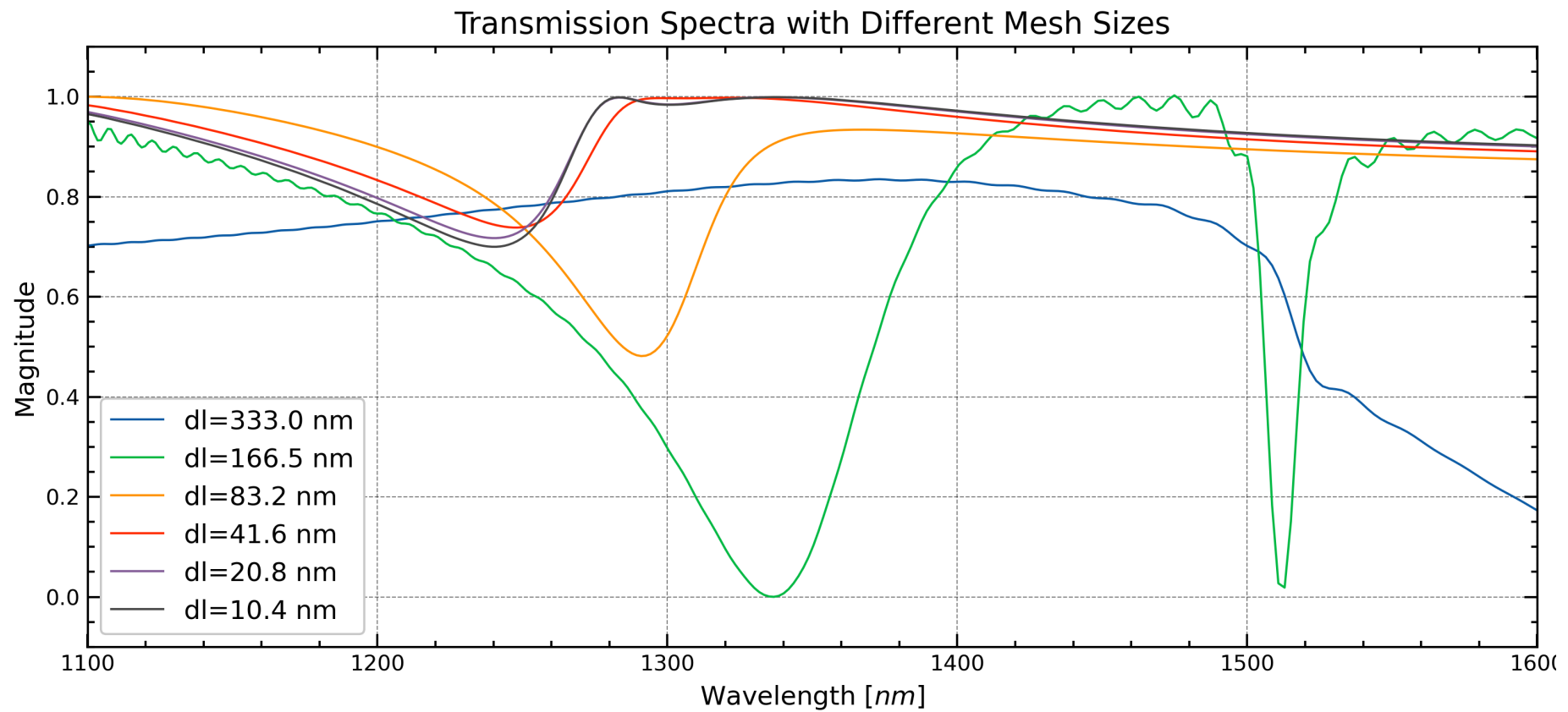


# Mesh Study of Huygen's Surface

Yankun (Alex) Meng



## Mesh-Study Results

# Initialization

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import tidy3d as td
4 import tidy3d.web as web
5 import scienceplots
```

```
1 # Set logging level to ERROR to reduce output verbosity
2 td.config.logging_level = "ERROR"
```

```
1 # 0 Define a FreqRange object with desired wavelengths
2 fr = td.FreqRange.from_wvl_interval(wvl_min=1.1, wvl_max=1.6)
3 N = 301 # num_points
```

```
1 # 1 Computational domain size
2 h = 0.220 # Height of cylinder
3 spc = 2
4 Lz = spc + h + h + spc
5
6 Px = Py = P = 0.666 # periodicity
7 sim_size = [Px, Py, Lz]
```

```
1 # 3 Structures
2 r = 0.242 # radius of the cylinder
3 n_Si = 3.5
4 Si = td.Medium(permittivity=n_Si**2, name='Si')
5 cylinder = td.Structure(
6     geometry=td.Cylinder(center=[0, 0, h / 2], radius=r, length=h, axis=2), medium=Si
7 )
```



# Mesh Study Loop Assignment

```
1 dls = [P/2, P/4, P/8, P/16, P/32, P/64] # mesh study list
2 sims = {}
```

```
1 for i, dl in enumerate(dls):
2     # 2 Grid Specifications
3     horizontal_grid = td.UniformGrid(dl=dl)
4     vertical_grid = td.AutoGrid(min_steps_per_wvl=32)
5     grid_spec=td.GridSpec(
6         grid_x=horizontal_grid,
7         grid_y=horizontal_grid,
8         grid_z=vertical_grid,
9     )
10
11     # 4 Sources
12     source = td.PlaneWave(
13         source_time=fr.to_gaussian_pulse(),
14         size=(td.inf, td.inf, 0),
15         center=(0, 0, Lz/2 - spc + 2 * dl),
16         direction="-",
17         pol_angle=0
18     )
19
20     # 5 Monitor
21     monitor = td.FluxMonitor(
22         center=(0, 0, -Lz/2 + spc - 2*dl),
23         size=(td.inf, td.inf, 0),
24         freqs=fr.freqs(N),
```



# Plotting

```
1 # this uses scienceplots to make plots look better
2 plt.style.use(['science', 'notebook', 'grid'])
```

```
1 x = td.C_0 / fr.freqs(N) * 1000
2 Ts = []
3 for i in range(len(dls)):
4     Ts.append(batch_data[f"actual{i}"]["flux"].flux / batch_data[f"norm{i}"]["flux"].flux)
```

```
1 plt.figure(figsize=(12, 5))
2 for i, T in enumerate(Ts):
3     plt.plot(x, T, "-", lw=1, label=f"dl={dls[i] * 1000:.1f} nm")
4 plt.xlabel(r"Wavelength [$nm$]")
5 plt.ylabel("Magnitude")
6 plt.ylim(-0.1, 1.1)
7 plt.legend(fontsize=12)
8 plt.tick_params(axis='both', labelsize=10) # change tick label size to 10
9 plt.title("Transmission Spectra with Different Mesh Sizes", fontsize=14)
10 plt.show()
```

