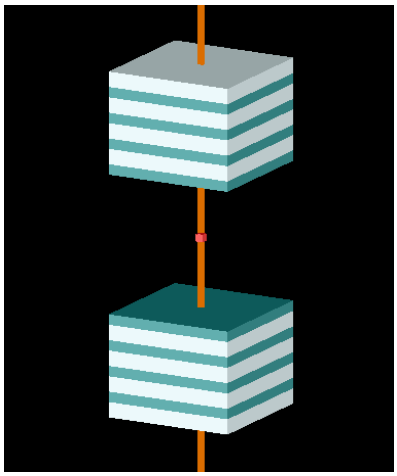
Maria Florencia Nardone
July 25, 2025
Script Report

Acknowledgements:
My script is an adaptation of a script provided to me by one of the graduate students in my lab (Benedict S. Morris). I received help from him as well as from Luis Santos and guidance from Professeur Bruno Palpant to write this script.

1.0 Introduction



This Lumerical FDTD script models a hybrid optical cavity that contains a gold rectangular nanorod (AuNR) placed in a defect layer of Alumina, embedded in a multi-layer Bragg reflector. My goal is to probe how the cavity's geometry influences light–matter coupling, with special attention to the relationship between coupling strength and cavity order (the number of standing-wave antinodes spanned by the defect layer). Additionally, this simulation can characterize a variety of other properties observed such as the interaction of the E-field on the nanorod, changes in the background index of the sample, etc.

1.1 Parameters of interest

Symbol	Meaning	Sweep Range (Array)
Gap	Edge-to-edge spacing between adjacent nanorods	`sweep`
Length	Total length of the AuNR	`sweep2`
Order	Defect-layer height expressed as an integer number of antinodes (cavity order)	`sweep3`

1.2 Key Outputs

The script can generate a large number of varying plots and datasets. Some of the key outputs include:

- **Optical spectra**: Absorption (A), Reflection (R), and Transmission (T) for every parameter triplet.
- **Coupling strength**: Extracted as the Rabi splitting energy between hybridized absorption peaks.
- **Field profiles**: Spatial maps of E intensity with respect to nanorod length

- **Index sensitivity**: Shift in resonance wavelengths versus changes in background refractive index

2.0 Script Initializing and Configuration

2.1 Function Library

The script imports `myfunction_library1_withSomeTweaks`, which houses helper routines such as `BuildSimulationDomain`, `BuildAuNR`, and `BuildCavity`. These user-defined functions set materials, geometry, boundary conditions, etc. thus keeping the main file readable.

2.2 Sweep Parameter Definiton

Three sets of arrays are built and specify the values to be swept.

- `sweep = {...}` – nanorod **gap** (in meters).
- `sweep2 = {...}` – nanorod **length** (in meters).
- `sweep3 = {...}` – integer cavity orders

2.3 Control Flags

Several boolean flags (set to `1` for ON, `0` for OFF) are defined to control what the script executes. This helps cut down on the running of programs that are not useful to you and thus also decreases the processing time.

Flags	Description
`BuildAuNRFlag`	Creates gold nanorod
`BuildCavityFlag`	Adds Bragg mirrors, generating hybrid cavity Note: for the script I wrote this must always be ON
`flagexporttextfiles`	Boolean check if exporting files for A,R,T or not
`flagsavefsp`	Saving data as fsp files. Note: these are very large and aren't suggested unless you really need them
`flagrelativevariation`	Slightly increases background index of whole simulation
`flagstationarycalculation`	Stationary measurements at resonance to determine Electric fields
`flagtimecalculation`	Note: only runs if is `flagstationarycalculation` also on Adds time components to E-field data

2.4 Data Storage Initialization

Before the sweep begins, the script pre-allocates:

- **Matrices** (e.g., `maxAmat, CS`) for scalar metrics such as peak absorption or computed coupling strength.
- **Nested Cell Arrays** (e.g., `Tmat, Amat, Lambdamat`) are used to store entire data vectors, such as a full spectrum over all wavelengths. The script uses a nested cell structure (`Tmat{sp3}{sp2}{sp}`) to create a 3D data container that map directly to the three swept parameters (cavity order, length, and gap).

3.0 Main Simulation Workflow

The heart of the program is a set of three nested for loops. This structure automates the process of simulating every possible combination of the geometric parameters defined in the initialization stage.

3.1 Three-Level Loop Structure

The simulation iterates through the parameters in a specific hierarchical order:

1. Outer loop (`sp3`) – **Cavity order.**
   Each pass fixes the defect-layer height (an integer number of antinodes) before any other parameter changes.
2. Middle loop (`sp2`) – **Nanorod length.**
   Within a given cavity order, the script sweeps every rod length in the array
3. Inner loop (`sp`) – **Gap.**
   Finally, it scans the edge-to-edge spacing between adjacent rods.

This ordering guarantees that for every cavity order and rod length, all gap values are evaluated.

3.2 Dynamic Folder Managemt

In order to organize the output, the code creates hierarchical folder structure for storing results, similar to the structure of the nested for loops. At the beginning of each loop, a `system("if not exist\" + ...")` command is executed to create a new directory if it doesn't already exist. This results in a clear and navigable file path for each simulation run, formatted as: `/CavityOrder_X/Plots_Length_Ynm/GapZnm/`. Grouping files this way makes it easy to track down.

4.0 Simulation Sequence (per iteration)

Each pass through the triple-nested loop runs two main stages. First it locates the nanorod’s

bare resonance, then models the the full hybrid cavity at zero detuning, before extracting information.

4.1 Stage 1: Nanorod Resonance Identification

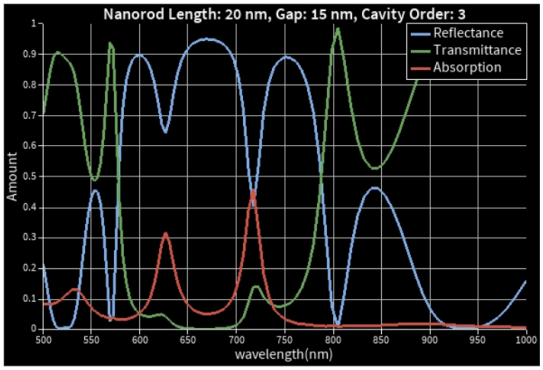
The script builds a bare cavity using the `BuildSimulationDomain` and `BuildAuNR` functions. This creates a gold nanorod inside an Alumina defect layer.It then runs an FDTD Simulation and gets data from the reflection and transmission monitors. From the R and T we can get the absorption spectrum. It then locates the peak in the peak (the wavelength of maximum absorption) and stores it in resLambdaA variable, which represents the resonance wavelength of the bare cavity.

4.2 Stage 2: Hybrid Cavity Simulation

With the nanorod's resonance identified, the script simulates the full hybrid cavity system**.**

The code calls the BuildCavity function, and passes the resLambdaA value it had previously found, as an input. This tunes the Bragg cavity to match with the calculated resonance wavelength in order to be at zero detuning and thus be able to observe strong coupling.

A second FDTD simulation is then run on the completed hybrid cavity. The final reflection (`R_final`), transmission (`T_final`), and absorption (`A_final`) spectra are collected and used for calcualtion. The full spectra and corresponding wavelengths are stored in the main 3D cell arrays (`Rmat`, `Tmat`, `Amat`, `Lambdamat`) for later analysis. A plot of these spectra is also immediately saved to the corresponding results' folder.



Example RTA Graph produced from one of the simulation runs. Specifically for Cavity Order: 3, Nanorod Length: 20 nm and Gap: 15 nm.

4.3 Coupling Strength Calculation

Immediately after the main simulation (still within the loop), an analysis is performed on the final absorption spectrum to quantify the interaction strength.

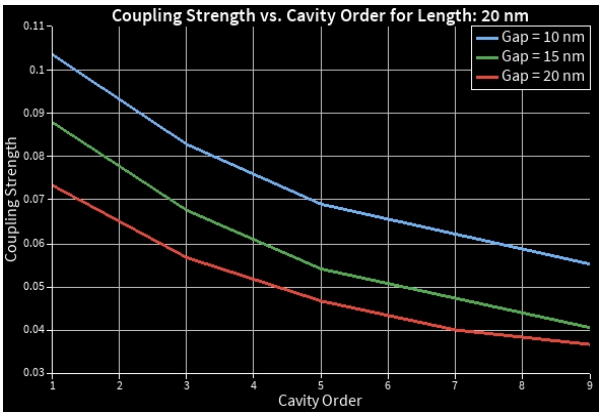
1. **Peak Finding:** The script analyzes the absorption spectrum `A_final` to search for the two most prominent peaks within set boundaries. These two split peaks are characteristic of strong coupline known as Rabi splitting.

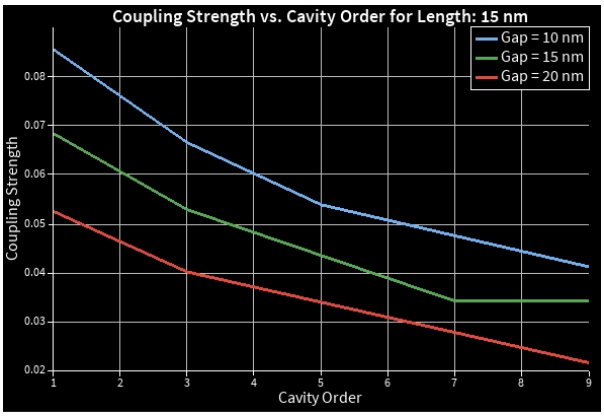
   The peak finding algorithm has a left and right bound, which encompass the photonic band gap (plus or minus a certain amount from the resonance wavelength). Moreover, a tolerance check (`if (A_final_cut(pos(i)) > 0.135)`) is implemented. This ensures that both identified peaks have a significant amplitude and are not simply noise, which is vital for distinguishing true peak splitting from a noisy single peak in weak-coupling scenarios.

2. **Calculation:**

- If two valid peaks are found, their wavelengths are converted to energy (in eV). The energy splitting (EVDif) between them is calculated. This value is then used to determine the normalized **Coupling Strength (`CS`)**.
- If the validation condition fails (and the two peaks found are not above the tolerance), the coupling strength **(`CS`)** is set to `0`. As well as printing out a “Warning” correctly identifying the result as a probable weak coupling case.

Some Pictures:





5.0 Conditional Analysis

The script includes several optional routines that can be activated using the control flags defined during initialization. These include E-field analysis WAWAAA

5.1 Index Background (`flagrelativevariation == 1`)

Purpose: Gauge how a small change in the surrounding refractive index (+ 0.01) perturbs the spectra.

When this flag is active, the script performs an additional FDTD simulation for each parameter set. In this run, the background refractive index is slightly increased (indexbackground + 0.01). The script then compares the resulting spectra with the original ones, calculating the maximum variation in absorption, reflection, and transmission. It also tracks the shift in the resonance wavelength.

5.2 E-Field Calculation (`flagstationarycalculation` & `flagtimecalculation`)

1. **E-field Profile (`flagstationarycalculation = 1`).**

Purpose: To analyze spatial distribution and enhancement of the electric field at resonance

This significantly modifies the simulation setup and replaces the source with a stationary, clean, single-frequency, phase-coherent standing wave at resonance. Monitors are then added to record the electric field components. After the simulation, the script calculates the electric field intensity and plots them with respect to the length of the nanorod in order to visualize how the E-field interacts.

Note: the Original vs Perturbed Resonance Wavelengths graphs are not correct. They are incorrectly comparing the resonance wavelength of the bare cavity to that of the hybrid cavity with the increased background index

2. **Time-Domain E-field (**`flagstationarycalculation == 1` and `flagtimecalculation == 1`**)**

Purpose: To observe the E-field change with respect to time

Note: This section of the code needs more working on. It has issues and does not run as expected. The information is not being stored in the matrices correctly and does not want to run. I chose to move on from resolving this issue as it was not part of my main pursuit (observing changes to coupling strength based on different parameters).

6.0 Post-Simulation Data Calculations and Visualization

After all the sweeps are completed and all the data has been collected/stored, the code's main focus becomes to analyze and create visualizations for collected data. The plots created here are used for interpreting the results, identifying trends, and drawing physical conclusions.

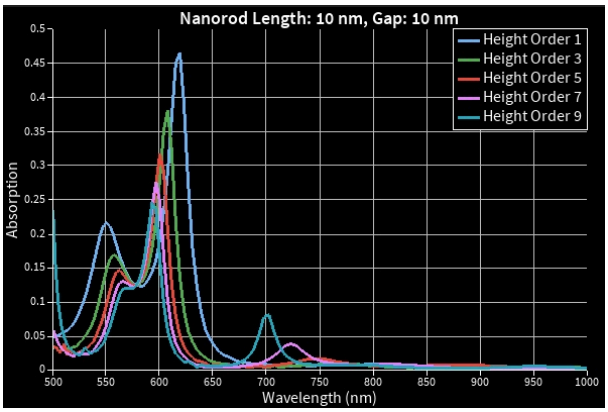
Moreover, the script within these plotting algorithms loops through the stored data systematically (like the nested for loops), extracting the relevant vectors for potting. This design allows multiple data sets to be overlaid on the same x and y azis for easy comparison and interpretation.

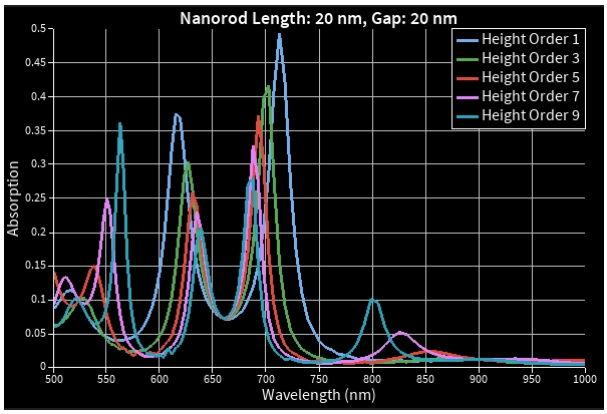
The script systematically loops through the stored data to create plots summarizing the relationships between the geometric parameters and the system's optical response.

6.1 Key Plots generated in this part:

- Maxs (MaxR, MinT, MaxA & EAuNR) vs. Gap
- MaxA vs. Gap
- Absoption vs. Cavity Order
- Max RV vs. Gap (Note: only if `flagrelativevariation == 1`)
- Resoance Wavelength Comparison (Note: only if `flagrelativevariation == 1`)
- Resonance Wavelength vs. Gap (Note: only if `flagrelativevariation == 0`)

Some Pictures (specifically the Absoption vs. Cavity Order Plot):





7.0 Final Data Export

Here, the code re-uses the three-level nested loop structure to iterate through every parameter combination in order to ensure that the data is saved in a structured manner. The data is aggregated and then exported as a txt file for later analysis. It is stored in the same hierarchical folders made previously so that the raw data is stored with its corresponding plots.

Note: Right now it is only exporting optical data such as the RTA data as well as the E-field data. Including extra matrices the user would like to export to txt, such as the coupling strength matrix, would be an easy addition to the code in this section.