsilon_1(\boldsymbol{r}')$, and moving the s-factor to the right side changes the equation into equation (2.4.3).

$$\nabla'\times\left(\frac{1}{\varepsilon_1(\boldsymbol{r}')}\nabla'\times\boldsymbol{H}_2(s\boldsymbol{r}')\right) = \left(\frac{s\omega_2}{c}\right)^2 \boldsymbol{H}_2(s\boldsymbol{r}') \tag{2.4.3}$$

Equation (2.4.3) is now of the same form as the eigenvalue equation for the ε_1-structure. This means that the valid solutions for one of the structures is also a valid solution for the other provided the scaling relations in equations (2.4.4) to (2.4.6) are used.

$$\varepsilon_1(\boldsymbol{r}) = \varepsilon_2(s\boldsymbol{r}) \tag{2.4.4}$$

$$\boldsymbol{H}_1(\boldsymbol{r}) = \boldsymbol{H}_2(s\boldsymbol{r}) \tag{2.4.5}$$

$$\omega_1 = s\omega_2 \tag{2.4.6}$$

Ref. [12] mentions that this property is very useful when testing new structures. They can then be fabricated on a larger scale, and the tests can be done with low frequency microwaves instead of waves with optical frequencies. This can substantially ease the fabrication of such test structures since it can be done on a much larger scale than would be required otherwise. In this thesis the property will be used the other way around. If a measurement or a simulation can be performed over a broad frequency spectrum, this can be utilized to obtain a result for a range of scaling factors. One example is if a result is desired only for a single wavelength, and for a range of structures that are simple scaled copies of each other. In such a context, it is desirable to express the length scale of the desired structure based on the length scale of the tested or simulated structures and the two wavelengths involved. Based on equation (2.4.4) and (2.4.6) such a relation can be obtained, and is here stated in equation (2.4.7).

$$l_2 = l_1 \cdot \frac{\lambda_2}{\lambda_1} \tag{2.4.7}$$

l_1	Length of arbitrary feature in ε_1-structure
l_2	Length of corresponding feature in ε_2-structure
λ_1	Vacuum wavelength of light in ε_1-structure
λ_2	Vacuum wavelength of light in ε_2-structure

According to [12], equation (2.2.15) also has a similar property regarding scaling of the dielectric constant ε. This is however not relevant for the structure in mind because it contains air which has an invariable dielectric constant. If this was not the case, the results might have been relevant also for other materials than what will be discussed here.

2.5 Waves in periodic media

2.5.1 Periodic media

In the previous sections light propagation has been discussed for media of arbitrary form. In this section photonic crystal structures will be the primary topic. In the beginning of this chapter photonic crystals were defined as structures with a periodic dielectric constant. Mathematically this can be defined by equation (2.5.1). The following analyses in this section will assume that $\varepsilon(\boldsymbol{r})$ satisfies this equation.

$$\varepsilon(\boldsymbol{r}) = \varepsilon(\boldsymbol{r} + \ell_1\boldsymbol{a}_1 + \ell_2\boldsymbol{a}_2 + \ell_3\boldsymbol{a}_3) \tag{2.5.1}$$

ℓ_i Arbitrary integer constants
$\boldsymbol{a}_i$ Primitive lattice vectors. These are specific for a particular photonic crystals. ($i = 1, 2, 3$)

The primitive lattice vectors $\boldsymbol{a}_i$, determine the spatial period and direction of the periodicity of the structure as illustrated in Figure 2.5.1. When they are equally long, their length is often denoted as the lattice constant a. For periodicity in one or two dimensions only one or two lattice vectors are needed. Equation (2.5.1) can however still be valid if the spare vectors are set to zero. The *unit cell* is also a useful notion when dealing with periodic structures. As indicated in Figure 2.5.2, this is a segment which if replicated infinitely in all directions will result in the entire structure.

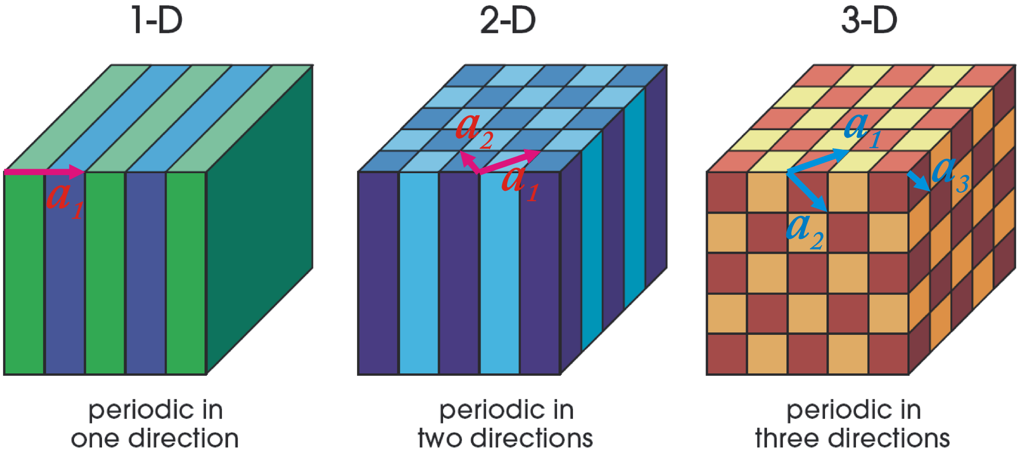
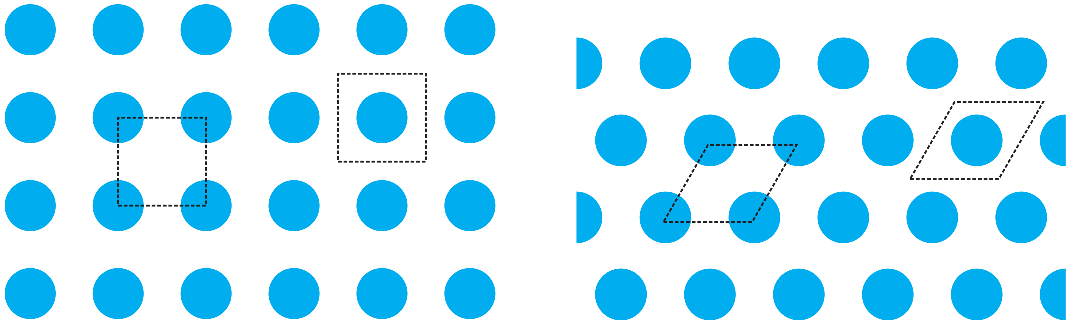

1-D
2-D
3-D
a_1
a_2
a_1
a_1
a_3
a_2
periodic in one direction
periodic in two directions
periodic in three directions

Figure 2.5.1: Examples of lattice vectors [2].

2.5.2 Reciprocal space

It is useful to look at the spatial frequencies of the structure $\varepsilon(\boldsymbol{r})$. Its Fourier transform holds this information, and will here be denoted $\breve{\varepsilon}(\boldsymbol{k})$. Because of the periodicity of $\varepsilon(\boldsymbol{r})$, $\breve{\varepsilon}(\boldsymbol{k})$ must be zero everywhere except at certain points in $\boldsymbol{k}$-space. On these points the function has an infinite value in a way such that a definite integral over the dot is finite. They are therefore a three dimensional version of Dirac's delta function. This concept is just a generalization of the concept that the only frequency components of a periodic function are the function's

[2] Figure 2.5.1 is from ref. [12]



(a) Square grating (b) Hexagonal grating

Figure 2.5.2: Examples of unit cells in two 2D photonic crystal structures. Both the stippled shapes can represent the unit cell.

actual frequency and its harmonics. Although the value at the different points in general is different, their placement must according to [12] form a periodic structure in $\boldsymbol{k}$-space. This structure is called the reciprocal lattice, and the $\boldsymbol{k}$-space is called the reciprocal space. The reciprocal lattice can also be described by primitive lattice vectors, which are called the reciprocal lattice vectors, and have the symbol $\boldsymbol{b}_i$. A photonic crystal has the same number of primitive reciprocal lattice vectors $\boldsymbol{b}_i$ as primitive lattice vectors $\boldsymbol{a}_i$. The relation between these is derived in [12], Appendix B. The result from this derivation is that they must satisfy equation (2.5.2).

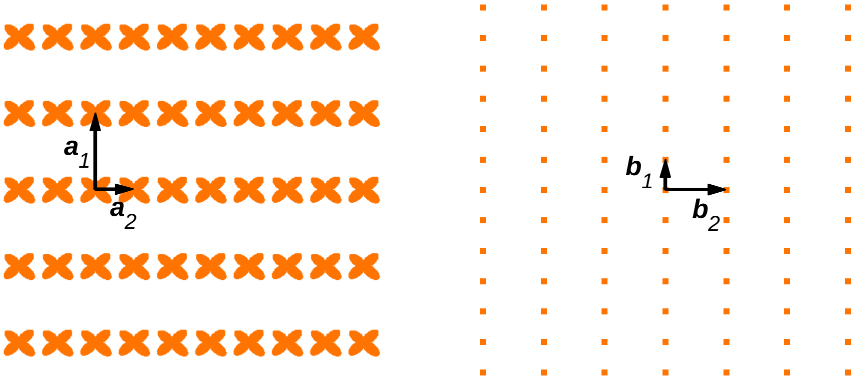
$$\boldsymbol{a}_i \cdot \boldsymbol{b}_k = 2\pi\delta_{ik} \tag{2.5.2}$$

$\boldsymbol{b}_j$ Primitive reciprocal lattice vectors $(j = 1, 2, 3)$
δ_{ik} Kronecker delta function. (1 if $i = k$, 0 otherwise).

The $\boldsymbol{b}_i$-vectors must be orthogonal to all but one $\boldsymbol{a}_i$-vector and vice versa. If the $\boldsymbol{a}_i$-vectors are mutually orthogonal like the ones illustrated in Figure 2.5.1, then the $\boldsymbol{b}_i$-vectors stated by equation (2.5.3) satisfy equation (2.5.2). Here the $\boldsymbol{b}_i$-vectors are parallel to the $\boldsymbol{a}_i$-vectors, and their lengths are inversely dependent on each other. Also their units are the inverse of each other.

$$\boldsymbol{b}_i = 2\pi\frac{\boldsymbol{a}_i}{\boldsymbol{a}_i \cdot \boldsymbol{a}_i} \tag{2.5.3}$$

Figure 2.5.3 illustrates such a structure in two dimensions. The reciprocal lattice is here calculated from a finite grating using the fft-function in Matlab. This causes the points in the reciprocal lattice to have a finite width and a finite value. The actual structures which will be considered in this thesis can all be regarded as such a rectangular grating, and equation (2.5.3) will therefore be sufficient.



(a) Rectangular grating (b) Reciprocal lattice of (a)

Figure 2.5.3: Example of a 2D rectangular grating and its reciprocal lattice

2.5.3 Bloch states

To analyze how a periodic medium affects the modes a new operator $\hat{T}_{ai}$ is defined in equation (2.5.4). It operates on functions of position in space, and shifts the argument by the lattice vector $\boldsymbol{a}_i$.

$$\hat{T}_{ai} f(\boldsymbol{r}) = f(\boldsymbol{r} + \boldsymbol{a}_i) \tag{2.5.4}$$

$f(\boldsymbol{r})$ Arbitrary function of position

Note that when this operator is operated on $\varepsilon(\boldsymbol{r})$ in equation (2.5.1), the result is by definition equal to $\varepsilon(\boldsymbol{r})$. In equation (2.5.5) this operator is used on the definition of the $\hat{\Theta}$-operator from equation (2.2.16) when the dielectric constant is periodic. It is here evident that the two operators *commute*, which means their order can be swapped without changing the result.

$$\begin{aligned}\hat{T}_{ai}\hat{\Theta}H(\boldsymbol{r}) &= \nabla\times\left(\frac{1}{\varepsilon(\boldsymbol{r}+\boldsymbol{a}_i)}\nabla\times\boldsymbol{H}(\boldsymbol{r}+\boldsymbol{a}_i)\right)\\ &= \nabla\times\left(\frac{1}{\varepsilon(\boldsymbol{r})}\nabla\times\boldsymbol{H}(\boldsymbol{r}+\boldsymbol{a}_i)\right)\\ &= \hat{\Theta}\hat{T}_{ai}H(\boldsymbol{r}) \end{aligned} \tag{2.5.5}$$

In equation (2.5.6), the $\hat{T}_{ai}$-operator has been used on equation (2.2.17) together with the commutative property with the $\hat{T}_{ai}$-operator.

$$\begin{aligned}\hat{T}_{\boldsymbol{a}i}\hat{\Theta}\boldsymbol{H} &= \hat{T}_{\boldsymbol{a}i}\left(\frac{\omega}{c}\right)^2\boldsymbol{H}\\ \hat{\Theta}\hat{T}_{\boldsymbol{a}i}\boldsymbol{H} &= \left(\frac{\omega}{c}\right)^2\hat{T}_{\boldsymbol{a}i}\boldsymbol{H}\end{aligned} \tag{2.5.6}$$

Equation (2.5.6) states that if $\boldsymbol{H}$ are modes of the system, then $\hat{\boldsymbol{T}}_{\boldsymbol{a}i}\boldsymbol{H}$ must also be modes of the same system. Since both equations (2.5.6) and (2.2.17) are valid for *all* modes of the system, then $\boldsymbol{H}$ and $\hat{\boldsymbol{T}}_{\boldsymbol{a}i}\boldsymbol{H}$ must be the same mode. This again means that one of them can be expressed as a scaled version of the other, which is expressed in equation (2.5.7).

$$\hat{T}_{\boldsymbol{a}i}\boldsymbol{H} = \alpha_{\boldsymbol{a}i}\boldsymbol{H} \tag{2.5.7}$$

$\alpha_{\boldsymbol{a}i}$ Arbitrary complex scaling constant

Equation (2.5.7) can be used recursively by applying the $\hat{\boldsymbol{T}}_{\boldsymbol{a}i}$-operator to it, and replacing $\hat{\boldsymbol{T}}_{\boldsymbol{a}i}\boldsymbol{H}$ on the right side by $\alpha_{\boldsymbol{a}i}\boldsymbol{H}$. This means equation (2.5.7) can be generalized to equation (2.5.8).

$$\left(\hat{T}_{\boldsymbol{a}i}\right)^l\boldsymbol{H} = (\alpha_{\boldsymbol{a}i})^l\boldsymbol{H} \tag{2.5.8}$$

l Integer which determines how many $\boldsymbol{a}_i$ the $\boldsymbol{r}$-argument is shifted .

Looking at equation (2.5.8), it is apparent that $|\alpha_{\boldsymbol{a}i}| = 1$ in order to avoid a diverging field. $|\alpha_{\boldsymbol{a}i}|$ is therefore restricted to a pure phase constant. Put in a different way, the mode field in two neighbouring unit cells must be identical except for a constant phase shift. Such a field which is a solution of equation (2.5.7) is shown in equation (2.5.9). Here $\boldsymbol{u}_{k_{Bi}}(\boldsymbol{r})$ is a periodic function which is unaffected by the $\hat{\boldsymbol{T}}_{\boldsymbol{a}i}$-operator, and $\alpha_{\boldsymbol{a}i} = \mathrm{e}^{jk_{Bi}\boldsymbol{b}_i\cdot\boldsymbol{r}}$

$$\boldsymbol{H} = \boldsymbol{u}_{k_{Bi}}(\boldsymbol{r})\cdot\mathrm{e}^{jk_{Bi}\boldsymbol{b}_i\cdot\boldsymbol{r}} \tag{2.5.9}$$

$\boldsymbol{u}_{k_{Bi}}(\boldsymbol{r})$	Vector function that is periodic with the structure.
$\boldsymbol{b}_i$	Primitive reciprocal lattice vector.
k_{Bi}	Real constant.

The derivation above has taken into account the periodicity in one arbitrary dimension numbered $\boldsymbol{i}$. If the same procedure is repeated for the two remaining dimensions, then equation (2.5.9) transforms into equation (2.5.10). Here the three k_{Bi}-constants are represented by the vector $\boldsymbol{k_B}$.

$$\boldsymbol{H} = \boldsymbol{u}_{\boldsymbol{k_B}}(\boldsymbol{r}) \cdot \mathrm{e}^{j\boldsymbol{k_B}\cdot\boldsymbol{r}} \tag{2.5.10}$$

$\boldsymbol{k}_B$ Vector constant ($= k_{B1}\boldsymbol{b_1} + k_{B2}\boldsymbol{b_2} + k_{B3}\boldsymbol{b_3}$)

The fact that the field can be expressed as the product of a plane wave and a periodic function with the same periodicity as the structure is called Bloch's theorem. The solutions $\boldsymbol{H}$, which have this property in three dimensions are called Bloch states, and the vector constant here labeled $\boldsymbol{k_B}$ is called a Bloch wave vector. Take note that until now, there is not any limitation to what value $\boldsymbol{k_B}$ can hold, and the function $\boldsymbol{u}_{\boldsymbol{k_B}}(\boldsymbol{r})$ is dependent on this value.

2.5.4 Discrete bands

The previous section showed how the form of the allowed solution must be. This section will consider the effect this form has on the eigenvalue equation, and find some relations between the defined parameter $\boldsymbol{k}_B$, and a corresponding eigenvalue, or frequency ω. For a given value of $\boldsymbol{k}_B$, $\boldsymbol{H}$ from equation (2.5.10), can be inserted into the eigenvalue equation in equation (2.2.15). Because $\boldsymbol{k}_B$ is given, the periodic function $\boldsymbol{u}_{\boldsymbol{k_B}}(\boldsymbol{r})$ is the only degree of freedom for the $\boldsymbol{H}$-field. Since this is a periodic function the only real degree of freedom is within one unit cell of the photonic crystal. For 3D photonic crystals this unit cell will have a finite volume, and its modes will be similar to that of cavities. These only have modes at discrete frequencies. The eigenvalue equation only has solutions for a non continuous discrete set of eigenvalues or frequencies ω. It is therefore possible to number these solutions or modes with increasing ω where the mode one has the lowest frequency. 1D and 2D photonic crystals will have the same property if the propagation is in the line or plane of the crystal. This means that the $\boldsymbol{H}$-field is constant in the directions perpendicular to the crystal.

2.5.5 Brillouin zone

In section 2.5.3, there were not any limitations to the values of $\boldsymbol{k}_B$. Although this is true, it will in this section be shown that it is only *necessary* to consider a limited range of $\boldsymbol{k}_B$-vectors. Because $\boldsymbol{u}_{\boldsymbol{k_B}}$ in equation (2.5.10) is periodic in the same way as the structure itself, its Fourier transform consists of distinct points as discussed in section 2.5.2. The inverse transform is therefore a Fourier series summing the contribution of all these points. Using this property, equation (2.5.10) can be expressed as equation (2.5.11).

$$\boldsymbol{H} = \mathrm{e}^{j\boldsymbol{k_B}\cdot\boldsymbol{r}} \sum_{m_1}\sum_{m_2}\sum_{m_3} \breve{\boldsymbol{u}}_{\boldsymbol{k_B}}(m_1, m_2, m_3)\, \mathrm{e}^{j(m_1\boldsymbol{b}_1+m_2\boldsymbol{b}_2+m_3\boldsymbol{b}_3)\cdot\boldsymbol{r}} \tag{2.5.11}$$

$\breve{\boldsymbol{u}}_{\boldsymbol{k_B}}$ Fourier constants for $\boldsymbol{u}_{\boldsymbol{k_B}}$
m_i Summing indices (i=1,2,3)
$\boldsymbol{b}_i$ Reciprocal lattice vectors of the structure (i=1,2,3)

$\boldsymbol{H}$ in equation (2.5.11) can be inserted into the eigenvalue equation to solve for an arbitrary value of $\boldsymbol{k_B}$. If a value of $\boldsymbol{k_{Bp}} = \boldsymbol{k_{B0}} + p\boldsymbol{b_1}$ were to be chosen where p is an integer, then equation (2.5.11) can be written as equation (2.5.12).

$$\begin{aligned}\boldsymbol{H} &= \mathrm{e}^{j(\boldsymbol{k_{B0}}+p\boldsymbol{b}_1)\cdot\boldsymbol{r}} \sum_{m_1}\sum_{m_2}\sum_{m_3} \breve{\boldsymbol{u}}_{\boldsymbol{k_{Bp}}}(m_1, m_2, m_3)\, \mathrm{e}^{j(m_1\boldsymbol{b}_1+m_2\boldsymbol{b}_2+m_3\boldsymbol{b}_3)\cdot\boldsymbol{r}} \\ &= \mathrm{e}^{j\boldsymbol{k_{B0}}\cdot\boldsymbol{r}} \sum_{m_1}\sum_{m_2}\sum_{m_3} \breve{\boldsymbol{u}}_{\boldsymbol{k_{Bp}}}(m_1, m_2, m_3)\, \mathrm{e}^{j((m_1+p)\boldsymbol{b}_1+m_2\boldsymbol{b}_2+m_3\boldsymbol{b}_3)\cdot\boldsymbol{r}} \\ &= \mathrm{e}^{j\boldsymbol{k_{B0}}\cdot\boldsymbol{r}} \sum_{m_1}\sum_{m_2}\sum_{m_3} \breve{\boldsymbol{u}}_{\boldsymbol{k_{Bp}}}(m_1 - p, m_2, m_3)\, \mathrm{e}^{j(m_1\boldsymbol{b}_1+m_2\boldsymbol{b}_2+m_3\boldsymbol{b}_3)\cdot\boldsymbol{r}} \end{aligned} \tag{2.5.12}$$

p Arbitrary integer

Although the $\breve{\boldsymbol{u}}_{\boldsymbol{k_{Bp}}}$-constants will be shifted by the integer p, the form of $\boldsymbol{H}$ was the same as if $\boldsymbol{k_B}$ would be chosen to be $\boldsymbol{k_{B0}}$. $\boldsymbol{H}$ would be able to form the same functions in both cases. Since insertion into the eigenvalue equation would yield all solutions in both cases, they would have to be the same solutions. The same principle would apply if an arbitrary number of any primitive reciprocal lattice vectors were be added to the $\boldsymbol{k_{B0}}$-vector. Because of this it is only necessary to consider $\boldsymbol{k_B}$-vector within the range of one primitive reciprocal lattice vector, or more precicely; only within one unit cell of the reciprocal lattice of the structure. This region in the reciprocal space is called the *Brillouin zone.* Choosing a $\boldsymbol{k_B}$-vector outside this zone would only yield another way of expressing the same solutions.

After finding the exact solutions to the eigenvalue equation, the result is a function which maps $\boldsymbol{k_B}$ and the mode number to the field function $\boldsymbol{u}_{\boldsymbol{k_B}}(\boldsymbol{r})$, and a corresponding frequency ω. Figure 2.5.4 is an example from [12] which shows a mapping from $\boldsymbol{k_B}$ to ω. The scaling properties described in section 2.4 are here used to properly scale the ω-axis. The structure which the bands in Figure 2.5.4 applies to are illustrated inside the figure as green dielectric rods. The rods have dielectric constant ($\varepsilon = 8.9$), and the surrounding material has dielectric constant ($\varepsilon = 1$). Since the structure is a two dimensional photonic crystal, $\boldsymbol{k_B}$ has two non-zero components, and the plot in figure 2.5.4 should have illustrated a function of two variables. Instead the figure plots ω along a path in the k_{B1}, k_{B2}-plane. The most common path to follow for two dimensional square photonic crystals is the closed path defined by the three corners $(0, 0)$, $(|\boldsymbol{b}_1|, 0)$, $(|\boldsymbol{b}_1|, |\boldsymbol{b}_2|)$ in the (k_{B1}, k_{B2})-plane. These corners have the respective symbols Γ, X and M, and are shown in a small figure inside Figure 2.5.4. The figure shows ω as a function of $\boldsymbol{k_B}$ along this path for two polarizations. According to [12], TE polarization means that the electric field is perpendicular to the rods in the structure, while TM polarization means that the magnetic field is perpendicular to these rods. Note also that Figure 2.5.4 assumes that the field is constant along the dimension parallel to the rods in the photonic crystal. This is the dimension perpendicular to the small figure with green dots inside Figure 2.5.4, and is called the out of plane dimension.

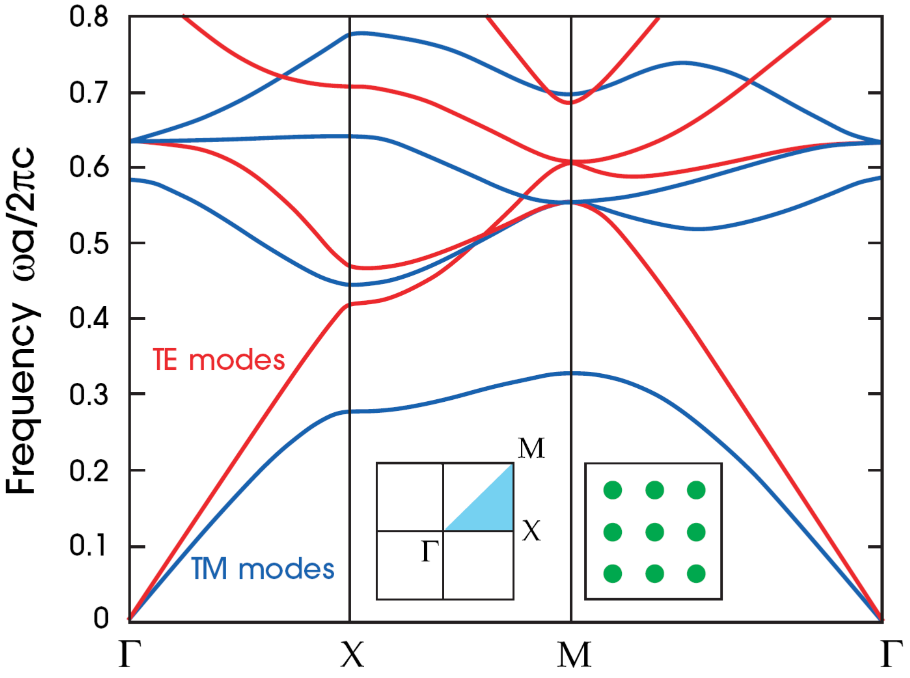


Figure 2.5.4: Band structure of square grating where the the lattice constant is a. [3]

2.5.6 Out of plane propagation

In two dimensional photonic crystals there are modes which are not constant along the out of plane direction. Because the structure is constant along this dimension, the variation must according to [12] be a pure phase shift similar to plane waves in homogeneous media. It is therefore convenient to define a $\boldsymbol{k}$-vector

[3] Figure 2.5.4 is from ref. [12]

which has the non-zero component k_3 in the out of plane direction. The index 3 here denotes the out of plane dimension since 1 and 2 denote the two dimensions which have the photonic crystal periodicity. There is not any Brillouin zone in this dimension, and k_3 can therefore hold any real value. This difference compared to the $\boldsymbol{k_B}$-components is why it is here defined as an independent vector. Using this definition, equation (2.5.10) can instead be written as equation (2.5.13).

$$\boldsymbol{H} = \boldsymbol{u}_{\boldsymbol{k_B},\boldsymbol{k}}(\boldsymbol{r}) \cdot \mathrm{e}^{j(\boldsymbol{k_B}+\boldsymbol{k})\cdot\boldsymbol{r}} \tag{2.5.13}$$

Figure 2.5.5 from [12] illustrates this by plotting ω as a function of both $\boldsymbol{k_B}$ and k_3 for a hexagonal photonic crystal. In this figure k_3 is labeled k_z. Because it is hexagonal the plotted path in the (k_{B1}, k_{B2})-plane is here somewhat different than in Figure 2.5.4. The figure only plots the lowest band, but higher bands should have a similar pattern of raising ω with increased k_3.

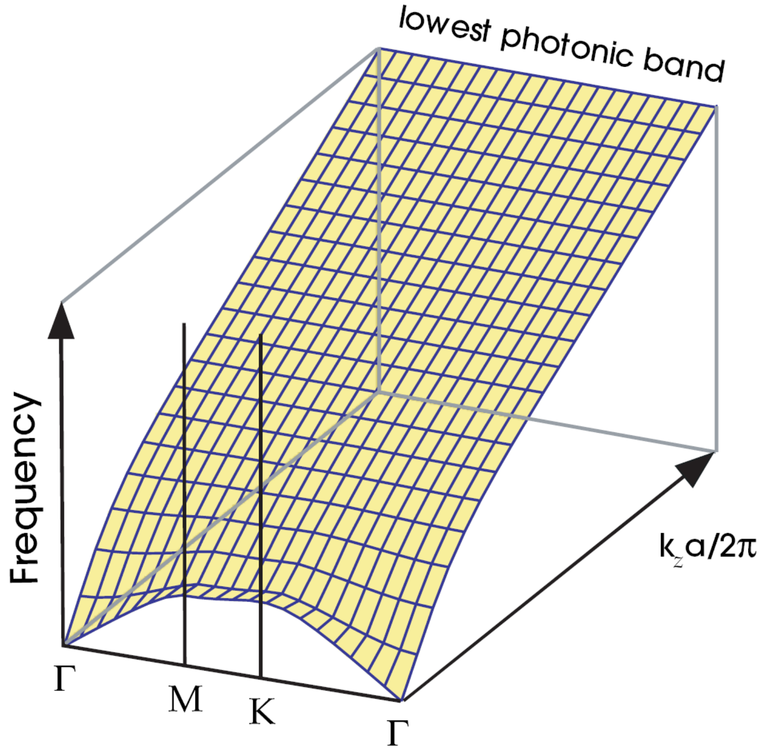


Figure 2.5.5: Lowest band with a non-zero out of plane wave vector for a hexagonal 2D photonic crystal. [4]

[4] Figure 2.5.5 is from ref. [12]

2.6 Mode coupling

Previous sections have discussed the modes of electromagnetic waves in photonic crystals. This section will consider how such modes can be excited by waves in homogeneous media, and vice versa. Because two dimensional photonic crystals are relevant for this thesis, only those will be discussed. More specifically the structure shown in Figure 2.6.1 will be the basis of the coming discussion. Here an arbitrary semi infinite photonic crystal is next to a homogeneous material with dielectric constant ε_h. Although it will not be directly utilized here, this structure actually satisfies the periodic media equation in equation (2.5.1) for two dimensions.

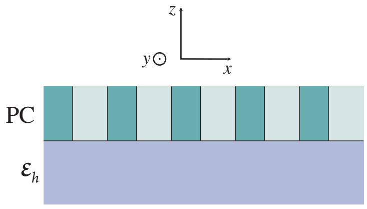


Figure 2.6.1: Infinite layer of 2D photonic crystal bounded by uniform media at the top and bottom.

Bienstman presents in [3] a coupled mode theory which gives quantitative precise measures of how this coupling occurs. This is however very complex, and not really necessary for the work which will be presented here. Instead this section will be used to find the modes which *can* couple to each other, while not finding quantitative measures of how good this coupling is. The basis of the coupled mode theory are the boundary conditions imposed by the Maxwell equations. These require some sort of continuity across the interface between the photonic crystal and the homogeneous media. In particular these all require that the scalar fields of the modes are not orthogonal on the interface. In mathematical terms this means that $\langle H_{PC}, H_h \rangle$ in equation (2.6.1) must be non-zero. Here * denotes complex conjugation. The equation is also a very common definition of an inner product of two scalar functions on the inferface plane.

$$\langle H_{PC}, H_h \rangle = \int_{Interface} H_{PC}^*(\boldsymbol{r}) \cdot H_h(\boldsymbol{r}) \mathrm{d}A \tag{2.6.1}$$

H_{PC} Scalar magnetic mode field in the photonic crystal
H_h Scalar magnetic mode field in the homogeneous media

Using a two dimensional version of equation (2.5.11) for H_{PC} and the plane wave solution in equation (2.3.4) for H_h, equation (2.6.1) transforms into equation

(2.6.2). Note that the vector scaling constant $\boldsymbol{H_0}$ in equation (2.3.4) has been omitted as it does not affect the form of the mode. The coordinate system indicated in Figure 2.6.1 has been used. The integral is performed on the interface between the two media which is a plane parallel to the xy-plane. Note that since the photonic crystal is two dimensional in the xy-plane, the primitive reciprocal lattice vectors $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ must also be in this plane.

$$\begin{aligned}\langle H_{PC}, H_h\rangle &= \int\limits_{x=-\infty}^{\infty}\int\limits_{y=-\infty}^{\infty} \mathrm{e}^{-j\boldsymbol{k_B}\cdot\boldsymbol{r}} \sum_{m_1}\sum_{m_2} \breve{u}_{\boldsymbol{k_B}}(m_1, m_2)\, \mathrm{e}^{-j(m_1\boldsymbol{b}_1+m_2\boldsymbol{b}_2)\cdot\boldsymbol{r}} \cdot \mathrm{e}^{j\boldsymbol{k}\cdot\boldsymbol{r}}\mathrm{d}A \\ &= \sum_{m_1}\sum_{m_2} \breve{u}_{\boldsymbol{k_B}}(m_1, m_2)\, \mathrm{e}^{k_z z} \int\limits_{x=-\infty}^{\infty}\int\limits_{y=-\infty}^{\infty} \mathrm{e}^{j(\boldsymbol{k_{xy}}-\boldsymbol{k_B}-m_1\boldsymbol{b}_1-m_2\boldsymbol{b}_2)\cdot\boldsymbol{r}}\mathrm{d}A \qquad (2.6.2)\end{aligned}$$

$\boldsymbol{r}$	Position in space ($= x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}}$) where $\hat{\boldsymbol{x}}$ is the unit vector in the x-direction
k_z	Component of $\boldsymbol{k}$ which is in the z-direction
$\boldsymbol{k_{xy}}$	Component of $\boldsymbol{k}$ which is in the xy-plane.

For the result of equation (2.6.2) to be non-zero, the integral must be non-zero for at least one term in the double sum. From Fourier analysis it is known that such an integral can only be non-zero if the vector sum in the exponential function is zero. Otherwise the integrand will oscillate over the integration plane resulting in an average value of zero. Another way of putting this requirement is that expressed in equation (2.6.3). *The difference between the $\boldsymbol{k}$-vector projection on the interface plane and the $\boldsymbol{k_B}$-vector of the photonic crystal mode must be a reciprocal lattice vector.* That is a vector composed of an integer amount of each primitive reciprocal lattice vector.

$$\boldsymbol{k_{xy}} - \boldsymbol{k_B} = m_1\boldsymbol{b}_1 + m_2\boldsymbol{b}_2 \qquad (2.6.3)$$

The actual shape of the $\boldsymbol{u}$-function can cause some of its Fourier coefficients $\breve{u}_{\boldsymbol{k_B}}(m_1, m_2)$ to be zero and therefore impose additional limits to the coupling. Also the polarization of the modes can cause additional constraints on the coupling. However to find these the exact mode fields of the photonic crystal must be found, and an exact coupled mode theory as that described by Bienstman [3] must be used.

Chapter 3

LED

The purpose of the photonic crystal described in this thesis is to increase the external quantum efficiency of a LED. The fabrication of the LED structure had been done by Tron Arne Nilsen before the work described in this thesis started. The LED and its principle of operation, will therefore only be shortly presented here. The optical properties which in turn will affect the performance of the photonic crystal will however be analyzed in more detail.

3.1 General LED theory

A pn-junction forming a diode is illustrated in Figure 3.1.1. Here two separate parts of a single semiconductor crystal have different doping types [23]. Because of the high differences in hole and electron concentrations in the two parts, positive holes will diffuse to the n-side and negative electrons will diffuse to the p-side. This transfer of charge causes a buildup of space charge in a region very close to the interface between the two doping types. The charge causes again an electrical field which counteracts the diffusion current. In thermal equilibrium the net current is therefore zero, and the band diagram [23] looks like Figure 3.1.2. The region with space charge is called the depletion zone because it does not contain any free charge carriers. In Figure 3.1.2 the depletion zone can be identified as the region where the band diagrams have a slope.

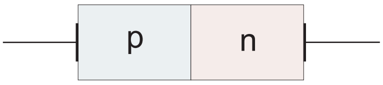


Figure 3.1.1: Regular pn-junction

If an external voltage source applies a positive voltage to the p-side, the counteracting electric field will decrease, causing a net diffusion current. Charge carriers

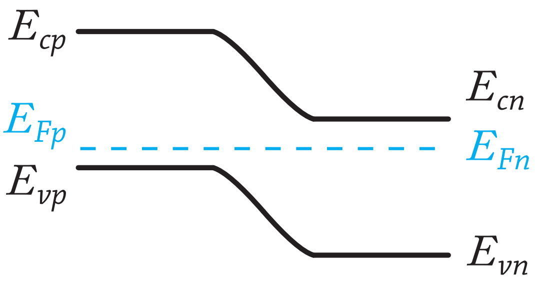


Figure 3.1.2: Band edges for valence and conduction band on p and n-side. Dotted line is Fermi level

can then move across the junction and later recombine [23] on the other side. This recombination releases energy as the charge carriers shift to a lower state. If the bandgap in the semiconductor is a a direct gap, meaning the electrons can change energy state without changing their momentum [23], the released energy can be in the form of electromagnetic radiation. The released energy is equal to the bandgap of the semiconductor and decides the wavelength of the emitted radiation through Eq. (3.1.1)

$$\lambda = \frac{ch}{E_g} \tag{3.1.1}$$

λ	wavelength in vacuum
c	Speed of light
h	Planck's constant
E_g	Bandgap energy $E_c - E_v$ in Figure 3.1.2

3.2 LED structure

The actual LED which is the topic of this thesis consists of several layers grown on a gallium antimonide (GaSb) substrate, and is illustrated in Figure 3.2.1. The substrate itself is n-doped and the first grown layer is also n-doped GaSb. The purpose of this layer is to set the doping levels to a desired value and to get a better crystal quality in the active region of the LED. Next is the transition between the n and p-doped side of the crystal. This boundary is the pn junction, and it is here the charge carriers will move across and into the side of opposite doping when an external voltage is applied. The lowest and first p-doped layer is composed of gallium indium antimonide ($\mathrm{Ga_{0.925}In_{0.075}Sb}$). Here the indices indicate that 7.5% of the gallium in GaSb, measured in mole, is replaced by indium. Above there is a p-doped layer of GaSb where the doping concentration is gradually decreased as illustrated in Figure 3.2.1. The indium in the lowest p-doped layer causes a slight narrowing of the bandgap, and the injected electrons will therefore need to pass an energy barrier to reach the higher layers with pure GaSb. This causes a slight confinement of minority carriers [23], but a significant amount are still able to diffuse into the higher layer. The different doping concentrations in the graded layer also have a similar effect. According to [22], there is an increased narrowing of the bandgap with increased doping, causing each sub layer to have a small confinement of electrons. Another apparent effect of non uniform bandgap is that the light emitted from the different regions will have a correspondingly different wavelength. The effects of this will however not be considered here since the width of each layers' emission spectrum is believed broader than the wavelength spread due to different doping. The top layer is made of intrinsic GaSb forming an insulator. As illustrated in Figure 3.2.1, a hole must be etched in this intrisic layer for the p-side metal contact. It is in this layer the photonic crystal is supposed to be etched out. Since it typically will be made by etching out holes in the material, it is beneficial to keep it away from the emitting regions. The alternative would lead to a net decrease of emitting material, and a possibility of undesired recombination caused by surface effects.

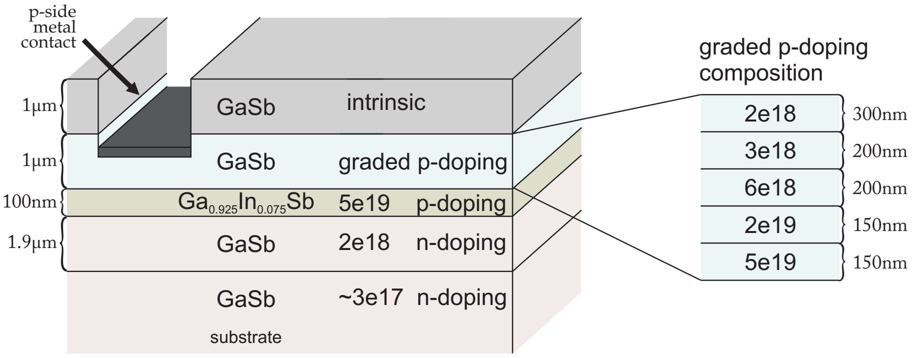


Figure 3.2.1: Different crystal layers and corresponding doping. The doping concentrations are measured in cm^{-3}

The combined emission spectrum from all layers is shown in Figure 3.2.2. The main peak at 0.7eV is most probably generated in the pure GaSb layers, and the smaller shoulder around 0.65eV is generated in the In-containing layer.

Figure 3.2.3 shows the structure of the LED after all processing steps are done. On the top one can see the intrinsic top layer shown in Figure 3.2.1. This is where the light is able to couple out of the device . The p-side contact surrounds this window and has an extended area for wire bonding. The n-side contact is not visible in Figure 3.2.3 as it is on the bottom side of the substrate. Outside the contact the layers shown in Figure 3.2.1 are etched away such that charge carriers in the p-layers are confined to the LED. This etched surface has also been passivated by a layer of insulating silicon nitride which has been deposited. This prevents currents from flowing along this surface effectively shorting the LED. Figure 3.2.3 also indicates a coordinate system. This defines the directions used

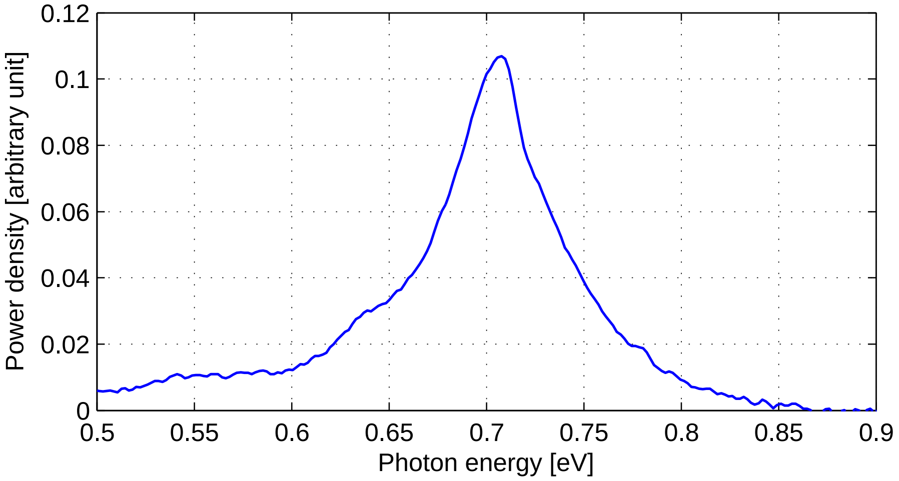


Figure 3.2.2: Meassured emission spectrum from the LED. The measurement is done by Tron Arne Nilsen using an FTIR spectrometer.

throughout this thesis.

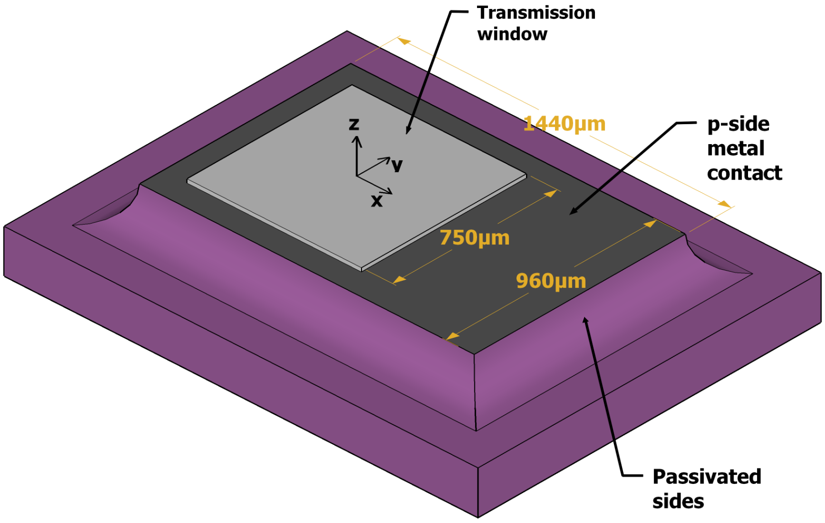


Figure 3.2.3: Illustration of the LED structure

3.3 Internal radiation distribution

3.3.1 General

The first step towards determining the radiation exiting the device is to determine the properties of the emitted radiation inside the semiconductor. More precisely, it is necessary with an angular power density function valid just below the interface between the topmost p-doped layer and the intrinsic layer in Figure 3.2.1. Such a function is expected to vary across the emission spectrum of the LED shown in Figure 3.2.2. For simplicity the coming analysis will only consider two points in this spectrum. These are the main peak at 0.7eV and the small shoulder at 0.65eV. The following analysis will primarily rely on two assumptions which are:

- Each volume element in the emitting region of the semiconductor will emit isotropically.

- Light moving downwards in the substrate will be absorbed before returning to the top side after reflecting on the bottom side. The light emitted in the downwards direction will therefore not contribute to the power density function which is to be constructed. Also light that is reflected downwards by the top surface will be considered lost.

3.3.2 Distribution function

Radiometric units such as radiance or radiant intensity would usually be the preferred quantities for these analyses. Due to the chosen simulation method that will be described in a later chapter, it will in this case be better with a power density function in the two transversal wave vector components k_x and k_y. Using the notation of [4] and the principles of fourier optics [20] these variables are related to the spherical angles θ and ϕ in the far field radiation pattern. The far field radiation pattern is a function of θ and ϕ which measures the power density per unit solid angle at a distance that is significantly larger than the width of the LED. In general the distance should be even bigger, but since the LED is not expected to have any significant spatial coherence, the requirement can be relaxed. It is definetly worth mentioning that the radiation pattern discussed here is not what one would expect to find by measuring the emitted light from the physical LED at different angles. Since the goal is to find the radiation distribution inside the LED, the effect of the semiconductor to air interface must be omitted. This means that the structure to evaluate must be the actual LED surrounded by a material of the same refractive index as the LED itself. The transversal wave vector components of light traveling at an angle described by θ and ϕ are stated in equations (3.3.1) and (3.3.2). Using these equations it is possible to relate the far field radiation pattern to a power density function in k_x and k_y.

$$k_x = k_T \cos(\phi) = k \sin(\theta) \cos(\phi) \tag{3.3.1}$$
$$k_y = k_T \sin(\phi) = k \sin(\theta) \sin(\phi) \tag{3.3.2}$$

k_x, k_y	Wave vector components
k_T	Transversal component of wave vector $\sqrt{k_x^2 + k_y^2}$
k	Medium dependent wave number $(n \cdot k_0)$
θ	Angle from z-axis
ϕ	Angle from x-axis in the xy-plane

When changing the variables of a power density function the power obtained by integrating over parts of the domain, must be preserved. For example the total power emitted is the integral of the radiation pattern over the upper hemisphere, and is stated in equation (3.3.3)

$$P = \int_{\substack{Upper\\hemisphere}} I(\theta, \phi) \mathrm{d}\Omega = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} I(\theta, \phi) \sin\theta \mathrm{d}\theta \mathrm{d}\phi \tag{3.3.3}$$

P	Total power
$\mathrm{d}\Omega$	Differential solid angle
$I(\theta, \phi)$	Power density per unit solid angle in the far field

An identical value for P should be obtained by integrating the power density function in k_x and k_y, I_k, over all possible k_x and k_y. The possible k_x and k_y are limited by the fact that the total transversal k vector component, $\sqrt{k_x^2 + k_y^2}$, must be less than k. The integration area will therefore be a surface in the $k_x\, k_y$-plane limited by a circle with center in the origin and radius k. This is stated in equation (3.3.4).

$$P = \int_{k_x^2 + k_y^2 < k^2} I_k(k_x, k_y) \mathrm{d}k_x \mathrm{d}k_y \tag{3.3.4}$$

To accommodate the mentioned requirement, the power density function must according to [5] be scaled with the Jacobian determinant. Equation (3.3.5) states this requirement when changing from a density in k_x and k_y to one in θ and ϕ.

$$\mathrm{d}P = I_k(k_x, k_y)\, \mathrm{d}k_x \mathrm{d}k_y = I_k\left(k_x(\theta, \phi),\ k_y(\theta, \phi)\right)\, |J_T|\, \mathrm{d}\theta \mathrm{d}\phi \tag{3.3.5}$$

I_k	Power density function in k_x, k_y
$\lvert J_T \rvert$	Jacobian determinant

The Jacobian determinant is given by equation (3.3.6)

$$|J_T| = \left|\frac{\partial(k_x, k_y)}{\partial(\theta, \phi)}\right| = \begin{vmatrix} \frac{\partial k_x}{\partial \theta} & \frac{\partial k_x}{\partial \phi} \\ \frac{\partial k_y}{\partial \theta} & \frac{\partial k_y}{\partial \phi} \end{vmatrix} = k^2 \cos(\theta) \sin(\theta) \tag{3.3.6}$$

I_k	Power density function in k_x, k_y
$\lvert J_T \rvert$	Jacobian determinant

Inserting equation (3.3.6) into (3.3.5) and recognizing the factor $\sin(\theta)\mathrm{d}\theta\mathrm{d}\phi$ as $\mathrm{d}\Omega$ transforms equation (3.3.5) into equation (3.3.7)

$$\mathrm{d}P = \underbrace{k^2\, I_k \left(k_x(\theta, \phi),\ k_y(\theta, \phi)\right)\, \cos(\theta)}_{I(\theta,\phi)}\ \mathrm{d}\Omega \tag{3.3.7}$$

Recognizing the power density per unit solid angle in the far field from equation (3.3.7) and expressing θ and ϕ as functions of k_x and k_y, gives the relation between the two power density functions expressed in equation (3.3.8)

$$\begin{aligned} I_k\left(k_x,\ k_y\right) &= \frac{1}{k^2 \cos\left(\theta(k_x, k_y)\right)}\, I\left(\theta(k_x, k_y),\ \phi(k_x, k_y)\right) \\ &= \frac{1}{k^2 \sqrt{1 - \frac{k_x^2 + k_y^2}{k^2}}}\, I\left(\theta(k_x, k_y),\ \phi(k_x, k_y)\right) \end{aligned} \tag{3.3.8}$$

Because of equation (3.3.8), it is possible to construct a power density function for the far field, and subsequently transform it to apply to the transversal wave vector components. The following sections will describe the formation of such a radiation pattern. The effect shaping the function and the needed material parameters will first be described before a proper radiation pattern is presented.

3.3.3 Absorption

The material interacts with the light mainly through spontaneous emission and absorption. The most simple model to analyze is a thick layer of light emitting material where each volume element emits isotropically. In this context a thick layer means that the thickness is much larger than the penetration depth of the light. Penetration depth is here defined as the length in which the irradiance of a plane wave has fallen to e^{-1}. This is the inverse of the absorption coefficient α used in [20]. This finite absorption will influence the angular distribution of the radiation because oblique rays will have a longer path length in the material than perpendicular rays. The light originating from a depth δ in the emitting layer, and propagating at an angle θ to the face normal will have an optical path length in the layer equal to $\delta/\cos\theta$. This corresponds to an attenuation of $\mathrm{e}^{-\alpha\delta/\cos\theta}$. Figure 3.3.1 illustrates such a beam path. The radiation pattern can then be constructed by integrating the contribution to optical power of all volume elements in the emitting layer. The assumption that the penetration depth is significantly less than the layer thickness allows integration of δ from zero to infinity. The result is, as shown in equation (3.3.9), the familiar cosine distribution.

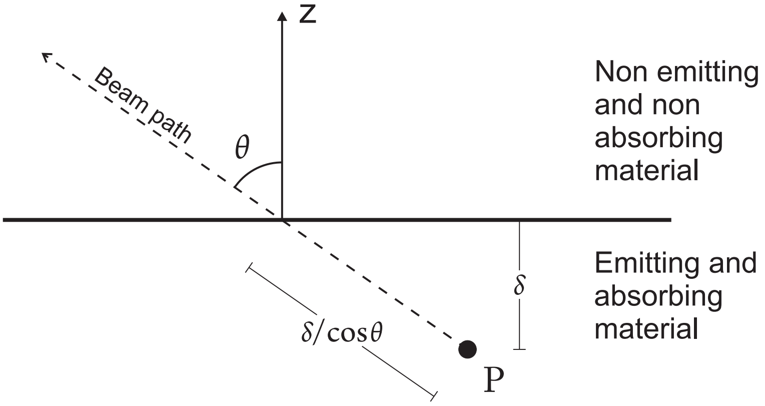

Z
Beam path
θ
Non emitting and non absorbing material
Emitting and absorbing material
δ
δ/cosθ
P

Figure 3.3.1: Beam path of light emitted at a point P, and traveling at an angle θ.

$$I(\theta,\phi) = \int_{LED} U\mathrm{e}^{-\alpha\delta/\cos\theta}\mathrm{d}V = UA_{LED}\int_{\delta=0}^{\infty}\mathrm{e}^{-\alpha\delta/\cos\theta}\mathrm{d}\delta$$

$$= \frac{UA_{LED}}{\alpha}\cos\theta \qquad (3.3.9)$$

U	Generated optical power per unit volume
α	Optical absorption coefficient. (Inverse of penetration depth)
A_{LED}	Area of the LED surface

Several of the assumptions that equation (3.3.9) are based on, are indeed not necessarily fulfilled by the physical structure. The most apparent being that the penetration depth probably is not much smaller than the overall thickness of the emitting layer. According to [21], the penetration depth of most direct gap semiconductors is around $1\mu m$ for wavelengths with a photon energy that is higher than the bandgap. This value must at best be considered an indication of what order of magnitude to expect. Measured or calculated values for the material compositions and doping levels used are not available in the literature. In addition the emitted light has a photon energy very close to the bandgap of the material, meaning the absorption coefficients of the different layers will vary significantly over the emission spectrum of the LED. To make matters even more complicated the different layers contribute to different parts of the emission spectrum, meaning that both the beam paths and absorption coefficients are dependent on wavelength. A final complication is that the layers also generate a different amount of optical power per unit volume. This is both because of varying minority carrier concentrations and because the layers have different radiative recombination rates [21]. The following sections will address these issues with regard to the structure in mind. However, since exact parameters are not known, the resulting conclusion should be considered preliminary.

Effect of finite thickness

The effect of final thickness of the emitting layer is easily determined by narrowing the limits of the integral in equation (3.3.9). Equation (3.3.10) shows this with a layer thickness t.

$$\begin{aligned} I(\theta, \phi) &= UA_{LED} \int_{\delta=0}^{t} \mathrm{e}^{-\alpha\delta/\cos\theta} \mathrm{d}\delta \\ &= \frac{UA_{LED}}{\alpha} \left(1 - \mathrm{e}^{-\alpha t/cos\theta}\right) \cos\theta \qquad (3.3.10) \end{aligned}$$

t Thickness of light emitting layer

Figure 3.3.2 plots equation (3.3.10) for different values of αt. Note that it is only the angular variations that are of interest in this context. The plots in Figure 3.3.2 are therefore normalized to the maximum value, excluding the effect of the $1/\alpha$-factor in equation (3.3.10).

The effect of a thinner emitting layer looks apparent from the plots in figure 3.3.2. Although the power density still goes to zero as θ approaches $\pi/2$, the curves are more constant in the area around $\theta = 0$. A physical reason for this effect is that absorption plays a less significant role as long as the effective beam path is less then the penetration depth. As a result the power density does not start decaying until θ reaches a value where the beam path length is equal to the penetration depth.

Effect of absorbing cap layer

As mentioned the different layers in figure 3.2.1 emit at slightly different wavelengths. Because of this, it is not correct to consider all layers as a block of

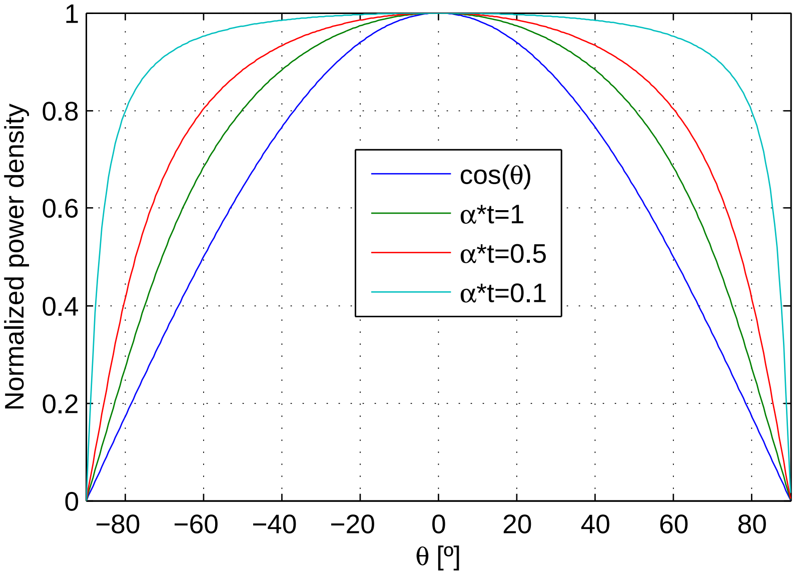


Figure 3.3.2: Normalized plots of equation (3.3.10) showing the angular distribution for a finitely thick emitting layer.

emitting material. When treating each wavelength independently, there will typically be one or some layers contributing to emission as characterized by equation (3.3.10). There will however be layers above not contributing much to emission, but still affecting the radiation due to their absorption. Therefore it is necessary to take into account the effect of an absorbing layer on top of the emitting layer. This can be done very easily by adding a θ dependent absorption factor to equation (3.3.10). Equation (3.3.11) states the results, and Figure 3.3.3 shows three different plots illustrating the effect.

$$I(\theta, \phi) = \frac{UA_{LED}}{\alpha} \left(1 - \mathrm{e}^{-\alpha t / \cos\theta}\right) \mathrm{e}^{-\alpha_c t_c / \cos\theta} \cos\theta \tag{3.3.11}$$

α_c	Absorption coefficient in cap layer
t_c	Thickness of cap layer

The plots in Figure 3.3.3 show the angular distribution when an absorbing cap layer is added on top of the emitting layer. The product $\alpha_c t_c$ is for each plot set to the same as αt, for example illustrating a cap layer with the same thickness and absorption as the emitting layer, or twice the thickness and half the absorption. The main difference from Figure 3.3.2 is a less flat curve because the net absorption is larger and therefore shapes the curve more. Another distinct difference is the rapid exponential decay when θ approaches 90° which comes from the exponential nature of this absorption.

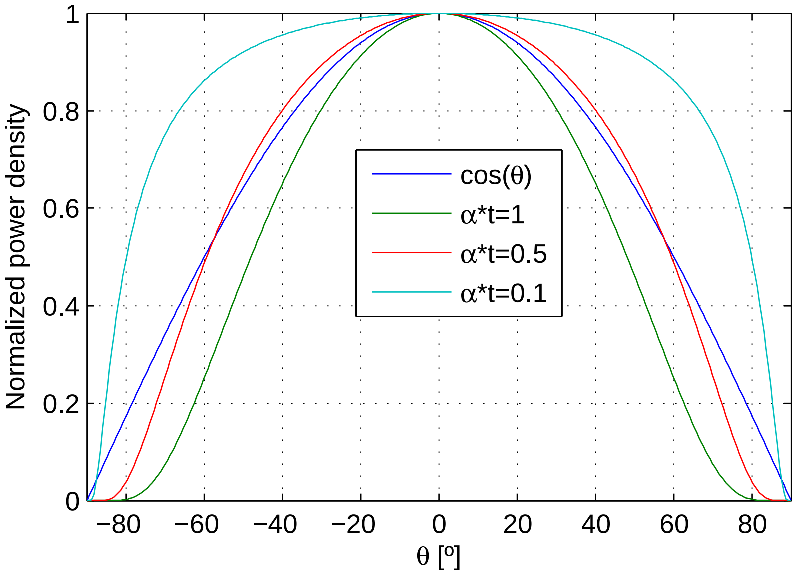


Figure 3.3.3: Normalized plots of equation (3.3.11) showing the angular distribution when an absorbing cap layer is added to the emitting layer. $\alpha_c t_c$ is for each plot, set to the same value as αt

3.3.4 Absorption coefficient

p-doped layers

From section 3.3.3 it is evident that the absorption coefficient is critical for the angular power density. It is therefore important to get a good estimate of it. Measurements of the absorption coefficient made by Iluridze et. al. [10] in p-doped GaSb cooled to 77K is shown for four different doping concentrations in Figure 3.3.4.

The doping levels in Figure 3.3.4 can, with some interpolation give a reasonable estimate of the absorption coefficients in the graded p-doped region illustrated in Figure 3.2.1. Because the use of the LED most probably will be in tempered environments, it is of much greater interest to find an absorption coefficient valid for room temperature. Such measurements with both the desired doping levels and the desired temperature have not been found, but Becker et. al. [2] have performed measurements of lightly p-doped GaSb in both 77K and 300K environments. These measurements are illustrated in Figure 3.3.5.

It is clear from Figure 3.3.5 that the increased temperature causes shift in the absorption spectrum toward lower photon energies. Applying a similar shift for the absorption spectra in figure 3.3.4 might give an indication to the real absorption coefficients to expect for room temperature. This must however only be considered as a vague estimate since the effect of temperature might be signif-

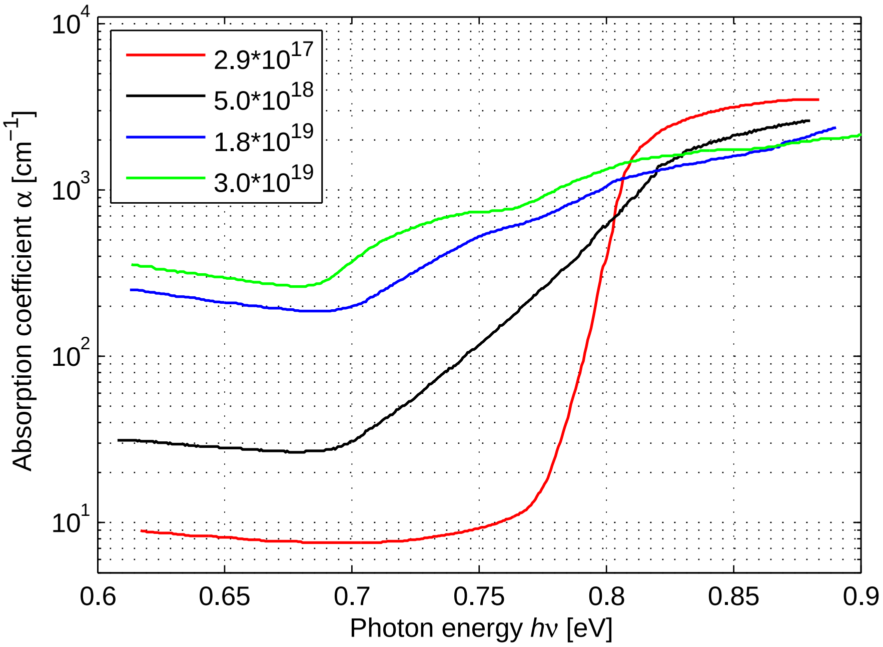


Figure 3.3.4: Absorption coefficient measured by Iluridze et. al. [10] for four different p type doping concentrations. The measurements are done with a temperature of 77K.

icantly different for the different doping concentrations. Figure 3.3.6 illustrates these shifted absorption spectra and an overlay of the emission spectrum of the LED. Beware that the emission spectrum is plotted as if the y axis was a linear scale.

Another approach would be to consider band gap narrowing as a function for doping concentrations, and to shift the curve for 300K in Figure 3.3.5 accordingly. Stollwerck *et al* have in [22] cited values calculated by Jain *et al* [11] for the bandgap narrowing. According to Stollwerck *et al* these values correspond well to measurements done by Titkov *et al* [24] and Hjelt *et al* [9]. The calculated function is illustrated in Figure 3.3.7.

Shifting the absorption curve for 300K in Figure 3.3.5 for the same doping concentrations as in Figure 3.3.4 results in the absorption spectra illustrated in figure 3.3.8.

The differences between the absorption spectra in Figure 3.3.8 and Figure 3.3.6 are very significant, and indicate that finding the real absorption coefficient is a much more complicated process than any of the two approaches discussed here. As indicated by Figure 3.3.3, the absorption coefficient does noticeably affect the radiation pattern. Finding it exactly will however require transmission measurements on epitaxially grown layers with the desired doping concentrations. Since this would be much too time consuming and costly, and really not within the

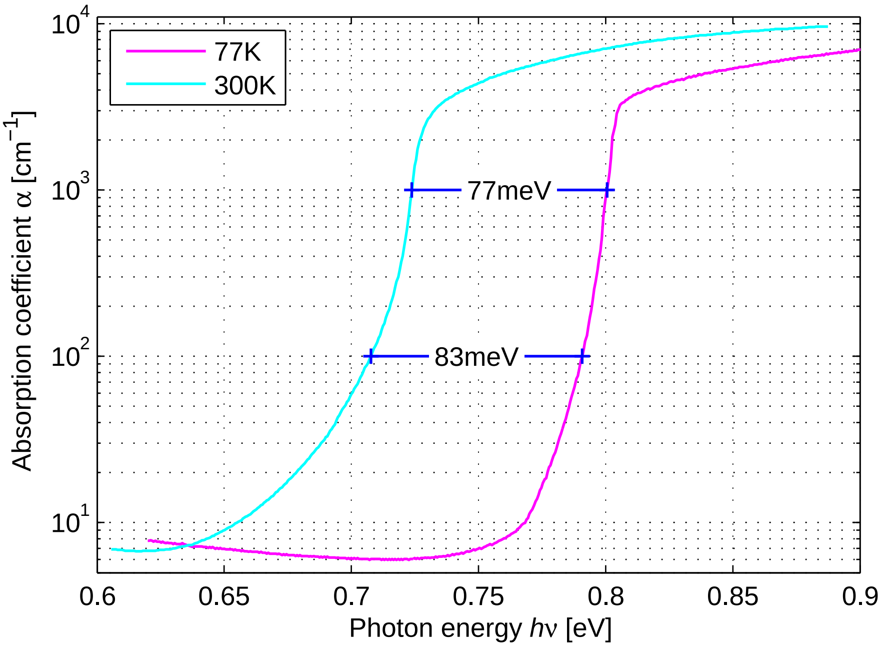


Figure 3.3.5: Absorption coefficient measured by Becker *et al* [2] for low doping concentration $(1.2 \cdot 10^{17})$. The measurements are done with a temperature of 77K and 300K.

scope of this thesis, such work has not been done here. Instead the analysis must rely on the assumed absorption spectra in Figure 3.3.6 and 3.3.8. It is believed that the shifted measurements by Iluridze *et al* in Figure 3.3.6 are somewhat more accurate. This is because it only assumes that the deformation of the absorption curve due to doping is equal for 77K and 300K. The alternative assumes no deformation for 300K despite measured deformation for 77K. Since the first seems to be more reasonable, it will be the main assumption for estimating the absorption coefficient. This assumption can be divided into two more precise assumptions. The first being that the shift in photon energy caused by increased temperature is equal despite different doping concentrations. The second being that the increased temperature does not deform the absorption curve, the only effect being the assumed shift. Ghezzi *et al* [8] have done measurements on n-doped GaSb which show an increased temperature dependence with increasing doping concentration. These measurements are shown in Figure 3.3.9, but only cover the part of the absorption spectrum where the absorption coefficient is larger than 1000cm^{-1}. In addition n-doping does not necessarily cause the same effects as p-doping. In absence of any definite absorption coefficient, the spectrum in Figure 3.3.6 will here be the basis of the coming discussions. It must however be clear that this can introduce a notable source of error.

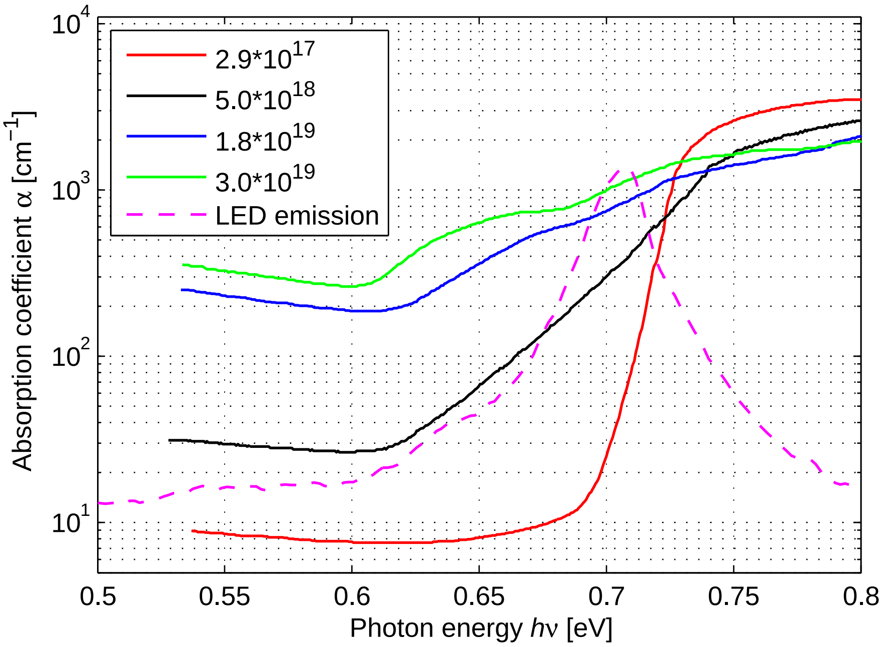


Figure 3.3.6: The absorption spectra from Figure 3.3.4 shifted 80meV. The LED emission spectrum is plotted as if the scale was linear to make it resemble Figure 3.2.2

The green curve in Figure 3.3.6 applies for a doping concentration of $3.0 \cdot 10^{19}\text{cm}^{-3}$, and the blue for a doping concentration of $1.8 \cdot 10^{19}\text{cm}^{-3}$. Since the two lowest layers of p-doped GaSb in Figure 3.2.1 have a similar doping concentration, an absorption coefficient of 1000cm^{-1} will be assumed for these layers for the light generated within them. The three layers above have lower doping concentrations and correspond best to the black curve in figure 3.3.6. These layers will work as a cap layer for the light emitted below, and has an assumed absorption coefficient of 300cm^{-1} for the light emitted from the lowest p-doped layers. Estimating the absorption spectrum for the p-doped layers containing indium is also difficult. It is however not unreasonable to believe that the absorption coefficient for the light emitted here is the same as the absorption coefficient in the highly p-doped layers for the light emitted there. As it is believed that the indium containing layer emits at a photon energy of approximately 0.65eV, the two lowest p-doped GaSb layers will work as a cap layer with an absorption coefficient of 400cm^{-1} according to Figure 3.3.6. The absorption in the higher levels is assumed too small to make a significant difference for this light. To improve the overview over these absorption coefficients, Table 3.3.1 lists them for the different layers.

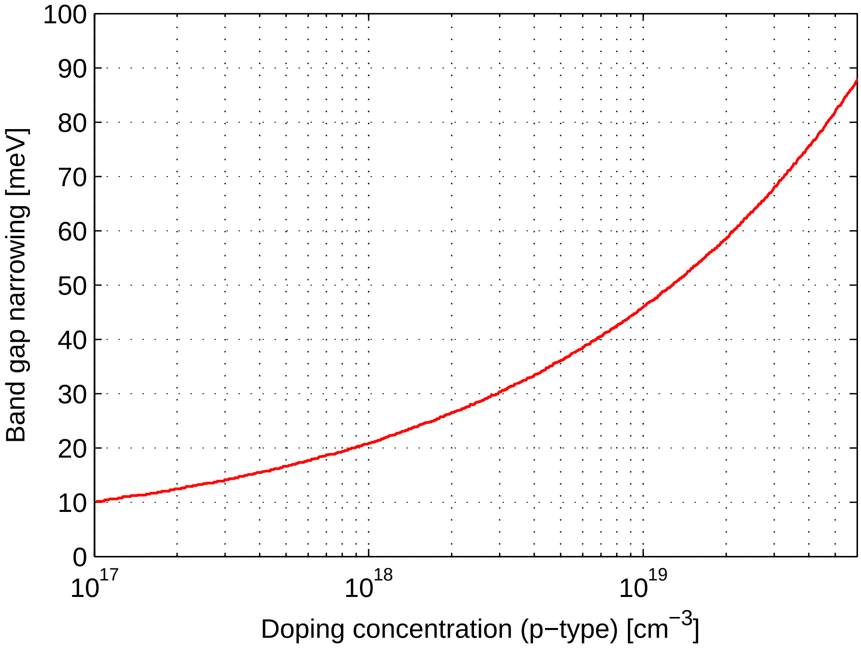


Figure 3.3.7: Band gap narrowing as a function of doping concentration. The curve is calculated by Jain *et al* [11]

n-doped layers

The n-doped layers below the pn junction are somewhat more difficult. Mainly because it is difficult to know the wavelength of the light emitted from the two layers. It will in the following, be assumed that the emission from these layers also is centered around the peak in the emission spectrum shown in Figure 3.2.2. As mentioned Ghezzi *et al* [8] have done measurements here shown in Figure 3.3.9, of the absorption coefficient in n-doped GaSb. These are however only presented for the photon energy spectrum between 0.73eV and 0.86eV which only touches the emission spectrum of the LED. Using extrapolation, a value of approximately 500cm^{-1} is estimated for the grown layer, and a value of approximately 200cm^{-1} is estimated for the substrate. This has been done while expecting a turning point in the graph around a value of 1000cm^{-1} This is similar to the measurements on p-doped GaSb shown in Figure 3.3.5. The absorption coefficients discussed here are also listed in Table 3.3.1 in the rows for the n-doped layers.

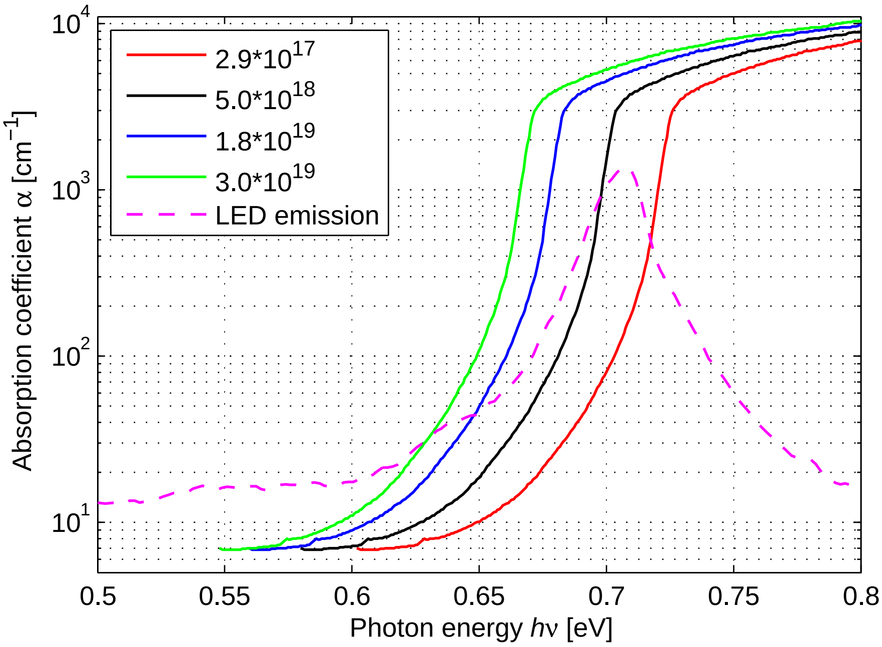


Figure 3.3.8: The absorption spectrum from Figure 3.3.5 shifted in photon energy by values for band gap narrowing in figure 3.3.7. The different plots are for different p-type doping concentrations. The LED emission spectrum is plotted as if the scale was linear to make it resemble Figure 3.2.2

Table 3.3.1: Absorption coefficients for different layers

Layer doping cm^{-3}	α @ 0.65eV [cm^{-1}]	α @ 0.7eV [cm^{-1}]	Thickness [μm]
GaSb p = $2 \cdot 10^{18}$	0	300	0.3
GaSb p = $3 \cdot 10^{18}$	0	300	0.2
GaSb p = $6 \cdot 10^{18}$	0	300	0.2
GaSb p = $2 \cdot 10^{19}$	400	1000	0.15
GaSb p = $5 \cdot 10^{19}$	400	1000	0.15
GaInSb p = $5 \cdot 10^{18}$	1000	1000	0.1
GaSb n = $2 \cdot 10^{18}$	-	500	1.9
GaSb n = $3 \cdot 10^{17}$	-	200	500

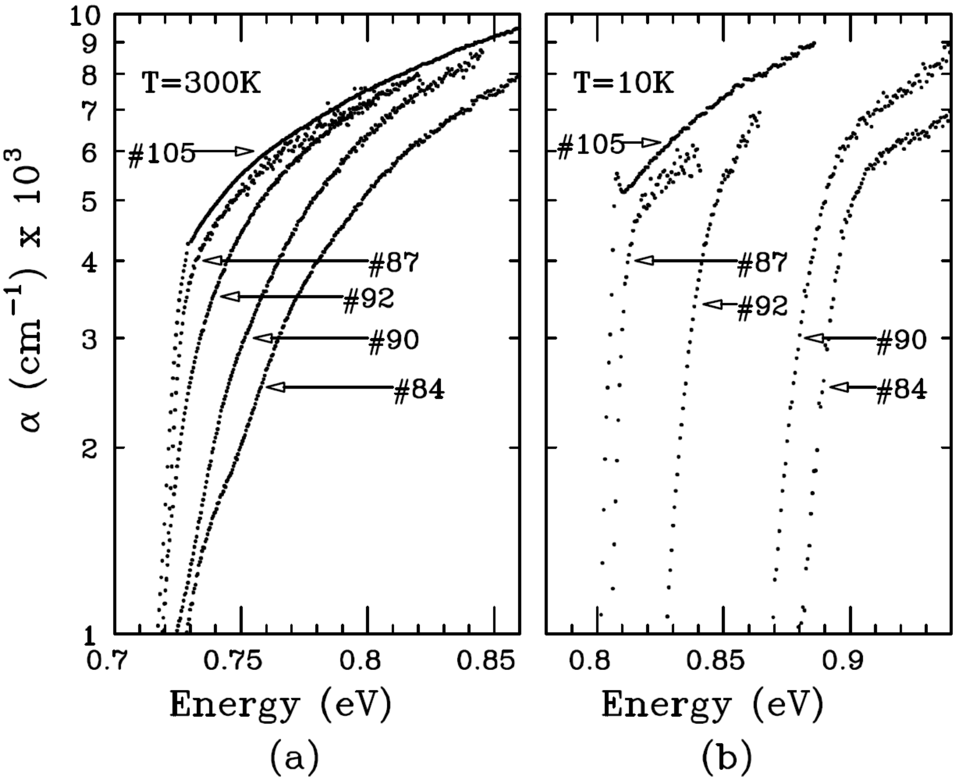


Figure 3.3.9: Absorption coefficient of n-doped GaSb measured by Ghezzi *et al* [8].The doping concentrations measured in cm^{-3} are as follows: #105: $p = 2.9 \cdot 10^{16}$, #87: $n = 8.0 \cdot 10^{15}$, #92: $n = 1.4 \cdot 10^{17}$, #90: $n = 6.4 \cdot 10^{17}$, #84: $n = 7.6 \cdot 10^{17}$

3.3.5 Emission distribution

In section 3.3.4 the absorption coefficients for the different layers have been stated for two points in the emission spectra. For the light with a photon energy of 0.65eV this is sufficient for calculating a radiation pattern based on the absorption in the emitting material and the above layers. Because there are several layers emitting light with a photon energy of 0.7eV, it is a somewhat more difficult task to calculate its radiation pattern. The contribution to the total radiation pattern from each of these layers, must be scaled properly with a measure of their emission rate. Mathematically this must be done by finding relative values for the U parameter in equations (3.3.9), (3.3.10), and (3.3.11). This parameter expresses light generation per unit volume of the emitting material. A quantity that is proportional to U is the spontaneous emission rate R_{spon} given by equation (3.3.12) [21] for p-doped layers. The equivalent expression for n-doped layers can be found by replacing all n symbols by p and vice versa.

$$R_{spon} = \frac{\Delta n}{\tau_r} \approx \frac{n}{\tau_r} \tag{3.3.12}$$

R_{spon}	Photon emission rate per volume
τ_r	Minority carrier radiative lifetime
Δn_p	Excess electron density in p-doped material
n_p	Electron density in p-doped material

As stated by equation (3.3.12), this requires values for radiative lifetime for the different layers, and a minority carrier distribution (n_p and p_n) for the doped layers. Take note that the indices of the minority carrier symbols indicate on which side of the junction they apply for (n or p), and the symbol itself indicates whether it is an electron or hole concentration.

Minority carrier injection

The first step towards finding the minority carrier concentrations for the different layers in Figure 3.2.1 is to find relative numbers for the amount of holes injected into the n-doped region, and of electrons injected into the p-doped region. The minority carriers are injected through the pn junction from the side of opposite doping. According to [23], the minority carrier concentration at the edge of the depletion zone for a normal diode is determined by the potential difference between the p and n-side and the majority carrier concentration on the opposite side. Equations (3.3.13) and (3.3.14) state this when the doping concentrations are much greater than the intrinsic carrier concentrations, and when all doping atoms are ionized to form majority carriers.

$$n_p(0) = N_d \mathrm{e}^{\frac{-q(V_0 - V)}{kT}} \tag{3.3.13}$$

$$p_n(0) = N_a \mathrm{e}^{\frac{-q(V_0 - V)}{kT}} \tag{3.3.14}$$

$n_p(0)$	Electron concentration on the p-side at the edge of the depletion zone
$n_p(0)$	Hole concentration on the n-side at the edge of the depletion zone
N_d	Doping concentration on n-side
N_a	Doping concentration on p-side
q	Elementary charge
V_0	Built in contact potential for the pn junction [23]
V	Applied external voltage
k	Boltzmann constant
T	Temperature in Kelvin

Equations (3.3.13) and (3.3.14) give the concentrations of charge carriers at the edges of the depletion zone. More precisely they take into account the diffusion mechanisms through a short potential barrier. Short means here that the recombination rate within the depletion zone is negligible. The actual pn junction is between the n-doped layer and the indium containing p-doped layer. This would usually lead to complications because the two layers have a different bandgap. Such an asymmetry causes the diffusion barrier due to the depletion zone to be different for electrons and holes. In this case the electron diffusion would be increased at the expense of the hole diffusion. The electrons diffusing into the p-doped layers will however have to overcome an equivalent barrier when diffusing from the indium containing layer to the p-doped pure GaSb layers. For the emission in the pure GaSb containing layers this means that the effect of smaller bandgap in the indium containing layer is eliminated. In a sense the bandgap is equal on both sides of the barrier if it is considered to start at the depletion zone and extend all the way to the p-doped pure GaSb layers. This would however challenge the mentioned criterion that the barrier must be short for equation (3.3.13) and (3.3.14) to be valid. The pn junction will because of this be treated as if the barrier is equal for both holes and electrons. The only error introduced by this method is a too low electron concentration in the indium containing layer. As mentioned at the start of this section, the amount of emission in this layer does not need to be comparable to the other regions because it emits at another wavelength.

Only a relative measure of the hole to the electron concentrations is needed in this context. The common factors in equations (3.3.13) and (3.3.14) can therefore be contained in a single constant. Using the doping concentrations in Figure 3.2.1 therefore gives the relative numerical values for the minority carrier concentrations stated in equations (3.3.15) and (3.3.16).

$$n_p(0) \approx 1 \cdot K_1 \tag{3.3.15}$$
$$p_n(0) \approx 25 \cdot K_1 \tag{3.3.16}$$

K_1 Constant containing all the common factors in equation (3.3.13) and (3.3.14)

Diffusion

The next step is to find out how the minority charge is transported away from the pn junction and, how it is distributed over the layers in figure 3.2.1. In a uniform medium which extends far beyond the pn junction, this process is, according to [23], characterized by the diffusion length L_p for holes and L_n for electrons. The minority carrier concentration will then decay exponentially in the same way light does in an absorbing medium. For example the hole concentration in a n-doped region will decay as e^{-x/L_p} when x is the distance to the edge of the depletion region. The requirement for this to be true is that the semiconductor medium extends several diffusion lengths from the pn junction. If this is not true, and the semiconductor is instead terminated by a metal contact at a distance smaller than the diffusion length, the distribution is somewhat different. The metal contact will pin the minority carrier concentration to zero, and it will linearly increase toward the pn junction where it is given by equations (3.3.13) and (3.3.14). This is believed to be the situation on the p-doped side under the metal contacts. When instead the semiconductor medium is terminated by an insulator at a distance significantly smaller than the diffusion length, the concentration gradient is expected to fall in comparison to the situation where it decays exponentially. This is expected to be the situation on the p-side where the surface is not covered by metal contacts, and a constant minority carrier concentration will therefore be assumed.

Finding out which regime to use for the different layers requires the diffusion lengths L_p and L_n. According to [23] these can be found using their definition and the Einstein relation. The resulting expression for electrons (L_n) is given by equation 3.3.17. The equivalent expression for holes can be found by changing all n-indices to p.

$$L_n = \sqrt{\frac{\mu_n \tau_n k T}{q}} \tag{3.3.17}$$

L_n Electron diffusion length
μ_n Electron mobility [23]
τ_n Electron lifetime in p-doped material

Mobility

The mobility is a parameter which gives the speed of a charge carrier when subjected to an electric field. This is according to [23] dependent on both doping and temperature, but that doping loses its relevance at high temperatures. Measurements done by Mathur *et al* [15] suggest that a value of $2500\frac{\text{cm}^2}{\text{Vs}}$ would be reasonable for the electron mobility for all pure GaSb layers at room temperature. The same value will also be assumed for the indium containing layer since this is heavily dominated by GaSb. Ref. [26] has collected measurements of the hole mobility at room temperature from four sources. According to these measurements, the mobility is $430\frac{\text{cm}^2}{\text{Vs}}$ at a doping concentration of $2 \cdot 10^{18}$ and $610\frac{\text{cm}^2}{\text{Vs}}$ at a doping concentration of $3 \cdot 10^{17}$.

Carrier lifetime

The carrier lifetime is also needed in equation (3.3.17). This parameter expresses the mean time a minority carrier exists before it recombines with a majority carrier. This process can emit radiation, but there are also other non radiative sources of recombination. In this context the total lifetime is interesting. Stollwerck *et al* [22] have calculated the different lifetimes for both holes and electrons at room temperature. The electron lifetimes are shown in Figure 3.3.10, and the hole lifetimes are shown in Figure 3.3.11.

Table 3.3.2: Minority carrier diffusion length for different layers

Layer doping [cm^{-3}]	Diffusion length $L_{n/p}$ [μm]	Thickness [μm]
GaSb p = $2 \cdot 10^{18}$	4.7	0.3
GaSb p = $3 \cdot 10^{18}$	4.0	0.2
GaSb p = $6 \cdot 10^{18}$	2.8	0.2
GaSb p = $2 \cdot 10^{19}$	1.3	0.15
GaSb p = $5 \cdot 10^{19}$	0.62	0.15
GaInSb p = $5 \cdot 10^{18}$	0.62	0.1
GaSb n = $2 \cdot 10^{18}$	6	1.9
GaSb n = $3 \cdot 10^{17}$	13	500

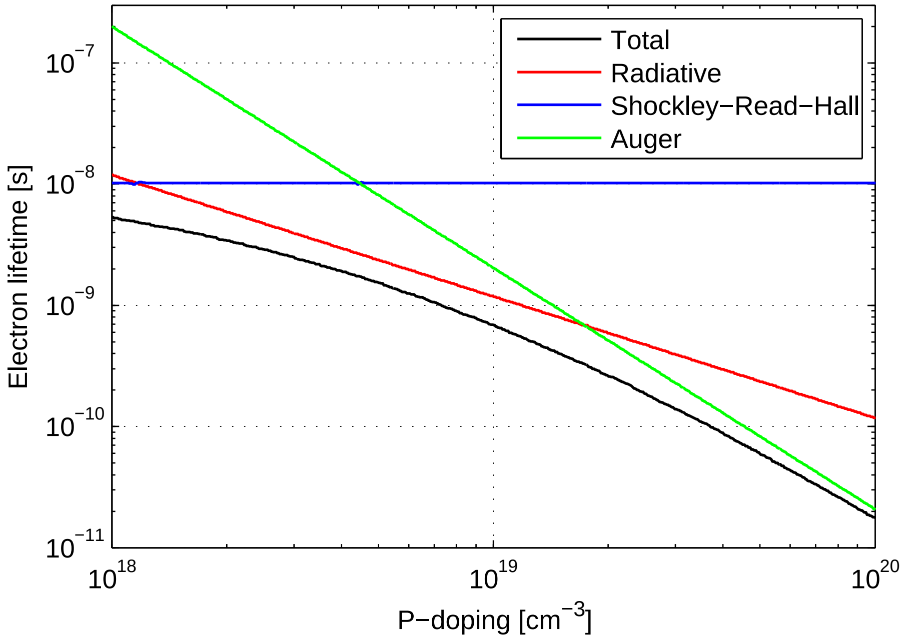


Figure 3.3.10: Calculated electron lifetime as a function of p-doping concentration. Calculations are performed by Stollwerck *et al* [22]

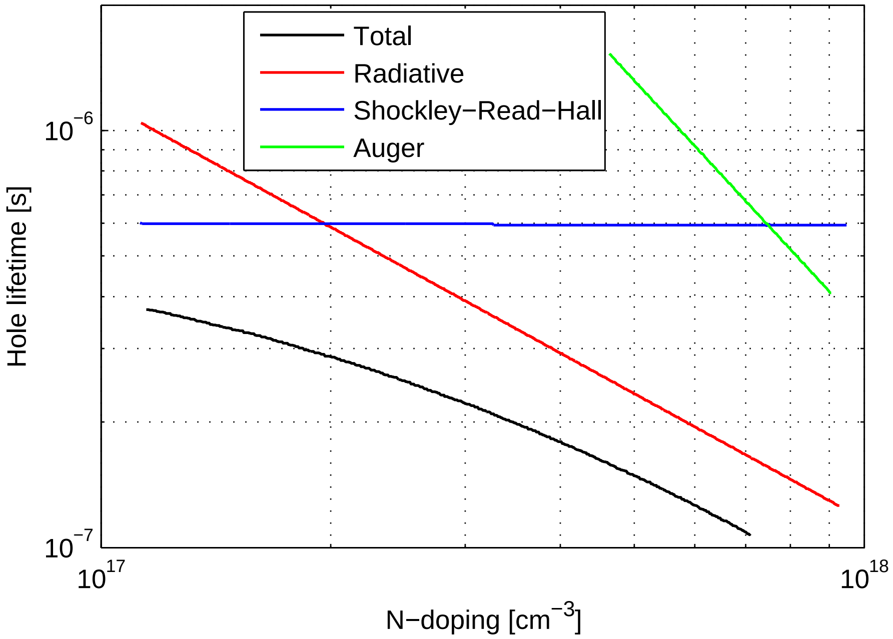


Figure 3.3.11: Calculated hole lifetime as a function of n-doping concentration. Calculations are performed by Stollwerck *et al* [22]

Diffusion lengths

Using the stated mobilities and carrier lifetimes, the diffusion lengths of the different layers are calculated using equation (3.3.17). The resulting values are stated in Table 3.3.2. The diffusion length in all grown layers is larger than the layer thicknesses. As discussed in a previous section, this causes a constant minority carrier concentration across the layers. Also the fact that the p-doped layers end with an intrinsic layer can lead to a more constant distribution than one would normally assume for an unbounded p-doped layer. This can be assumed since the intrinsic layer will act as a barrier and thereby cause pooling, or poor drainage, of the minority charge carriers. The n-doped side is on the other hand not bounded, and an exponential decay from the pn junction should be assumed. For simplicity, it will however be assumed that the minority charge concentration is constant to a depth of one diffusion length, and zero deeper into the substrate. The diffusion lengths for the grown layer and the substrate are in the same order of magnitude, and the value for the substrate will be chosen in this context since most of the charge containing region is in the substrate. In this way the substrate gets an effective optical thickness equal to its diffusion length subtracting the thickness of the overlaying grown layer. The electron concentration in the pure GaSb p-doped layers will therefore be given by equation (3.3.15) for all layers, and the hole concentration will be given by equation (3.3.16) for the top 13μm of the n-doped side.

Emission rates

As stated in equation (3.3.12), the emission rate per volume is also dependent on the *radiative* lifetime of the minority carriers. These are calculated by Stollwerck *et al* [22], and are shown in Figure 3.3.10 and Figure 3.3.11. As these are only calculations, the accuracy should not be expected to be very high. They should however give an indication of the emission rates for the different layers. Using equation (3.3.12), the minority carrier concentrations from equations (3.3.15) and (3.3.16), and the radiative lifetimes from Figures 3.3.10 and 3.3.11, gives the values for R_{spon} shown in Table 3.3.3. Note that the minority carrier lifetime for the grown n-doped layer is extrapolated since the curve in Figure 3.3.11 only covers doping concentrations up to $1 \cdot 10^{18}$. K_2 is a common scaling factor without denotation.

Table 3.3.3 sums up the discussion in this section and gives the necessary relative light emission rates needed to construct a radiation pattern for the entire structure.

Table 3.3.3: Photon emission rate at 0.7eV from different layers

Layer doping [cm^{-3}]	R_{spon}/K_2 [cm^{-3}s^{-1}]	Effective thickness [μm]
GaSb p = $2 \cdot 10^{18}$	1.7	0.3
GaSb p = $3 \cdot 10^{18}$	2.5	0.2
GaSb p = $6 \cdot 10^{18}$	5.1	0.2
GaSb p = $2 \cdot 10^{19}$	17	0.15
GaSb p = $5 \cdot 10^{19}$	43	0.15
GaInSb p = $5 \cdot 10^{18}$	-	0.1
GaSb n = $2 \cdot 10^{18}$	4.3	1.9
GaSb n = $3 \cdot 10^{17}$	0.64	11.1

Photon recycling

Before ending the discussion about material interaction the principle of photon recycling deserves some attention. When a photon is absorbed by the material, an electron is excited across the badgap, and an electron hole pair is created. This can again recombine and generate either thermal energy or a new photon. If the probability of generating a photon is high enough, the process which earlier has been described as absorption, is really isotropic scattering. Since the absorbed and emitted photon is not correlated in any way, this does not affect the theory for calculating radiation patterns discussed in previous sections. It merely provides another way of moving minority carriers, which will add to the effect of diffusion. According to von Roos [25], the effect can cause an effective diffusion length larger than otherwise expected, but that this effect is highly limited. von Roos did these calculations on GaAs, and they are therefore not directly transferable to GaSb. In this context the effect will be fully ignored because of the highly complicated theory required, and because of von Roos's conclusion that this effect is small in general. For the p-side of the diode, such an increased diffusion length would only make the applied assumptions more correct. For the n-doped side the effective thickness of the substrate should be slightly larger than what is calculated in previous sections. This would however only counteract the error introduced by assuming a constant diffusion length on the n-doped side.

3.3.6 Radiation pattern

Finding the combined radiation pattern from all layers is performed by summing their different contributions. Each layer must be represented by a term with the form of equation (3.3.10). The cap layer factor must also be split into a factor for each overlying layer. The resulting expression is shown in equation (3.3.18). Here the layers are numbered with number one assigned to the topmost p-doped layer in figure 3.2.1, and increasing by one for each step downwards.

$$I(\theta,\phi) = A_{LED}\sum_{i=1}^{N}\left[\frac{U_i}{\alpha_i}\left(1-\mathrm{e}^{-\frac{\alpha_i t_i}{\cos\theta}}\right)\cos(\theta)\cdot\mathrm{e}^{-\sum_{k=1}^{i-1}\frac{\alpha_k t_k}{\cos\theta}}\right] \tag{3.3.18}$$

t_i	Thickness of layer i
t_k	Thickness of layer k
U_i	Emission per unit volume for layer i
α_i	Absorption coefficient for layer i
α_k	Absorption coefficient for layer k
A_{LED}	Area of the LED surface

Equation (3.3.18) has been evaluated using layer thicknesses and emission per unit volume from Table 3.3.3 and absorption coefficients from Table 3.3.1. As the result contains a lot of terms, only the resulting plots are presented here. Figure 3.3.12 shows the result for 0.7eV, and Figure 3.3.13 shows the result for 0.65eV. The plots are normalized such that the power through the upper hemisphere is unity. These powers have been calculated by integrating equation (3.3.18) as stated in equation (3.3.3) using the trapezoidal method.

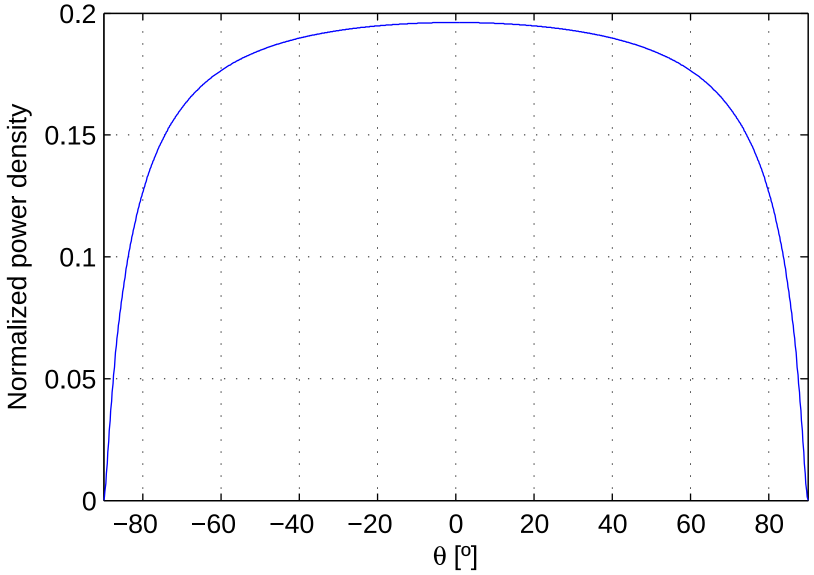


Figure 3.3.12: Normalized plot of equation (3.3.18) for light at 0.7eV. The plot is normalized such that the total power through the upper hemisphere is unity.

In section 3.3.2, a preferred way of representing the radiation pattern is presented. Instead of using the power per unit solid angle as a function of θ and ϕ, the power per unit area in the $k_x k_y$ plane as a function of k_x and k_y is used. Here k_x and k_y are the two transversal components of the k-vector. The transformation can be done

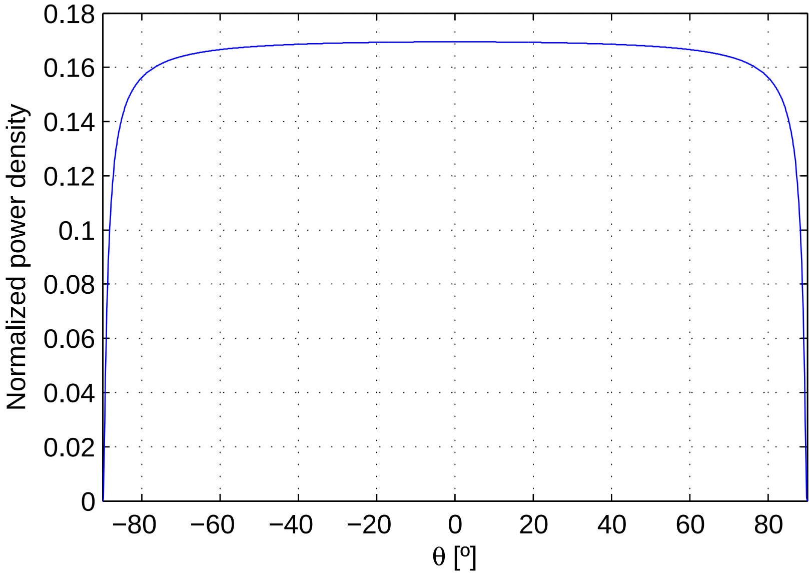


Figure 3.3.13: Normalized plot of equation (3.3.18) for light at 0.65eV. The plot is normalized such that the total power through the upper hemisphere is unity.

using equation (3.3.8), and the result is shown in Figure 3.3.14 and Figure 3.3.15 for 0.7eV and 0.65eV, respectively. These figures show the power density as a function of k_x with k_y equal to zero. This fully describes the entire power density function because it is still constant in the ϕ-direction. Since the transformation is wavelength dependent through the k-factor in equation (3.3.8), and because the range of k_x and k_y is wavelength dependent, the axis of Figure 3.3.14 and Figure 3.3.15 have been properly normalized. The x axis measures k_x/k and the y axis measures the power density multiplied by k^2.

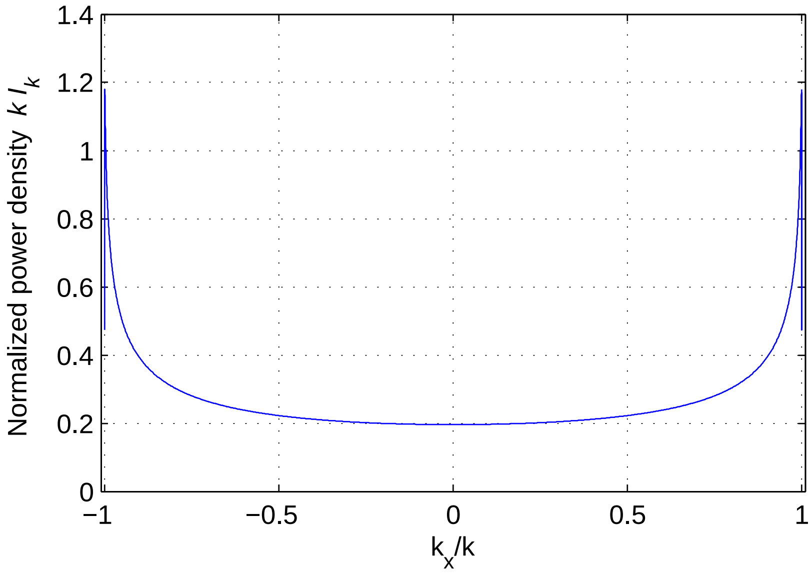


Figure 3.3.14: Radiated power per unit area in the $k_x\,k_y$ plane at a photon energy of 0.7eV

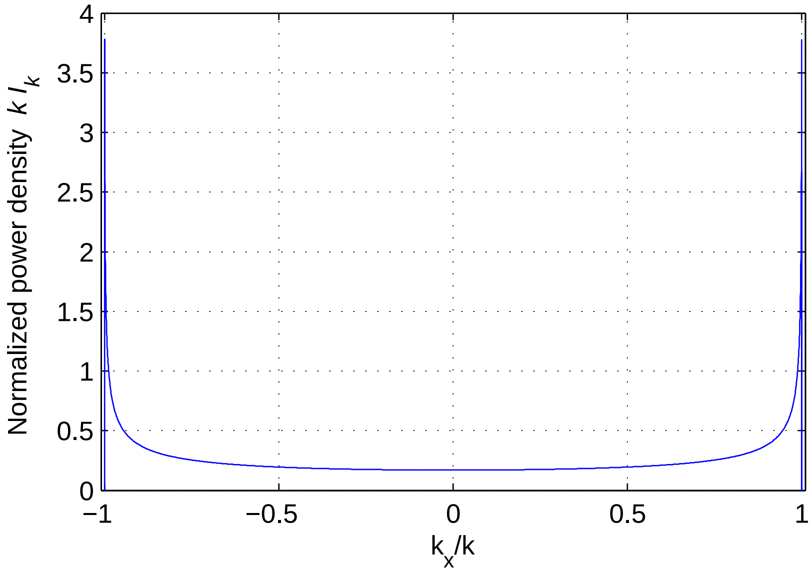


Figure 3.3.15: Radiated power per unit area in the $k_x\,k_y$ plane at a photon energy of 0.65eV

3.4 Transmission to air

The prime goal of this thesis is to find a measure of how much more of the optical power it is possible to extract from a LED by etching a photonic crystal grating on its surface. In this section it will be determined how much of the optical power is extracted from a LED with an unpatterned surface. To measure the transmission property of the surface, a parameter called the surface transmission efficiency, η_s, is defined. It is here defined as the ratio between the power exiting the top of the LED, and the total power incident on the semiconductor air interface. The transmission through this interface is governed by the Fresnel equations, which are derived in [20]. The transmission is dependent on the θ-angle of the light, the refractive indices of the two media on each side of the boundary, and the polarization state of the light. At a given propagation angle (θ, ϕ), all possible polarization states can be described as a linear combination of TE and TM-polarization. Pure TE-polarization is when the electric field is perpendicular to the plane of incidence or equivalently, parallel to the boundary and perpendicular to the propagation direction. Pure TM polarization is when the magnetic field possesses the mentioned properties. These two polarizations have independent transmission coefficients. Because the emitted light is assumed to be unpolarized, the optical power is assumed to be equally distributed over the two polarizations, and the combined power transmission coefficient is the arithmetic mean of the two. In ref. [20], the field reflection coefficients are given, and reproduced here in equation (3.4.1) for TE polarized light and in equation (3.4.2) for TM polarized light. In these equations n_1 is the refractive index of the GaSb semiconductor which Edwards *et al* [6] have measured to be 3.8 for the wavelengths of interest, and n_2 is the refractive index of air which has the numerical value 1.

$$r_{TE} = \frac{n_1 \cos\theta_1 - n_2 \cos\theta_2}{n_1 \cos\theta_1 + n_2 \cos\theta_2} \tag{3.4.1}$$

$$r_{TM} = \frac{n_2 \cos\theta_1 - n_1 \cos\theta_2}{n_2 \cos\theta_1 + n_1 \cos\theta_2} \tag{3.4.2}$$

n_1 Refractive index of the medium prior to the boundary (semiconductor)
n_2 Refractive index of the medium beyond the boundary (air)
θ_1 Propagation angle of the incident light, relative to the face normal of the boundary
θ_2 Propagation angle of the transmitted light, relative to the face normal of the boundary.

Equations (3.4.1) and (3.4.2) contain the angle θ_2 which is linked to the angle of incidence through Snell's law which is stated in equation (3.4.3).

$$n_2 \sin\theta_2 = n_1 \sin\theta_1 \tag{3.4.3}$$

$$\cos\theta_2 = \sqrt{1 - \left(\frac{n_1}{n_2}\right)^2 \sin^2\theta_1}$$

In this context, the most interesting form of these equations is the power transmission coefficients which are related to equations (3.4.1) and (3.4.2) through equation (3.4.4). The two polarizations are here also combined according to the discussion in the previous paragraph.

$$T = \frac{1}{2}\left(T_{TE} + T_{TM}\right) = \frac{1}{2}\left(1 - |r_{TE}|^2 + 1 - |r_{TM}|^2\right) \tag{3.4.4}$$

T	Combined transmission coefficients as a function of the angle of the incident light
T_{TE}	transmission coefficients for TE-polarized light
T_{TM}	transmission coefficients for TM-polarized light

The transmitted power density per unit solid angle for the incident light can be found by multiplying T in equation (3.4.4) with $I(\theta, \phi)$ from equation (3.3.18). θ_1 in equation (3.4.4) will then be the same as θ in equation (3.3.18). The easiest way of finding the total transmitted power is to integrate the transmission coefficient multiplied with the incident power density function per unit solid angle. The surface transmission efficiency can be found by dividing by the total incident power which can be found using equation (3.3.3). This is expressed in mathematical terms in equation (3.4.5).

$$\eta_s = \frac{\int\limits_{\substack{Upper\\hemisphere}} T(\theta,\phi) I(\theta,\phi) \mathrm{d}\Omega}{\int\limits_{\substack{Upper\\hemisphere}} I(\theta,\phi) \mathrm{d}\Omega} \tag{3.4.5}$$

η_s Surface transmission efficiency

The actual value of η_s is wavelength dependent, and here the values for the two photon energies 0.65eV and 0.7eV, discussed in section 3.3.6, will be calculated. Using the power densities shown in Figure 3.3.14 and Figure 3.3.15, and the transmission coefficients from equation (3.4.4) enables calculation of the actual η_s for the two wavelengths. Equation (3.4.5) can be simplified to equation (3.4.6) because both T and I are independent of ϕ, and because the transmission is zero above the critical angle θ_c. The integral in the nominator of equation (3.4.6) has been calculated using the trapezoidal method, and the denominator is unity because of the normalization used for the power density function. The results are shown in Table 3.4.1.

$$\eta_s = \frac{2\pi \int_{\theta=0}^{\theta_c} TI(\theta)}{P_{incident}} \tag{3.4.6}$$

θ_c Critical angle of incidence ($\arcsin(n_2/n_1)$)

Table 3.4.1: Surface transmission efficiency

Photon energy [eV]	Surface transmission efficiency
0.65	2.44%
0.7	2.82%

Table 3.4.1 shows that an unpatterned surface has an extremely low surface transmission efficiency. The main reason for this is that the high refractive index of the semiconductor causes a very small critical angle $\theta_c \approx 15^\circ$. Thus most of the light experiences total internal reflection. The surface transmission efficiency for the optical power at 0.65eV is smaller than what it is at 0.7eV. This is because more of the power is at large θ-angles at 0.65eV than at 0.7eV. Note that the surface transmission efficiency is only a measure of the transmittivity of the top surface. Other factors reducing the total efficiency of the LED, such as non radiative recombination, absorption, and the fact that half of the optical power is emitted downwards into the substrate, are not taken into consideration.

Chapter 4

Simulation model

4.1 Physical principle

4.1.1 General

This chapter will mainly discuss the simulation model used to calculate the effects of a photonic crystal on the top layer of a LED. However, this first section will focus on the physics of the photonic crystal. This is needed to make sure the simulation model incorporates all the physical effects. Figure 4.1.1 shows the structure which is to be examined. A two dimensional photonic crystal separates the bulk semiconductor from the surrounding air. As discussed in chapter 3, light is generated in the semiconductor, and is incident on the interface between the bulk semiconductor and the photonic crystal.

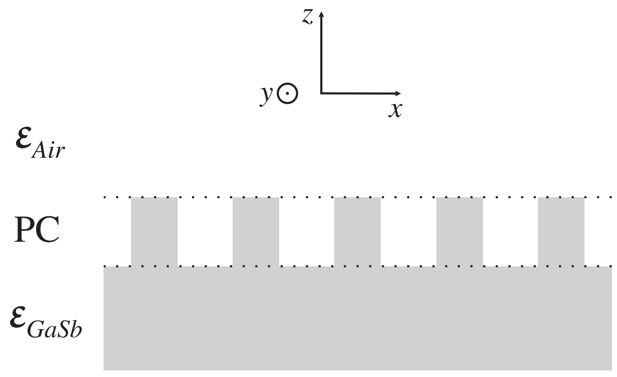


Figure 4.1.1: Photonic crystal (PC) at the semiconductor-air interface

The coupling mechanism from the semiconductor to the photonic crystal is based on the principle discussed in section 2.6. For light to couple to a mode in the photonic crystal its wave-vector parallel to the interface must satisfy equation (2.6.3).

Since it will be used extensively, the transversal wave-vector $\boldsymbol{k_T}$ is defined as the projection of the wave-vector on the interface plane. The word transversal is here used because the vector is transversal to the flow of energy which is of interest. As there are not any restrictions on the out of plane wave-vector component or the mode number, this makes light with a given $\boldsymbol{k_T}$-vector able to couple to all possible modes at a certain $\boldsymbol{k_B}$-point in the two dimensional Brillouin-zone. Note that since ω must be conserved, all these modes must have the same eigenvalue, however their out of plane wave-vector component can be different. If one want to imagine these modes in a band diagram, it is therefore beneficial to imagine curves defined by the intersection of a constant ω-plane and the possible mode surfaces. Figure 2.5.5 shows one such a mode plane, and a constant ω-plane would be parallel to the k_z-$\boldsymbol{k_B}$-plane. The number of possible modes would therefore be limited and increase with increasing ω.

At the interface between the photonic crystal and air, the modes at a certain $\boldsymbol{k_B}$-point can couple to plane waves in air according to (2.6.3). This means all these modes can couple to all plane waves with $\boldsymbol{k_T}$-vectors equal to those which coupled to the photonic crystal in the first place. This enables light to couple to plane waves in air despite it having a $\boldsymbol{k_T}$-vector in the semiconductor which is longer than the wave number k_0 in air. In the case of a plane surface, this would cause total internal reflection. If there is a combination of primitive reciprocal lattice vectors which when added to the $\boldsymbol{k_T}$-vector in the semiconductor results in a $\boldsymbol{k_T}$-vector which is shorter than k_0, coupling to air is possible. When studying equation (2.6.3) it is also possible conclude that if all points within the Brillouin zone have a $\boldsymbol{k_B}$-vector which have a length smaller than k_0, there will always be such a combination of primitive reciprocal lattice vectors.

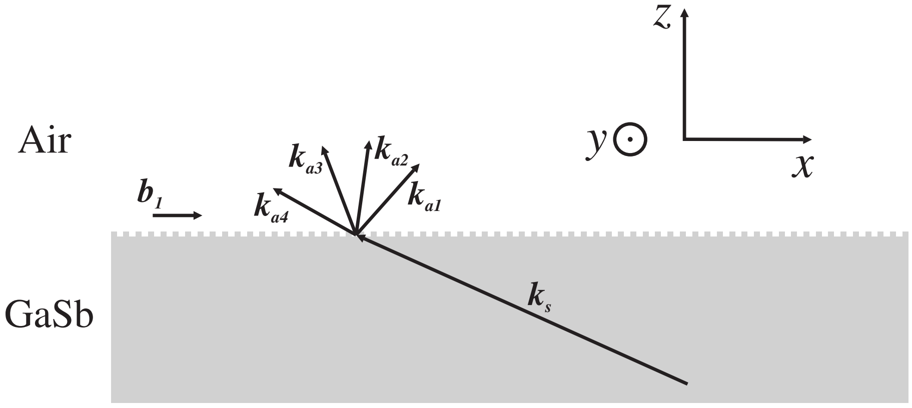


Figure 4.1.2: Diffraction through a photonic crystal surface

Figure 4.1.2 shows for one dimension, a plane wave in the semiconductor with wave-vector $\boldsymbol{k_s}$ that couples to several plane waves in air which have wave-vectors $\boldsymbol{k_{ai}}$ for $i = 1, 2, 3, 4$. The conclusion of this discussion is that a plane wave inside the semiconductor can couple out to plane waves in air which have wave-vectors which differ from the original wave-vectors by an integral number of primitive reciprocal lattice vectors. Using this rule, it is not really necessary to consider the modes of the photonic crystal, or the out of plane wave-vector component. The only essential parameter of the photonic crystal is its primitive reciprocal lattice vectors $\boldsymbol{b_i}$. All the other parameters will influence the coupling as well, but the $\boldsymbol{b_i}$-vectors apply the strictest limitations to how light couples through the surface.

4.1.2 Multiple incident plane waves

The previous section considered what happens when *one* plane wave from the semiconductor is incident on the photonic crystal. Since it in chapter 3 was concluded that light with a spectrum of wave-vectors is generated it is also necessary to consider what happens when multiple plane waves are incident on the surface. A given plane wave in air with wave-vector $\boldsymbol{k_a}$, and transversal wave vector $\boldsymbol{k_{Ta}}$ can get optical power from plane waves in the semiconductor which have transversal wave vectors $\boldsymbol{k_{Ts}} = \boldsymbol{k_{Ta}} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}$ when m_1 and m_2 are integers. Since typically a number of plane waves in the semiconductor satisfies this condition, their individual contribution must be summed. Considering the spatial Fourier transform of the scalar electric field, $E_{ks}(\boldsymbol{k_T})$, inside the bulk semiconductor and $E_{ka}(\boldsymbol{k_T})$ in air, equation (4.1.1) states this summation in mathematical terms. The scalar fields are here used for simplicity. The effect of polarization will be accounted for later.

$$E_{ka}(\boldsymbol{k_T}) = \sum_{m_1}\sum_{m_2} \gamma(\boldsymbol{k_T}, m_1, m_2)\, E_{ks}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \tag{4.1.1}$$

$\boldsymbol{k_T}$	Transversal wave-vector. Projection of the wave-vector on the interface plane.
$\gamma(\boldsymbol{k_T}, \boldsymbol{m_1}, \boldsymbol{m_2})$	Coupling coefficient.
$E_{ka}(\boldsymbol{k_T})$	Spatial Fourier transform of the scalar electric field in air.
$E_{ks}(\boldsymbol{k_T})$	Spatial Fourier transform of the scalar electric field in bulk semiconductor.
$\boldsymbol{b_1}, \boldsymbol{b_2}$	Reciprocal lattice vectors of the photonic crystal.

Finding the power flow through the surface would in reality involve the field directions and the dielectric constants of the materials in question. Although it is possible to account for these, it would complicate the equations. Since the point which will be made is not dependent on these effects they will therefore here be omitted. With these simplifications, the power flow is the square of the absolute value of equation 4.1.1. Equation (4.1.2) states this by multiplying with the complex conjugate. The four necessary summation symbols are here represented by only one because of space constraints.

$$
\begin{aligned}
S_{ka}(\boldsymbol{k_T}) &= |E_{ka}(\boldsymbol{k_T})|^2 \\
&= \sum_{\substack{m_1,\ m_2 \\ \hat{m}_1,\ \hat{m}_2}} \gamma(\boldsymbol{k_T}, m_1, m_2)\, E_{ks}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \cdot \gamma^*(\boldsymbol{k_T}, \hat{m}_1, \hat{m}_2)\, E^*_{ks}(\boldsymbol{k_T} + \hat{m}_1\boldsymbol{b_1} + \hat{m}_2\boldsymbol{b_2}) \\
&= \sum_{m_1,\ m_2} |\gamma(\boldsymbol{k_T}, m_1, m_2)|^2\ S_{ks}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \qquad (4.1.2) \\
&+ \sum_{\substack{m_1,\ m_2 \\ \hat{m}_1,\ \hat{m}_2 \\ m_1 \neq \hat{m}_1 \\ m_2 \neq \hat{m}_2}} \gamma(\boldsymbol{k_T}, m_1, m_2)\, \gamma^*(\boldsymbol{k_T}, \hat{m}_1, \hat{m}_2)\, E_{ks}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2})\, E^*_{ks}(\boldsymbol{k_T} + \hat{m}_1\boldsymbol{b_1} + \hat{m}_2\boldsymbol{b_2})
\end{aligned}
$$

$S_{ka}(\boldsymbol{k_T})$	Spatial Fourier transformed power density in air.
$S_{ks}(\boldsymbol{k_T})$	Spatial Fourier transformed power density in bulk semiconductor.

As usual when multiple sources of radiation are combined into one, the combined power density is a scaled sum of the power densities of the sources, and several interference terms. Take note that the interference here is caused by combining light generated by the same source, at roughly the same time, and at the same positions. The effect is therefore not characterized by temporal coherence or transversal spatial coherence. The spatial field distribution does determine the interference, but not in a way that is directly linked to spatial coherence. Instead it is necessary to think of the coherence in the transversal wave-vector space. The next section will analyze these properties in more detail.

4.1.3 Coherence in wave-vector space

This section will look at the statistical coherence between light with different $\boldsymbol{k_T}$. The analysis will be done with $\boldsymbol{k_T}$ in one dimension instead of two in order to ease notation and to provide a better overview. The results are readily translatable to two dimensions. The x-direction will therefore be chosen as the transversal direction in accordance with Figure 4.1.1. The $\boldsymbol{k_T}$-vector can then be replaced with the scalar k_x. A common way of specifying the coherence in the time domain is the autocorrelation function. A similar function can be defined for the k_x-dimension, and is here stated in equation (4.1.3).

$$G(q) = \int_{-\infty}^{\infty} E_{ks}^{*}(k_x) \cdot E_{ks}(k_x + q)\, \mathrm{d}k_x \tag{4.1.3}$$

$G(q)$ Autocorrelation function.
q Argument of G which specifies the shift in k_x of which G is the covariance.

$G(q)$ in equation (4.1.3) is a measure of how similar the amplitude is for a plane wave with transversal wave-vector k_x compared to one with wave vector $k_x + q$. The value at a specific q is proportional to the mean covariance obtained by integrating over k_x. $G(0)$ will therefore be the total power, and the function will typically fall towards zero for both increasing and decreasing q. The width around the peak near $q = 0$ tells how much difference there must be in k_x for two plane waves before their amplitudes are incoherent. The Wiener-Khinchin theorem [20] can be used to find the function $G(q)$ based on available parameters. Since it is not too complicated, and since it is rarely used in this context, it will be shown here. The plane wave amplitude $E_{ks}(k_x)$ is related to the regular electric field as a function of position, $E_s(x)$ through Fourier transform as stated in equations (4.1.4) and (4.1.5).

$$E_s(x) = \mathcal{F}^{-1}\left\{E_{ks}(k_x)\right\}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} E_{ks}(k_x)\, \mathrm{e}^{jk_x x}\, \mathrm{d}k_x \tag{4.1.4}$$

$$E_{ks}(k_x) = \mathcal{F}\left\{E_s(x)\right\}(k_x) = \int_{-\infty}^{\infty} E_s(x)\, \mathrm{e}^{-jk_x x}\, \mathrm{d}x \tag{4.1.5}$$

$\mathcal{F}$ Fourier transform operator

Using equation (4.1.5) to replace $E_{ks}(k_x)$, and changing the order of some of the integrals transforms equation (4.1.3) to equation (4.1.6). Recognizing the last integral as the Dirac delta function, and using this to solve the ξ-integral leads to the conclusion that $G(k_x)$ *is the Fourier transform of the transversal intensity profile.*

$$G(q) = \int_{x=-\infty}^{\infty} \int_{\xi-\infty}^{\infty} E_s^{*}(x) \cdot E_s(\xi)\, \mathrm{e}^{-q\xi} \underbrace{\int_{q=-\infty}^{\infty} \mathrm{e}^{jk_x(x-\xi)}\, \mathrm{d}k_x}_{\delta(x-\xi)}\, \mathrm{d}\xi\, \mathrm{d}x \tag{4.1.6}$$

$$= \int_{x=-\infty}^{\infty} E_s^{*}(x) \cdot E_s(x)\mathrm{e}^{-jqx}\mathrm{d}x = \mathcal{F}\left\{I_s(x)\right\}(q) \tag{4.1.7}$$

This means that the width of the intensity profile in a plane is inversely proportional to the width of $G(q)$. One example of this is a colinear laser beam with a relatively wide intensity profile. Since the beam is colinear, the plane wave amplitude $E_k k_x$ must decay very rapidly as $|k_x|$ increases. This again means that q in equation (4.1.3) must be very small to make sure both E_k-factors are non-zero for at least parts of the integral. A wide laser beam is therefore highly incoherent in the transversal wave vector space. An opposite example is a point source with an intensity profile width which is near-to-zero. The point source radiates isotropically, meaning $E_k(k_x)$ do not decay as $|k_x|$ increases. In addition all the plane waves are caused by the one and only oscillation, meaning they must have correlated amplitudes, and consequently $G(q)$ becomes a wide function. A point source is therefore highly coherent in the transversal wave vector space.

The width of $G(q)$, Δq is according to general properties of the Fourier transform approximately equal to the inverse of the width of the intensity profile Δx, which is stated in equation (4.1.8).

$$\Delta q = \frac{1}{\Delta x} \tag{4.1.8}$$

Δq Width of $G(q)$
Δx Width of $I(x)$

The interference terms in equation (4.1.2) are significant if Δq is larger than the primitive reciprocal lattice vectors $\boldsymbol{b_i}$. For the 1D case the one and only primitive reciprocal lattice vector is according to equation (2.5.3) equal to $2\pi/a$ where a is the grating period. Using this and equation (4.1.8) it is possible to transform the requirement into a minimum number of grating periods N_{per}.

$$\frac{2\pi}{a} > \Delta q = \frac{1}{\Delta x} \tag{4.1.9}$$

$$N_{per} = \frac{\Delta x}{a} > \frac{1}{2\pi} \tag{4.1.10}$$

N_{per} Number of periods in the grating (Width measured in grating periods).

Since the slopes of $G(q)$ are not necessarily steep, it is reasonable to add some clearance to the requirement in equation (4.1.10). However since the grating under investigation will have between 100 and 1000 periods, it will be assumed that all the interference terms in equation (4.1.2) are zero. The resulting equation describing the transmission through the two dimensional photonic crystal is therefore the one given in equation (4.1.11).

$$S_{ka}(\boldsymbol{k_T}) = \sum_{m_1,\ m_2} \Gamma(\boldsymbol{k_T}, m_1, m_2)\, S_{ks}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \qquad (4.1.11)$$

$\Gamma(\boldsymbol{k_T}, m_1, m_2)$ Power coupling coefficient.

4.1.4 Polarization

The previous section analyzed how multiple plane waves can mix and form one plane wave when moving through a two dimensional photonic crystal. The effects of polarization were however not discussed, and will therefore be lightly treated here. The polarization state of a plane wave can generally be described by a sum of two orthogonal linear polarization states. When describing light transmission through plane surfaces, the two polarizations used are typically TE and TM polarization. TE means transversal electric, and denotes light with an electric field transversal to the plane of incidence, and consequently an electric field parallel to the plane surface. TM means transversal magnetic, and denotes light where the magnetic field has the same properties as the electric field has for TE polarized light. As stated in equations (3.4.1) and (3.4.2), these two polarizations have different transmission coefficients when passing through an interface, and it is reasonable to believe the same for a photonic crystal surface. When either of the two polarizations are transmitted through a planar surface the polarization state is preserved. If the incident light is TE polarized, then the transmitted light is also TE polarized. This is however not necessarily true for a photonic crystal.

An effect of this is that transmitted light with TE polarization might originate from both TE and TM polarized light below the photonic crystal. Therefore the the question about interference effects when combining the two polarizations will appear in a similar way as the possibility for interference between different plane waves. It will be assumed that there is not any correlation between the TE and TM polarized light inside the LED. This effectively eliminates interference terms. Under these assumptions the introduction of polarization is merely a mild complication which doubles the terms in equation (4.1.11). Equations (4.1.12) and (4.1.13) state this new relation between incident and transmitted light. There are two equations which each describe the power of a transmitted plane wave with a certain polarization in air.

$$
\begin{aligned}
S_{ka,TE}(\boldsymbol{k_T}) = & \sum_{m_1,\ m_2} \Gamma_{TE \to TE}(\boldsymbol{k_T}, m_1, m_2)\, S_{ks,TE}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \\
& + \sum_{m_1,\ m_2} \Gamma_{TM \to TE}(\boldsymbol{k_T}, m_1, m_2)\, S_{ks,TM}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \qquad (4.1.12)
\end{aligned}
$$

$$
\begin{aligned}
S_{ka,TM}(\boldsymbol{k_T}) = & \sum_{m_1,\ m_2} \Gamma_{TM \to TM}(\boldsymbol{k_T}, m_1, m_2)\, S_{ks,TM}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \\
& + \sum_{m_1,\ m_2} \Gamma_{TE \to TM}(\boldsymbol{k_T}, m_1, m_2)\, S_{ks,TM}(\boldsymbol{k_T} + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}) \qquad (4.1.13)
\end{aligned}
$$

$S_{ka,TE}(\boldsymbol{k_T})$	Power density in air of TE polarized light.
$S_{ka,TM}(\boldsymbol{k_T})$	Power density in air of TM polarized light.
$S_{ks,TE}(\boldsymbol{k_T})$	Power density in bulk semiconductor of TE polarized light.
$S_{ks,TM}(\boldsymbol{k_T})$	Power density in bulk semiconductor of TM polarized light.
$\Gamma_{Tv \to Tw}(\boldsymbol{k_T}, m_1, m_2)$	Coupling coefficient for coupling of Tv polarized light in bulk semiconductor to Tw polarized light in air.

4.2 Simulation principle

4.2.1 Fresnel principle

The main purpose of the simulations described here is to find the light extraction from a LED with a 2D photonic crystal surface. In section 3.4, the surface transmission efficiency was calculated for a planar surface. The method used there was to construct an angular radiation pattern, and multiply it by the transmission coefficients given by the Fresnel equations. Integrating the result over the upper hemisphere then resulted in the total transmitted power, and the surface transmission efficiency was this power divided by the total power incident on the surface. The same principle will be used for the photonic crystal surface. The angular distribution of the incident light will be the same, but since there are no Fresnel equations giving the transmission coefficients, these will be obtained by simulation. The simulation task will therefore be to obtain transmission coefficients for different angles of incidence and polarization. Instead of operating in the angular domain, the principles stated in section 3.3.2 will be used to operate in the transversal wave-vector domain. The simulations can not be expected to produce transmission coefficients valid for the entire two dimensional continuous domain of angles of incidence. Instead it will be necessary to find coefficients for a discrete set of angles or values of $\boldsymbol{k_T}$. This will be done by making a uniform grid of points in the two dimensional space of $\boldsymbol{k_T}$. Because the maximum length of $\boldsymbol{k_T}$ is limited by the wavenumber in bulk semiconductor k_s, the point outside the circle of radius k_s will be omitted. In addition the photonic crystals which will be simulated will have a number of rotational and mirror symmetry-properties which can reduce the needed domain even further. Figure 4.2.1 illustrates this for a square photonic crystal with mirror symmetry axis along the x, y and the diagonal direction. Figure 2.5.2(a) illustrates an example of such a structure.

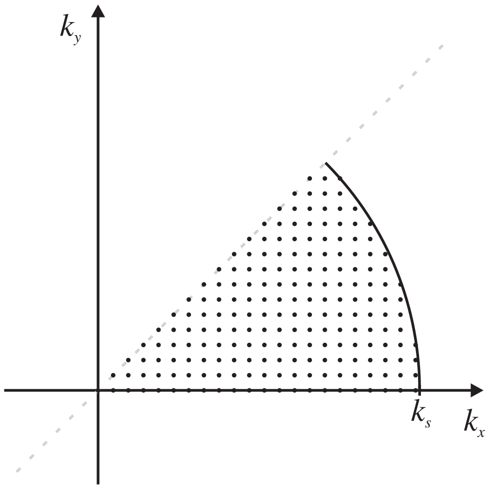


Figure 4.2.1: Grid of points which will have their transmission coefficients simulated for a square photonic crystal with three mirror symmetry axis

4.2.2 FDTD

The simulation software which was used for the simulations described in this thesis has the name MEEP [7], and is based on the finite-difference time-domain (FDTD) method. The key concept is to consider a finite space, and to divide it into a grid of voxels which are a 3D version of pixels. Each voxel holds a value of $\varepsilon(\boldsymbol{r})$ valid for its volume, and a value of all three components of the $\boldsymbol{H}(\boldsymbol{r})$ and $\boldsymbol{E}(\boldsymbol{r})$ fields. The Maxwell equations are approximated to form a set of difference equations which are used to update the fields in small time steps. In this way the software works by mimicking real wave propagation. The procedure is somewhat analog to to Euler's method for solving differential equations [14].The fields are excited by manipulating the $\boldsymbol{J}$-term in the Maxwell's equation for specific voxels in the grid. As the density of voxels, or resolution, is increased the result is supposed to converge to the exact solution. MEEP utilizes an algorithm for smoothing of the dielectric function in space which according to Farjadpour [7] makes the results converge quadratically to the actual solution. One of the main advantages of FDTD simulations is that since it operates in the time domain, the results from one simulation can give information about the properties of the structure over a wide band. Since it operates by approximating actual wave propagation it can be used for several purposes, and one very common is to find transmission and reflection coefficients. These are normally found by running two simulations. One with the scattering structure under investigation, and one without which serves as a reference that is used to calibrate the source.

4.2.3 Boundary conditions

In the previous section it was stated that the simulated space must be finite. This implies a need for boundary conditions on the edges of the simulated volume. The simplest forms are hard mirror walls which pin either the parallel magnetic or electric field to zero.

Another possibility which is more suitable for the current problem are periodic boundary conditions (PBC). These use the fields on the left side as boundary conditions for the right side and vice versa. The top and bottom side, and the front and rear sides also form identical pairs. The structure would therefore feel an identical structure with identical excitation right next to it. An infinite periodic structure can in this way be simulated only by considering one unit cell. As outlined here this method does however only take modes with $\boldsymbol{k_B} = 0$ in equation (2.5.10) into consideration. To account for the rest, the fields on the left side is multiplied by a phase constant $\mathrm{e}^{j\phi_1}$ before they are used as a boundary condition on the right side. The inverse phase constant $\mathrm{e}^{-j\phi_1}$ is used for the opposite direction, and different constants are used for the other two directions. If the dimensions of the unit cell is D_1, D_2 and D_3 in the first, second, and third direction respectively, and the phase coefficients used are ϕ_1, ϕ_2 and ϕ_3, then equation (4.2.1) gives the elements of $\boldsymbol{k_B}$ of the modes that are taken into account. In this way all modes can be simulated for an infinite structure by only considering one unit cell. The downside is that only one $\boldsymbol{k_B}$ can be considered simultaneously. Figure 4.2.2 illustrates this principle for a 2D unit cell. Since the simulation software knows the dimensions of the unit cell, the phase coefficients are set indirectly by specifying the $\boldsymbol{k_B}$-vector. In the following sections this kind of boundary conditions will be called PCB with a certain $\boldsymbol{k_B}$.

$$\begin{aligned} k_{B1} &= \phi_1/D_1 \\ k_{B2} &= \phi_2/D_2 \\ k_{B3} &= \phi_3/D_3 \end{aligned} \tag{4.2.1}$$

k_{B1}, k_{B2}, k_{B3}	The elements of $\boldsymbol{k_B}$ of the modes accounted for in the simulation.
ϕ_1, ϕ_2, ϕ_3	Phase added/subtracted when generating boundary conditions.
D_1, D_2, D_3	The length of the unit cell in all three directions.

In addition to mirror walls and PBC, perfectly matched layers (PML) can be used to form the desired boundary conditions. These are not really boundary

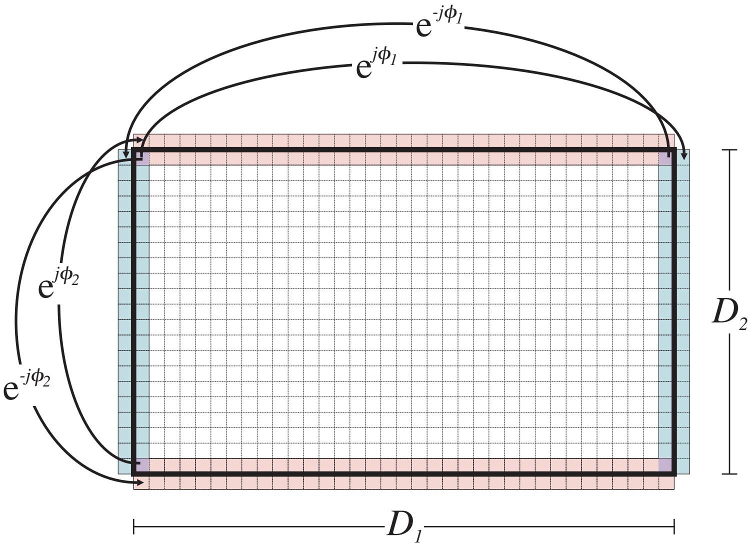


Figure 4.2.2: Principle of PBC for a 2D unit cell

conditions, and must be used together with mirror walls or PBC. Instead a layer of finite thickness is placed between the interior of the simulation cell and the walls. Waves which are excited within the cell will start decaying exponentially once they enter the PML. Before they reach the actual boundary, their field strengths should be satisfactorily attenuated such that the actual boundary condition will not have any effect. One can say that the PML insulates the interior of the unit cell from the boundaries. The unit cell will therefore see boundaries padded with PML as an infinite empty medium which does not reflect any radiation. It should be noted that PML is not simulated as a dielectric with an absorption coefficient. For example waves incident on the PML will in the limit of infinite resolution, not be partially reflected like they would if there was a transition in the complex refractive index. There are also more differences between PML and ordinary absorbing media which will not be discussed here.

4.3 Simulation structure

4.3.1 General

This section will describe the actual simulation structure used for finding the properties of the photonic crystal surface of the LED. In section 4.2.1, the principle was outlined. The simulations must provide transmission coefficients for different incident plane waves. The typical way of achieving this in a FDTD simulation is to excite a broadband pulse under the surface and record the transmitted waves. These can subsequently be compared to the total incident power to obtain a transmission coefficient. The total incident power is most easily obtained by performing a reference simulation which basically is the same, only without the surface. Figure 4.3.1 illustrates a cross section of the applied model and its reference.

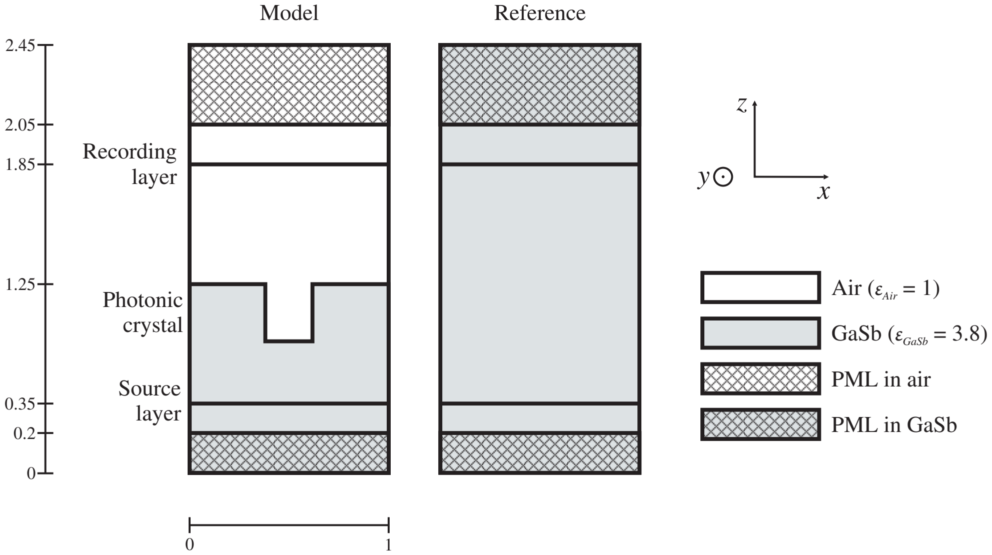


Figure 4.3.1: Simulation model and the reference structure. The units of the dimensions are explained in section 4.5.1.

4.3.2 Excitation and boundary conditions

A wave is excited by current densities in the source layer. In addition to the physical electric current density $\boldsymbol{J}$ in equation (2.2.2), excitation can also be achieved using magnetic current densities. This is a source term added to equation (2.2.1) such that it becomes equal to equation (4.3.1). Note that in order to distinguish between electric and magnetic current densities they are denoted $\boldsymbol{J_e}$ and $\boldsymbol{J_m}$ respectively.

$$\nabla\times\boldsymbol{E_t} + \frac{\partial\boldsymbol{B_t}}{\partial t} = -\boldsymbol{J_{t\,m}} \tag{4.3.1}$$

$\boldsymbol{J_{t\,m}}$ Magnetic current density.
t-index denotes time dependent variable instead of frequency dependent.

Even though the $\boldsymbol{J_m}$-term is non physical, it is a convenient way of exciting the desired waves. In fact the explicit use of either $\boldsymbol{J_e}$ or $\boldsymbol{J_m}$ in such simulations will in most cases not be related to the physical origin of the wave anyway. It is just a method of achieving an appropriate wave for the specific simulation.

Periodic boundary conditions are applied to the model, and it is therefore effectively repeated indefinitely in all directions. There are however PML layers at the top and bottom which insulate the repeated blocks in the z-direction. Therefore, the effective structure being simulated is that shown in Figure 4.3.1 repeated indefinitely in the x and y-directions. The source layer generates fields with a certain $\boldsymbol{k_T}$-vector by linearly modulating the phase over the plane according to equation (4.3.2).

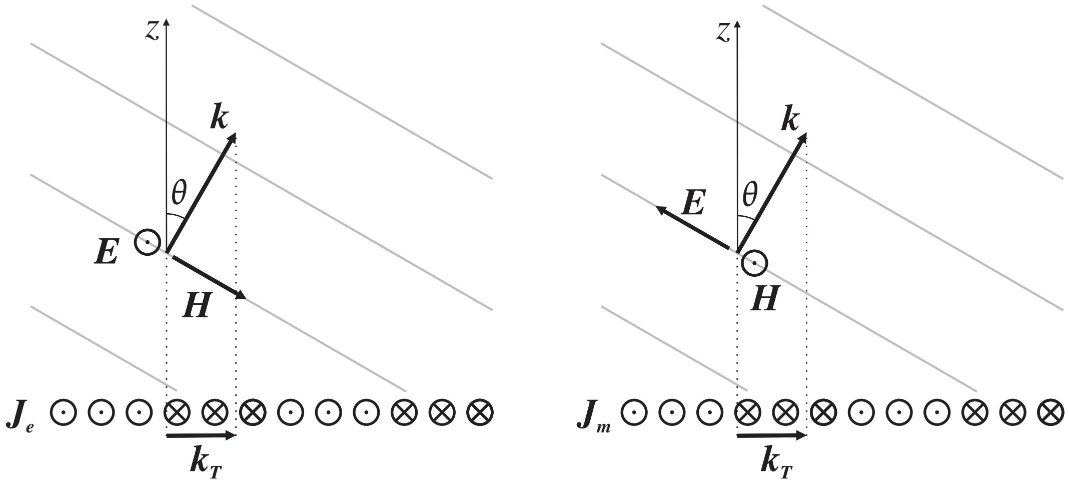
$$\boldsymbol{J} = \boldsymbol{J}_0\, \mathrm{e}^{j\boldsymbol{k_T r}} = \boldsymbol{J}_0\, \mathrm{e}^{j(k_x x + k_y y)} \tag{4.3.2}$$

$\boldsymbol{J}$	Magnetic or electric current density in the source plane
$\boldsymbol{J}_0$	Current density amplitude
$\boldsymbol{k_T}$	Transversal wave vector
$\boldsymbol{r}$	Position vector in the source plane
k_x	Component of $\boldsymbol{k}$-vector in the x-direction of the excited wave
k_y	Component of $\boldsymbol{k}$-vector in the y-direction of the excited wave

The $\boldsymbol{k_B}$ of the PBC must then be chosen such that the current density on the effective infinite source plane is a continuous function. This is most easily achieved by specifying $\boldsymbol{k_B}$ equal to $\boldsymbol{k_T}$. This can also be done when $\boldsymbol{k_T}$ extends outside one unit cell in the reciprocal lattice of the 2D photonic crystal. In fact adding an arbitrary integral number of reciprocal lattice vectors to $\boldsymbol{k_B}$ will not alter the boundary conditions because the added phase constants in Figure 4.2.2 do not change. This also means that the entire set of plane waves on the upper side of the photonic crystal will have suitable boundary conditions as long as they are suitable for the incident plane wave.

4.3.3 Polarization and beam steering

For a given optical frequency the source plane excites a plane wave where the angles of propagation θ and ϕ are specific. In fact equations (3.3.1) and (3.3.2) states the relation between the excited $\boldsymbol{k_T}$ and the angles. The simulation must be conducted separately for two orthogonal polarizations. To ensure mutual orthogonality, pure TE and TM waves are excited. TE waves are excited using only electric currents for the source, and TM waves are excited using only magnetic currents. For both TE and TM, the direction of $\boldsymbol{J}_0$ in equation (4.3.2) is orthogonal to k_T, but still within the xy-plane of the source. This also ensures a non diverging current density which otherwise would cause complications since the charge density would be non zero. When using only one of the available currents simultaneously it is impossible to direct the optical power upwards in the model. If both were used according to *Love's Equivalence Principle* [1], a pure upwards propagating wave could be excited. This would however require a definite angle of propagation θ. Since the excited pulse is highly broadband and $\boldsymbol{k_T}$ is pinned, this angle is different for each optical frequency. Therefore only one of the currents are used simultaneously, and the lower side PML must be used to absorb the downwards emitted wave. Figure 4.3.2 illustrates the upwards moving part of an excited monochromatic TE and TM wave.

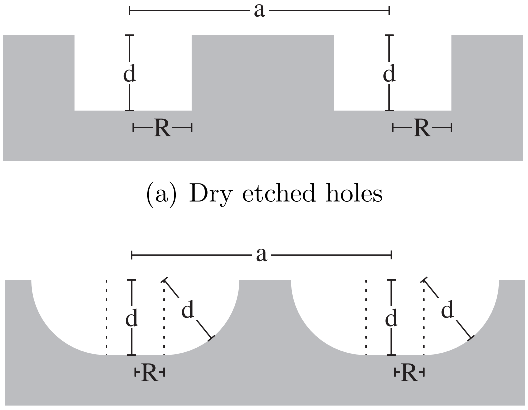


(a) TE wave (b) TM wave

Figure 4.3.2: Cross section illustration of excited monochromatic TE and TM waves

4.3.4 Grating structures

Figure 4.3.1 shows a typical cross section of the 3D structure being simulated. This is the unit cell of the simulated photonic crystal which is under investigation. The simulations described here are focusing on three kinds of photonic crystals. These are a square grating of holes, a hexagonal grating of holes, and a square grating of isotropically etched holes. The forms of the two first are shown in Figure 2.5.2. Both of these are typically cylindrically shaped holes in the surface. They can be fabricated using photolithography to protect the surrounding areas, but leaving an opening for the holes. They can subsequently be etched out using an anisotropic dry etch method such as sputter etching or reactive ion etching. These gratings are typically characterized by the radius and depth of the holes and the lattice constant. The last grating which is a square grating of isotropically etched holes can be made using the same kind of photolithography. The difference is that the holes are etched using a liquid chemical which etches equally deep in all directions. The hole radius and depth, and the lattice constant are sufficient for characterizing these holes as well, but here it is important to understand that all the material that is closer to the exposed area than the hole depth will be etched away. Figure 4.3.3 illustrates the holes caused by both these etching techniques. Here a is the lattice constant, d is the hole depth, and R is the hole radius.



(a) Dry etched holes

(b) Isotropically etched holes

Figure 4.3.3: Cross section of the simulated gratings.

The simulation software, MEEP, which was used for this work only allows cuboid shaped models, which means only rectangular unit cells can be simulated. For the square gratings, this is not an issue, but the unit cell of the hexagonal grating can not be used directly. To overcome this problem a slightly larger unit cell is defined as indicated in Figure 4.3.4. The length D_x is equal to the lattice constant a, while D_y is equal to $\sqrt{3}a$. Compared to square gratings, the hexagonal gratings will have a longer total running time of the simulations because the unit cell is larger, and because it lacks the diagonal symmetry line.

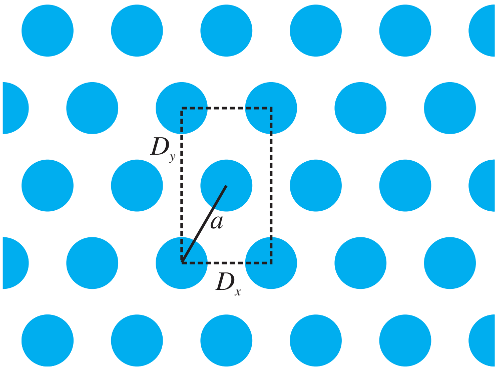


Figure 4.3.4: Rectangular unit cell of hexagonal grating

4.3.5 Simulation series

A simulation series is the set of simulations of a certain kind of structure. A total of six such series are described here, and each of the three grating structures presented in section 4.3.4 had one such series. These three can be regarded as the primary simulation series described here. The other three secondary series which were meant as a more detailed investigation of the square grating of holes illustrated in Figure 2.5.2(a). These were started after the primary series were completed to further investigate the most interesting results, and as a way of estimating the accuracy of the simulations. One of secondary series was performed using a source with lower lower frequency than the primary, one was performed with higher spatial resolution, and one was performed with both low frequency and high resolution. To easily distinguish between simulation series with different frequencies and spatial resolutions, they will be denoted as low or normal frequency series, and normal or high resolution series. The exact differences between the different simulation series are explained in greater detail in section 4.5.

4.4 Simulation method

4.4.1 Transmission spectra

As mentioned in section 4.2.2, the simulations can give transmission coefficients for a broad frequency band. The source emits a short Gaussian shaped pulse with the spatial modulation from equation (4.3.2), and the polarization as described in section 4.3.3. The time dependence of a Gaussian pulse is illustrated in equation (4.4.1). This should not be confused with a Gaussian *beam* where the *spatial* distribution is Gaussian. The pulse propagates through the photonic crystal and reaches the recording layer indicated in Figure 4.3.1. The $\boldsymbol{E_t}$ and $\boldsymbol{H_t}$ fields in this layer are recorded and stored for further processing. The Fourier transform of the recorded fields, $\boldsymbol{E}(\omega)$ and $\boldsymbol{H}(\omega)$ can then easily be calculated using an fft-algorithm. The total transmitted power spectrum can then be calculated using equation (4.4.1). The ω-dependence which for most preceding derivations has been implicit, is here stated explicitly.

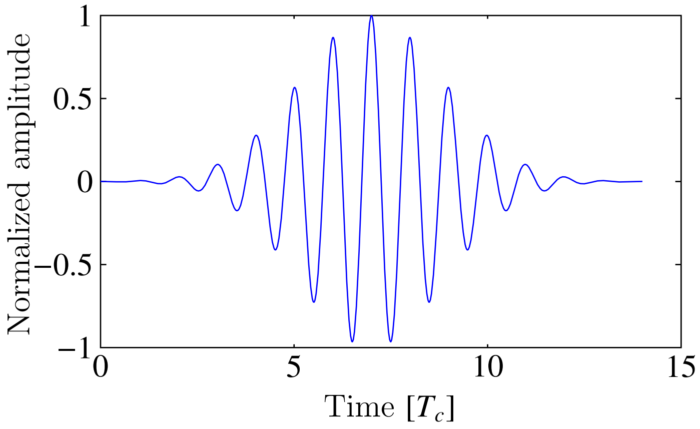


Figure 4.4.1: Real part of a Gaussian pulse.

$$P_{transCont}(\omega) = \int\limits_{\substack{Recording\\plane}} \frac{1}{2}\Re\left[\boldsymbol{S}(\omega)\right]\cdot\hat{\mathfrak{z}}\,\mathrm{d}A = \int\limits_{\substack{Recording\\plane}} \frac{1}{2}\Re\left[\boldsymbol{E}(\omega)\times\boldsymbol{H}^{*}(\omega)\right]\cdot\hat{\mathfrak{z}}\,\mathrm{d}A \tag{4.4.1}$$

$P_{transCont}(\omega)$	Transmitted power spectrum for a continuous space
$\hat{\mathfrak{z}}$	Unit vector in the z-direction
$\Re$	Real part operator

Because the simulations are conducted using discretized space, equation (4.4.1) must be approximated by a sum. Since the final result is supposed to be transmission coefficients, the scaling constants can be omitted. This also includes the constants normally introduced when approximating an integral by a sum. Equation (4.4.2) states the resulting expression for the simulated transmitted power spectrum.

$$P_{trans}(\omega) = \sum_{\substack{Recording\\plane}} \Re\left[\boldsymbol{E}(\omega)\times\boldsymbol{H}^{*}(\omega)\right]\cdot\hat{\mathfrak{z}} \tag{4.4.2}$$

$P_{trans}(\omega)$	Transmitted power spectrum for a discrete space

The transmission coefficient spectrum is the ratio of $P_{trans}(\omega)$ to $P_{inc}(\omega)$ which is the power spectrum of the wave moving upwards in the model from the source layer. $P_{inc}(\omega)$ can be found by performing a reference simulation with the same model, but without the scattering photonic crystal surface. This model must consist of a uniform medium which has the same refractive index as the medium surrounding the source layer. The impedance seen by the source should then be the same in both simulations and the emitted power spectrum in the upwards direction should be the same. The transmission spectrum $T(\omega, \boldsymbol{k_T})$ can therefore be found using equation (4.4.3).

$$T(\omega, \boldsymbol{k_T}) = \frac{P_{trans}(\omega)}{P_{inc}(\omega)} = \frac{\sum\limits_{\substack{Recording\\ plane\ in\\ model}} \Re\left[\boldsymbol{E}(\omega) \times \boldsymbol{H}^*(\omega)\right] \cdot \hat{\boldsymbol{z}}}{\sum\limits_{\substack{Recording\\ plane\ in\\ reference}} \Re\left[\boldsymbol{E}(\omega) \times \boldsymbol{H}^*(\omega)\right] \cdot \hat{\boldsymbol{z}}} \tag{4.4.3}$$

$\boldsymbol{k_T}$	Part of wave vector which is parallel to the surface, (xy-plane in Figure 4.1.1)
$T(\omega, \boldsymbol{k_T})$	Transmission spectrum for incident light with transversal wave vector $\boldsymbol{k_T}$
$P_{inc}(\omega)$	Power spectrum incident on the photonic crystal surface.

$T(\omega, \boldsymbol{k_T})$ in equation (4.4.3) measures the power transmission of a single incident plane wave with transversal wave vector $\boldsymbol{k_T}$, and angular frequency ω. It does not take into account how the transmitted field distribution is, only its total power.

4.4.2 Surface transmission efficiency

The procedure described in section 4.4.1 gives the transmission spectrum for one particular value of $\boldsymbol{k_T}$ of the incident wave. Repeating it for a range of $\boldsymbol{k_T}$-vectors, and for both polarizations will give a frequency specific lookup table with the same role as the Fresnel equations have for plane surfaces. For a given frequency ω_0, a 3D plot of the function $T(\omega_0, \boldsymbol{k_T})$ in $\boldsymbol{k_T}$-space will give insight into how the photonic crystal surface affects the transmission properties of the surface. For example it would be possible to see the transmission for incident angles which would undergo total internal reflection if the surface was plane. The key parameter describing the performance of the photonic crystal is the surface transmission efficiency η_s introduced in section 3.4. For a given frequency ω_0 this property is dependent on the transmission coefficient $T(\omega_0, \boldsymbol{k_T})$ and the angular radiation distribution of the light incident on the surface. This distribution was derived in chapter 3 and is plotted in Figures 3.3.12 to 3.3.15. It is important to note that this is the actual light hitting the surface in the physical LED, it is not the simulated $P_{trans}(\omega)$ in equation (4.4.2). Equation (3.4.5) states the definition of η_s when the incident radiation distribution is given by $I(\theta, \phi)$ in spherical coordinates. Since the simulated transmission spectrum is a function of $\boldsymbol{k_T}$ rather than θ and ϕ, it is more convenient to move to the $\boldsymbol{k_T}$-domain. Using the theory from section 3.3.2, this results in the expression in equation (4.4.4).

$$\eta_s(\omega) = \frac{\int\limits_{\mathrm{k_x^2+k_y^2<k^2}} T(\omega, k_x, k_y)\, I_k(k_x, k_y)\, \mathrm{d}k_x\, \mathrm{d}k_y}{\int\limits_{\mathrm{k_x^2+k_y^2<k^2}} I_k(k_x, k_y)\, \mathrm{d}k_x\, \mathrm{d}k_y} \tag{4.4.4}$$

η_s	Surface transmission efficiency
k_x	x-component of $\boldsymbol{k_T}$
k_y	y-component of $\boldsymbol{k_T}$
$I_k(k_x, k_y)$	Radiation distribution incident on the photonic crystal surface

$T(\omega, k_x, k_y)$ is known for a discrete set of k_x and k_y, and $I_k(k_x, k_y)$ can be calculated rapidly for any k_x and k_y. Because I_k has sharp peaks as shown in Figures 3.3.14 and 3.3.15, a sharp resolution in k_x,k_y-space is necessary. This is however not feasible for $T(\omega, k_x, k_y)$ because the simulations would become too time consuming. $T(\omega, k_x, k_y)$ is therefore linearly interpolated to provide approximate transmission coefficients for a high resolution version of $I_k(k_x, k_y)$. The integral is subsequently calculated using the trapezoidal method.

4.4.3 Radiation pattern

Purpose

Finding the surface transmission efficiency η_s, and the geometry that maximizes it, was the primary objective of the work this thesis is based on. Although the quantity is not directly measurable, the improvement from an unpatterned LED can be measured. In the interest of quality proofing the simulations it is however desirable to have a more revealing result that contains more information than this single number. A radiation pattern is suitable for such a task. That is a power density function $I_{k,ext}(k_x, k_y)$ that measures the radiation at different angles on the outside of the LED. This function can be converted to a power density function that measures power per unit solid angle. Such a function is directly measurable by placing a detector at a distance from the LED and rotating the LED to specific angles.

Principle

The transmission spectrum which was calculated in equation (4.4.3) is the ratio of the simulated transmitted power to the simulated incident power. To caclulate a radiation pattern it is necessary to find the propagation direction of the light on the outside of the LED for each simulation. That way it is possible to find the fraction of the incident light wave that is converted to the different refraction orders on the outside. This fraction would then be the coupling coefficient Γ in equation (4.1.11). This equation states the output power density with a certain $\boldsymbol{k_T}$ as a sum of the incident power density function at several distinct $\boldsymbol{k_T}$-points. Note that there is a difference between the simulated incident and transmitted waves, and the actual incident and transmitted waves. The ratios between the transmitted and incident waves are the same, and the simulations give this ratio which is applied to the real incident waves found in Chapter 3. Since equation (4.1.11) relies on the coupling from several incident plane waves, it is important to simulate all those needed. With reference to Figure 4.2.1, it therefore becomes important to perform simulations with a suitable set of $\boldsymbol{k_T}$-values for the incident plane wave. If a simulation is performed with an incident beam with $\boldsymbol{k_T} = \boldsymbol{k_{T}}_0$, then there must also be simulations for $\boldsymbol{k_T} = \boldsymbol{k_{T}}_0 + m_1\boldsymbol{b_1} + m_2\boldsymbol{b_2}$ for all integers m_1 and m_2 that give a $\boldsymbol{k_T}$-vector which is shorter than the wave number in the semiconductor. The first task is then to find the propagation direction of the transmitted light for each simulation.

Propagation direction

When a plane wave is transmitted through the surface, it couples to a number of plane waves due to diffraction. Finding the power in each of these plane waves can be done by calculating the spatial Fourier transform of the fields. Since the simulations effectively simulate an infinite surface, it should be possible to get an arbitrarily high resolution in such a Fourier transform. The recorded fields that are Fourier transformed along the time axis are only valid for one unit cell. However since the phase difference between neighbouring cells is known through the $\boldsymbol{k_B}$-vector, the fields for an arbitrarily high number of unit cells can be found. A consecutive spatial Fourier transform could then give the radiation as a function of $\boldsymbol{k_T}$ with an arbitrarily high resolution. Another, less computationally demanding, method could however also be used. If a function of time is strictly periodic with the period T its Fourier transform has all its energy concentrated at 0, $1/T$, $1/(2T)$ and so on. The power at these points can then be found as the absolute square of the corresponding terms in the function's Fourier series. When finding such a Fourier series, it is only necessary to consider one period of the function. The recorded fields are in general not periodic functions, and can be expressed with the form of equation (4.4.5). Note that this is not the same as the mode field in equation (2.5.10) where $\boldsymbol{u}$ and $\boldsymbol{H}$ are supposed to form a single mode of the structure. Here $\boldsymbol{E}$ and $\boldsymbol{v}$ are in general composed of several modes, but all having the same $\boldsymbol{k_B}$-vector.

$$\boldsymbol{E} = \boldsymbol{v}(\boldsymbol{r}) \cdot \mathrm{e}^{j\boldsymbol{k_B}\cdot\boldsymbol{r}} \tag{4.4.5}$$

$\boldsymbol{k}_B$	Vector constant within the plane of the crystal and within the Brillouin zone.
$\boldsymbol{v}(\boldsymbol{r})$	Function that is periodic in the plane of the crystal. Arbitrary in the z-direction.

$\boldsymbol{E}$ in equation (4.4.5) consists of two periodic functions, and is therefore periodic only if the ratio of the two functions' frequency is a rational number. Although $\boldsymbol{k_B}$ in the simulations could be chosen such that this was true, it is not something which will be assumed here. It is however possible to calculate the spatial Fourier transform of the two periodic functions individually and combine them afterwards by convolution as stated in equation (4.4.6).

$$\boldsymbol{E_k}(\boldsymbol{k_t}) = \mathcal{F}^2\left\{\boldsymbol{E}(\boldsymbol{r})\right\}(\boldsymbol{k_T}) = \frac{1}{4\pi^2}\mathcal{F}^2\left\{\boldsymbol{v}(\boldsymbol{r})\right\}(\boldsymbol{k_T}) * \mathcal{F}^2\left\{\mathrm{e}^{j\boldsymbol{k_B}\cdot\boldsymbol{r}}\right\}(\boldsymbol{k_T}) \tag{4.4.6}$$

$E_k(\boldsymbol{k_t})$	2D spatially Fourier transform of the electric field
$\mathcal{F}^2$	2D Fourier transform operator
$*$	Convolution operator

The Fourier transform of the complex exponential function is stated in equation (4.4.7). The easiest way of verifying this is to take the inverse Fourier transform of the result.

$$\mathcal{F}^2\left\{\mathrm{e}^{j\boldsymbol{k_B}\cdot\boldsymbol{r}}\right\}(\boldsymbol{k_T}) = 2\pi\delta(k_{Tx} - k_{Bx}) \cdot 2\pi\delta(k_{Ty} - k_{By}) = 4\pi^2\delta(\boldsymbol{k_T} - \boldsymbol{k_B}) \tag{4.4.7}$$

$\delta(a_1 - a_2)$	Dirac's delta function

Convolution with Dirac's delta function causes a pure shift of the argument. The spatially Fourier transformed electric field is therefore the Fourier transform of the $\boldsymbol{v}(\boldsymbol{r})$-function shifted by the vector $\boldsymbol{k_B}$. This is here stated in equation (4.4.8).

$$\boldsymbol{E_k}(\boldsymbol{k_T}) = \mathcal{F}^2\left\{\boldsymbol{v}(\boldsymbol{r})\right\}(\boldsymbol{k_T} - \boldsymbol{k_B}) \tag{4.4.8}$$

Assuming that the photonic crystal has a rectangular unit cell, equation (4.4.8) is the Fourier transform of a purely periodic function in both the x and y-direction. Since the simulation software used only supports rectangular unit cells, this will be valid for all photonic crystals discussed here. The non-zero values of the Fourier transform can therefore be calculated as terms in the function's 2D Fourier series. The propagation direction or $\boldsymbol{k_T}$-vector corresponding to the different

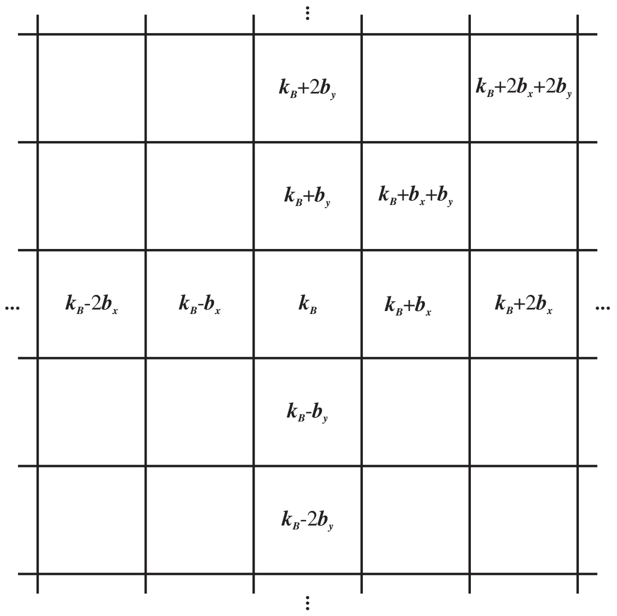


Figure 4.4.2: Diagram of $\boldsymbol{k_T}$-vectors associated with the terms in the 2D Fourier series of the electric field

terms will however be shifted by $\boldsymbol{k_B}$. For example, the center-point that normally is associated with an upwards propagating wave with $\boldsymbol{k_T} = 0$ will be the amplitude of the wave propagating with $\boldsymbol{k_T} = \boldsymbol{k_B}$. Figure 4.4.2 illustrates some of the $\boldsymbol{k_T}$-values associated with other points in the diagram of the Fourier series terms.

Since the simulations produce the electric field, and not the $\boldsymbol{\nu}$-function, the method for calculating $E_k(\boldsymbol{k_T})$ would firstly involve dividing the field with $e^{j\boldsymbol{k_B}\cdot\boldsymbol{r}}$ to effectively remove the $\boldsymbol{k_B}$-dependence. Then the discrete Fourier transform (DFT) [19] of exactly one unit cell could be calculated using an fft-algorithm. The result of this would then give the terms in the Fourier series. In contrast to the time continuous Fourier series, the DFT would provide a radiation distribution only for limited values of k_x and k_y. Typically the maximum k_x-value would be $k_{Bx} + N_x/2 * b_{xx}$. Here N_x is the number of voxels along the x-direction, in other words the width of the unit cell measured in voxels. This should not cause any problems as long as the maximum k_x is higher than the wavenumber for the light in air. The radiation at such k_x-points must be zero anyway since the field would be evanescent in the z-direction. The requirement is therefore that the spatial resolution must be higher than a minimum value. This requirement is very much like the sampling theorem [19] used in signal theory. Other concerns require the spatial resolution to be significantly higher anyway, meaning this would not become a problem. In fact storing these spatially Fourier transformed fields could remove a lot of redundant information and thereby save storage capacity. This because the points for the Fourier transformed version of the fields that lie within the wave number usually are within a 3×3 matrix while the original non transformed fields usually fill a 30×30 matrix. Disregarding the points outside the wave number limit could therefore reduce the needed storage space by approximately 100 times.

Coupling coefficients

Equation (4.4.8) gives the spatially Fourier transformed electric field in a certain propagation direction $\boldsymbol{k_T}$ as a result of one single incident plane wave with transversal wave vector which here will be labeled $\boldsymbol{k_{Ts}}$. To further distinguish between the wave vectors in air and the bulk semiconductor, the wave vector in air will here be labeled $\boldsymbol{k_{Ta}}$. This is in conjunction with figure 4.1.2. The simulated quantities will typically have two arguments that are transversal wave-vectors. The first represents $\boldsymbol{k_T}$ of the resulting wave in air, while the second one is $\boldsymbol{k_T}$ of the incident wave in the semiconductor. For example, the spatially Fourier transformed electric field will be denoted $\boldsymbol{E_k}(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}})$. Note that the difference $\boldsymbol{k_{Ta}}-\boldsymbol{k_{Ts}}$ must be a valid reciprocal lattice vector. Although the same quantities are frequency dependent, this will not be stated explicitly to simplify the notation. A similar equation to equation (4.4.8) can also be found for the magnetic field, and the power flow at different $\boldsymbol{k_{Ta}}$ can then be found with equation (4.4.9).

$$\boldsymbol{S_k}(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}}) = \Re\left[\boldsymbol{E_k}(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}})\times\boldsymbol{H_k}^*(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}})\right]\cdot\hat{\boldsymbol{z}} \tag{4.4.9}$$

$\hat{\boldsymbol{z}}$ Unit vector in the z-direction

Here equation (4.4.9) gives the simulated power transmitted through the surface at different $\boldsymbol{k_{Ta}}$ as a result of a single incident plane wave with wave vector $\boldsymbol{k_{Ts}}$. Dividing this result by the total power of the incident wave i.e. the total power in the reference simulation could subsequently give the coupling coefficients Γ used in equation (4.1.11). Since the spatially Fourier transformed fields were calculated using discrete Fourier transform [19], it is also necessary to divide by the number of pixels in the cross section of the model. Equation (4.4.10) states this mathematically.

$$\Gamma(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}}) = \frac{\boldsymbol{S_k}(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}})}{N_x\,N_y\,P_{inc}(\boldsymbol{k_{Ts}})} \tag{4.4.10}$$

$\Gamma(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}})$	Coupling coefficient from $\boldsymbol{k_{Ts}}$ to $\boldsymbol{k_{Ta}}$.
N_x	Number of pixels along the x-axis of the model.
N_y	Number of pixels along the y-axis of the model.
$P_{inc}(\boldsymbol{k_{Ts}}\ )$	Simulated incident power on the surface (reference simulation).

The coupling coefficient Γ is stated somewhat differently in equation (4.4.10) compared to equation (4.1.11). The integers m_1 and m_2 in equation (4.1.11) represent the $\boldsymbol{k_T}$ of the source plane wave in the semiconductor, while in equation (4.4.10) this is stated explicitly through $\boldsymbol{k_{Ts}}$. The final radiation pattern can subsequently be constructed by applying the coupling coefficients to the internal radiation distribution $I_k(\boldsymbol{k_T})$ found in section 3.3.6, and summing all the terms that couple to identical plane waves in air. Since the two possible polarizations for the incident light are simulated independently, the contribution from both must also be combined as the incident field is assumed to be unpolarized. Equation (4.4.11) states this and gives the complete radiation pattern in air. The sums are over the points in $\boldsymbol{k_{Ts}}$ where the vector difference $\boldsymbol{k_{Ts}} - \boldsymbol{k_{Ta}}$ is a valid reciprocal lattice vector of the photonic crystal.

$$I_{ka}(\boldsymbol{k_{Ta}}) = \frac{1}{2}\left[\sum_{\boldsymbol{k_{Ts}}} \Gamma_{TE}(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}})\, I_{ks}(\boldsymbol{k_{Ts}}) + \sum_{\boldsymbol{k_{Ts}}} \Gamma_{TM}(\boldsymbol{k_{Ta}};\boldsymbol{k_{Ts}})\, I_{ks}(\boldsymbol{k_{Ts}})\right] \tag{4.4.11}$$

$I_{ka}(\boldsymbol{k_{Ta}})$	External radiation pattern.
$I_{ks}(\boldsymbol{k_{Ts}})$	Internal radiation pattern found in section 3.3.6.

4.4.4 Resolution

The resolution of the external radiation pattern is the number of valid points in the $I_{ka}(\boldsymbol{k_{Ta}})$-function which are within the light cone, meaning that $|\boldsymbol{k_{Ta}}| < 2\pi/\lambda$. If the spacing between points in the x and y direction is Δk_x and Δk_y, respectively, equation 4.4.12 states this resolution.

$$Res_{ext} = \frac{4\pi^3 \Delta k_x \Delta k_y}{\lambda^2} \tag{4.4.12}$$

Res_{ext}	Resolution of the external radiation pattern.
Δk_x	Spacing between points in the x-direction.
Δk_y	Spacing between points in the y-direction.
λ	Wavelength in vacuum.

The spacing between the points is preserved through the surface, but the light cone is substantially widened because of the high refractive index of the semiconductor. This means that to achieve a particular Δk_x and Δk_y, the number of points in $\boldsymbol{k_{Ts}}$ for which a simulation must be performed, is significantly higher than the resolution obtained in the external radiation pattern. This number, $\boldsymbol{Res_{int}}$, is given by equation (4.4.13). The ratio between the two resolutions is n^2 where $n = 3.8$, meaning the obtained resolution of the external radiation pattern would be 14 times less than the number of simulated points in $\boldsymbol{k_{Ts}}$.

$$Res_{int} = \frac{4\pi^3 n^2 \Delta k_x \Delta k_y}{\lambda^2} \tag{4.4.13}$$

Res_{int}	Number of points in $\boldsymbol{k_{Ts}}$ for which a simulation must be performed.
n	Refractive index of the semiconductor (3.8).

4.5 Simulation parameters

4.5.1 Units

The simulation software, MEEP, which has been used here uses its own units for length and time. They are defined in such a way that the physical constants c, ε_0 and μ_0 are all unity. To relate the simulation to the real world it is therefore necessary to utilize the electromagnetic scaling properties discussed in section 2.4. Equation (2.4.7) states the relation between the length ratio and the wavelength ratio between two similar structures. Rearranging the equation can give the relations in equations (4.5.1) and (4.5.2) which states the physical length of a structure relative to the physical wavelength and in absolute terms, respectively. Both these equations are based on a light velocity of 1 in the simulations. The index s means the symbol denotes a quantity of the simulated structure, and the index r means it is a real physical quantity.

$$\frac{l_r}{\lambda_r} = l_s \cdot \nu_s \tag{4.5.1}$$

$$l_r = l_s \frac{c\, h\, \nu_s}{E_p} \tag{4.5.2}$$

l_s	Length in simulation.
ν_s	Optical frequency in simulation.
l_r	Physical length.
λ_r	Physical vacuum wavelength.
E_p	Physical photon energy.
c	Physical speed of light in vacuum.
h	Planck's constant.

As can be seen in figure 4.3.1, the width of one unit cell is 1. This means that the lattice constant also is 1, something which is true for all the simulation series presented here. Looking at equation (4.5.1), the real lattice constant, measured in wavelengths, has the same numerical value as the simulated optical frequency. For normal frequency simulation series the range of simulated optical frequencies was between $\nu_s = 0.4$ to $\nu_s = 1.3$, and for two low frequency series, it was between

$\nu_s = 0.2$ to $\nu_s = 0.6$. The lattice constant of the corresponding real physical structure, measured in wavelengths, would have the same value, and the absolute values for the two photon energies 0.65eV and 0.7eV are listed in Table 4.5.1.

Table 4.5.1: Simulated physical lattice constants (a)

Photon energy [eV]	Vacuum wavelength [μm]	a @ $\nu_s = 0.2$ [μm]	a @ $\nu_s = 0.4$ [μm]	a @ $\nu_s = 1.3$ [μm]
0.65	1.91	0.38	0.76	2.5
0.70	1.77	0.35	0.71	2.3

Another peculiar concept regarding the units is that the Bloch k-vector $\boldsymbol{k_B}$ is related to wavelength as $1/\lambda$ instead of the usual $2\pi/\lambda$. To ensure consistency, all kinds of k-vectors in the post processing software also follow this rule. These kinds are in addition to Bloch k-vectors, the plane wave k-vectors and reciprocal lattice vectors. Since rounding errors can be avoided because the irrational π-factor is removed, this actually makes the parameter passing slightly easier. As a consequence of this, equation (4.4.12) would lose a $(2\pi)^2$-factor if the k-vector were denoted in this way. The rest of the equations remain unchanged as they typically consist of fractions of the form $\boldsymbol{k}/k_0$. Changing the denomination of both the k-vector and the wave number does not affect the ratio between them.

4.5.2 Variable parameters

The actual simulation where one plane wave is excited and the transmitted field is recorded, is typically characterized by a set of parameters. The variable parameters are those which are varied within a simulation series. The simulations presented here had 4 such parameters which had their values scanned to cover all possible combinations. All these scans can be characterized by a certain scanning range and a resolution which together determine the number of simulations to run. Since all possible combinations of these parameters were simulated their resolution ranges had a vital influence of the total running time of the simulations. 2 of these parameters were the 2 components of the $\boldsymbol{k_T}$-vector of the source ($\boldsymbol{k_{Ts}}$) which were scanned in order to cover all angles of incidence. The scanning range was limited by the maximum wave number in the semiconductor ($nk_{0,max}$) which with the applied units is equal to $n\nu_{max}$. With the refractive index of GaSb at 3.8, and a maximum optical frequency of 1.3, the maximum wavenumber became 4.94. For the low frequency simulation series where ν_{max} was 0.6, the maximum wave number was instead 2.28. As mentioned in section 4.2.1 it was not necessary to simulate the entire domain limited by $nk_{0,max}$ since the unit cells possess certain symmetry properties. The square gratings had a horizontal, vertical and a diagonal line of mirror symmetry which reduced the scanning range as illustrated in Figure 4.2.1. The hexagonal grating lacked the diagonal line meaning the entire quadrant where $k_x > 0$ and $k_y > 0$ must be simulated. The chosen resolution for k_x and k_y was 0.1, and 0.05 for the low frequency simulations. The other two variable parameters were the depth and radius of the grating holes, and are defined in Figure 4.3.3. The upper bound for the depth is determined by the thickness of the top layer of intrinsic GaSb of the LED shown in Figure 3.2.1. This is 1μm thick, and the hole depth should be limited to slightly less than this. At the simulation frequency $\nu_s = 1$, 1μm real length corresponds to 0.56 in simulation units, and an upper bound was therefore set at approximately 0.55. This will mean somewhat deeper than 1μm for the highest simulation frequencies and somewhat shallower for the lower frequencies. For the low frequency simulations, the limit was extended to 1.35. The upper bounds for the radius of the holes were when neighbouring holes overlapped completely, effectively creating a lowered plane surface. The holes were however allowed to partially overlap effectively creating a grating of pillars instead of holes. Keeping in mind that the grating period was unity, this limit is $1/\sqrt{2}$ for the square gratings , and $1/\sqrt{3}$ for the hexagonal grating. The ranges and resolutions of all these four variables are listed in Table 4.5.2. They are given for six types of simulations where three of them are for the three different grating types presented in section 4.3.4, and the other three are more detailed simulations of the square grating of holes.

4.5.3 Constant parameters

The simulations are also characterized by a set of constant parameters. These are identical for the individual simulations within a simulation series, and typically describe the physical structure being simulated. The size of the simulation cell, the spatial resolution, the distances between all the layers shown in Figure 4.3.1, and the thicknesses and the strength of the two PML layers are such constant parameters. Also these parameters affect both the total running time and the accuracy of the results. Large structures are typically more realistic, but lead to a higher number of voxels, and consequently longer running time for the simulations. The next sections will present different groups of constant parameters which are used in the simulations. The variables are also listed in Table 4.5.3 where the given values are valid for the square gratings with optical frequencies ranging from 0.6 to 1.3.

Layer spacings

The distances between the different layers in Figure 4.3.1 are as mentioned constant parameters. It is typically best with large spacing between each layer in Figure 4.3.1 and with thick PML. This reduces any influence the surface and bottom PML might have on the source, and possible evanescent fields from the surface are allowed to decay properly before reaching the recording layer and the top PML. Two sets of values were used for these parameters, one for each of the two frequency ranges. The spacings in the low frequency simulations were twice the ones in the normal frequency simulations. This because evanescent decay rates were assumed scale approximately inversely proportional to the wavelength. The spacings shown in Figure 4.3.1 were used for the simulations with highest frequency. These are also stated in Table 4.5.3, and can be recognized as the first letter in the parameters' names is a lower case "d".

Simulation time

The simulation time, that is the number of time units, for each simulation was controlled by two parameters. One of these controlled the Gaussian source which was used to excite the fields. The time response is supposed to be Gaussian which implies it never actually reaches zero, and that it has been non zero for an infinitely long time before it reaches its peak value. This is impossible to achieve, and the source must instead be abruptly started finitely before the peak and abruptly stopped finitely after the peak. The time separating the peak and both turn on and cut off is defined by a parameter the parameter "sourceCutoff". A long waiting time would naturally increase the simulation time while a short one could introduce errors by exciting unexpected high frequency components. The simulation must also at one time be stopped. This time is for all the simulations set to 10 times the e^{-2}-width of the Gaussian pulse. The second parameter concerning the simulation time is the stop threshold. In this case it is a measure of how much the Poynting vector in the z-direction in the center of the recording plane must decay from its maximum value before the simulation is stopped. A too low value here would alter the frequency content of the recorded field, while a very large value would cause a needlessly long simulation time. The actual value was set to $1 \cdot 10^{-4}$ for most of the simulations, and $1 \cdot 10^{-6}$ for the low frequency simulations. Both of these time related parameters are stated in Table 4.5.3 as "sourceCutoff" and "stopTh" where the latter is only valid for simulations with frequency range $\nu_s = 0.6$ to $\nu_s 1.3$.

Spatial resolution

The spatial resolution is the most critical parameter with regards to the total running time of the simulations. It specifies a three dimensional resolution, meaning the amount of voxels is cubically dependent on this parameter. In addition the resolution in time is linearly dependent on the resolution. This concept is explained in more detail in section 4.5.3. This means that the running time of the simulations are biquadratically dependent on the resolution. Choosing just high enough resolution is therefore essential if the simulations is to be completed within reasonable time. The developers of the simulation software have recommended a rule of thumb stating that even crude simulations should have eight voxels per wavelength in the region of highest refractive index. For the refractive index of GaSb which is 3.8, and an optical frequency of 1, this translates to a resolution of “30.4”. The chosen resolution was 30 which means that the accuracy for the frequencies above 1 can not be considered to be very good. This was however necessary for the simulations to complete within reasonable time. The discretized Maxwell equations are however linear, meaning that the power at different frequencies should not mix. This ensures that imperfections for high frequencies do not affect the lower frequencies. Some simulations were alse performed with a resolution of 60 as a way of estimating the accuracy of the simulations with lower resolution. As can be seen from Table 4.5.2, these were only performed for very few gratings. The ones that looked most interesting after the regular simulations were selected for the high resolution simulations. The resolution in the regular low frequency simulation was conducted with a resolution of only 20, while the low frequency high resolution simulation had a resolution of 60. The resolution for the simulations with neither high resolution or low frequency is stated in Table 4.5.3.

Sampling frequency

The fields are sampled in the recording plane as the simulation progresses. It is important that this frequency is high enough, but too high a value would result in needless disk writing and thereby increasing the simulation time. Also the post processing would require more time as the resulting dataset would be higher. The minimum value for this frequency is limited by the folding in frequency domain which occurs when either sampling a continuous signal or down-sampling a discrete signal [19]. Sampling of real signals must therefore be performed at a frequency which is twice the highest frequency component in the signal. For complex signals where the prefix of the frequency is known, this can be relaxed to a sampling frequency that is simply higher than the highest frequency component in the signal. This is because the spectrum of such signals do not have inversion symmetry around the origin. The size of the simulation time steps determines the highest possible sampling frequency, and to get maximum accuracy the applied sampling frequency must be such that the time between consecutive samples is a multiple of the simulation step size. In fact the simulation software rounds off the specified sampling frequency to achieve this. By default the simulation software sets the time steps to half the numerical value of the distance between adjacent voxels. This is according to the developers of the simulation software, very near the limit of numerical stability. The similarities in numerical values come from the fact that the speed of light in the simulations is unity. The highest frequency utilized in the simulations is 1.3, however since the spectrum falls off exponentially there is also power at higher frequencies. A sampling frequency of 5 was therefore chosen as it leaves a significant margin down to the 1.3-limit, and it is also a valid sampling frequency for all the spatial resolutions which have been used in these simulations. The same sampling frequency has been used for all the simulations as this simplified the post processing. The value is stated in table 4.5.3 where it has the symbol “Fs”.

Since the total running time was a highly limiting factor for these simulations, it was essential that all constant parameters were just at the limit where they started to affect the results. In order to approach this limit, a series of 2D simulations were conducted where these parameters were swept. The unit cell was like the cross section of the 3D cell, and effectively formed a surface with grooves instead of holes. The chosen values for these parameters are stated in Table 4.5.3.

Subpixel smoothing

Subpixel smoothing, or subpixel averaging, is a feature of the simulation software which make voxels on the boundary between two materials take a weighted average value. As mentioned in section 4.2.2 Farjadpour [7] shows how this improves the accuracy of the simulations. The reason it is mentioned here is that not all simulation series had this feature enabled. For most of them there was not a downside to having it enabled, but for the square grating of isotropically etched holes it significantly increased the running time. This feature was therefore turned off for that specific simulation series. The reason for this increase in running time is probably that the this grating had to be defined using a programming function that returned a certain ε as a function of position. The other gratings could instead be defined using predefined geometric shapes. It is reasonable to believe that the latter increases the efficiency of the calculation of this average ε.

Table 4.5.2: Variable parameters

Square grating of holes

	k_x	k_y	d	R
Minimum	0	0	0.05	0.05
Resolution	0.1	0.1	0.05	0.05
Maximum	5.9	$\min\left(k_x, \sqrt{k_{max}^2 - k_x^2}\right)$	0.55	0.7

Square grating of isotropically etched holes

	k_x	k_y	d	R
Minimum	0	0	0.05	0.05
Resolution	0.1	0.1	0.08	0.05
Maximum	5.9	$\min\left(k_x, \sqrt{k_{max}^2 - k_x^2}\right)$	0.53	0.65

Hexagonal grating of holes

	k_x	k_y	d	R
Minimum	0	0	0.05	0.05
Resolution	0.1	0.0980	0.08	0.05
Maximum	5.9	$\sqrt{k_{max}^2 - k_x^2}$	0.53	0.55

Square grating of holes (Low frequency)

	k_x	k_y	d	R
Minimum	0	0	0.15	0.1
Resolution	0.05	0.05	0.15	0.05
Maximum	2.25	$\min\left(k_x, \sqrt{k_{max}^2 - k_x^2}\right)$	1.35	0.7

Square grating of holes (High resolution)

	k_x	k_y	d	R
Minimum	0	0	0.35	0.45
Resolution	0.1	0.1	-	0.05
Maximum	5.9	$\min\left(k_x, \sqrt{k_{max}^2 - k_x^2}\right)$	0.35	0.55

Square grating of holes (High resolution and low frequency)

	k_x	k_y	d	R
Minimum	0	0	0.9	0.5
Resolution	0.05	0.05	-	0.05
Maximum	2.25	$\min\left(k_x, \sqrt{k_{max}^2 - k_x^2}\right)$	0.9	0.6

Table 4.5.3: Constant parameters for the square gratings with $\nu_s \in [0.6, 1.3]$

Parameter	Description	Value
dPMLTop	Thickness of top PML	0.4
dTransPML	Distance between top PML and recording layer	0.2
dSurfTrans	Distance between recording layer and semiconductor surface	0.6
dSourceSurf	Distance between semiconductor surface and source layer	0.9
dReflSource	Distance between source layer and recording layer for reflected wave (this recording layer was not used)	0.1
dPMLRefl	Distance between recording layer for reflected wave and bottom PML	0.05
dPMLBot	Thickness of bottom PML	0.2
sourceCutoff	Waiting time before the Gaussian source is stopped/started (measured in multiples of the Gaussian e^{-2}-width)	10
stopTh	Fraction of maximum value before simulation is terminated	$1 \cdot 10^{-4}$
TPMLSt	Top PML strength (related to exponential decaying coefficient)	6
BPMLSt	Bottom PML strength (related to exponential decaying coefficient)	1
Dx	Cell size in the x-direction	1
Dy	Cell size in the y-direction	1
Fs	Sampling frequency	5
resolution	Spatial resolution	30
eps-averaging?	Subpixel smoothing	On

4.6 Semi analytic reference

4.6.1 Purpose

When calculating both the surface transmission coefficient and the radiation pattern, the simulations need to be compared to a reference simulation. That is a simulation without a semiconductor-air interface. The temporal Fourier transform of the recorded fields are used to find the power that is emitted upwards by the source in the semiconductor. The usable result from these reference simulations is therefore a power spectrum that calibrates the source. In general this spectrum can be expected to depend on the size of the simulation cell, the resolution and the $\boldsymbol{k_T}$-vector of the source, $\boldsymbol{k_{Ts}}$. For most of the simulation series, it is therefore necessary with a dedicated set of reference power spectra that covers all the simulated $\boldsymbol{k_{Ts}}$-vectors. Even though the reference simulations seem to be very easy with just a homogeneous material as seen in Figure 4.3.1, they are in fact much more time consuming than the actual simulations. The wave must pass through much more high index material with low velocity, and there is not any structure that can disturb the wave to make it move more upwards. The latter especially apply to waves with a long $\boldsymbol{k_{Ts}}$-vector, and thereby a very oblique angle of propagation. Since the different frequencies in the pulse have equal $\boldsymbol{k_{Ts}}$-vector, they must have different angle of propagation and also different velocity in the z-direction. This will cause the pulse to be very chirped when arriving at the recording plane. This can in turn lead to that the lowest frequencies with lowest velocity to a greater extent than high frequencies, arrive after the simulation has stopped causing errors in the power spectrum. To reduce these problems, attempts towards an analytical model of the reference was made, and is presented in the following section.

4.6.2 Dependence on the transversal wave vector

The following section will concentrate on the power spectrum's dependence of the transversal wave vector. The analysis will be carried out with a 2D model in the xz-plane. It is assumed that it can be extended to 3D by replacing the k_x component in the 2D case with the length of the $\boldsymbol{k_T}$-vector. Since the polarization direction is relative to the propagation direction as a TE or TM wave, this can be shown to be true by rotating the coordinate system around the z-axis. The objective is to find the emitted power as a function of $\boldsymbol{k_{Ts}}$ and the optical frequency while assuming the current densities are independent of $\boldsymbol{k_{Ts}}$. The procedure will be shown for TE polarized light, while only the result will be presented for TM polarized light. The model for TE polarized light is illustrated in Fiugre 4.6.1 and shows a planar monochromatic source which radiates equally in both the positive and negative z-direction in an infinite homogeneous dielectric medium.

Since k_x is pinned for both the upwards and downwards propagation, k_z must be equal with opposite sign, meaning $\theta_+ = \theta_-$.

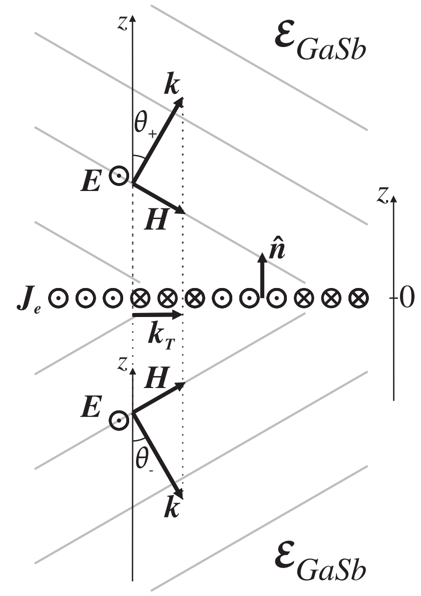


Figure 4.6.1: Monochromatic TE-wave excited by planar current source

In the plane of the source, the electric current density is given, in the units of the simulation software, by equation (4.6.1). The electric and magnetic fields are assumed to be plane waves with k_x identical to the source as stated in equations 4.6.2 and 4.6.3.

$$\boldsymbol{J_e} = \boldsymbol{J_{e0}} \mathrm{e}^{j2\pi k_x x} \tag{4.6.1}$$

$\boldsymbol{J_e}$ Electric current density.
$\boldsymbol{J_{e0}}$ Complex amplitude of electric current density (Only in the y-direction).

$$\boldsymbol{E} = \begin{cases} \boldsymbol{E_0}\mathrm{e}^{j2\pi(k_x x + k_z z)} & \text{if } z \geq 0 \\ \boldsymbol{E_0}\mathrm{e}^{j2\pi(k_x x - k_z z)} & \text{if } z < 0 \end{cases} \tag{4.6.2}$$

$$\boldsymbol{H} = \begin{cases} \boldsymbol{H_0}\mathrm{e}^{j2\pi(k_x x + k_z z)} & \text{if } z \geq 0 \\ \boldsymbol{H_0}\mathrm{e}^{j2\pi(k_x x - k_z z)} & \text{if } z < 0 \end{cases} \tag{4.6.3}$$

$\boldsymbol{E_0}$ Complex vector amplitude of the electric field.
$\boldsymbol{H_0}$ Complex vector amplitude of the magnetic field.

The vector amplitudes of both the current density and the the electric field are assumed to only have y-components with respective amplitudes g and a as stated in equations (4.6.4) and (4.6.5). The objective is first to find the relation between the amplitudes a and g.

$$\boldsymbol{J_{e0}} = \hat{\boldsymbol{y}}g \tag{4.6.4}$$

$$\boldsymbol{E_0} = \hat{\boldsymbol{y}}a \tag{4.6.5}$$

$\hat{\boldsymbol{y}}$ Unit vector in the y-direction.

If a general plane wave with arbitrary polarization is inserted into the first of Maxwell's equations (2.2.11), then the relation in equation (4.6.6) can be derived. Note that right side has the factor ν instead of the usual ω. This is because $\boldsymbol{k}$ is denoted in the units of the simulation software.

$$\boldsymbol{k} \times \boldsymbol{E_0} = \nu\mu_0\boldsymbol{H_0} \tag{4.6.6}$$

ν Optical frequency

Equation (4.6.6) can be used to find the complex vector amplitude of the magnetic field, and the result is stated in equation (4.6.7). Here the relations in equations (4.6.8) to (4.6.10) have been used to replace the $\boldsymbol{k}$-vector components. Note that since $\boldsymbol{k}$ is different above and under the source plane, $\boldsymbol{H_0}$ will also be different.

$$\boldsymbol{H_0} = \begin{cases} \frac{a\eta_0}{n}\left[-\hat{\boldsymbol{x}}\cos(\theta) + \hat{\boldsymbol{z}}\sin(\theta)\right] & \text{if } z \geq 0 \\ \frac{a\eta_0}{n}\left[\hat{\boldsymbol{x}}\cos(\theta) + \hat{\boldsymbol{z}}\sin(\theta)\right] & \text{if } z < 0 \end{cases} \tag{4.6.7}$$

η_0 Vacuum wave impedance $\left(\sqrt{\frac{\mu_0}{\varepsilon_0}}\right)$.

$$k_z = k\cos(\theta) \tag{4.6.8}$$

$$k_x = k\sin(\theta) \tag{4.6.9}$$

$$\frac{k}{\nu\mu_0} = \frac{n\nu}{c} = \frac{n\nu\sqrt{\mu_0\varepsilon_0}}{\nu\mu_0} = n\sqrt{\frac{\varepsilon_0}{\mu_0}} = \frac{n}{\eta_0} \tag{4.6.10}$$

n Refractive index
k Wave number (= $|\boldsymbol{k}|$)
θ Propagation angle, equal to θ_+ and θ_- in Figure 4.6.1

The electromagnetic equivalence principle is presented by Balanis in [1] chapter 12. It is presented as a way of constructing imaginary sources based on the knowledge of the fields on a closed surface which encloses real sources. In its most basic form, it is actually the boundary conditions for the surface and is here stated in equations (4.6.11) and (4.6.12).

$$\boldsymbol{J_m} = -\boldsymbol{\hat{n}} \times (\boldsymbol{E_+} - \boldsymbol{E_-}) \tag{4.6.11}$$
$$\boldsymbol{J_e} = \boldsymbol{\hat{n}} \times (\boldsymbol{H_+} - \boldsymbol{H_-}) \tag{4.6.12}$$

$\boldsymbol{J_m}$	Magnetic current density
$\boldsymbol{J_e}$	Electric current density
$\boldsymbol{\hat{n}}$	Normal vector to the source surface (= $\boldsymbol{\hat{z}}$)
$\boldsymbol{E_+}$	Electric field just above the source surface
$\boldsymbol{E_-}$	Electric field just below the source surface
$\boldsymbol{H_+}$	Magnetic field just above the source surface
$\boldsymbol{H_-}$	Magnetic field just below the source surface

It can easily be verified that equation (4.6.11) is zero by inserting the electric field form equation (4.6.2). This is as expected since the source does not have a magnetic current density. Inserting the magnetic field into equation (4.6.12) does however give a result as $\boldsymbol{H_0}$ is not the same above and under the source plane. The result is a relation between the amplitudes $\boldsymbol{a}$ and g, and is stated in equation (4.6.13). The fact that the left and right sides of equation (4.6.12) could be matched also verifies the plane waves assumed in equations (4.6.2) and (4.6.3).

$$\boldsymbol{J_{e0}} = \boldsymbol{\hat{y}} g = -2\cos(\theta)\frac{a\eta_0}{n}\boldsymbol{\hat{y}} \tag{4.6.13}$$

Since $\boldsymbol{a}$ is known, it is now possible to find expressions for the fields expressed by the amplitude of the current density g. Equation (4.6.14) uses this to find the time average Poynting vector above the source plane.

$$\boldsymbol{\bar{S}} = \frac{1}{2}(\boldsymbol{E} \times \boldsymbol{H}^*) = \frac{n\,g^2}{8\eta_0\cos^2(\theta)}\left[\boldsymbol{\hat{x}}\sin(\theta) + \boldsymbol{\hat{z}}\cos(\theta)\right] \tag{4.6.14}$$

$\boldsymbol{\bar{S}}$	Time average Poynting vector

The power flow per unit area out of the source is the component of $\boldsymbol{\bar{S}}$ which is normal to the source surface. Multiplying this component by the area of the source gives the total power $\boldsymbol{P}$ emitted by the source at a certain angle. This is performed in equation (4.6.15).

$$P_s = A_s \bar{\mathbf{S}} \cdot \hat{\mathbf{z}} = \frac{n\, A_s\, g^2}{8\eta_0 \cos(\theta)} \tag{4.6.15}$$

A_s	Area of the source
P_s	Power emitted upwards by the source

In the simulations g is kept constant for all angles, and it is therefore useful to express the emitted power as a function of the power emitted at $\theta = 0$. Equation (4.6.16) states this relatively simple relation, and equation (4.6.17) states the same relation expressed with $\boldsymbol{k}$-vectors instead of the angle θ

$$P_s(\theta) = \frac{P_s(\theta = 0)}{\cos(\theta)} \tag{4.6.16}$$

$$P_s(\boldsymbol{k_T}) = \frac{P_s(\theta = 0)}{\sqrt{1 - \left(\frac{|\boldsymbol{k_T}|}{k}\right)^2}} \tag{4.6.17}$$

If the source instead was emitting TM polarized radiation, the relations in equations (4.6.16) and (4.6.17) would still apply. The derivation would only have been slightly different. For example equation (4.6.11) would have been used instead of equation (4.6.12) and η_0 would be in the nominator instead of the denominator as in equation (4.6.15). In the simulations this relation has been used to form the reference for the individual simulations. For each simulation series and polarization, the only necessary reference simulation was the one with $\theta = 0$. The other reference spectra were calculated using equation (4.6.16).

Chapter 5

Software

5.1 MEEP

5.1.1 Background

The simulation software which has been used to perform the simulations presented here is called MEEP, and was developed at MIT. The properties of the software are described in detail by Oskooi in [17]. This chapter will not explain the details of MEEP nor the details of how to use it. Only a brief overview will be given so that it is possible to understand the simulations that have been performed. As mentioned in section 4.2.2, MEEP uses the FDTD method to simulate electromagnetic wave propagation.

5.1.2 Usage

MEEP runs in a Linux command window where all the input and output to and from the program is performed through command line parameters, script files and output files. There isn't any graphical user interface to either construct the simulation model or to visualize the results. MEEP must therefore be set to output data to files which can be loaded by another tool which has visualizing properties. For all the work presented here MATLAB has been used both to visualize the results and to do post processing on outputs from the simulations. The typical way of starting MEEP is from the command line to invoke it with a *control file* as an argument. An arbitrary number of arguments of the form “parameter=value” can be added before the command file to specify certain parameters defined in the control file. An example of such an invocation is shown in Listing 5.1.

Listing 5.1: Invoking MEEP from a Linux command window

```
meep HoleDepth=0.35 CtrlFile.txt
```

The contents of the control file defines the conditions of the simulations. The number of dimensions, the geometry, the sources, the recording planes/lines/-points, the placement of the output files etc. The succeeding parameters define parameter values which are valid for that single execution. Typically the variable parameters are defined in this way, while the constant parameters are defined inside the control file. The control file is really a script that is executed when MEEP starts. The scripting language is called Scheme, and is described in its definition [13]. The next section will give a short introduction into the parts of this language which have been used in the work presented in this thesis.

5.1.3 Scheme

Variables

Variables are in Scheme defined and assigned through the `define`-statement. It should immediately be followed by the variable name and the value to assign to the newly created variable. All this, including the `define`-statement itself, must be encapsulated in parentheses. Later the variable can be reassigned to another value using the `set!`-statement in a similar manner. Listing 5.2 shows an example where a variable first is defined and assigned the number 5, and then reassigned the number -1.1.

Listing 5.2: Example of variable definition and reassigning

```
(define varName 5)
(set! varName -1.1)
```

As mentioned in section 5.1.2, variables can be set through the command line. To allow this the variable must be defined with the `define-param`-statement instead of the `define`-statement. The value assigned in the control file with the `define-param`-statement will then act as a default if the variable is not assigned through the Linux command line. Variables do not need to be numbers, but are in general objects of any type.

Calling functions

Functions are in Scheme called using the function name followed by a list of all input parameters. All of this must be encapsulated inside a pair of parentheses. Listing 5.3 gives such an example.

Listing 5.3: Calling a function with 3 input parameters

```
(FunctionName var1 var2 var3)
```

If a variable is to be set to the output value of a function, the `define` or `set!`-statement must be used with the content of Listing 5.3 in the place of the value. Listing 5.4 gives such an example.

Listing 5.4: Assigning the output value from `FunctionName` to the variable `varName`

```
(set! varName (FunctionName var1 var2 var3))
```

Operators

Operators such as + - / and * are available in Scheme as functions. This means that the function calling syntax must be used instead of the more usual <*operand*> <*operator*> <*operand*>-syntax. Predefined functions with the names "+", "-", "/", and "*" exists. These take any number of parameters, but two is the most common. As can bee seen in Listing 5.5, the syntax is not a big problem for simple calculations, but easily becomes difficult with slightly more complicated arithmetics.

Listing 5.5: Two examples of operator usage

```
;Calculating 2 + 3
(+ 2 3)

;Calculating (2 + 4*7.6)/2
(/ (+ 2 (* 4 7.6)) 2)
```

Outputs

The outputs from the simulations are handled by dedicated functions, but if outputs from the control file script is wanted, the Scheme output functions must be used. Two simple ways of doing this is to save text to a file or to print the text to the screen making it appear in the Linux command window. Printing to the screen can be performed using the `display`-function as shown in Listing 5.6.

Listing 5.6: Print to the screen

```
;Printing text
(display "Hello World")

;Printing the value of a variable
(display varName)
```

To save text to a file, it is necessary to first open the file using the `open-output-file`-function. The function outputs an object that must be used as a reference to that file. Writing text to the file can subsequently be done using the `display`-function with the file reference as the second argument. The `display`-function can then be called an arbitrary number of times to append more text to the file. Before the script ends the file must be closed by calling the function `close-output-port`. Listing 5.7 gives an example of this process.

Listing 5.7: Writing text to a file

```
;Opening (and creating) an output file
(define fileRef (open-output-file ”FileName.txt”))

;Writing text to the file
(display ”Start of file \n”, fileRef)
(display ”Hole raduis = ”, fileRef)
(display HoleRaduis, fileRef)

;Closing the file
(close-output-port fileRef)
```

Defining functions

New functions can also be defined in Scheme. As when defining new variables, the `define`-statment is used, but the name of the function and its arguments must be encapsulated inside a pair of parentheses. The body of the function then follows, and the last statements are regarded the output. Listing 5.8 states the syntax for defining functions, and Listing 5.9 shows an example of a function that calculates the square of its input, prints the result to the screen and finally returns the result.

Listing 5.8: Syntax for defining a function

```
(define (FunctionName arg1 arg2 arg3)
   (Line 1)
   (Line 2)
   (Line 3)
   (Line 4)
)
```

Listing 5.9: Function for calculating the square of a number

```
(define (calcSquare number)
   (define result (* number number))
   (display ”The Result is: ”)
   (display result)
   (result)
)
```

Objects

As mentioned earlier, not only numeric variables are allowed. Examples of strings have been shown, and many other types are also possible. In particular it is possible to define new types of variables that are not part of the Scheme language. Such types are generally called objects, and are defined by a *class* which specifies the object properties. These properties are usually a list of variables with a given name that are of a certain type. These variables can both be primitive types like numbers or they can be objects themselves. Class definition will not be discussed here as it was not used in the simulations, only object creation based on predefined classes will be mentioned. To create an object, the `make`-statement must be used followed by the class name and a list of properties and their corresponding values. The order of these property-value pairs can be arbitrary. Listing 5.10 states the general syntax for creating an object from a certain class.

Listing 5.10: Syntax for making an object

```
(make ClassName (PropertyName1 value1) (PropertyName2 value2))
```

If the object is to be used later in the program, it must be assigned to a variable. Listing 5.11 states the general syntax for this.

Listing 5.11: Syntax for making an object and assigning it to a variable

```
(define varName (make ClassName (PropertyName1 value1) (PropertyName2 value2)))
```

Listing 5.12 shows an example from the actual simulations. Here an object is created from the `dielectric`-class and assigned to the variable `GaSb`. The only parameter necessary is named `epsilon`, and its value is set to the square of the variable `refrIndex`. This variable must have been defined before the line in Listing 5.12.

Listing 5.12: Creating an object from the `dielectric`-class and assigning it to the variable `GaSb`

```
(define GaSb (make dielectric (epsilon (* refrIndex refrIndex))))
```

Lists

Lists or arrays of numbers or objects are very common in programming. In Scheme a list of any type of variable can be created using the `list`-function. Calling it with any number of input parameters returns a list containing all these parameters. Listing 5.13 shows an example where a list containing the number from 0 to 5 is assigned to the variable `myList`.

Listing 5.13: List of all numbers from 0 to 5

```
(define myList (list 0 1 2 3 4 5))
```

5.1.4 Configuring MEEP

MEEP is configured through a set of predefined variables. For example the spatial resolution is set by assigning the number to the predefined variable `resolution` using the `set!`-statement. Other variables are more complicated as they consist of more than a single number. Because the number of such variables is very high, not all will be explained. Only the ones used to define the geometry and the source will be presented here. These are the most complicated as they involve many parameters. In addition the functions for starting the simulations and saving the results will be presented. For the rest of the parameters the reader is referred to the control file for square gratings of holes in Appendix A and to the MEEP Reference [16].

Defining Geometry

The geometry illustrated in Figure 4.3.1 is defined through a single variable named `geometry`. This is a variable that holds a list of objects from the predefined `geometry-objects`-class. These objects are typically geometric shapes such as blocks, spheres, cylinders and cones. When constructing them it is always necessary with parameters defining the material they consist of and three coordinates that define their center position. Depending on the actual shape, it is also necessary with parameters defining their sizes and directions. For example the radius is the only necessary additional parameter for a sphere, while a cylinder must be specified by the radius, the height and the direction of its axis. Geometric shapes can also be specified by a user defined function. Such a function must take a point in three dimensions as input, and return the corresponding dielectric constant. The geometry in the simulations is set as in Listing 5.14.

Listing 5.14: Geometry in simulations

```
(set! geometry (if ref? (list allSemiconductor) (list semiConductorSlab hole)))
```

The `if`-function returns the second input argument if the first is `true`, or it returns the third if the first one is `false`. In the simulations `ref?` is used to determine if it is a reference simulation or not. If this is the case, the geometry will be defined by the object `allSemiconductor` which fills the entire cell with semiconductor material. If it is not a reference, then the two objects `semiConductorSlab` and `hole` determine the outcome, and will make the simulation model shown in Figure 4.3.1. The object `allSemiconductor` is a block placed in the center of the simulation cell and which is slightly bigger than it to make sure it is completely filled. Also `semiConductorSlab` is a block, but it is centered further down in the model and does not extend to the top. This only fills some of the model with semiconductor, and makes a plane interface between the semiconductor and air which is the default material. The holes making the photonic crystal are introduced by the object `hole` which is made of air and extends into the mentioned `semiConductorSlab`-object. Because `hole` is last in the list in Listing 5.14, it has priority over `semiConductorSlab`. Listing 5.15 shows the construction of all of these objects. The positions and sizes of the different objects have previously been defined based on the variables discussed in sections 4.5.2 and 4.5.3.

Listing 5.15: Definitions of geometric objects

```
(define semiConductorSlab
    (make block
        (center 0 0 (- cSemiConductor 1))
        (size 1.1 1.1 (+ dSemiConductor 2 ))
        (material (make dielectric (epsilon (* refrIndex refrIndex))) )
    )
)

(define hole
    (make cylinder
        (center 0 0 cDitch)
        (radius holeR)
        (height (* 2 holeDepth))
        (material air)
    )
)

(define allSemiconductor
    (make block
        (center 0 0 0)
        (size 1.1 1.1 (+ geomHeight 2 ))
        (material (make dielectric (epsilon (* refrIndex refrIndex))) )
    )
)
```

Defining the source

The source is set through the predefined variable `sources` which is a list of an arbitrary number of `source`-objects. The bottom line of Listing 5.16 sets this variable to the output of the function `defSource` which is the rest of Listing 5.16. The function returns a list with two elements. This is necessary because the source must be able to have an arbitrary polarization within its plane, and each source element can only be polarized in either the x, y or z direction. The two sources are therefore polarized in the y and x direction respectively, and their amplitudes `AmY` and `AmX` determine the effective polarization. The top of the `defSource`-function is concerned with setting these amplitudes such that the polarization fits the direction of the $\boldsymbol{k_{T_s}}$-vector. The last parameter used when making the `source`-objects has the name `amp-func`, and is set to a function called `sourcePhaseFunction`. This function is used to specify the source's spatial dependence. It must have a position in three dimensions as the input, and the output must be the amplitude in that point. This amplitude will be multiplied with the `AmX` or `AmY` to form a final amplitude. This is used to phase modulate the source as a function of position to implement the specified $\boldsymbol{k_{T_s}}$-vector. As indicated in Listing 5.16, there are several other parameters involved in the creation of a `source`-object as well. These will however not be explained in more detail here, and the reader is instead referred to the MEEP Reference [16].

Listing 5.16: Defining the source

```
(define (sourcePhaseFunction P)
    (make-polar 1 (+ (* 2 pi (vector3-x P) kxSource ) (* 2 pi (vector3-y P) kySource )) )
)

(define (defSource)
    (define AmX 0.0)
    (define AmY 1.0)
    (define ADenom (sqrt (+ (* kxSource kxSource) (* kySource kySource))) )
    (if (not (= ADenom 0.0))
        (begin
         (set! AmX  (/ (* -1.0 kySource) ADenom)  )
         (set! AmY  (/ (*  1.0 kxSource) ADenom)  )
        )
     )

    (list
        (make source
            (src (make gaussian-src (frequency Fcenter) (fwidth Fwidth) (cutoff sourceCutoff)))
            (component (if TE? Ey Hy))
            (center 0 0 cSource)
            (size 1 1 0)
            (amplitude AmY)
            (amp-func sourcePhaseFunction )
        )
        (make source
            (src (make gaussian-src (frequency Fcenter) (fwidth Fwidth) (cutoff sourceCutoff)))
            (component (if TE? Ex Hx))
            (center 0 0 cSource)
            (size 1 1 0)
            (amplitude AmX)
            (amp-func sourcePhaseFunction )
        )
    )
)

(set! sources (defSource))
```

Starting the simulation

When all settings have been configured the simulation can be started by running one of five available run-functions. They differ mainly in how the simulation is terminated. The run-function used in the simulations described in this thesis is called `run-sources+`, and the execution of it is shown in Listing 5.17. First it runs the simulation until all sources have shut down. Then it relies on the output of the function which is given as the first input parameter to `run-sources+`, namely the function `stop-when-fields-decayed`. For every timestep in the simulation this function is called to determine if the simulation is to be terminated. A specific field component at a given point is checked and compared to its maximum value. A return value signaling termination is given if the field component has decayed to a given fraction of its maximum value. More precisely, the maximum value within a given time block is compared to the maximum value since simulation startup. If it has not decayed enough, the simulation continues for another equally sized time block. Evaluating the maximum value within a time block instead of continuously evaluating it avoids termination of the simulation if the field is below the threshold only for a short time before increasing again. As shown in Listing 5.17, the function `stop-when-fields-decayed` takes four input parameters. These respectively set the time block length to 5 time units, specifies that the z-component of the Poyntings vector is the field to consider, defines the point at which to check the field, and specifies the fraction the field must have decayed. The latter is also mentioned in section 4.5.3.

Subsequent to the first input parameter to `run-sources+`, an arbitrary number of step functions can be entered. These are functions that will be called for every or some of the timesteps in the simulation. The ones that are used as input parameter to the `run-sources+`-function will be called for every time step. Depending on the actual step function used, it might also take an arbitrary number of step functions as input parameters. In such a way a chain of step functions can be constructed. The last function in the chain typically performs an action or returns a particular value. The intermediate step functions typically either control the execution of the consecutive step functions by calling them only at certain timesteps, or they manipulate the data returned from the consecutive step functions before they return it themselves. Two such chains are used in the simulations presented here, and both are shown in Listing 5.17. The first one consists of two step functions, where the first is called `at-beginning`. It will be called every timestep, but it will only call its consecutive step functions on the first timestep. Its only consecutive step function is `output-epsilon` which saves the $\varepsilon(\boldsymbol{r})$-grid to a file with a predefined name. The second chain is concerned with saving the fields in the recording layer. The first function called `to-appended` saves the output it gets from its consecutive step functions. The second function in the chain called `at-every` calls its consecutive functions and relays the output with a certain time interval. This is how the actual sampling frequency is realised. The third function called `in-volume` expects its consecutive functions to return data for each voxel in the simulation grid. It relays only the part of the grid which lies within a given "volume". This "volume" can actually also be a plane, line, or even a point, and is here used to specify the recording layer. The forth link in the chain is two step functions which return the actual electric and magnetic fields for the entire volume. Together this chain saves the fields in the recording plane with a certain sampling frequency, such that they can be loaded for post processing.

Listing 5.17: Starting simulation

```
(run-sources+
    (stop-when-fields-decayed 5 Sz (vector3 0 0 cTrans) stopTh)
    (at-beginning output-epsilon)
    (to-appended (string-append filePrefix " T Rxi_" (number->string Rxi) " Ryi_" (number->string Ryi))
        (at-every (/ 1 Fs)
            (in-volume
                (volume (center 0 0 cTrans) (size 1 1 0))
                output-efield output-hfield
            )
        )
    )
)
```

5.2 Execution

5.2.1 Simulation groups

Section 5.1.4 described how MEEP is configured and how a simulation is started. This section will describe how the execution of all the different simulations was organized. As explained in section 4.2.1, a separate simulation must be run for every $\boldsymbol{k_T}$-vector of the source ($\boldsymbol{k_{Ts}}$) that is to be simulated. To achieve adequate resolution in the $\boldsymbol{k_{Ts}}$-space, it was necessary with a rather high number of simulations. To be able to form a radiation pattern from the results it was necessary that for every $\boldsymbol{k_{Ts}}$ that was simulated, a simulation was also necessary for all the $\boldsymbol{k_{Ts}}$-vectors which have the same point in the Brillouin zone ($\boldsymbol{k_B}$). This is most easily accomplished by generating a grid of points to simulate in the Brillouin zone, and for each of these points run a simulation for all $\boldsymbol{k_{Ts}}$ that corresponds to the current $\boldsymbol{k_B}$-point. More precisely it means that for all the points in the Brillouin zone, a simulation must be run with $\boldsymbol{k_{Ts}}$ equal to $\boldsymbol{k_B}$ added an arbitrary number of reciprocal lattice vectors while limiting the length of $\boldsymbol{k_{Ts}}$ to less than the wave number. All these simulations which share a common $\boldsymbol{k_B}$-point will for the rest of this chapter be denoted as a *simulation group*. The control file which was used and is shown in Appendix A performed a simulation for all simulations in such a simulation group. This means it was only necessary to start MEEP from the command line once for every $\boldsymbol{k_B}$-point. The actual $\boldsymbol{k_B}$-point then had to be given as an argument as described in section 5.1.2. Simulating the entire group in one MEEP startup reduced the influence of the overhead running time caused by the initiation process, while allowing external applications to control the resolution in $\boldsymbol{k_{Ts}}$-space.

5.2.2 Parallel computing

The amount of simulations which had to be executed was very large, meaning a single computer would not be able to complete the task within reasonable time. It was therefore necessary to split up the task into smaller pieces and distribute the load on a large amount of computers. A total of 76 computers in a student computer room were used to perform these calculations. As the simulations were performed during the summer holiday most of these rooms were vacant, and could be used for this purpose. This section will focus on how the simulation task was split up between these computers and how this was organized.

Splitting up a simulation series

The term simulation series was introduced in section 4.3.5, and denote all simulations for a certain type of grating. That is all the simulation groups in the $\boldsymbol{k_{T_s}}$-space for all combinations of the geometry parameters that are to be scanned. Each simulation series had approximately 100 combinations of geometry parameters which each consisted of about 100 simulation groups. Keeping in mind that both polarizations must be simulated this means that each simulation series consists of approximately 20000 simulation groups. Although the results must be combined in the end, the different simulation groups do not depend on each other during the simulation phase. This means that the simulation task can easily be split up and shared by multiple computers. The same control file was used for all simulations groups in a simulation series, and the differences in geometry and $\boldsymbol{k_B}$ were specified through parameters as described in section 5.1.2. The simulation series could then be represented by a list of Linux commands which all would invoke MEEP with different parameters. Each line in such a list would then correspond to a simulation group with a particular polarization. For each simulation series such a list was made using MATLAB, and saved to a text file. These files will be denoted as command files in the following sections. Listing 5.18 shows an example of a single line in such a command file.

Listing 5.18: One line of the command file for square grating of holes

```
#meep Fcenter=0.85 Fwidth=1.8 Fs=5 sourceCutoff=10 dSourceSurf=0.9 ref?=false TE?=true kx=0 ky=0 holeDepth=0.05 holeR=0.05 outputDir=\"../res/Surf_d_0.05__R_0.05_TE\" filePrefix=\"xi_1_yi_1\" ../3DLed_firstEd.ctl
```

Load distribution

The easiest way of distributing the different simulation groups between the different computers, or nodes, would be to give each of them a subset of the command file and let them go through it individually. Such a process could probably be automated, but it would be difficult to make adjustments after the process was initiated. Instead the entire command file was placed on a computer with a shared disk. In this way all individual nodes would have access to this single file. It would then be easier to make modifications as only that single file would have to be edited or replaced.

Node simulation loop

To automate the load distribution all nodes were running a simulation loop which is shown in the flow chart in Figure 5.2.1. The main simulation loop ran in a Python script located on a shared disk. Python is a programming language which has a C-like syntax and has code that is easy to read, and is easy to edit. More information about the programming language can be found in the Python documentation [18]. In the simulation loop the nodes would first fetch a command from the shared command file, execute it, save the result on the same shared disk, and then fetch a new command. The first character in each line of the command file was used to indicate whether the command had been executed and completed, was currently being executed or if it had not yet been fetched. In Listing 5.18 the first character is # which was used to indicate that the command had not yet been used by any of the nodes. When a node fetched the command the character was changed to @, and when it was completed it was changed to %. Although the system was rather primitive it was very resilient. If a single node failed or was intentionally stopped, only the single simulation group it was currently simulating would fail. All the other nodes would carry on unnoticed. The single simulation group that failed could then easily be recognized by inspecting the command file. The lines where the first character had not been replaced to % had to be simulated again. This was most easily done by manually changing the first character back to #. In addition new nodes could be added while the simulation was in progress, and without disturbing the simulation process. The communication between the nodes did in this way go through a shared file. To make sure that two or more nodes did not modify this command file simultaneously, each node had to lock it before using it. A file can only be locked by one node simultaneously, and an error is returned if a second node tried to lock it. If such an error was returned the node would wait a few seconds before trying again. This waiting time was set to increase if several attempts to lock it failed.

As shown in Figure 5.2.1, parts of the post processing were also distributed. This was essential in order to sufficiently reduce the amount of data generated to fit it on a regular hard drive. Also the process of loading the files generated by MEEP into MATLAB was slightly tedious. Saving the data after post processing in MATLAB's data format therefore significantly reduced the loading time for the next processing step.

Not all the program execution was performed in the Python script or in the MATLAB post processing code. Figure 5.2.1 also show that parts of it was performed in a Linux batch script. This is a script that runs regular Linux commands in the command window. This was in the form of another loop of which the only purpose was to execute the Python simulation loop, and when it had finished it performed a time delay before starting the simulation loop again. Having two nested loop like this had mainly three purposes. The first was to ease the initiation of a new simulation series. When the command file no longer contained any unstarted commands, the Python script would finish. Since the batch script constantly tried to restart it, a new simulation series could be started simply by replacing the command file. It was not necessary to manually restart the nodes. The second purpose was increased resiliency against errors. If an error occurred in the Python script, MEEP or the MATLAB post processing script, the program execution would fall back to the batch script. It would then automatically restart the Python script, meaning the manual intervention would not be necessary. The third reason was that since the Python script was constantly called when the system was not running simulations, it was possible to change the content of the Python script effectively changing the entire simulation process. This flexibility was used a few times to optimize the simulation loop, and to eliminate some sources of errors.

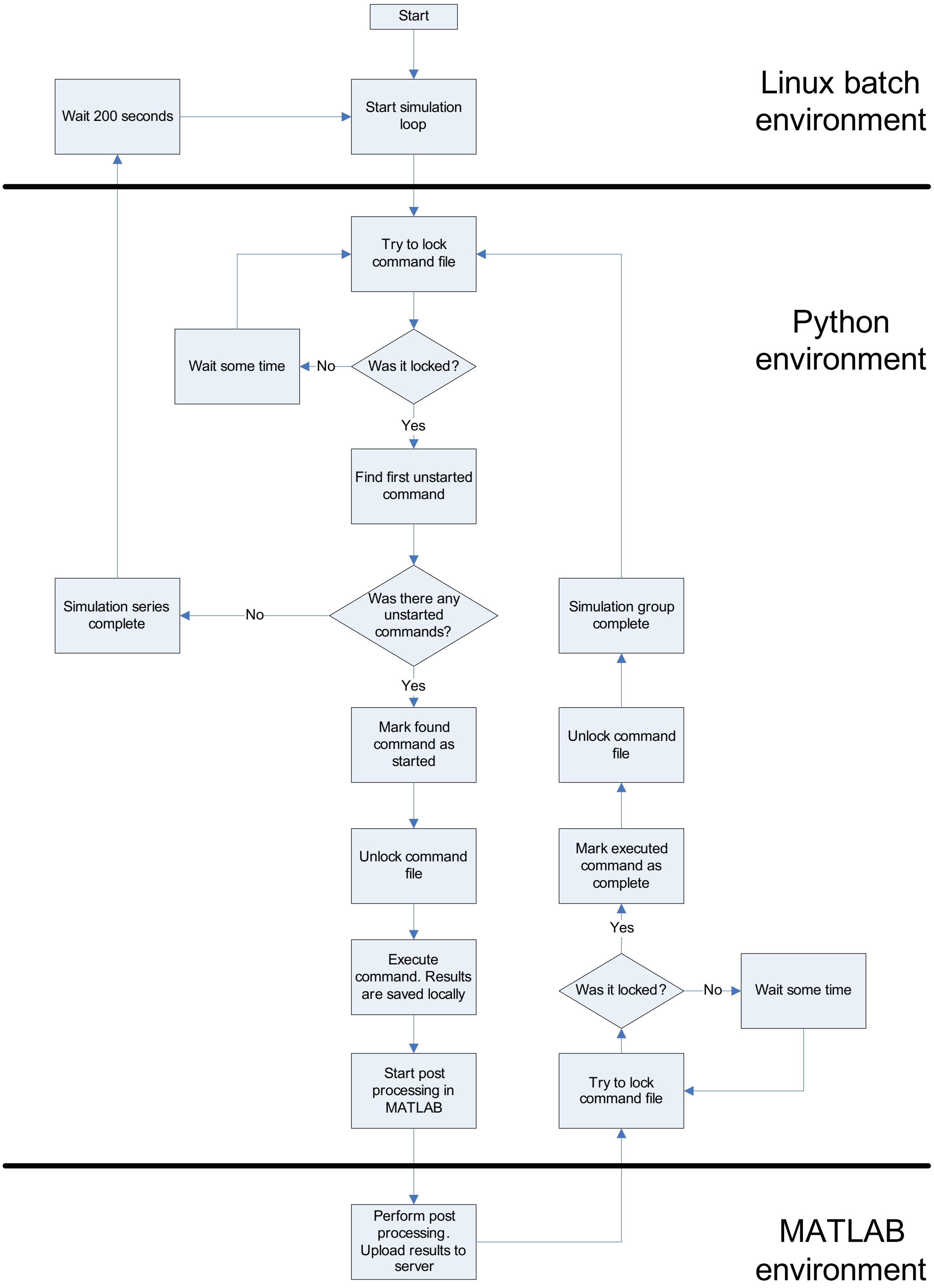

Start
Linux batch environment
Wait 200 seconds
Start simulation loop
Python environment
Try to lock command file
Wait some time
No
Was it locked?
Yes
Find first unstarted command
Simulation series complete
No
Was there any unstarted commands?
Simulation group complete
Yes
Mark found command as started
Unlock command file
Unlock command file
Mark executed command as complete
Yes
Execute command. Results are saved locally
Was it locked?
No
Wait some time
Start post processing in MATLAB
Try to lock command file
Perform post processing. Upload results to server
MATLAB environment

Figure 5.2.1: Flowchart for program running on each simulation node

5.3 Post processing

As mentioned in section 5.2.2, parts of the post processing in MATLAB was carried out directly after MEEP had completed a simulation on the individual nodes. The program performing the calculations was compiled MATLAB code. This is MATLAB code that can run without having MATLAB installed. It was started by invoking an executable file as an ordinary program. The actual code will not be presented here as it is too extensive. It can however be found on the enclosed disc. Although the mathematics can be expressed compactly, the actual details concerned with finding files, determining resolutions, keeping track of matrix indices etc. are rather substantial.

MEEP produced one result file per simulation, meaning each simulation group contained about 10-20 files. The MATLAB code first loaded these and put all the data in the same data structure. Secondly the data was Fourier transformed producing the transmitted power spectrum. This spectrum consisted of all frequencies from zero to the sampling frequency. To reduce the amount of data, only the interval where the source had been emitting was kept, the rest was rejected. Next the Poyntings vector and the total transmitted power flux was calculated. This is the denominator of equation (4.4.3), and was done at this time to ease the calculation of the surface transmission efficiency in later processing steps. Next the spatial Fourier transform of the fields was calculated. This was done as described in section 4.4.3 by removing the Bloch wave vector to make the fields periodic. As mentioned in section 4.4.3 it is possible to reduce the needed storage space about 100 times by removing the points in the spatial Fourier transform that lie outside the light cone. This was utilized here and was definitively the most storage space saving technique utilized. Finally the spatial and temporal Fourier transformed fields and the total power flux were saved to a shared disk. The calculated Poyntings vector was rejected since it was redundant, and since storage space constraints were a concern. The data from the simulation of one simulation group was saved as a 2D array of structures to a MATLAB data file. Each element in the 2D array corresponded to a certain $\boldsymbol{k_{Ts}}$, and the axes in the 2D system was k_{xs} and k_{ys}. The size in the two directions varied depending on where in the Brillouin zone the simulation group belonged, but for square gratings it was approximately 5 elements in the k_{xs}-direction and 3 in the k_{ys}-direction. Each array element was as mentioned, a structure containing the fields shown in Listing 5.19.

Listing 5.19: One element in the saved 2D array

```
    params: [1x1 struct]
        Pf: [93x1 single]
         f: [1x93 double]
       Efk: [4-D single]
       Hfk: [4-D single]
```

Here `Pf` is the power spectrum , `f` is the corresponding frequency array, and `Efk` and `Hfk` are the fields. The fields are a 4D array where the dimensions are k_{xa}, k_{ya}, frequency, and since its a vector a fourth dimension must be used to list the three elements. The field `params` is itself a structure with many fields. This is supposed to be all the parameters used in the actual simulation. At least all parameters listed in Tables 4.5.2 and 4.5.3 can be found. Appendix A has a complete list starting at line 172. Here all these parameters are saved by calling the function `outputVar` for each of them. One such file was created for each simulation group with a certain polarization. All the files from one certain grating geometry and a certain polarization were placed in the same folder. The results from the entire simulation series therefore consisted of a set of such folders covering all the simulated geometry settings and both polarizations.

5.4 Surface transmission efficiency

5.4.1 Step one

The most important parameter for the grating which the simulations are supposed to give, is the surface transmission efficiency η_s which is described in sections 3.4 and 4.4.2. It specifies how much of the optical power incident on the surface is transmitted through to air, and is calculated as described in section 4.4.2. The calculation of this parameter was done in two steps where the intermediate result was saved to a file. The code for both these steps is shown in Appendix B. The calculation was divided into two steps to ease the programming and debugging of the software. The amounts of data which were to be processed were very large, meaning it was a time consuming process to complete the calculations. Debugging code that requires much time to run is very tedious as it typically is necessary to run it several times. Dividing the calculation into steps therefore makes the programming more efficient. After the simulations, the data was saved in files which covered a simulation group. This kind of data organization is suited for calculating the radiation pattern, but not for calculating η_s. The first step therefore loaded all files for a certain geometry and extracted the transmitted power spectra. These were then organized in a regular three dimensional array with k_{xs}, k_{xa} and frequency as the axes. Although this is the same as organization as the saved files, there is a difference because the different simulation groups are no longer separated. Everything is instead gathered in the same array which is saved to a single file. This step takes care of the rather lengthy process of loading all the result files. The next step can therefore load only one file per geometry setting. Listing B.1 in Appendix B shows the Matlab-function for performing the first step to the result of a single geometry setting and polarization. Another function was therefore used to run it for all these settings and polarizations.

5.4.2 Step two

The second step loaded the files created by the first and then calculated η_s. This process involved four Matlab functions which all are listed in Table 5.4.1 and which can be found in Appendix B. The function `extractEtaS` is the top level function and is responsible for processing all geometry settings for a simulation series. The root folder of the simulation series and the pathname of a data file with the incident radiation in Figure 3.3.14 or 3.3.15 are the only input parameters. The function will automatically find the results from all geometry settings using the `dir`-function. It also knows the location of the appropriate reference simulation with normal incidence. The reference for all angles can then be found using the theory from section 4.6. This information is partly used by the function itself and partly used as input to lower level functions. The final result is a three dimensional array of surface transmission efficiency values. The axes of this array are optical frequency and depth and radius of the photonic crystal holes. This structure and vectors defining the axes were saved to a single file which rapidly could be loaded for plotting purposes.

Table 5.4.1: Matlab functions involved in step two

Function name	Description
`extractEtaS`	This is the top level function for a simulation series. - Loads all data for a simulation series - Loads the reference with normal incidence. - Calls `artificialRef` for each simulation and calculates transmission spectra - Calls `calcEtaS` to get the η_s for each geometry setting. The path name of a data file containing the incident light in Figure 3.3.14 or 3.3.15 are given as a input. - Sorts the results after the geometry parameters and saves them to a single file
`calcEtaS`	The function calculates η_s based on the transmission spectra and the incident radiation. The integral in equation (4.4.4) is calculated using `trapezint2D`
`trapezint2D`	The function calculates the 2D integral of a function defined by a matrix and two vectors that define the axes. The spacing between the rows and columns does not need to be constant.
`artificialRef`	Function that calculates the reference for light emitted at a given angle based on a simulation with light emitted in the forward direction.

5.5 Radiation pattern

The most advanced results from the simulations are the radiation patterns described in section 4.4.3. The overall procedure was to first calculate the coupling coefficients Γ in equation (4.1.11) for all simulations. This requires the spatially Fourier transformed fields and the reference. Secondly these coefficients are weighted by the incident power from Figures 3.3.14 and 3.3.15 to find the power that is coupled by Γ. Thirdly the combined radiation pattern can be be formed by summing the contribution from all simulations in the same simulation group. This is what equation (4.1.11) states. The Matlab code performing this is not included in this thesis, but can be found on the enclosed disc. For a given simulation series, the calculation starts with the top level function `findRadPatsAndSave`. The enumerated list below goes through the main steps in the calculation and is recommended to read prior to looking at the actual Matlab code. The Matlab-function responsible for performing each item in the list is stated in parentheses.

1. Function `findRadPatsAndSave` is called with the root folder of the simulation series as input.
2. Find the geometry settings (radius and depth) for which a simulation has been performed (`getSimmedRandd`)
3. For each geometry setting, run the function `calcTransRadPat` to calculate the radiation pattern (`findRadPatsAndSave`)
   (a) Load reference with normal incidence. (`calcTransRadPat`)
   (b) Find all simulated simulation groups. (`calcTransRadPat`)
   (c) Organize the simulation groups in a matrix with k_{Bx} and k_{By} as axes. (`calcTransRadPat`)
   (d) Load simulation data from all simulation groups (`calcTransRadPat`)
   (e) Load files with incident power from Figures 3.3.14 and 3.3.15. (`getPinc`)
   (f) For each simulation group:
      i. For each simulation in group:
         A. Calculate Poyntings vector from simulated fields. (`calcTransmissionCoeff`)
         B. Calculate reference (`artificialRef`)
         C. Calculate coupling coefficient, Γ, which defines the coupling $\boldsymbol{k_{Ts}}$ -> $\boldsymbol{k_{Ta}}$ for all $\boldsymbol{k_{Ta}}$ in simulation group. (`calcTransmissionCoeff`)
         D. Scale by incident power to find transmitted power from current $\boldsymbol{k_{Ts}}$. (`calcTransmissionCoeff`)
         E. Add contribution from current simulation ($\boldsymbol{k_{Ts}}$) to the combined transmitted power from all simulations ($\boldsymbol{k_{Ts}}$) in the group. (`calcTransmissionCoeff`)
      ii. Generate matrices defining k_{xa} and k_{ya} valid for the calculated transmitted powers. This means to combine $\boldsymbol{k_B}$ of the simulation group with the correct number of reciprocal lattice vectors. For reference see Figure 4.4.2. (`calcTransmissionCoeff`)
      iii. Reorganize the transmitted power in a single three dimensional array. The axis of the matrix is k_{xa} and k_{ya} and optical frequency. (`formRadPattern`)
      iv. Unwrap diagonal mirror symmetry if present
      v. Unwrap mirror symmetry along x and y axis
   (g) Save the radiation pattern for the current geometry setting. (`calcTransRadPat`)

4. Save all the radiation patterns for all geometry settings to a single file. (`findRadPatsAndSave`)

Chapter 6

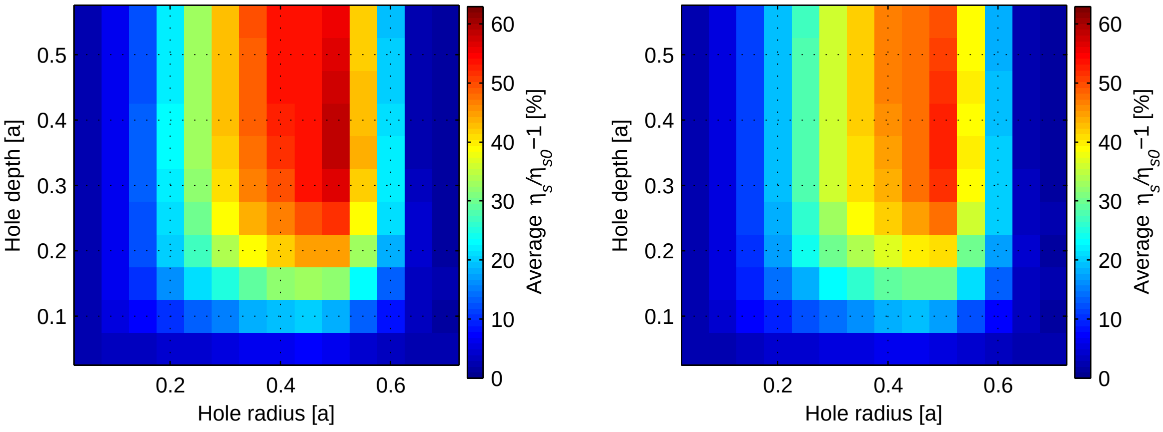
Results

6.1 Surface transmission efficiency

The surface transmission efficiency η_s is defined in section 3.4, and is the ratio of the power transmitted through the surface to the total incident power. In this section η_s is plotted for the simulated gratings. Instead of showing the actual η_s, the plots show the improvement over η_s for a plane surface which is shown in Table 3.4.1. Because η_s is dependent on the three variables hole depth, d, radius R and frequency, it is not possible to show the results for all possible combinations. Instead the average over all frequencies is shown for each d, R-combination. In addition the spectra of a few d, R-points are plotted. Since the physical frequency is pinned by a photon energy of either 0.65eV or 0.7eV, the frequency axis is translated to a lattice constant axis according to the principles established in section 2.4. Other physical lengths, for example hole depth and radius, are originally denoted in MEEP units described in section 4.5.1. Since the lattice constant, a, always has the length of unity, this means that all lengths are effectively fractions of the physical a in μm. Because of this, the axes measuring lengths other than the lattice constant are labeled as having the unit of one lattice constant a. In addition to the η_s-results, some of the simulated gratings are also shown in this section. The plots are based on the $\varepsilon(\boldsymbol{r})$-grid output from the simulation software. The grids are replicated some times in each direction, and are illustrated as iso-surfaces of the $\varepsilon(\boldsymbol{r})$-functions. As mentioned in section 4.3.5, the three grating structures presented in section 4.3.4 each have a normal frequency and normal resolution simulation series. Sections 6.1.1, 6.1.2 and 6.1.3 show the η_s-results from these series. Section 6.1.4 shows the results form the low frequency simulation of the square grating of holes, and sections 6.1.5 and 6.1.6 show the results from the high resolution simulations with normal and low frequency respectively.

6.1.1 Square grating of holes

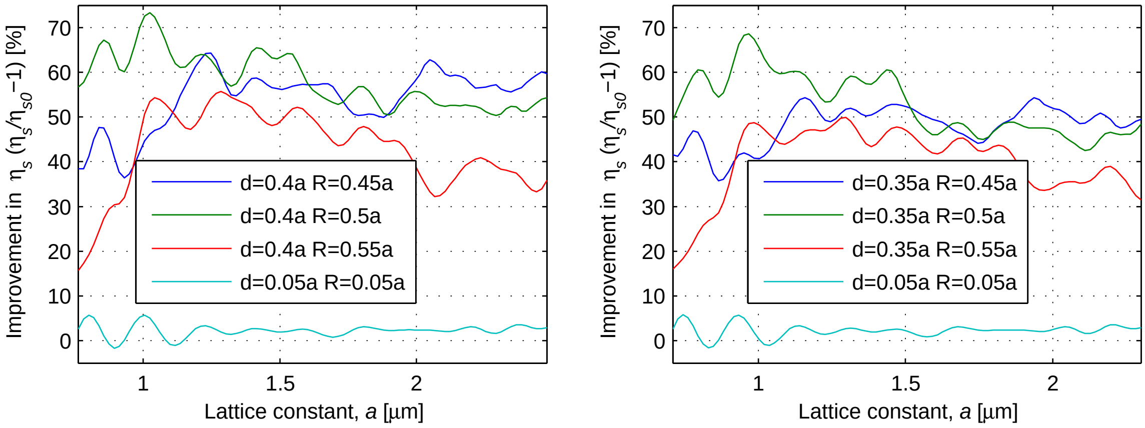
The η_s-results from the normal frequency, normal resolution simulation series of the square grating of holes are plotted in this section. Figure 6.1.1(a) shows that the maximum average improvement for light with photon energy of 0.65eV is 58.2%, and that this can be acheived when $R = 0.5a$ and $d = 0.4a$ which is the structure shown in Figure 6.1.3(a). Figure 6.1.1(b) likewise shows that the maximum average improvement for light with photon energy of 0.7eV is 52.8%, and that this can be achieved when $R = 0.5a$ and $d = 0.35a$. Figure 6.1.2 shows four transmission spectra for each of the photon energies. These are for the structures with highest average improvement, one with smaller, and one with larger hole radius, and one which is close to a plane surface. The peak improvement for 0.65eV is 73.3%, and can be achieved when $a = 1.02\mu$m, $R = 0.5a = 0.512\mu$m, and $d = 0.4a = 0.41\mu$m. The peak improvement for 0.7eV is 68.6%, and can be achieved when $a = 0.97\mu$m, $R = 0.5a = 0.48\mu$m, and $d = 0.35a = 0.34\mu$m. Figure 6.1.3 shows two of the simulated structures, where the on in Figure 6.1.3(a) caused maximum average improvement for light at 0.65eV.



(a) 0.65eV

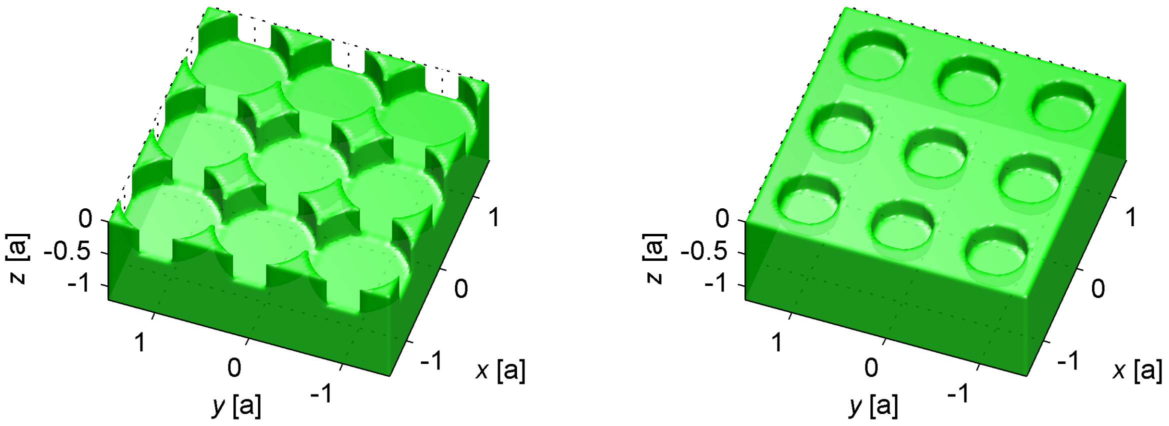
(b) 0.7eV

Figure 6.1.1: Improvement in η_s with a square grating of holes compared to a plane surface. The plotted values are averages over the simulated spectrum.



(a) 0.65eV (b) 0.7eV

Figure 6.1.2: Spectra showing the improvement in η_s with a square grating of holes compared to a plane surface. The figures show the spectra for four points in Figure 6.1.1



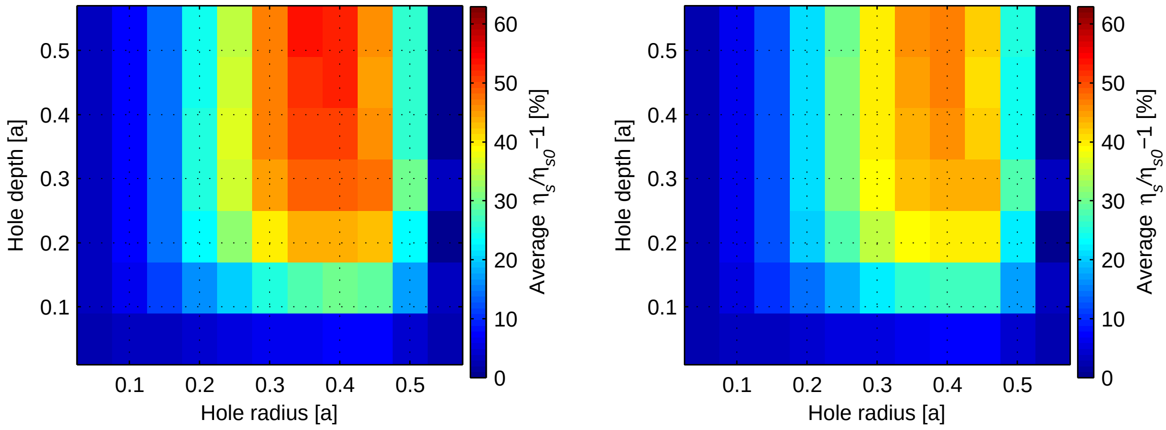
(a) d=0.4, R=0.5 (b) d=0.25, R=0.3

Figure 6.1.3: $\varepsilon(\boldsymbol{r})$ iso-surface of two square gratings of holes

.

6.1.2 Hexagonal grating of holes

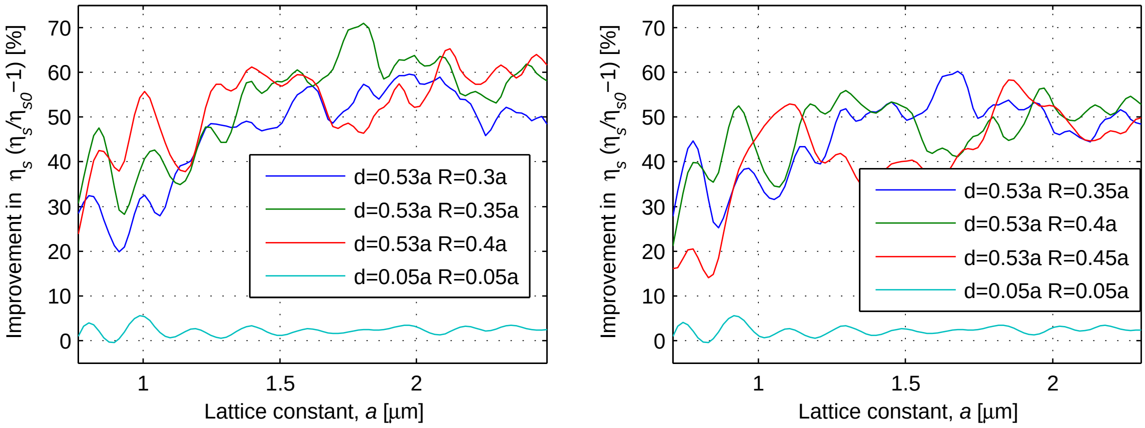
The η_s-results from the normal frequency, normal resolution simulation series of the hexagonal grating of holes are plotted in this section. Figure 6.1.4(a) shows that the maximum average improvement for light with photon energy of 0.65eV is 53.21%, and that this can be acheived when $R = 0.35a$ and $d = 0.53a$ which is the structure shown in Figure 6.1.6(a). Figure 6.1.4(b) likewise shows that the maximum average improvement for light with photon energy of 0.7eV is 47.1%, and that this can be achieved when $R = 0.4a$ and $d = 0.53a$. Figure 6.1.5 shows four transmission spectra for each of the photon energies. These are for the structures with highest average improvement, one with smaller, and one with larger hole radius, and one which is close to a plane surface. The peak improvement for 0.65eV is 70.1%, and can be achieved when $a = 1.81\mu$m, $R = 0.35a = 0.63\mu$m, and $d = 0.53a = 0.96\mu$m. The peak improvement for 0.7eV is 60.2%, and can be achieved when $a = 1.68\mu$m, $R = 0.35a = 0.59\mu$m, and $d = 0.53a = 0.89\mu$m. Figure 6.1.6 shows two of the simulated structures, where the on in Figure 6.1.6(a) caused maximum average improvement for light at 0.65eV.



(a) 0.65eV

(b) 0.7eV

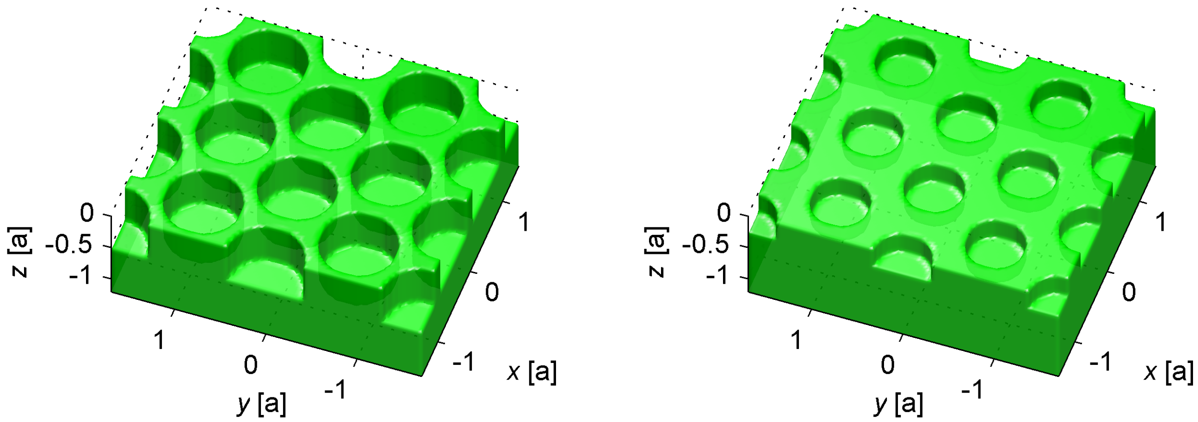
Figure 6.1.4: Improvement in η_s with a hexagonal grating of holes compared to a plane surface. The plotted values are averages over the simulated spectrum.



(a) 0.65eV

(b) 0.7eV

Figure 6.1.5: Spectra showing the improvement in η_s with a hexagonal grating of holes compared to a plane surface. The figures show the spectra for four points in Figure 6.1.4



(a) d=0.53, R=0.35

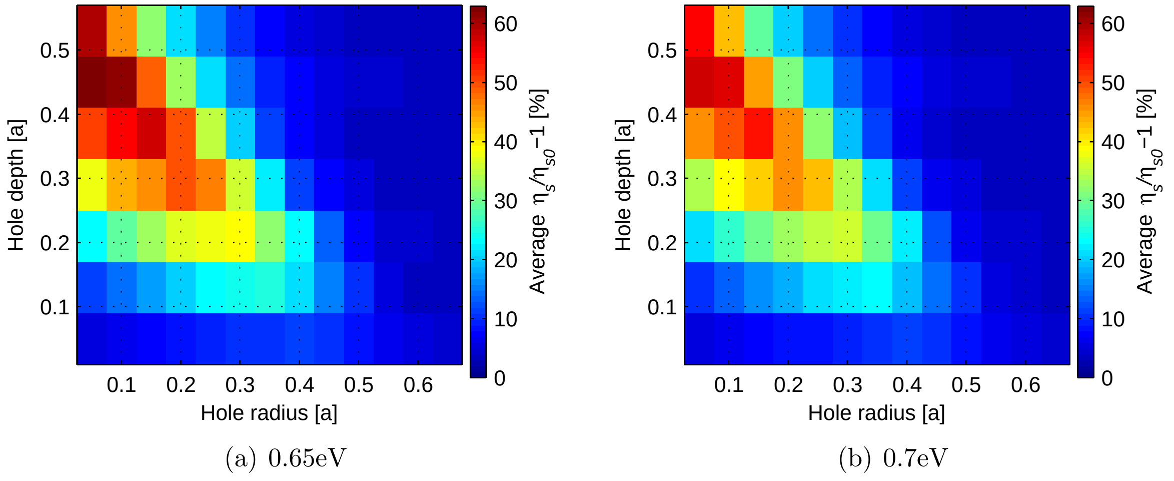
(b) d=0.29, R=0.3

Figure 6.1.6: $\varepsilon(\boldsymbol{r})$ iso-surface of two square gratings of holes

.

6.1.3 Square grating of isotropically etched holes

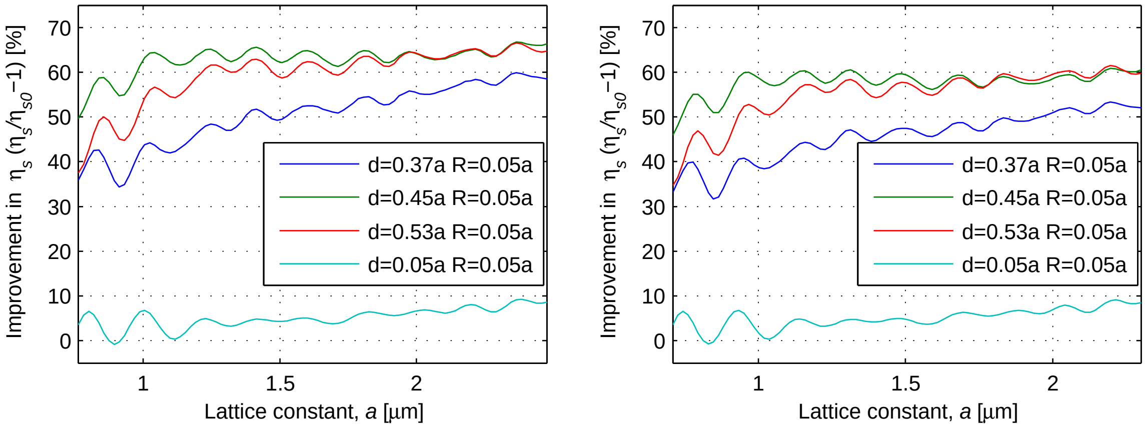
The η_s-results from the normal frequency, normal resolution simulation series of the square grating of isotropically etched holes holes are plotted in this section. Figure 6.1.7(a) shows that the maximum average improvement for light with photon energy of 0.65eV is 62.8%, and that this can be acheived when $R = 0.05a$ and $d = 0.45a$ which is the structure shown in Figure 6.1.9(a). Figure 6.1.7(b) likewise shows that the maximum average improvement for light with photon energy of 0.7eV is 57.8%, and that this can be achieved when $R = 0.05a$ and $d = 0.45a$. Figure 6.1.8 shows four transmission spectra for each of the photon energies. These are for the structures with highest average improvement, one with smaller, and one with larger hole depth, and one which is close to a plane surface. The peak improvement for 0.65eV is 66.7%, and can be achieved when $a = 2.37\mu$m, $R = 0.05a = 0.118\mu$m, and $d = 0.45a = 1.06\mu$m. The peak improvement for 0.7eV is 61.6%, and can be achieved when $a = 2.20\mu$m, $R = 0.05a = 0.110\mu$m, and $d = 0.53a = 1.16\mu$m. Figure 6.1.9 shows two of the simulated structures, where the on in Figure 6.1.9(a) caused maximum average improvement for light at 0.65eV.



(a) 0.65eV

(b) 0.7eV

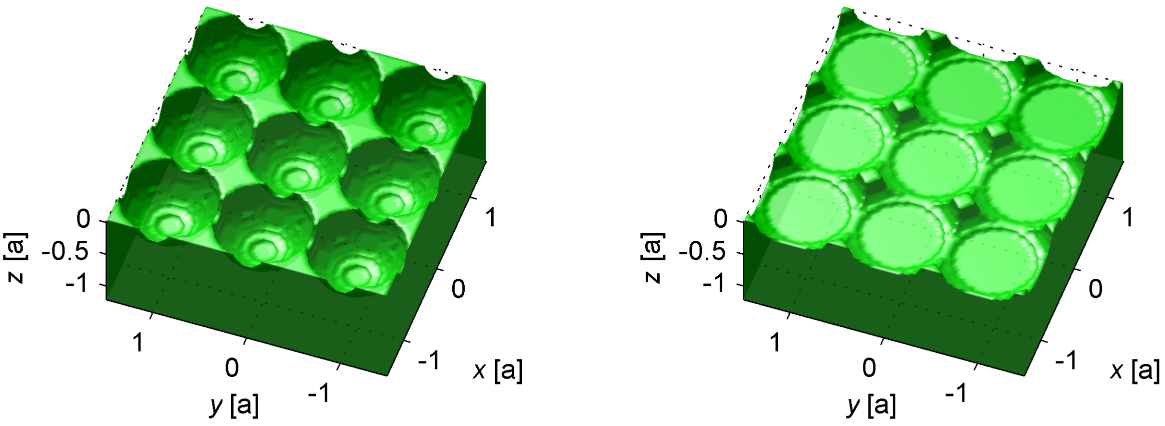
Figure 6.1.7: Improvement in η_s with a square grating of isotropically etched holes compared to a plane surface. The plotted values are averages over the simulated spectrum.



(a) 0.65eV

(b) 0.7eV

Figure 6.1.8: Spectra showing the improvement in η_s with a square grating of isotropically etched holes compared to a plane surface. The figures show the spectra for four points in Figure 6.1.7



(a) d=0.45, R=0.05

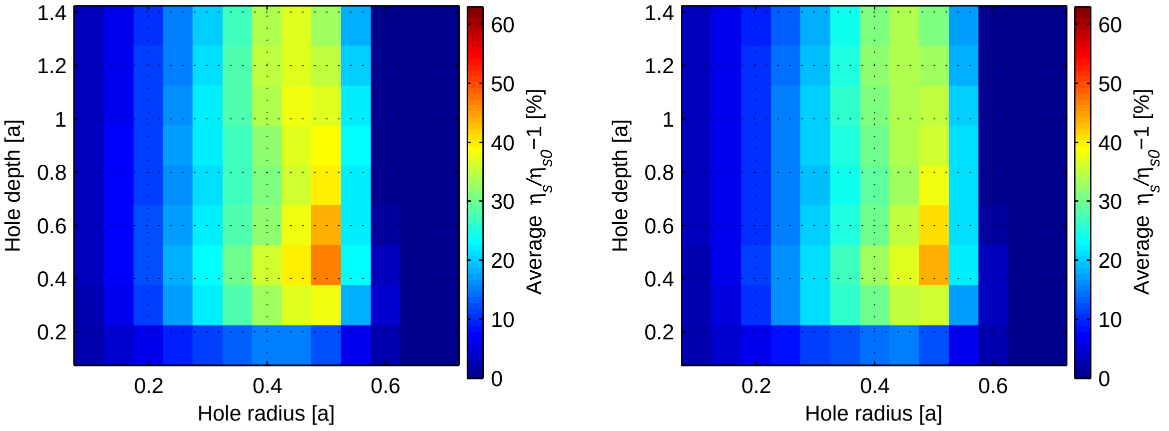
(b) d=0.21, R=0.4

Figure 6.1.9: $\varepsilon(\boldsymbol{r})$ iso-surface of two square gratings of holes

.

6.1.4 Square grating of holes, low frequency

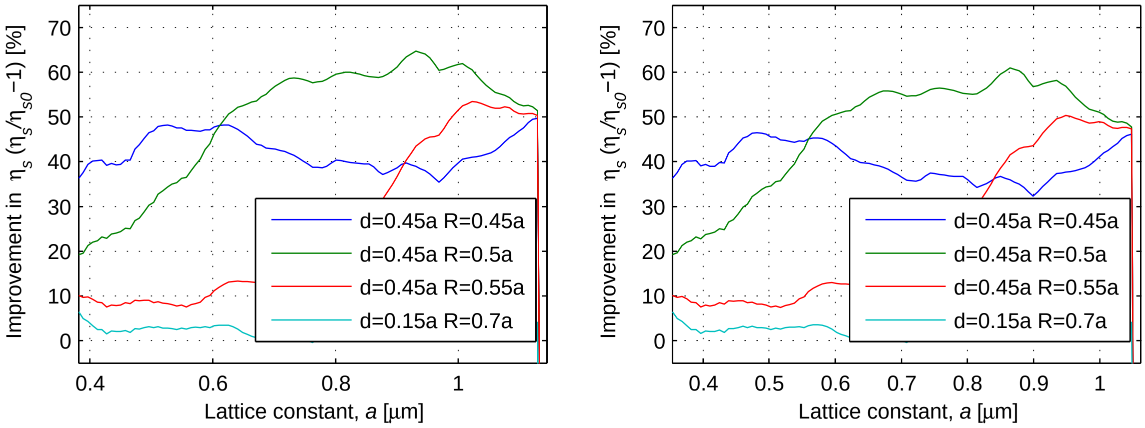
The η_s-results from the low frequency, normal resolution simulation series of the square grating of holes are plotted in this section. Figure 6.1.10(a) shows that the maximum average improvement for light with photon energy of 0.65eV is 46.5%, and that this can be acheived when $R = 0.5a$ and $d = 0.45a$ which is the structure shown in Figure 6.1.12(a). Figure 6.1.4(b) likewise shows that the maximum average improvement for light with photon energy of 0.7eV is 44.1%, and that this can be achieved when $R = 0.5a$ and $d = 0.45a$. Figure 6.1.11 shows four transmission spectra for each of the photon energies. These are for the structures with highest average improvement, one with smaller, and one with larger hole radius, and one which is close to a plane surface. The peak improvement for 0.65eV is 64.7%, and can be achieved when $a = 0.93\mu\text{m}$, $R = 0.5a = 0.47\mu\text{m}$, and $d = 0.45a = 0.42\mu\text{m}$. The peak improvement for 0.7eV is 61.0%, and can be achieved when $a = 0.86\mu\text{m}$, $R = 0.5a = 0.43\mu\text{m}$, and $d = 0.45a = 0.39\mu\text{m}$. Figure 6.1.12 shows two of the simulated structures, where the on in Figure 6.1.12(a) caused maximum average improvement for light at 0.65eV.



(a) 0.65eV

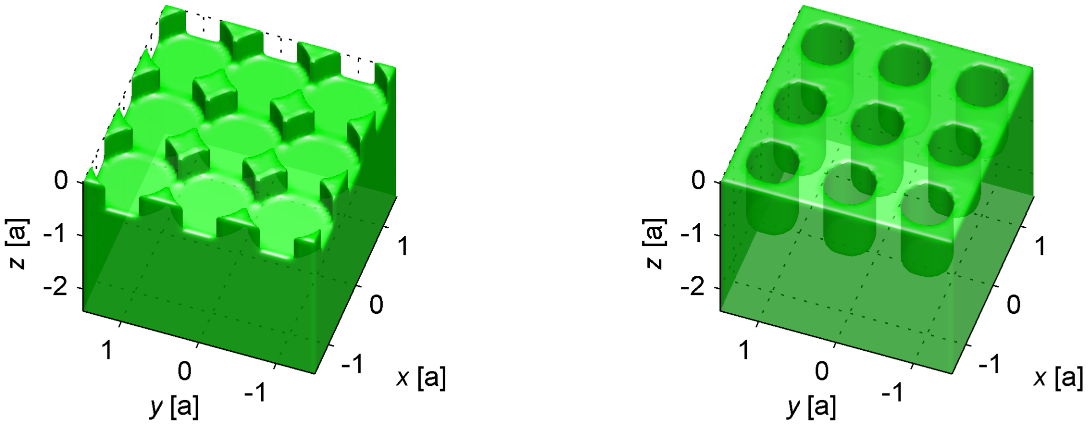
(b) 0.7eV

Figure 6.1.10: Improvement in η_s with a square grating of holes compared to a plane surface. The plotted values are averages over the simulated spectrum. Low frequency simulation



(a) 0.65eV (b) 0.7eV

Figure 6.1.11: Spectra showing the improvement in η_s with a square grating of holes compared to a plane surface. The figures show the spectra for four points in Figure 6.1.10. Low frequency simulation



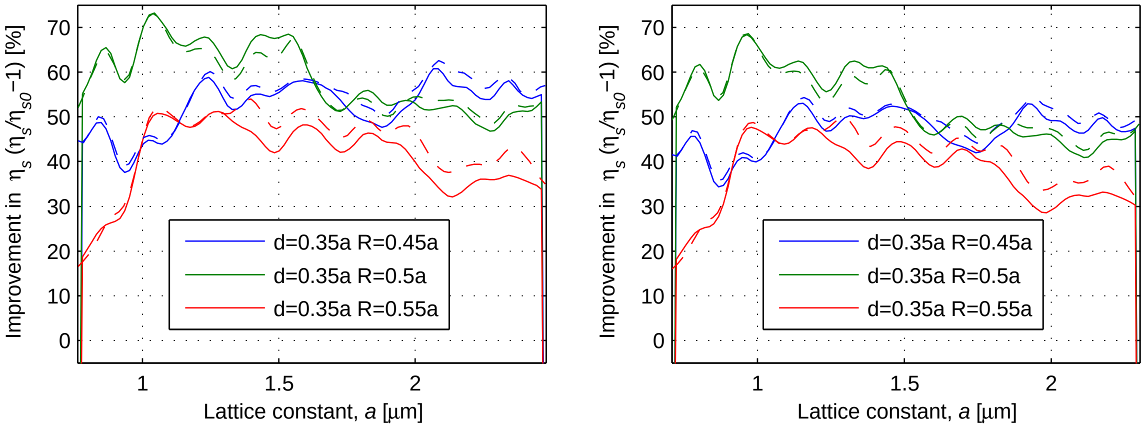
(a) d=0.45, R=0.5 (b) d=1.05, R=0.3

Figure 6.1.12: $\varepsilon(\boldsymbol{r})$ iso-surface of two square gratings of holes. Low frequency simulation

.

6.1.5 Square grating of holes. High resolution

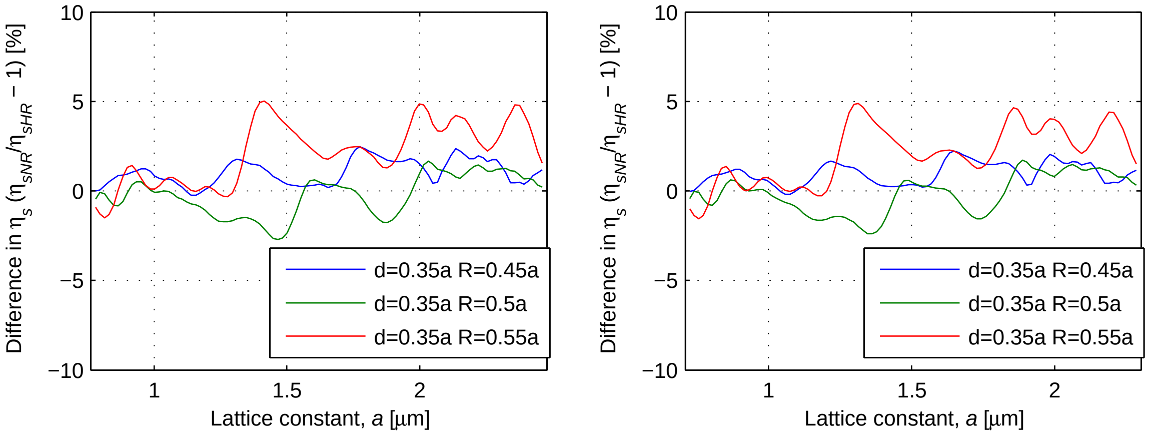
The η_s-results from the normal frequency, high resolution simulation series of the square grating of holes are plotted in this section. Figure 6.1.13 plots three spectra showing the improvement in η_s for each of the two photon energies 0.65eV and 0.7eV. The solid lines are the results from the high resolution simulation series, while the stippled lines are from normal resolution simulations of the exact same structure. Figure 6.1.2(b) also shows the η_s improvement spectra for 0.7eV for the same structures. Figure 6.1.14 shows the relative difference between η_s calculated with normal and high resolution. If the high resolution simulations were assumed to be exact, this would be the error in the results from the normal resolution simulations.



(a) 0.65eV

(b) 0.7eV

Figure 6.1.13: Spectrum showing the improvement in η_s with a square grating of holes compared to a plane surface. The solid line is for the high resolution simulation, and the stippled lines if for the normal resolution simulations.



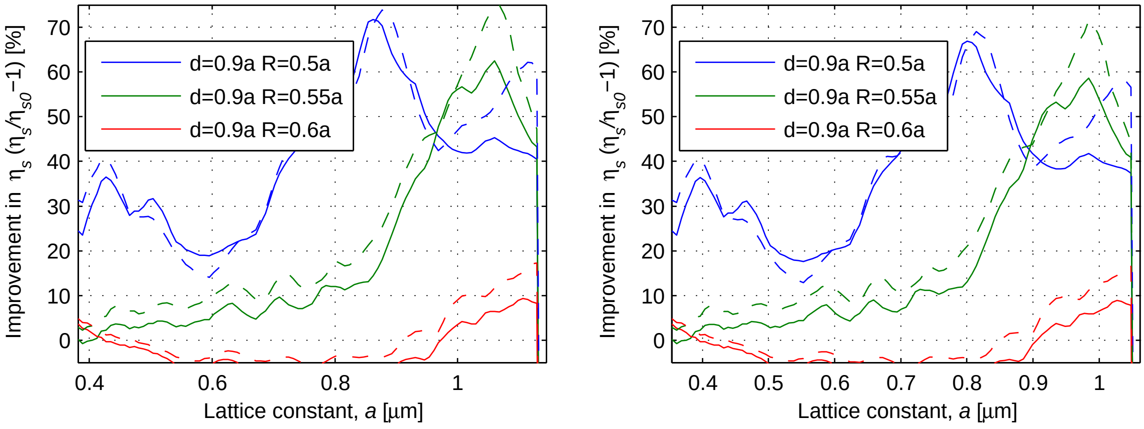
(a) 0.65eV

(b) 0.7eV

Figure 6.1.14: Spectrum showing the relative difference in η_s between the high resolution and the normal resolution simulations with normal frequency

6.1.6 Square grating of holes. Low frequency, high resolution

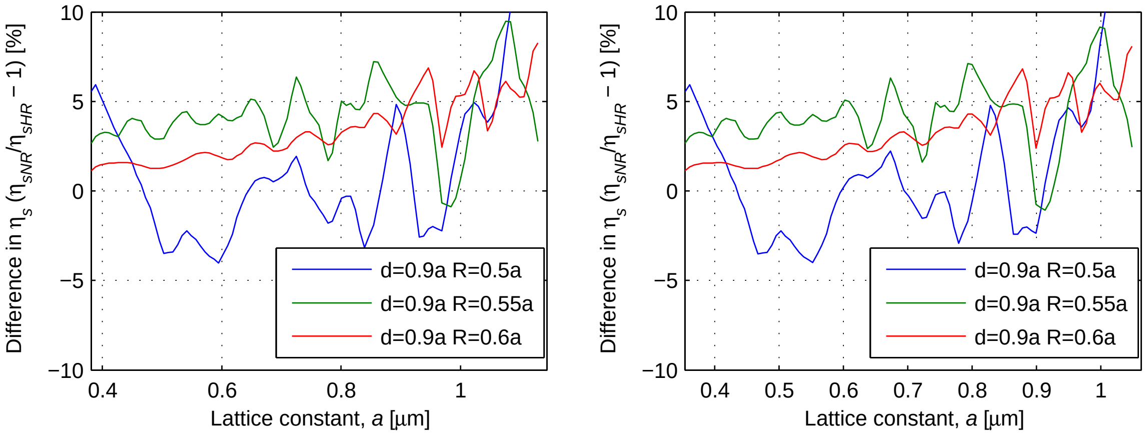
The η_s-results from the low frequency, high resolution simulation series of the square grating of holes are plotted in this section. Figure 6.1.15 plots three spectra showing the improvement in η_s for each of the two photon energies 0.65eV and 0.7eV. The solid lines are the results from the high resolution simulation series, while the stippled lines are from normal resolution simulations of the exact same structure. Figure 6.1.16 shows the relative difference between η_s calculated with normal and high resolution. If the high resolution simulations were assumed to be exact, this would be the error in the results from the normal resolution simulations.



(a) 0.65eV

(b) 0.7eV

Figure 6.1.15: Spectrum showing the improvement in η_s with a square grating of holes compared to a plane surface. The solid line is for the high resolution low frequency simulation, and the stippled lines if for the normal resolution low frequency simulations.



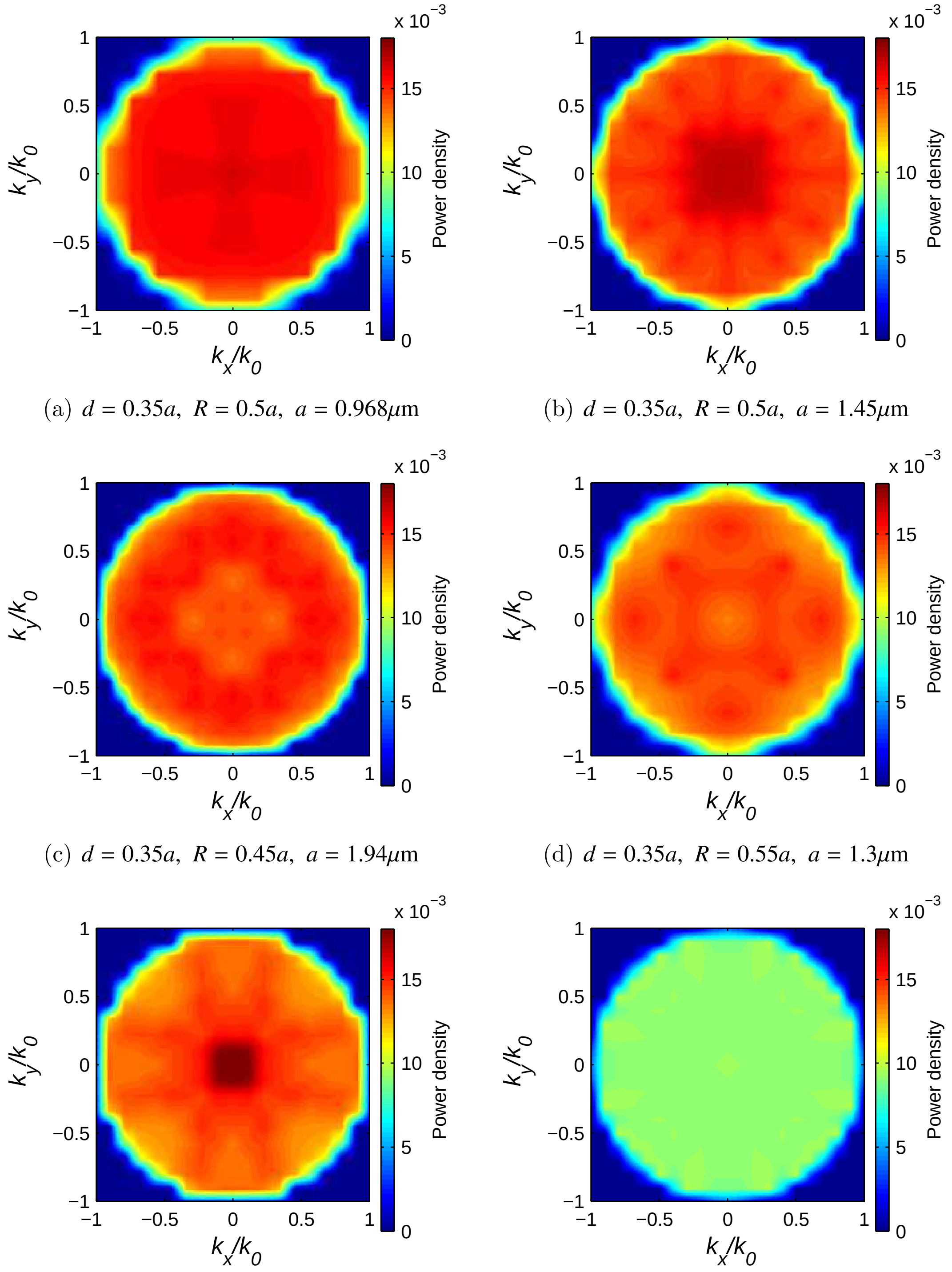
(a) 0.65eV

(b) 0.7eV

Figure 6.1.16: Spectrum showing the relative difference in η_s between the high resolution and the normal resolution simulations with low frequency

6.2 Radiation patterns

In this section a selection of radiation patterns for the light with photon energy of 0.7eV are plotted. Data for all radiation patterns for both the photon energies of 0.65eV and 0.7eV can be found on the enclosed disc. The radiation patterns plot the power density per unit area in the $k_x/k_0 - k_y/k_0$-plane. The applied incident power distribution function has a total power of unity, meaning integrating the radiation patterns over the $k_x/k_0 - k_y/k_0$-plane, should give the surface transmission efficiency η_s. This ensures that the absolute amplitudes in the plots are directly comparable, even for different frequencies. Because the figures illustrate power densities over a plane with normalized axes (e.g. k_x/k_0), and since they show the transmitted power per unit incident power, the power density axes are unitless. The plots are all linearly interpolated to make them smooth. Figure 6.2.1 shows six radiation patterns from various configurations of the normal frequency simulation of the square grating of holes, Figure 6.2.2 shows the same for the hexagonal grating of holes, Figure 6.2.3 shows it for the square grating of isotropically etched holes, and Figure 6.2.4 shows six patterns from the low frequency simulation of the square grating of holes. With the exception of Figure 6.2.3, sub figure (a) of the radiation pattern Figures 6.2.1 to 6.2.4 are for the configuration with highest improvement in η_s. Figure 6.2.3(a) shows the radiation pattern for a configuration which has very near to maximum improvement in η_s for the square grating of isotropically etched holes. This configuration was chosen as it lies closer to the center of the simulated spectrum, where the accuracy is believed to be somewhat better. For all four Figures 6.2.1 to 6.2.4, the sub figure (f) is from a grating configuration that is very close to a plane surface, and is meant as a reference. The other four sub figures (b)-(e) of Figures 6.2.1 to 6.2.4 are for various grating configurations which all have η_s improvement spectra which are plotted in Figures 6.1.2(b), 6.1.5(b), 6.1.8(b), or 6.1.11(b).



(a) $d = 0.35a,\ R = 0.5a,\ a = 0.968\mu$m

(b) $d = 0.35a,\ R = 0.5a,\ a = 1.45\mu$m

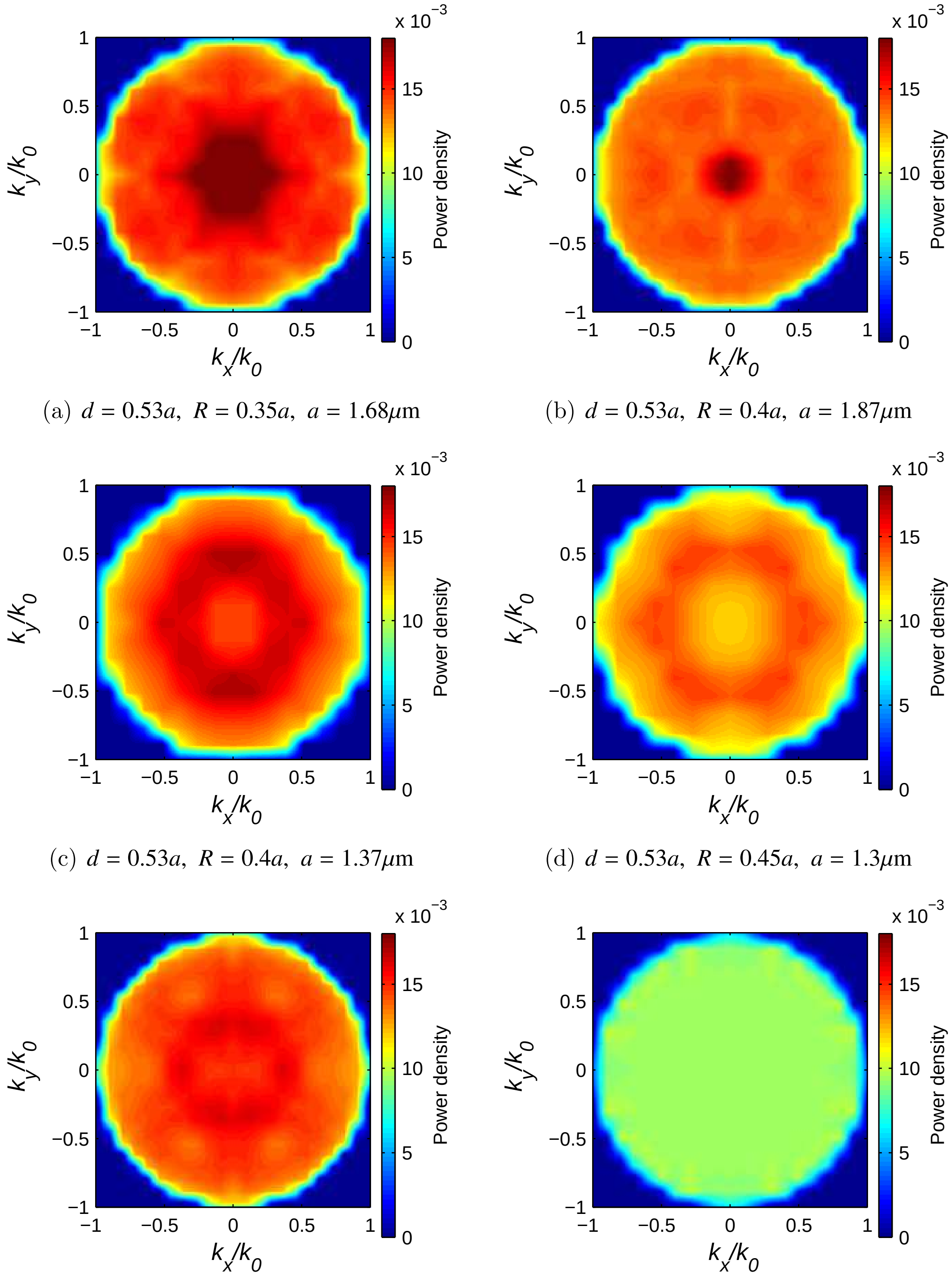
(c) $d = 0.35a,\ R = 0.45a,\ a = 1.94\mu$m

(d) $d = 0.35a,\ R = 0.55a,\ a = 1.3\mu$m

(e) $d = 0.55a,\ R = 0.5a,\ a = 1.56\mu$m

(f) $d = 0.05a,\ R = 0.05a,\ a = 1.76\mu$m

Figure 6.2.1: Radiation patterns for the square grating of holes at an electron energy of 0.7eV



(a) $d = 0.53a, \ R = 0.35a, \ a = 1.68\mu$m

(b) $d = 0.53a, \ R = 0.4a, \ a = 1.87\mu$m

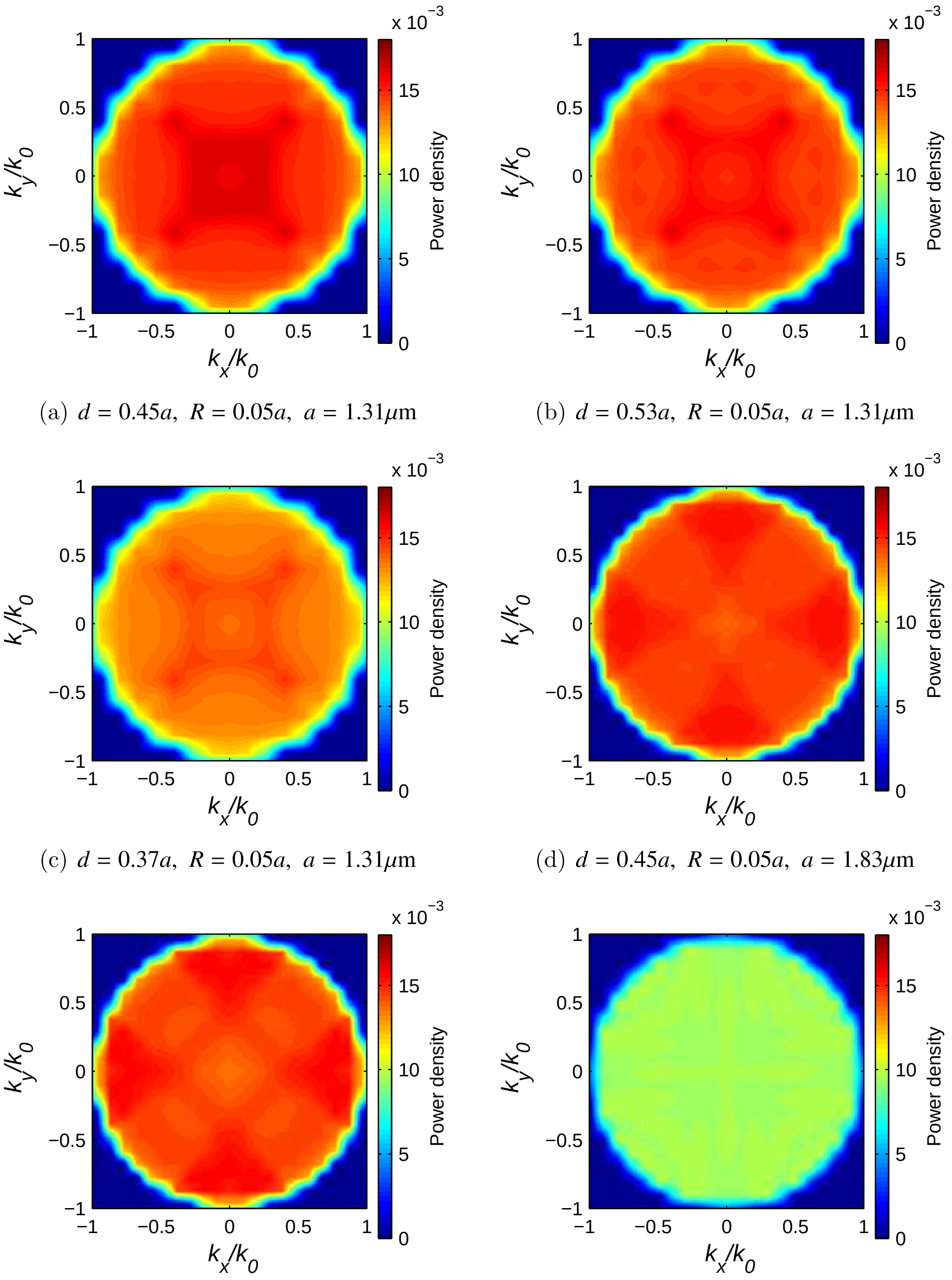
(c) $d = 0.53a, \ R = 0.4a, \ a = 1.37\mu$m

(d) $d = 0.53a, \ R = 0.45a, \ a = 1.3\mu$m

(e) $d = 0.53a, \ R = 0.45a, \ a = 1.97\mu$m

(f) $d = 0.05a, \ R = 0.55a, \ a = 1.76\mu$m

Figure 6.2.2: Radiation patterns for the hexagonal grating of holes at an electron energy of 0.7eV



(a) $d = 0.45a,\ R = 0.05a,\ a = 1.31\mu\text{m}$

(b) $d = 0.53a,\ R = 0.05a,\ a = 1.31\mu\text{m}$

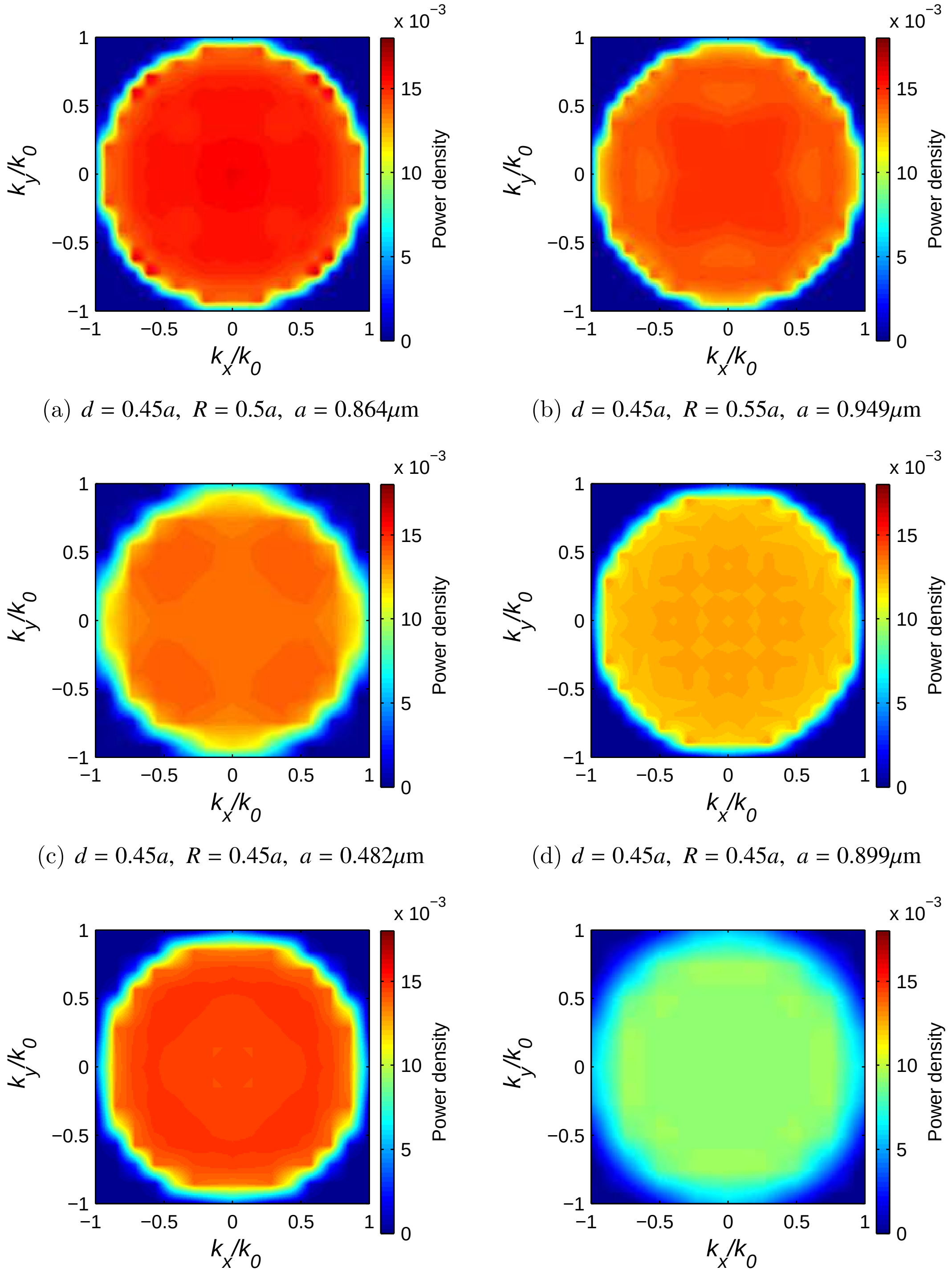
(c) $d = 0.37a,\ R = 0.05a,\ a = 1.31\mu\text{m}$

(d) $d = 0.45a,\ R = 0.05a,\ a = 1.83\mu\text{m}$

(e) $d = 0.53a,\ R = 0.05a,\ a = 1.83\mu\text{m}$

(f) $d = 0.05a,\ R = 0.05a,\ a = 1.76\mu\text{m}$

Figure 6.2.3: Radiation patterns for the square grating of isotropically etched holes at an electron energy of 0.7eV



(a) $d = 0.45a,\ R = 0.5a,\ a = 0.864\mu\text{m}$

(b) $d = 0.45a,\ R = 0.55a,\ a = 0.949\mu\text{m}$

(c) $d = 0.45a,\ R = 0.45a,\ a = 0.482\mu\text{m}$

(d) $d = 0.45a,\ R = 0.45a,\ a = 0.899\mu\text{m}$

(e) $d = 0.45a,\ R = 0.5a,\ a = 0.623\mu\text{m}$

(f) $d = 0.15a,\ R = 0.1a,\ a = 0.354\mu\text{m}$

Figure 6.2.4: Radiation patterns for the square grating of holes at an electron energy of 0.7eV Low frequency simulation

Chapter 7

Discussion

7.1 Improvement in surface transmission efficiency

The results clearly indicate that the transmission properties of the LED surface can be improved by etching out a 2D photonic crystal into the top layer. Although the specific geometry is highly influential, most of the simulated geometries seem to feature an improvement as opposed to a deterioration. Most of the results also indicate that η_s is improved more for the light at a photon energy of 0.65eV compared to that of 0.7eV. This is most probably because η_s for a plane surface, according to Table 3.4.1 is less for light at 0.65eV than for light at 0.7eV. This is in turn because light at 0.65eV has a greater amount of its power at large angles as shown in Figure 3.3.13. The grating with the largest spectral average improvement in η_s is the square grating of isotropically etched holes illustrated in Figure 6.1.9(a). The spectra in Figure 6.1.8 however indicate a very flat response, and other gratings have greater improvements for narrow spectral regions. Figure 6.1.2 shows an improvement around 70% for both photon energies for the square grating of holes with a lattice constant of approximately 1μm. According to Figure 6.1.8(a), also one of the hexagonal gratings of holes shows an improvement of 70% for light at 0.65eV. As shown in Figure 6.1.8(b), the improvement is however only arround 60% for the light at 0.7eV. This peak is however still competetive with the grating of isotropically etched holes for both photon energies.

The actual geometry settings, hole depth d and hole radius R, that give maximum η_s vary somewhat between the three different gratings. From Figures 6.1.1 and 6.1.4, it seems as if the grating depths for the square and hexagonal gratings need to only be large enough. Although the square grating has a local maximum, η_s does not decrease much with increasing depth. For both gratings it seems as if a depth greater than 0.3 gives close to maximum η_s. The hole radius is however much more critical in order to maximize η_s. As shown in Figure 6.1.3(a), the optimum radius for the square grating is so large that the grating really is composed of rods rather than holes. The optimum radius for the hexagonal grating is significantly smaller. One explanation might be that the hexagonal grating requires smaller holes to achieve the same fill factor, but it does not fully justify the difference. It should be noted that the results from the low frequency simulation of the square grating of holes shown in Figure 6.1.10 show a different dependence on the depth. This might be because the simulations cover gratings that are much deeper measured in wavelengths. It might also be because the lattice constant is significantly smaller than the wavelength. This can cause the light to see the grating as a homogeneous material with an average refractive index. The improvement might then be caused by the grating acting more as a quarter wave transformer than a perturbation that reduces total internal reflection. In this case an optimal depth should resemble a quarter wavelength. Although not fully in agreement, both the normal and low frequency simulations indicate that the square grating of holes with $(d, R) \approx (0.4, 0.5)$ has highest η_s around a grating constant of 1μm. This might be because these dimensions allow both a quarter wave transformer like behavior and still cause adequate diffraction to reduce total internal reflection. Such effects can however not be observed with the results from the hexagonal grating.

The dependence on d and R for the square grating of isotropically etched holes is quite different from the other two gratings. This should be expected as the parameters influence the physical structure quite differently as shown in Figure 4.3.3. The optimum geometry for this grating seems to be when R is as small as possible, and when the depth is such that the holes barely touch each other at the top. This can be seen in Figure 6.1.9. A striking feature of the spectra in Figure 6.1.8 is the very broadband improvement in η_s which this grating causes. Although it climbs slightly with increasing lattice constant, it is significantly more flat than the spectra for the other grating types. This might be because the grating partially acts as a tapered adaptation between the semiconductor and air.

7.2 Accuracy assessment

7.2.1 High resolution simulations

Without experimental measurements it is not possible to validate the results obtained by the simulations described in this thesis. Until such measurements are available the only method is to consider the realism and consistency of the results. For this purpose it is reasonable to turn to the high resolution simulations. The consistency between these simulations and those of the regular simulation series indicates whether the spatial resolution was sufficient. The spectra in Figures 6.1.13 and 6.1.14 show a very good consistency at low frequencies (small lattice constant) for the simulations with normal frequency. The consistency deteriorates with increasing frequency as should be expected from the discussion around the spatial resolution in section 4.5.3. The difference is however not too large, and never gets larger than 5%, indicating that the spatial resolution, to a reasonable degree, was sufficient for this kind of simulation. The comparison between high and low spatial resolution with low frequency shown in Figure 6.1.15 and 6.1.16 show somewhat worse consistency. In particular the region with a lattice constant larger than $0.9\mu m$ has a rather large difference. The rest of the spectrum is generally worse than the results with normal frequency, but the error is rarely above 5%.

7.2.2 Simulation of weak gratings

Another comparison that is worth bringing up is that between the η_s spectra which show a significant improvement and those of very weak gratings. Weak gratings meaning those that have a near to plane surface. One such spectrum is included for all full simulation series shown in Chapter 6. For the square and hexagonal gratings of holes Figures 6.1.2 and 6.1.5 show that these gratings have near no improvement in η_s. This is as expected and indicates that the simulations are able to replicate the result obtained for plane surfaces in section 3.4. It is however important to note that these results mainly confirm the behavior of light with a $\boldsymbol{k_{Ts}}$-vector that is within the light cone in air. The simulations can still introduce errors for light that has a $\boldsymbol{k_{Ts}}$-vector outside the light cone, as this light is only transmitted for strong gratings. It should also be noted that the grating of isotropically etched holes has a higher η_s for its weak grating than the other two. This might be because the effective hole radius is larger, making the grating stronger, or it might be because as explained in section 4.5.3, the ε-smoothing was turned off for this simulation series. If this is the case, then similar errors must be expected from the results from stronger gratings in this simulation series as well.

7.2.3 Oscillations in the surface transmission efficiency spectra

A noticeable feature of η_s-spectra from all the simulations with normal frequency is the oscillating tendency most clearly shown in Figure 6.1.8. Since the effect is consistent between different kinds of gratings which one would not expect to have equal properties, the oscillations are not believed to be real. The effect can however not be caused by the discretization of space since Figure 6.1.13 show very good consistency with the high resolution simulations. Another possible explanation could be interference effects with delayed light. Such light could be caused by for example unintentional reflections at the PML boundaries, or insufficient attenuation in the PML layers. Using MEEP units, the period of one oscillation in the spectra is typically 0.22 inverse time units which for a light velocity of unity corresponds to an increased optical path length of almost 8. Even taking into account the high index semiconductor, this is longer than any dimension achievable in the simulation cell, and is therefore not a very likely source for such an error. Although not presented here, the spectra for the square grating of holes with maximum hole size have been examined to investigate whether interference could be the cause. These grating holes cover almost the entire surface, and the different depths really mean a different semiconductor thickness. If interference was the cause the periodicity in the η_s spectra for different depths should be different since the effective optical path length most likely would be altered. These spectra generally showed small differences, but the periodicity of the oscillations was near to identical making interference seem a highly unlikely cause. A third cause could be the finite resolution in $\boldsymbol{k_T}$-space. Even for plane surfaces there are abrupt variations in the Fresnel equations which conceivably could distort the calculated total transmittance depending on how good the sampling points match the transmission peaks. Since the resolution is defined by a certain grid in $\boldsymbol{k_T}$-space, the actual propagation angles varies for different frequencies, and consequently the match should be expected to vary with frequency. The fact that the oscillations seem to be strongest for low frequencies also support this theory. This is the region where the least $\boldsymbol{k_T}$-points are within the light cone, and consequently the resolution in propagation angle is at its lowest. How well the simulated points match the transmission function should therefore be a greater issue for these frequencies compared to higher frequencies. To further investigate this possible explanation it would be necessary to look at the calculated coupling coefficients Γ from equation (4.1.11). These can be calculated from the simulation results, and are in fact intermediate results to the radiation patterns.

7.2.4 Radiation patterns

The primary reason to calculate radiation patterns was to get a distinct fingerprint of the simulations which later could be measured to verify the simulations for certain gratings. The limited subset of simulated radiation patterns that is shown in section are however surprisingly uniform which is not the best for such a task. Note that a flat radiation pattern corresponds to a $\cos(\theta)$-distribution for the power per unit solid angle. The radiation patterns can however be used to estimate certain aspects of the quality of the simulations. The fact that the radiation outside the circle of unity radius is very near zero means that there is not any significant light outside the light cone in air. This means that the distance from the surface to the recording plane was most probably long enough for evanescent fields to decay sufficiently. The radiation patterns from the hexagonal grating can also be used to assess the accuracy of the simulations. While the three mirror symmetry axis of the square gratings was imposed by the simulation method, only two mirror axis was imposed on the hexagonal grating. The three fold rotational symmetry was not utilized in the same way. It could therefore be used to assess the consistency of the results. Figure 6.2.2 shows six radiation patterns where five are from non weak gratings. All of these five except Figure 6.2.2(c) show a rather clear three fold rotational symmetry, and therefore indicate good consistency. The reason Figure 6.2.2(c) does not show such good symmetry properties is not known. One reason might be that the resolution in the radiation patterns is insufficient to resolve the needed details to see the symmetry. Note that the resolution in these gratings is very low as discussed in section 4.4.4. If this is the case, then it is also possible that the simulated η_s-value for this grating is faulty. Another reason might be the spatial discretization of $\boldsymbol{\varepsilon}(\boldsymbol{r})$. Taking this into account, the grating does not actually have any rotational symmetry properties. If this is the reason, it implies that the spatial resolution might have been too low.

7.3 Future work

7.3.1 Increasing efficiency

There is much more work which can be be performed with the light extraction method described here. The most obvious one being to further increase the light extraction. Although the simulations indicate an increase up to 70%, the actual value of η_s is still less than 5%, meaning there is potential for a much larger increase. One method which probably would work would be to etch the substrate very thin near the active LED area. Then the downwards propagating and reflected light can be reflected from the back side and have another attempt at being transmittet through the surface instead of being absorbed.

7.3.2 Detailed analysis of the results

In this thesis the results are presented in the form of η_s and a few radiation patterns. More insight into the properties of the gratings could probably be achieved with a more thorough study of the radiation patterns. They might also become a better basis for comparison against measurements if they were split up into two polarizations. This would be the case if the radiation patterns for each polarization separately has more variations than the combined patterns presented here. It should be noted that such a split must be between the polarizations of the transmitted light, not the two polarizations used for the source in the simulations.

Another way of gaining more insight into the mode of operation of the photonic crystal based on the achieved results is to directly examine the coupling coefficients Γ from equation (4.1.11). These have only been very lightly studied in the work this thesis is based on, and are not presented themselves. They can for example indicate the transmission of different plane waves, which for example can show the transmission of light that otherwise would be completely reflected from the surface. They can also be used for comparing the results from the simulations with high and normal spatial resolution to look for specific discrepancies. As mentioned in section 7.2.3 these might also give the reason for the observed oscillations in the η_s spectra in section 6.1.

7.3.3 Distribution of incident light

In Chapter 3 the angular distribution of the light incident on the photonic crystal surface was calculated. The lack of precise material parameters made these calculations difficult, and might very well be a significant source of error. Direct measurements of the optical properties of the materials in the LED could therefore improve the accuracy of the simulation results. If a new and more precise distribution function for the incident light were to be obtained, it could be utilized without running the simulations again. Only the post processing steps for calculating η_s and the radiation patterns would have to be repeated.

7.3.4 Measurements

Measuring the effects of a photonic crystal grating an actual LED must be regarded as highly desirable for this work. Measuring η_s is not directly possible, but the improvement in η_s can be found by measuring a LED with and one without a patterned surface. This can for example be done by using either an integrating sphere or preferably by measuring the light at different angles and constructing a radiation pattern. The latter would also allow comparisons with the simulated radiation patterns.

Chapter 8

Conclusion

The simulations presented in this thesis indicate that it is possible to increase the total light transmission through the surface of gallium antimonide LED by up to 70% by patterning the the surface with a 2D photonic crystal grating. Although there are concerns about the exact number of this increase, the consistency of the the results quite clearly indicate a notable improvement. The three simulated gratings types, square and hexagonal graitngs of holes and square grating of isotropically etched holes, are all able to achieve an improvement of approximately 60-70%. The square grating of isotropically etched holes stand somewhat out because the improvement seems to have low sensitivity to the lattice constant. Any value between 1 and 2.4μm can give approximately 60% improvement. The hole radius should be as small as possible and the hole depth and undercut should be such that the undercuts from neighbouring holes barily meet. The other two gratings show improvements up to 70%. The optimum hole depth, hole radius and lattice cosntant is, for the square grating of holes, 0.4, 0.5 and 1μm respectively. Here the depth and radius are fractions of the lattice constant. The configuration does according the simulations cause an increased transmission of approximately 70% for both the simulated photon energies, 0.65eV and 0.7eV. For the hexagonal grating, the optimum hole depth, hole radius and lattice cosntant are 0.53, 0.35 and 1.78μm respectively. The improvement is approximately 70% for the light with photon energy og 0.65eV, and 60% for the light with photon energy at 0.7eV. The accuracy of the results is not beilieved to be very good, but certain conclusions can be made. The optimum hole radius is, because of good consistency between many simulations, beilieved to be true. The optimum hole depth and lattice constant are less certain. It is however very likely that if the depth is chosen sufficiently large, the improvement will be very notable and near optimum if a corresponding optimum lattice constant can be found.

Bibliography

[1] Constantine A. Balanis. *Antenna Theory : Analysis And Design.* John Wiley & Sons, Inc., third edition, 2005.

[2] W. M. Becker, A. K. Ramdas, and H. Y. Fan. Energy band structure of gallium antimonide. *Journal of Applied Physics*, 32(10):2094–2102, 1961.

[3] Peter Bienstman. *Rigorous and efficient modelling of wavelength scale photonic components.* PhD thesis, Ghent University, 2001.

[4] David K. Cheng. *Field and Wave Electromagnetics.* Addison - Wesly Publishing Company, second edition, 1992.

[5] C. Henry Edwards and David E. Penny. *Calculus.* Prentice Hall, fifth edition, 1999.

[6] DAVID F. EDWARDS and GEORGE S. HAYNE. Optical properties of gallium antimonide. *J. Opt. Soc. Am.*, 49(4):414–415, 1959.

[7] Ardavan Farjadpour, David Roundy, Alejandro Rodriguez, Mihai Ibanescu, Peter Bermel, J. D. Joannopoulos, Steven G. Johnson, and Geoffrey Burr. Improving accuracy by subpixel smoothing in FDTD. *Optics Letters*, 31:2972–2974, October 2006.

[8] C Ghezzi, R Magnanini, A Parisini, B Rotelli, L Tarricone, A Bosacchi, and S Franchi. Concentration dependence of optical absorption in tellurium-doped gasb. *Semiconductor Science and Technology*, 12(7):858–866, 1997.

[9] K. Hjelt and T. Tuomi. Photoluminescence and electrical properties of movpe-grown zinc-doped gallium antimonide on gallium arsenide. *Journal of Crystal Growth*, 170(1-4):794 – 798, 1997.

[10] G.N. Iluridze, A.N. Titkov, and E.I. Chaikina. Intraband absorption in p-type gallium antimonide. *Soviet Physics - Semiconductors*, 21(1):48 – 9, 1987/01/.

[11] S. C. Jain, J. M. McGregor, and D. J. Roulston. Band-gap narrowing in novel iii-v semiconductors. *Journal of Applied Physics*, 68(7):3747–3749, 1990.

[12] John D. Joannopoulos, Steven G. Jghnson, Joshua N. Winn, and Robert D. Meade. *Molding the Flow of Light.* Princeton University Press, second edition, 2008.

[13] Richard Kelsy, William Clinger, and Jonathan Rees. Revised(5) report on the algorithmic language Scheme. Technical report, Massachusetts Institute of Technology, 2007.

[14] Erwin Kreyszig. *Advanced Engineering Mathematics.* John Wiley & Sons, Inc., eighth edition, 1999.

[15] P. C. Mathur and Sushil Jain. Electrical transport mechanism in the conduction band of gallium antimonide studied from hall-mobility and transverse-magnetoresistance measurements. *Phys. Rev. B*, 19(6):3159–3166, Mar 1979.

[16] MEEP Reference. http://ab-initio.mit.edu/wiki/index.php/Meep_Reference, November 2009.

[17] Ardavan F. Oskooi, David Roundy, Mihai Ibanescu, Peter Bermel, J. D. Joannopoulos, and Steven G. Johnson. MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method. *Computer Physics Communications*, 181:687–702, January 2010.

[18] Python reference. http://www.python.org/doc/, 2009.

[19] Tor A. ramstad. *Representing Information by Signals.* NTNU Trykk, 2003.

[20] Bahaa E. A. Saleh and Malvin Carl Teich. *Fundamentals of Photonics.* John Wiley & Sons, Inc., 1991.

[21] Jasprint Singh. *Optoelectronics An Introduction to Materials and Devices.* McGraw-Hill, 1996.

[22] G. Stollwerck, O.V. Sulima, and A.W. Bett. Characterization and simulation of gasb device-related properties. *Electron Devices, IEEE Transactions on*, 47(2):448–457, Feb 2000.

[23] Ben Streetman and Sanjay Banerjee. *Solid State Electronic Devices.* Prentice Hall, sixth edition, 2005.

[24] A.N. Titkov, E.I. Chaikina, E.M. Komova, and N.G. Ermakova. Low-temperature luminescence of degenerate p-type crystals of direct-gap semiconductors. *Soviet Physics - Semiconductors*, 15(2):198 – 202, 1981/02/.

[25] Oldwig von Roos. Effect of reabsorbed recombination radiation on the diffusion length of minority carriers in wide-band-gap semiconductors. *Journal of Applied Physics*, 54(5):2495–2498, 1983.

[26] J. D. Wiley, R. K. Willardson, and A. C. Beer. *Semiconductors and Semimetals*, volume 10. Academic Press, 1975.

Appendix A

Control file

Listing A.1: Control file for square grating of cylindrical holes

```
;Transversale k-vektorer er 2pi-korrigerte. Behandles som f, ikke w. Som i meep
;Transversal geometri-lengde er 1. Resiprok gittervektor Kx er derfor lik 1 (transversal romlig frekvens)
(define Kx 1)
(define Ky 1)
(define-param ref? false) ;if true, the entire geometry will be filled with semiconcuctor

(define-param dPMLTop 0.4)
(define-param dTransPML 0.2)
(define-param dSurfTrans 0.6)
(define-param dSourceSurf 0.6)
(define-param dReflSource 0.1)
(define-param dPMLRefl 0.05)
(define-param dPMLBot 0.2)

(define-param holeDepth 0.3)
(define-param holeR 0.35)

(define geomHeight (+ dPMLBot dPMLRefl dReflSource dSourceSurf dSurfTrans dTransPML dPMLTop))
(define bot (/ geomHeight -2)) ; Bottom z-coordinate
(define top (/ geomHeight 2)) ; Top z-coordinate
(define dSemiConductor (+ dPMLBot dPMLRefl dReflSource dSourceSurf))

(define cRefl (+ bot dPMLBot dPMLRefl))
(define cSource (+ cRefl dReflSource))
(define cDitch (+ cSource dSourceSurf)) ;Ditch center is on semiconductor surface. The height must therefore be doubled.
(define cTrans (+ cDitch dSurfTrans))
(define cSemiConductor (+ bot (/ dSemiConductor 2)))

(define-param sourceCutoff 5)
(define-param stopTh 1e-4)
(define-param TPMLSt 6)
(define-param BPMLSt 1)
(define-param outputDir ”results”)
(define-param filePrefix ”standard”)
(define-param refrIndex 3.8)
(define-param Fs 2.5); time sampeling frequency
(define-param Fcenter 0.85) ; Center frequency
(define-param Fwidth 1.8) ; Frequency width
```

```
(define-param animer? false); Output data for animation or not
(define-param res 30)
(define-param TE? true)

(define-param kx 0) ;transversal k with noe added reciprocal lattice vectors. This is done by the script
(define-param ky 0)

(define Rxi 0)
(define Ryi 0)

(define kxSource (+ kx (* Rxi Kx) ))
(define kySource (+ ky (* Ryi Ky) ))

(define Fmax (+ Fcenter (/ Fwidth 4))) ;Fmax is Fs+Fwidth/4 (also equal to k0, max)
(define Nx (inexact->exact (floor (* refrIndex (/ Fmax Kx)))))
(define Ny (inexact->exact (floor (* refrIndex (/ Fmax Ky))))); Maximum number of added reciprocal lattice vectors (zero means one simulation, no adding)


(if (not (access? outputDir F_OK))
    (mkdir outputDir)
)

(define outputFile (open-output-file (string-append outputDir "/" filePrefix ".out")))

(define (sourcePhaseFunction P)
   (make-polar 1 (+ (* 2 pi (vector3-x P) kxSource ) (* 2 pi (vector3-y P) kySource )) )
)

(define (outputVar string var)
    (display (string-append ":Var Rxi_" (number->string Rxi) " Ryi_" (number->string Ryi) " " string " = " ) outputFile)
    (display var outputFile)
    (display "\n" outputFile)
)

(define (seqList length) ; Procedure that makes a list with increasing numbers (0 1 2 3 4 ... length-1)
    (define count 0)
    (define v (make-list length 0))
    (for-each (lambda (x) (begin (list-set! v count count) (set! count (+ count 1)) ) ) v )
    v
)

(define (SqkTotSource)
    (+ (* kxSource kxSource) (* kySource kySource))
)

(define (defSource)
    (define AmX 0.0)
    (define AmY 1.0)
    (define ADenom (sqrt (+ (* kxSource kxSource) (* kySource kySource))) )
    (if (not (= ADenom 0.0))
        (begin
         (set! AmX  (/ (* -1.0 kySource) ADenom)  )
         (set! AmY  (/ (*  1.0 kxSource) ADenom)  )
        )
     )

    (list
        (make source
            (src (make gaussian-src (frequency Fcenter) (fwidth Fwidth) (cutoff sourceCutoff)))
            (component (if TE? Ey Hy))
            (center 0 0 cSource)
            (size 1 1 0)
            (amplitude AmY)
            (amp-func sourcePhaseFunction )
        )
        (make source
            (src (make gaussian-src (frequency Fcenter) (fwidth Fwidth) (cutoff sourceCutoff)))
            (component (if TE? Ex Hx))
            (center 0 0 cSource)
            (size 1 1 0)
            (amplitude AmX)
            (amp-func sourcePhaseFunction )
        )
    )
)

(define semiConductorSlab
    (make block
        (center 0 0 (- cSemiConductor 1)) ; moves the semiconductor 1 down and extends it by 2. The only effect is that it is extended by 2 outside the cell. This avoids unwanted subpixel averaging
        (size 1.1 1.1 (+ dSemiConductor 2 ))
        (material (make dielectric (epsilon (* refrIndex refrIndex))) )
    )
)

(define hole
    (make cylinder
        (center 0 0 cDitch)
        (radius holeR)
        (height (* 2 holeDepth))
        ;(size ditchWidth infinity (* 2 ditchDepth) )
        (material air)
    )
)

(define allSemiconductor
    (make block
        (center 0 0 0)
        (size 1.1 1.1 (+ geomHeight 2 ))
        (material (make dielectric (epsilon (* refrIndex refrIndex))) )
    )
)


(use-output-directory outputDir)
(set! output-single-precision? true)
(set! ensure-periodicity false)
(set! geometry-lattice (make lattice (size 1 1 geomHeight)))
(set! force-complex-fields? true)
(set! geometry (if ref? (list allSemiconductor) (list semiConductorSlab hole)) )
(set! resolution res)
(set! sources (defSource))
(set! pml-layers
    (list
        (make pml (direction Z) (side Low)  (thickness dPMLBot) (strength BPMLSt))
        (make pml (direction Z) (side High) (thickness dPMLTop) (strength TPMLSt))
    )
```

```
)
(set! k-point (vector3 kx ky 0))

;Calculating real Fs (compensates for rounding error)
(define dN (round (/ resolution (* Courant Fs)) ));Timesteps between sampeling
(define Fsr (/ resolution (* Courant dN)))


(define (runFunction) ; "Rxi og Ryi" beskriver hvor mange resiproke gitterveektorer som skal legges til kx og ky
    (restart-fields)

    (set! kxSource (+ kx (* Rxi Kx) ))
    (set! kySource (+ ky (* Ryi Ky) ))

    (change-sources! (defSource))


    (outputVar "Kx" Kx)
    (outputVar "Ky" Ky)
    (outputVar "Rxi" Rxi)
    (outputVar "Ryi" Ryi)
    (outputVar "dPMLTop" dPMLTop)
    (outputVar "dTransPML" dTransPML)
    (outputVar "dSurfTrans" dSurfTrans)
    (outputVar "dSourceSurf" dSourceSurf)
    (outputVar "dReflSource" dReflSource)
    (outputVar "dPMLRefl" dPMLRefl)
    (outputVar "dPMLBot" dPMLBot)
    (outputVar "holeDepth" holeDepth)
    (outputVar "holeR" holeR)
    (outputVar "geomHeight" geomHeight)
    (outputVar "bot" bot)
    (outputVar "top" top)
    (outputVar "dSemiConductor" dSemiConductor)
    (outputVar "cRefl" cRefl)
    (outputVar "cSource" cSource)
    (outputVar "cDitch" cDitch)
    (outputVar "cTrans" cTrans)
    (outputVar "cSemiConductor" cSemiConductor)
    (outputVar "Nx" Nx)
    (outputVar "Ny" Ny)
    (outputVar "kx" kx)
    (outputVar "ky" ky)
    (outputVar "n" refrIndex)
    (outputVar "Fs" Fsr)
    (outputVar "FsInput" Fs)
    (outputVar "Courant" Courant)
    (outputVar "dN" dN)
    (outputVar "Fcenter" Fcenter)
    (outputVar "Fwidth" Fwidth)
    (outputVar "animer" animer?)
    (outputVar "res" res)
    (outputVar "TE" TE?)
    (outputVar "kxSource" kxSource)
    (outputVar "kySource" kySource)
    (outputVar "k_point" k-point)
    (outputVar "resolution" resolution)
    (outputVar "filePrefix" filePrefix)
    (outputVar "ref" ref?)
    (outputVar "outputDir" outputDir)
    (outputVar "eps_averaging" eps-averaging?)
    (outputVar "BPMLSt" BPMLSt)
    (outputVar "TPMLSt" TPMLSt)
    (outputVar "stopTh" stopTh)
    (outputVar "sourceCutoff" sourceCutoff)


    (run-sources+
        (stop-when-fields-decayed 5 Sz (vector3 0 0 cTrans) stopTh)
        (at-beginning output-epsilon)
        (to-appended (string-append filePrefix " T Rxi_" (number->string Rxi) " Ryi_" (number->string Ryi))
            (at-every (/ 1 Fs)
                (in-volume
                    (volume (center 0 0 cTrans) (size 1 1 0))
                    output-efield output-hfield
                )
            )
        )
        (if animer?
            (to-appended (string-append filePrefix " Animation  Rxi_" (number->string Rxi) " Ryi_" (number->string Ryi))
                (at-every (/ 1 Fs)
                    (in-volume (volume (center 0 0 0) (size 1 0 geomHeight))
                        (if TE? output-efield-y output-hfield-y)
                    )
                )
            )
            (to-appended "nothing"
                (at-every infinity
                    (in-volume (volume (center 0 0 0) (size 0 0 0))
                        (if TE? output-efield-y output-hfield-y)
                    )
                )
            )
        )
    )

)

(define (testFunction)
    (set! kxSource (+ kx (* Rxi Kx) ))
    (set! kySource (+ ky (* Ryi Ky) ))
    (display (string-append "kxSource: " (number->string kxSource) "   kySource: " (number->string kySource) ))

    (if (and (<= kySource kxSource) (<= (SqkTotSource) (* refrIndex Fmax refrIndex Fmax)) )
        (begin
            (display "   GO!   ")
            (display (SqkTotSource))
            (display "\n")
            (runFunction)
        )
        (begin
            (display "   Skip  \n")
            (display (SqkTotSource))
        )
    )

)

(define (runLoop)
    (define RxiList (seqList (+ Nx 1)))
    (define RyiList (seqList (+ Ny 1)))
    (for-each
        (lambda (RxiElement)
            (set! Rxi RxiElement)
            (for-each
                (lambda (RyiElement)
                    (set! Ryi RyiElement)
                    (testFunction)
                )
                RyiList
            )
        )
        RxiList
    )
)



(runLoop)

(close-output-port outputFile)
(quit)
```

Appendix B

Surface transmission efficiency

Listing B.1: Step 1: Reorganizing power spectra

```
function extractPower(folder)

try
    load([folder '/xi_1_yi_1.mat']); clear data;
    Pdata.epsilon = epsilon;
catch
    Pdata.epsilon = [];
end


%if NBy == 12 % This is a hack. remove for general purpouse
%    NBy = 6;
%end

try
    NBx = length(dir( [folder '/xi_*_yi_1_Pf.mat' ]));
    NBy = length(dir( [folder '/xi_1_yi_*_Pf.mat' ]));
    load([folder '/xi_1_yi_1_Pf.mat']); testData = PfData;
    postFix = '_Pf';
catch
    NBx = length(dir( [folder '/xi_*_yi_1.mat' ]));
    NBy = length(dir( [folder '/xi_1_yi_*.mat' ]));
    load([folder '/xi_1_yi_1.mat']); testData = data;
    postFix = '';
end


[Nx, Ny] = size(testData); %Maximum number of added reciprocal lattice vectors for x and y
%Nx = max(Nx, Ny);
%Ny = max(Nx, Ny);

pointsX = Nx * NBx; %The total number of simulated plane waves
pointsY = Ny * NBy; %

nF = 101;

fstart = testData(1,1).params.Fcenter - testData(1,1).params.Fwidth/4;
fstop = testData(1,1).params.Fcenter + testData(1,1).params.Fwidth/4;
Pdata.f = (0:(nF-1))/(nF-1)*(fstop - fstart) + fstart;

Pdata.P = NaN(pointsX, pointsY, nF);
Pdata.kxSource = NaN(pointsX, pointsY);
Pdata.kySource = NaN(pointsX, pointsY);

for bx = 1:NBx
    for by = 1:NBy

        filename = sprintf('%s/xi_%i_yi_%i%s.mat', folder, bx, by, postFix);
        load(filename);
        try
            BrilPoint = PfData;
        catch
            BrilPoint = data;
        end
        for px = 1:Nx
            for py = 1:Ny

                %This unwraps the diagonal symmetry line. Must be removed
                %for hexagonal gratings (Only valid for square gratings!!!)
                %if(py > px)
                %     pxS = py;
                %     pyS = px;
                %else
                    pxS = px;
                    pyS = py;
                %end

                if(prod(double((size(BrilPoint) >= [pxS pyS])))   && BrilPoint(pxS, pyS).params.exists)

                    if ~isequal(BrilPoint(pxS, pyS).f, Pdata.f)
                        %warning('Non equal frequency array, expected length: %i, actual length: %i, Interpolating..', length(Pdata.f), length(BrilPoint(pxS, pyS).f));
                        BrilPoint(pxS, pyS).Pf = interp1(BrilPoint(pxS, pyS).f, BrilPoint(pxS, pyS).Pf ,Pdata.f);
                    end
                    Pdata.P( bx + (px-1)*NBx , by + (py-1)*NBy , : ) = BrilPoint(pxS, pyS).Pf;
                    Pdata.kxSource( bx + (px-1)*NBx , by + (py-1)*NBy) = BrilPoint(pxS, pyS).params.kxSource;
                    Pdata.kySource( bx + (px-1)*NBx , by + (py-1)*NBy) = BrilPoint(pxS, pyS).params.kySource;
                end
            end
        end

    end
end

Pdata.params = testData(1,1).params;
save([folder '/PowerData.mat'],  'Pdata' )
```

Listing B.2: Step 2: extractEtaS

```
function file = extractEtaS(folder, IncRadFile)


TEfolders = dir([folder '/Surf*TE']);
%TMfolders = dir([folder '/Surf*TM']);
RefTE = load( [folder '/Ref_TE/PowerData' ] );
RefTM = load( [folder '/Ref_TM/PowerData' ] );

nf = length(RefTE.Pdata.f);
nsims = length(TEfolders);
d = zeros(1,nsims);
R = zeros(1,nsims);

etaS = zeros(nf, nsims);

clear PrTE
clear PrTM
%For each geometry (folder)
for p = 1:nsims
    p
    TED = load([folder '/' TEfolders(p).name '/PowerData']);
    TMD = load([folder '/' strrep(TEfolders(p).name, '_TE', '_TM') '/PowerData' ]);
    TED.Pdata

    if isnan(TED.Pdata.P(1,2,1))
        disp('Unwrapping diagonal symmetry')
        TED.Pdata = unwrapDiagSymP(TED.Pdata);
        TMD.Pdata = unwrapDiagSymP(TMD.Pdata);
    else
        disp('There is no diagonal symmetry')
    end

    %Calculating reference power
    if (~exist('PrTE', 'var') || ~exist('PrTM', 'var'))
        PrTE = artificialRef(squeeze(RefTE.Pdata.P(1,1,:)), RefTE.Pdata.f, TED.Pdata.kxSource, TED.Pdata.kySource);
        PrTM = artificialRef(squeeze(RefTM.Pdata.P(1,1,:)), RefTM.Pdata.f, TMD.Pdata.kxSource, TMD.Pdata.kySource);
    end
    TED.Pdata.T = TED.Pdata.P ./ PrTE;
    TMD.Pdata.T = TMD.Pdata.P ./ PrTM;

    if (length(TED.Pdata.f) ~=nf) | (length(TMD.Pdata.f) ~=nf)
        error(sprintf('Wrong length of frequency-array'))
    end

    %for each frequency
    for pf = 1:nf
        etaS_TE = calcEtaS(TED.Pdata.T(:,:,pf), TED.Pdata.f(pf), TED.Pdata.kxSource, TED.Pdata.kySource, IncRadFile);
        etaS_TM = calcEtaS(TMD.Pdata.T(:,:,pf), TMD.Pdata.f(pf), TMD.Pdata.kxSource, TMD.Pdata.kySource, IncRadFile);
        etaS(pf, p) = 0.5* (etaS_TE + etaS_TM);

    end

    d(p) = TED.Pdata.params.holeDepth;
    R(p) = TED.Pdata.params.holeR;

end

%sorting a d-R matrix f is 1st dim, d is second, R is third
```

```
[dax, a, nd] = unique(d);
[Rax, a, nR] = unique(R);
d= dax;
R = Rax;

etaSMtx = zeros(nf, length(dax), length(Rax));
for p = 1:nsims
    etaSMtx(:,nd(p), nR(p)) = etaS(:,p);
end


%deciding where to save the results
if ~isempty(findstr(IncRadFile, '07'))
    elEn = 0.7;
    file = [folder '/etaS07'];
elseif ~isempty(findstr(IncRadFile, '65'))
    elEn = 0.65;
    file = [folder '/etaS065'];
else
    file = 'bacup';
    error('result is not saved!')
end

save(file, 'etaSMtx', 'etaS', 'd', 'R', 'elEn');
```

Listing B.3: Step 2: calcEtaS

```
%Calculates etaS for ONE frequency
function etaS = calcEtaS(P, f, kx, ky, radPattDataFile)

res = 400;
n = 3.8;

k = n*f;

kxAx = (0:res-1)/(res-1)*k;
kyAx = (0:res-1)/(res-1)*k;


kx(isnan(kx)) = -1;
ky(isnan(ky)) = -1;
P(isnan(P)) = 0;

%Interpolating transmission coefficients
T = griddata(kx, ky, P, kxAx, kyAx');

global IncRad
if isempty(IncRad) || ~strcmp(IncRad.file, radPattDataFile)
    disp('Loading File');
    IncRad = load(radPattDataFile);
    IncRad.file = radPattDataFile;
end

[kxM, kyM] = meshgrid(kxAx, kyAx);

ktM = sqrt(kxM.^2 + kyM.^2);
ktMVect = reshape(ktM, size(ktM,1)*size(ktM,2), 1);

%Interpolating incident power
PincMVect = interp1(k*IncRad.kt, IncRad.radkt/k^2, ktMVect);
PincM = reshape(PincMVect, res, res);

T(isnan(T)) = 0;
PincM(isnan(PincM)) = 0;

%Integrating. Multiplying by 4 because only one quadrant is used
etaS = 4*trapezint2D(kxAx, kyAx, T.*PincM);
```

Listing B.4: Step 2: trapezint2D

```
function res = trapezint2D(x, y, Z)

temp = zeros(1, length(y));

for p = 1:length(y)
    temp(p) = trapezint(x, squeeze(Z(:,p)).');
end

res = trapezint(y, temp);
```

Listing B.5: Step 2: artificialRef

```
%kx, and ky are matrices, fvec  and PnormalInc are vectors,
%Part is a 3d structure with respective axis (kx, ky, f)
function Part = artificialRef(PnormalInc, fvec, kx, ky)

n = 3.8;
kvec = n*fvec;

[a, a, k] = ndgrid(1:size(kx,1), 1:size(ky,2),  kvec);
clear a
[a, a, P] = ndgrid(1:size(kx,1), 1:size(ky,2),  PnormalInc);
clear a

%Creating anf extruding 3d kt-structure
kt = sqrt(kx.^2 + ky.^2);
ktV = reshape(kt, [numel(kt), 1]);
ktextV = ndgrid(kt, kvec);
kt = reshape(ktextV, [size(kt), length(kvec)]);

Part = P .* k./ sqrt(k.^2 - kt.^2);
Part(imag(Part)~=0) = NaN;
```