Lab 2: Uniform Grating Coupler

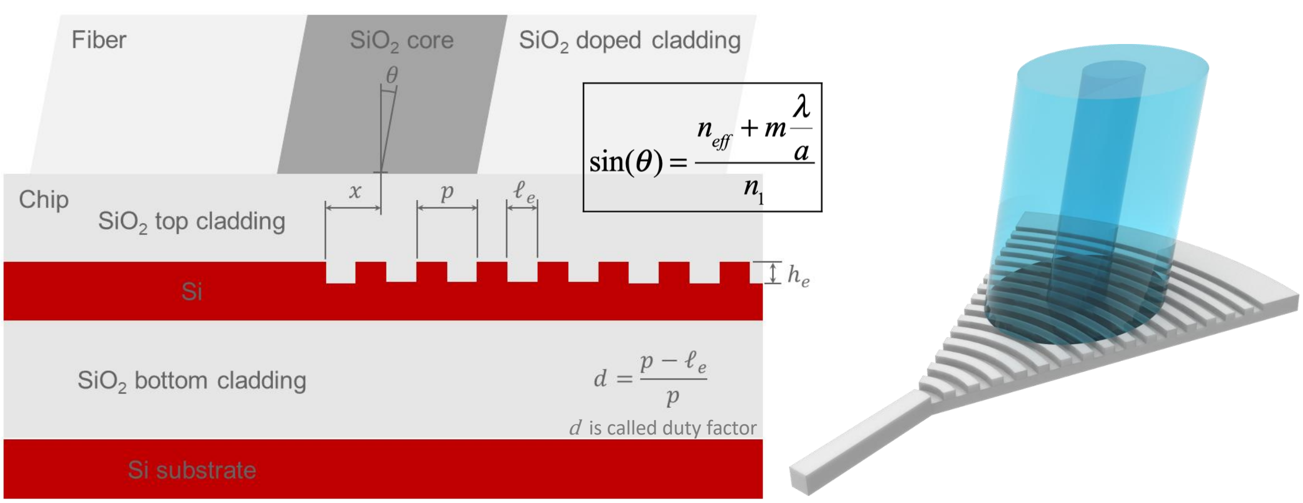
Part 1: Scripting the Setup with Lumerical Language

Objective:

- Learn how to set up simulations using the Lumerical scripting language.
- Understand the fundamental design of a grating coupler, including its structure and key design parameters. Explore which design parameters can be modified and how they affect performance.

Background:

Grating couplers are optical devices used in integrated photonics to efficiently couple light between optical fibers and planar waveguides. These couplers typically consist of a periodic grating structure etched into the chip's surface. The grating diffracts incident light and couples it into or out of the waveguide with minimal reflection. Grating couplers are essential components for the integration of photonic devices, enabling the seamless interfacing of on-chip optical circuits with external optical communication systems. Their design and optimization play a crucial role in achieving high coupling efficiency and minimizing signal loss in integrated photonics applications.



Procedure:

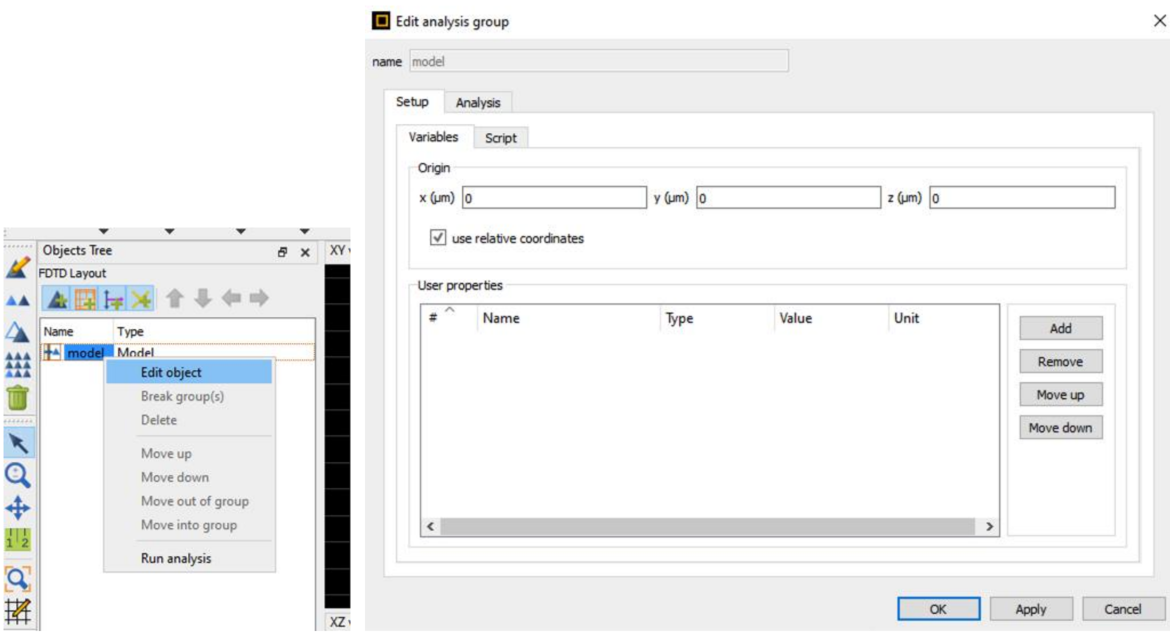
Open Lumerical and click on the Solvers tab. Start a new FDTD project.

There are several ways to script in Lumerical: you can write commands in the *Script Prompt* at the bottom of the window, create a script file and run it using the *Script File Editor* on the right side, or script within specific objects in the *Object Tree* on the left side. When scripting under objects, you typically script under the *model* (the top layer of the *Object Tree*), which creates global variables and scripts the entire setup, including all components, sources, monitors, simulation solvers, meshes, etc. Alternatively, you can script within a specific *group*, where the variables and script only affect the objects within that group.

In this example, we will script under *model* for global scripting.

Define the Simulation Variables

1. Right click on ***model*** and edit the object. A window with the header "*Edit analysis group*" will appear.



2. Add ***Variables*** to the *User Properties* list. This list is an easily-tractable interface to quicky modify parameters in your simulation. We will use the following as starting points to design our grating. Since we have defined our variables here, they will be easily accessible in other aspects of the program (sweeps, optimizations, monte carlo, etc…)

Name	Type	Value	Unit
`wavelength`	Length	1.55	um
`pitch`	Length	0.6	um
`etch_depth`	Length	0.07	um
`duty_factor`	Number	0.5	
`N`	Number	15	
`cladding`	Length	2	um

Script the Photonic Structures

1. Navigate to the ***Script*** tab (next to *Variables*). Begin the script by initializing the workspace.

```
deleteall;
```

   If you want to add a comment, use a "#." Everything after the # on each line will be treated as a comment and appear in green. For example:

```
deleteall; # delete all previously existing setups in the file
```

   You can see my comments in the code within the uploaded file ***grating_coupler_2D.fsp***.

2. Define the optical waveguide. We will use a 220 nm thick silicon rectangle to guide the optical mode. The waveguide will begin at x = -10 µm, and finish at 50 µm. Have the bottom of the silicon begin at y = 0 µm. Once again, we will start our design procedure with a 2D simulation, so set the z position to 0 µm and the z span to 10 µm.

```
addrect;
set('name', 'waveguide');
set('x min', -10e-6);
set('x max', 50e-6);
set('y min', 0);
set('y max', 0.22e-6);
set('z', 0);
set('z span', 10e-6);
set('material','Si (Silicon) - Palik');
```

3. Insert the grating coupler teeth. Use a *for* loop to create a rectangle for each individual grating tooth. Each tooth is constructed from *etch* material, has a length of *pitch * (1-duty_factor*), and a depth of *etch_depth*.

```
x0 = 0;
for (i = 1:N){
    x1 = x0 + (1-duty_factor)*pitch;
    addrect;
    set('name', 'tooth_' + num2str(i));
    set('x min', x0);
    set('x max', x1);
    set('y min', 0.22e-6 - etch_depth);
    set('y max', 0.22e-6);
    set('z', 0);
    set('z span', 10e-6);
    set('material', 'etch');
    x0 = x0 + pitch;
    addtogroup('grating_teeth');
}
```

Note that the last sentence adds each grating tooth into a *structure group*. We use a *structure group* here primarily to make the object tree clearer (to avoid having a long list of individual teeth).

In other cases, the *structure group* allows you to edit the properties of the entire group at once, rather than editing each object individually. You can refer to the other uploaded file ***grating_coupler_2D_fiber.fsp***, which contains a structure group for the *fiber*. With the structure group, you can view all the components inside the *fiber* as a whole, making it much easier to move or rotate the fiber as a unit.

4. Add the bottom silicon dioxide cladding. Since this is a simulation variable, we need to use the variable name *cladding* in our rectangle definition. Make the cladding width as wide as the waveguide, and the thickness be bound by [0, -cladding]. Set the z dimensions to be larger than the grating teeth.

```
addrect;
set('name', 'cladding');
set('x min', -10e-6);
set('x max', 50e-6);
set('y min', -cladding);
set('y max', 0);
set('z', 0);
set('z span', 20e-6);
set('material','SiO2 (Glass) - Palik');
```

5. Introduce the substrate to our simulation. Typically, photonics wafers (Silicon on Insulator) have a 600 - 700 µm 'handle' silicon substrate. Set the x dimensions to equal those of the waveguide. Make sure the y dimension is a function of the cladding thickness – the substrate should start where the cladding ends.

```
addrect;
set('name', 'substrate');
set('x min', -10e-6);
set('x max', 50e-6);
set('y min', -cladding-20e-6);
set('y max', -cladding);
set('z', 0);
set('z span', 20e-6);
set('material','Si (Silicon) - Palik');
```

Introduce the Optical Source

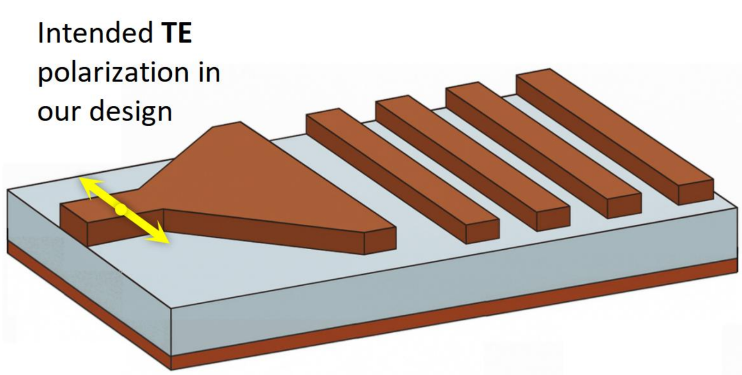
1. We will use a confined optical source – a waveguide mode propagating in the positive x direction. Place it 2.5 microns behind the first grating tooth, and ensure the mode is wide enough to encapsulate most of the cladding, but none of the bottom substrate.

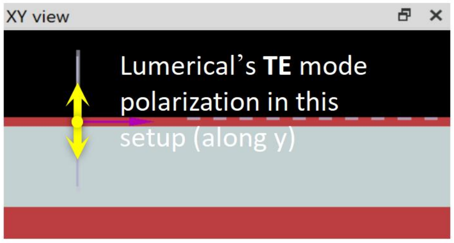
Set the single wavelength equal to wavelength. We aim to use the fundamental TE mode but will select the ***fundamental TM mode*** for *mode selection* here (the reason is explained in detail below).

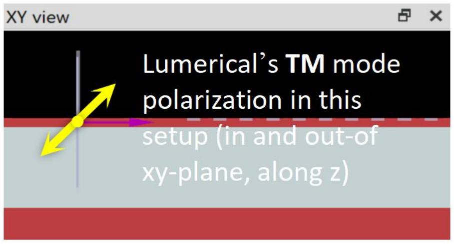
```
addmode;
set('name', 'mode');
set('injection axis', 'x-axis');
set('direction', 'forward');
set('x', -2.5e-6);
set('y', 0.22e-6/2);
set('y span', 0.22e-6 + 2*(3/4*cladding));
set('z', 0);
set('z span', 15e-6);
set('wavelength start', wavelength);
set('wavelength stop', wavelength);
set('mode selection', 'fundamental TM mode');
```

Note that for this setup, we aim to use the TE mode. In this context, the TE mode refers to the electric field being polarized parallel to the surface of the chip, as illustrated in the 3D plot below. In the FDTD setup, the TE mode corresponds to the electric field being polarized in the xy-plane (the transverse plane), as shown in the right picture below. However, in this 2D setup, we are drawing the cross-section of the chip with the grating in the xy-plane. Therefore, the TE mode in FDTD is actually polarized perpendicular to the chip surface, and the TM mode is defined in the transverse plane of the input waveguide.

To simulate the TE mode as shown in the left 3D diagram below, we need to inject the fundamental TM mode of the waveguide in this 2D simulation, given the current setup. If you refer to the example of grating coupler design on Lumerical's website ([websitehttps://optics.ansys.com/hc/en-us/articles/360042305334-Grating-coupler](https://optics.ansys.com/hc/en-us/articles/360042305334-Grating-coupler)), you'll notice that they switched the mode selection from TM to TE when transitioning from a 2D setup with a cross-section in the xy-plane to a 3D setup with a top view in the xy-plane, in order to correctly simulate the TE mode.







Monitor the Simulation

1. We will use multiple monitors to extract information from the simulation. The first one, *slice*, will collect a 1-D view of the light scattering out of the grating. The second monitor, *profile*, will give a 2-D profile of the system. A third monitor called *through* will capture the light still propagating in the waveguide, and a fourth called *leakage* will measure the light scattered through the substrate.

```
addpower;
set('name', 'slice');
set('monitor type', 'linear x');
set('x', 0);
set('x span', 200e-6);
set('y', 0.5e-6);
set('z', 0);

addpower;
set('name', 'profile');
set('monitor type', '2D z-normal');
set('x', 0);
set('x span', 200e-6);
set('y', 0);
set('y span', 200e-6);
set('z', 0);

addpower;
set('name', 'through');
set('monitor type', 'linear Y');
set('x', x0 + 1.25e-6);
set('y', 0);
set('y span', 0.22e-6+cladding/2);
set('z', 0);

addpower;
set('name', 'leakage');
set('monitor type', 'linear x');
set('x', 0);
set('x span', 200e-6);
set('y', -cladding-1.25e-6);
set('z', 0);
```

Define the Simulation Region

1. Insert an FDTD region. We want to minimize the size of the region to reduce computation, while keeping it large enough to be accurate. In this case, the FDTD size will be a function of the cladding and the pitch.

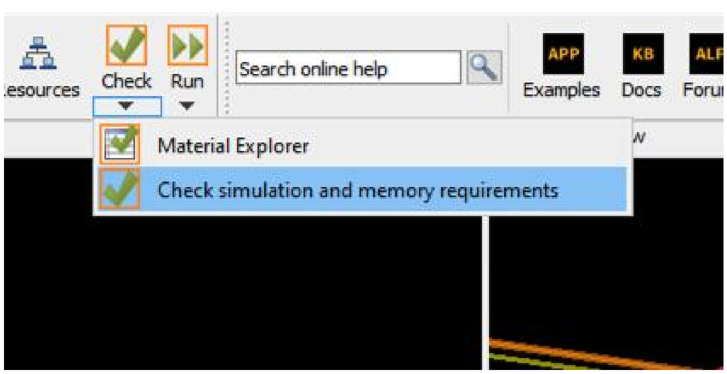
```
fdtd_xmin = -5e-6;
fdtd_xmax = x0+2.5e-6;
fdtd_ymin = -cladding-2.5e-6;
fdtd_ymax = 2.5e-6;
fdtd_z = 0;

addfdtd;
set('dimension', '2D');
set('auto shutoff min', 1e-4);
set('x min', fdtd_xmin);
set('x max', fdtd_xmax);
set('y min', fdtd_ymin);
set('y max', fdtd_ymax);
set('z', 0);
```

2. Next, add a simulation mesh. We want high accuracy around the scattering region – i.e. the grating teeth. We will make a uniform mesh of size λ/40 centered around the grating coupler (in a real simulation, you should run convergence tests for this mesh density).

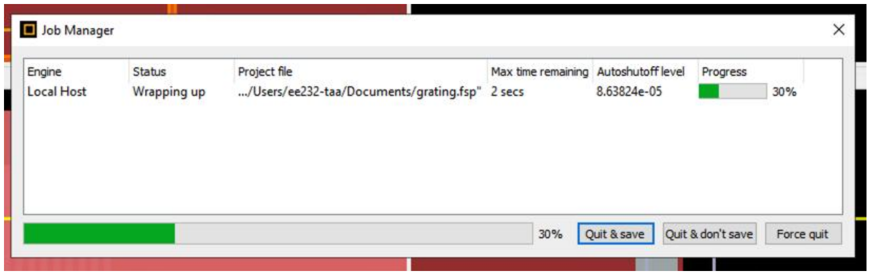
```
addmesh;
set('dx', wavelength/40);
set('dy', wavelength/40);
set('dz', wavelength/40);
set('x', 0);
set('x span', 200e-6);
set('y min', -0.25e-6);
set('y max', 0.22e-6+0.25e-6);
set('z', 0);
```

Run the Simulation

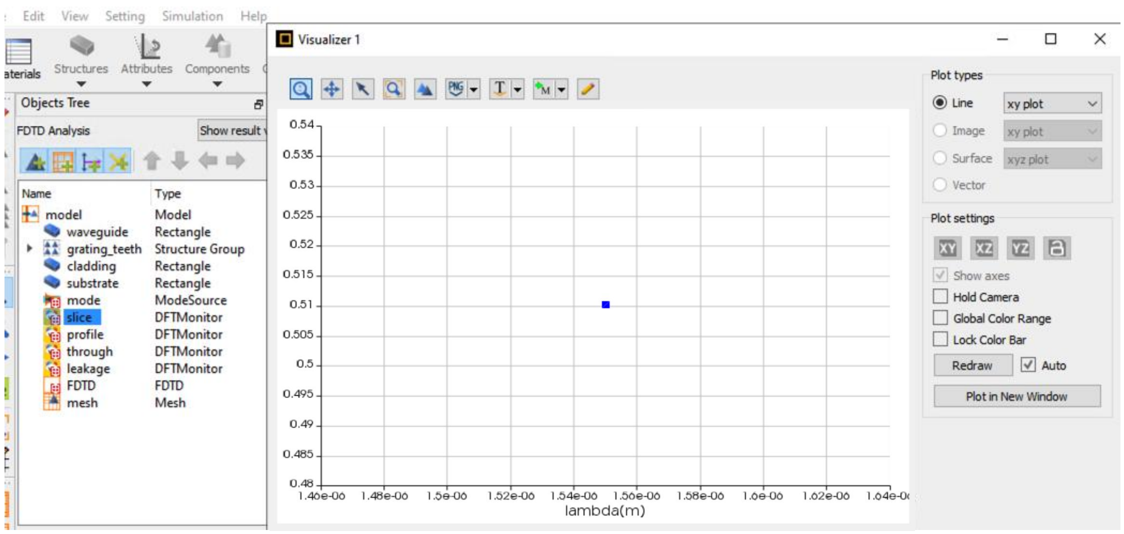
1. Check the simulation memory requirements to ensure the setup is correct. A simulation of this size should have around the order of ~100's of megabytes.



2. Run the simulation. Be sure to monitor the auto shutoff level to verify it approaches 10^{-4}, the end condition we previously specified.



3. Visualize the 'directionality' of the grating, or the transmission of the power scattered upward. This is one of the key figures of merit of a grating coupler.



Follow-up Questions:

1. Examine the transmission of the other monitors (*through*, *leakage*). In this scenario, they are both approximately equal to that of *slice* – i.e. light is uniformly scattered upward, scattered downward, and passed through. What are some ways we can force the directionality upward?
    a. There are many ways to do this – let's try to think of at least 3+
2. Examine the power vs position plot (P) of *slice*. What does the shape look like (hint – what kind of function)? What does this tell us about our grating?
3. Check out the farfield of the *slice* monitor. This is the angular performance of the grating at a long distance – i.e. its emission angle. We see sidelobes in the angular spectrum. Why does this occur, and how can we mitigate this?

Reference

[1] "Grating Coupler." *Ansys Optics*, Ansys, https://optics.ansys.com/hc/en-us/articles/360042305334-Grating-coupler.