Mesh Study of Huygen’s Surface

Yankun (Alex) Meng

```python
import matplotlib.pyplot as plt
import numpy as np
import tidy3d as td
import tidy3d.web as web
import scienceplots
```

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

Initialization

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

```python
# 1 Computational domain size
h = 0.220  # Height of cylinder
spc = 2
Lz = spc + h + h + spc

Px = Py = P = 0.666 # periodicity
sim_size = [Px, Py, Lz]
```

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

# Spin_on glass + substrate
n_glass = 1.4
n_SiO2 = 1.45
glass = td.Medium(permittivity=n_glass**2, name='glass')
SiO2 = td.Medium(permittivity=n_SiO2**2, name='oxide')

substrate = td.Structure(
```

Mesh Study Loop Assignment

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

```python
for i, dl in enumerate(dls):
    # 2 Grid Specifications
    horizontal_grid = td.UniformGrid(dl=dl)
    vertical_grid = td.AutoGrid(min_steps_per_wvl=32)
    grid_spec=td.GridSpec(
        grid_x=horizontal_grid,
        grid_y=horizontal_grid,
        grid_z=vertical_grid,
    )

    # 4 Sources
    source = td.PlaneWave(
        source_time=fr.to_gaussian_pulse(),
        size=(td.inf, td.inf, 0),
        center=(0, 0, Lz/2 - spc + 2 * dl),
        direction="-",
        pol_angle=0
    )

    # 5 Monitor
    monitor = td.FluxMonitor(
        center=(0, 0, -Lz/2 + spc - 2*dl),
        size=(td.inf, td.inf, 0),
        freqs=fr.freqs(N),
```

Plotting

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

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

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