# Design Proposal For a Bragg Grating Resonator

Anusika Nijher, anusikanijher@gmail.com, edX Edge Username: ded2610b63ab4f4d9cfb44931ca9dc

***Abstract*—This document describes the design and simulation of multiple Waveguide Bragg grating resonators using a Fabry-Perot cavity.**

## I. Introduction

This document describes the design and simulation of a Waveguide Bragg grating resonator using a Fabry-Perot cavity. The goal is to design a resonator with the highest possible Quality Factor (QF). A high QF resonator is a key component of a laser. The resonator will be designed to operate at a central wavelength of 1310[nm]. In this document, we will first model a 220[nm] height by 335[nm] width stripe waveguide in Lumerical MODE. Next, a compact waveguide model from the effective and group index will be generated using the simulation data in MATLAB. A unit cell Bragg grating is simulated using Lumerical FDTD to extract the grating strength. The grating strength and effective index found from simulations are inserted into the Transfer Matrix Model (TMM) in MATLAB. The TMM is used to simulate a full Bragg grating resonator with a Fabry-Perot cavity. Using the TMM different resonator designs will be constructed varying the cavity length and the number of gratings. The designed resonators are also simulated in Lumerical INTERCONNECT. Lastly, a layout of the resonators with different design variations is done in KLayout.

## II. Waveguide Modelling

For Waveguide modelling, first a simulation is done in Lumerical MODE and then a compact model for the effective and group index is generated in Lumerical MODE. The simulations are done for a 220[nm] height by 335[nm] width stripe waveguide.

### A. Lumerical MODE

We are considering only the first quasi-TE mode for our design and the mode profile can be seen in Figure 1. The simulation is at a wavelength of 1310[nm].

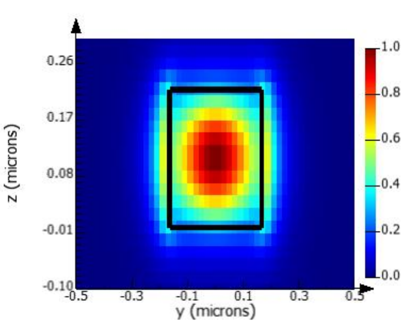


Fig. 1. First Quasi-TE Mode Electric Field Intensity for Stripe Waveguide (1310nm)

A wavelength sweep is then done from 1.2[$\mu$m] to 1.4[$\mu$m] in MODE for the first quasi TE mode and the data is saved for use in MATLAB.

### B. MATLAB

In Figures 2 and 3 a plot is shown of the effective and group index is shown alongside a compact model generated from the data points from MODE. The MATLAB script was provided Dr. Lukas Chrostowski [1].

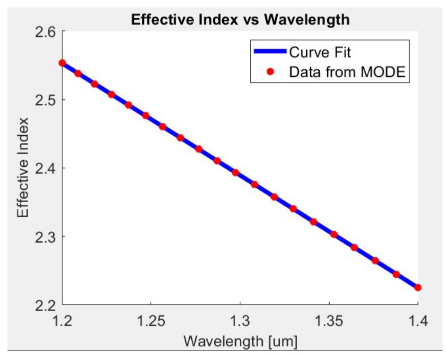


Fig. 2. Effective Index of Waveguide vs. Wavelength First Quasi TE Mode

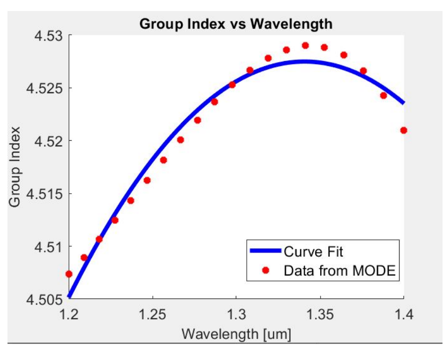


Fig. 3. Group Index of Waveguide vs. Wavelength First Quasi TE Mode

The compact model for the effective index is Equation 1 and the compact model for the group index is Equation 2. The units for $\lambda$ in both equations is $[\mu m]$.

$$n_{eff}(\lambda) = 2.373 - 1.642(\lambda - 1.31) - 0.0423\lambda - 1.31)^2 \quad (1)$$

$$n_g(\lambda) = 4.526 + 0.0693(\lambda - 1.31) - 1.129(\lambda - 1.31)^2 \quad (2)$$

## III. Bragg Grating Analytical Calculations

A diagram of a Bragg Grating is shown in Figure 4. Each periodic structure that makes up the overall grating has length $\Lambda$ (also called the Bragg Period). Peak light reflection through the Bragg grating occurs at a resonance wavelength called the Bragg wavelength( $\lambda_B$). At this wavelength, transmission is at a minimum. The Bragg gratings in this design will act as mirrors on either side of the Fabry-Perot cavity to form a resonator.

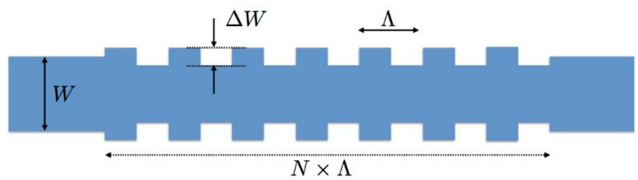


Fig. 4. Bragg Grating Diagram Source: L.Chrostowski Slides on Bragg Gratings [2]

Consider the diagram in Figure 5 to see a vector representation of the Bragg Grating.

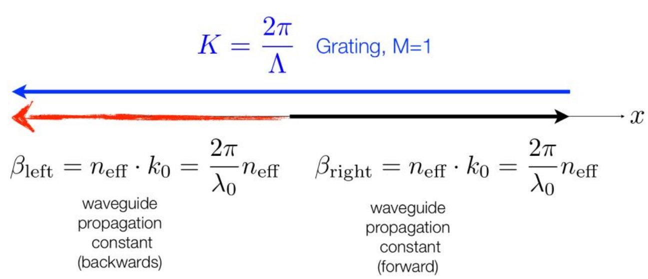


Fig. 5. Bragg Condition Diagram Source: L.Chrostowski Slides on Bragg Gratings [2]

The Bragg condition is when wave vector matching occurs:

$$\beta_{left} - K = \beta_{right}$$
$$\frac{2\pi}{\lambda_B} n_{eff} - \frac{2\pi}{\Lambda} = \frac{2\pi}{\lambda_B} n_{eff}$$
$$\lambda_B = 2 n_{eff} \Lambda$$

From the waveguide modelling we know that $n_{eff} = 2.373$ at $\lambda_B = 1310[nm]$. Therefore the Bragg Period is $\Lambda = 276[nm]$.

## IV. Bragg Grating Unit Cell Modelling

A unit cell (one Bragg period long) Bragg grating is simulated in Lumerical FDTD using a script provided by Dr. Chrostowski and modified as needed [1]. The width of the cell is kept fixed at $W = 335[nm]$ and the Bragg period is kept fixed at $\Lambda = 276[nm]$. The corrugation width $dW$ is swept from $10[nm]$ to $100[nm]$ in steps of $10[nm]$. Both rectangular and sinusoidal Bragg gratings are simulated for comparison. The central wavelength ($\lambda_B$) and bandwidth ($\Delta\lambda$) are extracted directly from the simulation. To find the grating strength ($\kappa$) the following equation is used [2]:

$$\kappa = \frac{n_g(\lambda_B)\pi\Delta\lambda}{\lambda_B^2} \tag{3}$$

In this case $n_g$ is a function of $\lambda_B$ using the compact model found in the waveguide modeling section.

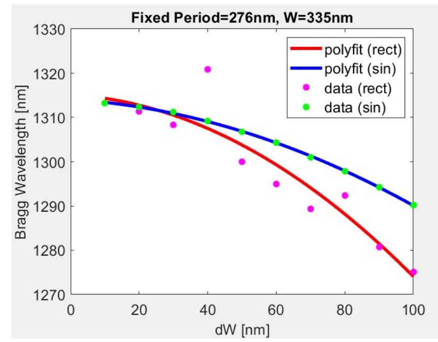


Fig. 6. Bragg Wavelength ($\lambda_B$) vs Corrugation Width ($dW$) for a fixed Bragg Period and Waveguide Width

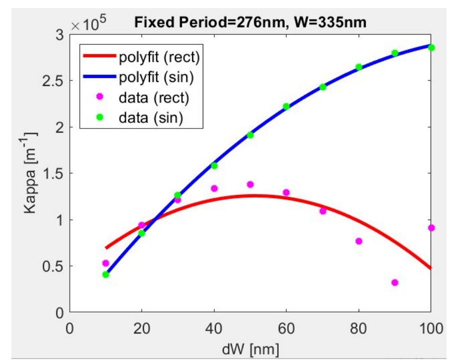


Fig. 7. Grating Strength ($\kappa$) vs Corrugation Width ($dW$) for a fixed Bragg Period and Waveguide Width

The sinusoidal Bragg grating behaves nicely along a polyfit curve whereas the rectangular Bragg grating has seemingly random "jumps" in the Bragg wavelength at corrugation widths at 40[nm] and 80[nm]. The Bragg wavelength tends to be lower for a rectangular grating relative to a sinusoidal grating at the same corrugation width. The rectangular grating also has the grating strength past a corrugation width of 50[nm] but this is not the case for sinusoidal gratings. As a result, resonator designs will be done using both rectangular and sinusoidal Bragg gratings to study the differences between these options.

## V. Transfer Matrix Method

The Transfer Matrix Method (TMM) is useful for testing the propagation of light through multi-layer film transmission [2]. A matrix can be used to represent a unit Bragg grating as a low and high effective index homogeneous waveguides alongside step matrices for the change in the effective index. The Fabry-Perot cavity of a Bragg Grating resonator can be represented as a high-index waveguide.

### A. Matrix Modelling of Bragg Grating and Fabry-Perot Cavity

The Bragg Grating Fabry-Perot Resonator can be thought of as three matrices multiplied together:

$$T_{LeftBragg} \cdot T_{cavity} \cdot T_{RightBragg} \tag{4}$$


Fig. 8. Block Diagram of a Bragg Grating Fabry-Perot Resonator

The cavity is just a homogeneous waveguide with a higher effective index connected to the Bragg gratings so

$$T_{cavity} = T_{hw2} \cdot T_{is21} \cdot T_{hw1} \cdot T_{is1c} \cdot T_{hwC} \cdot T_{isc1} \cdot T_{hw1} \cdot T_{is12} \cdot T_{hw2} \tag{5}$$

Where $c$ refers to the effective index of the cavity which is $n_c = 2.373$ which is the effective index of the 335nm wide stripe waveguide.

The left Bragg Grating can be represented as:

$$T_{LeftBragg} = (T_{hw2} \cdot T_{is21} \cdot T_{hw1} \cdot T_{is12})^{NG-1} \tag{6}$$

where $T_{is*}$ are the step matrices for change in the effective index, $T_{hw*}$ are the homogeneous waveguides and NG is the number of gratings.

Lastly, the right Bragg Grating can be represented as:

$$T_{RightBragg} = (T_{is21} \cdot T_{hw1} \cdot T_{is12} \cdot T_{hw2})^{NG-1} \tag{7}$$

$NG - 1$ gratings are used for the Bragg gratings as for the cavity a Bragg grating unit cell is added to each side.

The propagation matrix for a homogeneous waveguide and the step from a change in the effective index is given as:

$$T_{hwn} = \begin{bmatrix} e^{j\beta L} & 0 \\ 0 & e^{-j\beta L} \end{bmatrix} \tag{8}$$

$$T_{is12} = \begin{bmatrix} \frac{n_1 + n_2}{2\sqrt{n_1 n_2}} & \frac{n_1 - n_2}{2\sqrt{n_1 n_2}} \\ \frac{n_1 - n_2}{2\sqrt{n_1 n_2}} & \frac{n_1 + n_2}{2\sqrt{n_1 n_2}} \end{bmatrix} \tag{9}$$

where $L$ is the length of the waveguide and $\beta$ is the propagation constant (found as $\beta = \frac{2\pi n_{eff}}{\lambda} - j \cdot \frac{\alpha}{2}$, with $\alpha$ as the loss)

To find the effective indices $n_1$ and $n_2$ the change in effective index $\Delta n$ is used:

$$n_1 = n_{eff} - \frac{\Delta n}{2} \tag{10}$$

$$n_2 = n_{eff} + \frac{\Delta n}{2} \tag{11}$$

Where $n_{eff}$ is found as a function of $\lambda$ using the compact model found in the Waveguide Modelling $n_{eff}(\lambda)$.

$\Delta n$ is calculated as [2]:

$$\Delta n = \frac{\kappa \lambda_B}{2} \tag{12}$$

Where $\kappa$ is found as a function of $dW$ using the polyfit equations found through the Bragg Grating Unit Cell simulations.

### B. Parameter Sweeps of Number of Gratings and Cavity Length

The TMM is implemented in MATLAB. Various simulations and sweeps of cavity length and the number of gratings were done to understand the effects on the quality factor (QF), free spectral range (FSR), central wavelength, and peak transmission at the central wavelength. Some key findings are shown in the figures below. The corrugation width is fixed at $50[nm]$ and a sinusoidal Bragg Grating is selected as at this corrugation width the grating strengh is quite large at $1.9 \cdot 10^5 m^{-1}$ and the central wavelength is 1307[nm] which is quite close to the desired 1310[nm]. The loss $\alpha$ is conservatively 4dB/cm.

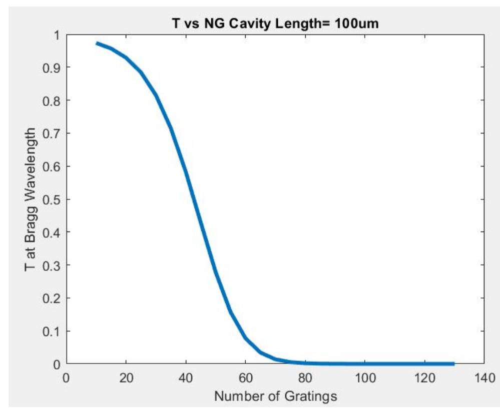


Fig. 9. Peak Transmission vs Number of Gratings

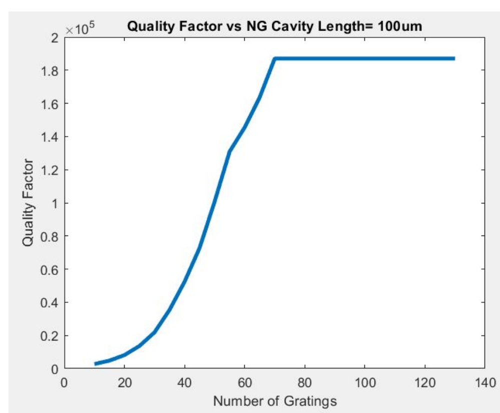


Fig. 10. Quality Factor vs Number of Gratings

From Figures 9 and 10, it is seen that as peak transmission at the Bragg Wavelength decreases, then the quality factor increases. There is a trade off to consider here as the peak transmission must be high enough that light is still able to enter the cavity to resonate. As a result, the selected designs keep the number of gratings between 40-60 on each side as that is before the peak transmissions completely goes to 0.

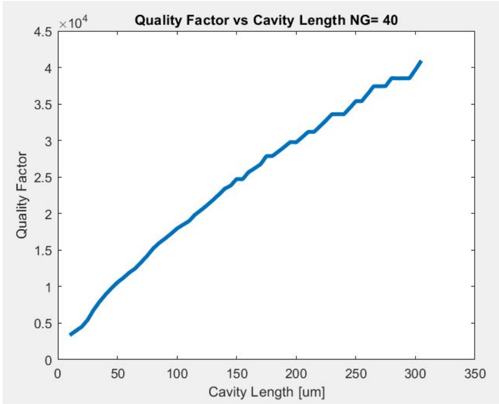


Fig. 11. Quality Factor vs Cavity Length

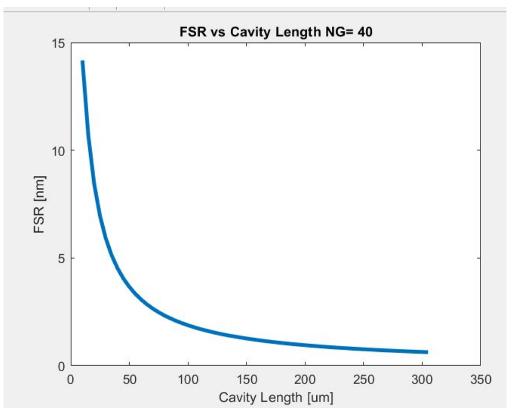


Fig. 12. FSR vs Cavity Length

From Figures 11 and 12, it is seen that a longer cavity results in both a lower FSR and a higher QF. As such the selected designs use longer cavities to maximize QF and minimize FSR. The selected cavity lengths are 100[$\mu$m] and 150[$\mu$m] due to size limitations of the provided floorplan.

### C. Selected Resonator Designs

In total there will be 16 designs fabricated. 8 will be using rectangular Bragg gratings and 8 using sinusoidal Bragg gratings. Out of the 8 designs, only 4 are unique as half the designs will be fabricated on a different side of the chip. This is to compare the effects of fabrication and chip variations on QF and FSR. The design are picked to vary cavity length and the number of gratings. The central design that is considered "safe" as in it should resonate and have a decently high-quality factor (around 52,000) uses a sinusoidal Bragg grating with a corrugation width of 50[nm] (this gives $\kappa = 1.9258 \cdot 10^5$), with 40 gratings on each side and a cavity length of 100 [$\mu$m]. The transmission and reflection spectrum of this design is shown in Figures 13 and 14.

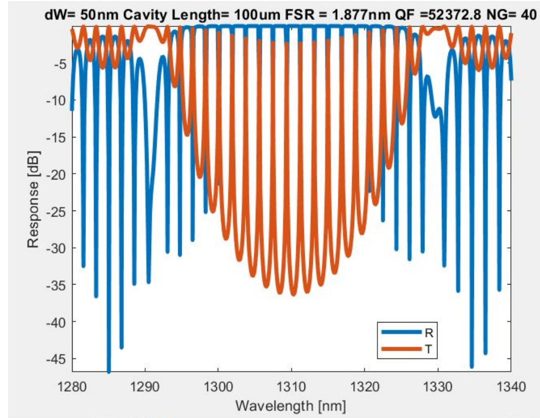


Fig. 13. Transmission and Reflection of a Bragg Grating Fabry Perot Cavity Resonator (Sinuosidal Bragg Grating), dB scale

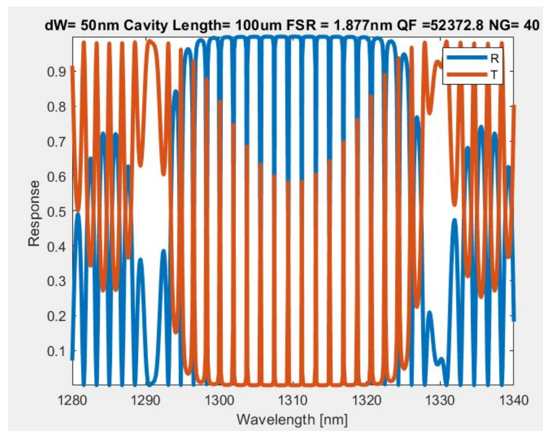


Fig. 14. Transmission and Reflection of a Bragg Grating Fabry Perot Cavity Resonator (Sinuosidal Bragg Grating)

The unique designs and expected results are summarized in Tables I and II. The design parameters are the corrugation width ($dW$), the cavity length ($CL$), and the number of gratings ($NG$). The results are the expected central wavelength ($\lambda_B$), the transmission at $\lambda_B$ ($T_{peak}$) the quality factor (QF), and the Free Spectral Range (FSR).

| $dW$ $nm$ | $CL$ $\mu m$ | $NG$ | $\lambda_B$ $nm$ | QF | FSR $nm$ | $T_{peak}$ dB |
|---|---|---|---|---|---|---|
| 50 | 100 | 40 | 1309.3 | 52370 | 1.877 | -2.3454 |
| 50 | 150 | 40 | 1310.6 | 68980 | 1.264 | -3.3047 |
| 50 | 100 | 60 | 1309.3 | 145480 | 1.876 | -11.1076 |
| 50 | 150 | 60 | 1310.6 | 163820 | 1.264 | -13.7666 |

TABLE I
MATLAB SIMULATION RESULTS FOR DIFFERENT BRAGG GRATING FABRY-PEROT RESONATORS USING SINUSOIDAL BRAGG GRATINGS

| $dW$ $nm$ | $CL$ $\mu m$ | $NG$ | $\lambda_B$ $nm$ | QF | FSR $nm$ | $T_{peak}$ dB |
|---|---|---|---|---|---|---|
| 50 | 100 | 40 | 1309.3 | 15225 | 1.857 | -0.6028 |
| 50 | 150 | 40 | 1310.6 | 21843 | 1.256 | -0.8790 |
| 50 | 100 | 60 | 1310.3 | 50359 | 1.851 | -2.1801 |
| 50 | 150 | 60 | 1310.6 | 65529 | 1.252 | -3.0674 |

TABLE II
MATLAB SIMULATION RESULTS FOR DIFFERENT BRAGG GRATING FABRY-PEROT RESONATORS USING RECTANGULAR BRAGG GRATINGS

A key result to verify will be that the rectangular Bragg grating designs perform worse than the sinusoidal designs.

## VI. Lumerical INTERCONNECT

The MATLAB results are verified by using a circuit designed in Lumerical INTERCONNECT. The circuit is shown in Figure 15.

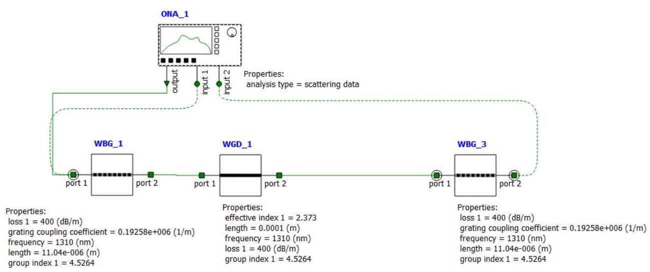


Fig. 15. Lumerical INTERCONNECT Circuit

The spectrum of the "safe" design (sinusoidal Bragg grating with a corrugation width of 50[nm], with 40 gratings on each side and a cavity length of 100 [$\mu$m]) found using INTERCONNECT is shown in Figure 16. Overall, the INTERCONNECT simulation is quite close to the TMM results.

Results for the unique designs are shown in Table III and IV.

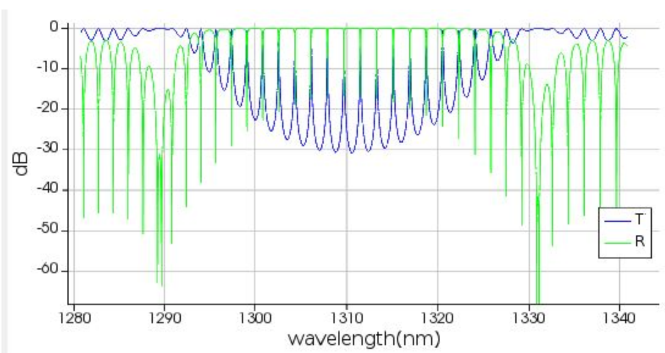


Fig. 16. Lumerical INTERCONNECT Simulation of Safe Design (dW=50[nm], NG=40, CL=100[$\mu$m])

| $dW$ $nm$ | $CL$ $\mu m$ | $NG$ | QF $nm$ | FSR |
|---|---|---|---|---|
| 50 | 100 | 40 | 34213 | 1.8 |
| 50 | 150 | 40 | 47125 | 1.22 |
| 50 | 100 | 60 | 137935 | 1.8 |
| 50 | 150 | 60 | 157841 | 1.22 |

TABLE III
INTERCONNECT Simulation Results for Different Bragg Grating Fabry-Perot Resonators Using Sinusoidal Bragg Gratings

| $dW$ $nm$ | $CL$ $\mu m$ | $NG$ | QF | FSR $nm$ |
|---|---|---|---|---|
| 50 | 100 | 40 | 8884 | 1.77 |
| 50 | 150 | 40 | 12806 | 1.2 |
| 50 | 100 | 60 | 32195 | 1.75 |
| 50 | 150 | 60 | 44259 | 1.2 |

TABLE IV
INTERCONNECT Simulation Results for Different Bragg Grating Fabry-Perot Resonators Using Rectangular Bragg Gratings

Overall INTERNCONNECT gives the about same FSR but a lower quality factor. However, the QF values are still quite high so this is not a real concern but shows that the TMM may be overly optimistic.

## VII. Layout

The layout of the selected designs are completed using KLayout. There are a total of 4 cells created. 1 cell for the rectangular Bragg Grating designs, 1 cell for the sinusoidal Bragg Grating designs. The other two cells are copies of the first two, just place on different sides of the chip. A pair of fibre-gating couplers connected by a 350[nm] stripe waveguide is also added to each cell to act as a de-embedding structure in the analysis of the experimental results. The floor size for each cell is 605[nm] by 410[nm]. Table V lists the 16 structure names, which of the four cells the structure is found in and its various parameters. For the type of grating R refers to rectangular and S to sinusoidal. All gratings have a corrugation width of 50[nm]. To account for fabrication bias the waveguides in KLayout are of width 350[nm].

| Structure Name | Cell Name | Type of Grating | Number of Gratings | Cavity Length [$\mu$m] |
|---|---|---|---|---|
| Bragg1 | A | R | 40 | 100 |
| Bragg2 | A | R | 40 | 150 |
| Bragg3 | A | R | 60 | 100 |
| Bragg4 | A | R | 60 | 150 |
| Bragg5 | B | S | 40 | 100 |
| Bragg6 | B | S | 40 | 150 |
| Bragg7 | B | S | 60 | 100 |
| Bragg8 | B | S | 60 | 150 |
| Bragg9 | C | R | 40 | 100 |
| Bragg10 | C | R | 40 | 150 |
| Bragg11 | C | R | 60 | 100 |
| Bragg12 | C | R | 60 | 150 |
| Bragg13 | D | S | 40 | 100 |
| Bragg14 | D | S | 40 | 150 |
| Bragg15 | D | S | 60 | 100 |
| Bragg16 | D | S | 60 | 150 |

TABLE V
KLayout Structures and Parameters

The layout of cell A is shown in Figure 17.

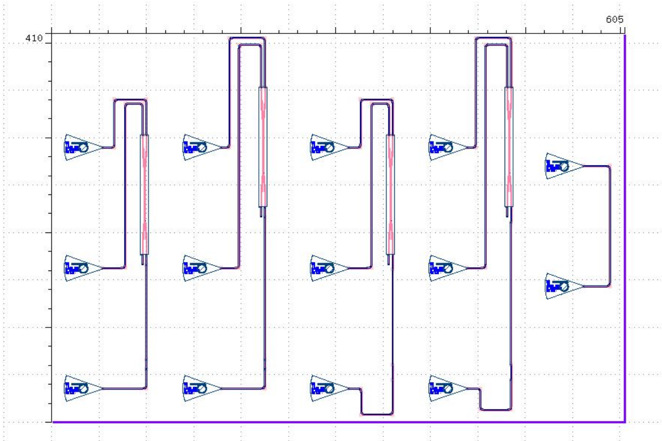


Fig. 17. Layout of Cell A

## VIII. Conclusion

This design proposal covers the simulation and layout of various Bragg Grating Fabry-Perot resonators. The main parameters varied are the type of grating (sinusoidal or rectangular), the cavity length, and the number of gratings. Simulations of the selected designs were completed in MATLAB and Lumerical INTERCONNECT.

## Acknowledgments

I acknowledge the excellent instruction from the UBC ELEC 413: Semiconductor Lasers teaching group, especially Dr. Lukas Chrostowski. Software to perform simulations and layout was provided by Lumerical Solutions, Mathworks and KLayout. MATLAB and Lumerical scripts for simulation including generating the compact model for the waveguide are provided by Dr. Lukas Chrostowski.

## References

[1] L. Chrostowski, "Lukasc-UBC/siliconphotonicsdesign," GitHub. [Online]. Available: https://github.com/lukasc-ubc/SiliconPhotonicsDesign.
[2] L. Chrostowski. Bragg Gratings [Lecture Notes].

## IX. Appendix: MATLAB Code

### A. Code for Compact Models of Effective and Group Index

Original Script: Dr.Chrostowski with modifications by Anusika Nijher

```
% Phot1x_fit_wg_compactmodel.m
% by Lukas Chrostowski, 2015

% User provides a matrix of neff values vs. wavelength
% Matlab curve fits to an expression.

% url='https://www.dropbox.com/s/xv4he4preyfa9v2/wg-export-TM.mat?dl=1'
%url='https://s3.amazonaws.com/edx-course-phot1x-chrostowski/Phot1x/wg-export-TM.
    mat'
%a=websave('wg.mat',url); % get data from Dropbox
load("wg_model_1310_220_335.mat");

neff = real(neff); % take the real part of the effective index.

c=299792458; % speed of light, m/s
lambdas = c ./ f; % f is the matrix of frequency points,
                  % where the effective index is recorded.
lambdas = lambdas * 1e6; % convert to microns.
lambda0 = 1.31; % replace with desired centre wavelength



% use Matlab anonymous function for the effective index expression:
neff_eq = @(nx, lambda) ...
            (nx(1) + nx(2).*(lambda-lambda0) + nx(3).*(lambda-lambda0).^2);

% initial guess.
X=[2.4 0 0];

%plot ( lambdas, neff_eq(X, lambdas), 'r')

% curve fit to find expression for neff.
format long
X = lsqcurvefit (neff_eq, X, lambdas, neff);

r=corrcoef(neff,neff_eq(X, lambdas));
r2=r(1,2).^2;
disp (['Goodness of fit, r^2 value: ' num2str(r2) ])

lambdas2=linspace(min(lambdas), max(lambdas), 100);
figure; hold on;
plot ( lambdas2, neff_eq(X, lambdas2), "b",LineWidth=4)
plot (lambdas, neff, '*r', LineWidth=4);
xlabel ('Wavelength [um]');
ylabel ('Effective Index');

legend ('Curve Fit','Data from MODE')
title("Effective Index vs Wavelength")
set(findall(gcf,'-property','FontSize'),'FontSize',14)

ng = c./vg;
% use Matlab anonymous function for the effective index expression:
ng_eq = @(nx, lambda) ...
            (nx(1) + nx(2).*(lambda-lambda0) + nx(3).*(lambda-lambda0).^2);

% initial guess.
Y=[4.498 0 0];


%plot ( lambdas, ng_eq(Y, lambdas), 'r')

% curve fit to find expression for ng.
format long
Y = lsqcurvefit (ng_eq, Y, lambdas, ng);

r=corrcoef(neff,neff_eq(Y, lambdas));
r2=r(1,2).^2;
disp (['Goodness of fit, r^2 value: ' num2str(r2) ])

lambdas2=linspace(min(lambdas), max(lambdas), 100);
figure; hold on;
plot ( lambdas2, ng_eq(Y, lambdas2),'b', LineWidth=4);
plot (lambdas, ng, '*r', LineWidth=4);
xlabel ('Wavelength [um]');
ylabel ('Group Index');
legend ('Curve Fit','Data from MODE')
title("Group Index vs Wavelength")
set(findall(gcf,'-property','FontSize'),'FontSize',14)

group_index_compact = Y;
effective_index_compact = X;
save('compact_models.mat', 'group_index_compact', 'effective_index_compact')
```

### B. Code for Generating Polyfit of Kappa vs dW

Source: Anusika Nijher

```
dW_vals_s=(10);
kappa_vals_s = (10);
center_wavelength_vals_s = (10);
ng_eq_coeffs = load("compact_models.mat","group_index_compact").group_index_compact
    ;

ng_eq = @(lambda) (ng_eq_coeffs(1)+ng_eq_coeffs(2)*(lambda-1.31)+ng_eq_coeffs(3)*(
    lambda-1.31)^2);


for i=1:10
    dW=i*1e-8;
    file_name = "raw_data/335_sinusoidal_dW_varying_"+num2str(dW) + ".mat";
    load(file_name);
    dW_vals_s(i)=dW;
    ng=ng_eq(center_wavelength*1e6);
    kappa_vals_s(i)=pi*ng*bandwidth/(center_wavelength^2);
    center_wavelength_vals_s(i)=center_wavelength;

end
kappa_eq_dW_s = polyfit(dW_vals_s, kappa_vals_s, 2);
center_wavelengths_eq_dW_s = polyfit(dW_vals_s, center_wavelength_vals_s, 2);

dW_test_s = linspace(1e-8,1e-7);
kappa_test_vals_s = polyval(kappa_eq_dW_s,dW_test_s);
center_wavelengths_test_vals_s = polyval(center_wavelengths_eq_dW_s, dW_test_s);
save("kappa_dw_eq_sinusoidal","kappa_eq_dW_s");

figure;
plot(dW_test_s*1e9, kappa_test_vals_s);
hold on;
plot(dW_vals_s*1e9, kappa_vals_s, "*r");
legend("polyfit", "test points");
xlabel("dW [nm]")
ylabel("kappa [m-1]")
title("Sinusoidal Fixed Period=276nm, W=335nm")


figure;
plot(dW_test_s*1e9, center_wavelengths_test_vals_s*1e9);
hold on;
plot(dW_vals_s*1e9, center_wavelength_vals_s*1e9, "*r");
legend("polyfit", "test points");
xlabel("dW [nm]")
ylabel("Bragg Wavelength [nm]")
title("Sinusoidal Fixed Period=276nm, W=335nm")



dW_vals_r=(10);
kappa_vals_r = (10);
```

```
center_wavelength_vals_r = (10);

for i=1:10
    dW=i*1e-8;
    file_name = "raw_data/335_rect_dW_varying_"+num2str(dW) + ".mat";
    load(file_name);
    dW_vals_r(i)=dW;
    ng=ng_eq(center_wavelength*1e6);
    kappa_vals_r(i)=pi*ng*bandwidth/(center_wavelength^2);
    center_wavelength_vals_r(i)=center_wavelength;

end
kappa_eq_dW_r = polyfit(dW_vals_r, kappa_vals_r, 2);
center_wavelengths_eq_dW_r = polyfit(dW_vals_r, center_wavelength_vals_r, 2);

dW_test_r = linspace(1e-8,1e-7);
kappa_test_vals_r = polyval(kappa_eq_dW_r,dW_test_r);
center_wavelengths_test_vals_r = polyval(center_wavelengths_eq_dW_r, dW_test_r);
save("kappa_dw_eq_rect","kappa_eq_dW_r");

figure;
plot(dW_test_r*1e9, kappa_test_vals_r);
hold on;
plot(dW_vals_r*1e9, kappa_vals_r, "*r");
legend("polyfit", "test points");
xlabel("dW [nm]")
ylabel("kappa [m-1]")
title("Rectangular Fixed Period=276nm, W=335nm")


figure;
plot(dW_test_r*1e9, center_wavelengths_test_vals_r*1e9);
hold on;
plot(dW_vals_r*1e9, center_wavelength_vals_r*1e9, "*r");
legend("polyfit", "test points");
xlabel("dW [nm]")
ylabel("Bragg Wavelength [nm]")
title("Rectangular Fixed Period=276nm, W=335nm")



figure;
plot(dW_test_r*1e9, center_wavelengths_test_vals_r*1e9, "r", LineWidth=3);
hold on;
plot(dW_test_r*1e9, center_wavelengths_test_vals_s*1e9, "b", LineWidth=3);
plot(dW_vals_r*1e9, center_wavelength_vals_r*1e9, "*m", LineWidth=3);
plot(dW_vals_s*1e9, center_wavelength_vals_s*1e9, "*g", LineWidth=3);
legend("polyfit (rect)", "polyfit (sin)", "data (rect)", "data (sin)");
xlabel("dW [nm]")
ylabel("Bragg Wavelength [nm]")
title("Fixed Period=276nm, W=335nm")
set(findall(gcf,'-property','FontSize'),'FontSize',14)


figure;
plot(dW_test_r*1e9, kappa_test_vals_r, "r", LineWidth=3);
hold on;
plot(dW_test_r*1e9, kappa_test_vals_s, "b", LineWidth=3);
plot(dW_vals_r*1e9, kappa_vals_r, "*m", LineWidth=3);
plot(dW_vals_r*1e9, kappa_vals_s, "*g", LineWidth=3);
legend("polyfit (rect)", "polyfit (sin)", "data (rect)", "data (sin)");
xlabel("dW [nm]")
ylabel("Kappa [m^{-1}]")
title("Fixed Period=276nm, W=335nm")
set(findall(gcf,'-property','FontSize'),'FontSize',14)
```

### *C. Code for Transfer Matrix of Homogeneous Waveguide*

Source: Dr.Chrostowski

```
function T_hw=HomoWG_Matrix(wavelength,l,neff,loss)
%Calculate the transfer matrix of a homgeneous waveguide.

%Complex propagation constant
beta = 2*pi*neff/wavelength-1i*loss/2;

v=[exp(1i*beta*l) exp(-1i*beta*l)];
T_hw=diag(v);

end
```

### *D. Code for Transfer Matrix of Index Step*

Source: Dr.Chrostowski

```
function T_is=IndexStep_Matrix(n1,n2)
%Calculate the transfer matrix for a index step from n1 to n2.

a=(n1+n2)/(2*sqrt(n1*n2));
b=(n1-n2)/(2*sqrt(n1*n2));
T_is=[a b; b a];
end
```

### *E. Code for TMM of Bragg Grating with Fabry Perot Cavity*

Original Source: Dr.Chrostowski, with modifications by Anusika Nijher

```
function T = Resonator_Grating_Matrix(wavelength, period, NG, n1, n2, loss,
        cavity_length, neff_eq)
% Calculate the total transfer matrix of the gratings
nc = neff_eq(wavelength);
l=period/2;
T_hw1=HomoWG_Matrix(wavelength,l,n1,loss);
T_is12=IndexStep_Matrix(n1,n2);
T_hw2=HomoWG_Matrix(wavelength,l,n2,loss);
T_is21=IndexStep_Matrix(n2,n1);
T_is1c=IndexStep_Matrix(n1,nc);
T_isc1=IndexStep_Matrix(nc,n1);
T_hwC=HomoWG_Matrix(wavelength,cavity_length,nc,loss);


T_left=T_hw2*T_is21*T_hw1*T_is12;
T_c = T_hw2*T_is21*T_hw1*T_is1c*T_hwC*T_isc1*T_hw1*T_is12*T_hw2*T_is21;
T_right = T_hw1 * T_is12 * T_hw2 * T_is21;
T=((T_left)^(NG-1))*T_c*((T_right)^(NG-1));

end
```

### *F. Code to Find Reflection and Transmission from TMM Results*

Source: Dr.Chrostowski

```
function [R,T]=Resonator_Grating_RT(wavelength, period, NG, n1, n2, loss,
        cavity_length, neff_eq)
%Calculate the R and T for a certain wavelength
M=Resonator_Grating_Matrix(wavelength, period, NG, n1, n2, loss, cavity_length,
        neff_eq);
T=abs(1/M(1,1))^2;
R=abs(M(2,1)/M(1,1))^2;
end
```

### *G. Code to Sweep Wavelength and Find TMM Results*

Source: Dr.Chrostowski

```
function [R, T, Lambda] = Bragg_Grating_Resonator(Bragg_wavelength, delta_n, period
        , NG, loss, cavity_length, N, res, neff_eq);
Lambda=zeros(N+1,1);
R=zeros(N+1,1);
T=zeros(N+1,1);

for i=1:N+1
    wavelength=Bragg_wavelength+(i-1-N/2)*res; %Wavelength sweep
    n_eff = neff_eq(wavelength*1e6);
    n1=n_eff-delta_n/2;
    n2=n_eff+delta_n/2;
    [r,t]=Resonator_Grating_RT(wavelength, period, NG, n1, n2, loss, cavity_length,
            neff_eq); %Calculate the R and T
    Lambda(i)=wavelength*1e9; %in nm
    R(i)=r;
    T(i)=t;
end

end
```

### *H. Code for Plotting Response of One Bragg Grating Resonator*

Original Script: Dr.Chrostowski with modifications by Anusika Nijher

```
%This file is used to plot the reflection/transmission spectrum,
%clear;
%clc;
%close all;
ng_eq_coeffs = load("compact_models.mat","group_index_compact").group_index_compact
        ;
ng_eq = @(lambda) (ng_eq_coeffs(1)+ng_eq_coeffs(2)*(lambda-1.31)+ng_eq_coeffs(3)*(
        lambda-1.31)^2);

neff_eq_coeffs = load("compact_models.mat","effective_index_compact").
        effective_index_compact;
neff_eq = @(lambda) (neff_eq_coeffs(1)+neff_eq_coeffs(2)*(lambda-1.31)+
        neff_eq_coeffs(3)*(lambda-1.31)^2);

kappa_eq_coeffs = load("kappa_dw_eq_sinusoidal.mat","kappa_eq_dW_s").kappa_eq_dW_s;
kappa_eq = @(dW) (kappa_eq_coeffs(1)*dW^2+kappa_eq_coeffs(2)*dW+kappa_eq_coeffs(3))
        ;

span=60e-9; %Set the wavelength span for the simulation
resolution=0.001e-9; %Set the wavelength resolution
N=round(span/resolution);

Bragg=1310e-9; %Bragg wavelength
Period=276e-9; %Bragg period
NG=40; %Number of grating periods

L=NG*Period; %Grating length
dW=50e-9;
kappa = kappa_eq(dW);
delta_n = kappa*Bragg/2; %Index contrast between n1 and n2
cavity_length=100e-6;

lossdBcm=4;
```

```
loss=log(10)*lossdBcm/10*100;

[R,T, Lambda] = Bragg_Grating_Resonator(Bragg, delta_n, Period, NG, loss,
        cavity_length, N, resolution, neff_eq);
[max_bragg_lambda, max_t_at_bragg, quality_factor,FSR] = Find_Max_T_Q_FSR(Lambda,T,
         Bragg);

figure;
plot (Lambda,[10*log10(R), 10*log10(T)],'LineWidth',3);
legend("R","T")
xlabel('Wavelength [nm]')
ylabel('Response [dB]');
title("dW= "+num2str(dW*1e9)+"nm Cavity Length= "+num2str(cavity_length*1e6) + "um
       FSR = " +num2str(FSR)+ "nm QF =" +num2str(quality_factor)+ " NG= "+num2str(NG
       ))
legend
axis tight;

figure;
plot (Lambda,[R, T],'LineWidth',3);
hold on
legend("R","T")
xlabel('Wavelength [nm]')
ylabel('Response ');
title("dW= "+num2str(dW*1e9)+"nm Cavity Length= "+num2str(cavity_length*1e6) + "um
       FSR = " +num2str(FSR)+ "nm QF =" +num2str(quality_factor)+ " NG= "+num2str(NG
       ))
legend
axis tight;

%Fixed_Cavity_Length_Vary_NG(Bragg,delta_n,Period,loss,cavity_length, N, resolution
       , neff_eq);
%Fixed_NG_Vary_Cavity_Length(Bragg,delta_n,Period,loss,NG, N, resolution, neff_eq);
```

## I. Code for Sweeping Cavity Length with Fixed Number of Gratings

Source: Anusika Nijher

```
function Fixed_NG_Vary_Cavity_Length(Bragg_wavelength,delta_n,period,loss,NG, N,
       res, neff_eq);
  T_at_Bragg_new=(60);
  Bragg_new_wavelength=(60);
  cavity_length_plot = (60);
  Q_factor = (60);
  FSR = (60);
  for i=1:60
       cavity_length=10e-6+(i-1)*5e-6;
       cavity_length_plot(i)=cavity_length;
       %NG=45;
       [R, T, Lambda] = Bragg_Grating_Resonator(Bragg_wavelength, delta_n, period,
              NG, loss, cavity_length, N, res, neff_eq);
       [Bragg_new_wavelength(i), T_at_Bragg_new(i), Q_factor(i) ,FSR(i)] =
              Find_Max_T_Q_FSR(Lambda,T, Bragg_wavelength);

  end
  figure;
  plot(cavity_length_plot*1e6, T_at_Bragg_new, 'LineWidth',3);
  xlabel("Cavity Length [um]")
  ylabel("T at Bragg Wavelength");
  title("T vs Cavity Length NG= "+num2str(NG));
  figure;
  plot(cavity_length_plot*1e6, Bragg_new_wavelength, 'LineWidth',3);
  xlabel("Cavity Length [um]")
  ylabel("Bragg Wavelength [nm]");
  title("Bragg Wavlength vs Cavity Length NG= "+num2str(NG));
  figure;
  plot(cavity_length_plot*1e6, Q_factor, 'LineWidth',3);
  xlabel("Cavity Length [um]")
  ylabel("Quality Factor");
  title("Quality Factor vs Cavity Length NG= "+num2str(NG));
  figure;
  plot(cavity_length_plot*1e6, FSR, 'LineWidth',3);
  xlabel("Cavity Length [um]")
  ylabel("FSR [nm]");
  title("FSR vs Cavity Length NG= "+num2str(NG));

end
```

## J. Code for Sweeping Number of Gratings with Fixed Cavity Length

Source: Anusika Nijher

```
function Fixed_Cavity_Length_Vary_NG(Bragg_wavelength,delta_n,period,loss,
       cavity_length, N, res, neff_eq);

  T_at_Bragg_new=(25);
  Bragg_new_wavelength=(25);
  NG_plot = (25);
  Q_factor = (25);
  FSR = (25);
  for i=1:25
       NG=10+(i-1)*5;
       NG_plot(i)=NG;
       %NG=45;
       [R, T, Lambda] = Bragg_Grating_Resonator(Bragg_wavelength, delta_n, period,
              NG, loss, cavity_length, N, res, neff_eq);
       [Bragg_new_wavelength(i), T_at_Bragg_new(i), Q_factor(i) ,FSR(i)] =
              Find_Max_T_Q_FSR(Lambda,T, Bragg_wavelength);

  end
  figure;
  plot(NG_plot, T_at_Bragg_new, 'LineWidth',3);
  xlabel("Number of Gratings")
  ylabel("T at Bragg Wavelength");
  title("T vs NG Cavity Length= "+num2str(cavity_length*1e6)+"um");
  figure;
  plot(NG_plot, Bragg_new_wavelength, 'LineWidth',3);
  xlabel("Number of Gratings")
  ylabel("Bragg Wavelength [nm]");
  title("Bragg Wavlength vs NG Cavity Length= "+num2str(cavity_length*1e6)+"um");
  figure;
  plot(NG_plot, Q_factor, 'LineWidth',3);
  xlabel("Number of Gratings")
  ylabel("Quality Factor");
  title("Quality Factor vs NG Cavity Length= "+num2str(cavity_length*1e6)+"um");
  figure;
  plot(NG_plot, FSR, 'LineWidth',3);
  xlabel("Number of Gratings")
  ylabel("FSR [nm]");
  title("FSR vs NG Cavity Length= "+num2str(cavity_length*1e6)+"um");
end
```

## K. Code for Finding FSR, QF and Central Wavelength from Transmission Spectrum

Source: Anusika Nijher

```
function [max_bragg_lambda, max_t_at_bragg, quality_factor,FSR] = Find_Max_T_Q_FSR(
       Lambda,T, Bragg_wavelength)

       [local_max_T_index local_max_T_nu] = islocalmax(T); %nu is the local array
              and is not used
       Lambda_local_max = Lambda(local_max_T_index);
       [Lambda_closest_to_Bragg_min, Lambda_closest_to_Bragg_index] = min(abs(
              Lambda_local_max-Bragg_wavelength*1e9));
       Lambda_bragg = Lambda_local_max(Lambda_closest_to_Bragg_index);
       Lambda_right_bragg = Lambda_local_max(Lambda_closest_to_Bragg_index+1);
       main_bragg_index = find(Lambda==Lambda_bragg);
       second_peak_index = find(Lambda==Lambda_right_bragg);
       T_local_max = T(main_bragg_index);
       max_t_at_bragg=T_local_max;
       max_bragg_lambda=Lambda_bragg;
       T_local_max_dB = 10*log10(T_local_max);
       T_dB = 10*log10(T);

      T_bw_right=0;
      T_bw_right_index=length(T);
      T_bw_left=0;
      T_bw_left_index=1;

       for j=main_bragg_index:1:length(Lambda)
           if(T_dB(j) <= T_local_max_dB-3)
               T_bw_right=T_dB(j);
               T_bw_right_index=j;
               break
           end
       end

       for j=main_bragg_index:-1:1
           if(T_dB(j) <= T_local_max_dB-3)
               T_bw_left=T_dB(j);
               T_bw_left_index=j;
               break
           end
       end

        quality_factor= (Lambda(main_bragg_index))/(Lambda(T_bw_right_index)-Lambda
               (T_bw_left_index));
        FSR = Lambda(second_peak_index)-Lambda(main_bragg_index);

%
%         figure;
%         plot(Lambda, T_dB)
%         hold on
%         plot(Lambda(T_bw_right_index), T_bw_right, "*r")
%         plot(Lambda(T_bw_left_index), T_bw_left, "*r")
%         plot(Lambda(main_bragg_index), T_local_max_dB, "*g")
%         plot(Lambda(second_peak_index), T_dB(second_peak_index), "*g")
%
%         figure;
%         plot(Lambda, T);
%         hold on
%         plot(Lambda_bragg, T_local_max, "*r");
end
```

## L. Code for Finding FSR, QF and Central Wavelength of Specified Designs from a CSV

Source: Anusika Nijher

```
ng_eq_coeffs = load("compact_models.mat","group_index_compact").group_index_compact
       ;
ng_eq = @(lambda) (ng_eq_coeffs(1)+ng_eq_coeffs(2)*(lambda-1.31)+ng_eq_coeffs(3)*(
       lambda-1.31)^2);

neff_eq_coeffs = load("compact_models.mat","effective_index_compact").
       effective_index_compact;
neff_eq = @(lambda) (neff_eq_coeffs(1)+neff_eq_coeffs(2)*(lambda-1.31)+
       neff_eq_coeffs(3)*(lambda-1.31)^2);

kappa_eq_coeffs = load("kappa_dw_eq_rect.mat","kappa_eq_dW_r").kappa_eq_dW_r;
kappa_eq = @(dW) (kappa_eq_coeffs(1)*dW^2+kappa_eq_coeffs(2)*dW+kappa_eq_coeffs(3))
       ;

span=60e-9; %Set the wavelength span for the simulation
resolution=0.001e-9; %Set the wavelength resolution
N=round(span/resolution);
```

```
Bragg=1310e-9; %Bragg wavelength
Period=276e-9; %Bragg period

lossdBcm=4;
loss=log(10)*lossdBcm/10*100;

design_params = readtable("design_params.csv");
dWs = design_params.dW*1e-9;
cavity_lengths = design_params.CL*1e-6;
NGs = design_params.NG;

max_bragg_lambda = (length(dWs));
max_t_at_bragg = (length(dWs));
quality_factor = (length(dWs));
FSR = (length(dWs));

for i=1:length(dWs)
    NG= NGs(i);
    L=NGs(i)*Period; %Grating length
    kappa = kappa_eq(dWs(i));
    delta_n = kappa*Bragg/2; %Index contrast between n1 and n2
    cavity_length=cavity_lengths(i);
    [R,T, Lambda] = Bragg_Grating_Resonator(Bragg, delta_n, Period, NG, loss,
        cavity_length, N, resolution, neff_eq);
    [max_bragg_lambda(i), max_t_at_bragg_val, quality_factor(i),FSR(i)] =
        Find_Max_T_Q_FSR(Lambda,T, Bragg);
    max_t_at_bragg(i) = 10*log10(max_t_at_bragg_val);
end
```