Appendix

A Manual of the Computer Program

The computer program used to simulate the model described in Section 3.2 is a useful tool to identify and quantise essential characteristics of cavities. We implemented a graphical user interface to provide a convenient access to the program's abilities. It enables to simulate cavities for various design parameters in a user-friendly and intuitively comprehensible environment that makes the program a helpful tool for further projects. In the following sections, the abilities and the manual is described in more detail.

Introduction

The provided computer program simulates a planar cavity. The Transfer Matrix Method is used to calculate the intensity and phase difference after one roundtrip (RT), i.e. a plane wave starts between two mirrors and get reflected once at every mirror (incident perpendicular to the surface) [2]. After that the wave returns to the starting point. They are two conditions which are important for lasing.

$$\left|\frac{E_{\mathrm{RT}}}{E_{\mathrm{in}}}\right|^2 \stackrel{!}{=} 1 \tag{4.38}$$

$$\mathrm{Arg}\,(E_{\mathrm{in}}) - \mathrm{Arg}\,(E_{\mathrm{RT}}) \mod 2\pi \stackrel{!}{=} 0 \tag{4.39}$$

Where E_{in} denotes the field amplitude at the starting time and E_{RT} denotes the field amplitude after one RT.
More details can be found in Section 2.3.

Libraries and Starting

The program is written in Python version 2.7 and should be started with this interpreter. To start the script, first go to the folder containing `metalbraggmirror.py` , and then run it with the interpreter. In a Linux environment, this can be achieved with

```
1   cd 'Path to folder which contains metalbraggmirror.py '
2  python2.7 metalbraggmirror.py
```

In addition to Python 2.7, the following libraries must be installed. Table 4.5 summarises all the libraries, their installation command for Ubuntu and the developer page. Make sure `pip` is installed. If not, install it with following command.

```
1   sudo apt-get install python-pip
```

Table 4.5: All libraries needed in order to run the script. In addition the installing command in Ubuntu is listed, as well as the developer page for more information.

Library	Installation Command	Developer's Page
NumPy	`python2.7 -m pip install -users numpy`	`numpy.org`
PyQt4	`sudo apt-get install python-qt4`	`riverbankcomputing.com`
Matplotlib	`python2.7 -m pip install -users matplotlib`	`matplotlib.com`

The Graphical User Interface

In Figure 4.14 the user interface for the parameter is displayed. The following enumeration gives an overview and a short explanation of the functionality of the single parts.

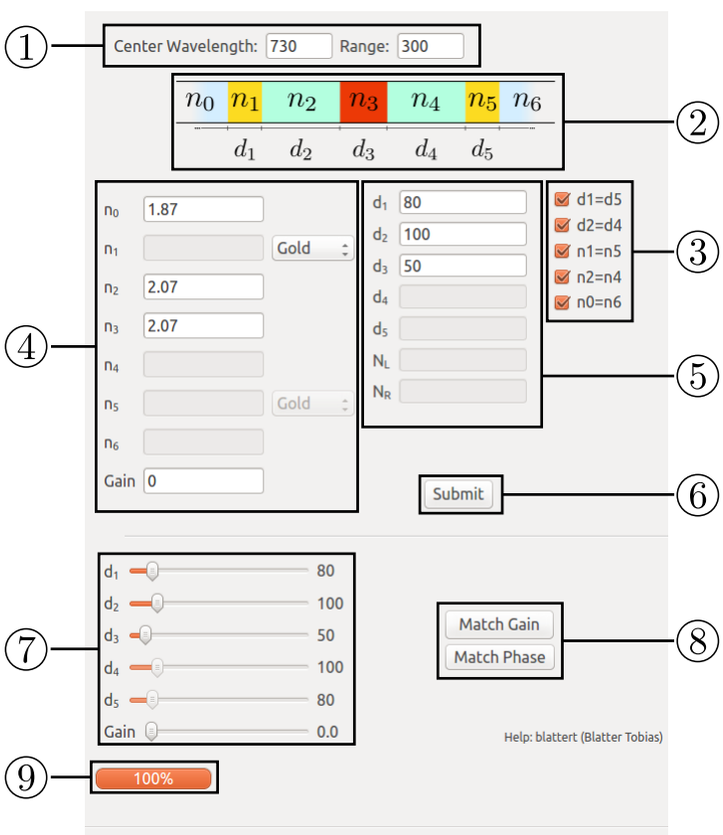

Figure 4.13: Overview of the user interface.

① Defines the set of wavelengths which are going to simulate. Precisely, the set is $\Lambda = [\lambda_c - R, \lambda_c + R]$ with $2R$ evenly spaced samples. Where λ_c is the `Center Wavelength` and R the `Range`. The unit is nm. As long as no wavelength depended material is chosen (see ④), the unit $m := [\lambda_c]$ can be arbitrary. In consequence the units of the thicknesses d and gain g change, in particular $[d] = m$ and $[g] = 1/m$. The choice of λ_c will affect the results of the optimisation routines (see ⑧).

② A sketch of the cavity which is going to be simulated. The refractive indices

of each layer are denoted with n and the length of each layer is denoted with d. The values can be manipulated with the boxes below (see ③,④,⑤).

③ Symmetry options: If a check box is checked the statement on the right will be respected during the simulation. For example, if $n_1 = n_5$ is checked the refractive index of layer 1 and 5 are equal.

④ With the first lines one can set the refractive index of a specific layer to a certain value. The last line `Gain` allows to submit a gain parameter. This value is 10^6 times higher then the actual value for the simulation and the unit is nm^{-1} or likewise $[\lambda_c]^{-1}$. Some lines are inactivated if a particular option is chosen (see ③). With the two combo boxes one can choose from different options for the mirrors. They are freely combinable:

(a) `Gold`: A gold mirror with wavelength depended refractive index of gold. Values from Babar and Weaver [5].

(b) `Silver`: A silver mirror with wavelength depended refractive index of silver. Values from Babar and Weaver [5].

(c) `Bragg`: Allows to set up a Bragg mirror. Two refractive indices (second line would appear) for the slices of one single Bragg module.

(d) `Custom`: Wavelength independent refractive index like for example n_0

It is possible to specify complex refractive index. For this the format $a + bi$ $(a, b \in \mathbb{R})$ is to be used.

⑤ In the first six lines the lengths of the layers can be specified. The unit is nm or likewise $[\lambda_c]$. The parameters $\mathrm{N_L}$ and $\mathrm{N_R}$ determine the number of single Bragg elements of the left (L) and right (R) Bragg mirror. They are only activated if Bragg mirror is chosen. The lengths of the two single Bragg modules layer, are automatically set to $d_{1,2} = \lambda_c/4\,\mathrm{Re}(n_{1,2})$.

⑥ Starts the simulation with the values specified.

⑦ Sliders can be used to manipulate the length of the layer and to change the gain. The above values are not changed, so the simulation can be started at any time with the original values, by pressing the `Submit` button.

⑧ The `Match Gain` button starts an optimisation routine to find the minimal gain to fulfil the intensity condition (Equation 4.38) for the wavelength λ_c. The `Match Phase` button starts an optimisation routine to find the nearest value for d_2 in order to satisfy the phase condition (Equation 4.39) for λ_c. The result is rounded to an integer. The length d_4 will be changed only if the option $d_2 = d_4$ is checked. Both results are displayed as label of the slider. The optimisation routine uses the Secant Method [36].

⑨ A feedback for the user that the simulation is running. The percentage indicates how many wavelengths have already been simulated.