Echelle Layout Program 2

Marc Schneider, 2021 - 2023

marc.schneider@kit.edu

When designing integrated wavelength division multiplexing systems using photonic chips, different methods for wavelength filtering are available. One of them are planar concave gratings or Echelle gratings. For such Echelle gratings, different construction methods exist, like Rowland Circle method and Two Stigmatic Point (TSP) method.

This MATLAB app implements my version of the TSP method to calculate arbitrary Echelle gratings with Bragg grating reflectors. From this, a simulation project for COMSOL Multiphysics can be generated to simulate the Echelle grating in 2D, as well as a GDSII layout for using the grating in a chip design. Additionally, a sample file for Synopsys OptoDesigner is generated.

To run the program to its full extent, you still need

- GDSII Toolbox v1.41 from Ulf Griesmann (https://github.com/ulfgri/gdsii-toolbox)
- COMSOL Multiphysics (installed with MATLAB support)

To speed up calculations, it is preferred to install the

- Parallel Computing Toolbox (MATLAB)

The file “EffectiveRefractiveIndices_Wavelength1480-1630nm_SiliconThickness205-225nm+245-255nm.xlsx” contains the effective refractive indices of the layer stack used for the Echelle grating. The given file contains calculated effective refractive indices for a layer stack of silicon dioxide - silicon - silicon dioxide for wavelengths between 1480nm and 1630nm and for silicon thicknesses between 205nm and 225nm as well as between 245nm and 255nm in 1nm steps. The data is used for the Material Chooser. Extend the file to your own needs.

The main program is "MarcEchelle.mlapp" and runs in MATLABs App Designer. Or at least it should, if all required parts are included in MATLABs search path.

'MarcEchelle_exported.m' and 'MarcEchelle_ITUChooser_exported.m' are the exported versions of the respective .mlapp files for better readability and easier tracking of changes.

Documentation

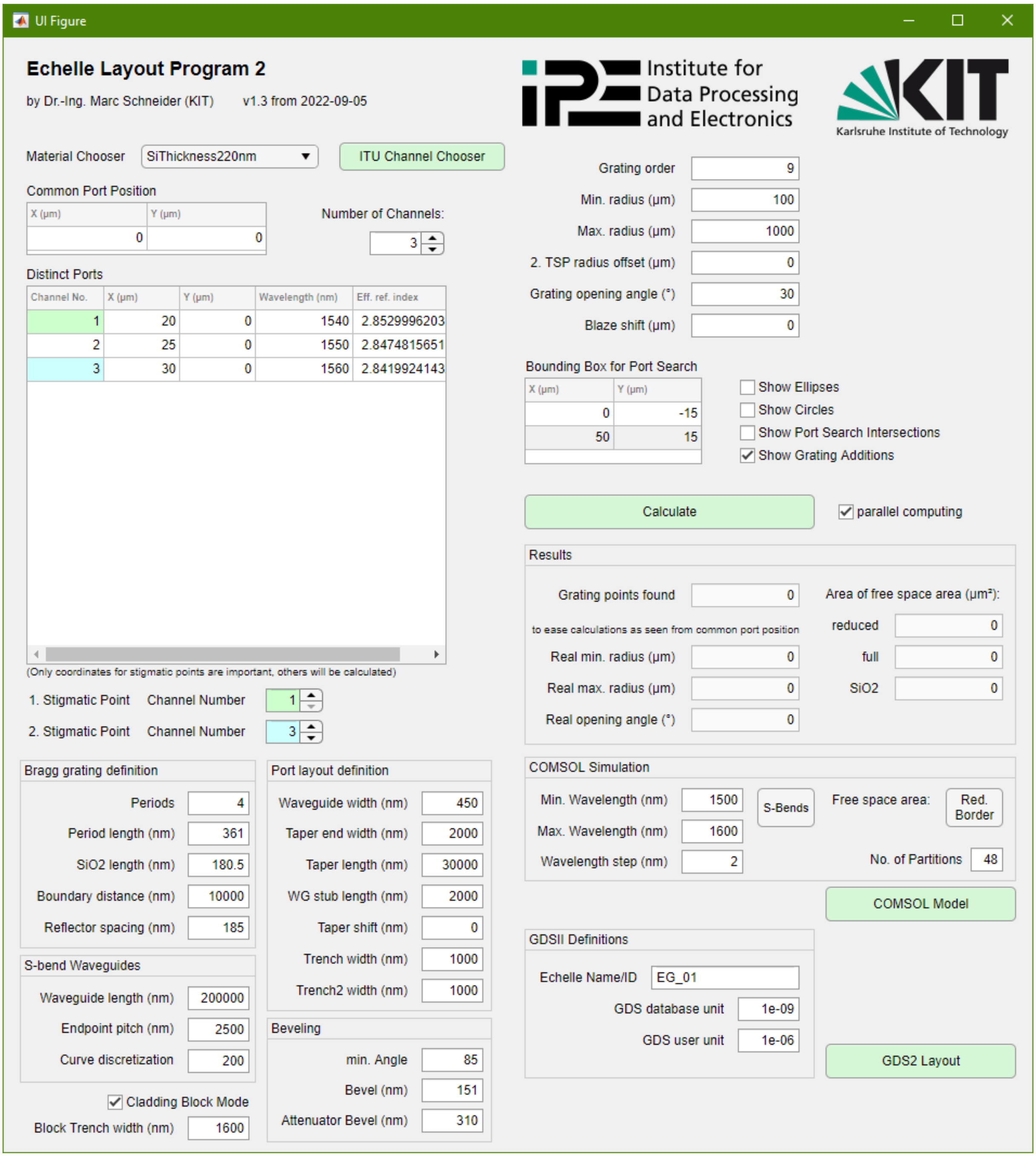
The “Echelle Layout Program 2” is provided as (raw) MATLAB™ application. So, you’ll need MATLAB to run it. (I’m still using MATLAB R2019b.)

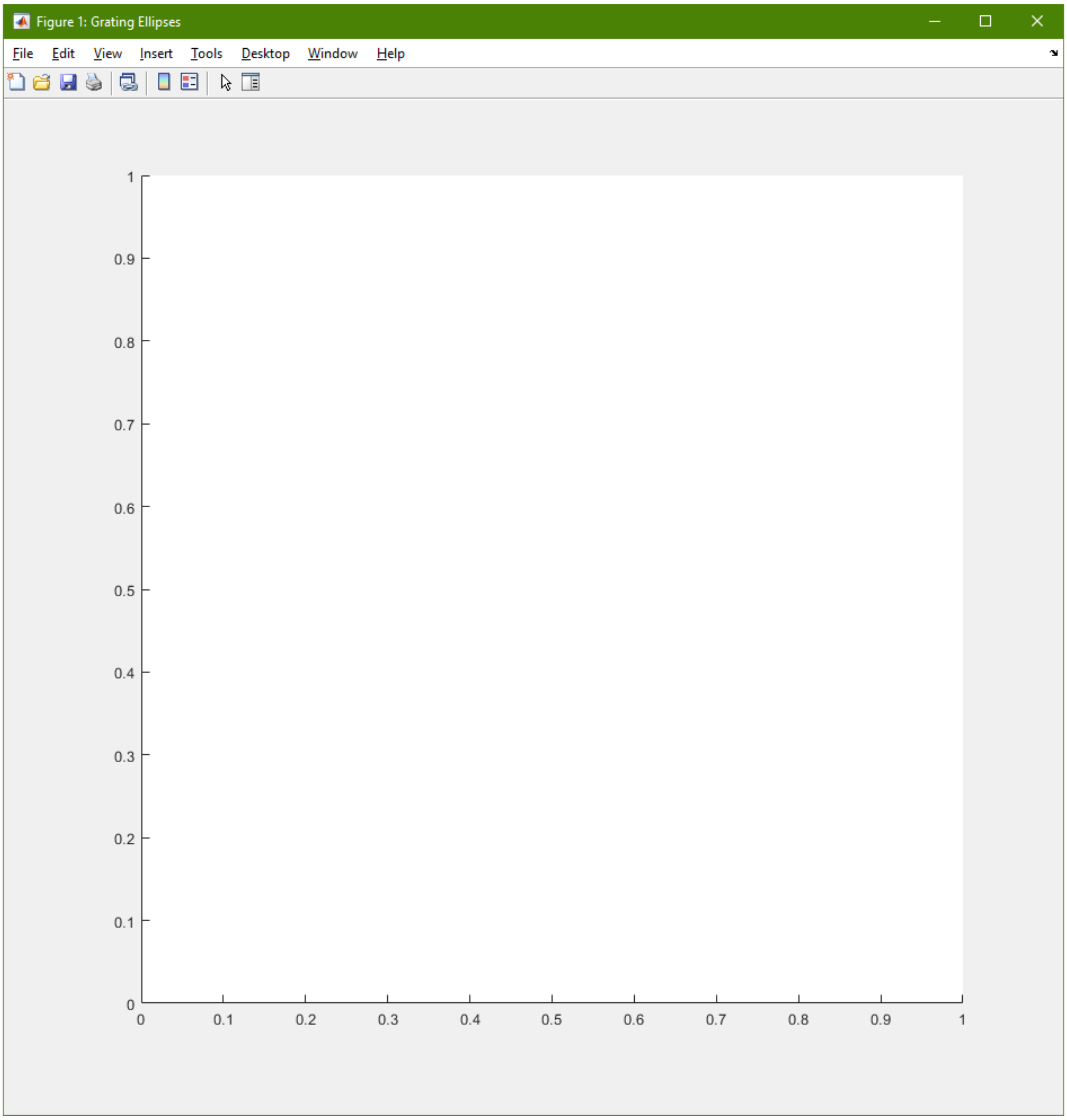
For export of the calculated Echelle grating to GDSII, you’ll also need to install the GDSII Toolbox from Ulf Griesmann (https://github.com/ulfgri/gdsii-toolbox). I used version 1.41, but maybe other versions work, too. Don’t forget to set the path in MATLAB.

To export the calculated Echelle grating to a COMSOL Multiphysics™ project for simulation, you will need COMSOL Multiphysics and have to start MATLAB through the link “COMSOL Multiphysics with MATLAB”, which should have been generated when installing COMSOL. This is required, because COMSOL model libraries are needed to generate the project and it seems that those are only be found in this way.

To start the program, navigate to the respective folder and open “MarcEchelle.mlapp”. The AppDesigner should come up and show you the user interface. Then run the application.

Two windows will appear, the UI and a Figure. The latter one shows (later on) the generated Echelle grating.

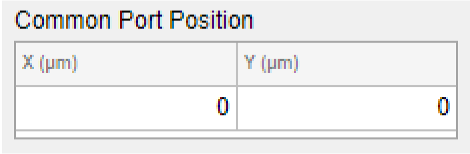

UI Figure
Echelle Layout Program 2
by Dr.-Ing. Marc Schneider (KIT)
v1.3 from 2022-09-05
Institute for Data Processing and Electronics
Karlsruhe Institute of Technology
Material Chooser
SiThickness220nm
ITU Channel Chooser
Common Port Position
X (µm)
Y (µm)
0
0
Number of Channels:
3
Distinct Ports
Channel No. | X (µm) | Y (µm) | Wavelength (nm) | Eff. ref. index
1 | 20 | 0 | 1540 | 2.8529996203
2 | 25 | 0 | 1550 | 2.8474815651
3 | 30 | 0 | 1560 | 2.8419924143
(Only coordinates for stigmatic points are important, others will be calculated)
1. Stigmatic Point Channel Number 1
2. Stigmatic Point Channel Number 3
Grating order 9
Min. radius (µm) 100
Max. radius (µm) 1000
2. TSP radius offset (µm) 0
Grating opening angle (°) 30
Blaze shift (µm) 0
Bounding Box for Port Search
X (µm)
Y (µm)
0
-15
50
15
Show Ellipses
Show Circles
Show Port Search Intersections
Show Grating Additions
Calculate
parallel computing
Results
Grating points found 0
to ease calculations as seen from common port position
Real min. radius (µm) 0
Real max. radius (µm) 0
Real opening angle (°) 0
Area of free space area (µm²):
reduced 0
full 0
SiO2 0
Bragg grating definition
Periods 4
Period length (nm) 361
SiO2 length (nm) 180.5
Boundary distance (nm) 10000
Reflector spacing (nm) 185
S-bend Waveguides
Waveguide length (nm) 200000
Endpoint pitch (nm) 2500
Curve discretization 200
Cladding Block Mode
Block Trench width (nm) 1600
Port layout definition
Waveguide width (nm) 450
Taper end width (nm) 2000
Taper length (nm) 30000
WG stub length (nm) 2000
Taper shift (nm) 0
Trench width (nm) 1000
Trench2 width (nm) 1000
Beveling
min. Angle 85
Bevel (nm) 151
Attenuator Bevel (nm) 310
COMSOL Simulation
Min. Wavelength (nm) 1500
Max. Wavelength (nm) 1600
Wavelength step (nm) 2
S-Bends
Free space area:
Red. Border
No. of Partitions 48
COMSOL Model
GDSII Definitions
Echelle Name/ID EG_01
GDS database unit 1e-09
GDS user unit 1e-06
GDS2 Layout



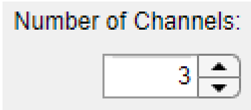
I will go through the UI and explain, what can be done with the controls. To generate an Echelle grating for the given values, you can just hit the ‘Calculate’ button. At first it will take some time as MATLAB has to set up its worker threads for parallel computing, which speeds up calculations for larger gratings significantly on multi core CPUs.

First choose a material system, which means here a certain silicon thickness in a SiO_2-Si-SiO_2 layer stack, using the drop down box. This is for choosing the correct effective refractive index of the silicon layer and is essential for the results. The values are stored in the Excel table ‘EffectiveRefractiveIndices_Wavelength1480-1630nm_SiliconThickness205-225nm+245-255nm.xlsx’. They were generated using COMSOL with a very simple model of a sheet waveguide with 2µm lower SiO_2 layer and 1µm upper SiO_2 layer. The middle Si layer has the stated thickness. Material parameters came from the COMSOL material library, for silicon ‘Si (Li-293K)’, for SiO_2 ‘SiO2 (Malitson)’. The table can be extended by your own values for your own material systems by adding columns with the variable name in the first line.

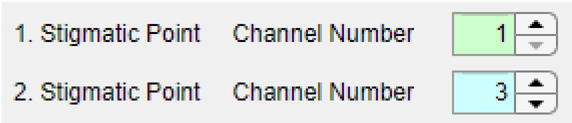
The ‘Common Port Position’ is the formal position of the waveguide end for the common port of the Echelle grating. I never had a reason to change it, but here you can. The common port always looks straight up from the given position. The position unit is micrometers.



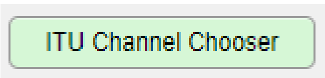
Then choose the number of channels your Echelle grating should have. This also affects, how many ports are listed in the ‘Distinct Ports’ table.

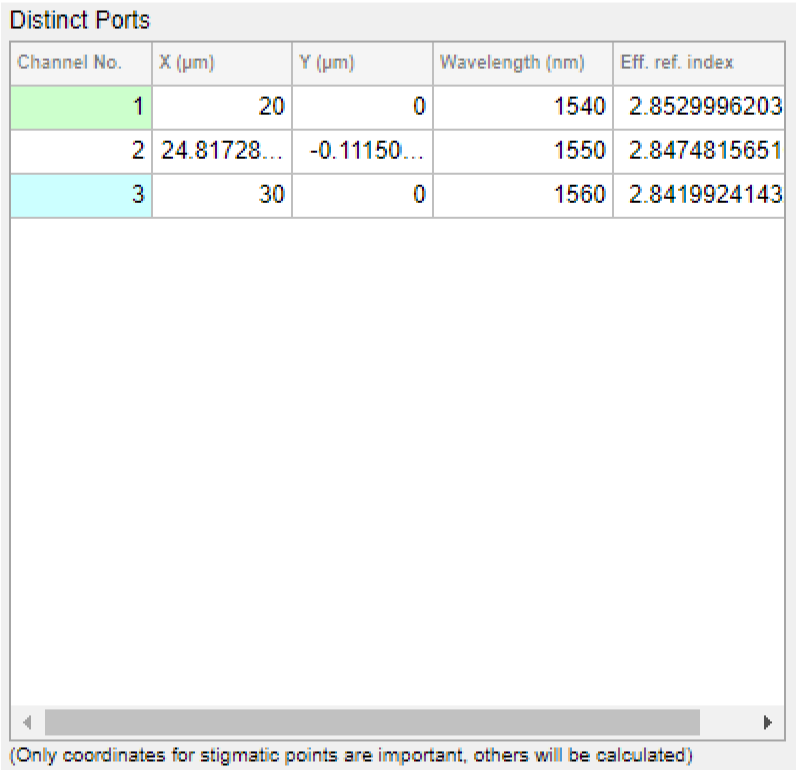


As the program uses the two stigmatic points (TSP) method to calculate the Echelle grating, you will have to choose, which channels/ports should be used for the stigmatic points. This can be selected with the controls below the table. The channel for the first stigmatic point is also indicated with green background, the second with a blue one.



Then you will have to choose, where the output ports for the stigmatic point channels should be positioned. To do that, enter the positions into the table. The following calculations only work, if the distinct ports are located on the right of the common port, which means positive x-values. You don’t have to choose any other channel positions, they are calculated afterwards. You can enter all wavelengths for the channels here or use the ‘ITU Channel Chooser’:

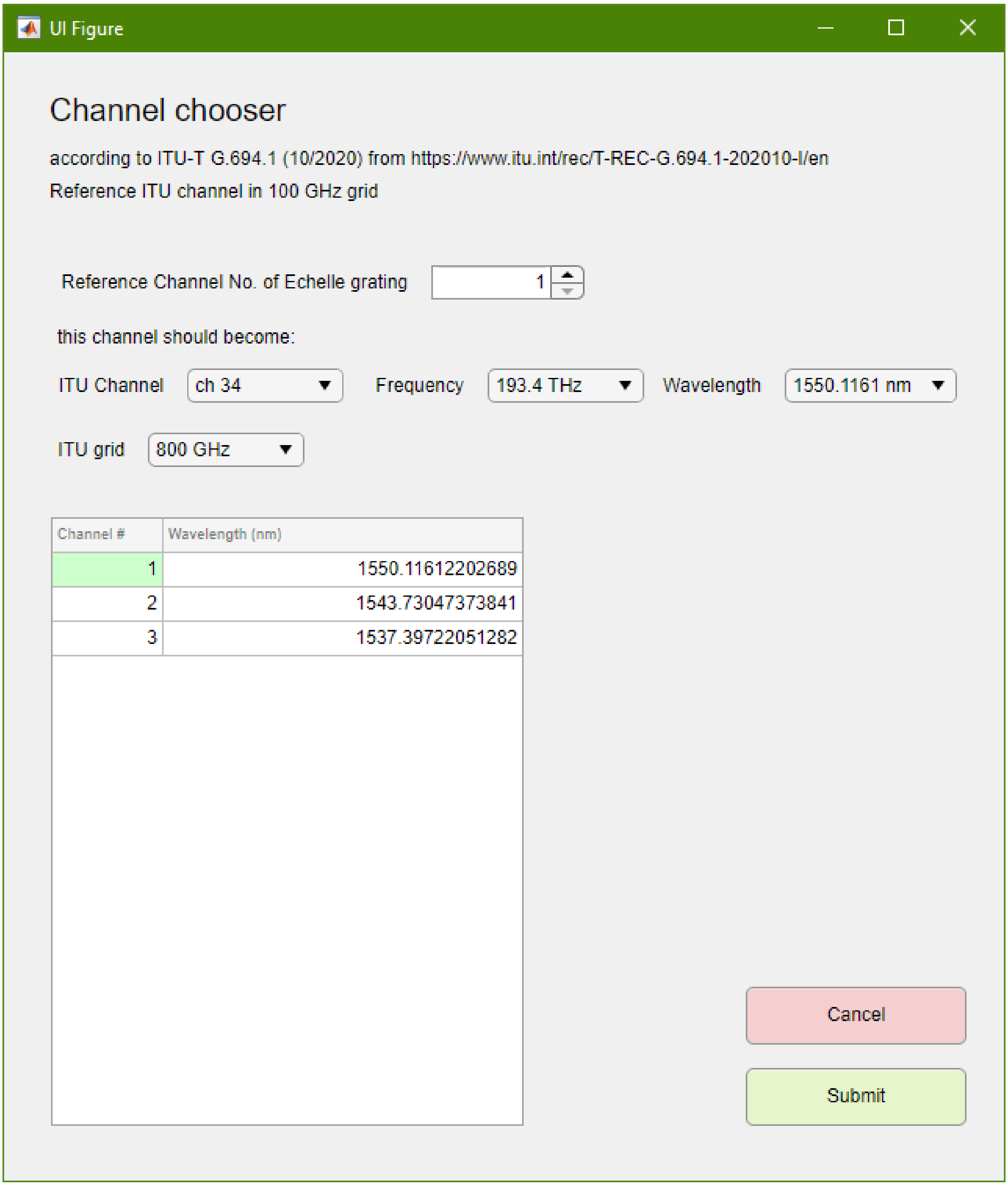



Distinct Ports

Channel No.	X (µm)	Y (µm)	Wavelength (nm)	Eff. ref. index
1	20	0	1540	2.8529996203
2	24.81728...	-0.11150...	1550	2.8474815651
3	30	0	1560	2.8419924143

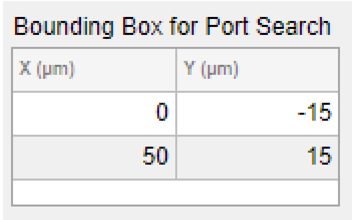
(Only coordinates for stigmatic points are important, others will be calculated)

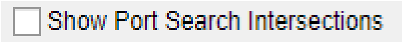
In the most right column, the effective refractive index for the respective wavelength is shown. This value is used for calculating the optical path lengths.



The ‘ITU Channel Chooser’ lets you choose a channel wavelength distribution according to ITU-T G.694.1 with channel spacings between 1600 GHz and 12.5 GHz either in increasing or decreasing order. For that you can choose a reference channel for which you enter either a ITU channel number or its frequency or its wavelength. According to the choosen ITU grid, the rest of the channels are assigned the respective wavelengths. Pushing ‘Submit’ submits the wavelengths to the main interface table and closes the window.

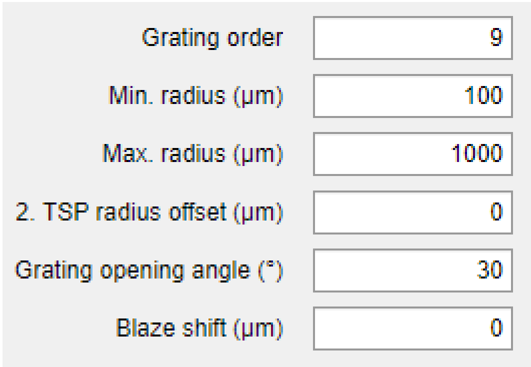
Another important table is the ‘Bounding Box for Port Search’. As the non-stigmatic-point ports are searched by calculating intersections from circles and these intersections being not unique, I limit the valid intersections to those within the bounding box. Therefore it should (must) contain the intersections on the final port position but not the second intersections. Choose the coordinates for a close bounding box not too large. To verify the bounding box you can activate





until you get a feeling, what are correct values. Unfortunately activating this option increases the time for drawing the figure significantly.

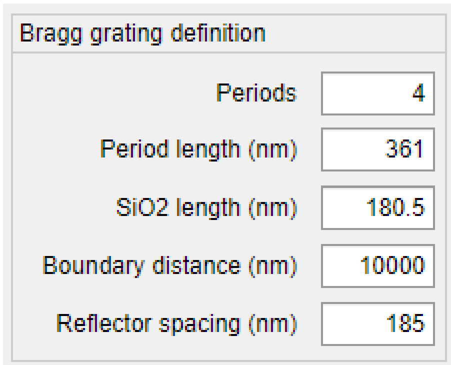
Further controls for the grating are found on the top right. You can choose the ‘Grating Order’, which determines how fine or coarse the grating will be. The ‘Min. radius’ is the radius of the large half axis of the smallest ellipse defining the grating points with the common port and the 1. stigmatic point as foci. Approximately it is the smallest distance between a grating element and the ports. Accordingly ‘Max. radius’ is the maximum Radius of the large half axis of the largest ellipse defining the grating points with the common port and the 1. stigmatic point as foci. This radius is not necessarily reached, as the ‘Grating opening angle’ also determines the length of the grating trajectory. Whichever is reached first, maximum radius or opening angle, stops the calculation.



The option ‘2. TSP radius offset’ lets you add an offset to the radius used for the ellipse calculation for the second stigmatic point. This means the path length between common port and second stigmatic port can be tuned to be different than between common port and first stigmatic port. With this you can rotate the grating trajectory around the ports. Several investigations have shown that a grating trajectory right at the top of the ports achieves best filter results.

The grating elements are small (Bragg) reflectors, whose orientations are calculated to (geometrically) reflect light from the common port to the center of the port block. Using ‘Blaze shift’ you can shift the position onto which the common port light is reflected. Most often positive values result in a flatter, more even channel response of the filtered channels.

The reflective grating elements used here are Bragg reflectors (another type could be a corner reflector, but this is not implemented). The Bragg gratings used are rectangles fully etched through the silicon layer and filled with SiO_2. Their properties are defined in the ‘Bragg grating definition’ block. First you can choose, how many periods should be used and how long one period should be. It is the SiO_2 length plus the Si length to the next SiO_2 strip. You can also name it the pitch of the Bragg grating. The next value is the length (in incident direction) of the SiO_2 strip. Splitting the period length symmetrically is rather good.

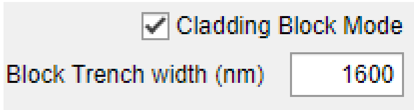


Combinations I simulated and found to be good for a wavelength of 1550 nm are:

Si-thickness (nm)	No. of Periods	Period length (nm)
215	4	366
250	4	361

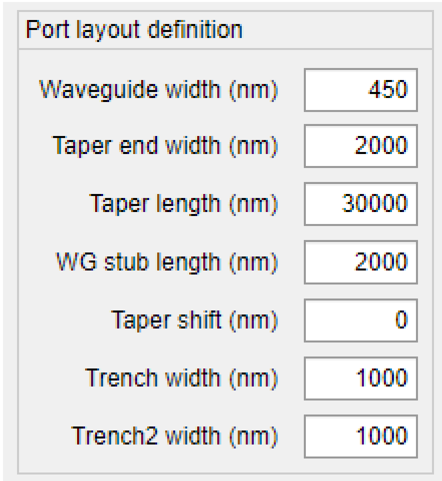
210	4	368
211	4	367.5
211	8	373 (this is no typo)

'Boundary distance' is a bit misleading, but is for an boundary extension of the silicon layer. In most parts of the design it is really the width of the boundary, especailly at the grating trajectory the boundary vertices are calculated to a direction perpendicular to the bragg grating. This results in a boundary much narrower than given here, but as long as the boundary doesn't cross the bragg reflectors, it should be okay.

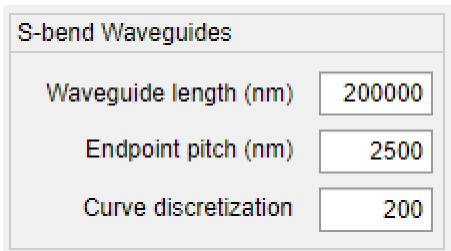


Another option worth mentioning in conjunction with the boundary is the 'Cladding Block Mode'. In general there is no need to surround the free space area of an Echelle grating with a boundary made of SiO_2, even worse it might trap stray light. But a wafer fab might have a design rule that all structures have to be surrounded by cladding of a certain width. For this you can activate the 'Cladding block mode' and define, how wide the surrounding cladding should be.

The last point in the Bragg reflector definition box is 'Reflector spacing' and needs further explanation, too. Especially with low grating orders, portions of one Bragg reflector might touch an adjacent Bragg reflector. From a physical point of view, this is unproblematic and good, but this might trigger errors in the design rule check from the wafer fab which should fabricate the structure. To overcome this, you can define a distance between adjacent Bragg reflectors. But for fine Echelle gratings with low orders this might also lead to too thin Bragg reflector structures.

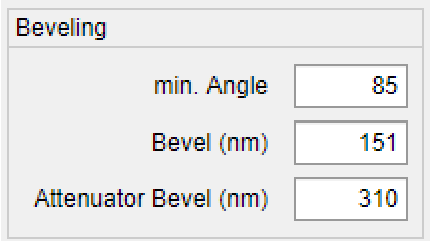


The layout of the port waveguides can be defined in the box 'Port layout definition'. The port waveguides consist of a linear taper, opening to the free space region and a short sub at the other end to connect further (single mode) waveguides to. The width of these waveguides is defined by 'Waveguide width'. The width of the taper end at the free space opening is defined by 'Taper end width', the length of the taper by 'Taper length'. The length of the additional stub is defined by 'Stub length'. Normally the calculated port position is in the middle of the open taper end, but you can shift it further down the taper by 'Taper shift', if required (or up/away from the taper using negative values). 'Trench width' defines the width of the cladding around port waveguides. For the outermost port waveguides it is overridden by 'Block Trench width', if 'Cladding Block Mode' is activated. You should choose the trench width small enough that the stubs of adjacent channels don't overlap, because this will harm the port definitions for the COMSOL simulations. If you don't need the simulation or want to edit the COMSOL simulation project anyway, you can also let the trenches overlap. 'Trench2 width' defines the cladding width at the end of added S-bend waveguids. These are used to bring the waveguide ends to a defined grid and orientation for further use of the layout.



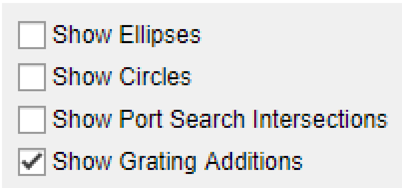
Further definitions of the S-bends are made in the 'S-bend Waveguides' block. The waveguide for the common port is always straight and has the length 'Waveguide length', which adds up to the port taper. (For that the whole design is rotated around the common port position to have the common port vertical.) Then bezier curves between the port taper stubs and the endpoints are generated, where the endpoints are on a horizontal line with a distance of ‘Endpoint pitch’ to each other. The discretization of the curve can be adjusted by ‘Curve discretization’.

When ‘Cladding Block Mode’ is off, a cladding for each waveguide is generated, maybe overlapping and generating sharp angles. When ‘Cladding Block Mode’ is on, the whole port and S-bend waveguide region gets a single cladding block for all waveguides.

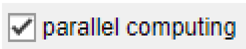


Speaking about sharp angles, angles sharper than ‘min. Angle’, defined in the ‘Bevelling’ box, can be deburred. The length of the bevel is defined by ‘Bevel’ and should be larger than the smallest feature size defined by the process design rules.

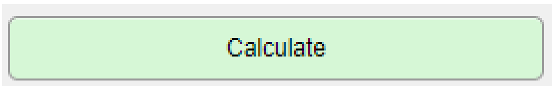
The generated Echelle grating consists of an undoped silicon film waveguide. But to attenuate stray light, on the left and right of of the required free propagation region, doped stripes are added, which might feature sharp angles, too. Most often the design rules for doped regions demand larger minimum feature sizes. Therefore the length of the bevel used for the attenuators can be defined seperately with ‘Attenuator Bevel’.



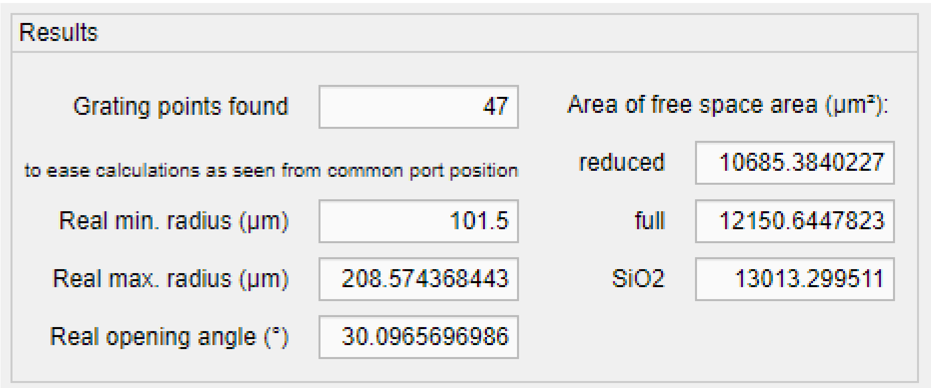
Before you calculate the Echelle grating, you can choose, which additional elements should be shown in the figure. ‘Show Ellipses’ draws the calculated ellipses for finding the grating positions. ‘Show circles’ shows the circles used for finding the ports other than the stigmatic ports. ‘Show Port Search Intersections’ just shows the intersections of all these circles, but only works, if ‘parallel computing’ is off. ‘Show Grating Additions’ shows the port tapers, the Bragg reflectors and their direction, the boundary, and some more features.

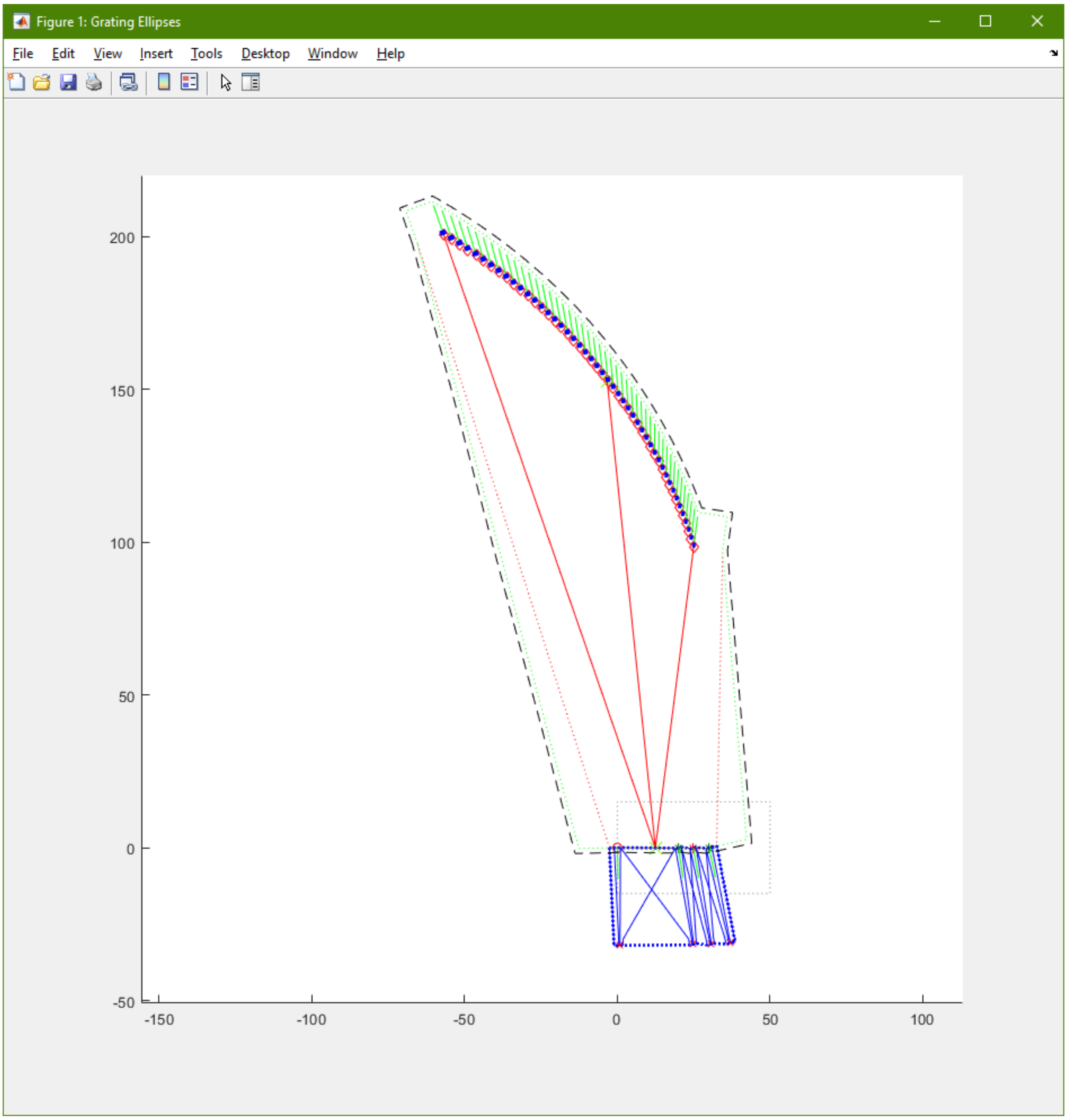


Activating ‘parallel computing’ delays the first calculation a bit, as MATLAB has to initialize its worker threads, but it can speed up the calculation of subsequent and larger gratings significantly, if you have a multicore processor (should be standard nowadays) and MATLABs ‘Parallel Computing Toolbox’ installed.



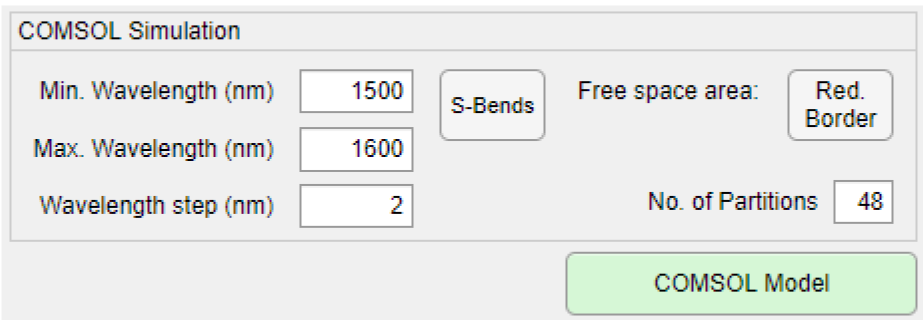
If you hit the ‘Calculate’ button now, the Echelle grating will be calculated. If there is a problem, an appropriate message should show up, but if everything runs well, you can find some parameters in the ‘Results’ box and a graphical representation in the Figure window.





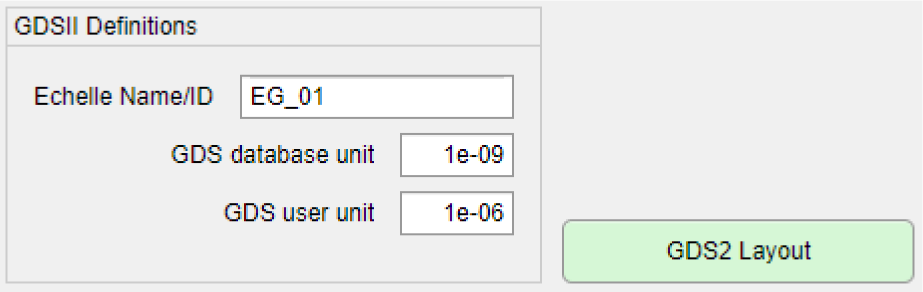
If you don’t like the result, you can change the parameters. But each new set of parameters described up to here requires a new calculation of the grating. It is not checked, which parts are really affected and which parts of the calculation could be reused.

The next possible step is generating a COMSOL model and simulation project from the calculated data. In the ‘COMSOL Simulation’ box you can define the



minimum and maximum wavelength for the simulation as well as the wavelength step size for the wavelength sweep. If you wish, you can add the S-bends to the simulation, but generally it just blows up the size and calculation time without any new results. Additionally, you can choose, if you wish to use the full border of the Echelle grating or a slightly reduced version to speed up simulation time a little and also save a little bit of memory. The last parameter is 'No. of Partitions' and affects, how the free space region is modeled. COMSOL meshes regions using a single core per region. The free space region is large and the required mesh rather fine, so it can take several hours to mesh it on a single core. To speed up this process, the only way is to split up the large free space region into several smaller ones and the parameter defines, how many regions are generated. To ease things, the regions are generated by cutting the free space region into the given number of horizontal stripes. This speeds up simulation setup significantly on multi-core processors (but doesn't look really nice, so generating a second project file with just one partition for publications is recommended).

When hitting the 'Comsol Model' button, you are asked for a file name and location for the project and after hitting 'Save', the simulation project is generated as .mph-file. The file can be loaded into COMSOL Multiphysics, and if the port stubs weren't too close together you can just hit run to start the simulation. The design parameters of the Echelle grating are saved in the comments of the properties of the component 'comp1'.



To use the Echelle grating in an own layout, you can set a name for the Echelle grating with 'Echelle Name/ID'. For GDSII export, a 'GDS database unit' and a 'GDS user unit' is required. The given values are 1 nm as database unit and 1 µm as user unit. This is often requested by the wafer fab and should be entered according to their rules.

When hitting the 'GDS2 Layout' button, you are asked for a file name and location for the GDSII file and after hitting 'Save', the layout is generated. Additionally a '…_comments.txt' file is generated with all the parameters. Another file is also generated, a Synopsys OptoDesigner™ project '…_ports.spt' with the BuildingBlock definition of the Echelle grating together with input and output port definitions for using the grating as multiplexer "MUX" or demultiplexer "DEMUX".

A currently awkward thing is the definition of the GDS layers. This is done at the beginning of the file 'BuildGDSFile.m'. Also in this file you might change the description of the BuildingBlock from line 581 on.

Apologies

If you go through the code you'll notice that there is a lot of code commented out. I didn't clean up the code to better orient myself, but it might be a little annoying for you. Sorry!

Help

If you find a bug in the software, please send a message to marc.schneider@kit.edu and I will try to fix it.